Data Mining

ALGORITMA & IMPLEMENTASI

Anjar Wanto • Muhammad Noor Hasan Siregar
Agus Perdana Windarto • Dedy Hartama
Ni Luh Wiwik Sri Rahayu Ginantra • Darmawan Napitupulu
Edi Surya Negara • Muhammad Ridwan Lubis
Sarini Vita Dewi • Cahyo Prianto

Data Mining : Algoritma dan Implementasi

UU 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Perlindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

a. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
b. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
c. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
d. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).

Data Mining : Algoritma dan Implementasi

Penulis:
Anjar Wanto, Muhammad Noor Hasan Siregar, Agus Perdana Windarto,
Dedy Hartama, Ni Luh Wiwik Sri Rahayu Ginantra,
Darmawan Napitupulu, Edi Surya Negara, Muhammad Ridwan Lubis,
Sarini Vita Dewi, Cahyo Prianto

Penerbit Yayasan Kita Menulis

Data Mining : Algoritma dan Implementasi

Penulis:
Anjar Wanto, Muhammad Noor Hasan Siregar
Agus Perdana Windarto, Dedy Hartama
Ni Luh Wiwik Sri Rahayu Ginantra, Darmawan Napitupulu
Edi Surya Negara, Muhammad Ridwan Lubis
Sarini Vita Dewi, Cahyo Prianto

Editor: Tonni Limbong
Desain Cover: Tim Kreatif Kita Menulis

Penerbit
Yayasan Kita Menulis
Web: kitamenulis.id
e-mail: press@kitamenulis.id

Anjar Wanto, dkk.
Data Mining : Algoritma dan Implementasi
Yayasan Kita Menulis, 2020
xiv; 168 hlm; 16 x 23 cm
ISBN: 978-623-7645-79-5 (print)
E-ISBN: 978-623-7645-80-1 (online)
Cetakan 1, April 2020
I. Data Mining : Algoritma dan Implementasi
II. Yayasan Kita Menulis

Katalog Dalam Terbitan

Kata Pengantar

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan taufik dan hidayahnya, sehingga kami mampu menyelesaikan buku Data Mining : Algoritma dan Implementasi ini.

Data mining dapat diterapkan untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Terdapat beberapa teknik yang digunakan dalam data mining, salah satu teknik data mining adalah clustering. Terdapat dua jenis metode clustering yang digunakan dalam pengelompokkan data, yaitu *hierarchical clustering* dan *non-hierarchical clustering*.

Buku ini terdiri dari 10 (sepuluh) bab, yaitu :

Bab 1 Pengelompokkan Data dengan Algoritma K-Means
Bab 2 Pengelompokan Data dengan Algoritma K-Medoids
Bab 3 Asosiasi Data Mining dengan Algoritma A Priori
Bab 4 Pengklasifikasian Data dengan Algoritma C4.5
Bab 5 Klasifikasi Citra dengan K-NN
Bab 6 Penerapan Data Mining dengan Particle Swarm Optimization dan Decision Tree C4.5
Bab 7 Klasifikasi Data Menggunakan Algoritma Naive Bayes
Bab 8 Implementasi Data Mining dengan Regresi Linear Berganda
Bab 9 Performa klasifikasi Dataset dengan metode Correlation Based Feature Selection (CFS)
Bab 10 Text Mining : Twitter Analysis

Akhir kata penulis mengucapkan banyak terima kasih kepada teman-teman sejawat yang telah memberikan masukan-masukan positif selama penulisan buku ini.

Medan, Maret 2020

Penulis

Daftar Isi

Daftar Gambar

Daftar Tabel

Bab 1

Pengelompokkan Data dengan Algoritma K-Means

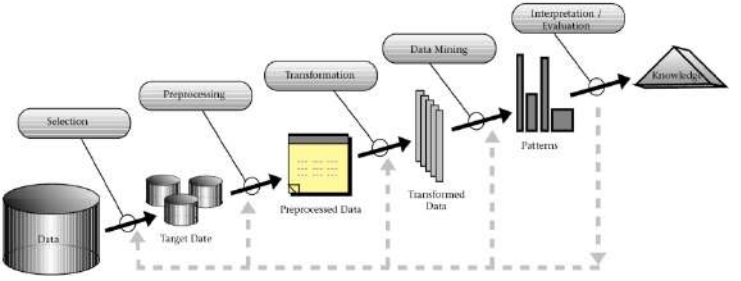
1.1 Pendahuluan

Istilah data mining sering digunakan sebagai sinonim untuk proses menemukan pola-pola yang berwawasan, menarik, dan baru, serta model deskriptif, dapat dipahami, dan prediktif dari data skala besar (Zaki and Jr, 2014). Istilah data mining telah banyak digunakan oleh ahli statistik, analis data, dan komunitas Sistem Informasi Manajemen (SIM) (Fayyad and Stolorz, 1997). Tan, Steinbach and Kumar (2006) mendefinisikan data mining sebagai proses untuk mendapatkan informasi yang berguna dari gudang basis data yang besar, yang dapat juga diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan (Knowledge Discovery). Menurut Sudirman, Windarto and Wanto (2018), data mining merupakan proses yang menggunakan berbagai teknik dan alat analisis data untuk menemukan hubungan dan pola yang tersembunyi. Pendekatan dasar dalam data mining adalah untuk meringkas data dan untuk mengekstrak informasi berguna yang masuk akal dan sebelumnya tidak diketahui.

Data Mining dapat menemukan tren dan pola tersembunyi yang tidak muncul dalam analisis kueri sederhana sehingga dapat memiliki bagian penting dalam hal menemukan pengetahuan dan membuat keputusan. Tugas-tugas semacam itu dapat bersifat prediksi seperti klasifikasi dan regresi atau deskriptif seperti *Clustering* dan asosiasi (Hemeida *et al.*, 2019). Karena itu Data Mining

sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (artificial intelligent), *machine learning*, statistik dan basis Data (Aprilla *et al.*, 2013). Data mining perlu dipelajari dan dipahami, karena manusia menghasilkan banyak sekali data yang sangat besar baik dalam bidang bisnis, kedokteran, cuaca, olahraga, politik dan sebagainya (Nofriansyah and Nurcahyo, 2015).

Sebagai suatu rangkaian proses, data mining dapat dibagi menjadi beberapa tahap yang dapat dilihat pada gambar 1.1. Tahap-tahap tersebut bersifat interaktif di mana pemakai terlibat langsung atau dengan perantaraan knowledge base.



Gambar 1.1: Tahapan Data Mining (Fayyad, Piatetsky-Shapiro and Smyth, 1996)

Ada beberapa tugas yang dapat dilakukan oleh Data Mining dalam proses pemecahan masalah dan pencarian pengetahuan baru (Larose, 2005; Han and Kamber, 2006; Witten, Frank and Hall, 2011), di antaranya adalah sebagai berikut:

1. Klastering (*Clustering*)
   Digunakan untuk mengelompokan atau mengidentifikasi data yang memiliki karakteristik tertentu. Contoh algoritma : *K-Means*, K-Medoids, dan lain-lain.
2. Klasifikasi (*Classification*)
   Digunakan untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Contoh algoritma : C4.5, K-Nearest Neighbor (KNN), dan lain-lain.
3. Asosiasi (*Association*)
   Digunakan untuk mengatasi masalah bisnis yang khas, yakni dengan menganalisa tabel transaksi penjualan dang mengidentifikasi produk-

produk yang seringkali dibeli bersamaan oleh customer, misalnya apabila orang membeli sambal, biasanya juga dia membeli kecap.
Contoh Algoritma : Apriori, Frequent Pattern Growth (FP- Growth), dan lain-lain.

4. Estimasi (*Estimation*)
   Digunakan untuk memperkirakan atau menilai sesuatu hal yang belum pernah ada sebelumnya yang disajikan dalam bentuk hasil kuantitatif (angka).
   Contoh algoritma : Regresi Linier, Confidence Interval Estimations, dan lain-lain.
5. Prediksi (*Predictions*)
   Digunakan untuk memperkirakan atau meramalkan suatu kejadian yang belum pernah terjadi.
   Contoh algoritma : Decision Tree, K-Nearest Neighbor (KNN), dan lain-lain.

Pada buku ini, akan dibahas beberapa algoritma (metode) data mining yang sering digunakan dalam kehidupan sehari-hari beserta contoh kasus dan perhitungannya, diantaranya : *K-Means*, *K-Medoids*, *Apriori*, *C4.5*, *K-Nearest Neighbor (KNN)*, *Naive Bayes*, Regresi Berganda (*Multiple Regression*), *Corelationbased Feature Selection* (CFS) dan *text mining* dengan *Lexicon Based*. Sedangkan untuk bab ini akan membahas pengelompokkan data dengan algoritma *K-Means*.

1.2 Algoritma *K-Means*

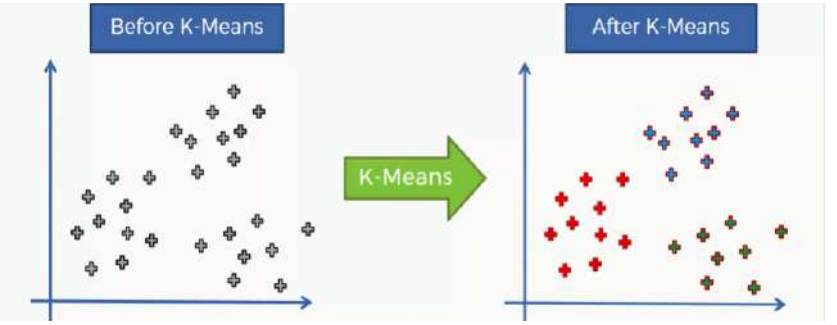
Algoritma *K-Means* ditemukan oleh beberapa orang yaitu Lloyd (1957), Forgey (1965), Friedman dan Rubin (1967), dan McQueen (1967). Ide dari pengelompokkan (*Clustering*) pertama kali ditemukan oleh Lloyd pada tahun 1957, namun hal tersebut baru dipublikasi pada tahun 1982. Pada tahun 1965 Forgey juga mempublikasikan teknik yang sama sehingga terkadang dikenal sebagai Lloyd-Forgy (Primartha, 2018).

K-Means merupakan salah satu algoritma *Clustering* yang masuk dalam kelompok *Unsupervised learning* yang digunakan untuk membagi data menjadi beberapa kelompok dengan sistem partisi. Algoritma ini menerima masukan berupa data tanpa label kelas. Hali ini berbeda dengan *K-Nearest*

Neighbor (KNN) dan algoritma *supervised learning* lainnya yang menerima masukan berupa vektor. Pada algoritma *K-Means*, komputer mengelompokkan sendiri data-data yang menjadi masukannya tanpa mengetahui terlebih dahulu target kelasnya. Masukan yang diterima adalah data atau objek dan *k* buah kelompok (*cluster*) yang diinginkan. Algoritma ini akan mengelompokkan data atau objek kedalam k buah kelompok tersebut.

Pada setiap *cluster* terdapat titik pusat (*Centroid*) yang mempresentasikan *cluster* tersebut. Secara sederhana algoritma *K-Means* dapat dijelaskan sebagai algoritma data mining yang digunakan untuk menyelesaikan masalah pengelompokkan (*Clustering*). Pada pemrosesan data algoritma *K-Means Clustering*, akan diawali dengan pengelompokan *Centroid* pertama yang dipilih secara acak sebagai titik awal untuk setiap *cluster*, kemudian menghitung secara berulang agar posisi *Centroid* optimal.

Pada dasarnya algoritma *K-Means* hanya mengambil sebagian dari banyaknya komponen yang didapatkan untuk kemudian dijadikan pusat *cluster* awal, pada penentuan pusat *cluster* ini dipilih secara acak dari populasi data. Kemudian algoritma *K-Means* akan menguji masing-masing dari setiap komponen dalam populasi data tersebut dan menandai komponen tersebut ke dalam salah satu pusat *cluster* yang telah didefinisikan sebelumnya tergantung dari jarak minimum antar komponen dengan tiap-tiap pusat *cluster*. Selanjutnya posisi pusat *cluster* akan dihitung kembali hingga semua komponen data digolongkan ke dalam tiap-tiap *cluster* dan terakhir akan terbentuk *cluster* baru (Sihombing, 2017).

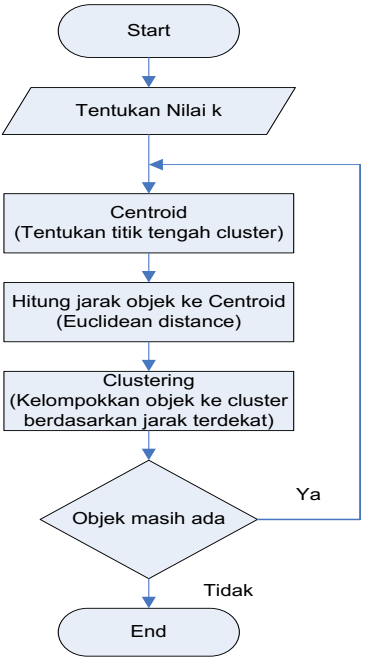


Gambar 1.2: *K-Means Clustering*

1.3 Konsep *K-Means*

Terdapat dua jenis data *Clustering* yang sering dipergunakan dalam proses pengelompokan data yaitu *Hierarchical* dan *Non-Hierarchical*. *K-Means* merupakan salah satu metode data *Clustering* non-hierarchical atau Partitional *Clustering*. Algoritma *K-Means* berusaha mengelompokkan data yang ada ke dalam beberapa kelompok, di mana data dalam satu kelompok mempunyai karakteristik yang sama antara satu dengan yang lainnya dan mempunyai karakteristik yang berbeda dengan data yang ada di dalam kelompok yang lain.

Berikut ini akan disajikan Diagram alir dari algoritma *Clustering K-Means*.



Gambar 1.3: Diagram alir dari algoritma *Clustering K-Means* (Younus *et al.*, 2015)

Langkah-langkah algoritma *K-Means* dapat dijelaskan sebagai berikut (Larose, 2005; Prasetyo, 2012; Khotimah, 2014; Parlina *et al.*, 2018; Primartha, 2018):

1. Tentukan jumlah *cluster* (k) pada data set.
2. Tentukan nilai pusat (*Centroid*).

Penentuan nilai *Centroid* pada tahap awal dilakukan secara *random*, sedangkan pada tahap iterasi digunakan rumus seperti pada persamaan (1) berikut ini.

$V_{ij} = \frac{1}{Ni} \sum_{k=0}^{Ni} Xkj$ .....(1)

Keterangan :

V_{ij} = *Centroid* rata-rata *cluster* ke-I untuk variabel ke-j
N_i = Jumlah anggota *cluster* ke-i
i, *k* = Indeks dari *cluster*
j = Indeks dari variabel
Xkj = nilai data ke-k variabel ke-j untuk *cluster* tersebut

3. Pada masing-masing *record*, hitung jarak terdekat dengan *Centroid.* Ada beberapa cara yang dapat digunakan untuk mengukur jarak data ke pusat kelompok, diantaranya *Euclidean*, *Manhattan/City Block*, dan *Minkowsky*. Setiap cara memiliki kelebihan dan kekurangan masing-masing. Untuk penulisan pada bab ini, jarak *Centroid* yang digunakan adalah *Euclidean Distance,* dengan rumus seperti dibawah ini:

$De = \sqrt{(xi - si)^2 + (yi - ti)^2}$.....(2)

Keterangan :

De = *Euclidean Distance*
i = Banyaknya objek [2]
(x, y) = Koordinat objek
(s, t) = Koordinat *Centroid*

4. Kelompokkan objek berdasarkan jarak ke *Centroid* terdekat
5. Ulangi langkah ke-3 hingga langkah ke-4, lakukan *iterasi* hingga *Centroid* bernilai optimal.

1.4 Karakteristik *K-Means*

Karakteristik *K-Means* dapat diringkas seperti berikut:

1. *K-Means* merupakan salah satu metode pengelompokkan sederhana yang dapat digunakan dengan mudah dan sangat cepat dalam proses *Clustering*.
2. Pada jenis dataset tertentu *K-Means* tidak dapat melakukan segmentasi data dengan baik, di mana hasil segmentasinya tidak

dapat memberikan pola kelompok yang mewakili karakteristik bentuk alami data.
3. *K-Means* bisa mengalami masalah ketika mengelompokkan data yang mengandung outlier.
4. *K-Means* sangat sensitif pada pembangkitan *Centroid* awal secara random
5. Memungkinkan suatu *cluster* tidak mempunyai anggota.
6. Hasil *Clustering* dengan *K-Means* bersifat tidak unik (selalu berubah-ubah) – terkadang baik, terkadang jelek
7. *K-Means* sangat sulit untuk mencapai global optimum.

Memperhatikan input dalam algoritma *K-Means*, dapat dikatakan bahwa algoritma ini hanya mengolah data kuantitatif atau numerik. Sebuah basis data tidak mungkin hanya berisi satu macam tipe data saja, akan tetapi beragam tipe. Sebuah basis data dapat beris data data dengan tipe seperti: binary, nominal, ordinal, interval dan ratio. Berbagai macam atribut dalam basis data yang berbeda tipe disebut sebagai data multivariate. Tipe data seperti nominal dan ordinal harus diolah terlebih dahulu menjadi data numerik (bisa dilakukan dengan cara diskritisasi), sehingga dapat diberlakukan algoritma *K-Means* dalam pembentukan *cluster* nya.

1.5 Contoh Kasus Pengelompokkan Data dengan *K-Means*

Pada pembahasan ini akan diberikan contoh kasus dan penjelasan mengenai pengelompokkan data dengan menggunakan algoritma *K-Means*.

1.5.1 Contoh Kasus dan cara Penyelesaian

Contoh Kasus:

Jumlah wisatawan mancanegara yang berkunjung ke Indonesia setiap tahun bervariasi dan tidak sama. Para wisatawan yang hadir sebagian besar untuk menikmati destinasi wisata yang ada di Indonesia. Semakin banyak wisatawan yang berkunjung, maka akan semakin banyak pula Devisa negara yang akan diterima oleh Indonesia. Oleh karena itu, akan dilakukan pengelompokkan jumlah wisatawan mancanegara yang berkunjung ke Indonesia dengan *K-*

Means. Hal ini bertujuan sebagai informasi agar nantinya pemerintah Indonesia maupun pihak swasta dapat membuat kebijakan yang lebih baik lagi untuk dapat meningkatkan jumlah kunjungan wisatawan mancanegara ke Indonesia. Sampel dataset yang digunakan adalah data jumlah wisatawan mancanegara tahun 2017-2018 yang bersumber dari Badan Pusat Statistik Indonesia (Tabel 1.1).

Penyelesaian (Perhitungan Manual):

Berikut ini adalah proses perhitungan manual dengan menggunakan algoritma *K-Means Clustering* berdasarkan langkah-langkah yang sudah dijelaskan sebelumnya.

1. Menentukan data awal yang akan di *cluster*.
   Tabel di bawah merupakan data yang akan di proses menggunakan metode *K-Means Clustering.*

Tabel 1.1: Jumlah Kunjungan Wisatawan Mancanegara

No	Kebangsaan (A)	Tahun 2017 (X)	Tahun 2018 (Y)
1	Afganistan	595	1034
2	Albania	592	706
3	Armenia	772	794
4	Austria	27208	26676
5	Azerbaijan	4405	1138
6	Bahrain	2457	2208
7	Bangladesh	56503	50423
8	Belarus	4576	4340
9	Belgium	48477	47693
10	Bhutan	610	594
11	Bosnia Herzegovina	1784	982
12	Brunei Darussalam	23455	14394
13	Bulgaria	8695	6908
14	Canada	96139	89533
15	China	2093171	1994159
16	Croasia	6620	3723
17	Cyprus	1909	1815
18	Czech	19904	20250
19	Denmark	43721	44140
20	Egypt	20345	16297

No	Kebangsaan (A)	Tahun 2017 (X)	Tahun 2018 (Y)
21	Estonia	7569	7039
22	Finland	24447	24230
23	France	274117	275020
24	Georgia	1265	587
25	Germany	267823	257233
26	Hong Kong	98272	84825
27	Hongaria	12600	12488
28	I t a l y	90022	89411
29	Iceland (Islandia)	4776	2232
30	India	536902	535550
31	Irak	2167	2189
32	Iran	16301	11029
33	Irlandia (Ireland)	29400	25972
34	Israel	688	460
35	Japan	573310	485888
36	Jordan	6773	5726
37	Kamboja	6506	7950
38	Kazakhstan	7219	6671
39	Kirgistan	1254	1147
40	Korea Utara	140	608
41	Kuwait	5760	5199
42	Laos	4036	3577
43	Latvia	3932	3873
44	Libanon	6115	4868
45	Lithuania	8550	7790
46	Luxemburg	1720	1976
47	Macao	618	294
48	Macedonia	1381	944
49	Maladewa	2668	2204
50	Malaysia	2121888	2255115
51	Malta	2173	1181
52	Moldova	1030	738
53	Mongolia	2414	2992
54	Myanmar/Burma	48133	25320
55	Nepal	12821	13630

No	Kebangsaan (A)	Tahun 2017 (X)	Tahun 2018 (Y)
56	Netherlands	210426	196169
57	Norway	22838	23155
58	Oman	18615	24366
59	Pakistan	11424	12107
60	Palestina	2035	2339
61	Philippines	308977	198182
62	Polandia	32704	29431
63	Portugal	33223	33700
64	Qatar	1859	1970
65	R u s i a	117532	110028
66	Romania	18787	12807
67	Saudi Arabia	182086	152569
68	Serbia-Montenegro	7054	3658
69	Singapore	1554119	1526918
70	Slovakia	9264	8494
71	Slovenia	5264	4149
72	South Korea	423191	328471
73	Spain	81690	81022
74	Srilanka	35669	29709
75	Sweden	51417	44041
76	Switzerland	61191	56384
77	Syiria	2328	1792
78	Taiwan	264278	191986
79	Tajikhistan	350	488
80	Thailand	138235	113565
81	Timor Leste	960026	1610578
82	Turki	34433	19491
83	Turkmenistan	233	408
84	Ukraine	32964	23468
85	Uni Emirat Arab	8387	6540
86	United Kingdom	378131	362080
87	USA	344766	353030
88	Uzbekistan	4057	3184
89	Vietnam	77466	70963
90	Yaman	8453	9036

No	Kebangsaan (A)	Tahun 2017 (X)	Tahun 2018 (Y)
91	Yunani (Greece)	9896	6985

2. Menentukan Jumlah *Cluster*.
   Pada tahap ini adalah penulis menetapkan sebanyak ***2 (dua) cluster*** yang akan diterapkan dalam perhitungan manual *K-Means* yaitu ***cluster tinggi*** dan ***cluster rendah***.
3. Menentukan nilai *Centroid*.
   Untuk mendapatkan nilai titik tengah pada *Centroid* dari data, maka perlu membuat suatu ketentuan bahwa *cluster*isasi yang diinginkan adalah 2, Penentuan *cluster* dibagi menjadi 2 bagian yaitu *cluster* tingkat tinggi (C1), *cluster* tingkat rendah (C2). Untuk nilai titik *cluster* ditentukan dengan cara mengambil nilai terbesar (maksimum) untuk *cluster* tingkat tinggi (C1), dan nilai terkecil (minimum) untuk *cluster* tingkat rendah (C2). Nilai titik *cluster* dapat dilihat pada tabel berikut (diambil berdasarkan tabel 1.1):

Tabel 1.2: Nilai *Centroid* Awal

Centroid	C1 (Max)	2121888	2255115
	C2 (Min)	140	294

4. Menghitung Jarak *Centroid*.
   Untuk menghitung jarak antara titik *Centroid* dengan titik tiap objek menggunakan *Euclidian Distance*. Rumus untuk menghitung jarak dari *Centroid* adalah :

   $De = \sqrt{(xi - si)^2 + (yi - ti)^2}$

   Menghitung jarak *Centroid* 1 (C1) dan *Centroid* 2 (C2)

   $D_{A1C1} = \sqrt{(595 - 2121888)^2 + (1034 - 225115)^2} = 3095281{,}11$

   $D_{A1C2} = \sqrt{(595 - 140)^2 + (1034 - 294)^2} = 868{,}69$

   $D_{A2C1} = \sqrt{(592 - 2121888)^2 + (706 - 225115)^2} = 3095522{,}03$

   $D_{A2C2} = \sqrt{(592 - 140)^2 + (706 - 294)^2} = 611{,}59$

   $D_{A3C1} = \sqrt{(772 - 2121888)^2 + (794 - 225115)^2} = 3095334{,}60$

   $D_{A3C2} = \sqrt{(772 - 140)^2 + (794 - 294)^2} = 805{,}87$

   $D_{A4C1} = \sqrt{(27208 - 2121888)^2 + (26676 - 225115)^2} = 3058369{,}61$

   $D_{A4C2} = \sqrt{(27208 - 140)^2 + (26676 - 294)^2} = 37797{,}97$

$D_{A5C1} = \sqrt{(4405-2121888)^2+(1138-225115)^2} = 3092595{,}44$
$D_{A5C2} = \sqrt{(4405-140)^2+(1138-294)^2} = 4347{,}71$

$D_{A6C1} = \sqrt{(2457-2121888)^2+(2208-225115)^2} = 3093150{,}13$
$D_{A6C2} = \sqrt{(2457-140)^2+(2208-294)^2} = 3005{,}31$
$D_{A7C1} = \sqrt{(56503-2121888)^2+(50423-225115)^2} = 3021006{,}79$
$D_{A7C2} = \sqrt{(56503-140)^2+(50423-294)^2} = 75430{,}13$

$D_{A8C1} = \sqrt{(4576-2121888)^2+(4340-225115)^2} = 3090145{,}34$
$D_{A8C2} = \sqrt{(4576-140)^2+(4340-294)^2} = 6004{,}02$

$D_{A9C1} = \sqrt{(48477-2121888)^2+(47693-225115)^2} = 3028488{,}91$
$D_{A9C2} = \sqrt{(48477-140)^2+(47693-294)^2} = 67698{,}82$

$D_{A10C1} = \sqrt{(610-2121888)^2+(594-225115)^2} = 3095591{,}27$
$D_{A10C2} = \sqrt{(610-140)^2+(594-294)^2} = 557{,}58$

Dan seterusnya menggunakan cara yang sama hingga D_{A91C1} dan D_{A91C2} sehingga dihasilkan jarak terpendek dari *Centroid*. Menghitung jarak terpendek dapat dlilakukan dengan menggunakan rumus =**MIN(C1;C2)** pada *Ms. Excel*. Berikut tabel hasil dari perhitungan *Centroid* 1 dan *Centroid* 2 serta jarak terpendeknya.

Tabel 1.3: Hasil Perhitungan Iterasi 1

No	Kebangsaan	C1	C2	Jarak Terpendek
1	Afganistan	3095281,11	868,69	868,69
2	Albania	3095522,03	611,59	611,59
3	Armenia	3095334,60	805,87	805,87
4	Austria	3058369,61	37797,97	37797,97
5	Azerbaijan	3092595,44	4347,71	4347,71
6	Bahrain	3093150,13	3005,31	3005,31
7	Bangladesh	3021006,79	75430,13	75430,13
8	Belarus	3090145,34	6004,02	6004,02
9	Belgium	3028488,91	67698,82	67698,82
10	Bhutan	3095591,27	557,58	557,58
11	Bosnia Herzegovina	3094504,25	1782,16	1782,16
12	Brunei Darussalam	3069894,40	27247,00	27247,00

No	Kebangsaan	C1	C2	Jarak Terpendek
13	Bulgaria	3085452,86	10813,56	10813,56
14	Canada	2965367,50	131070,24	131070,24
15	China	262531,33	2890722,47	262531,33
16	Croasia	3089194,82	7331,33	7331,33
17	Cyprus	3093811,86	2332,98	2332,98
18	Czech	3068054,48	28086,61	28086,61
19	Denmark	3034334,94	61820,51	61820,51
20	Egypt	3070633,33	25774,76	25774,76
21	Estonia	3086128,73	10034,19	10034,19
22	Finland	3062042,89	34113,96	34113,96
23	France	2708326,77	387991,97	387991,97
24	Georgia	3095147,56	1162,53	1162,53
25	Germany	2725635,62	371041,56	371041,56
26	Hong Kong	2967352,42	129519,80	129519,80
27	Hongaria	3078712,67	17434,03	17434,03
28	I t a l y	2969638,57	126572,56	126572,56
29	Iceland (Islandia)	3091544,12	5024,77	5024,77
30	India	2338607,37	758031,95	758031,95
31	Irak	3093362,68	2774,84	2774,84
32	Iran	3077242,04	19401,50	19401,50
33	Irlandia (Ireland)	3057381,98	38929,52	38929,52
34	Israel	3095635,41	572,59	572,59
35	Japan	2351224,79	751215,94	751215,94
36	Jordan	3087630,54	8573,41	8573,41
37	Kamboja	3086193,70	9956,92	9956,92
38	Kazakhstan	3086636,58	9527,77	9527,77
39	Kirgistan	3094747,21	1403,07	1403,07
40	Korea Utara	3095903,16	314,00	314,00
41	Kuwait	3088708,42	7459,45	7459,45
42	Laos	3091071,09	5094,79	5094,79
43	Latvia	3090926,75	5214,26	5214,26
44	Libanon	3088706,36	7524,77	7524,77
45	Lithuania	3084909,59	11265,79	11265,79
46	Luxemburg	3093824,12	2307,71	2307,71
47	Macao	3095804,28	478,00	478,00
48	Macedonia	3094808,04	1400,92	1400,92

No	Kebangsaan	C1	C2	Jarak Terpendek
49	Maladewa	3093008,47	3168,42	3168,42
50	Malaysia	0,00	3096131,83	0,00
51	Malta	3094092,78	2218,08	2218,08
52	Moldova	3095198,59	994,60	994,60
53	Mongolia	3092608,61	3528,50	3528,50
54	Myanmar/Burma	3045069,05	54126,04	54126,04
55	Nepal	3077729,46	18402,63	18402,63
56	Netherlands	2809438,66	287379,92	287379,92
57	Norway	3063928,25	32215,28	32215,28
58	Oman	3065941,68	30344,47	30344,47
59	Pakistan	3079795,96	16336,33	16336,33
60	Palestina	3093343,89	2788,02	2788,02
61	Philippines	2741827,80	366796,88	366796,88
62	Polandia	3052598,74	43696,44	43696,44
63	Portugal	3049132,02	47015,38	47015,38
64	Qatar	3093733,24	2400,82	2400,82
65	Rusia	2935786,30	160693,60	160693,60
66	Romania	3074244,46	22456,31	22456,31
67	Saudi Arabia	2860687,24	237259,40	237259,40
68	Serbia-Montenegro	3088945,04	7688,95	7688,95
69	Singapore	923381,02	2178401,15	923381,02
70	Slovakia	3083907,60	12267,33	12267,33
71	Slovenia	3089813,12	6412,21	6412,21
72	South Korea	2568565,48	535417,87	535417,87
73	Spain	2981457,40	114749,35	114749,35
74	Srilanka	3050367,45	46125,40	46125,40
75	Sweden	3029141,53	67402,75	67402,75
76	Switzerland	3013451,53	82905,44	82905,44
77	Syiria	3093541,52	2651,67	2651,67
78	Taiwan	2776187,35	326365,91	326365,91
79	Tajikhistan	3095846,64	285,90	285,90
80	Thailand	2919095,00	178607,25	178607,25
81	Timor Leste	1328665,21	1874672,16	1328665,21
82	Turki	3058673,41	39300,57	39300,57
83	Turkmenistan	3095985,08	147,12	147,12

No	Kebangsaan	C1	C2	Jarak Terpendek
84	Ukraine	3056771,47	40180,21	40180,21
85	Uni Emirat Arab	3085931,95	10345,31	10345,31
86	United Kingdom	2573765,72	523226,82	523226,82
87	USA	2603092,38	493142,74	493142,74
88	Uzbekistan	3091342,97	4867,75	4867,75
89	Vietnam	2991685,36	104754,08	104754,08
90	Yaman	3084068,48	12063,52	12063,52
91	Yunani (Greece)	3084574,32	11830,00	11830,00

5. Menentukan posisi *cluster.*

   Dalam menentukan posisi *cluster* berdasarkan tabel 1.3, dapat dilakukan dengan mengikuti pernyataan berikut. "Jika nilai jarak terpendek berada di kolom C1 maka pada tabel posisi *cluster*, kolom C1 diberi nilai 1" dan "Jika jarak terpendek berada di kolom C2 maka pada tabel posisi *cluster*, kolom C2 diberi nilai 1". Nilai 1 hanya untuk simbolis atau pertanda bahwa pada kolom tersebut terdapat nilai jarak terpendek mewakili kolom tersebut. Berikut tabel posisi *Cluster* berdasarkan tabel 1.3:

Tabel 1.4: Posisi *Cluster* Iterasi 1

Kebangsaan	C1	C2
Afganistan		1
Albania		1
Armenia		1
Austria		1
Azerbaijan		1
Bahrain		1
Bangladesh		1
Belarus		1
Belgium		1
Bhutan		1
Bosnia		1
Brunei		1
Bulgaria		1
Canada		1
China	1	
Croasia		1
Cyprus		1
Czech		1

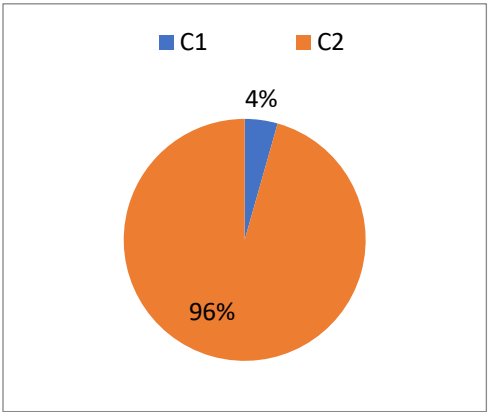
Kebangsaan	C1	C2
Iran		1
Irlandia		1
Israel		1
Japan		1
Jordan		1
Kamboja		1
Kazakhstan		1
Kirgistan		1
Korea Utara		1
Kuwait		1
Laos		1
Latvia		1
Libanon		1
Lithuania		1
Luxemburg		1
Macao		1
Macedonia		1
Maladewa		1

Kebangsaan	C1	C2
Portugal		1
Qatar		1
Rusia		1
Romania		1
Saudi Arabia		1
Serbia		1
Singapore	1	
Slovakia		1
Slovenia		1
South Korea		1
Spain		1
Srilanka		1
Sweden		1
Switzerland		1
Syiria		1
Taiwan		1
Tajikhistan		1
Thailand		1

Kebangsaan	C1	C2
Denmark		1
Egypt		1
Estonia		1
Finland		1
France		1
Georgia		1
Germany		1
Hong Kong		1
Hongaria		1
I t a l y		1
Islandia		1
India		1
Irak		1

Kebangsaan	C1	C2
Malaysia	1	
Malta		1
Moldova		1
Mongolia		1
Myanmar		1
Nepal		1
Netherlands		1
Norway		1
Oman		1
Pakistan		1
Palestina		1
Philippines		1
Polandia		1

Kebangsaan	C1	C2
Timor Leste	1	
Turki		1
Turkmenistan		1
Ukraine		1
Uni Emirat Arab		1
United Kingdom		1
USA		1
Uzbekistan		1
Vietnam		1
Yaman		1
Yunani		1



Gambar 1.4: Grafik *Cluster* iterasi 1

Diperoleh hasil dari posisi *cluster* 1 adalah C1 berjumlah 4 data (China, Malaysia, Singapura dan Timor Leste) dan C2 berjumlah 87 data. Setelah mendapatkan hasil perhitungan iterasi 1, maka selanjutnya menghitung iterasi yang kedua. Dalam hal ini, nilai *Centroid* yang di gunakan berbeda, harus menghitung nilai *Centroid* baru dengan cara menjumlahkan nilai *cluster* yang ada di kolom C1 dan C2 setelah itu dibagi dengan jumlah data itu sendiri.

Berikut perhitungannya :

$C1_X$ = 2093171 + 2121888 + 1554119 + 960026 / 4 = 1682301

$C1_Y$ = 1994159 + 2255115 + 1526918 + 1610578 / 4 = 1846692,5

$C2_X$ = 595 + 592 + 772 + 27208 + 4405 + 2457 + 56503 + 4576 + 48477 + 610 + 1784 + 23455 + 8695 + 96139 + 6620 + 1909 + 19904 + 43721 + 20345 + 7569 + 24447 + 274117 + 1265 + 267823 + 98272 + 12600 + 90022 + 4776 + 536902 + 2167 + 16301 + 29400 + 688 + 573310 + 6773 + 6506 + 7219 + 1254 + 140 + 5760 + 4036 + 3932 + 6115 + 8550 + 1720 + 618 + 1381 + 2668 + 2173 + 1030 + 2414 + 48133 + 12821 + 210426 + 22838 + 18615 + 11424 + 2035 + 308977 + 32704 + 33223 + 1859 + 117532 + 18787 + 182086 + 7054 + 9264 + 5264 + 423191 + 81690 + 35669 + 51417 + 61191 + 2328 + 264278 + 350 + 138235 + 34433 + 233 + 32964 + 8387 + 378131 + 344766 + 4057 + 77466 + 8453 + 9896 / 87
= 61734,39

$C2_Y$ = 1034 + 706 + 794 + 26676 + 1138 + 2208 + 50423 + 4340 + 47693 + 594 + 982 + 14394 + 6908 + 89533 + 3723 + 1815 + 20250 + 44140 + 16297 + 7039 + 24230 + 275020 + 587 + 257233 + 84825 + 12488 + 89411 + 2232 + 535550 + 2189 + 11029 + 25972 + 460 + 485888 + 5726 + 7950 + 6671 + 1147 + 608 + 5199 + 3577 + 3873 + 4868 + 7790 + 1976 + 294 + 944 + 2204 + 1181 + 738 + 2992 + 25320 + 13630 + 196169 + 23155 + 24366 + 12107 + 2339 + 198182 + 29431 + 33700 + 1970 + 110028 + 12807 + 152569 + 3658 + 8494 + 4149 + 328471 + 81022 + 29709 + 44041 + 56384 + 1792 + 191986 + 488 + 113565 + 19491 + 408 + 23468 + 6540 + 362080 + 353030 + 3184 + 70963 + 9036 + 6985 / 87
= 54715,59

NB : Nilai $C1_X$ merupakan nilai penjumlahan data awal yang berdasarkan nilai posisi *cluster* pada tabel 3.4. disesuaikan dengan nilai 1, jika nilai 1 pada kolom C1 berada pada urutan 15 maka sesuaikan pada nilai data awal setelah itu dijumlahkan sebanyak nilai 1 yang ada pada kolom C1 dan seterusnya sama dengan C1y, C2x, C2y.

Dari perhitungan di atas maka diperoleh hasil *Centroid* baru sebagai berikut:

Tabel 1.5: Nilai *Centroid* Baru Iterasi ke 2

Centroid	C1 (Max)	1682301	1846692,5
	C2 (Min)	61734,39	54715,59

Untuk menghitung Iterasi ke-2, dapat dilakukan dengan cara yang sama menggunakan persamaan:

$$De = \sqrt{(xi - si)^2 + (yi - ti)^2}$$

Sehingga menghasilkan perhitungan seperti tabel 1.6. berikut:

Tabel 1.6: Hasil Perhitungan Iterasi 2

No	Kebangsaan	C1	C2	Jarak Terpendek
1	Afganistan	2496916,17	81361,77	81361,77
2	Albania	2497160,65	81580,80	81580,80
3	Armenia	2496974,38	81387,66	81387,66
4	Austria	2460039,21	44477,97	44477,97
5	Azerbaijan	2494274,72	78467,93	78467,93
6	Bahrain	2494794,37	79188,73	79188,73
7	Bangladesh	2422767,68	6767,11	6767,11
8	Belarus	2491791,31	76189,12	76189,12
9	Belgium	2430181,08	15002,50	15002,50
10	Bhutan	2497231,33	81641,52	81641,52
11	Bosnia Herzegovina	2496153,97	80506,82	80506,82
12	Brunei Darussalam	2471656,90	55598,04	55598,04
13	Bulgaria	2487119,63	71405,48	71405,48
14	Canada	2367175,41	48948,23	48948,23
15	China	436532,39	2808589,58	436532,39
16	Croasia	2490872,01	75085,55	75085,55
17	Cyprus	2495453,92	79859,56	79859,56
18	Czech	2469707,67	54200,17	54200,17
19	Denmark	2436009,02	20888,40	20888,40
20	Egypt	2472336,03	56471,85	56471,85
21	Estonia	2487780,59	72159,17	72159,17
22	Finland	2463706,45	48163,48	48163,48
23	France	2110245,63	306007,20	306007,20
24	Georgia	2496795,46	81156,95	81156,95
25	Germany	2127705,22	288939,12	288939,12
26	Hong Kong	2369245,65	47345,26	47345,26
27	Hongaria	2480364,40	64787,02	64787,02
28	Italy	2371368,95	44765,62	44765,62
29	Islandia	2493215,73	77451,82	77451,82
30	India	1740986,37	676007,39	676007,39
31	Irak	2495003,69	79418,61	79418,61

No	Kebangsaan	C1	C2	Jarak Terpendek
32	Iran	2478954,72	63029,44	63029,44
33	Irlandia	2459086,18	43263,22	43263,22
34	Israel	2497277,86	81672,09	81672,09
35	Japan	1755462,88	669043,54	669043,54
36	Jordan	2489287,39	73625,63	73625,63
37	Kamboja	2487822,92	72368,47	72368,47
38	Kazakhstan	2488288,33	72665,05	72665,05
39	Kirgistan	2496388,83	80792,77	80792,77
40	Korea Utara	2497537,51	81984,75	81984,75
41	Kuwait	2490359,02	74733,02	74733,02
42	Laos	2492719,03	77099,02	77099,02
43	Latvia	2492570,20	76981,07	76981,07
44	Libanon	2490364,83	74688,01	74688,01
45	Lithuania	2486564,86	70926,65	70926,65
46	Luxemburg	2495462,17	79894,88	79894,88
47	Macao	2497447,72	81834,73	81834,73
48	Macedonia	2496453,40	80832,64	80832,64
49	Maladewa	2494655,26	79033,57	79033,57
50	Malaysia	600038,06	3014297,68	600038,06
51	Malta	2495744,94	80084,40	80084,40
52	Moldova	2496842,04	81231,78	81231,78
53	Mongolia	2494243,75	78703,48	78703,48
54	Myanmar/Burma	2447019,17	32389,79	32389,79
55	Nepal	2479371,21	63879,15	63879,15
56	Netherlands	2211480,01	205227,34	205227,34
57	Norway	2465584,41	50089,92	50089,92
58	Oman	2467534,19	52729,30	52729,30
59	Pakistan	2481437,87	65928,95	65928,95
60	Palestina	2494981,70	79418,66	79418,66
61	Philippines	2145601,47	285852,27	285852,27
62	Polandia	2454304,31	38497,71	38497,71
63	Portugal	2450795,80	35419,69	35419,69
64	Qatar	2495373,00	79794,48	79794,48
65	Rusia	2337628,20	78567,40	78567,40
66	Romania	2475967,46	60006,73	60006,73
67	Saudi Arabia	2262896,26	155112,22	155112,22

No	Kebangsaan	C1	C2	Jarak Terpendek
68	Serbia-Montenegro	2490628,17	74811,91	74811,91
69	Singapore	344508,86	2096328,16	344508,86
70	Slovakia	2485563,62	69925,51	69925,51
71	Slovenia	2491469,38	75801,61	75801,61
72	South Korea	1972398,16	453423,54	453423,54
73	Spain	2383180,20	33018,99	33018,99
74	Srilanka	2452106,44	36121,10	36121,10
75	Sweden	2430912,39	14845,72	14845,72
76	Switzerland	2415202,30	1754,67	1754,67
77	Syiria	2495188,80	79561,46	79561,46
78	Taiwan	2179183,98	244677,50	244677,50
79	Tajikhistan	2497484,78	81906,50	81906,50
80	Thailand	2321178,74	96517,34	96517,34
81	Timor Leste	759888,96	1796562,18	759888,96
82	Turki	2460515,04	44566,10	44566,10
83	Turkmenistan	2497622,71	82047,15	82047,15
84	Ukraine	2458548,38	42475,25	42475,25
85	Uni Emirat Arab	2487599,10	71880,67	71880,67
86	United Kingdom	1976090,50	441111,89	441111,89
87	USA	2005000,63	411215,74	411215,74
88	Uzbekistan	2492995,49	77344,59	77344,59
89	Vietnam	2393472,51	22615,53	22615,53
90	Yaman	2485708,86	70182,13	70182,13
91	Yunani	2486254,65	70465,79	70465,79

Setelah menghitung jarak *Centroid* pada iterasi ke-2, selanjutnya menempatkan posisi *cluster* dengan mengelompokkan nilai seperti pada tabel 1.4. yang telah dibahas sebelumnya. Proses *K-Means* akan terus berlangsung sampai iterasi ke-n sama hasilnya dengan iterasi sebelumnya. Berikut posisi *cluster* pada iterasi ke-2.

Tabel 1.7: Posisi *Cluster* Iterasi 2

Kebangsaan	C1	C2
Afganistan		1
Albania		1
Armenia		1
Austria		1
Azerbaijan		1

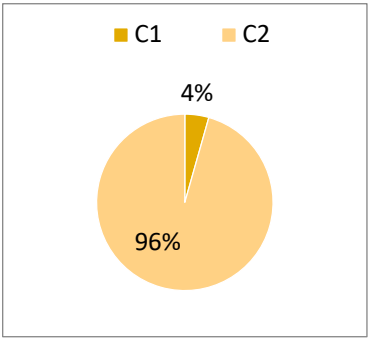
Kebangsaan	C1	C2
Iran		1
Irlandia		1
Israel		1
Japan		1
Jordan		1

Kebangsaan	C1	C2
Portugal		1
Qatar		1
Rusia		1
Romania		1
Saudi Arabia		1

Kebangsaan	C1	C2
Bahrain		1
Bangladesh		1
Belarus		1
Belgium		1
Bhutan		1
Bosnia		1
Brunei		1
Bulgaria		1
Canada		1
China	1	
Croasia		1
Cyprus		1
Czech		1
Denmark		1
Egypt		1
Estonia		1
Finland		1
France		1
Georgia		1
Germany		1
Hong Kong		1
Hongaria		1
I t a l y		1
Islandia		1
India		1
Irak		1

Kebangsaan	C1	C2
Kamboja		1
Kazakhstan		1
Kirgistan		1
Korea Utara		1
Kuwait		1
Laos		1
Latvia		1
Libanon		1
Lithuania		1
Luxemburg		1
Macao		1
Macedonia		1
Maladewa		1
Malaysia	1	
Malta		1
Moldova		1
Mongolia		1
Myanmar		1
Nepal		1
Netherlands		1
Norway		1
Oman		1
Pakistan		1
Palestina		1
Philippines		1
Polandia		1

Kebangsaan	C1	C2
Serbia		1
Singapore	1	
Slovakia		1
Slovenia		1
South Korea		1
Spain		1
Srilanka		1
Sweden		1
Switzerland		1
Syiria		1
Taiwan		1
Tajikhistan		1
Thailand		1
Timor Leste	1	
Turki		1
Turkmenistan		1
Ukraine		1
Uni Emirat Arab		1
United Kingdom		1
USA		1
Uzbekistan		1
Vietnam		1
Yaman		1
Yunani		1

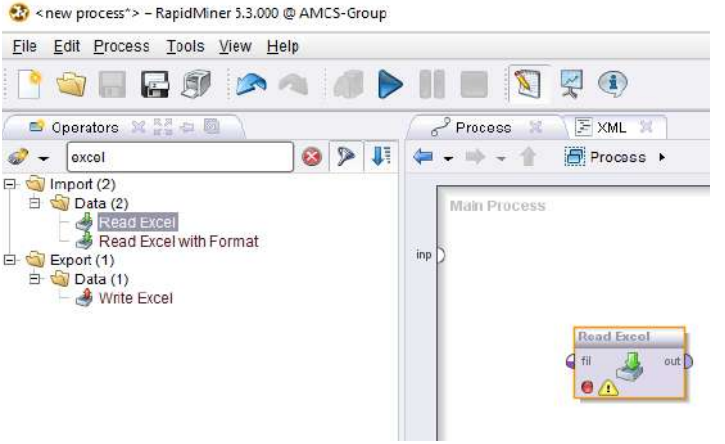


Gambar 1.5: Grafik *Cluster* iterasi 2

Berdasarkan tabel 1.4 dan tabel 1.7 serta gambar 1.4 dan gambar 1.5, dapat dilihat bahwa posisi *Cluster* 1 dan posisi *Cluster* 2 memiliki nilai *cluster* yang sama dan tidak ada perubahan. Sehingga proses perhitungan *K-Means* berhenti pada iterasi ke-2, karena iterasi ke-2 sama hasilnya dengan iterasi sebelumnya. Diperoleh hasil dari posisi *Cluster* 2 adalah C1 berjumlah 4 data dan C2 berjumlah 87, maka proses perhitungan dihentikan.

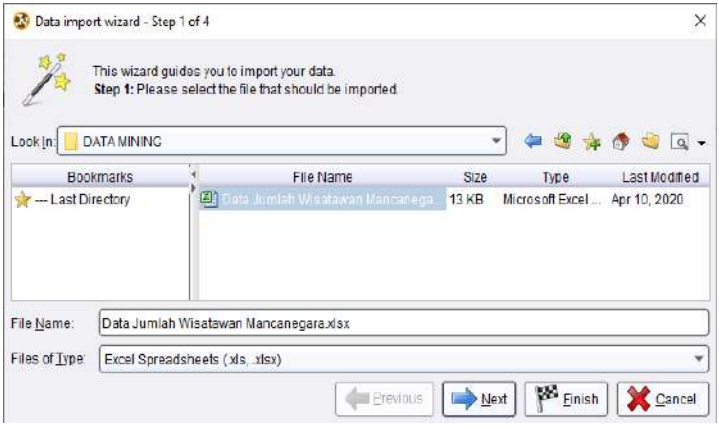
1.5.2 Pengujian *K-Means* dengan RapidMiner

Dalam menjalankan proses *Clustering*, terlebih dahulu pada saat membuka aplikasi *rapidminer* yang harus dilakukan adalah mengklik *new process* pada halaman awal. Setelah itu *import* data dari data yang telah di transformasi ke dalam *Microsoft excel* menggunakan operator *read excel* pada *Software Rapidminer*. Dalam melakukan *importing* data, dibutuhkan operator *read excel* seperti gambar berikut.



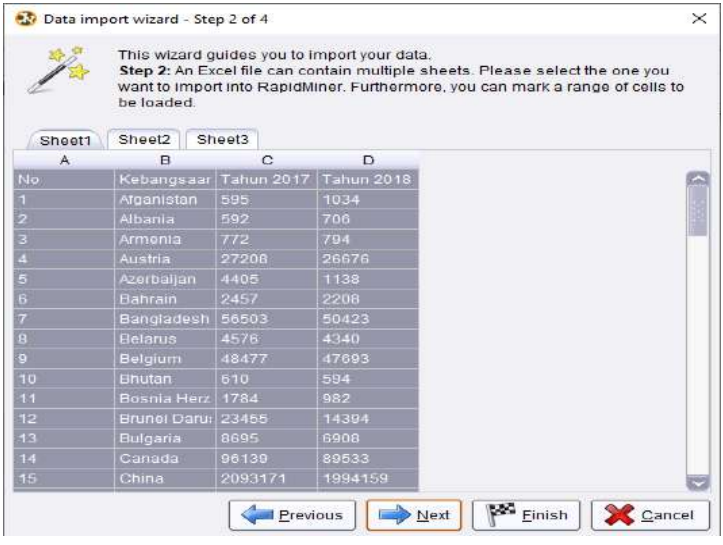
Gambar 1.6: *New Process* untuk *Import* Data *Excel*

Pada gambar 1.6. dilakukan pemilihan operator import data → *read excel*. Tarik operator *read excel* kedalam halaman *process*. Setelah itu pilih disebelah kanan menu *parameters* klik *import configuration wizard*. Kemudian akan muncul gambar berikut:



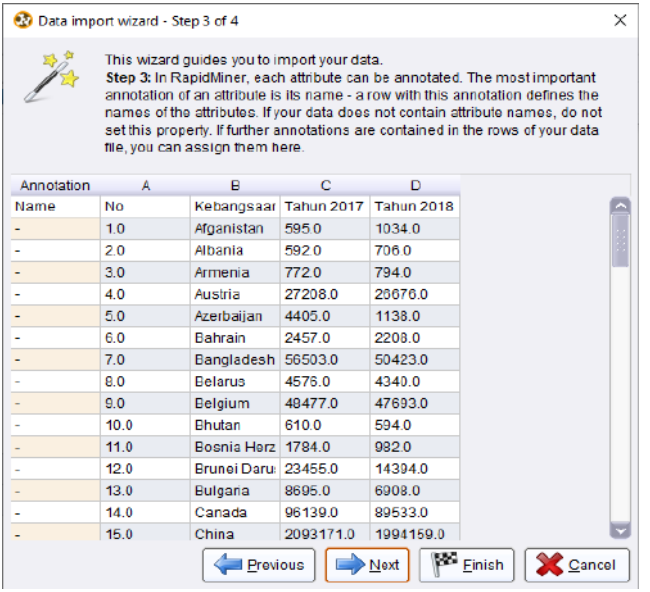
Gambar 1.7: Proses *Import* Data *Step* 1

Gambar 1.7 merupakan tampilan *Step 1* setelah diklik bagian *import configuration wizard*. Pilih lokasi data yang sebelumnya disimpan dalam format **.xlsx* (*Ms. Excel*) yang akan di import di *rapidminer*. Setelah data ditemukan, selanjutnya klik *next* di bagian bawah halaman dan akan diarahkan ke *Step 2*.



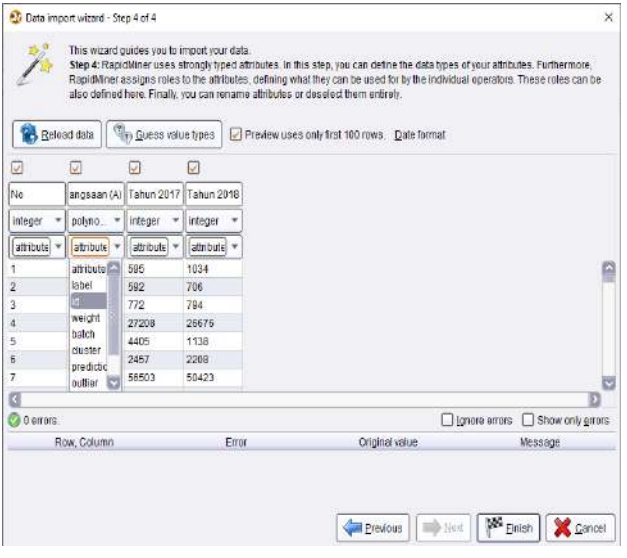
Gambar 1.8: Proses *Import* Data *Step* 2

Setelah muncul gambar 1.8, kemudian klik *Next* kembali.



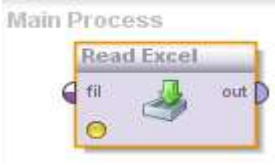
Gambar 1.9: Proses *Import* Data *Step* 3

Setelah muncul gambar 1.9, kemudian klik *Next* kembali.

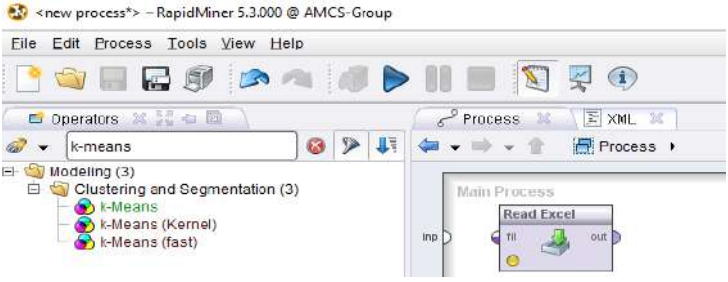


Gambar 1.10: Proses *Import* Data *Step* 4

Pada gambar 1.10 (Step 4), *attribute* dari *field* Kebangsaan diganti menjadi *id*, agar nanti hasil *cluster* nya berupa nama negara, bukan berupa numerik. Proses *import* berhasil apabila *fill* pada *Read Excel* berwarna kuning.

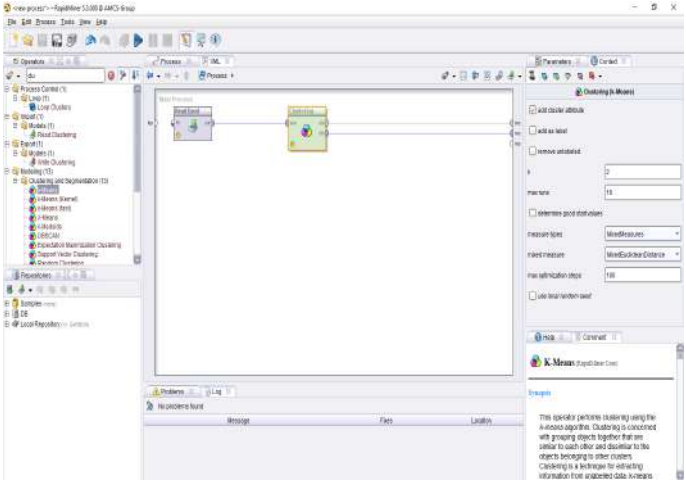


Gambar 1.11: *Read Excel* berhasil di *import*



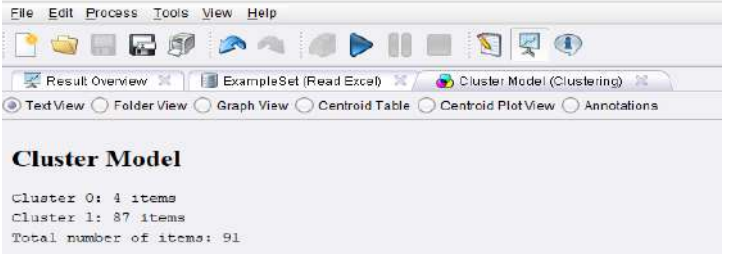
Gambar 1.12: Memanggil Algoritma *K-Means*

Pada kotak teks **[*filter*]** ketikkan *K-Means* untuk memanggil algoritma *K-Means*. Kemudian tarik icon *K-Means* kedalam halaman process.



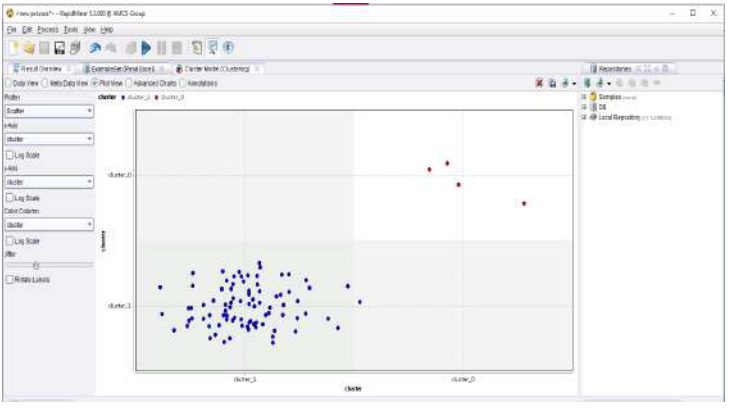
Gambar 1.13: Menghubungkan *Read Excel* dengan *K-Means*

Gambar 1.13 merupakan proses menghubungkan *read excel* dengan *K-Means* dan *output* yang akan di eksekusi. Kemudian pada *Parameters View Clustering (K-Means*) yang ada di sebelah kanan, ubah nilai *k* menjadi 2 (2 adalah banyak nya *Clustering* yang ingin diproses seperti yang dijelaskan pada contoh kasus, yakni *Clustering* tinggi dan rendah), dan ubah pilihan *measures types* menjadi *MixedMeasures.* Selanjutnya klik *tool run* yang berbentuk segitiga kesamping berwarna biru. Pada saat *tool run* di klik akan muncul hasil *cluster* seperti gambar dibawah.



Gambar 1.14: Hasil *Clustering* menggunakan *K-Means*

Berdasarkan gambar 1.14 dapat dijelaskan bahwa terdapat 2 *cluster* yang dimulai dari *cluster* 0 dan *cluster* 1. *Cluster* 0 merupakan *cluster* tinggi (C1 pada perhitungan manual) dan *cluster* 1 merupakan *cluster* rendah (C2 pada perhitungan manual). Sehingga didapatkan grafik *plot view* dari pengujian dengan rapidminer sebagai berikut.



Gambar 1.15: Grafik *Clustering* pada *Plot View*

Untuk membuat tampilan seperti gambar 1.15 adalah dengan cara: Pilih tab *ExampleSet (ReadExcel)* → Klik option button *Plot View* → pada *x-Axis* pilih *cluster*, begitu juga untuk *y-Axis* dan *Color Column* → Kemudian Scroll *Jitter* sedikit kekanan.

Sedangkan hasil pengelompokkan negara-negara yang terbagi kedalam 2 bagian, dapat dilihat pada gambar 1.16 berikut.

Gambar 1.16: Hasil *cluster* jumlah kunjungan Wisatawan Mancanegara

Untuk menampilkan gambar 1.16 dapat dilakukan dengan cara: Pilih tab *Cluster Model (Clustering)* → Klik option button *Folder View* → Klik masing-masing root pada *cluster*_0 dan *cluster*_1.

Setelah dilakukan pengujian menggunakan aplikasi *rapidminer*, diperoleh hasil bahwa jumlah negara yang termasuk *cluster* tinggi (*cluster* 0) yaitu negara China, Malaysia, Singapore dan Timor Leste. Sedangkan negara yang termasuk *cluster* rendah (*cluster* 1) ada 87 negara, yakni Afganistan hingga Yunani. Hal ini berarti bahwa perhitungan manual *K-Means* menggunakan *Microsoft Excel* dengan pengujian menggunakan aplikasi *rapidminer* hasilnya sama.

1.6 Kelebihan dan kekurangan *K-Means*

Berdasarkan uraian dan penjelasan yang telah dituliskan sebelumnya, *K-Means* memiliki beberapa kelebihan dan kekurangan sebagai berikut:

1.6.1 Kelebihan

Beberapa kelebihan *K-Means* antara lain:

1. Relatif efisien dan cepat
2. Dapat diimplementasikan untuk *machine learning* dan data mining.
3. Dapat bekerja dengan baik pada dataset yang berbentuk spherical (circular dua dimensi)

1.6.2 Kekurangan

Beberapa kekurangan *K-Means* antara lain:

1. Kurang cocok untuk dataset yang bersifat non-spherical (spherical dataset disebut juga non-linier dataset).
2. Harus menentukan nilai k di awal. Seringkali hasil perhitungan akan berbeda walaupun nilai k sama.
3. Sangat sensiitif dengan kondisi awal atau initial condition. Perbedaan kondisi awal dapat menyebabkan perbedaan hasil perhitungan *cluster*.
4. Algoritma *K-Means* dapat terjebak pada nilai lokal optimum.

Bab 2

Pengelompokan Data dengan Algoritma K-Medoids

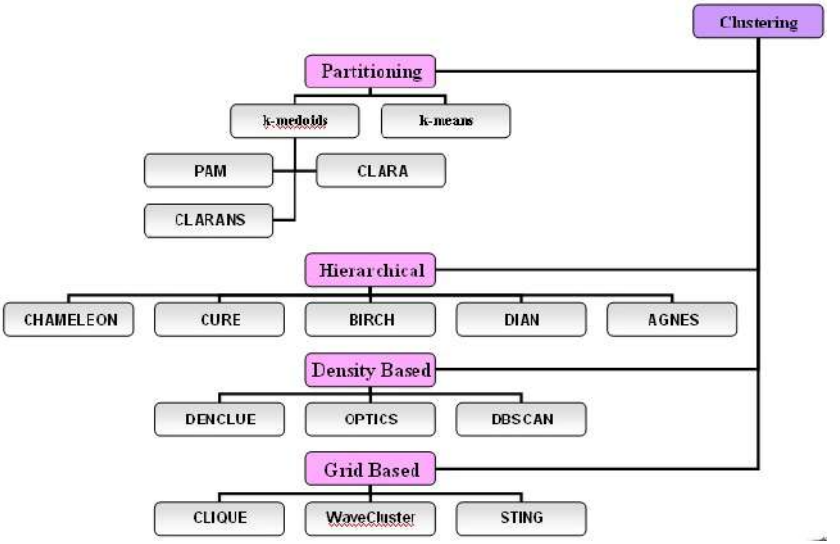
2.1 Pendahuluan

2.1.1 Clastering

Salah satu proses yang umum dilakukan oleh data mining adalah *clustering* atau proses membagi (atau mempartisi) satu set objek data (atau observasi) menjadi beberapa *subset*. Masing-masing *subset* adalah satu kluster, sehingga objek-objek di dalam suatu kluster mirip satu sama lain, namun tidak mirip dengan objek-objek di dalam kluster lainnya (Han, Kamber and Pei, 2012). Tidak ada definisi yang diterima secara umum dari sebuah kluster, karena sebagian besar mendefinisikannya berdasarkan cara pandang dalam pengelompokannya (Estivill-Castro, 2002), yang berarti bahwa setiap pengguna bisa memiliki pengertian yang berbeda berdasarkan kebutuhan dan niat untuk memakai cara pengelompokannya. Oleh karena itu, selama bertahun-tahun penelitian, ratusan algoritma pengelompokan dan langkah-langkah evaluasi telah diusulkan, dengan masing-masing kelebihan dan kekurangannya.

Secara umum metode pada *clustering* dapat digolongkan ke dalam beberapa metode berikut di antaranya, metode partisi, metode hierarki, metode berbasis kerapatan (*density based method*), metode berbasis *grid*. Metode ini telah digunakan secara luas dan berulang dan setelah 60 hingga 20 tahun yang lalu, metode ini telah diteliti dan dipahami dengan baik, tetapi masih ada banyak

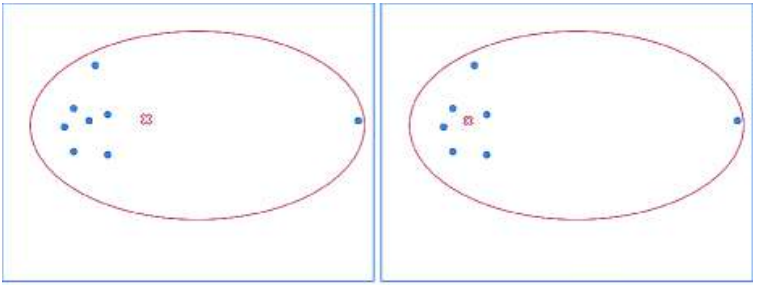
publikasi ilmiah yang mencoba menjelaskan algoritma ini dengan lebih baik lagi (Schubert and Rousseeuw, 2019).



Gambar 2.1: metode pada *clustering* secara umum (Lamiaa, Fattouh Ibrahim ; Manal, 2012)

2.1.2 K-Medoids

Pada bab sebelumnya telah dijelaskan tentang pengelompokan dengan algoritma *K-means* yang sensitif terhadap *outlier*, karena nilai rerata (*mean*) mudah dipengaruhi oleh nilai-nilai ekstrim. *K-Medoids* hadir untuk mengatasi kelemahan *K-Means*. K-Medoids merupakan varian dari *K-means* yang lebih kuat terhadap kebisingan dan *outlier*. K-medoid merupakan metode pengelompokan partisi yang menggunakan titik aktual dalam kluster untuk mewakilinya. Objek yang mewakili sebuah kluster disebut dengan *medoids*. Medoid merupakan objek yang letaknya terpusat di dalam suatu kluster dengan jarak minimum ke titik lain sehingga robust terhadap *outlier*. Kluster dibangun dengan menghitung kedekatan yang dimiliki antara medoids dengan objek non medoids (Han, Kamber and Pei, 2012). Seperti yang ditunjukkan pada Gambar 2.2. Kelebihan lainnya yaitu hasil proses *Clustering* tidak bergantung pada urutan masuk data set (Pramesti *et al.*, 2017).



Gambar 2.2: Mean & Medoid (Jin *et al.*, 2011)

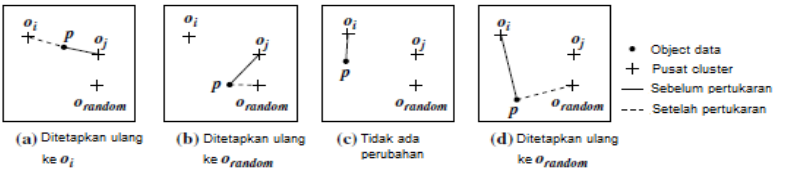
PAM (*Partitioning Around Medoids*)

Algoritma ini merupakan wujud umum dari *clustering k-medoids* atau sering juga disebut Algoritma K-Medoids (Pramesti *et al*., 2017) dan diperkenalkan pertama kali oleh (Kaufman and Rousseeuw, 1987). Algoritma ini mengatasi masalah iterasi, namun boros dan secara komputasi tidak efisien. Seperti algoritma *k-means*, objek di pusat kluster pada awalnya dipilih secara acak. Selanjutnya, mempertimbangkan apakah mengganti objek yang merupakan pusat kluster dengan objek lain akan meningkatkan kualitas *clustering* atau tidak. Semua kemungkinan perubahan dicoba. Proses iterasi dalam mengganti objek yang merupakan pusat kluster dengan objek lain berlanjut hingga kualitas *clustering* yang dihasilkan tidak dapat lagi ditingkatkan dengan mengubah atau dengan kata lain mencapai titik stabil. Kualitas ini diukur dengan fungsi perbedaan rata-rata (jarak deviasi) antara suatu objek dan objek di pusat kluster

$$E = \sum_{i=1}^{k} \sum_{p \epsilon C_i} dist(p, o_i)$$

Untuk lebih spesifik, misalnya, $\boldsymbol{o_1,\ldots,o_k}$ adalah pusat dari kluster (atau medoid nya). Untuk menentukan apakah objek bukan pusat kluster yang ditulis dalam $\boldsymbol{o_{random}}$ adalah pengganti yang baik untuk medoid saat ini (atau objek pusat kluster) $\boldsymbol{o_j}$ $\boldsymbol{(1 \leq j \leq k)}$, dapat dihitung jarak dari setiap objek $\boldsymbol{p}$ ke objek terdekat dalam $\{\boldsymbol{o_1,\ldots,o_{j-1},o_{random},o_{j+1},\ldots,o_k}\}$, dan dapat menggunakan jarak itu untuk memperbarui fungsi jarak / penyimpangan (perbedaan jarak antara objek dan pusat kluster). Mengatur objek ke $\{\boldsymbol{o_1,\ldots,o_{j-1},o_{random},o_{j+1},\ldots,o_k}\}$ itu mudah. Katakanlah objek $\boldsymbol{p}$ saat ini memasuki kluster yang disajikan dengan medoid $\boldsymbol{o_j}$ (gambar (a) / (b) di bawah). Apakah akan mengatur ulang $\boldsymbol{p}$ ke kluster lain jika diganti oleh $\boldsymbol{o_{random}}$? Objek $\boldsymbol{p}$ harus ditetapkan kembali

ke o_{random} atau ke objek lain yang disajikan dengan o_i $(i \neq j)$ terdekat. Misalnya, dalam diagram (a), p paling dekat dengan o_j. dan karena itu harus ditetapkan ulang ke o_i . Namun, dalam diagram (b), p paling dekat dengan o_{random} dan dengan demikian diatur ulang ke o_{random}. Sebaliknya, bagaimana jika p saat ini ditugaskan untuk grup yang diwakili oleh objek lain oi i ≠j?



Gambar 2.3: Empat kasus fungsi untuk pengelompokan k-medoid (Han, Kamber and Pei, 2012)

Objek o masih dimasukkan ke dalam kluster yang disajikan dengan o_i selama o masih lebih dekat ke o_i dibanding ke o_{random} (gambar 2.3 (c)). tetapi jika sebaliknya, maka o dimasukkan ke dalam o_{random} (gambar 2.3 (d)).
Setiap penetapan ulang dilakukan, selisih dalam simpangan mutlak, E, berkontribusi pada fungsi jarak/simpangan. Jadi, fungsi jarak/simpangan menghitung perbedaan/selisih nilai simpangan mutlak jika objek yang menjadi pusat kluster saat ini diganti dengan objek lain. Jumlah total jarak/simpangan dari pertukaran ini adalah jumlah simpangan-simpangan yang terjadi akibat pertukaran dengan objek-objek lain ketika dicoba untuk menjadi pusat kluster (medoid). Jika jumlah totalnya adalah negatif, maka o_j diganti dengan o_{random} karena simpangan mutlak *(absolute-error)* E berkurang. Jika jumlah totalnya adalah positif, maka objek yang sedang menjadi pusat kluster, o_j, tetap dipertahankan, dan tidak ada yang berubah dalam iterasi tersebut (Han, Kamber and Pei, 2012).

CLARA (Clustering LARge Applications)

Pada kasus dengan dataset yang besar, PAM tidak akan berjalan dengan baik. Maka metode berbasis sampling yang disebut dengan CLARA (Clustering LARge Applications) yang diperkenalkan oleh (Kaufman and Rousseeuw, 1990) bisa digunakan. Dalam hal ini CLARA menggunanakan sample dataset secara acak. Dalam banyak kasus, sample yang semakin besar akan bekerja dengan baik, sehingga setiap objek memiliki probabilitas yang sama untuk dipilih sebagai sample. Objek-objek yang dipilih menjadi pusat kluster (medoids) akan cenderung mirip dengan yang sudah dipilih dari seluruh dataset.

CLARA membuat clustering dari banyak sample secara acak dan menghasilkan clustering terbaik sebagai outputnya. Kompleksitas dalam menghitung medoids pada sample acak adalah $O(ks_2 + k(n-k))$, di mana s adalah ukuran sample, k adalah jumlah kluster, dan n adalah jumlah total objek. CLARA bisa mengatasi dataset yang lebih besar dibandingkan dengan PAM. Tingkat keefektivitasan CLARA bergantung pada ukuran sample.

Perhatikan bahwa PAM mencari k-medoids terbaik di antara dataset, tetapi CLARA mencari k-medoids terbaik di antara sample dataset yang terpilih. CLARA tidak bisa menghasilkan clustering yang baik jika medoids terbaik yang disamplekan sangat jauh dari k-medoids terbaik. Jika objek merupakan salah satu dari k-medoids terbaik tetapi tidak dipilih selama proses sampling, maka CLARA tidak akan pernah menghasilkan clustering terbaik (Han, Kamber and Pei, 2012).

CLARANS (Clustering LARge Applications based upon RANdomized Search)

CLARANS pertama kali diperkenalkan oleh (Ng and Han, 1994) dan pada (Ng and Han, 2002). CLARANS atau disebut juga dengan Algoritma teracak (randomized) memberikan 'pertukaran' antara biaya (kompleksitas komputasi) dan keefektifan dalam menggunakan sample untuk menghasilkan clustering, di mana pada saat mencari medoids yang lebih baik, PAM menguji setiap objek di dalam dataset terhadap medoid saat ini (objek yang sedang menjadi pusat kluster), karena CLARA membatasi kandidat medoids hanya ke suatu sampel acak dari dataset, maka CLARANS merupakan cara optimal yang dilakukan untuk hasil yang lebih baik.

CLARANS secara acak memilih objek-objek sebanyak k di dalam dataset sebagai medoids. Kemudian memilih secara acak satu medoid x dan satu objek y yang bukan salah satu medoids. Pertanyaannya, Bisakah dengan menggatikan x dengan y meningkatkan kriteria simpangan mutlak? Jika ya, penggantian dilakukan. CLARANS melakukan pencarian acak sebanyak m kali. Sekumpulan medoids yang dihasilkan dari sejumlah m langkah, dianggap sebagai nilai optimum lokal. CLARANS mengulang proses acak ini sebanyak m kali yang akan menghasilkan optimal lokal terbaik sebagai hasil akhir (Han, Kamber and Pei, 2012).

2.2 Clastering Data dengan Algoritma K-medoids

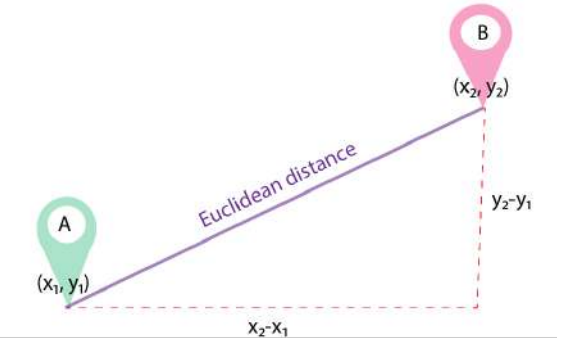
Tahapan Algoritma K-Medoids adalah sebagai berikut (Ningsih *et al*., 2019):

1. Inisialisasi pusat cluster sebanyak *k* (jumlah cluster).
2. Alokasikan setiap data (objek) ke cluster terdekat.
3. Pilih secara acak objek pada masing-masing cluster sebagai kandidat medoid baru.
4. Hitung jarak setiap objek yang berada pada masing-masing cluster dengan kandidat medoid baru.
5. Hitung total simpangan (S) dengan menghitung nilai total distance baru – total distance lama. Jika $S < 0$, maka tukar objek dengan data *cluster* untuk membentuk sekumpulan k objek baru sebagai medoid.
6. Ulangi langkah 3 sampai 5 hingga tidak terjadi perubahan medoid, sehingga didapatkan cluster beserta anggota cluster masing-masing

Pada proses clustering, pengukuran simpangan/jarak atau distance space memegang peran yang sangat penting dalam menentukan kemiripan atau keteraturan di antara data dan item. hal ini dilakukan untuk mengetahui, dengan cara seperti apa data dikatakan saling terkait, mirip, tidak mirip, dan metode pengukuran jarak seperti apa yang diperlukan untuk membandingkannya (Pramesti et al., 2017). Pemilihan metode pengukuran simpangan pada langkah 4 dilakukan berdasar pada kasus clustering seperti apa yang akan dilakukan untuk menentukan atau mendeskripsikan nilai kuantitatif dari tingkat kemiripan atau ketidakmiripan data (proximity measure). Beberapa metode yang sering digunakan, yaitu Euclidean distance, Manhattan distance atau Minkowski distance.

Euclidean Distance

Euclidean distance merupakan salah satu metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 (dua) buah titik dalam Euclidean space (meliputi bidang euclidean dua dimensi, tiga dimensi, atau bahkan lebih) (Nishom, 2019). Seperti contoh gambar 2.4, jarak garis lurus antara 2 titik data A dan B .



Gambar 2.4: Euclidean Distance (Gohrani, 2019)

Untuk mengukur jarak tersebut dalam banyak kasus digunakan untuk mengukur tingkat kemiripan data, maka digunakan rumus *euclidean distance* sebagai berikut:

$$d(x,y) = |x-y| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

di mana,

d = jarak antara *x* dan *y*
x = data pusat klaster
y = data pada atribut
i = setiap data
n = jumlah data,
xi = data pada pusat klaster ke *i*
yi = data pada setiap data ke *i*

Contoh *K-Medoid* dengan perhitungan metode *Euclidean Distance*

Tabel 2.1: Contoh kelompok data

Objek	X1	X2
A	2	6
B	3	4
C	3	8
D	4	7
E	6	2
F	6	4
G	7	3
H	7	4
I	8	5
J	7	6

Langkah pertama, kita tentukan jumlah k sebanyak 2,

Langkah kedua, kita pilih B (3, 4) sebagai medoid-1 dan H (7, 4) sebagai medoid-2

Langkah ketiga adalah menghitung jarak obyek ke masing-masing medoid yang telah dipilih, dengan rumus *Euclidean Distance*

Tabel 2.2: Iterasi 1 dengan *Euclidean Distance*

Objek	X1	X2	Jarak objek ke medodid 1	Jarak objek ke medodid 1
A	2	6	$d_{A,M_1} = \sqrt{((2-3)^2 + (6-4)^2)} = 2{,}23$	$d_{A,M_2} = \sqrt{((2-7)^2 + (6-4)^2)} = 5{,}38$
B	3	4	0	$d_{B,M_2} = \sqrt{((3-7)^2 + (4-4)^2)} = 4$
C	3	8	$d_{C,M_1} = \sqrt{((3-3)^2 + (8-4)^2)} = 4$	$d_{C,M_2} = \sqrt{((3-7)^2 + (8-4)^2)} = 5{,}65$
D	4	7	$d_{D,M_1} = \sqrt{((4-3)^2 + (7-4)^2)} = 3{,}16$	$d_{D,M_2} = \sqrt{((4-7)^2 + (7-4)^2)} = 4{,}24$
E	6	2	$d_{E,M_1} = \sqrt{((6-3)^2 + (2-4)^2)} = 3{,}6$	$d_{E,M_2} = \sqrt{((6-7)^2 + (2-4)^2)} = 2{,}23$
F	6	4	$d_{F,M_1} = \sqrt{((6-3)^2 + (4-4)^2)} = 3$	$d_{F,M_2} = \sqrt{((6-7)^2 + (4-4)^2)} = 1$

G	7	3	$d_{G,M_1} = \sqrt{((7-3)^2 + (3-4)^2)} = 4{,}12$	$d_{G,M_2} = \sqrt{((7-7)^2 + (3-4)^2)} = 1$
H	7	4	$d_{H,M_1} = \sqrt{((7-3)^2 + (4-4)^2)} = 4$	0
I	8	5	$d_{I,M_1} = \sqrt{((8-3)^2 + (5-4)^2)} = 5{,}1$	$d_{I,M_2} = \sqrt{((8-7)^2 + (5-4)^2)} = 1{,}414$
J	7	6	$d_{J,M_1} = \sqrt{((7-3)^2 + (6-4)^2)} = 4{,}47$	$d_{J,M_2} = \sqrt{((7-7)^2 + (6-4)^2)} = 2$

Total jarak terdekat pada medoid awal adalah 2.23 + 0 + 4 + 3.16 + 2.23 + 1 + 1 + 0 + 1.414 + 2 = 17.034

Dengan demikian anggota cluster-1 adalah A, B, C, dan D serta anggota cluster-2 adalah E, F, G, H, I, dan J

Langkah keempat adalah melakukan iterasi pada medoid yang akan berhenti jika anggota *cluster* yang terbentuk dari medoid tidak mengalami perubahan lagi.

Kita pilih B (3, 4) sebagai medoid-1 dan G (7, 3) sebagai medoid-2

Tabel 2.3: Iterasi 2 dengan *Euclidean Distance*

Objek	X1	X2	Jarak objek ke medodid 1	Jarak objek ke medodid 1
A	2	6	$d_{A,M_1} = \sqrt{((2-3)^2 + (6-4)^2)} = 2{,}23$	$d_{A,M_2} = \sqrt{((2-7)^2 + (6-3)^2)} = 5{,}83$
B	3	4	0	$d_{B,M_2} = \sqrt{((3-7)^2 + (4-3)^2)} = 4{,}12$
C	3	8	$d_{C,M_1} = \sqrt{((3-3)^2 + (8-4)^2)} = 4$	$d_{C,M_2} = \sqrt{((3-7)^2 + (8-3)^2)} = 6{,}4$
D	4	7	$d_{D,M_1} = \sqrt{((4-3)^2 + (7-4)^2)} = 3{,}16$	$d_{D,M_2} = \sqrt{((4-7)^2 + (7-3)^2)} = 5$
E	6	2	$d_{E,M_1} = \sqrt{((6-3)^2 + (2-4)^2)} = 3{,}6$	$d_{E,M_2} = \sqrt{((6-7)^2 + (2-3)^2)} = 1{,}141$
F	6	4	$d_{F,M_1} = \sqrt{((6-3)^2 + (4-4)^2)} = 3$	$d_{F,M_2} = \sqrt{((6-7)^2 + (4-3)^2)} = 1{,}141$
G	7	3	$d_{G,M_1} = \sqrt{((7-3)^2 + (3-4)^2)} = 4{,}12$	0
H	7	4	$d_{H,M_1} = \sqrt{((7-3)^2 + (4-4)^2)} = 4$	$d_{H,M_2} = \sqrt{((7-7)^2 + (4-3)^2)} = 1$
I	8	5	$d_{I,M_1} = \sqrt{((8-3)^2 + (5-4)^2)} = 5{,}1$	$d_{I,M_2} = \sqrt{((8-7)^2 + (5-3)^2)} = 2{,}23$
J	7	6	$d_{J,M_1} = \sqrt{((7-3)^2 + (6-4)^2)} = 4{,}47$	$d_{J,M_2} = \sqrt{((7-7)^2 + (6-3)^2)} = 3$

Total jarak terdekat pada medoid awal adalah 2.23 + 0 + 4 + 3.16 + 1.414 + 1.414 + 0 + 1+2.23 + 3 = 18.448

Dengan demikian, anggota cluster-1 adalah A, B, C, dan D serta anggota cluster-2 adalah E, F, G, H, I, dan J

Langkah kelima adalah menghitung total simpangan.

S = b — a

S = 18.448 — 17.034

S = 1.414

Karena b > a, maka iterasi dihentikan.

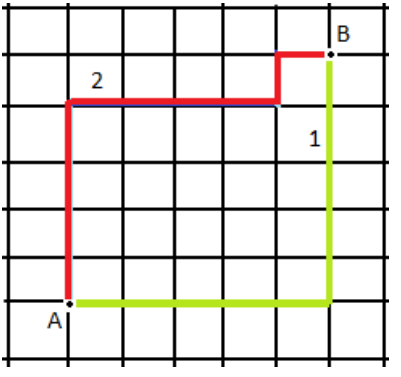
Sehingga, anggota *cluster* yang terbentuk pada masing-masing medoid adalah:

anggota **cluster-1** adalah **A, B, C,** dan **D**

anggota **cluster-2** adalah **E, F, G, H, I,** dan **J**

Manhattan Distance

Manhattan distance digunakan untuk menghitung perbedaan absolut (mutlak) antara koordinat sepasang objek. Sebagai ilustrasi, pada gambar di bawah, bayangkan setiap sel merupakan kotak, dan setiap sisi merupakan jalan. Jika ingin melakukan perjalanan dari titik A ke titik B yang ditandai pada gambar, kita bisa melalui jalur merah atau kuning. Akan terlihat bahwa jalannya tidak lurus dan ada belokan. Untuk menghitung jarak tersebut, dapat menggunakan Manhattan Distance.



Gambar 2.5: *Manhattan Distance* (Gohrani, 2019)

Rumus yang digunakan adalah sebagai berikut:

$$d(x, y) = \sum_{i=1}^{n} |x_i - x_i|$$

di mana,

d = jarak antara *x* dan *y*
x = data pusat klaster
y = data pada atribut
i = setiap data
n = jumlah data,
xi = data pada pusat klaster ke *i*
yi = data pada setiap data ke *i*

Contoh perhitungan dengan menggunakan data yang sama pada table 2.1:

Langkah pertama, kita tentukan jumlah k sebanyak 2,

Langkah kedua, kita pilih B (3, 4) sebagai medoid-1 dan H (7, 4) sebagai medoid-2

Langkah ketiga adalah menghitung jarak obyek ke masing-masing medoid yang telah dipilih, dengan rumus *manhattan distance*

Tabel 2.4: Iterasi 1 dengan *manhattan distance*

Objek	X1	X2	Jarak objek ke medodid 1	Jarak objek ke medodid 1
A	2	6	|2-3|+|6-4|=3	|2-7|+|6-4|=5
B	3	4	|3-3|+|4-4|=0	|3-7|+|4-4|=4
C	3	8	|3-3|+|8-4|=4	|3-7|+|8-4|=7
D	4	7	|4-3|+|7-4|=4	|4-7|+|7-4|=6
E	6	2	|6-3|+|2-4|=5	|6-7|+|2-4|=3
F	6	4	|6-3|+|4-4|=3	|6-7|+|4-4|=1
G	7	3	|7-3|+|3-4|=5	|7-7|+|3-4|=1

H	7	4	|7-3|+|4-4|=4	|7-7|+|4-4|=0
I	8	5	|8-3|+|5-4|=6	|8-7|+|5-4|=2
J	7	6	|7-3|+|6-4|=7	|7-7|+|6-4|=2

Total jarak terdekat pada medoid awal adalah = 3 + 0 + 4 + 4 + 3 + 1 + 1 + 0 + 2 + 2 = 20

Didapat anggota cluster-1 adalah A, B, C, dan D dan anggota cluster-2 adalah E, F, G, H, I, dan J, Kemudian mencari pusat medoid yang baru didapat yaitu B (3,4) dan G (7,3)

Langkah keempat adalah melakukan iterasi pada medoid yang akan berhenti jika anggota *cluster* yang terbentuk dari medoid tidak mengalami perubahan lagi.

Tabel 2.5: Iterasi 2 dengan *manhattan distance*

Objek	**X1**	**X2**	**Jarak objek ke medodid 1**	**Jarak objek ke medodid 1**
A	2	6	|2-3|+|6-4|=3	|2-7|+|6-3|=8
B	3	4	|3-3|+|4-4|=0	|3-7|+|6-3|=6
C	3	8	|3-3|+|8-4|=4	|3-7|+|6-3|=9
D	4	7	|4-3|+|7-4|=4	|4-7|+|7-3|=7
E	6	2	|6-3|+|2-4|=5	|6-7|+|2-3|=2
F	6	4	|6-3|+|4-4|=3	|6-7|+|4-3|=2
G	7	3	|7-3|+|3-4|=5	|7-7|+|3-3|=0
H	7	4	|7-3|+|4-4|=4	|7-7|+|4-3|=1
I	8	5	|8-3|+|5-4|=6	|8-7|+|5-3|=3
J	7	6	|7-3|+|6-4|=7	|7-7|+|6-3|=3

Total jarak terdekat pada medoid =3 + 0 + 4 + 4 + 2 + 2 + 0 + 1 + 3 + 3=22

Langkah kelima adalah menghitung total simpangan.

S = b — a

S = 22 — 20

S = 2

Karena b > a, maka iterasi dihentikan.

Sehingga, anggota *cluster* yang terbentuk pada masing-masing medoid adalah:

anggota cluster-1 adalah **A, B, C,** dan **D**

anggota cluster-2 adalah **E, F, G, H, I,** dan **J**

Minkowski Distance

Minkowski distance merupakan sebuah metrik dalam ruang vektor di mana suatu norma didefinisikan (normed vector space) sekaligus dianggap sebagai generalisasi dari Euclidean distance dan Manhattan distance. Dalam pengukuran jarak objek menggunakan *minkowski distance* biasanya digunakan nilai *p* adalah 1 atau 2. Berikut rumus yang digunakan menghitung jarak dalam metode ini.

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

di mana,

d = jarak antara *x* dan *y*
x = data pusat klaster
y = data pada atribut
i = setiap data
n = jumlah data,
xi = data pada pusat klaster ke *i*
yi = data pada setiap data ke *i*
p = power

Jarak Minkowski adalah metrik jarak umum. Kita dapat memanipulasi rumus di atas dengan mengganti 'p' untuk menghitung jarak antara dua titik data dengan cara yang berbeda (Gohrani, 2019).

Algoritma K-medoids memiliki kinerja yang lebih optimal jika jumlah data yang digunakan berjumlah sedikit(Ningsih *et al.*, 2019).

2.3 Aplikasi Clastering dengan Algoritma K-medoids

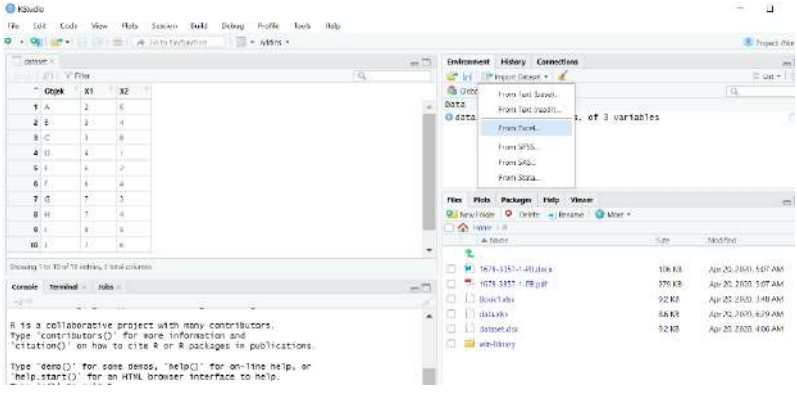
Pada dataset besar, tentunya perhitungan manual sangat melelahkan, beberapa software yang bebas digunakan untuk melakukan clastering menggunakan algoritma K-medoids di antaranya dapat menggunakan bahasa pemrograman R atau aplikasi RStudio dan RapidMiner 5.3. Di bawah ini akan dipaparkan secara singkat, penggunaan aplikasi untuk melakukan *clastering* dengan algoritma K-medoids

2.3.1 Algoritma K-Medoids pada Aplikasi RStudio

Sebagai contoh, data yang kita gunakan dalam menjalankan software ini adalah data pada tabel 2.1

Adapun langkah langkahnya :

1. Import data yang telah disiapkan terlebih dahulu di lembar kerja MS. Excel seperti tampilan pada gambar 2.6, pada contoh ini disimpan dengan nama dataset

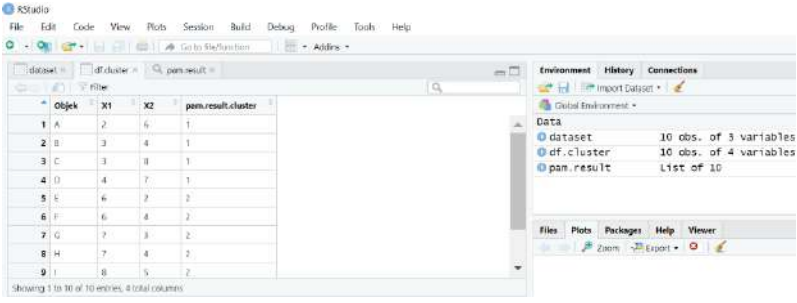


Gambar 2.6: Rstudio import data

2. Lakukan Proses clastering algoritma k-medoids dengan memasukan script R (asumsi semua modul *clustering* yang diperlukan telah di install dan di aktifkan) dengan memakai *library cluster* dan fungsi pam serta jumlah k = 2 (jumlah claster yang diinginkan)
   - library(cluster)
   - pam.result <- pam(dataset[,2:3],2)
   - print(pam.result)
   - pam.result$diss
   - df.cluster = data.frame(dataset,pam.result$cluster)
   - View(df.cluster)

Hasil :

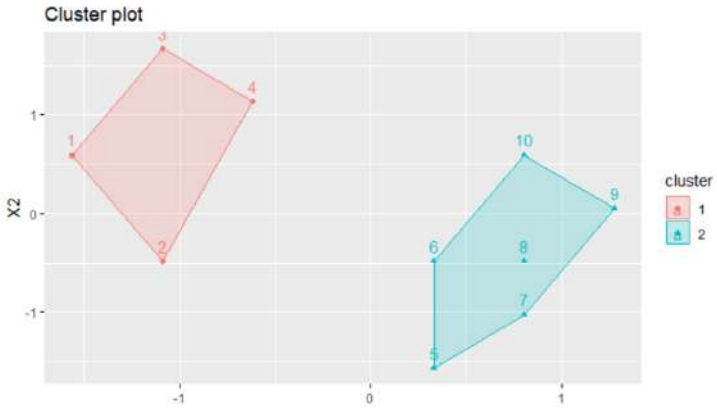
```
Medoids:
     ID X1 X2
[1,]  1  2  6
[2,]  8  7  4
Clustering vector:
 [1] 1 1 1 1 2 2 2 2 2 2
Objective function:
   build     swap
1.553663 1.435849

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
 [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```



Gambar 2.7 Rstudio hasil *cluster*

Pada tabel df.cluster terlihat pembagian klaster di kolom pam.result.cluster

3. Menampilkan plot visualisasi K-Medoids dengan perintah :
   - fviz_cluster(pam.result, data = dataset[,2:3])

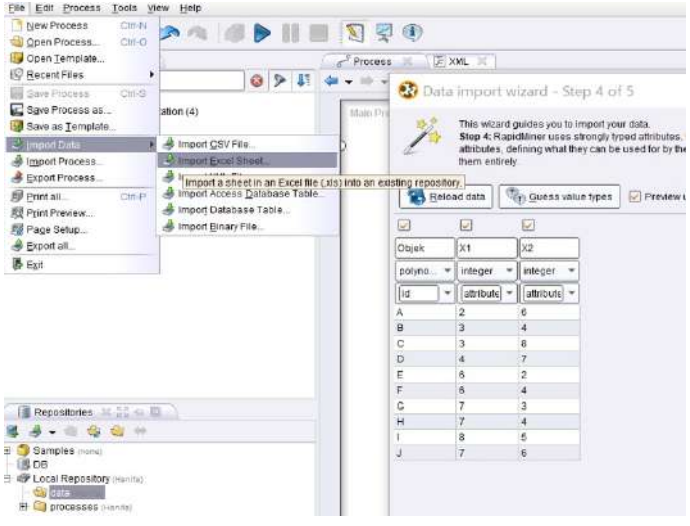


Gambar 2.8: Rstudio visualisasi K-Medoids

2.3.2 Algoritma K-Medoids pada Aplikasi RapidMiner

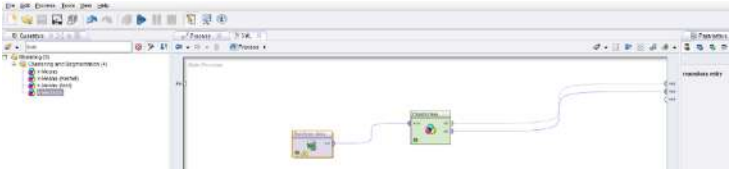
Data yang digunakan dalam aplikasi ini adalah data pada tabel 2.1

1. Seperi halnya di RStudio, pada RapidMiner ini juga dapat mengimport data melalui *tool Data Import Wizard* yang tersedia, seperti pada Gambar 2.9.



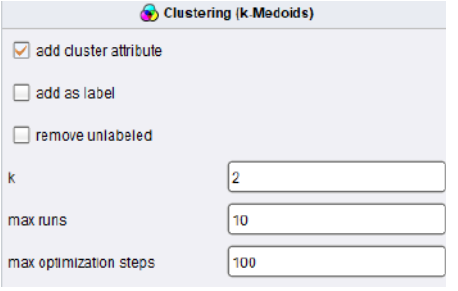
Gambar 2.9: RapidMiner 3.5 import data

2. Pada lembar main process, tarik data dan modeling clastering dalam hal ini adalah k-medoids, seperti padaGambar 2.10



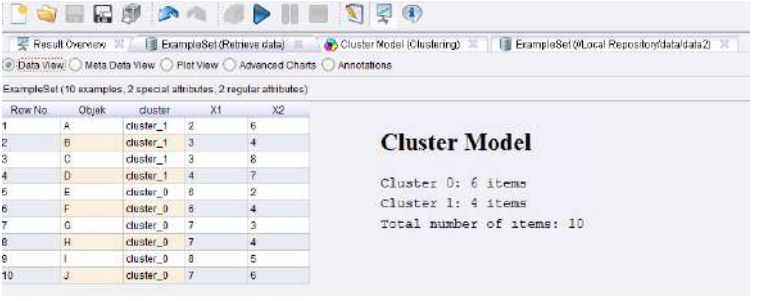
Gambar 2.10: RapidMiner 3.5 main process

3. Pilih jumlah k yang diinginkan serta parameter lainnya



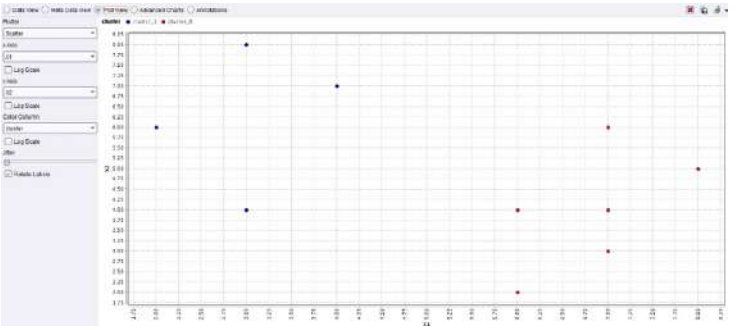
Gambar 2.11: RapidMiner 3 Paremeter K-medoids

4. Menjalankan proses dengan menekan F11 atau mengklik run pada program, maka akan tampil hasi berupa inisialisasi claster untuk masing masing objek seperti yang ditunjukkan pada Gambar 2.12



Gambar 2.12: Hasil clastering K-Medoids RapidMiner 3.5

5. Pada Plot View, kita dapat melihat hasil berupa Visualisasi data, di mana terdapat 2 claster yang dibedakan dengan warna biru dan merah



Gambar 2.13: RapidMiner 3.5 visualisasi data

Bab 3

Asosiasi Data Mining dengan Algoritma A Priori

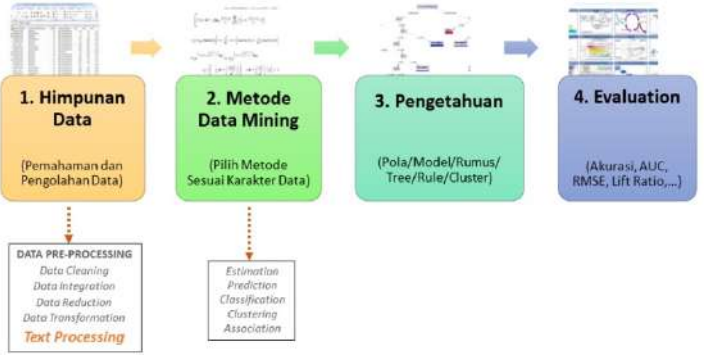
3.1 Pendahuluan

Algoritma A Priori termasuk dalam cabang ilmu data mining untuk kategori assosiasi (Purnia and Warnilah, 2017). Algoritma A Priori merupakan salah satu algoritma yang digunakan untuk menemukan pola frekuensi tinggi yang sangat terkenal (Bagus et al., 2018). Pola frekuensi tinggi merupakan pola-pola item di dalam suatu database yang memiliki frekuensi atau Support di atas ambang batas tertentu yang disebut dengan istilah minimum Support. Pola frekuensi tinggi ini digunakan untuk menyusun aturan assosiatif dan juga beberapa teknik data mining lainnya. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut sebagai *affinity analysis* atau *market basket analysis* (Panjaitan et al., 2019).

3.2 Assosiasi

Aturan assosiasi (Association rule) adalah salah satu metode yang ada di data mining yang bertujuan mencari pola yang sering muncul di antara banyak transaksi, di mana setiap transaksi terdiri dari beberapa item (Aditya, Marisa and Purnomo, 2016). *Association rule* merupakan bagian dari *frequent pattern mining*. *Frequent pattern mining* merupakan salah satu task data mining yang sangat penting. Task ini mencari hubungan/ relasi, assosiasi, dan korelasi dalam data (Aribowo, 2015). Tugas assosiasi dalam penambangan data adalah menemukan atribut yang muncul sekaligus. Dalam dunia bisnis lebih sering

disebut analisis keranjang belanja (market basket analysis) (Pahlevi, Sugandi and Sintawati, 2018).



Gambar 3.1: Proses secara umum data mining (Assosiasi)(Supriyadi et al., 2018)

3.2.1 Aturan Assosiasi dalam Penjualan

Beberapa aturan assosiasi dalam penjualan yang menjadi hal terpenting yang dilakukan oleh aturan asosiasi adalah:

a. Tersedianya database market basket (transaksi pembelian) pada pusat penjualan (apapun) yang secara otomatis menemukan assosiasi produk atau item-item yang tersimpan dalam database tersebut. Dalam hal ini database market basket tersebut mengandung record dalam jumlah yang amat besar di mana tiap record mencatat semua item yang dibeli oleh pelanggan dalam transaksi tunggal (Aqra et al., 2018).
b. Manfaat yang diperoleh dari aturan assosiasi:

1. Manfaat bagi manajer:
   (a) Dapat ditentukannya penempatan barang-barang dalam layout dengan tepat. Contoh: susu diletakkan berdekatan dengan diapers;
   (b) Promosi produk;
   (c) Segmentasi pembeli;
   (d) Pembuatan katalog.
   (e) Melihat pola kecenderungan pola belanja pelanggan

2. Aturan asosiasi juga dapat diterapkan dalam bentuk sistem rekomendasi, Contoh:
   (a) Sistem rekomendasi pembelian buku atau dvd online (www.amazon.com);
   (b) Sistem rekomendasi pencarian artikel dalam search engine;
   (c) Sistem rekomendasi peminjaman atau pengadaan buku pada perpustakaan; dalan lain-lain (Riszky and Sadikin, 2019; Bagus et al., 2018).
3. Penyajian informasi transaksi ke dalam bentuk "if-then"atau "jika maka". Aturan ini dihitung dari sifat probabilistik yang dimiliki oleh data yang ada (Nurjoko and Kurniawan, 2016).

Istilah - istilah umum dalam Aturan Assosiasi

a) Itemset
   - Sekumpulan satu/ lebih item. Misal: {Milk, Bread, Diaper}
   - k-itemset merupakan suatu itemset yang terdiri dari k-item

b) Support count (σ)

Frekuensi kemunculan suatu itemset. Misal: σ ({Milk, Bread, Diaper}) = 2

c) Support

Pecahan transaksi yang terdiri dari suatu itemset. Misal: s ({Milk, Bread, Diaper}) = 2/5

d) Frequent Itemset (Φ)

Suatu itemset yang memiliki nilai Support lebih tinggi atau sama dengan batas minimum Support (minsup)

e) Association Rule

Persamaan dalam bentuk $X \rightarrow Y$, di mana X dan Y merupakan itemset. Misal: {Milk, Diaper} $\rightarrow$ {Beer}

f) Rule Evaluation Metrics
   - Support (s)

Pecahan transaksi yang terdiri dari kedua item X dan Y

- Confidence (c)

Ukuran seberapa sering item dalam y muncul darlam transaksi yang terdiri dari x.

Contoh:

Tabel 3.1: Contoh aturan assosiasi

Transaksi	***Items***
Transaksi 1	Bread, Milk
Transaksi 2	Bread, Diaper, Beer, Eggs
Transaksi 3	Milk, Diaper, Beer, Coke
Transaksi 4	Bread, Milk, Diaper, Beer
Transaksi 5	Bread, Milk, Diaper, Coke

(Sumber: Data Olahan sendiri)

Misal:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

3.3 Algoritma A Priori

Arti A Priori secara umum merupakan anggapan atau sikap yang sudah ditentukan sebelum (melihat, menyelidiki) terhadap sesuatu. Oleh karena itulah, Algoritma A Priori termasuk dari jenis aturan asosiasi pada data mining seperti yang telah disebutkan sebelumnya (Riszky and Sadikin, 2019). Analisis assosiasi atau *association* adalah teknik data mining untuk menemukan aturan

assosiatif antara suatu kombinasi item (Murnawan, Sinaga and Nughraha, 2018). Salah satu contoh dari aturan assosiatif adalah hasil analisa pembelian di suatu toko di mana dapat diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik toko dapat mengatur penempatan barangnya atau merancang pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu. Analisis assosiasi menjadi terkenal karena aplikasinya dapat menganalisa isi keranjang belanja di toko, analisis assosiasi juga sering disebut dengan istilah market basket analysis (Purnia and Warnilah, 2017). Berikut Tabel 3.2 di bawah ini merupakan contoh transaksi pada suatu took.

Tabel 3.2: Tabel Transaksi Barang yang Dibeli (Purnia and Warnilah, 2017)

Transaksi	Barang yang dibeli
Transaksi 1	Barang1, Barang2, Barang3
Transaksi 2	Barang1, Barang2
Transaksi 3	Barang2, Barang5
Transaksi 4	Barang1, Barang2, Barang5

Mempelajari aturan assosiasi berarti komputer diminta untuk mencari barang-barang yang sering dibeli bersamaan. Pada Tabel 3.2 di atas, barang-barang yang paling sering dibeli bersamaan adalah Barang1 dan Barang2. Untuk selanjutnya barang disebut dengan item dan himpunan barang disebut itemset. Penting tidaknya suatu aturan assosiasi dapat diketahui dengan dua parameter, Support (nilai penunjang) yaitu: persentase kombinasi item tersebut dalam database dan confidence (nilai kepastian) yaitu: kuatnya hubungan antar item dalam aturan asosiasi (Bagus et al., 2018).

Aturan asosiasi biasanya dinyatakan dalam bentuk:

Contoh:

{roti, mentega} -> {susu} (Support = 40%, confidence = 50%)

Penjelasan: 50% dari transaksi di database yang memuat item roti dan mentega juga memuat item susu. Sedangkan 40% dari seluruh transaksi yang ada di database memuat ketiga item itu. Hal tersebut dapat juga diartikan: Seorang konsumen yang membeli roti dan mentega punya kemungkinan 50% untuk juga membeli susu. Aturan ini cukup signifikan karena mewakili 40% dari catatan transaksi selama ini.

Dalam hal analisis assosiasi digunakan untuk menemukan semua aturan assosiasi yang memenuhi syarat minimum untuk Support (minimum Support) dan syarat minimum untuk confidence (minimum confidence).

3.3 Langkah Penyelesaian Algoritma A Priori

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

a. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai Support dalam database. Nilai Support sebuah item diperoleh dengan rumus berikut:

$$Support\ (A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ transaksi} \quad (3.1)$$

sedangkan nilai *Support* dari 2 item diperoleh dari rumus 2 berikut:

$$Support\ (A, B) = P\ (A \cap B)$$

$$Support\ (A, B) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi} \quad (3.2)$$

b. Pembentukan aturan assosiasi

Setelah semua pola frekuensi tinggi ditemukan, dicari aturan assosiasi yang memenuhi syarat minimum untuk confidence dengan menghitung confidence aturan assosisasi $A \rightarrow B$. Nilai confidence dari aturan $A \rightarrow B$ diperoleh dari rumus berikut:

$$Confidence = P\ (B|A) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi\ A} \quad (3.3)$$

Sedangkan cara kerja dari Algoritma A Priori sendiri terbagi dari beberapa tahap yang disebut iterasi. Tahapan tersebut adalah :

1. Pembentukan kandidat itemset, kandidat k-itemset dibentuk dari kombinasi (k-1)-itemset yang didapat dari iterasi sebelumnya. Satu ciri dari algoritma A Priori adalah adanya pemangkasan kandidat k-itemset

yang subsetnya yang berisi k-1 item tidak termasuk dalam pola frekuensi tinggi dengan panjang k-1.

2. Perhitungan Support dari tiap kandidat k-itemset. Support dari tiap kandidat k-itemset didapat dengan menscan database untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat k-itemset tersebut. Ini juga merupakan ciri dari algoritma A Priori di mana diperlukan perhitungan dengan scan seluruh database sebanyak k-itemset terpanjang.
3. Tetapkan pola frekuensi tinggi. Pola frekuensi tinggi yang memuat k-item atau k-itemset ditetapkan dari kandidat k-itemset yang Supportnya lebih besar dari minimum Support.
4. Bila tidak didapat pola frekuensi tinggi maka seluruh proses dihentikan. Bila tidak, maka k tambah satu dan kembali ke bagian 1.

Kasus 1: Berikut adalah contoh penyelesaian algoritma A Priori kasus pertama. Ada beberapa transaksi yang disediakan (Penulis menggunakan simbol karakter huruf untuk contoh kasus pertama).

Tabel 3.3: Contoh kasus pertama transaksi barang yang dibeli

Transaksi	Item yang dibeli
T1	D, E, C
T2	C, A
T3	C, B, D
T4	A, C, E, D
T5	E, D
T6	E, C, B
T7	B, D, A

(Sumber: Data Olahan sendiri)

Berdasarkan item yang dibeli pada tabel 3.3, langkah yang harus dilakukan adalah:

a. Pisahkan masing-masing item yang dibeli dalam bentuk tabel atau sejenisnya. Hal ini dilakukan agar kita mengetahui item apa yang dari dari sekian transaksi yang dilakukan dalam satu periode (hari atau bulan). Dalam hal ini item yang dibeli adalah: A, B, C, D, E.
b. Buat tabel seperti di bawah ini dan hitung jumlahnya berdasarkan tabel 3.4.

Tabel 3.4: Perhitungan item yang dibeli berdasarkan item

Transaksi	A	B	C	D	E
T1	0	0	1	1	1
T2	1	0	1	0	0
T3	0	1	1	1	0
T4	1	0	1	1	1
T5	0	0	0	1	1
T6	0	1	1	0	1
T7	1	1	0	1	0
$\sum$ = (k=1)	3	3	5	5	4

(Sumber: Data Olahan sendiri)

c. Tentukan Frequent Itemset (Φ)

Misalkan kita tentukan Φ = 3, maka kita dapat menentukan frekuent itemset. Dari tabel 3.4 di atas diketahui total Φ untuk transaksi k=1, semuanya lebih besar dari Φ. Maka:

F1= {{A}, {B}, {C}, {D}, {E}}.

Nilai k=1 ≥ Φ.

Untuk k=2 (2 unsur), diperlukan tabel untuk tiap-tiap pasang item. Himpunan yang mungkin terbentuk adalah {A,B}, {A,C}, {A,D}, {A,E}, {B,C}, {B,D}, {B,E}, {C,D}, {C,E}, {D,E}.

Tabel 3.5: himpunan yang mungkin terbentuk untuk k=2 (2 unsur)

T	A	B	Hasil
T1	0	0	S
T2	1	0	S
T3	0	1	S
T4	1	0	S
T5	0	0	S
T6	0	1	S
T7	1	1	P
		∑	1

T	A	C	Hasil
T1	0	1	S
T2	1	1	P
T3	0	1	S
T4	1	1	P
T5	0	0	S
T6	0	1	S
T7	1	0	S
		∑	2

T	A	D	Hasil
T1	0	1	S
T2	1	0	S
T3	0	1	S
T4	1	1	P
T5	0	1	S
T6	0	0	S
T7	1	1	P
		∑	2

T	A	E	Hasil
T1	0	1	S
T2	1	0	S
T3	0	0	S
T4	1	1	P
T5	0	1	S
T6	0	1	S
T7	1	0	S
		∑	1

T	B	C	Hasil
T1	0	1	S
T2	0	1	S
T3	1	1	P
T4	0	1	S
T5	0	0	S
T6	1	1	P
T7	1	0	S
		∑	2

T	B	D	Hasil
T1	0	1	S
T2	0	0	S
T3	1	1	P
T4	0	1	S
T5	0	1	S
T6	1	0	S
T7	1	1	P
		∑	2

T	B	E	Hasil
T1	0	1	S
T2	0	0	S
T3	1	0	S
T4	0	1	S
T5	0	1	S
T6	1	1	P
T7	1	0	S
		∑	1

T	C	D	Hasil
T1	1	1	P
T2	1	0	S
T3	1	1	P
T4	1	1	P
T5	0	1	S
T6	1	0	S
T7	0	1	S
		∑	3

T	C	E	Hasil
T1	1	1	P
T2	1	0	S
T3	1	0	S
T4	1	1	P
T5	0	1	S
T6	1	1	P
T7	0	0	S
		∑	3

T	D	E	Hasil
T1	1	1	P
T2	0	0	S
T3	1	0	S
T4	1	1	P
T5	1	1	P
T6	0	1	S
T7	1	0	S
		∑	3

Dari tabel-tabel 2 unsur di atas, P merupakan item-item yang dijual bersamaan, sedangkan S merupakan item-item yang dijual secara bersamaan atau tidak terjadi transaksi. Σ melambangkan jumlah Frequent Itemset (Φ). Jumlah Frequent Itemset (Φ) harus lebih besar atau sama dengan jumlah Frekuensi item set (Σ >= Φ). Dari tabel 3.5 di atas, maka didapat:

F2 = {{C,D},{C,E},{D,E}}.

Kombinasi dari itemset dalam F2, dapat kita gabungkan menjadi calon 3-itemset. Itemset-itemset yang dapat digabungkan adalah itemset-itemset yang memiliki kesamaan dalam k-1 item pertama.

Untuk k=3 (3 unsur), himpunan yang mungkin terbentuk adalah: {C, D, E}.

Tabel 3.6: himpunan yang mungkin terbentuk untuk k=3 (3 unsur)

T	C	D	E	Hasil
1	1	1	1	P
2	1	0	0	S
3	1	1	0	S
4	1	1	1	P
5	0	1	1	S
6	1	0	1	S
7	0	1	0	S
			∑	2

Dari tabel 3.5 di atas, didapat F3= { }, karena tidak ada Σ >= Φ sehingga F4, F5, F6 dan F7 juga merupakan himpunan kosong.

d. Tentukan (ss-s) sebagai *antecedent* dan s sebagai *consequent* dari F_k (*frequent itemset of size* k) yang telah didapat pada F2. Di mana himpunan F2= {{C,D},{C,E},{D,E}}

Maka dapat disusun:

1) Untuk {C, D}:
   Jika (ss-s) = C, Jika s = D, Maka →If beli C then beli D
   Jika (ss-s) = D, Jika s = C, Maka →If beli D then beli C

2) Untuk {C, E}:
   Jika (ss-s) = C, Jika s = E, Maka →If beli C then beli E
   Jika (ss-s) = E, Jika s = C, Maka →If beli E then beli C
3) Untuk {D, E}:
   Jika (ss-s) = D, Jika s = E, Maka →If beli D then beli E
   Jika (ss-s) = E, Jika s = D, Maka →If beli E then beli D

e. Dari langkah di atas, maka kita memperoleh 6 *rule* yang dapat digunakan:

If beli C then beli D
If beli D then beli C
If beli C then beli E
If beli E then beli C
If beli D then beli E
If beli E then beli D

f. Hitung *support* dan *confidence* dengan menggunakan rumus 3.2 dan 3.3 seperti pada tabel 3.7 berikut:

Tabel 3.7: Perhitungan *support* dan *confidence*

If *antecedent* then *consequent*	***Support***	***confidence***
If beli C then beli D	(3/7)x100%=42,86%	(3/5)x100%=60%
If beli D then beli C	(3/7)x100%=42,86%	(3/5) x100%=60%
If beli C then beli E	(3/7)x100%=42,86%	(3/5) x100%=60%
If beli E then beli C	(3/7)x100%=42,86%	(3/4) x100%=75%

If *antecedent* then *consequent*	***Support***	***confidence***
If beli D then beli E	(3/7)x100%=42,86%	(3/5) x100%=60%
If beli E then beli D	(3/7)x100%=42,86%	(3/4) x100%=75%

g. Setelah perhitungan nilai support dan confidence diperoleh (tabel 3.7), kemudian lakukan perkalian antara support dan confidence, di mana confidence nya diambil 70% (untuk kasus ini nilai confidence diambil 70% keatas. Proses penentuan confidence berbeda disetiap kasus. Disesuaikan dengan contoh kasus. Pemberian nilai dilakukan berdasarkan hasil perhitungan support dan *confidence*. Gunakan konsep penalaran dalam mengambil nilai *confidence* yang dijadikan patokan.

Tabel 3.8: Nilai confidence nya 70% ke atas

If *antecedent* then *consequent*	***Support***	***confidence***	***Support x confidence***
If beli E then beli C	42,86%	75%	0.32145
If beli E then beli D	42,86%	75%	0.32145

h. Setelah perkalian support dan confidence dilakukan, hasil perkalian yang paling tinggi adalah rule yang dipakai pada saat menjual. Karena dari hasil ke-2 penjualan bernilai sama, maka ke-2 penjualan dapat dijadikan rule.
   1. Jika membeli E maka akan membeli C dengan support 42,86% dan confidence 75%
   2. Jika membeli E maka akan membeli D dengan support 42,86% dan confidence 75%

Kasus 2: Berikut adalah contoh penyelesaian algoritma A Priori kasus kedua. Ada beberapa transaksi yang disediakan.

Tabel 3.9: Contoh kasus kedua transaksi barang yang dibeli

Transaksi	Item yang dibeli
T1	Susu, Teh, Gula
T2	Teh, Gula, Roti
T3	Teh, Gula
T4	Susu, Roti
T5	Susu, Gula, Roti
T6	Teh, Gula
T7	Gula, Kopi, Susu
T8	Gula, Kopi, Susu
T9	Susu, Roti, Kopi
T10	Gula, Teh, Kopi

(Sumber: Data Olahan sendiri)

Berdasarkan item yang dibeli pada tabel 3.9, langkah yang harus dilakukan adalah:

1. Pisahkan masing-masing item yang dibeli dalam bentuk tabel atau sejenisnya. Dalam hal ini item yang dibeli adalah: Susu, Teh, Gula, Roti, Kopi.
2. Buat tabel seperti di bawah ini dan hitung jumlahnya berdasarkan tabel 3.10.

Tabel 3.10: Perhitungan item yang dibeli berdasarkan item

Transaksi	Susu	Teh	Gula	Roti	Kopi
T1	1	1	1	0	0
T2	0	1	1	1	0
T3	0	1	1	0	0
T4	1	0	0	1	0
T5	1	0	1	1	0

Transaksi	Susu	Teh	Gula	Roti	Kopi
T6	0	1	1	0	0
T7	1	0	1	0	1
T8	1	0	1	0	1
T9	1	0	0	1	1
T10	0	1	1	0	1
∑ = (k=1)	6	5	8	4	4

(Sumber: Data Olahan sendiri)

3. Tentukan Frequent Itemset (Φ)

Misalkan kita tentukan Φ = 2, Dari tabel 3.10 di atas diketahui total Φ untuk transaksi k=1, semuanya lebih besar dari Φ. Maka:

F1= {{Susu}, {Teh}, {Gula}, {Roti}, {Kopi}}.

Nilai k=1 ≥ Φ.

Untuk k=2 (2 unsur), diperlukan tabel untuk tiap-tiap pasang item. Himpunan yang mungkin terbentuk adalah {Susu, Teh}, {Susu, Gula}, {Susu, Roti}, {Susu, Kopi}, {The, Gula}, {The, Roti}, {Teh, Kopi}, {Gula, Roti}, {Gula, Kopi}, {Roti, Kopi}.

Tabel 3.11: himpunan yang mungkin terbentuk untuk k=2 (2 unsur)

T	Susu	Teh	H
T1	1	1	P
T2	0	1	S
T3	0	1	S
T4	1	0	S
T5	1	0	S
T6	0	1	S
T7	1	0	S
T8	1	0	S
T9	1	0	S

T	Susu	Gula	H
T1	1	1	P
T2	0	1	S
T3	0	1	S
T4	1	0	S
T5	1	1	P
T6	0	1	S
T7	1	1	P
T8	1	1	P
T9	1	0	S

T	Susu	Roti	H
T1	1	0	S
T2	0	1	S
T3	0	0	S
T4	1	1	P
T5	1	1	P
T6	0	0	S
T7	1	0	S
T8	1	0	S
T9	1	1	P

T10	0	1	S
		∑	1

T10	0	1	S
		∑	4

T10	0	0	S
		∑	3

T	Susu	Kopi	H
T1	1	0	S
T2	0	0	S
T3	0	0	S
T4	1	0	S
T5	1	0	S
T6	0	0	S
T7	1	1	P
T8	1	1	P
T9	1	1	P
T10	0	1	S
		∑	3

T	Teh	Gula	H
T1	1	1	P
T2	1	1	P
T3	1	1	P
T4	0	0	S
T5	0	1	S
T6	1	1	P
T7	0	1	S
T8	0	1	S
T9	0	0	S
T10	1	1	P
		∑	5

T	Teh	Roti	H
T1	1	0	S
T2	1	1	P
T3	1	0	S
T4	0	1	S
T5	0	1	S
T6	1	0	S
T7	0	0	S
T8	0	0	S
T9	0	1	S
T10	1	0	S
		∑	1

T	Teh	Kopi	H
T1	1	0	S
T2	1	0	S
T3	1	0	S
T4	0	0	S
T5	0	0	S
T6	1	0	S
T7	0	1	S
T8	0	1	S
T9	0	1	S
T10	1	1	P
		∑	1

T	Gula	Roti	H
T1	1	0	S
T2	1	1	P
T3	1	0	S
T4	0	1	S
T5	1	1	P
T6	1	0	S
T7	1	0	S
T8	1	0	S
T9	0	1	S
T10	1	0	S
		∑	2

T	Gula	Kopi	H
T1	1	0	S
T2	1	0	S
T3	1	0	S
T4	0	0	S
T5	1	0	S
T6	1	0	S
T7	1	1	P
T8	1	1	P
T9	0	1	S
T10	1	1	P
		∑	3

T	Roti	Kopi	H
T1	0	0	S
T2	1	0	S
T3	0	0	S
T4	1	0	S
T5	1	0	S
T6	0	0	S
T7	0	1	S
T8	0	1	S
T9	1	1	P
T10	0	1	S
		∑	1

Dari tabel-tabel 2 unsur di atas, P merupakan item-item yang dijual bersamaan, sedangkan S merupakan item-item yang dijual secara bersamaan atau tidak terjadi transaksi. Σ melambangkan jumlah Frequent Itemset (Φ). Jumlah Frequent Itemset (Φ) harus lebih besar atau sama dengan jumlah Frekuensi item set (Σ >= Φ). Dari tabel 3.11 di atas, maka didapat:

F2= {{Teh, Gula}, {Susu, Gula}, {Susu,Roti}, {Susu, Kopi}, {Gula, Kopi}, {Gula, Roti}}

Kombinasi dari itemset dalam F2, dapat kita gabungkan menjadi calon 3-itemset. Itemset-itemset yang dapat digabungkan adalah itemset-itemset yang memiliki kesamaan dalam k-1 item pertama.

Untuk k=3 (3 unsur), himpunan yang mungkin terbentuk adalah: {Teh, Gula, Susu, Roti, Kopi}.

Tabel 3.12: himpunan yang mungkin terbentuk untuk k=3 (3 unsur)

T	Teh	Gula	Susu	Hasil
T1	1	1	1	P
T2	1	1	0	S
T3	1	1	0	S
T4	0	0	1	S

T	Teh	Gula	Roti	Hasil
T1	1	1	0	S
T2	1	1	1	S
T3	1	1	0	S
T4	0	0	1	S

T5	0	1	1	S
T6	1	1	0	S
T7	0	1	1	S
T8	0	1	1	S
T9	0	0	1	S
T10	1	1	0	S
			∑	1

T5	0	1	1	S
T6	1	1	0	S
T7	0	1	0	S
T8	0	1	0	S
T9	0	0	1	S
T10	1	1	0	S
			∑	1

T	Teh	Gula	Kopi	Hasil
T1	1	1	0	S
T2	1	1	0	S
T3	1	1	0	S
T4	0	0	0	S
T5	0	1	0	S
T6	1	1	0	S
T7	0	1	1	S
T8	0	1	1	S
T9	0	0	1	S
T10	1	1	1	P
			∑	1

T	Teh	Susu	Roti	Hasil
T1	1	1	0	S
T2	1	0	1	S
T3	1	0	0	S
T4	0	1	1	S
T5	0	1	1	S
T6	1	0	0	S
T7	0	1	0	S
T8	0	1	0	S
T9	0	1	1	S
T10	1	0	0	S
			∑	0

T	Teh	Susu	Kopi	Hasil
T1	1	1	0	S
T2	1	0	0	S
T3	1	0	0	S
T4	0	1	0	S
T5	0	1	0	S
T6	1	0	0	S
T7	0	1	1	S
T8	0	1	1	S
T9	0	1	1	S

T	Teh	Roti	Kopi	Hasil
T1	1	0	0	S
T2	1	1	0	S
T3	1	0	0	S
T4	0	1	0	S
T5	0	1	0	S
T6	1	0	0	S
T7	0	0	1	S
T8	0	0	1	S
T9	0	1	1	S

T10	1	0	1	S
			∑	0

T10	1	0	1	S
			∑	0

T	Gula	Susu	Roti	Hasil
T1	1	1	0	S
T2	1	0	1	S
T3	1	0	0	S
T4	0	1	1	S
T5	1	1	1	P
T6	1	0	0	S
T7	1	1	0	S
T8	1	1	0	S
T9	0	1	1	S
T10	1	0	0	S
			∑	1

T	Gula	Susu	Kopi	Hasil
T1	1	1	0	S
T2	1	0	0	S
T3	1	0	0	S
T4	0	1	0	S
T5	1	1	0	S
T6	1	0	0	S
T7	1	1	1	P
T8	1	1	1	P
T9	0	1	1	S
T10	1	0	1	S
			∑	2

T	Susu	Roti	Kopi	Hasil
T1	1	0	0	S
T2	0	1	0	S
T3	0	0	0	S
T4	1	1	0	S
T5	1	1	0	S
T6	0	0	0	S
T7	1	0	1	S
T8	1	0	1	S
T9	1	1	1	P
T10	0	0	1	S
			∑	1

Dari tabel 3.12 di atas, maka

F3= {Gula, Susu, Kopi}

karena hanya kombinasi inilah yang memiliki frekeunsi kemunculan >= ϕ

4. Tentukan (ss-s) sebagai *antecedent* dan s sebagai *consequent* dari F_k (*frequent itemset of size* k) yang telah didapat pada F2 dan F3. Di mana himpunan F2= {{Teh, Gula}, {Susu, Gula}, {Susu, Roti}, {Susu, Kopi}, {Gula, Kopi}, {Gula, Roti}} dan F3= {Gula, Susu, Kopi}.

   Maka dapat disusun himpunan F2:

   a) Untuk {Teh, Gula}:
      Jika (ss-s) = Teh, Jika s = Gula, Maka →If beli Teh then beli Gula
      Jika (ss-s) = Gula, Jika s = Teh, Maka →If beli Gula then beli Teh
   b) Untuk {Susu, Gula}:
      Jika (ss-s) = Susu, Jika s = Gula, Maka →If beli Susu then beli Gula
      Jika (ss-s) = Gula, Jika s = Susu, Maka →If beli Gula then beli Susu
   c) Untuk {Susu, Roti}:
      Jika (ss-s) = Susu, Jika s = Roti, Maka →If beli Susu then beli Roti
      Jika (ss-s) = Roti, Jika s = Susu, Maka →If beli Roti then beli Susu
   d) Untuk {Susu, Kopi}:
      Jika (ss-s) = Susu, Jika s = Kopi, Maka →If beli Susu then beli Kopi
      Jika (ss-s) = Kopi, Jika s = Susu, Maka →If beli Kopi then beli Susu
   e) Untuk {Gula, Kopi}:
      Jika (ss-s) = Gula, Jika s = Kopi, Maka →If beli Gula then beli Kopi
      Jika (ss-s) = Kopi, Jika s = Gula, Maka →If beli Kopi then beli Gula
   f) Untuk {Gula, Roti}:
      Jika (ss-s) = Gula, Jika s = Roti, Maka →If beli Gula then beli Roti
      Jika (ss-s) = Roti, Jika s = Gula, Maka →If beli Roti then beli Gula

   Maka dapat disusun himpunan F3:

   a) Untuk {Gula, Susu, Kopi}:
      Jika (ss-s) = Gula dan Susu, Jika s = Kopi, Maka →If beli Gula and Susu then beli Kopi
   b) Untuk {Gula, Kopi, Susu}:
      Jika (ss-s) = Gula dan Kopi, Jika s = Susu, Maka →If beli Gula and Kopi then beli Susu
   c) Untuk {Kopi, Susu, Gula}:

Jika (ss-s) = Kopi dan Susu, Jika s = Kopi, Maka →If beli Kopi and Susu then beli Gula

5. Dari langkah di atas, maka kita memperoleh 12 *rule* untuk F2 dan 3 *rule* untuk F3 yang dapat digunakan:

If beli Teh then beli Gula
If beli Gula then beli Teh
If beli Susu then beli Gula
If beli Gula then beli Susu
If beli Susu then beli Roti
If beli Roti then beli Susu
If beli Susu then beli Kopi
If beli Kopi then beli Susu
If beli Gula then beli Kopi
If beli Kopi then beli Gula
If beli Gula then beli Roti
If beli Roti then beli Gula
If beli Gula and Susu then beli Kopi
If beli Gula and Kopi then beli Susu
If beli Kopi and Susu then beli Gula

6. Hitung *support* dan *confidence* dengan menggunakan rumus 3.2 dan 3.3 seperti pada tabel 3.13 berikut:

Tabel 3.13: Perhitungan *support* dan *confidence*

If *antecedent* then *consequent*	***Support***	***confidence***
If beli Teh then beli Gula	(5/10)x100%=50%	(5/5)x100%=100%
If beli Gula then beli Teh	(5/10)x100%=50%	(5/8)x100%=62,5%
If beli Susu then beli Gula	(4/10)x100%=40%	(4/6)x100%=66,7%
If beli Gula then beli Susu	(4/10)x100%=40%	(4/8)x100%=50%
If beli Susu then beli Roti	(3/10)x100%=30%	(3/6)x100%=50%
If beli Roti then beli Susu	(3/10)x100%=30%	(3/4)x100%=75%
If beli Susu then beli Kopi	(3/10)x100%=30%	(3/6)x100%=50%
If beli Kopi then beli Susu	(3/10)x100%=30%	(3/4)x100%=75%
If beli Gula then beli Kopi	(3/10)x100%=30%	(3/8)x100%=37,5%

If *antecedent* then *consequent*	*Support*	*confidence*
If beli Kopi then beli Gula	(3/10)x100%=30%	(3/4)x100%=75%
If beli Gula then beli Roti	(2/10)x100%=20%	(2/8)x100%=25%
If beli Roti then beli Gula	(2/10)x100%=20%	(2/4)x100%=50%
If beli Gula and Susu then beli Kopi	(2/10)x100%=20%	(2/4)x100%=50%
If beli Gula and Kopi then beli Susu	(2/10)x100%=20%	(2/3)x100%=66,7%
If beli Kopi and Susu then beli Gula	(2/10)x100%=20%	(2/3)x100%=66,7%

7. Setelah perhitungan nilai *support* dan *confidence* diperoleh (tabel 3.13), kemudian lakukan perkalian antara *support* dan *confidence*, di mana *confidence* nya diambil 60% (untuk kasus ini nilai *confidence* diambil 60% keatas. Proses penentuan *confidence* berbeda disetiap kasus. Disesuaikan dengan contoh kasus. Pemberian nilai dilakukan berdasarkan hasil perhitungan *support* dan *confidence*. Gunakan konsep penalaran dalam mengambil nilai *confidence* yang dijadikan patokan.

Tabel 3.14: Nilai *confidence* nya 60% ke atas

If *antecedent* then *consequent*	*Support*	*confidence*	*Support x confidence*
If beli Teh then beli Gula	50%	100%	0,50000
If beli Gula then beli Teh	50%	62,5%	0,31250
If beli Susu then beli Gula	40%	66,7%	0,26680
If beli Roti then beli Susu	30%	75%	0,22500
If beli Kopi then beli Susu	30%	75%	0,22500
If beli Kopi then beli Gula	30%	75%	0,22500
If beli Gula and Kopi then beli Susu	20%	66,7%	0,13340
If beli Kopi and Susu then beli Gula	20%	66,7%	0,13340

8. Setelah perkalian *support* dan *confidence* dilakukan, hasil perkalian yang paling tinggi adalah *rule* yang dipakai pada saat menjual. Karena

dari hasil ke-8 penjualan bernilai sama, maka ke-8 penjualan dapat dijadikan *rule*.

a) If beli Teh then beli Gula dengan *support* 50% dan *confidence* 100%.
b) If beli Gula then beli The dengan *support* 50% dan *confidence* 62,5%.
c) If beli Susu then beli Gula dengan *support* 40% dan *confidence* 66,7%.
d) If beli Roti then beli Susu dengan *support* 30% dan *confidence* 75%.
e) If beli Kopi then beli Susu dengan *support* 30% dan *confidence* 75%.
f) If beli Kopi then beli Gula dengan *support* 30% dan *confidence* 75%.
g) If beli Gula and Kopi then beli Susu dengan *support* 20% dan *confidence* 66,7%.
h) If beli Kopi and Susu then beli Gula dengan *support* 20% dan *confidence* 66,7%.

Bab 4

Pengklasifikasian Data dengan Algoritma C4.5

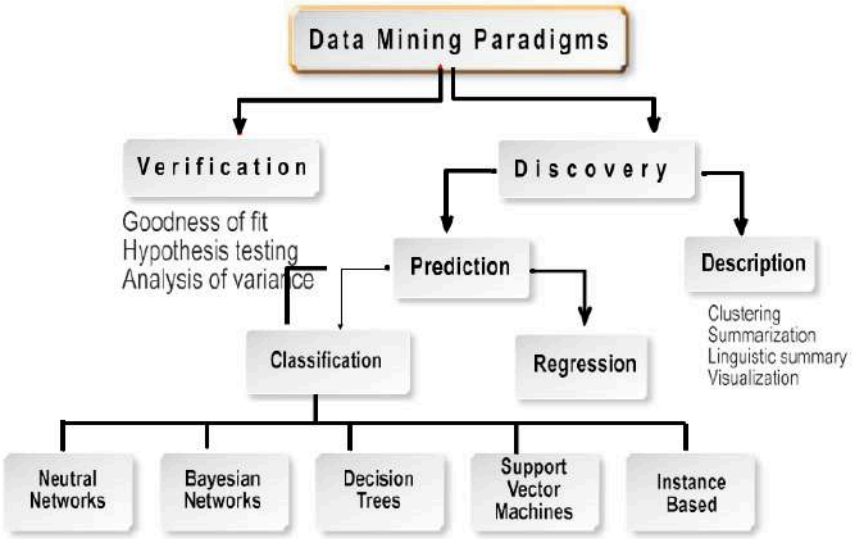
4.1 Pendahuluan

Klasifikasi adalah bagian dari ilmu Data Mining. *Classification* adalah sebuah model dalam data mining di mana, *classifier* dikontruksi untuk memprediksi *categorical label*, seperti "aman" atau "berisiko" untuk data aplikasi peminjaman uang; "ya" atau "tidak" untuk marketing; atau "treatment A", "treatment B" atau "treatment C" untuk data medis. Kategori tersebut dapat direpresentasikan dengan nilai yang sesuai dengan kebutuhannya, di mana pengaturan dari nilai tersebut tidak memiliki tertentu. (Han dan Kamber, 2006:285) dalam (Retno Tri Vulandari, 2017).

Dalam Klasifikasi memiliki ciri data yang bersifat atribut categorical dan atribut *categorical* sebagai label yang merupakan keputusan dari atribut data yang lain. Klasifikasi data adalah proses dua langkah, yang terdiri dari langkah belajar (di mana model klasifikasi didekonstruksi) dan langkah klasifikasi (di mana model digunakan untuk memprediksi label kelas untuk data yang diberikan) (Han, Kamber and Pei, 2012) Proses data klasifikasi memiliki dua tahapan, yang pertama adalah learning: yaitu training data dinalisa dengan menggunakan sebuah algoritma klasifikasi. Dan yang kedua adalah *classification* yaitu pada tahap ini test data digunakan untuk mengestimasi ketepatan dari *classification rules*. Jika keakuratan yang dikondisikan dan yang diperkirakan dapat diterima, rule tersebut dapat diaplikasikan pada klasifikasi lainnya dari tuple data yang baru (Vulandari, 2017).

Klasifikasi merupakan bagian dari Data Mining. Data Mining dapat diartikan sebagai proses penambangan data yang menghasilkan output (keluaran) berupa

pengetahuan (Nofriansyah and Nurcahyo, 2009). Taksonomi Metode Data Mining berguna untuk membedakan antara dua jenis utama penambangan data: berorientasi verifikasi (sistem memverifikasi hipotesis pengguna) dan berorientasi penemuan (sistem menemukan aturan dan pola baru secara mandiri). Gambar 4.1 mengilustrasikan taksonomi Metode Data Mining, yang secara otomatis mengidentifikasi pola dalam data, melibatkan metode prediksi dan deskripsi (Rokach and Maimon, 2012).



Gambar 4.1: Taxonomy dari Metode Data Mining

4.2 Atribut Data

Dalam penambangan data mining, Data atau data set sangat penting untuk diketahui. Data merupakan himpunan atribut yang belum mengandung arti, sedangkan informasi merupakan pengolahan data yang sudah mengandung arti dan berguna kepada si pemakai. Suyanto, (2018) menjelaskan himpunan data dibangun dari objek objek data, di mana objek data menyatakan entitas. Misalnya pada himpunan data universitas, objek data bias berupa mahasiswa, dosen dan mata kuliah.

Himpunan data dan jenis-jenis atribut dibagi atas 4 bagian (Suyanto, 2019)

a. Atribut Nominal

Atribut nominal disebut juga kategorikal karena nilainya menggambarkan kategori, kode, atau status yang tidak memilki urutan. Misalnya atribut warna yang mempunyai dua kemungkinan nilai : Merah dan Biru. Atribut kategori Pelanggan yang bisa bernilai : Silver, Gold dan Platinum. Atribut prestasi akademik bisa Sangat memuaskan, memuaskan, cukup, kurang dan sangat kurang. Atribut nominal dapat bernilai numerik. Misalnya atribut warna bisa bernilai 0 atau 1, di mana 0 menyatakan merah sedangkan 1 menyatakan putih. Kategori pelanggan bias bernilai 1, 2, atau 3, di mana 1 menyatakan Silver, 2 adalah Gold dan 3 menyatakan Platinum.

b. Atribut Biner

Atribut biner atau disebut juga dengan atribut Boolean adalah atribut nominal yang hanya memiliki dua kategori nilai : 0 atau 1, di mana 0 biasanya menyatakan Tidak (sesuatu yang negative atau yang berdampak kecil) sedangkan 1 menyatakan Ya (sesuatu yang positif atau yang berdampak besar). Atribut biner dibedakan menjadi dua, yaitu:

1. Atribut biner simetris, jika nilainya dianggap memberikan dampak yang setara, misalnya atribut jenis kelamin yang bernilai Pria atau Wanita.
2. Atribut biner asimetris, jika nilainya memberikan dampak berbeda, yang secara konvensi bernilai 1 untuk yang jarang terjadi dan bernilai 0 untuk umum yang terjadi, misalnya atribut Hasil Tes Buta Warna yang bernilai 1 (Buta Warna) dan 0 (Tidak Buta Warna)

c. Atribut Numerik

Atribut Numerik adalah kuantitatif, yang memiliki nilai berupa kuantitas yang terukur dan dinyatakan dalam nilai-nilai bulat (integer) atau riil (real). Atribut numerik bisa diskalakan secara interval dan secara rasio. Atribut numerik secara interval contohnya adalah temperature udara dalam satuan Celcius yang memiliki nilai 15 °C diruangan A dan 30 °C diruangan B, sedangkan atribut numeric diskalakan berdasarkan rasio adalah atribut gaji pegawai yang memiliki nilai 0 (tidak ada gaji) sampai 30 juta (gaji terbesar).

d. Atribut Ordinal

Atribut ordinal adalah memiliki nilai yang menggambarkan urutan atau peringkat (ranking). Namun ukuran perbedaan antara dua nilai yang berurutan tidak diketahui. Miasalnya atribut pelanggan yang bernilai : Silver, Gold, atau Platinum. Ketiga nilai tersebut memiliki urutan dan tingkatan namun tidak menjelaskan. Atribut ordinal sangat berguna dengan surveu, yaitu untuk penilaian subjektif (kualitatif) yang tidak dapat diukur secara objektif. Misalnya, survey tentang kepuasan pelanggan yang menghasilkan atribut bernilai ordinal, yaitu: 1 (Sangat Tidak Puas), 2 (Tidak Puas), 3 (Cukup Setuju), 4 (Setuju) dan 5 (Sangat Setuju). Jawaban ini adalah untuk survey linker lima.

e. Atribut Diskrit dan Kontinu

Semua jenis atribut di atas pada dasarnya dapat dikelompokkan kedalam atribut diskrit atau atribut kontinu. Atribut diskrit memiliki nilai terbatas atau nilai terbatas tapi masih dapat dihitung, yang bisa bernilai bulat. Misalnya atribut Temperatur Udara dan Atribut Nomor Pelanggan. Sedangkan atribut kontinu memiliki nilai riil atau pecahan yang dalam computer, misalnya atribut gaji pegawai yang bisa bernilai Rp. 15.550.000,50.

Jenis atribut tergantung pada beberapa property (sifat) berikut, yang mana yang dimiliki (Vulandari, 2017).

1. Distinctness : = ≠
2. Order : < >
3. Addition : + -
4. Multiplication : * /
5. Nominal attribute : distinctness
6. Ordinal attribute : distinctness & order
7. Interval attribute : distinctness, order & addition
8. Ratio attribute : all 4 properties

Karakteristik dari nilai atribut dijelaskan pada tabel 4.1 – 4.2.

Tabel 4.1: Deskripsi karakteristik masing-masing tipe atribut (P. Tan, M. Steinbach and V. Kumar, 2006) dalam (Vulandari, 2017)

Tipe Atribut	Deskripsi	Contoh	Operasi
Nominal	Nilai dari atribut nominal hanya nama yang berbeda. Atribut nominal menyediakan hanya informasi yang cukup untuk membedakan satu obyek dari yang lainnya. (=, ≠)	Kode zip, nomor ID karyawan, warna mata, jenis kelamin:{laki-laki, perempuan}	*Mode, Entropy, contingency, correlation, test* X^2
Ordinal	Nilai atribut ordinal menyediakan cukup informasi untuk mengurutkan atau menggolongkan obyek	Tingkat kekerasan mineral {baik, lebih baik, paling baik}, tingkatan, nomor jalan	*Median, percentile, rank correlation, run test, sign test*
Interval	Untuk atribut *interval*, perbedaan antar nilai sangat berarti. Hal ini menyatakan satuan pengukuran.	Tanggal kalender, suhu dalam *Celcius* dan *fahrenheit*	Rata-rata (*mean*), simpangan baku, korelasi *Pearson's test t* dan *F*
Ratio	Untuk variabel *ratio*, baik selisih maupun perbandingan sangat berarti	Suhu dalam *Kelvin*, kuantitas moneter, perhitungan, umur, massa, panjang, arus listrik	Rata-rata *geometric*, rata-rata *harmonc*, presentase variasi

Tabel 4.2: Transformasi dari masing-masing tipe atribut (P. Tan, M. Steinbach and V. Kumar, 2006) dalam (Vulandari, 2017)

Tipe Atribut	Transformasi	Keterangan
Nominal	Semua permutasi dari nilai atribut	Jika semua nomor ID karyawan *reassigned*, apakah itu akan membuat perbedaan?
Ordinal	Perintah menjaga perubahan nilai *new_value = f(old_value) di mana f adalah fungsi monotonic*	Setiap atribut mencakup dugaan baik, lebih baik atau paling baik yang bias ditunjukkan. Hal ini sama dengan tingkat kebaikan oleh nilai {1, 2, 3} atau nilai {0.5, 1, 10}
Interval	*New_value = a*old_value + b, di mana adan b konstan*	Dengan begitu, skla suhu *Fahrenheit* dan *Celcius* adalah berbeda, tergantung pada di mana nilai nol masing-masing dan ukuran rentang satuan
Ratio	*New_value = a*old_value*	Panjang bisa diukur dalam meter atau kaki (*feet*)

4.3 Model Klasifikasi

Permasalahan dalam klasifikasi data dalam Data Mining adalah bagaimana menentukan faktor dominan yang belum diketahui untuk mengambil sebuah keputusan dari sebuah atribut (x) ke salah satu atribut label (y) yang sudah didefinisikan sebelumnya.

Klasifikasi data dalam data mining dapat di modelkan sebagai berikut :

$$Fungsi(f) = x_{ak} \rightarrow y_{al}; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(1)$$

Di mana :

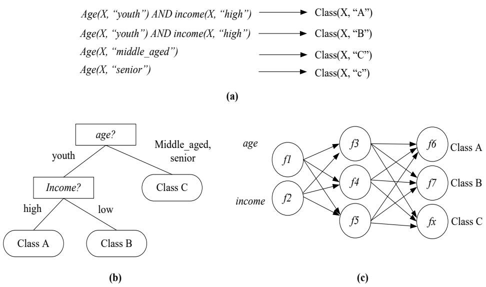
$$x_{ak} \in atribut_{kategorikal} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots.....(2)$$

$$y_{al} \in atribut_{label} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots...(3)$$

Dalam menentukan proses pembelajaran klasifikasi data mining yang dibutuhkan adalah data yang bersifat kategori, dan data kategori yang memiliki sifat label sebagai keputusan (1). x_{ak} adalah atribut data yang bersifat kategorikal yang dapat di mining dalam pembelajaran data mining. Jika data tidak bersifat kategorikal atau tunggal maka data tidak ikut dalam proses pembelajaran (2). y_{al} adalah atribut data yang besifat class label.

Klasifikasi adalah proses menemukan model (atau fungsi) yang menggambarkan dan membedakan kelas atau konsep data. Model diturunkan berdasarkan analisis dari set data pelatihan (yaitu, objek data yang label kelasnya diketahui). Model ini digunakan untuk memprediksi label kelas objek yang label kelasnya tidak diketahui.

"Bagaimana model turunan disajikan?" Model turunan dapat diwakili dalam berbagai bentuk, seperti aturan klasifikasi (mis., aturan IF-THEN), pohon keputusan, rumus matematika, atau jaringan saraf, seperti pada Gambar 4.2 di bawah ini. Pohon keputusan adalah struktur pohon seperti bagan alur, di mana setiap node menunjukkan tes pada nilai atribut, setiap cabang mewakili hasil tes, dan daun pohon mewakili kelas atau distribusi kelas. Pohon keputusan dapat dengan mudah dikonversi menjadi aturan klasifikasi (Han, Kamber and Pei, 2012)



Gambar 4.2: Model klasifikasi dapat di representasikan dalam berbagai bentuk; (a) aturan IF-THEN, (b) Pohon Keputusan, atau (c) Jaringan Saraf Sumber (Han, Kamber and Pei, 2012)

4.4 Algoritma C 4.5

Algoritma C 4.5 adalah salah satu metode untuk membuat decision tree berdasarkan training data yang telah disediakan. Algoritma C 4.5 merupakan pengembangan dari ID3. Beberapa pengembangan yang dilakukan pada C 4.5 adalah sebagai antara lain bisa mengatasi missing value, bisa mengatasi continue data, dan pruning. (Rokach and Maimon, 2012). Pohon keputusan adalah hasil dari proses perhitungan Entropy dan information gain, setelah perhitungan berulang-ulang sampai semua atribut pohon memiliki kelas dan tidak bisa lagi dilakukan proses perhitungan. (Cynthia and Ismanto, 2018).

Variabel tujuan biasanya dikelompokkan dengan pasti dan model pohon keputusan lebih mengarah pada perhitungan probability dari tiap-tiap record terhadap kategori-kategori tersebut atau untuk mengklasifikasi record dengan mengelompokkannya dalam satu kelas. Pohon keputusan juga dapat digunakan untuk mengestimasi nilai dari variabel continue meskipun ada beberapa teknik yang lebih sesuai untuk kasus ini (Hartama, 2011). Banyak algoritma yang dapat dipakai dalam pembentukan pohon keputusan, antara lain ID3, CART, dan C4.5 (Larose, 2005).

Data dalam pohon keputusan biasanya dinyatakan dalam bentuk tabel dengan atribut dan record. Atribut menyatakan suatu parameter yang dibuat sebagai kriteria dalam pembentukan pohon. Misalkan untuk menentukan main tenis, kriteria yang diperhatikan adalah cuaca, suhu, kelembaban dan berangin. Mengubah *tree* yang dihasilkan dalam beberapa *rule*. Jumlah rule sama dengan jumlah path yang mungkin dapat dibangun dari *root* sampai *leaf node*.

Tree Pruning dilakukan untuk menyederhanakan tree sehingga akurasi dapat bertambah. Pruning ada dua pendekatan, yaitu (Cynthia and Ismanto, 2018) :

a. *Pre-pruning*, yaitu menghentikan pembangunan suatu subtree lebih awal (yaitu dengan memutuskan untuk tidak lebih jauh mempartisi data training). Saat seketika berhenti, maka node berubah menjadi leaf (node akhir). Node akhir ini menjadi kelas yang paling sering muncul di antara subset sampel.
b. *Post-pruning*, yaitu menyederhanakan tree dengan cara membuang beberapa cabang subtree setelah tree selesai dibangun. Node yang jarang dipotong akan menjadi leaf (node akhir) dengan kelas yang paling sering muncul.

Untuk memudahkan penjelasan mengenai algoritma C 4.5 berikut ini disertakan contoh kasus yang dituangkan dalam Tabel 4.3

Tabel 4.3: Himpunan Kasus Bermain Tenis (Vulandari, 2017)

NO	Cuaca	Suhu	Kelembaban	Berangin	Main
1	Cerah	Panas	Tinggi	Salah	Tidak
2	Cerah	Panas	Tinggi	Benar	Tidak
3	Berawan	Panas	Tinggi	Salah	Ya
4	Hujan	Sejuk	Tinggi	Salah	Ya
5	Hujan	Dingin	Normal	Salah	Ya
6	Hujan	Dingin	Normal	Benar	Ya
7	Berawan	Dingin	Normal	Benar	Ya
8	Cerah	Sejuk	Tinggi	Salah	Tidak
9	Cerah	Dingin	Normal	Salah	Ya
10	Hujan	Sejuk	Normal	Salah	Ya
11	Cerah	Sejuk	Normal	Benar	Ya
12	Berawan	Sejuk	Tinggi	Benar	Ya
13	Berawan	Panas	Normal	Salah	Ya
14	Hujan	Sejuk	Tinggi	Benar	Tidak

Tabel 4.3 di atas, dapat dijelaskan klasifikasi adalah tugas pembelajaran sebuah fungsi target *f* yang memetakan setiap himpunan atribut *x* ke salah satu label kelas *y* yang telah didefinisikan sebelumnya.

$$Fungsi(f) \longrightarrow x, y : x \in atribut : y \in label$$

Atribut tabel di atas adalah : Cuaca, Suhu, Kelembaban dan Berangin

Label tabel di atas adalah : Main

Dalam kasus yang tertera pada Tabel 4.3 akan dibuat pohon keputusan untuk menentukan main tenis atau tidak dengan melihat keadaan cuaca, suhu, kelembaban dan berangin. Secara Umum Algoritma C 4.5 untuk membangun pohon keputusan adalah sebagai berikut :

Berikut ini algoritma dasar dari C4.5 (Cynthia and Ismanto, 2018):

Input : sampel training, label training, atribut

Output : Pohon Keputusan

1. Membuat simpul akar untuk pohon yang dibuat
2. Jika semua sampel positif, berhenti dengan suatu pohon dengan satu simpul akar, beri tanda (+)
3. Jika semua sampel negatif, berhenti dengan suatu pohon dengan satu simpul akar, beri tanda (-)
4. Jika atribut kosong, berhenti dengan suatu pohon dengan suatu simpul akar, dengan label sesuai nilai yang terbanyak yang ada pada label training
5. Untuk yang lain, Mulai
   (a) A ------ atribut yang mengklasifikasikan sampel dengan hasil terbaik (berdasarkan Gain rasio)
   (b) Atribut keputusan untuk simpul akar ----- A
   (c) Untuk setiap nilai, *vi*, yang mungkin untuk A
      1) Tambahkan cabang di bawah akar yang berhubungan dengan A= vi
      2) Tentukan sampel *Svi* sebagai subset dari sampel yang mempunyai nilai vi untuk atrribut A
      3) Jika sampel *Svi* kosong
         - Di bawah cabang tambahkan simpul daun dengan label = nilai yang terbanyak yang ada pada label training
         - Yang lain tambah cabang baru di bawah cabang yang sekarang C4.5 (sampel training, label training, atribut-[A])
   (d) Berhenti

Untuk mililih atribut sebagai akar, didasarkan pada nilai gain tertinggi dengan atribut-atribut yang ada (Vulandari, 2017):

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} -\frac{|S_i|}{|S|} * Entropy(S_i)$$

S	: Himpunan kasus
A	: Atribut
n	: jumlah partisi atribut A
$\| S_i \|$	: jumlah kasus pada partisi ke-i
|S|	: jumlah kasus dalam S

Sedangkan perhitungan nilai Entropy dapat dilihat pada rumus di bawah ini.

$$Entropy(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Dengan variabel :

S	: Himpunan kasus
n	: Jumlah partisi S
p_i	: Proporsi dari terhadap S

Berikut ini adalah penjelasan *step by step* mengenai masing masing langkah iterasi pembentukan *node* dan *leaf* dalam pembentukan pohon keputusan dengan menggunakan algoritma C 4.5 dari Pola Bermain Tenis pada Tabel di atas.

1. Himpunan Kasus adalah : 14 Kasus
2. Atribut dibagi 2 : Atribut Data yaitu : Cuaca, Suhu, Kelembaban dan Berangin dan Atribut Label adalah Main
3. Jumlah Partisi Atribut n adalah : Klasifikasi dari Himpunan data Kategorikal Distinc
4. Jumlah kasus pada partisi ke-i : Klasifikasi Jumlah kasus yang sama Pada Kolom terhadap label.
5. Jumlah kasus dalam S : Total Klasifikasi Jumlah kasus yang sama Pada Kolom

Hasil Klasifikasi dari Hubungan Algoritma untuk mencari Entropy dan Gain dengan Tabel 4.3 di atas dapat dilihat pada Tabel 4.4. di bawah ini

Tabel 4.4: Klasifikasi dataset dalam menghitung nilai Entropy dan Gain

***Node* 1**		**Jlh Kasus**	**Ya**	**Tidak**	***Entropy***	***Gain***
		(S)	(S_1)	(S_2)		
Total		14	10	4		
Cuaca	Berawan	4	4	0		
	Hujan	5	4	1		
	Cerah	5	2	3		
	Total *Gain* Cuaca					
Suhu	Dingin	4	4	0		
	Panas	4	2	2		
	Sejuk	6	4	2		
	Total *Gain* Suhu					
Kelembaban	Tinggi	7	3	4		
	Normal	7	7	0		
	Total *Gain* Kelembaban					
Berangin	Salah	8	6	2		
	Benar	6	2	4		
	Total *Gain* Berangin					

Proses perhitungan manual dari Tabel 4.4.di atas berdasarkan nilai Entropy dan Gain dapat dilihat di bawah ini .

Hitungan Manual di Node 1. (Iterasi 1)

1. Mengitung Entropy Total dari Jumlah kasus rumusnya adalah :

$$Entropy(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Entropy [Total] = (-(S_2/S) * ($\log_2$(S_2/S)) + (-(S_1/S) * ($\log_2$(S_1/S)) di mana

S2 : Label Jlh Keputusan Tidak
S1 : Label Jlh Keputusan Ya
S : Jumlah Kasus

Entropy [Total] = (-(4/14) * ($\log_2$(4/14)) + (-(10/14) * ($\log_2$(10/14))
= **0,863120569**

1. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Cuaca.

Entropy (Cuaca, Berawan) = (-(0/4) * ($\log_2$ (0/4)) + (-(4/4) * ($\log_2$(4/4)) = **0**
Entropy (Cuaca, Hujan) = (-(1/5)*($\log_2$(4/5)) + (-(1/5) * ($\log_2$(4/5)) = **0,72192**
Entropy (Cuaca, Cerah) =(-(3/5) * ($\log_2$(3/5)) + (-(3/5) * ($\log_2$ (2/5)) = **0,9709**

2. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Cuaca

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$ di mana

S : Jumlah Kasus
A : Atribut
Si : Jumlah Kasus pada Kolom Atribut, sehingga *Gain* [S,Cuaca]
= **0,863120569**– ((4/14) * 0) + ((5/14) ***0,72192**) + ((5/14) * **0,9709**)
= **0,258521037**

3. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Suhu.

Entropy (Suhu, Dingin) = (-(0/4) * ($\log_2$ (0/4)) + (-(4/4) * ($\log_2$(4/4)) = **0**
Entropy (Suhu, Panas) =(-(2/4) * ($\log_2$ (2/4)) + (-(2/4) * ($\log_2$(2/4)) = 1

Entropy (Suhu, Sejuk) =(-(2/6) * ($\log_2$ (2/6)) + (-(4/6) * ($\log_2$ (4/6))
= 0,918295834

4. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Suhu

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Gain [S,Suhu]

= 0,863120569– ((4/14) * 0) + ((4/14) *1) + ((6/14) *0,918295834)
= 0,183850925

5. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Kelembaban

Entropy (Kelembaban,Tinggi)
= (-(4/7) * ($\log_2$ (4/7)) + (-(3/7) * ($\log_2$ (3/7)) = 0,985228136

Entropy (Kelembaban,Normal)

=(-(0/7) * ($\log_2$ (0/7)) + (-(7/7) * ($\log_2$ (7/7)) = 0

6. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Kelembaban

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Gain [S,Kelembaban]

= **0,863120569**– ((7/14) * 0,985228136) + ((7/14) *0)) = 0,370506501

7. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Berangin

Entropy (Berangin,Salah)

= (-(2/8) * ($\log_2$ (2/8)) + (-(6/8) * ($\log_2$ (6/8)) = 0,811278124

Entropy (Berangin,Benar)

=(-(4/6) * ($\log_2$ (4/6)) + (-(2/6) * ($\log_2$ (2/6)) = 0,918295834

8. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Berangin

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Gain [S,Berangin]

= **0,863120569**– ((8/14) * 0,811278124) + ((6/14) *0,918295834))

= 0,005977711

Setelah di Hitung Semua Entropy dan Gain, Kemudian dicari Gain Tertinggi dari Seluruh Gain dari Himpunan Semesta, seperti Tabel 4.5 di bawah ini :

Tabel 4.5: Hasil Perhitungan Nilai Entropy dan Gain pada Node 1

		Jml Kasus	Ya	Tidak	*Entropy*	Gain
		(S)	(S1)	(S2)		
***node* 1**		14	10	4	0,863120569	
Cuaca						
	Berawan	4	4	0	0	
	Hujan	5	4	1	0,721928095	
	Cerah	5	2	3	0,970950594	
						0,258521037
Suhu						
	Dingin	4	4	0	0	
	Panas	4	2	2	1	
	Sejuk	6	4	2	0,918295834	
						0,183850925
Kelembaban						

	Tinggi	7	3	4	0,985228136	
	Normal	7	7	0	0	
						0,370506501
Berangin						
	Salah	8	6	2	0,811278124	
	Benar	6	2	4	0,918295834	
						0,005977711

Dari hasil pada Tabel 4.5 dapat diketahui bahwa atribut dengan Gain tertinggi adalah kelembaban yaitu sebesar 0,370506501. Dengan demikian kelembaban dapat menjadi node akar. Ada 2 nilai atribut dari kelembaban yaitu tinggi dan normal. Dari kedua nilai atribut tersebut, nilai atribut normal sudah mengklasifikasikan kasus menjadi 1 yaitu keputusannya Ya, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut tinggi masih perlu dilakukan perhitungan lagi.

Dari hasil tersebut dapat digambarkan pohon keputusan sementara, tampak seperti Gambar 4.3

Gambar 4.3: Pohon Keputusan Node 1 (root node)

Hitungan Manual di Node 1.1 (Iterasi 2)

Menghitung jumlah kasus, jumlah kasus untuk keputusan Ya, jumlah kasus untuk keputusan Tidak, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut cuaca, temperatur dan angin yang dapat menjadi node akar dari nilai atribut tinggi. Setelah itu lakukan penghitungan Gain untuk masing-masing atribut. Hasil penambangan data di iterasi ke 2 di filter berdasarkan kelembaban yang tinggi ditunjukkan oleh Tabel 4.6

Tabel 4.6: Himpunan Bermain Tenis Berdasarkan Kelembaban Tinggi

NO	Cuaca	Suhu	Kelembaban	Berangin	Main
1	Cerah	Panas	Tinggi	Salah	Tidak
2	Cerah	Panas	Tinggi	Benar	Tidak
3	Berawan	Panas	Tinggi	Salah	Ya
4	Hujan	Sejuk	Tinggi	Salah	Ya
5	Cerah	Sejuk	Tinggi	Salah	Tidak
6	Berawan	Sejuk	Tinggi	Benar	Ya
7	Hujan	Sejuk	Tinggi	Benar	Tidak

Dari Tabel 4.6 di atas kita akan membagi atau melakukan Klasifikasi dari Pola Bermain Tenis pada Tabel 4.6 di atas.

1. Himpunan Kasus adalah : 7 Kasus
2. Atribut dibagi 2 : Atribut Data yaitu : Cuaca, Suhu, Kelembaban dan Berangin dan Atribut Label adalah Main
3. Jumlah Partisi Atribut n adalah : Klasifikasi dari Himpunan data Kategorikal Distinc
4. Jumlah kasus pada partisi ke-i : Klasifikasi Jumlah kasus yang sama Pada Kolom terhadap label.
5. Jumlah kasus dalam S : Total Klasifikasi Jumlah kasus yang sama Pada Kolom

Hasil Klasifikasi kategorikal atribut dari Hubungan Algoritma Mencari Gain dan Entropy dengan Tabel 4.6 di atas adalah pada Tabel 4.7 di bawah ini:

Tabel 4.7: Klasifikkasi Kategori Atribut Kelembaban bernilai Tinggi

		Kelembaban Tinggi	Ya	Tidak	*Entropy*	*Gain*
		(S)	(S_1)	(S_2)		
***node* 2 (1.1)**		7	3	4		
Cuaca						
	Berawan	2	2	0		

	Hujan	2	1	1		
	Cerah	3	0	3		
	Total *Gain* Cuaca					
Suhu						
	Dingin	0	0	0		
	Panas	3	1	2		
	Sejuk	4	2	2		
	Total *Gain* Suhu					
Berangin						
	Salah	4	2	2		
	Benar	3	2	1		
	Total *Gain* Berangin					

Hitungan Manual untuk mencari *Entropy* Total, *Entropy* atribut dan *Gain* Total di bawah ini.

1. *Entropy* Kelembaban Tinggi
   a. Jumlah Kasus (S) : 7 Kasus
   b. Jumlah S(2) Tidak : 4 kasus
   c. Jumlah S(1) Ya : 3 kasus

$$Entropy(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Entropy [Total] = (-(S_2/S) * ($\log_2$(S_2/S)) + (-(S1/S) * ($\log_2$(S_1/S)) , di mana

S2 :Label Jlh Keputusan Tidak
S1 :Label Jlh Keputusan Ya
S :Jumlah Kasus

Entropy [Total] = (-(4/7) * ($\log_2$ (4/7)) + (-(3/7) * ($\log_2$ (3/7)) = **0,985228136**

2. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Cuaca.
   Entropy (Cuaca, Berawan) = (-(0/2) * ($\log_2$ (0/2)) + (-(2/2) * ($\log_2$(2/2))
   = **0**
   Entropy (Cuaca, Hujan) =(-(1/2) * ($\log_2$(1/2)) + (-(1/2) * ($\log_2$ (1/2)) = **1**

Entropy (Cuaca, Cerah) =(-(3/3) * ($\log_2$ (3/3)) + (-(0/3) * ($\log_2$(0/3)) = **0**

3. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Cuaca

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Gain [S,Cuaca] = ***0,985228136***– ((2/7) * 0) + ((2/7) *1) + ((3/7) * 0)

= **0,69951385**

4. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Suhu.

Entropy (Suhu, Dingin) = (-(0/0) * ($\log_2$(0/0)) + (-(0/0) * ($\log_2$(0/0)) = **0**

Entropy (Suhu, Panas) =(-(2/3) * ($\log_2$ (2/3)) + (-(1/3) * ($\log_2$ (1/3))
= 0,918295834

Entropy (Suhu, Sejuk) =(-(2/4) * ($\log_2$ (2/4)) + (-(2/4) * ($\log_2$(2/4)) = 1

5. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Suhu

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Gain [S,Suhu]

= **0,985228136**– ((0/7) * 0) + ((3/7) *0,918295834) + ((4/7) *1)

= **0,020244207**

6. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Berangin

Entropy (Berangin,Salah)

= (-(2/4) * ($\log_2$(2/4)) + (-(2/4) * ($\log_2$(2/4)) = 1

Entropy (Berangin,Benar)

=(-(1/3) * ($\log_2$ (1/3)) + (-(2/3) * ($\log_2$(2/3)) = 0,918295834

7. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Berangin

$$Gain(S, A) = Enthrophy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Enthropy(S_i)$$

Gain [S,Berangin]

= **0,985228136**– ((4/7) * 1) + ((3/7) *0,918295834)) = 0,020244207

Setelah di Hitung Semua Entropy dan Gain, Kemudian dicari Gain Tertinggi dari Seluruh Gain dari Himpunan Semesta, seperti Tabel 4.8. di bawah ini :

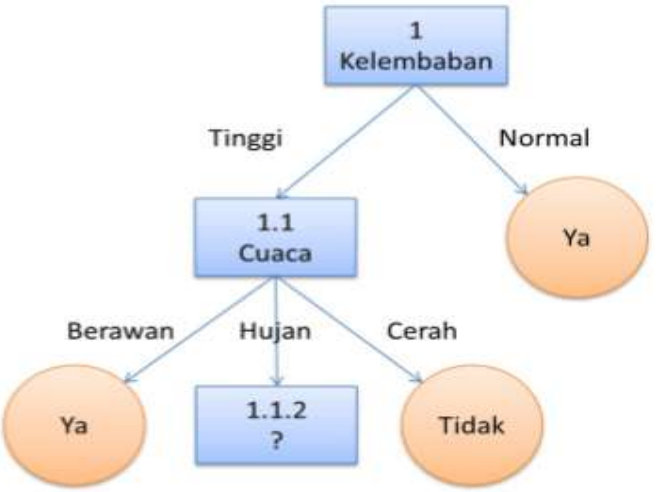
Tabel 4.8: Hasil Perhitungan Entropy dan Gain pada Node 1.1.

		Kelembaban Tinggi	Ya	Tidak	*Entropy*	*Gain*
***node* 2 (1.1)**		(S)	(S1)	(S2)		
Kelembaban Tinggi		7	3	4	0,985228136	
Cuaca						
	Berawan	2	2	0	0	
	Hujan	2	1	1	1	
	Cerah	3	0	3	0	
						0,69951385
Suhu						
	Dingin	0	0	0	0	
	Panas	3	1	2	0,918295834	
	Sejuk	4	2	2	1	
						0,020244207
Berangin						
	Salah	4	2	2	1	
	Benar	3	2	1	0,918295834	
						0,020244207

Dari hasil pada Tabel 4.8 dapat diketahui bahwa atribut dengan Gain tertinggi adalah cuaca yaitu sebesar 0.699. Dengan demikian cuaca dapat menjadi node cabang dari nilai atribut tinggi. Ada 3 nilai atribut dari cuaca yaitu mendung, hujan dan cerah. dari ketiga nilai atribut tersebut, nilai atribut mendung sudah

mengklasifikasikan kasus menjadi 1 yaitu keputusannya Ya dan nilai atribut cerah sudah mengklasifikasikan kasus menjadi satu dengan keputusan Tidak, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut hujan masih perlu dilakukan perhitungan lagi.

Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada Gambar 4.4 berikut:



Gambar 4.4: Pohon Keputusan Node 1.1

Hitungan Manual di Node 1.1.2 (Iterasi 3)

Menghitung jumlah kasus, jumlah kasus untuk keputusan Ya, jumlah kasus untuk keputusan Tidak, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut suhu dan angin yang dapat menjadi node cabang dari nilai atribut hujan. Setelah itu lakukan penghitungan Gain untuk masing-masing atribut. Pada penambangan data di iterasi ke 3 ini di filter berdasarkan cuaca hujan dan kelembaban tinggi, seperti pada Tabel 4.9 di bawah ini.

Tabel 4.9: Himpunan Bermain Tenis Berdasarkan Kelembaban Tinggi dan Cuaca Hujan

NO	Cuaca	Suhu	Kelembaban	Berangin	Main
1	Hujan	Sejuk	Tinggi	Salah	Ya

2	Hujan	Sejuk	Tinggi	Benar	Tidak

Penjelasan dari Algoritma dan Tabel 4.9

Dari Tabel 4.9. di atas Kita akan membagi atau melakukan Klasifikasi dari Pola Bermain Tenis

Himpunan Kasus adalah : 2 Kasus

1. Atribut dibagi 2 : Atribut Data yaitu : Suhu dan Berangin dan Atribut Label adalah Main
2. Jumlah Partisi Atribut n adalah : Klasifikasi dari Himpunan data Kategorikal Distinc
3. Jumlah kasus pada partisi ke-i : Klasifikasi Jumlah kasus yang sama Pada Kolom terhadap label.
4. Jumlah kasus dalam S : Total Klasifikasi Jumlah kasus yang sama Pada Kolom

Hasil Klasifikasi dari Hubungan Algoritma Mencari Gain dengan Tabel 4.9 Di atas, dapat dilihat pada Table 4.10 di bawah ini adalah :

Tabel 4.10: Klasifikasi data kategori Kelembaban Tinggi dan Cuaca Hujan

		Kelembaban Tinggi dan Cuaca Hujan	**Ya**	**Tidak**	***Entropy***	***Gain***
		(S)	(S_1)	(S_2)		
***node* 2 (1.1)**		2	1	1		
Suhu						
	Dingin	0	0	0		
	Panas	0	0	0		
	Sejuk	2	1	1		
Berangin						
	Salah	1	1	0		
	Benar	1	0	1		

Pada iterasi Ke 3 ini yang pertama di hitung adalah :

1. *Entropy* Kelembaban Tinggi
   a. Jumlah Kasus (S) : 2 Kasus
   b. Jumlah S(2) Tidak : 1 kasus
   c. Jumlah S(1) Ya : 1 kasus

$$Entropy(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Entropy [Total] = (-(S_2/S) * ($\log_2$ (S_2/S)) + (-(S_1/S) * ($\log_2$ (S_1/S))

S2 : (Label Jumlah Keputusan Tidak)
S1 : (Label Jumlah Keputusan Ya)
S : (Jumlah Kasus)

Entropy [Total] = (-(1/2) * ($\log_2$ (1/2)) + (-(1/2) * ($\log_2$ (1/2)) = **1**

1. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Suhu.
   Entropy (Suhu, Dingin) = (-(0/0) * ($\log_2$ (0/0)) + (-(0/0) * ($\log_2$(0/0)) = **0**
   Entropy (Suhu, Panas) = (-(0/0) * ($\log_2$ (0/0)) + (-(0/0) * ($\log_2$ (0/0)) = **0**
   Entropy (Suhu, Sejuk) =(-(1/2) * ($\log_2$ (1/2)) + (-(1/2) * ($\log_2$(1/2)) = 1

2. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Suhu

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Gain [S,Suhu] = 1– ((0/2) * 0) + ((0/2) *0) + ((2/2) *1) = **0**

3. Menghitung *Entropy* Untuk Setiap Klasifikasi dari Atribut Berangin

Entropy (Berangin,Salah) = (-(0/1) * ($\log_2$(0/1)) + (-(1/1) * ($\log_2$(1/1)) = 0

Entropy (Berangin,Benar) = (-(1/1) * ($\log_2$(1/1)) + (-(0/1) * ($\log_2$(0/1)) = 0

4. Mencari *Gain* Total Semesta dari *Gain* Total Atribut Berangin

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Gain [S, Berangin] = **1**– ((1/2) * 0) + ((1/2) *0)) = 1

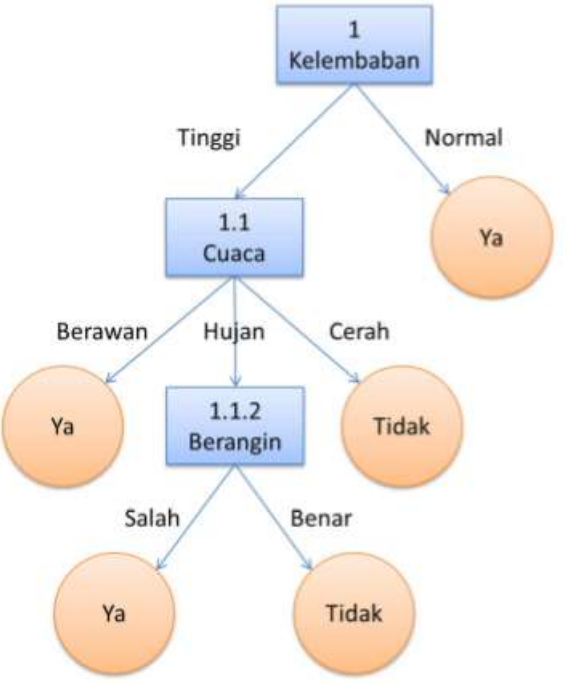
Setelah itu lakukan penghitungan Gain dan Entropy untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh Tabel 4.11

Tabel 4.11: Hasil Perhitungan Nilai Entropy dan Gain pada Node 1.1.2

		Kelembaban Tinggi dan Cuaca Hujan	**Ya**	**Tidak**	***Entropy***	***Gain***
***node* 2 (1.1.2)**		(S)	(S_1)	(S_2)		
Kelembaban Tinggi dan Cuaca Hujan		2	1	1	1	
Suhu						
	Dingin	0	0	0	0	
	Panas	0	0	0	0	
	Sejuk	2	1	1	1	
						0
Berangin						
	Salah	1	1	0	0	
	Benar	1	0	1	0	
						1

Dari hasil pada Tabel 4.11 dapat diketahui bahwa atribut dengan Gain tertinggi adalah angin yaitu sebesar 1. Dengan demikian angin dapat menjadi node cabang dari nilai atribut hujan. Ada 2 nilai atribut dari angin yaitu Salah dan Benar. Dari kedua nilai atribut tersebut, nilai atribut Salah sudah mengklasifikasikan kasus menjadi 1 yaitu keputusannya Ya dan nilai atribut Benar sudah mengklasifikasikan kasus menjadi satu dengan keputusan Tidak, sehingga tidak perlu dilakukan perhitungan lebih lanjut untuk nilai atribut ini.

Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada Gambar 4.5.



Gambar 4.5: Pohon Keputusan Hasil Perhitungan Node 1.1.2

Dengan memperhatikan pohon keputusan pada Gambar 4.5, diketahui bahwa semua kasus sudah masuk dalam kelas. Dengan demikian, pohon keputusan pada Gambar 4.5. merupakan pohon keputusan terakhir yang terbentuk.

4.5 Penerapan Algoritma C 4.5

Penerapan Algoritma dalam Rapidminer sebagai Tools yang digunakan untuk memvalidasi hitungan manual. RapidMiner merupakan perangakat lunak yang bersifat terbuka (open source). RapidMiner adalah sebuah solusi untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi.

RapidMiner menyediakan GUI (Graphic User Interface) untuk merancang sebuah pipeline analitis. GUI ini akan menghasilkan file XML (Extensible Markup Language) yang mendefenisikan proses analitis keingginan pengguna untuk diterpkan ke data. File ini kemudian dibaca oleh RapidMiner untuk menjalankan analis secara otomatis. (denis Aprilia dkk)

Pada bagian ini ini kita akan membahas penggunaan RapidMiner untuk mengelola dataset seperti pada Tabel 4.12 di bawah ini.

Table 4.12: Dataset Keputusan bermain Tenis

NO	Cuaca	Suhu	Kelembaban	Berangin	Main
1	Cerah	Panas	Tinggi	Salah	Tidak
2	Cerah	Panas	Tinggi	Benar	Tidak
3	Berawan	Panas	Tinggi	Salah	Ya
4	Hujan	Sejuk	Tinggi	Salah	Ya
5	Hujan	Dingin	Normal	Salah	Ya
6	Hujan	Dingin	Normal	Benar	Ya
7	Berawan	Dingin	Normal	Benar	Ya
8	Cerah	Sejuk	Tinggi	Salah	Tidak
9	Cerah	Dingin	Normal	Salah	Ya
10	Hujan	Sejuk	Normal	Salah	Ya
11	Cerah	Sejuk	Normal	Benar	Ya
12	Berawan	Sejuk	Tinggi	Benar	Ya
13	Berawan	Panas	Normal	Salah	Ya
14	Hujan	Sejuk	Tinggi	Benar	Tidak

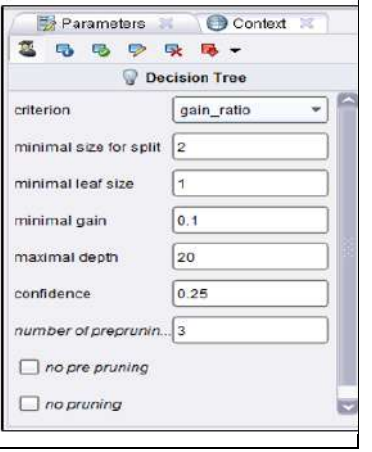
Atribut target dari Tabel 4.12 di atas adalah main. Atribut data adalah Cuaca, Suhu, Kelembaban dan Berangin.

Dalam penerapan Algoritma C 4.5 menggunakan rapid miner dibagi tiga, yaitu:

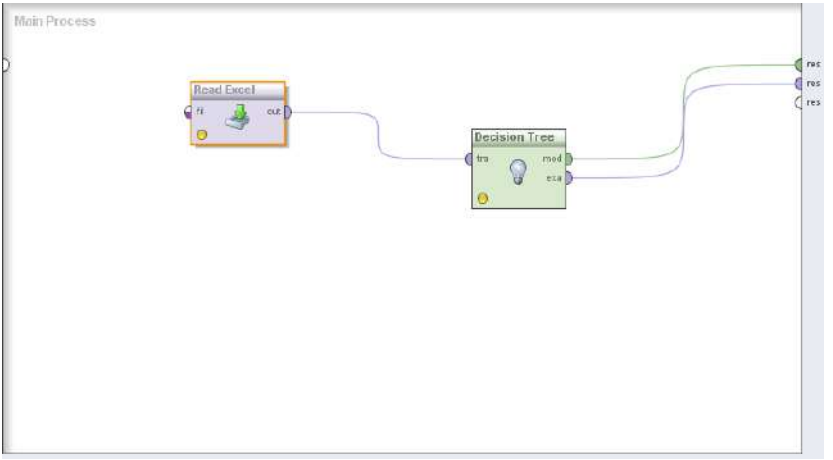
a. Menggunakan dasar Decision Tree

Software dan Operator RapidMinner dapat dilihat seperti pada Tabel 4.13 di bawah ini.

Tabel 4.13: Konfigurasi Penggunaan Operator RapidMiner Decision Tree

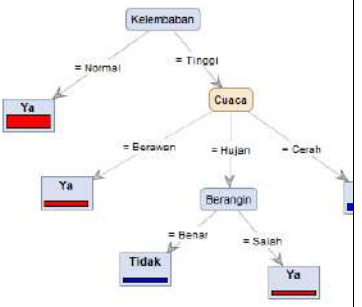
NO	*Software*	*Action*	Konfigurasi
1	Microsoft Excel	Data Set	Seperti pada tabel x.y
2	*RapidMinner*	Operator	*Import*, Data dan *Read* Excel
			1. Excel File : Mimilih File Excel yang dijadikan dataset 2. *Import Configuration wizard* (menentukan atribut data dan atribut label)
			

Configurasi Main Proses pada RapidMiner bisa dilihat pada gambar 4.6. di bawah ini.



Gambar 4.6: Configurasi Decesion Tree Main Proses

Hasil dari Eksekusi Pengolahan Pola Data Mining Bermain tenis seperti pada tabel di atas dapat dilihat pada Gambar 4.7 di bawah ini.

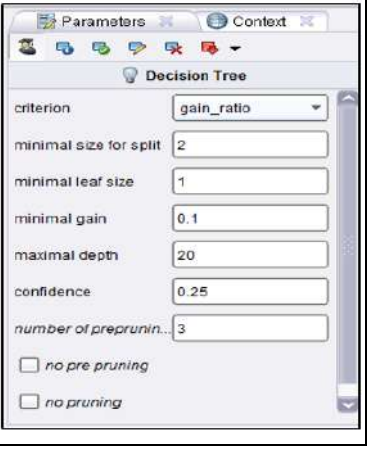
	Tree Kelembaban = Normal: Ya {Tidak=0, Ya=7} Kelembaban = Tinggi | Cuaca = Berawan: Ya {Tidak=0, Ya=2} | Cuaca = Cerah: Tidak {Tidak=3, Ya=0} | Cuaca = Hujan | | Berangin = Benar: Tidak {Tidak=1, Ya=0} | | Berangin = Salah: Ya {Tidak=0, Ya=1}
a. Hasil Model Grafik	b. Hasil Model *Text*

Gambar 4.7: Hasil Eksekusi Pohon Keputusan Bermain Tenis

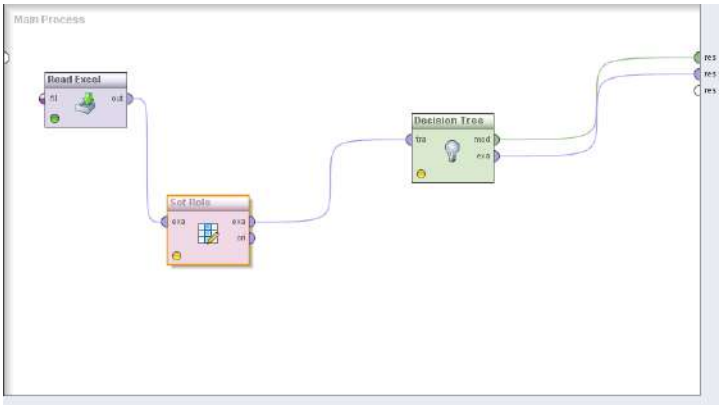
b. Menggunakan Set Role-Decesion Tree

Sofware dan Operator RapidMinner dapat dilihat seperti pada Tabel 4.14 di bawah ini.

Tabel 4.14: Konfigurasi Operator RapidMiner Set Role Decision Tree

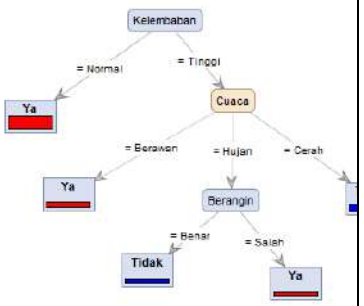
NO	***Software***	***Action***	**Konfigurasi**
1	Microsoft Excel	Data Set	Seperti pada tabel x.y
2	*RapidMinner*	Operator	
			1. Excel File : Mimilih File Excel yang dijadikan dataset 2. *Import Configuration wizard* (menentukan atribut data dan atribut label)
			1. Atribut *Name* : main 2. Target *role* : Label
			

Configurasi Main Proses pada Rapid Miner bisa dilihat pada gambar 4.8 di bawah ini.



Gambar 4.8: Konfigurasi Decesion Tree Main Proses Menggunakan Set Role

Hasil dari Eksekusi Pengolahan Pola Data Mining Bermain tenis seperti pada tabel di atas dapat dilihat pada Gambar 4.9 di bawah ini.

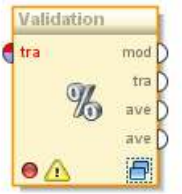
	*Tree* Kelembaban = Normal: Ya {Tidak=0, Ya=7} Kelembaban = Tinggi | Cuaca = Berawan: Ya {Tidak=0, Ya=2} | Cuaca = Cerah: Tidak {Tidak=3, Ya=0} | Cuaca = Hujan | | Berangin = Benar: Tidak {Tidak=1, Ya=0} | | Berangin = Salah: Ya {Tidak=0, Ya=1}
a. Hasil Model Grafik	b. Hasil Model *Text*

Gambar 4.9: Hasil Eksekusi Pohon Keputusan Bermain Tenis

c. Menggunakan Validasi Decesion Tree

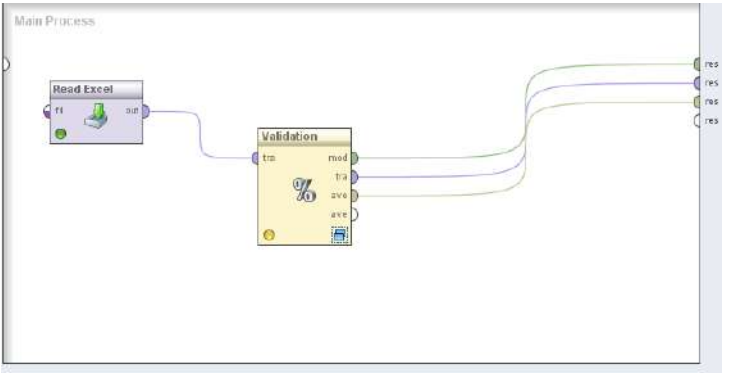
Sofware dan Operator RapidMinner dapat dilihat seperti pada Tabel 4.15 di bawah ini.

Tabel 4.15: Konfigurasi Operator RapidMiner X-Validation

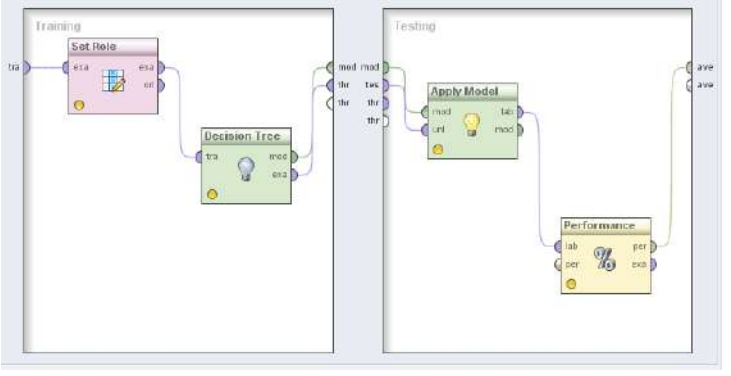
NO	*Software*	*Action*	Konfigurasi
1	Microsoft Excel	Data Set	Seperti pada tabel x.y
2	*RapidMinner*	Operator	
			3. Excel File : Mimilih File Excel yang dijadikan dataset 4. *Import Configuration wizard* (menentukan atribut data dan atribut label)
			Operator *X-Validation* merupakan operator bersarang yang memiliki dua subproses: *training* subprocess (subproses percobaan) dan *testing* subprocess (subproses pengujian). Subproses percobaan digunakan untuk melatih sebuah model. Model yang terlatih kemudian diterapkan dalam subproses pengujian. *Configurasi* main Proses dapat dilihat pada gambar x.y dan gambar x.y

Validasi decision tree digunakan untuk melihat keakuratan model pohon keputusan bermain tenis menggunakan algoritma C 4.5 yang menggunakan

software rapidminer. Dalam melakukan validasi rule decision tree, menggunakan utility X-Validation yang digunakan untuk membagi dua area training dan testing data. Dalam testing data menggunakan *utility apply model* dan %performance. Design model validasi rapidminer dapat dilihat pada Gambar 4.10 dan 4.11



Gambar 4.10: Konfigurasi Main Proses X-Validation RapidMiner



Gambar 4.11: Konfigurasi Training dan Testing RapidMiner

Pada Gambar 4.11. dapat dijelaskan untuk menu training terdiri dari Set Role untuk membuat label dengan menggunakan algoritma decision tree. Sementara menu testing menggunakan apply model dan performance. Dari testing didapatkan accuracy 75%, seperti yang terlihat pada Gambar 4.12.

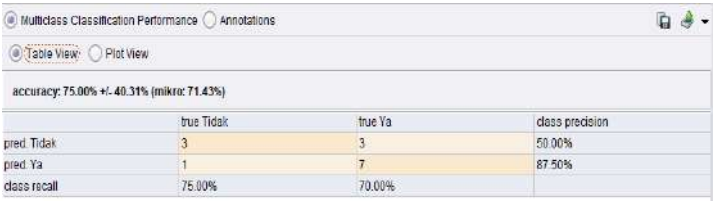

Multiclass Classification Performance Annotations

Table View Plot View

accuracy: 75.00% +/- 40.31% (mikro: 71.43%)

	true Tidak	true Ya	class precision
pred. Tidak	3	3	50.00%
pred. Ya	1	7	87.50%
class recall	75.00%	70.00%	

Gambar 4.12: Hasil Accuracy Validation Training dan Testing RapidMiner

Dari Gambar 4.12 di atas dapat dijelaskan bahwa predikat Tidak mempunyai nilai 3 dengan class precision 50%, dan Ya mempunyai nilai 7 dengan class precision 87,50 %.

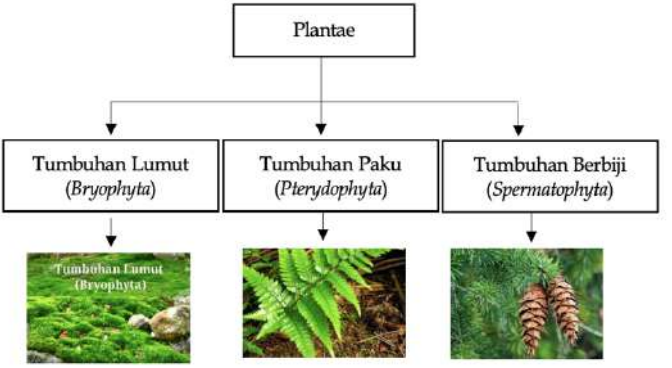
Bab 5

Klasifikasi Citra dengan K-NN

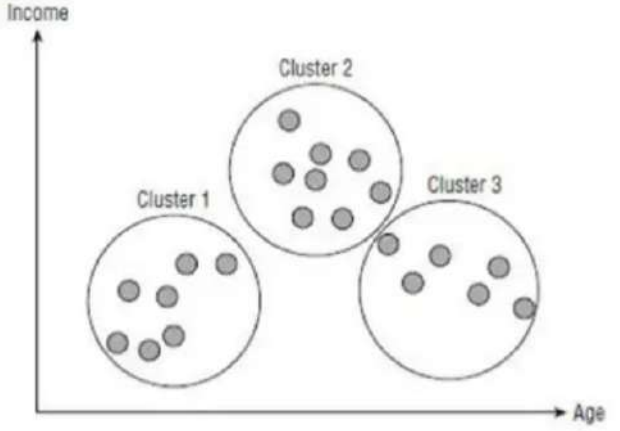
5.1 Klasifikasi Citra

Sering kali kita mendengar kata klasifikasi. Klasifikasi merupakan salah satu topik utama dalam data mining dan machine learning. Apa sebenarnya klasifikasi dan apa perbedaannya dengan klastering? Klasfikasi (clasification) merupakan proses pembelajaran suatu target yang memetakan tiap himpunan ke suatu label kelas yang telah diidentifikasikan sebelumnya. Sedangkan klastering (clustering) adalah metode untuk menemukan objek hingga objek-objek dalam satu kelompok sama (punya hubungan) dengan yang lain dan berbeda (tidak berhubungan) dengan objek-objek dalam kelompok lain (Herawati, 2013). Perbedaan lainnya klasifikasi (clasification) termasuk dalam supervised learning yakni pendekatan yang sudah terdapat data yang dilatih sebelumnya, sedangkan klastering (clustering) termasuk unsupervised learning yakni pendekatan yang tidak menggunakan data latih sebelumnya. Sehingga sederhananya klasifikasi adalah mengelompokkan *attribute* ke label klas yang didefiniskan sebelumnya sedangkan klastering belum ada label kelas yang didefinisikan sebelumnya.

Sebagai contoh perbedaan klasifikasi dengan klastering dapat dilihat pada gambar 5.1 dan gambar 5.2. Pada gambar 5.1 merupakan klasifikasi untuk tumbuhan yang dibagi menjadi tiga (3) klas yaitu tumbuhan lumut, tumbuhan paku dan tumbuhan berbiji sedangkan pada gambar 5.2 klaster antara data pelanggan yang memiliki attribute umur dan pendapatan.



Gambar 5.1: Klasifikasi tumbuhan



Gambar 5.2: Ilustrasi klastering antara data pelanggan yang memilik atribute umur (age) dan pendapatan(income).

Klasifikasi memiliki dua jenis model yaitu ;

1. Pemodelan deskriptif (descriptive modelling) yakni model klasifikasi yang berfungsi alat penjelas guna membedakan objek-objek dari klas yang berbeda. Contoh dari pemodelan deskriptif adalah : struktur organisasi, tabel pengelompokan tanaman.
2. Pemodelan prediktif (predivtive modelling) yaitu model klasifikasi yang digunakan untuk memprediksi label klas untuk record yang tidak

diketahui. Contoh dari pemodelan prediktif : prediksi analisa *Break even point* (BEP), *prediksi inventory* barang retail (Herawati, 2013).

Tahapan Klasifikasi dalam data mining ada tiga tahapan yaitu :

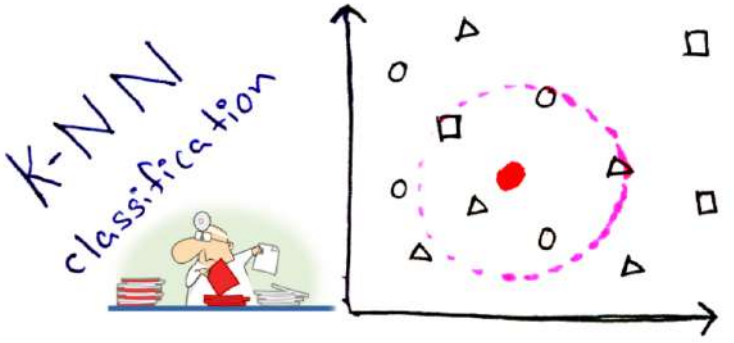
1. Pembangunan model (Model Construction) yakni merepresentasikan rule klasifikasi untuk menyelesaikan masalah klasifikasi berdasarkan data latih atau data training. Data training ini sudah harus memiliki informasi yang lengkap baik attribute maupun classnya.
2. Penerapan model (Model Usage) yakni model yang sebelumnya telah dirancang dipergunakan untuk menentukan attribute/class dari sebuah data baru yang attribute/classnya belum diketahui sebelumnya.
3. Evaluasi pada tahapan ini dilakukan pengujian dari penerapan model yang dirancang dengan menggunakan parameter terukur untuk menentukan apakah model tersebut dapat diterima atau tidak. Tahapan ini dapat dilakukan pengujian nilai akurasi untuk mengetahui tingkat keakuratan penerapan model.

Lalu apakah klasifikasi citra? Klasifikasi citra merupakan suatu proses pengelompokan seluruh piksel pada suatu citra kedalam kelompok sehingga dapat diinterpretasikan sebagai suatu property yang spesifik. (Chang and Ren, 2000). Ada beberapa algoritma standar yang dapat menyelesaikan klasifikasi yaitu: Backpropagation Neural Network, Suport Vector Classification (SVC), K-nearest Neighbor(K-NN), Naive Bayes.

5.2 K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) adalah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terdefinisi (dilabel) sebelumnya. K-NN termasuk dalam kelompok *supervised learning*, di mana hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam K-NN. Kelas yang baru dari suatu data akan dipilih berdasarkan group klasnya yang paling dekat jarak vectornya.

Sebagai gambaran kinerja algoritma K-NN dapat dilihat pada gambar 5.3



Gambar 5.3: Algoritma K-NN, Kelas yang baru dari suatu data akan dipilih berdasarkan group klasnya yang paling dekat jarak vectornya (Informatikalogi, 2017)

Algoritma K-Nearst Neighbor (K-NN) menggunakan klasifikasi dengan nilai ketetanggan sebagai nilai prediksi dari *query instance* yang baru. Dalam hal ini jumlah data terdekat ditentukan oleh user yang dinyatakan dengan *k*, misal user menentukan *k=6*, maka setiap data testing atau data latih dihitung jaraknya terhadap data training dan dipilih 6 data training yang jaraknya paling dekat ke data testing. Kemudian akan periksa outputnya atau labelnya masing-masing, dan kemudian ditentukan output mana yang frekuensinya paling banyak. Tujuan algoritma K-NN adalah untuk mengklasifikasikan objek baru berdasarkan data training sebelumnya.

Untuk menghitung nilai jarak antar dua titik yaitu titik pada data latih dan titik pada data testing maka dapat dipergunakan rumus *Euclidien Distance* seperti berikut:

$$d\ (x,y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}$$

di mana *d* adalah jarak pada data training *x* dan titik data testing y yang akan diklasifikasikan, dan $x = x_1,x_2,\ldots.x_i$ dan $y=y_1,y_2,\ldots.y_i$, di mana I merepresentasikan nilai atribut serta *n* merupakan dimensi atribut (Indrawaty et al., 2018).

Kelebihan dari algoritma K-NN sehingga sering dipergunakan untuk klasifikasi yaitu :

1. Sangat nonlinier

K-NN merupakan salah satu algoritma pembelajaran yang bersifat nonparametrik yakni model yang tidak mengasumsikan apa-apa mengenai distribusi instance di dalam dataset. Nonlinier adalah data yang tidak tetap dan sistem nonlinier memiliki tingkat senstivitas yang tinggi terhadap data yang tidak dapat dikontrol.

2. Mudah dipahami dan diimplementasikan. Untuk mengklasifikasi instance x menggunakan K-NN kita cukup mendefinisikan fungsi untuk menghitung jarak antar-instance. Dengan langkah dan ketersediaan rumus perhitungan jarak yang ada K-NN dapat diimplentasikan untuk klasifikasi teks maupun citra.
3. Memiliki konsistensi yang kuat. Ketika jumlah data mendekati tak hingga algoritma K-NN menjamin error rate yang lebih baik dari bayes error rate.
4. Tangguh terhadap data uji yang noise.

Kekurangan dari algoritma K-NN yaitu (Amroh, 2010):

1. Perlu penentu parameter k (jumlah tetangga terdekat)
2. Pembelajaran berdasarkan jarak tidak jelas mengenai jarak apa yang dipergunakan untuk menghasilkan solusi terbaik
3. Beban komputasi yang cukup tinggi karena perlu menghitung jarak data uji dengan semua data latih.

Tahapan algoritma K-NN yaitu (Informatikalogi, 2017):

1. Menentukan parameter k (jumlah tetangga paling dekat)
2. Menghitung kuadrat jarak dengan menggunakan eucliden objek terhadap data training yang diberikan.

   $d\ (x,y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}$

Di mana :

d adalah jarak antara titik pada data training *x* dan titik data testing *y* yang akan diklasifikasi.

i =nilai atribut

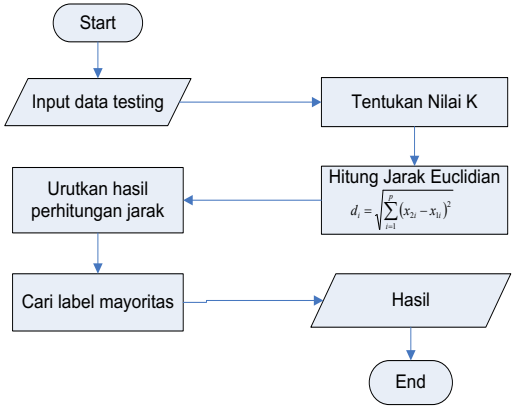
xi = citra query.

Yi=citra database dengan k vektor.

n = dimensi atribut

3. Kemudian mengurutkan objek-objek termasuk kedalam kelompok yang mempunyai jarak terkecil.
4. Mengumpulkan kategori Y (klasifikasi nearest neighbor berdasarkan nilai k)
5. Mencari label Mayoritas untuk menentukan hasil klasifikasi.

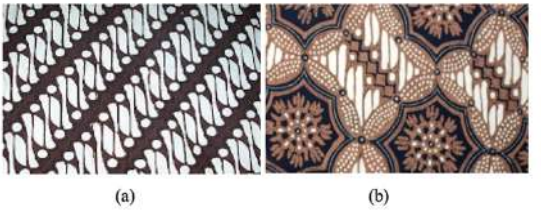
Tahapan algoritma K-NN dalam klasifikasi dapat digambarkan dalam flowchart pada gambar 5.4



Gambar 5.4: Flowchart klasifikasi dengan algoritma K-NN

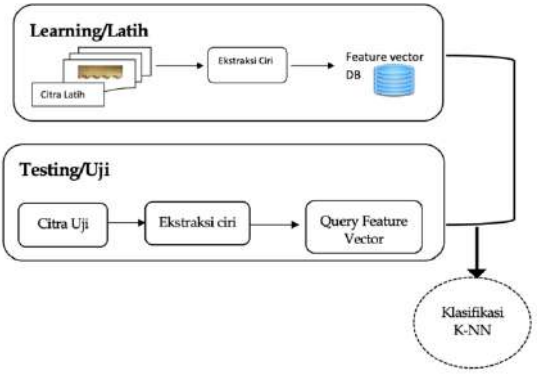
5.3 Studi Kasus Klasifikasi Citra Batik dengan K-NN

Pada klasifikasi citra dengan menggunakan algoritma K-NN, citra yang dipergunakan untuk data latih dan testing adalah citra batik. Motif batik yang beraneka macam jenisnya mencerminkan asal usul daerah dari kain batik tersebut, hal ini terlihat pada ragam hias maupun tata warna. Ragam hiasa, tata warna dan pola dari kain batik yang ada di Indonesia hampir memiliki kemiripan bentuk dan ciri yang sangat dekat. Identifikasi secara visual memerlukan kemampuan penglihatan dan pengetahuan dalam mengklasifikasikan pola yang terbentuk dari citra batik itu sendiri.



Gambar 5.5: Contoh Motif batik, (a) Batik Parang, (b) Batik Ceplok

Contoh klasifikasi citra batik yang akan dibahas yakni pada citra batik Parang yakni mengklasifikasikan citra batik tersebut tergolong batik parang atau bukan. Berikut adalah gambaran secara umum algoritma K-NN dalam klasifikasi citra batik.

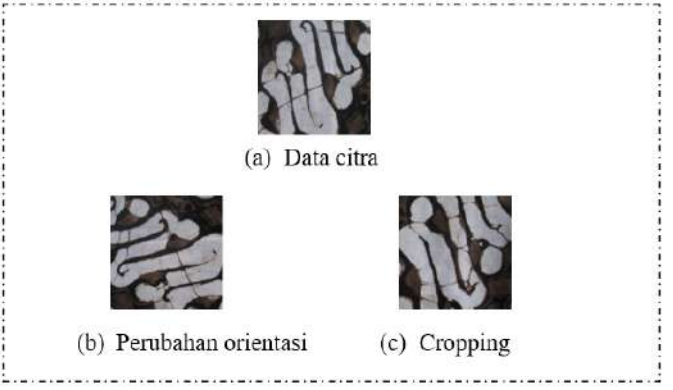


Gambar 5.6: Gambaran umum sistem klasifikasi citra batik

Masukkan merupakan citra yang nantinya sistem akan mencari kemiripan (similarity) antara citra masukkan dengan citra yang terdapat dalam basis data. Keluaran dari sistem berupa klasifikasi jenis batik yang ditentukan. Proses klasifikasi citra akan dilakukan beberapa tahap yaitu dari pemilihan jenis dan sumber data yang akan dijadikan sebagai basis data citra yang kemudian dilakukan proses ekstraksi ciri baru kemudian dilakukan klasifikasi citra.

Karakteristik citra masukan pada studi kasus ini diperoleh dengan menggunakan kamera digital dan citra yang dipergunakan sebagai data uji diperoleh melalui dua mekanisme yaitu :

1. Mengubah orientasi citra dengan merotasi citra pada sudut 900 dan 1800
2. Cropping, yakni memotong gambar menjadi ukuran 256x256 pixel.



Gambar 5.7: (a) data citra asli, (b) perubahan orientasi citra, (c) cropping ukuran citra 256x256 pixel

Untuk proses ekstraksi ciri dapat dipergunakan metode ekstraksi ciri yang ada pada proses pengolahan citra (image processing) seperti: Geometric Moment Invariant, paket Wavelet, GLCM dan lainnya. Pada studi kasus ini akan dipergunakan Geometric moment Invariant dalam proses ekstraksi ciri bentuk pada pola batiknya. Tahapan pada proses ekstraksi ciri bentuk yakni pertama dengan mensegmentasi background untuk memisahkan objek dengan background citra, kemudian menthreshod sehingga menjadi gambar biner dengan menggunakan tolak ukur pengubah nilai pixel apakah menjadi nilai 0 (hitam) atau 255 (putih).

Menghitung moment dan moment pusat menggunakan persamaan:

$$\mu_{pq} = = \iint_{\delta} (x - x_0)^p (y - y_0)^q f^i(x, y) dxdy \; ; p, q = 0,1,2 \ldots .n$$

Dan momen pusat yang ternormalisasi dinyatakan dengan persamaan berikut :

$$\eta_{ij} = \frac{\mu_{ij}}{(\mu_{00})^{\lambda}}$$

Dengan $\lambda = \frac{(i+j)}{2} + 1$, dengan $(i + j) \geq 2$ (momen tingkat ke-1 adalah selalu invariant. Dari momen ternormalisasi di atas, sekumpulan momen-momen invarian *(invariant moments)* dapat didefinisikan. Momen-momen ini sangat berguna dalam membuat vektor ciri untuk pengenalan objek.Berikut ini adalah persamaan dari momen-momen invariant.

$\phi_1 = \eta_{20} + \eta_{02}$

$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta^2_{11}$

$\phi_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$

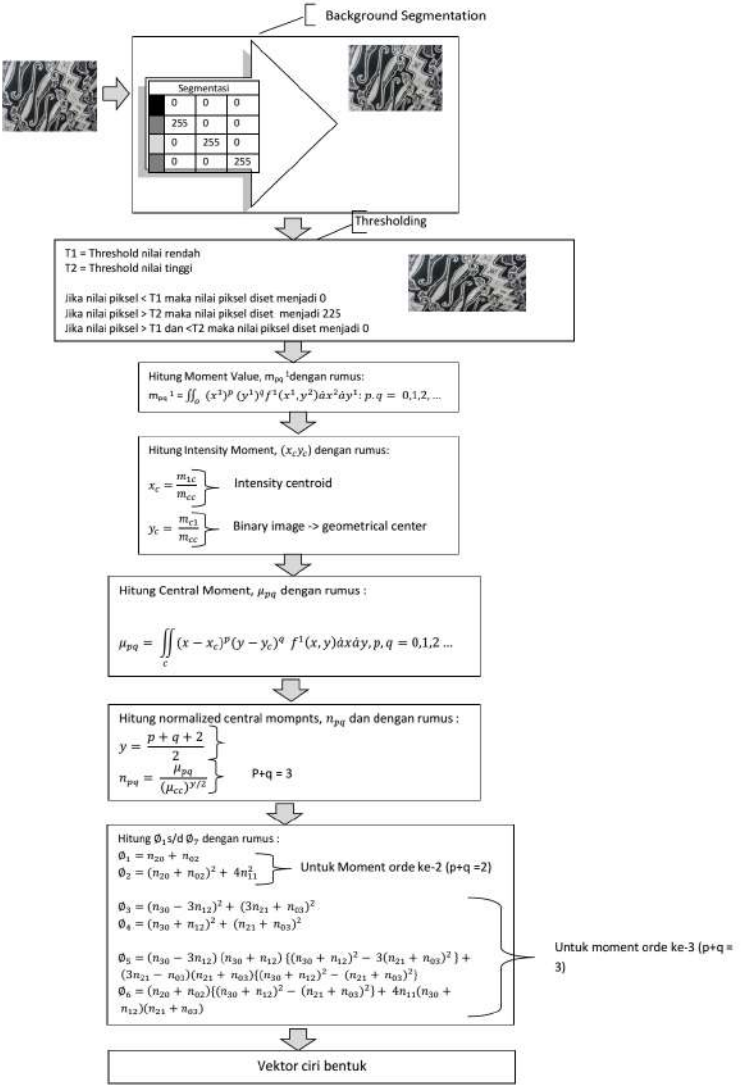
$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21}+\eta_{03})^2\} + (\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21}+\eta_{03})^2\}$

$\phi_6 = (\eta_{20} - \eta_{02})\{(\eta_{30}+\eta_{12})^2 - (\eta_{21}+\eta_{03})^2\} + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21}+\eta_{03})$

$\phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21}+\eta_{03})^2\} + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$

Secara diagram alir proses ekstraksi ciri geometric moment invariant dapat disajikan seperti pada gambar 5.8



Gambar 5.8: Diagram tahapan Ekstraksi ciri bentuk

Citra baru yang akan diklasifikasikan diperoleh dan fitur dibuat dari citra uji, kemudian dibandingkan dengan fitur yang berada pada basis data. Metode

klasifikasi K-NN digunakan untuk menentukan kelas dari citra batik yang baru. Klasifikasi K-NN dilakukan dengan mencari k buah tetangga terdekat dari data uji dan memilih kelas dengan anggota terbanyak.

Setelah proses klasifikasi maka dilakukan evaluasi terhadap kinerja klasifikasi. Penerapan algoritma K-NN pada suatu pengklasifikasian yang dilakukan melalui beberapa kali pengujian (testing) dan perlu dievaluasi atau diukur keakuratan klasifikasi dengan model tersebut. Keakuratan (accuracy) dari pengklasifikasian merupakan presentase dari banyaknya data uji yang diklasifikasikan dengan tepat oleh sebuah model klasifikasi. Confusion matrix atau yang dikenal dengan matrik klasifikasi sering digunakan untuk memprediksi keakuratan model klasifikasi tersebut. Confusion matrix memberikan informasi perbandingan klasifikasi actual dengan klasifikasi yang dihasilkan melalui sistem klasifikasi (Kohavi.Provost, 1998). Table confusion matrix menggunakan dua kategori yakni positif dan negatif.

		Prediksi	
		Positif	**Negatif**
Aktual	**Positif**	TP	FN
	Negatif	FP	TN

True Positif (TP) dan *True Negatif* (TN) adalah berisikan informasi jumlah klasifikasi kelas kategori positif dan negatif yang diprediksi dengan benar oleh sistem klasifikasi sesuai dengan kelas positif dan negatif pada keadaan actual. Sedangkan False Positif (FP) dan False negatif (FN) adalah jumlah klasifikasi kelas kategori yang salah diprediksi oleh sistem klasifikasi. Akurasi dapat dihitung dengan membandingkan hasil menjumlahkan banyaknya klasifikasi yang benar pada kelas positif dan negatif dengan keseluruhan data. Persamaan untuk menghitung akurasi yang menunjukan komposisi jumlah dari prediksi yang benar adalah (Kohavi.Provost, 1998):

$$\text{Akurasi} = \frac{TP + TN}{TP + FP + TN + FN}$$

Klasifikasi hasil perhitungan nilai keakuratan dapat didiagnosa dalam rentang sebagai berikut yang kemudian untuk presentasenya dikalikan 100%:

1. Akurasi bernilai 0.90-1.00 = excellent classification. Nilai 90%-100% dapat disimpulkan hasil klasifikasi mendekati sempurna (luar biasa).

2. Akurasi bernilai 0.80-0.90 = good classification. Nilai 80%-90% dapat disimpulkan hasil klasifikasi baik.
3. Akurasi bernilai 0.70-0.80 = fair classification. Nilai 70%-80% dapat disimpulkan hasil klasifikasi cukup baik (adil).
4. Akurasi bernilai 0.60-0.70 = poor classification. Nilai 60%-70% dapat disimpulkan hasil klasifikasi kurang baik.
5. Akurasi bernilai 0.50-0.60 = failure. Nilai 50%-60% dapat disimpulkan hasil klasifikasi gagal.

Dari hasil percobaan yang dilakukan pada studi kasus ini menggunakan 120 data citra dan metode klasifikasi K-NN, evaluasi klasifikasi menunjukkan rata-rata nilai presentase 83%. Dengan nilai tertinggi yaitu 88% pada nilai k=1. Sehingga dikategorikan klasifikasi citra dengan algoritma K-NN memiliki hasil nilai kategori klasifikasi yang baik (good clasification).

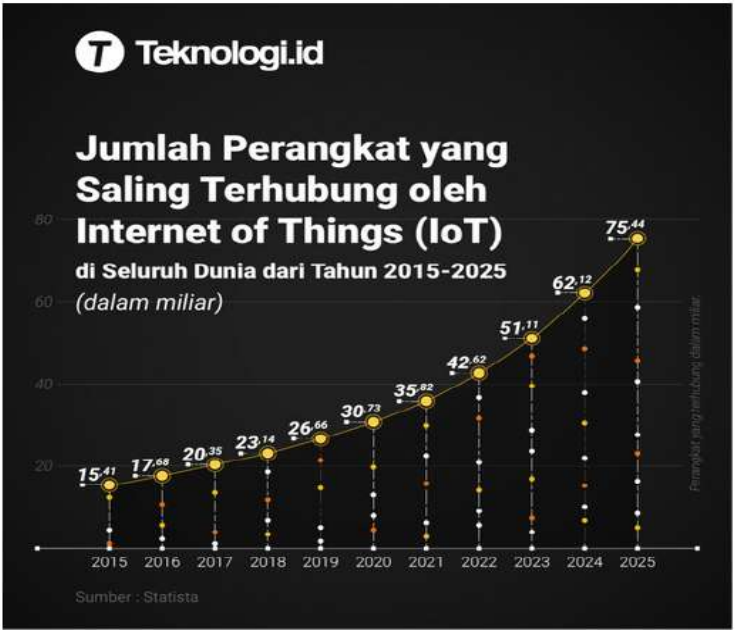
Bab 6

Penerapan Data Mining dengan Particle Swarm Optimization dan Decision Tree C4.5

6.1 Pendahuluan

Di era revolusi industri 4.0, data menjadi aset yang sangat penting bagi suatu organisasi. Data bisa diubah dan ditranformasikan menjadi informasi yang berguna dan bernilai tinggi sehingga dapat dimanfaatkan untuk kepentingan individu ataupun organisasi. Apalagi dengan perkembangan teknologi IOT (Internet of Things) yang merupakan teknologi pendorong utama pada transformasi digital di mana diperkirakan akan semakin banyak perangkat yang akan terhubung dengan Internet setiap tahunnya tentu akan menambah besarnya jumlah data yang saling dipertukarkan oleh berbagai perangkat tersebut. Besarnya data yang semakin signifikan jumlahnya yang karena pertumbuhannya bersifat eksponensial akan mengakibatkan traffic (lalu lintas) data juga akan semakin padat. Berdasarkan survei (Statistica, 2016) diprediksi jumlah perangkat yang terhubung ke Internet akan tumbuh mencapai angka 75 miliar pada tahun 2025, meningkat lima kali lipat dari 10 tahun sebelumnya seperti pada Gambar 6.1 di bawah. Untuk tahun 2020 ini diperkirakan ada sekitar 31 miliar perangkat di dunia yang terpasang atau terkoneksi ke Internet. Di Indonesia, pada akhir 2018 terdapat 10 juta perangkat yang telah terhubung ke Internet dan diprediksi jumlahnya akan terus naik. Pasar IOT juga akan terus meningkat di dunia dan diperkirakan nilainya bertumbuh melebihi angka

1 miliar dolar per tahunnya. Bahkan nilai investasi IOT bisa mencapai angka 15 triliun dolar hingga tahun 2025. Data ini diproyeksikan karena banyaknya perusahaan yang ingin berinvestasi dalam IOT untuk basis solusi bisnisnya pada tahun mendatang (Mastel, 2018).



Gambar 6.1: Proyeksi Pertumbuhan IOT 2015-2025 (Statistica, 2016)

Dengan pertumbuhan IOT yang semakin pesat maka tidak heran kebutuhan data juga akan meningkat pula. Pada tahun 2020, pemakaian data setiap harinya rata-rata mencapai 1,5 GB per orang. Jumlah ini meningkat hampir 1,5 kali lipat dari tahun 2016 di mana orang hanya menggunakan data sebesar 650 MB per hari. Sementara itu untuk industri mobil otonomous vehicle (self-driving car) diperkirakan mengkonsumsi data mencapai 4 kilo GB per hari. Industri pesawat terbang juga membutuhkan data lebih tinggi yakni hinga 40 Kilo GB per hari (Beritasatu, 2016). Dengan kata lain, industri manufaktur yang mengandalkan mesin akan lebih banyak mengkonsumsi data daripada manusia. Selain industri manufaktur, pemakaian data yang besar juga terjadi pada ritel dan s*mart city* karena didorong oleh kemajuan teknologi sensor dan IOT. Hal ini juga didukung oleh teknologi 5G sebagai kapasitas yang besar untuk menyalurkan data tersebut.

Secara tidak disadari banyak organisasi telah menghasilkan atau mengumpulkan data yang begitu besar dalam kurun waktu sekian tahun yang meliputi data transaksi, data nasabah, data penjualan, data pembelian, dll). Data yang dikumpulkan hampir semuanya digunakan untuk melakukan transaksi harian yang biasa disebut OLTP (Online Transaction Processing). Sebuah perusahaan atau organisasi biasanya memakai aplikasi untuk memasukkan data-data tersebut ke dalam tempat penyimpanan data (database) atau gudang data (data warehouse). Namun sayangnya, data dalam jumlah besar tersebut jarang sekali dipakai dan dimanfaaat kembali oleh penggunanya karena mungkin dianggap tidak penting lagi. Dapat dibayangkan berapa jumlah data penjualan yang dihasilkan dari transaksi sebuah retail seperti Giant atau Superindo yang telah memiliki basis pelanggan. Belum lagi banyaknya data pembelian (purchasing) dari para pemasoknya selama bertahun-tahun. Begitu juga dengan industri perbankan misalnya saja dari transaksi atau pemakaian kartu kredit oleh nasabahnya akan menyumbangkan data dalam ukuran yang sangat besar.

Pertanyaaan adalah apakah data dalam jumlah besar akan dibiarkan saja tidak berguna atau bahkan malah dibuang atau dihapus karena dianggap sebagai sampah yang memenuhi tempat penyimpanan data (database) saja? Atau mungkin kita dapat memanfaatkan data tersebut dengan meng"gali"nya untuk menemukan informasi yang berguna bagi organisasi? Dengan kemajuan teknologi informasi, database tersebut sebenarnya dapat dipakai untuk menghasikan informasi demi keuntungan organisasi dalam pengambilan keputusan atau untuk perkembangan ilmu pengetahuan dan riset.

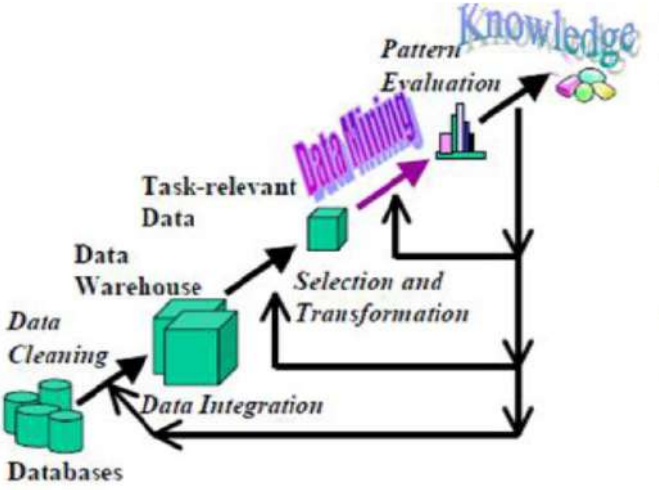
Sebagai contoh perusahaan yang menjual produk tertentu dapat menggali histori data pembelian untuk menemukan pembeli mana sajakah yang potensial dalam membeli suatu produk tertentu. Hal ini akan dapat membantu perusahaan untuk melakukan penghematan dalam melakukan pemasaran produknya kepada pelanggan (Hasibuan *et al*., 2020; Salmiah *et al*., 2020; Saputra *et al*., 2020) Selama ini mungkin perusahaan memperlakukan seluruh pelanggannya dengan cara yang sama. Semua pelanggan akan dihubungi melalui saluran telpon, menerima notifikasi SMS yang berisi penawaran produk tertentu atau bahkan dikirimi surat yang berisi brosur katalog produk. Walaupun teknik pemasaran tersebut cukup efektif tapi memakan biaya yang tidak sedikit. Misalkan saja perusahaan memiliki 1.000.000 pelanggan, untuk mengirimkan sebuah SMS penawaran produk dikenakan biaya sekitar Rp 500,- maka biaya yang yang dikeluarkan mencapai Rp 500.000.000,- untuk

satu kali SMS. Kalau hal ini dilakukan dua kali sebulan maka biayanya menjadi Rp 1 Milyar, dalam setahun maka total biaya pemasaran mencapai Rp 12 Milyar hanya dari SMS saja. Belum lagi pengeluaran dari biaya pulsa telpon, brosur, dll. Padahal dari total dana yang dihabiskan, mungkin hanya sekitar 20% konsumen yang melakukan pembelian dari produk yang ditawarkan sehingga perusahaan sebetulnya melakukan pemborosan dengan membuang dana hingga 80% atau sekitar hampir Rp 10 Milyar, yang seharusnya dapat dialokasikan untuk keperluan bisnis lainnya. Persoalan semacam ini dapat diatasi dengan memanfaatkan data mining di mana data mining dapat mengekstrak informasi yang penting dengan cara menggali data penjualan atau transaksi belanja konsumen untuk mencari pembeli yang potensial sehingga perusahaan hanya menyasar pada beberapa pembeli saja. Jika akurasi atau presisinya mencapai 20%, maka 80% dana yang tadi tidak akan terbuang dengan percuma.

Data Mining merupakan sebuah bidang ilmu yang relatif baru khususnya dalam ilmu komputer. Namun data mining masih sering diperdebatkan terkait posisi keilmuannya karena berkaitan erat atau menyangkut dengan database, kecerdasan buatan, machine learning, statistik regresi, dan lain-lain. Terlepas dari perdebatan tersebut, data mining telah berhasil menjadi pusat perhatian industri dalam beberapa tahun belakangan di mana semakin tingginya kebutuhan pengguna untuk mengubah data dalam ukuran besar menjadi informasi yang berguna (Fadli, 2003). Secara definisi, data mining adalah proses mengekstraksi informasi dan pengetahuan yang bersifat implisit yang sebelumnya tidak diketahui di mana ekstraksi dilakukan pada data dengan jumlah besar, tidak lengkap, acak dan berderau (Asriningtias *et al*., 2014).

Perbedaan antara data mining dengan data traditional adalah data traditional analysis fokus kepada proses query, laporan dan analisis aplikasi sedangkan data mining ditujukan untuk menggali informasi dan menemukan pengetahuan pada database (Sahu, Shrma dan Gondhalakar, 2008). Oleh karena itu data mining sering disebut dengan KDD (Knowledge Discovey in Database) yang berarti proses penemuan pengetahuan melalui pencarian pola yang teratur atau hubungan dalam set data berjumlah besar dengan menggunakan metode atau teknik tertentu (Mustafa, Ramadhan dan Thenata, 2018). Walaupun istilah data mining dan KDD sering tertukar namun sebenarnya data mining adalah bagian dari KDD (Sahu, Shrma dan Gondhalakar, 2008). Proses menggali informasi dan pengetahuan dapat dilakukan dari pengumpulan data mentah hingga

menghasilkan bentuk pengetahuan yang baru (Yuli, 2017) yang dapat disajikan pada Gambar 6.2 di bawah.



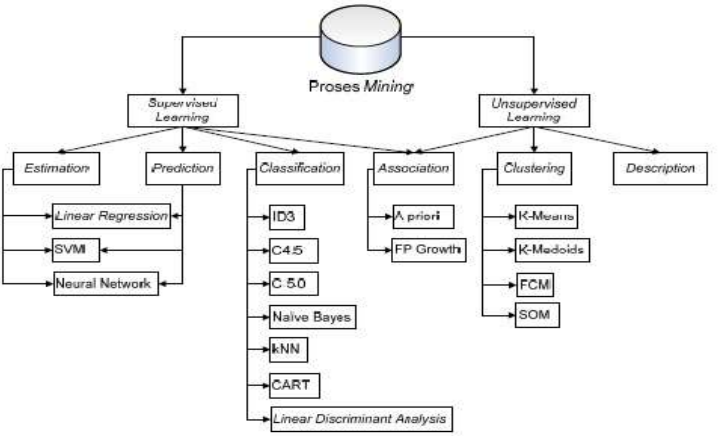
Gambar 6.2: Tahapan KDD (Sahu, Shrma dan Gondhalakar, 2008; Yuli, 2017)

Berdasarkan Gambar 6.2 di atas dapat dijelaskan tahapan-tahapan dalam KDD (Knowledge Discovery in Database) sebagai berikut:

1. Data Cleaning (Pembersihan Data)
   Pembersihan data adalah tahap awal dari tahapan KDD yang dilakukan dengan menghilangkan berbagai noise dan data yang tidak valid yang akan digunakan. Data mentah yang kita kumpulkan tidak terlepas dari berbagai kesalahan seperti kesalahan ketik, adanya duplikasi atau *data redundant*, data hilang, tidak konsisten atau data yang tidak relevan dengan kriteria. Data seperti ini harus dibersihkan terlebih dahulu dengan cara membuangnya agar tidak memengaruhi performa dari data mining. Selain itu data lain yang lebih relevan dapat ditambahkan ke dalam database.
2. Data Integration (Integrasi Data)
   Integrasi data adalah tahap untuk menggabungkan data dari berbagai sumber (database) yang bersifat heterogen ke dalam sebuah database (data set) termasuk atribut yang akan digunakan dalam proses ini. Data mining akan belajar dan menemukan pengetahuan dari berbagai data yang tersedia sehingga proses integrasi data menjadi sangat penting. Jika

beberapa atribut penting hilang maka keseluruhan studi akan gagal. Di lain pihak, mengumpulkan dan mengelola repositori daya yang kompleks akan sangat mahal. Oleh karena itu terjadi *trade-off* dengan peluang untuk memahami fenomena ini.

3. Data Selection (Seleksi Data)
   Seleksi data adalah tahap untuk menentukan data mana yang akan dipakai sehingga data dapat dipilih dengan tepat dan sesuai untuk dianalisa dari database (data set) atau tidak terjadi kesamaan dan perulangan dalam tahapan data mining. Seleksi data harus dilakukan sebelum tahap data mining dimulai. Data yang telah dipilih adalah hanya data yang relevan saja untuk disimpan dalam suatu berkas yang terpish dari basis data operasional. Misalnya, sebuah penelitian terkait faktor motivasi orang dalam membeli produk tertentu khususnya dalam kasus market basket analysis hanya membutuhkan atribut id pelanggan daripada nama pelanggan.
4. Data Transformation (Transformasi Data)
   Transformasi data merupakan tahap untuk mengubah dan menggabungkan data ke dalam format yang cocok untuk dapat diproses selanjutnya. Tahap ini dikenal dengan istilah Coding di mana prosesnya dalam KDD (Knowledge Discovery in Database) adalah proses kreatif yang tergantung pada pola informasi dan pengetahuan yang akan ditemukan pada database. Misalnya tahapan transformasi terjadi pada aplikasi clustering yang hanya dapat menerima data input bersifat kategori. Oleh karena itu data input yang bersifat angka numerik harus dibagi lagi ke dalam beberapa interval.
5. Data Mining (Penambangan Data)
   Penambangan data adalah tahap yang paling penting atau utama dalam KDD yang ditujukan untuk mengekstrak pola informasi dan menemukan pengetahuan berguna yang tersembunyi sebelumnya dari data. Terdapat berbagai metode atau teknik yang dapat digunakan dalam data mining seperti yang ditunjukkan pada Gambar 6.3 di bawah, di mana pemilihan metode atau teknik tersebut didasarkan pada tujuan dari KDD itu sendiri.
6. Patten Evaluation (Evaluasi Pola)
   Evaluasi pola merupakan tahap untuk mengidentifikasi atau mengenali pola-pola informasi menarik yang dapat merepresentasikan pengetahuan diidentifikasi berdasarkan ukuran yang ditentukan. Pola-pola informasi yang ditemukan tersebut pada tahap ini akan dievaluasi apakah bertentangan dengan hipotesis yang sudah ada sebelumnya.



Gambar 6.3: Metode atau Teknik Data Mining (Mustafa, Ramadhan dan Thenata, 2018)

7. Knowledge Presentation (Presentasi Pengetahuan)
   Presentasi Pengetahuan adalah tahap di mana pengetahuan baru yang telah ditemukan sebelumnya dapat direpresentasikan secara visual dan lengkap kepada pengguna. Oleh karena itu penyajian pengetahuan dengan baik kepada pengguna merupakan proses yang sangat diperlukan. Dalam tahap ini, teknik visualisasi biasanya dapat membantu untuk mengkomunikasikan hasil pengolahan data mining (Han, 2006). Tahap ini juga menjadi penting karena akan membantu pengguna untuk bisa memahami dan menginterpretasi hasil dari pengolahan data mining yang telah dilakukan. Selanjutnya pengguna akan menggunakan hasil tersebut sebagai pendukung dalam mengambil keputusan.

6.2 Penerapan Data Mining

Seperti disebutkan sebelumnya bahwa data mining bertujuan untuk mengekstraksi informasi dari data dengan ukuran besar sehingga menghasilkan pola informasi yang menarik sebagai pengetahuan baru yang tersembunyi pada data tersebut. Pola informasi yang menarik tersebut maksudnya tidak bersifat sepele, implisit dan belum diketahui sebelumnya (Kurniawan dan Rosadi, 2017). Pada bagian selanjutnya akan diberikan contoh pemanfaatan data mining yaitu berupa teknik klasifikasi yang

diimplementasikan untuk menganalisa siswa yang putus sekolah. Siswa putus sekolah menjadi problem klasik dari domain pendidikan yang tidak kunjung dapat diselesaikan. Ada banyak faktor yang menyebabkan siswa putus sekolah seperti faktor ekonomi keluarga yang tidak mampu membiayai pendidikan anaknya, faktor internal kemampuan siswanya, faktor lokasi, dll. Intinya siswa tidak dapat melanjutkan atau menamatkan pendidikannya. Angka putus sekolah di Indonesia lebih banyak dialami oleh siswa SMA (Sekolah Menengah Atas) dibandingkan siswa usia lainnya. Hal ini menjadi permasalahan bagi pemerintah terkait terhambatnya program yakni penggalakkan wajib belajar 12 tahun (Kurniawan dan Rosadi, 2017). Untuk mencegah siswa putus sekolah, diperlukan tindakan preventif khususnya terhadap siswa yang diprediksi putus sekolah. Dengan kata lain, pada studi kasus ini dibutuhkan suatu cara untuk mengetahui atau mengklasifikasikan siswa mana saja yang diprediksi akan putus sekolah.

Dalam data mining terdapat banyak teknik di mana klasifikasi menjadi salah satunya. Pada teknik klasifikasi akan didefinisikan karakteristik dari sebuah kelompok atau mengkategorikannya ke dalam beberapa jenis yang sudah diketahui. Yang membedakan klasifikasi dari teknik lainnya adalah sudah ada target kategorinya (Santoso *et al*., 2018). Misalnya dalam kasus ini dapat dikategorikan atau digolongkan menjadi lulus dan tidak lulus. Contoh yang lain misalnya untuk mendeteksi adanya penggunaan kartu kredit secara ilegal berdasarkan data transaksi sebelumnya. Lalu data mining akan mencari model klasifikasi aman atau tidak aman. Dengan demikian kita dapat menerapkan model tersebut terhadap transaksi yang baru untuk melakukan tindakan preventif. Salah satu algoritma yang telah digunakan secara luas dalam teknik klasifikasi adalah algoritma C4.5 atau Decision Tree. Pohon keputusan (Decision Tree) merupakan metode klasifikasi dan prediksi yang sangat popular dalam data mining karena fleksibilitasnya dan kemudahannya untuk diinterpretasi atau diamati (Hadiri, 2016). Pada dasarnya pohon keputusan mengubah data yang sangat besar menjadi decision tree yang merepresentasikan aturan. Proses yang terjadi pada decision tree yaitu mengubah bentuk tabel menjadi model pohon kemudian mengubah model pohon ke dalam rule (aturan) serta menyederhanakannya (Sulistiyanto, 2018).

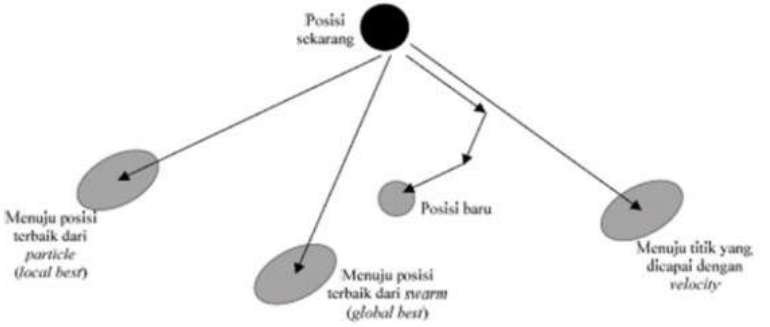
Terdapat beberapa tahapan untuk membuat pohon keputusan dengan algoritma C4.5 (Ramdhani, 2016) yaitu:

1. Menyiapkan data training yang diambil dari data histori atau data data masa lampau yang sudah ada sebelumnya serta telah dikelompokkan menjadi beberapa kelas.
2. Menghitung simpul akar dari pohon yang diambil atau dipilih dari atribut dan menentukan atribut yang akan menjadi simpul berikutnya. Pemilihan atribut ditentukan dari nilai gainnya di mana akar pertama adalah yang memiliki nilai gain tertinggi. Untuk mendapatkan nilai gain, terlebih dahulu harus dihitung besarnya entropi.
3. Mengulangi langkah 2 sampai semua record terpartisi di mana proses partisi dari pohon keputusan akan dihentikan ketika:
    - Semua record mempunyai kelas yang sama dalam simpul N
    - Tidak ada atribut dari record yang dapat dipartisi lagi
    - Tidak ada record dalam cabang yang kosong

Algoritma C4.5 adalah hasil pengembangan dari algoritma sebelumnya yaitu ID3 (Iterative Dichotometer 3) yang diciptakan oleh J. Rose Quinlan dan termasuk dalam top 10 besar algoritma yang direkomendasikan dalam data mining (Yuli, 2017). Namun demikian algoritma ini bukan tanpa kekurangan terutama pada saat membaca data dalam jumlah besar sehingga akurasinya menurun. Oleh karena itu untuk meningkatkan akurasinya perlu dikombinasikan dengan algoritma lain untuk memperbaiki kelemahan tersebut (Santoso *et al*., 2018). Algoritma PSO (Particle Swarm Optimization) dapat digunakan untuk mengoptimasi decision tree dengan cara menseleksi atribut dengan efektif sedemikian sehingga akurasi dari algoritma C4.5 dapat meningkat. PSO juga cukup populer dipakai pada teknik klasifikasi data mining karena hanya membutuhkan waktu komputasi yang sedikit.

Algoritma PSO adalah teknik optimasi atau algoritma yang berbasis pada populasi di mana secara teori setiap individu akan berpindah dari tempat satu ke tempat lainnya dengan kecepatan yang disesuaikan dan bergerak kepada posisi yang terbaik berdasarkan pengalaman pribadinya. Dalam algoritma PSO, istilah warm diartikan sebagai populasi dan particle merupakan individu. PSO awalnya dikembangkan atau didasarkan pada kecerdasan berkelompok dari kawanan serangga seperti burung, ikan, rayap, semut, dan lain-lain. Kawanan ini memiliki perilaku unik dalam mencari makanan (Ramdhani, 2016). Sebuah partikel diibaratkan sebagai seekor burung, populasinya adalah

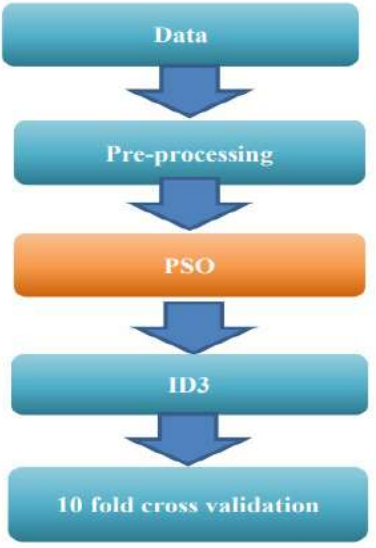
kawanan burung. Ketika seekor burung menemukan makanan di suatu daerah, maka apa yang dilakukan kawanan burung lainnya untuk mengetahui lokasi makanan tersebut adalah mencari posisi terbaik sebelumnya yakni posisi terbaik secara global. Dengan demikian kawanan tersebut dapat mengikuti arah tersebut walaupun lokasi atau posisi mereka cukup jauh. Model perpindahan partikel dalam PSO dapat ditunjukkan pada Gambar 6.4 sebagai berikut:



Gambar 6.4: Model Perpindahan Partikel (Kurniawan dan Rosadi, 2017)

Berdasarkan Gambar 6.4 dapat dilihat bahwa algoritma PSO sebagai metode optimasi didasarkan pada intelegensi atau kecerdasan populasi (swarm intelligence) untuk mencari solusi dari permasalahan. Populasi berupa kawanan diasumsikan memiliki ukuran tertentu dengan tiap artikel di mana posisi awal terletak pada suatu lokasi yang random di dalam ruang multidimensi. Dua karakteristik yang dimiliki oleh setiap partikel adalah kecepatan dan posisi.

Selanjutnya pada kasus klasifikasi siswa putus sekolah, dilakukan eksperimen dengan mengaplikasikan decision tree pada data yang telah dilakukan pre-processing, lalu diuji akurasinya dengan metode confusion matrix. Sementara itu algoritma PSO diterapkan pada decision tree untuk kemudian dibandingkan tingkat akurasinya dengan algoritma decision tree tanpa PSO (Kurniawan dan Rosadi, 2017). Desain eksperimen dari algoritma PSO dan decision tree dapat disajikan pada Gambar 6.5 di bawah



Gambar 6.5: Desain Eksperimen (Kurniawan dan Rosadi, 2017)

Analisa dari hasil eksperimen nantinya dilakukan dengan pengujian tingkat akurasinya dengan menghitung atau membandingkan Recall dan Precision dari tree yang diperoleh untuk algoritma C4.5 tanpa PSO maupun algoritma C4.5 dengan PSO. Klasifikasi yang lebih baik dari kedua model ini ditunjukkan dari seberapa tinggi tingkat akurasinya yang dapat dikatakan mendekati solusi ideal. Untuk mengevaluasi kinerja klasifikasinya, tingkat akurasi dihitung dengan confusion matrix yang akan menghasilkan nilai *precision, recall* dan *accuracy model* yang dapat dirumuskan yakni (Santoso *et al*., 2018):

$$\text{Precision} = \text{TP/TP+FP} \quad (1)$$

$$\text{Recall} = \text{TP/TP+FN} \quad (2)$$

$$\text{Accuracy} = \text{TP+TN/TP+TN+FN} \quad (3)$$

Di mana:

TP: Jumlah kasus positif yang diklasifikasikan sebagai positif

FP: Jumlah kasus positif yang diklasifikasikan sebagai positif

TN: Jumlah kasus positif yang diklasifikasikan sebagai negatif

FN: Jumlah kasus positif yang diklasifikasikan sebagai negatif

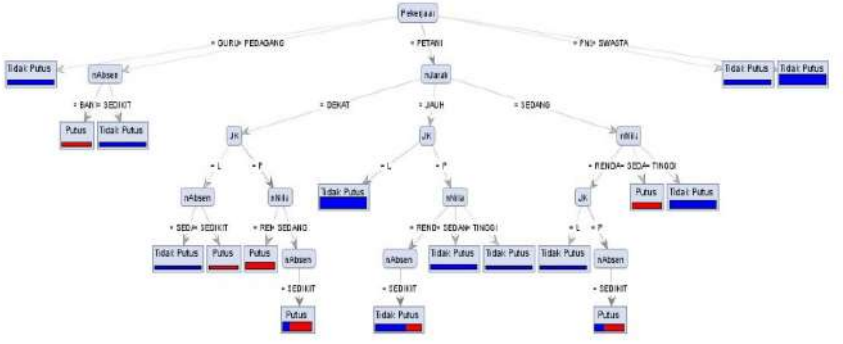
Data yang telah di *pre-processing* dapat disajikan pada Tabel 6.1 sebagai berikut:

Tabel 6.1: Data Siswa Pre-Processing (Kurniawan dan Rosadi, 2017)

ID	JK	Jarak Rumah-Sekolah	Pekerjaan Ortu	Absensi	Nilai	Keterangan
0001	L	Jauh	Petani	Sedikit	Rendah	Tidak Putus
0002	P	Dekat	Petani	Sedikit	Sedang	Tidak Putus
0003	L	Dekat	Petani	Sedikit	Sedang	Putus
0004	L	Sedang	Guru	Sedikit	Rendah	Tidak Putus
0005	P	Sedang	Pedagang	Banyak	Sedang	Putus
0006	P	Dekat	Guru	Sedikit	Tinggi	Tidak Putus
0007	P	Dekat	Petani	Sedikit	Rendah	Putus
0008	L	Jauh	Petani	Sedikit	Sedang	Tidak Putus
0009	L	Sedang	Swasta	Sedikit	Sedang	Tidak Putus
0010	P	Sedang	Petani	Sedikit	Rendah	Putus
0011	P	Jauh	Petani	Sedikit	Rendah	Tidak Putus
0012	P	Dekat	Petani	Sedikit	Sedang	Putus
0013	P	Dekat	Petani	Sedang	Rendah	Putus
0014	L	Dekat	Petani	Sedang	Rendah	Tidak Putus
0015	P	Sedang	Petani	Sedang	Sedang	Putus
0016	P	Jauh	Petani	Sedikit	Sedang	Tidak Putus
0017	P	Dekat	Pns	Sedikit	Tinggi	Tidak Putus
0018	P	Sedang	Petani	Sedikit	Rendah	Putus
0019	P	Dekat	Petani	Sedikit	Sedang	Putus
0020	L	Jauh	Petani	Sedikit	Sedang	Tidak Putus
0021	P	Jauh	Petani	Sedikit	Rendah	Putus
0022	P	Sedang	Petani	Sedikit	Rendah	Tidak Putus
0023	P	Jauh	Petani	Banyak	Sedang	Tidak Putus
0024	L	Sedang	Petani	Sedikit	Rendah	Tidak Putus
0025	P	Dekat	Swasta	Sedikit	Rendah	Tidak Putus
0026	P	Sedang	Petani	Sedikit	Tinggi	Tidak Putus
0027	P	Jauh	Petani	Sedikit	Tinggi	Tidak Putus
0028	P	Jauh	Swasta	Sedikit	Rendah	Tidak Putus
0029	P	Jauh	Swasta	Sedang	Sedang	Tidak Putus
0030	L	Jauh	Petani	Sedikit	Sedang	Tidak Putus

ID	JK	Jarak Rumah-Sekolah	Pekerjaan Ortu	Absensi	Nilai	Keterangan
0031	P	Sedang	Petani	Sedikit	Tinggi	Tidak Putus
0032	P	Sedang	Pedagang	Sedikit	Sedang	Tidak Putus
0033	P	Dekat	Petani	Sedikit	Sedang	Putus
0034	P	Dekat	PNS	Sedikit	Rendah	Tidak Putus
0035	P	Dekat	Petani	Sedang	Rendah	Putus
0036	P	Sedang	Swasta	Sedikit	Rendah	Tidak Putus
0037	P	Sedang	Petani	Sedikit	Sedang	Putus
0038	P	Sedang	Petani	Sedikit	Tinggi	Tidak Putus
0039	P	Jauh	Petani	Sedikit	Rendah	Tidak Putus
0040	L	Jauh	Petani	Sedikit	Rendah	Tidak Putus

Selanjutnya model *decision tree* yang dihasilkan akan diukur kinerjanya berdasarkan data berikut yang ditunjukkan pada Gambar 6.6 pemodelan *decision tree* untuk analisa data siswa putus sekolah sebagai berikut:



Gambar 6.6: Decision Tree (Kurniawan dan Rosadi, 2017)

Metode pengujian validasi (X-Validation) baik untuk algoritma decision tree tanpa PSO (Tabel 6.2) dan dengan PSO (Tabel 6.3) akan menghasikan confusion matrix yang menyajikan nilai precision, recall dan hasil true & false sebagai berikut:

Tabel 6.2: Confusion Matrix Decision Tree tanpa PSO (Kurniawan dan Rosadi, 2017)

Accuracy: 72.50% ± 13.46% (mikro: 72.50%)			
	True Tidak Putus	True Putus	Class Precision
Pred. Tidak Putus	21	5	80.77%
Pred. Putus	6	8	57.14%
Class Recall	77.78%	61.54%	

Tabel 6.3: Confusion Matrix Decision Tree yang dioptimasi PSO (Kurniawan, 2017)

Accuracy: 85.00% ± 16.58% (mikro: 85.00%)			
	True Tidak Putus	True Putus	Class Precision
Pred. Tidak Putus	24	3	88.89%
Pred. Putus	3	10	76.92%
Class Recall	88.89%	76.92%	

Hasil pemodelan Tree yang diperoleh memperlihatkan bahwa pekerjaan ortu (orang tua) mempunyai pengaruh utama dalam pemodelan ini di mana jika pekerjaan ortu (orang tua) adalah sebagai guru, PNS dan swasta maka dapat dikatakan siswa tidak putus sekolah. Sedangkan jika pekerjaan ortu (orang tua) adalah pedagang maka dapat dilihat jumlah absensinya. Kalau jumlah absensinya banyak maka siswa putus sekolah. Sementara itu jika pekerjaan ortu (orang tua) sebagai petani maka dapat dilihat parameter lainnya seperti jarak, absensi dan nilai. Secara keseluruhan perbandingan antara hasil decision tree tanpa algoritma PSO dengan *decision tree* yang telah dioptimasi dengan algoritma PSO dapat disajikan pada Tabel 6.4 sebagai berikut:

Tabel 6.4: Perbandingan Hasil (Kinerja) Accuracy Model Klasifikasi (Kurniawan dan Rosadi, 2017)

	Decision Tree	Decision Tree dengan PSO
Accuracy	72.5	85
Precision	57.14	76.92
Recall	65	75

Berdasarkan Tabel 6.4 di atas dapat ditunjukkan bahwa model klasifikasi decision tree menunjukkan akurasi lebih rendah 72.5% daripada model klasifikasi decision tree yang telah dioptimasi oleh PSO. Hal ini dapat

disimpulkan bahwa PSO cukup efektif untuk meningkatkan kinerja akurasi dari *decision tree* khususnya terhadap data siswa putus sekolah.

Bab 7

Klasifikasi Data Menggunakan Algoritma Naive Bayes

7.1 Pendahuluan

Salah satu tugas dari data mining yang sering digunakan adalah klasifikasi. Klasifikasi merupakan pengelompokan objek berdasarkan model objek atau kelompok yang telah ditentukan. Klasifikasi memerlukan data latih yang telah diberikan label atau kelas. Klasifikasi menurut (He & Tan, 2012) merupakan suatu model pada data mining yang *classifier* dibangun untuk memprediksi kategorikal label, seperti "aman" maupun "beresiko" untuk data penerapan peminjaman uang; "ya" maupun "tidak" untuk data sales; maupun "treatment A", "treatment B" maupun "treatment C" untuk data medis. Proses pengelompokan dilakukan dengan membangun terlebih dahulu model data atau disebut dengan data latih kemudian setelah model terbentuk dari proses pelatihan tersebut maka data baru dapat diujikan dalam proses pengelompokan yang disebut dengan proses uji.

Proses pada klasifikasi data mempunyai dua tahapan ialah :

1. *Learning* ialah proses latih data kemudian dianalisis dengan memakai algoritma klasifikasi.
2. *Classification* ialah proses pengujian data yang dipakai untuk mengetahui ketepatan dari *classification rules*. Akurasi yang keberadannya dapat diterima, *rule* dapat diimplementasikan pada suatu

klsifikasi dari *tuple* data baru. (Nikulin, 2008) lebih detail mengatakan bahwa, klasifikasi hanya bisa digunakan pada suatu data latih yang kuat di mana diperkirakan bahwa kelas "positif" telah mewakili minoritas tanpa harus kehilangan atribut pada umumnya.

7.2 Algoritma Naïve Bayes

Algoritma Naïve Bayes merupakan suatu algoritma klasifikasi berdasarkan teorema Bayesian pada statistika (Suntoro, et al., 2018). Algoritma Naïve Bayes berguna untuk memprediksi probabilitas keanggotaan suatu kelas. Naïve Bayes merupakan statistik yang fundamental pada data mining. Pendekatan ini didasari oleh kuantitatif *trade-off* pada berbagai keputusan klasifikasi dengan memakai probabilitas. Adapun ciri utama dari Naïve Bayes Classifier ini adalah asumsi yg sangat kuat (naif) akan independensi dari masing-masing kondisi / kejadian.

Menurut Olson Delen (2008) Naïve Bayes merupakan algoritma yang bekerja dengan menghitung untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informasi obyek. Algoritma Naïve Bayes berkerja dengan mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari "master" tabel keputusan.

Naive Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini telah dibuktikan oleh Xhemali, Hinde Stone dalam pada artikel yang berjudul „Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages" mengatakan bahwa Naive Bayes Classifier memiliki tingkat akurasi yg lebih baik dibandingmodel classifier lainnya (Xhemali, et al., 2009). Penelitian lain juga dilakukan oleh (Nurhachita & Negara, 2020) dengan judul "A Comparison Between Naïve Bayes and The K-Means Clustering Algorithm for The Application of Data Mining on The Admission of New Students" yang menunjukkan bahwa algoritma Naïve Bayes memiliki tinggkat arusasi yang lebih baik yaitu 9.08%.

Keuntungan penggunannya algoritma Naïve Bayes adalah metode ini hanya membutuhkan jumlah data latih yang kecil untuk menentukan estimasi parameter yg diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai *variable independent*, maka hanya varians dari suatu

variable dalam sebuah kelas yang dibutuhkan untuk menentukan suatu klasifikasi, bukan keseluruhan dari matriks kovarians.

Kelebihan Naïve Bayes:

1. Naïve Bayes dapat digunakan untuk data yang bersifat kuantitatif maupun kualitatif.
2. Pada proses analisis dan klasifikasi Naïve Bayes tidak memerlukan jumlah data yang banyak.
3. Naïve Bayes tidak perlu melakukan pelatihan data dalam jumlam yang banyak.
4. Jika ada nilai yang hilang, maka Naïve Bayes bisa mengabaikan dalam perhitungan.
5. Naïve Bayes memiliki mekanisme dan proses perhitungan cepat dan efisien.
6. Naïve Bayes mudah untuk dipahami bagi pemula.
7. Naïve Bayes mudah untuk digunakan ditambah dengan aplikasi RapidMiner yang telah menyediakan algoritma tersebut.
8. Pengklasifikasian dokumen dengan Naïve Bayes dapat di personalisasi dan juga dapat disesuaikan dengan kebutuhan setiap orang.
9. Jika digunakan dalaam bahasa pemrograman, kode Naïve Bayes cukup sederhana.
10. Naïve Bayes dapat digunakan dalam klasifikasi permasalahan biner ataupun multiclass.

Kekurangan Naïve Bayes:

1. Pada Naïve Bayes apabila probabilitas kondisionalnya bernilai nol, maka probabilitas prediksi juga akan bernilai nol.
2. Pada Naïve Bayes diasusikan bahwa masing-masing variabel independen membuat berkurangnya akurasi, karena biasanya ada korelasi antara variabel yang satu dengan variabel yang lain.
3. Keakuratan Naïve Bayes tidak dapat diukur hanya menggunakan satu probabilitas saja. Diperlukan pengujian dan bukti-bukti lain untuk membuktikannya.

4. Untuk membuat keputusan, pada Naïve Bayes diperlukan pengetahuan awal atau pengetahuan mengenai masa sebelumnya. Keberhasilan Naïve Bayes sangat bergantung pada pengetahuan awal tersebut. Terdapat beberap celah yang dapat mengurangi efektivitasnya. Naïve Bayes dirancang untuk menganalisis dan mendeteksi data yang bersifat *text* saja, tidak bisa berupa gambar.

Teorema Bayein menghitung nilai posterior probability P (H|X) menggunakan probabilitas P(H), P(X), dan P (Kantardzic, 2011), di mana nilai X adalah data uji yang kelasnya tidak diketahui. Nilai H adalah hipotesis data X adalah suatu kelas yang spesifik. Nilai P(X|H) atau disebut juga dengan likelihood, adalah probabilitas hipotesis X berdasarkan kondisi H. nilai P(H) atau disebut juga dengan *prior probability* adalah hipotesisH. Sedangkan nilai P(X) yang disebut juga dengan *predictor prior probability* adalah probabilitas X.(X|H).

$$P(H|X) = \frac{P(X|H).P(H)}{P(H)}$$

Algoritma Naïve Bayes ini sangat cocok untuk klarifikasi untuk dataset bertipe nominal. Untuk dataset bertipe nominal, perhitungan algoritma Naïve Bayes menggunakan persamaan. Apabila dataset bertipe numeric maka digunakan penghitungan distribusi Gaussian (Ryu & Baik, 2016). Perhitungan distribusi dapat dilihat dari persamaan, di mana dihitung terlebih dahulu rata-rata *u* sesuai persamaan, dan standard deviasi o sesuai persamaan.

$$g(x,\mu,\acute{o}) = \frac{1}{\sqrt{2\pi.\acute{o}}} \exp \frac{-(x-\mu)^2}{2\acute{o}^2}$$

$$\mu = \frac{\sum_i^n Xi}{n}$$

$$\acute{o} = \sqrt{\frac{\sum_i^n (Xi-\mu)^2}{n-1}}$$

Langkah-langkah algoritma Naïve Bayes adalah :

1. Menyiapkan dataset
2. Lalu hitunglah jumlah kelas yang terpadat di data latih
3. Lalu hitunglah jumlah kasus dengan kelas yang sama.

4. Kemudian dikalikan hasil sesuai pada data uji yang akan dicari kelasnya.
5. Lalu bandingkan hasil per kelas, nilai yang paling tinggi dapat ditetentukan sebagai kelas baru.

7.3 Penghitungan Manual Tipe Data Nominal Algoritma Naïve Bayes

Dataset yang digunakan pada perhitungan manual ini adalah data pembelian komputer. Dataset pembelian computer dibagi menjadi dua, yaitu data latih (dapat dilihat pada tabel 1) dan data uji (dapat dilihat pada tabel 2). dataset pembelian komputer bertipe data nominal terdiri dari 4 atribut dan 21 kelas. Berikut adalah langkah langkah perhitungan manual algoritma Naïve Bayes:

1. Siapkan dataset.
   Seperti yang telah dijelaskan di atas, dataset yang digunakan pada perhitungan manual ini menggunakan dataset pembelian computer. Dan dapat di lihat pada tabel 1 dan 2.
2. Hitung jumlah kelas pada data latih
   Kelas pada data latih terdiri dari dua kategori, yaitu beli komputer dan tidak beli komputer sehingga probabilitas untuk beli computer dan tidak beli komputer adalah sebagai berikut:
   Jumlah kelas beli komputer = 9
   Jumlah kelas tidak beli komputer = 5
   Maka,
   P(C = “beli”) = $\frac{9}{14}$ = 0,64
   P(C = “tidak beli”) = $\frac{5}{14}$ = 0,36

Tabel 7.1: Data Latih pada Pembelian Komputer

Usia	**Pendapatan**	**Pelajar**	**Kredit**	**Kelas**
Muda	Tinggi	Tidak	Macet	Tidak Beli
Muda	Tinggi	Tidak	Lancar	Tidak Beli
Tengah baya	Tinggi	Tidak	Macet	Beli
Tua	Sedang	Tidak	Macet	Beli
Tua	Rendah	Ya	Macet	Beli
Tua	Rendah	Ya	Lancar	Tidak Beli
Tengah baya	Rendah	Ya	Lancar	Beli
Muda	Sedang	Tidak	Macet	Tidak Beli
Muda	Rendah	Tidak	Macet	Beli
Tua	Sedang	Ya	Macet	Beli
Muda	Sedang	Ya	Lancar	Beli
Tengah baya	Sedang	Tidak	Lancar	Beli
Tengah baya	Tinggi	Ya	Macet	Beli
Tua	Sedang	Tidak	Lancar	Tidak Beli

Tabel 7.2: Data Uji pada Pembelian Komputer

Usia	**Pendapatan**	**Pelajar**	**Kredit**	**Kelas**
Tua	Tinggi	Tidak	Macet	?

3. Kemudian hitunglah jumlah kasus dan dengan kelas yang sama

P (usia = “tua” | C = “beli”) = $\frac{3}{9}$ = 0,33

P (usia = “tua” | C = “tidak beli”) = $\frac{2}{5}$ = 0,40

P (pendapatan = “tinggi” | C = “beli”) = $\frac{2}{9}$ = 0,22

P (pendapatan = “tinggi” | C = “tidak beli”) = $\frac{2}{5}$ = 0,40

P (pelajar = “tidak” | C = “beli”) = $\frac{3}{9}$ = 0,33

P (pelajar = “tidak” | C = “tidak beli”) = $\frac{4}{5}$ = 0,80

P (kredit = “macet” | C = “beli”) = $\frac{6}{9}$ = 0,67

P (kredit = “macet” | C = “tidak beli”) = $\frac{2}{5}$ = 0,40

4. Lalu dikalikan semua hasil yang telah didapatkan sesuai dengan data testing yang akan dicari kelasnya

P (X | C = “beli”) = 0,33 * 0,22 * 0,33 * 0,67 = 0,02

P (X | C = “tidak beli”) = 0,40 * 0,40 * 0,80 * 0,40 = 0,05

P (X | C = “beli” \ X) = 0,02 * 0,64 = 0,01

P (X | C = “tidak beli” \ X) = 0,05 * 0,36 = 0,02

5. Bandingkan hasil per kelas
   Dari perhitungan probabilitas beli komputer dan probabilitas tidak beli komputer pada langkah sebelumnya, dapat disimpulkan bahwa data usia = tua, pendapatan = tinggi, pelajar = tidak, dan krdit = macet masuk ke dalam kelas tidak beli komputer karena probabilitas beli komputer (0,01).

7.4 Analisis Data menggunakan Algoritma Naïve Bayes dan Aplikasi RapidMinner

RapidMiner adalah suatu perangkat lunak untuk sains data yang dibentuk oleh RapidMiner, Inc yang menyajikan terintegrasi data, *text mining*, *machine learning*, dan analisis prediktif. Serta membantu langkah dalam suatu proses pembelajaran mesin seperti hasil visualisasi, optimasi, persiapan data dan validasi model. RapidMiner membawa kecerdasan buatan ke perusahaan melalui platform sains data yang terbuka dan dapat diperluas. Dibangun untuk tim analitik, RapidMiner menyatukan seluruh siklus ilmu sains data dari persiapan data hingga *machine learning* hingga penyebaran model prediktif (Hofmann & Klinkenberg, 2016). Secara sederhana RapidMiner adalah sebuah aplikasi yang digunakan untuk mengolah sebuah data dengan berbagai teknik dan metode dalam *data mining*, sehingga data dapat menjadi informasi yang berguna. RapidMiner merupakan salah satu perangkat lunak yang bersifat *opensource* yang berguna untuk pengolahan suatu data mining.

Pada awal dikembangkan RapidMiner sebelumnya bernama YALE (Yet Another Learning Environment), dimana versi awalnya ini mulai dikembangkan pada tahun 2001 oleh RalfKlinkenberg, Ingo Mierswa, dan Simon Fischer di Artificial Intelligence Unit dari University of Dortmund. RapidMiner didistribusikan di bawah lisensi AGPL (GNU Affero General Public License) versi 3. Saat ini telah ribuan aplikasi yang dikembangkan mengunakan RapidMiner di lebih dari 40 negara. RapidMiner sebagai software open source untuk data mining tidak perlu diragukan lagi karena software ini sudah terkemuka di dunia. RapidMiner menempati peringkat pertama sebagai Software data mining pada polling oleh KDnuggets, sebuah portal data-mining pada 2010-2011.

RapiMiner juga menyediakan prosedur *data mining* dan *machine learning* yang didalamnya termasuk, *ETL (extraction, transformation, loading)*, data preprocessing, visualisasi, modelling dan evaluasi. Proses data mining tersusun atas operator-operator yang nestable, dideskripsikan dengan XML, dan dibuat dengan GUI.

RapidMiner sebagai perangkat lunak untuk sains data dan *data mining* memiliki beberapa sifat diantara nya sebagai berikut :

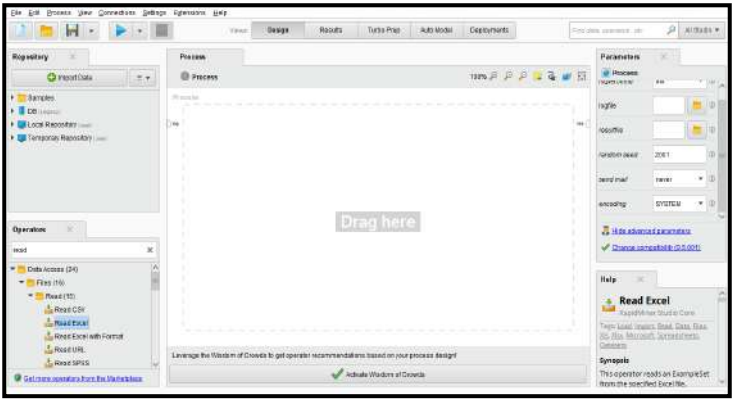
1. RapidMiner ditulis dengan bahasa pemrograman Java yang memungkinkan untuk dapat dijalankan diberbagai sistem operasi.
2. Proses analisis dengan berbagai tugas *data mining* untuk menemukan pengetahuan *(knowledge)* dimodelkan sebagai operator *trees.*
3. RapidMiner menggunakan representasi XML internal untuk memastikan format standar pertukaran data.
4. Bahasa dan kode program memungkinkan untuk melakukan eksperimen dengan skala besar dan melakukan otomatisasi eksperimen.
5. RapidMiner menggunakan konsep multi-layer untuk menjamin tampilan data yang efisien dan menjamin penanganan data selama melakukan proses analisis.
6. RapidMiner memiliki GUI, command line mode, dan Java API yang dapat dipanggil dari program lain.

Berikut ini adalah fitur-fitur yang ada dalam RapidMiner yaitu:

1. Terdapat berberapa algoritma data mining yang dapat digunakan secara langsung, seperti: *decision tree* dan *self organization map*, dan lain-lain.
2. RapidMiner memiliki bentuk grafis yang canggih, seperti diagram *histogram, tree chart* dan *3D scatter plots*.
3. RapidMiner memiliki banyaknya variasi *plugin*, seperti *text plugin* untuk melakukan analisis teks.
4. RapidMiner memiliki prosedur *data mining* dan *machine learning* termasuk: *ETL (extraction, transformation, loading)* data preprocessing, visualisasi, modeling dan evalualisasi.
5. Proses data mining tersusun atas operator-operator yang *nestable*, dideskripsikan dengan XML, dan dibuat dengan GUI.
6. RapidMiner dapat terintegrasi dengan berbagai proyek data mining dengan aplikasi Weka dan statistic R.

Langkah-langkah menggunakan Rapidminer pada algoritma naïve bayes sebagai berikut :

1. Buka aplikasi Rapidminer, pada bagian operators ke bagian *search*, kemudian ketikan *Read CSV* atau *Read Excel* (Sesuai Type data yang digunakan).

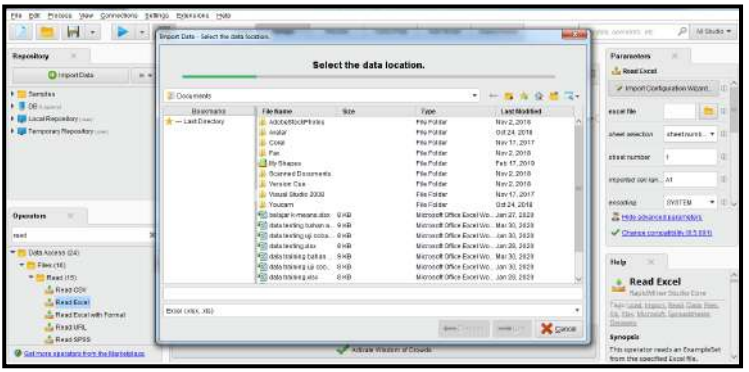


Gambar 7.1: Tampilan Utama Aplikasi Rapidminer

2. *Drag* dan *drop* operator *Read Excel* ke bagian *process*.

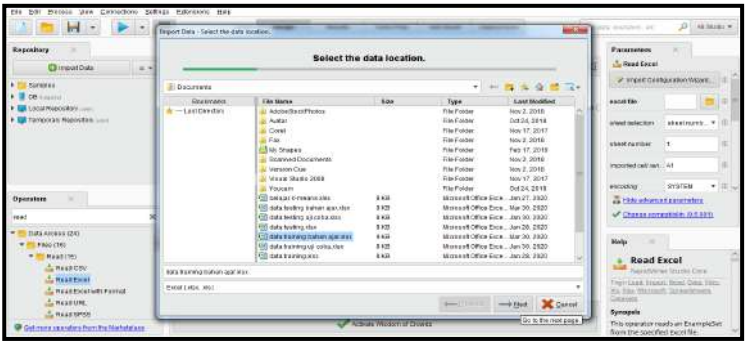
Gambar 7.2: Proses membaca dataset

3. Kemudian pada bagian *Parameters*, klik tombol *button Import Configuration Wizard* (berisi file data latih).



Gambar 7.3: Proses Import Data Latih

4. Pilih lokasi penyimpanan dataset, kemudian pilih Next.

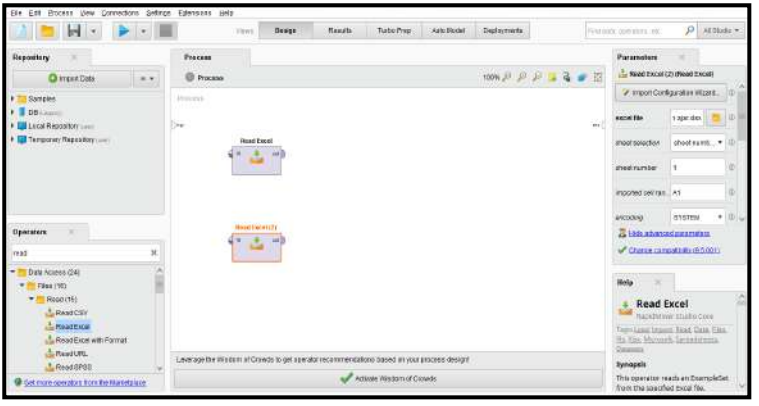


Gambar 7.4: Import Data Latih dari Direktori

5. Klik Next kembali pada langkah anotasi berikut.
6. Ubah tipe data pada atribut-atribut tersebut, kemudian klik *Finish*.

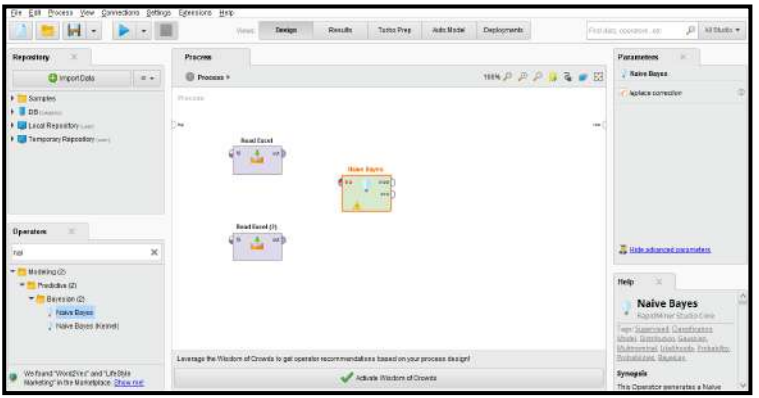
Gambar 7.5: Proses Format Tipe Data

7. Kemudian masukan data excel lagi seperti langkah-langkah di atas yang berisi data uji.



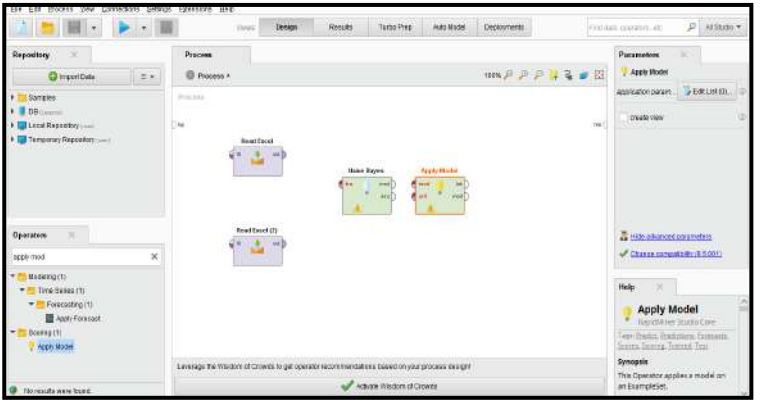
Gambar 7.6: Proses Import Data Uji

8. Lalu pada bagian operators ke bagian *search*, kemudian ketikan Naïve Bayes.
9. *Drag* dan *drop* operator Naïve Bayes ke bagian *process*.



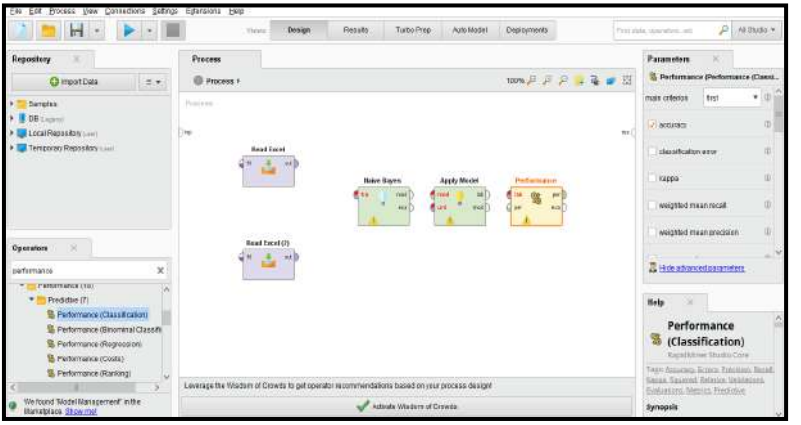
Gambar 7.7: Proses Import Fungsi/Model Algoritma Naive Bayes

10. Lalu pada bagian operators ke bagian search, kemudian ketikan *Apply Model*.
11. *Drag* dan *drop* operator *Apply Model* ke bagian *process*.



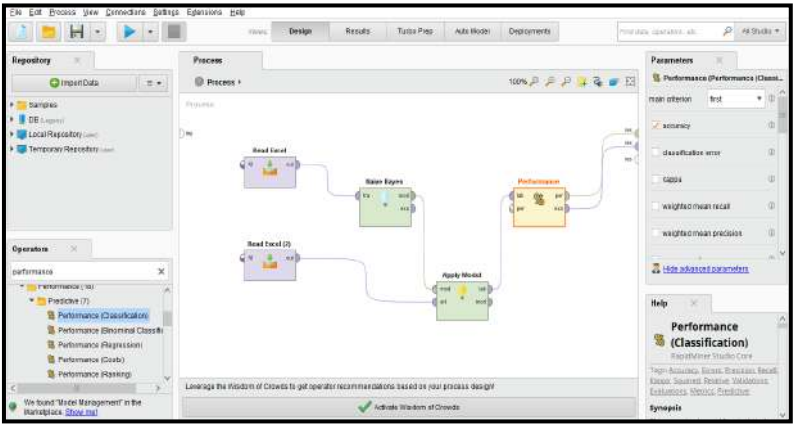
Gambar 7.8: Proses Aplly Model Data Latih

12. Lalu pada bagian operators ke bagian *search*, kemudian ketikan *Performance*.
13. *Drag* dan *drop* operator *Performance* ke bagian *process*.



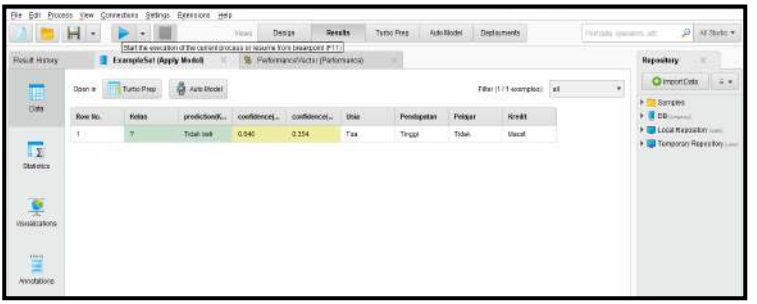
Gambar 7.9: Proses Pengkuran *Performance*

14. Hubungkan *port output* pada operator dengan *port* hasil.



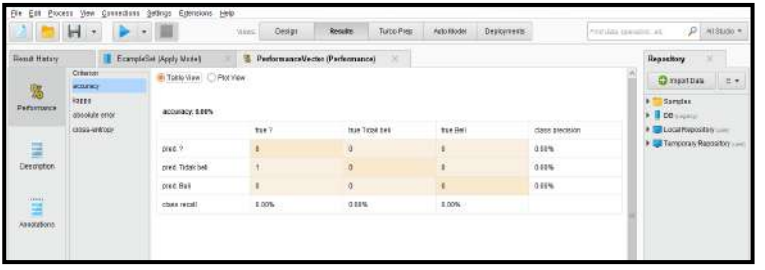
Gambar 7.10: Proses Penghubungan Semua Elemen dan Operator

15. Klik *Run Process*.

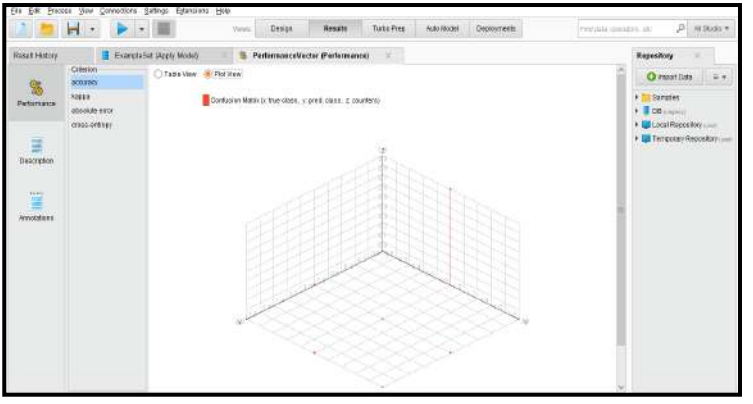


Gambar 7.11: Proses Klasifikasi dan Analisis Dataset

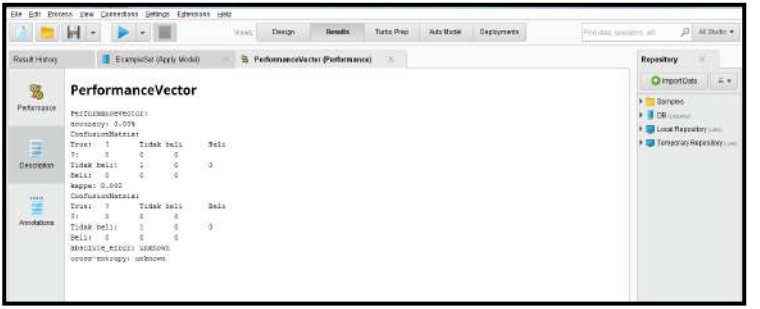
16. Tampilan hasil analisis menggunakan naive bayes dengan aplikasi rapidminer.



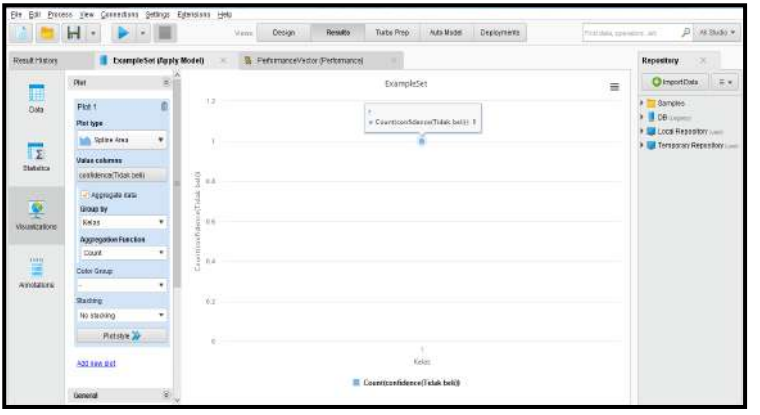
Gambar 7.12: Hasil Klasifikasi dan Analisis Performance



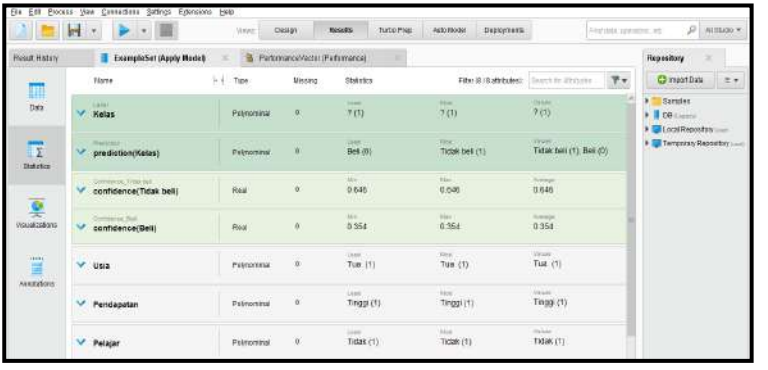
Gambar 7.13: Hasil Visualisasi Klasifikasi dan Analisis Performance Dalam Bentuk Plot View



Gambar 7.14: Hasil Performance Vector



Gambar 7.15: Hasil Klasifikasi dan Analisis Performance Dengan Plot View



Gambar 7.16: Hasil Klasifikasi dan Analisis Performance Dengan Statistics View

Bab 8

Implementasi Data Mining dengan Regresi Linear Berganda

8.1 Pendahuluan

Data mining merupakan sebuah proses untuk memperoleh informasi yang bermanfaat dan diperoleh tersebut diperoleh dari Gudang data yang besar. Dalam implementasinya, data mining dapat diolah dalam beberapa Teknik pengolahan data seperti Estimasi, Prediksi, Klasifikasi, Klastering dan Asosiasi (Panggabean, Buulolo and Silalahi, 2020). Pada pembahasan ini, penulis akan membahas mengenai teknik data mining dengan model estimasi menggunakan algoritma regresi linear berganda.

8.2 Definisi Regresi Linear Berganda

Estimasi digunakan untuk menerka sebuah nilai yang belum diketahui, missal menerka penghasilan seseorang ketika informasi mengenai orang tersebut diketahui, salah satu metode yang akan dibahas adalah Multiple Regression.

Multiple Regression atau Regresi berganda adalah regresi yang memiliki satu variabel dependent (tidak bebas) dan lebih dari satu variabel independent (bebas) (Gunawan, 2019).

Algoritma regresi merupakan salah satu teknik analisis statistika yang digunakan untuk menggambarkan hubungan antara satu variabel respon dengan satu atau lebih variabel penjelas (Nofriansyah, 2015). Tujuan regresi ini untuk mencari garis lurus sedekat mungkin dengan semua titik untuk mewakili titik-titik tersebut. Salah satu algoritma yang digunakan dalam prediksi adalah regresi linier. Algoritma regresi linier merupakan analisis satistika yang memodelkan hubungan beberapa variabel menurut bentuk hubungan persamaan linier eksplisit. Persamaan linier eksplisit adalah persamaan linier yang menempatkan suatu peubah secara tunggal pada salah satu persamaan.

Dalam analisis regresi dikenal 2 jenis variabel yaitu (Syahputra, Halim and Perangin-angin, 2018):

1. Variabel dependen disebut juga variabel respon yaitu variabel terikat yang keberadaannya dipengaruhi oleh variabel lainnya dan dinotasikan dengan variabel Y.
2. Variabel bebas atau prediktor disebut juga dengan variabel independen yaitu variabel yang tidak terikat (tidak dipengaruhi oleh variabel lainnya) dan dinotasikan dengan X.

Untuk data yang digunakan pada metode regresi linear berganda adalah data yang berhubungan dengan skala interval atau rasio dan umumnya digunakan untuk meramalkan atau memprediksi (Aspian Nur, 2019). Baiklah, pada pembahasan kali ini kita akan membahas permasalahan "Memprediksi Tingkat Pemahaman Mahasiswa terhadap Matakuliah yang diambil menggunakan Algoritma Regresi Linear Berganda".

Tahap yang dilakukan adalah sebagai berikut:

1. Mengumpulkan data
2. Menerapkan Metode
3. Menentukan Variabel
   a Variabel Dependent
   b Variabel dependent
4. Melakukan Tahap Pengujian dengan Regresi Linear Berganda

5. Hasil dan Analisis Hasil
6. Kesimpulan

Mari kita bahas tahap demi tahap pada kasus yang akan kita bahas:

1. Mengumpulkan data

Data yang digunakan adalah data mahasiswa yang mengampu matakuliah algoritma dan Pemrograman pada AMIK Tunas Bangsa

Tabel 8.1: Data set Yang digunakan

No	NIM	Nama Mahasiswa	Hadir	Tugas	Formatif	UTS	UAS	HASIL
1	201702030001	ADE TRI SUHARDI	86	70	70	70	70	79
2	201702030002	AYU ASTARI	100	80	80	80	80	90
3	201702030003	AYU LESTARI	100	80	80	80	80	90
4	201702030004	AYU SAFITRI	100	70	70	70	70	80
5	201702030005	DWI PUSPITA RANI	93	70	70	70	70	79
6	201702030006	DWIKY ROMARIO SARAGIH	86	70	70	70	70	79
7	201702030007	ERAWATI	100	80	80	80	80	90
8	201702030008	ETTY ISNAENY SUPRATMAN	100	80	80	80	80	90
9	201702030009	INDAH ANDRIANI DAMANIK	100	80	80	80	80	90
10	201702030010	KARINA NOVITRI	97	70	70	70	70	80
11	201702030011	MUHAMMAD FAISAL	79	70	70	70	70	78
12	201702030012	NIKO PIORADO BUTAR BUTAR	72	70	70	70	70	77
13	201702030013	NINDA PAJAR QARIAH	100	80	80	80	80	90
14	201702030014	PUTRI AYU INDAH SARI	100	80	80	80	80	90
15	201702030015	RAMITA JULIANA AMBARITA	100	80	80	80	80	90
16	201702030017	SALSA PRENNY PERANGIN-ANGIN	100	80	80	80	80	90
17	201702030018	SUSANTI	100	80	80	80	80	90
18	201702030019	WIDYA KRISTIANTI PANJAITAN	93	70	70	70	70	79
19	201702030020	WINDA SIAHAAN	100	80	80	80	80	90

(Sumber AMIK Tunas Bangsa Pematangsiantar)

2. Menerapkan Metode

Metode yang digunakan untuk menyelesaikan permasalahan yang dibahas sudah dijelaskan pada pendahuluan yaitu menggunakan Regresi Linear

Berganda Untuk memprediksi tingkat pemahaman mahasiswa dalam mengampu matakuliah. Pada pembahasan ini, kita hanya fokuskan pada satu matakuliah saja. Selanjutnya, kita akan kembangkan sesuai dengan kebutuhan masing-masing.

Dalam Regresi Linear Berganda, tahapan yang dilakukan adalah sebagai berikut:

Membedakan variable independent (X) dengan variable dependent (Y), berikut ini adalah persamaan yang digunakan:

$Y=a+b_1X_1+b_2X_2+\ldots b_nX_n$.....8.1

di mana:

Y → Variabel terikat

a → konstanta

b_1, b_2 → Koefisien Regresi

X_1, X_2 →Variabel Independent

3. Menentukan Variabel

Dataset pada table 8.2 merupakan data uji dalam memprediksi tingkat pemahaman mahasiswa terhadap matakuliah Algoritma dan Pemrograman.

a. Variabel Independent

Kita asumsikan variable Independent (X) sebagai variable bebas yang digunakan sebagai variable predictor

Tabel 8.2: Variabel Independent

X_1	X_2	X_3	X_4	X_5
86	70	70	70	70
100	80	80	80	80
100	80	80	80	80
100	70	70	70	70
93	70	70	70	70
86	70	70	70	70
100	80	80	80	80
100	80	80	80	80

X_1	X_2	X_3	X_4	X_5
100	80	80	80	80
97	70	70	70	70
79	70	70	70	70
72	70	70	70	70
100	80	80	80	80
100	80	80	80	80
100	80	80	80	80
100	80	80	80	80
100	80	80	80	80
93	70	70	70	70
100	80	80	80	80

Pada table 8.2 ada 5 variabel independent yang digunakan sebagai predictor yaitu X_1, X_2, X_3, X_4, X_5

b. Variabel Dependent

Untuk Variabel Dependent (Y) ada satu parameter terikat yang nilainya dipengaruhi oleh variable bebas. Jumlah Variabel maksimal 1 (satu).

Tabel 8.3: Variabel Dependent

Y
79
90
90
80
79
79
90
90
90
79
80
78
77

Y
90
90
90
90
90
79

Catatan :

Untuk mengingat kembali berikut hasil penentuan variable pada kasus prediksi ini:

Tabel 8.4: Variabel Independent dan Variabel Dependent

Variabel Pengujian					
Independent					**Dependent**
X_1	X_2	X_3	X_4	X_5	Y

4. Pengujian dengan Metode Regresi Linear Berganda

Tahap pertama gunakan table bantu dengan menggunakan rumus dari persamaan 8.2, 8.3 dan 8.4

Tabel bantu sebagai berikut:

Tabel 8.5: Tabel Bantu Perhitungan Pada Regresi Linear Berganda

X_1	X_2	X_3	X_4	X_5	Y	X_1Y	X_2Y	X_3Y	X_4Y	X_5Y	X_1^2	X_2^2	X_3^2	X_4^2	X_5^2
86	70	70	70	70	79	6760	5502	5502	5502	5502	7396	4900	4900	4900	4900
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	70	70	70	70	80	8000	5600	5600	5600	5600	10000	4900	4900	4900	4900
93	70	70	70	70	79	7375	5551	5551	5551	5551	8649	4900	4900	4900	4900
86	70	70	70	70	79	6760	5502	5502	5502	5502	7396	4900	4900	4900	4900
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
97	70	70	70	70	80	7731	5579	5579	5579	5579	9409	4900	4900	4900	4900

79	70	70	70	70	78	6154	5453	5453	5453	5453	6241	4900	4900	4900	4900
72	70	70	70	70	77	5558	5404	5404	5404	5404	5184	4900	4900	4900	4900
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400
93	70	70	70	70	79	7375	5551	5551	5551	5551	8649	4900	4900	4900	4900
100	80	80	80	80	90	9000	7200	7200	7200	7200	10000	6400	6400	6400	6400

Pada table 8.4 merupakan table bantu yang digunakan untuk menghitung persamaan 8.2,8.3 dan 8.4. tujuannya agar lebih memudahkan kita dalam melakukan proses perhitungan manualnya. Selanjutnya, dari perhitungan di atas mari kita lakukan perhitungan total dari record yang ada untuk menggunakan persamaan di atas.

Tabel 8.6: Hasil Perhitungan Jumlah Pada variable yang digunakan

No	Variabel	Total
1	X_1	1806
2	X_2	1440
3	X_3	1440
4	X_4	1440
5	X_5	1440
6	Y	1621
7	X_1Y	154712
8	X_2Y	123342
9	X_3Y	123342
10	X_4Y	123342
11	X_5Y	123342
12	X_1^2	172924
13	X_2^2	109600
14	X_3^2	109600
15	X_4^2	109600
16	X_5^2	109600

Selanjutnya, mari kita lakukan perhitungan manual dari persamaan 8.2, 8.3 dan 8.4 yang digunakan. Untuk melengkapi perhitungan manual sebanyak variable yang digunakan berikut disajikan matriks perkalian matriks dengan jumlah variable terikat sebanyak 5 parameter.

Tabel 8.7: Matriks Perkalian Variabel Depeden

X_1X_1	X_1X_2	$X1X_3$	X_1X_4	X_1X_5
7396	6020	6020	6020	6020
10000	8000	8000	8000	8000
10000	8000	8000	8000	8000
10000	7000	7000	7000	7000
8649	6510	6510	6510	6510
7396	6020	6020	6020	6020
10000	8000	8000	8000	8000
10000	8000	8000	8000	8000
10000	8000	8000	8000	8000
9409	6790	6790	6790	6790
6241	5530	5530	5530	5530
5184	5040	5040	5040	5040
10000	8000	8000	8000	8000
10000	8000	8000	8000	8000
10000	8000	8000	8000	8000
10000	8000	8000	8000	8000
10000	8000	8000	8000	8000
8649	6510	6510	6510	6510
10000	8000	8000	8000	8000

X_2X_1	X_2X_2	X_2X_3	X_2X_4	X_2X_5
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
7000	4900	4900	4900	4900
6510	4900	4900	4900	4900
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6790	4900	4900	4900	4900
5530	4900	4900	4900	4900
5040	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6510	4900	4900	4900	4900
8000	6400	6400	6400	6400

X_3X_1	X_3X_2	X_3X_3	X_3X_4	X_3X_5
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400

X_4X_1	X_4X_2	X_4X_3	X_4X_4	X_4X_5
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400

X_3X_1	X_3X_2	X_3X_3	X_3X_4	X_3X_5
8000	6400	6400	6400	6400
7000	4900	4900	4900	4900
6510	4900	4900	4900	4900
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6790	4900	4900	4900	4900
5530	4900	4900	4900	4900
5040	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6510	4900	4900	4900	4900
8000	6400	6400	6400	6400

X_4X_1	X_4X_2	X_4X_3	X_4X_4	X_4X_5
8000	6400	6400	6400	6400
7000	4900	4900	4900	4900
6510	4900	4900	4900	4900
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6790	4900	4900	4900	4900
5530	4900	4900	4900	4900
5040	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6510	4900	4900	4900	4900
8000	6400	6400	6400	6400

X_5X_1	X_5X_2	X_5X_3	X_5X_4	X_5X_5
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
7000	4900	4900	4900	4900
6510	4900	4900	4900	4900
6020	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6790	4900	4900	4900	4900

X_5X_1	X_5X_2	X_5X_3	X_5X_4	X_5X_5
5530	4900	4900	4900	4900
5040	4900	4900	4900	4900
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
8000	6400	6400	6400	6400
6510	4900	4900	4900	4900
8000	6400	6400	6400	6400

$\sum Y=an+b_1\sum X_1+b_2\sum X_2+b_3\sum X_3+ b_4\sum X_4+ b_5\sum X_5$
$1621=86a+1806b1+1440b_2+1440b_3+1440b_4+1440b_5$ ..... a

$\sum X_1Y=a\sum X_1+b_1 \sum X_1X_1+b_2 \sum X_1X_2+b_3 \sum X_1X_3+b_4 \sum X_1X_4+b_5 \sum X_1X_5$
$154712=1806.a+7396\ b_1+6020\ b_2+6020\ b_3+6020\ b_4+6020\ b_5$ ..... b

$\sum X_2Y=a\sum X_2+b_1 \sum X_2X_1+b_2 \sum X_2X_2+b_3 \sum X_2X_3+b_4 \sum X_2X_4+b_5 \sum X_2X_5$

$123342=1440.a+6020\ b_1+4900\ b_2+4900\ b_3+4900\ b_4+4900\ b_5$ c

$\sum X_3Y= a\sum X_3+b_1\sum X_3X_1+b_2. \sum X_3X_2+b_3. \sum X_3X_3+b_4. \sum X_3X_4+b_5. \sum X_3X_5$
$123342=1440.a+6020\ b_1+4900\ b_2+4900\ b_3+4900\ b_4+4900\ b_5$ ..... d

$\sum X_4Y= a\sum X_4+b_1 \sum X_4X_1+b_2 \sum X_4X_2+b_3 \sum X_4X_3+b_4 \sum X_4X_4+b_5. \sum X_4X_5$
$123342=1440.a+6020\ b_1+4900\ b_2+4900\ b_3+4900\ b_4+4900\ b_5$ ..... e

$\sum X_5Y= a\sum X_5+b_1 \sum X_5X_1+b_2 \sum X_5X_2+b_3 \sum X_5X_3+b_4 \sum X_5X_4+b_5 \sum X_5X_5$
$123342=1440.a+6020\ b_1+4900\ b_2+4900\ b_3+4900\ b_4+4900\ b_5$ ..... f

Dari persamaan a dan b lakukan metode subsitusi dengan cara mengalikan persamaan a dengan subsitusi persamaan a dan persamaan b untuk mencari koefisien regresi b_1, b_2, b_3, b_4 dan b_5.

Kalikan persamaan a =21 dan persamaan b =1

1621 =86 a+1806 b1+1440 b2+1440 b3+1440 b4+1440 b5

154712=1806.a+7396 b1+6020 b2+6020 b3+6020 b4+6020 b5

34033=1806 a+37926 b1+30240 b2+30240 b3+30240 b4+30240 b5 g

Selanjutnya, subsitusi persamaan a dan c

1621 =86 a+1806 b1+1440 b2+1440 b3+1440 b4+1440 b5

123342=1440.a+6020 b1+4900 b2+4900 b3+4900 b4+4900 b5

Kalikan 10 persamaan a dan persamaan c =3

Sehingga dapat persamaan h sebagai berikut:

-35820=-3460 a+0 b1 -300 b2 -300 b3 -300 b4 -300 b5 h

Kemudian lakukan subsitusi persamaan *a* dan *d* dengan mengalikan 245 pada persamaan *a* dan persamaan *b* 72

Sehingga hasil pada persamaan di bawah ini:

1621 =86 a+1806 b1+1440 b2+1440 b3+1440 b4+1440 b5

123342=1440.a+6020 b1+4900 b2+4900 b3+4900 b4+4900 b5

-8411577 =-82610 a+9030 b1+0 b2+0 b3+0 b4+0 b5 i

Dari subsitusi persamaan di atas, didapat nilai *a* dan koefisien regresi pada tabel berikut:

Tabel 8.8: Nilai Konstanta dan Koefisien Regresi Berdasarkan Data Uji

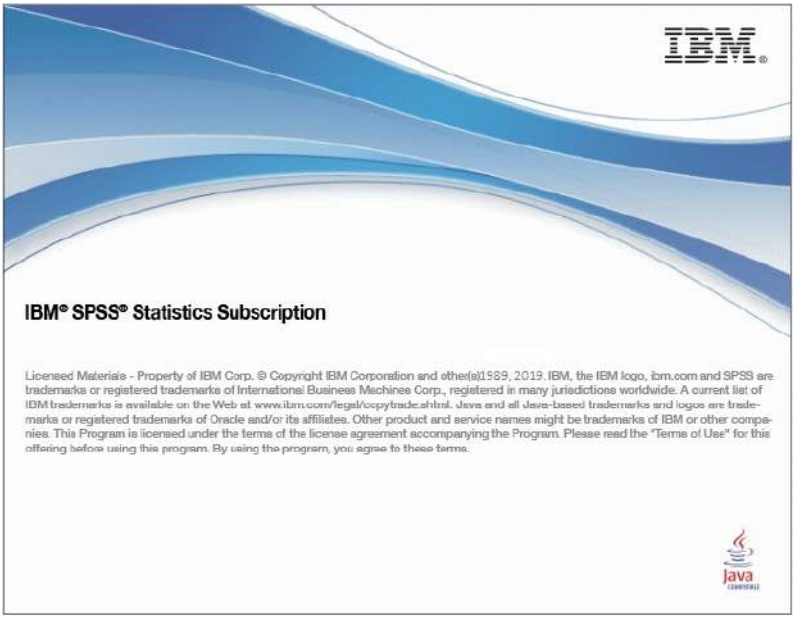
Nilai Konstanta dan Koefisien Regresi	
a	34,701
b_1	0,532
b_2	0,532
b_3	0,890
b_4	0,890
b_5	0,890

Persamaan yang diperoleh:
$Y=a+ b_1X_1+b_2X_2+b_3X_3+b_4X_4+b_5X_5$
$Y= 34,701+0,532+0,890+0,890+0,890+0,890$

Kemudian tentukan Koefisien Regresi Berganda (R), Koefisien determinasi (R^2) dan F Hitung dengan hasil sebagai berikut:

R= 0,781
Koefisien Determinasi (R^2) = 0,610
F hitung= 26,605

Berikut hasil pengujian prediksi dengan menggunakan Software



Gambar 8.1: Halaman Depan Aplikasi SPSS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.781[a]	.610	.587	3.661482333994832
2	1.000[b]	1.000	1.000	.000000063689908

a. Predictors: (Constant), X1

b. Predictors: (Constant), X1, X5

Gambar 8.2: Hasil Koefisien Korelasi dan Determinasi

Gambar 8.2 di atas adalah hasil Koefisien Korelasi, Determinasi dan Estimasi pada pengujian terhadap dataset yang digunakan.

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	T	Sig.
1	(Constant)	34.701	9.845		3.525	.003
	X1	.532	.103	.781	5.158	.000
2	(Constant)	.000	.000		.000	1.000
	X1	.100	.000	.147	39079593.379	.000
	X5	1.000	.000	.890	237034076.021	.000

a. Dependent Variable: Y

Gambar 8.3: Nilai Konstanta dan Koefisien Regresi

Model pada gambar 8.3 merupakan nilai konstanta yang diperoleh dengan aplikasi SPSS yang terdiri dari variabel X1 dan X5 yang memiliki pengaruh terhadap varibel dependent dengan nilai pada hasil lebih kecil dari nilai target yang menjadi ketetapan yaitu X1 dan X5 lebih kecil dari nilai target.

Kesimpulan yang diperoleh pada pengujian di atas, mengenai prediksi tingkat

pemahaman mahasiswa dalam mengampuh matakuliah adalah sebagai berikut:

1. Ada 2 model yang dihasilkan dengan menggunakan Software SPSS mengenai kasus yang dibahas dalam memprediksi tingkat pemahaman mahasiswa dalam mengampuh matakuliah.
2. Dari 5 variabel bebas sebagai predictor yaitu: Hadir (X1), Tugas (X2), Formatif (X3), UTS (X4) dan UAS (X5) terhadap Variabel terikat Hasil (Y) yang memiliki pengaruh signifikan adalah variabel X1 dan X5 karena nilai nya lebih kecil dari 0,5.
3. Prediksi dengan konsep Estimasi dengan metode Regresi berganda digunakan untuk mengetahui pengaruh variabel terikat dependent(terikat) terhadap beberapa variabel bebas.

Bab 9

Performa Klasifikasi Dataset dengan Metode Correlation Based Feature Selection (CFS)

9.1 Klasifikasi

Menurut KBBI "klasifikasi adalah penyusunan kelas yang dibuat bertingkat dalam suatu kelompok atau golongan menurut standar yang telah ditetapkan berdasarkan kebutuhan". Dalam data mining klasifikasi digunakan sebagai cara untuk menentukan kelas data berdasarkan kategori kelas data yang baru sesuai dengan standar yang telah ditetapkan menurut sumber, jenis, sifat dan fungsinya. Kelas data yang telah diklasifikasi dapat digunakan sebagai model data baru sesuai dengan kepentingan. Klasikasi juga dapat digunakan untuk meprediksi suatu kelas menjadi model data berdasarkan atribut yang dimiliki, atribut dari kelas yang dimiliki dapat berdiri sendiri tanpa ada hubungan denga atribut lainnya.

Data merupakan kumpulan fakta mentah yang belum memiliki makna sehingga perlu diolah agar menjadi informasi yang memiliki nilai. Ada beberapa cara yang dapat digunakan untuk mengolah data agar memiliki nilai salah satunya dengan melakukan penelitian dan uji coba. Menurut Liang Gie, Data adalah kumpulan fakta atau kejadian, yang berupa kenyataan atau apapun yang mengandung makna suatu pengetahuan yang dijadikan sebagai bahan menyusun informasi, keterangan, membuat kesimpulan, atau mengambil keputusan dalam suatu kejadian.

Dalam proses pengklasifikasian, data yang digunakan sangat memengaruhi hasil pengelompokan sehingga perlu diperhatikan jenis, sifat dan sumbernya.

9.1.1 Klasifikasi Data Menurut Jenisnya

1. Data Hitung

Data hitung merupakan data yang diperoleh dari proses perhitungan baik berupa persentase maupun jumlah total nilai hitung data yang tercatat dan tersimpan.

Contoh data hitung: Data pemilu, data nasabah/ karyawan dan lain-lain.

2. Data Ukur

Data ukur adalah data yang menunjukkan kapasitas atau ukuran dari suatu nilai.

Contoh : suhu yang ditunjukkan pada thermometer, IPK mahasiswa pada akhir semester dan lain-lain.

9.1.2 Klasifikasi Data Menurut Sifatnya

1. Kuantitatif

Data kuantitatif merupakan data yang dapat dihitung dan dijumlahkan, berupa bilangan atau variable namun kadang sulit untuk diukur.

Contoh data kuantitatif: nomor, angka, statistic, grafik dan lain-lain.

2. Kualitatif

Data kualitatif adalah data yang perlu dilakukan analisis untuk dapat dilakukan pengelompokan berdasarkan jenis datanya. Data ini tidak berbentuk bilangan atau angka tetapi berupa objek.

Contoh data kuantitatif: analisa hasil keputusan, pengamatan dan lain-lain..

9.1.3 Klasifikasi Data Menurut Sumbernya

1. Data Internal

Data internal diperoleh dari hasil pengolahan data yang dilakukan secara mandiri, tidak bersumber dari luar, dapat dipercaya dan dapat dipertanggung jawabkan.

Contoh : Data perusahaan.

2. Data Eksternal

Data ekternal adalah data yang diperoleh dari luar, tidak dioleh secara mandiri dan dapat dipengaruhi oleh berbagai aspek, termasuk metode pengolahan datanya.

Contoh: Data yang didapat dari penyedia layanan pengambilan data.

Contoh klasifikasi dalam dunia medis

Dari penelitian yang telah dilakukan untuk klasifikasi penyakit Orang-orang yang sakit dikelompokkan menurut kesamaan gejala, tanda, perubahan cairan tubuh atau jaringan, fungsi fisiologis, perilaku. Prognosis atau gabungan dari gambaran tesebut sesuai dengan diagnose yang dilakukan oleh para dokter atau pakar. Pengelompokan dilakukan tergantung pada kesamaan dalam hal pengalaman-pengalaman yang terbukti sebagai sebab terjadinya penyakit (Sarini Vita Dewi, Adhistya Erna Permanasari, 2014).

9.1.4 Performa Klasifikasi

Performa klasifikasi digunakan untuk menunjukkan hubungan antara hasil kerja dan kelas prediksi. Kelas predikasi bibagi ke dalam dua katagori yaitu positif dan negatif. Seperti ditunjukkan pada tabel 9.1 (Sarini Vita Dewi, Adhistya Erna Permanasari, 2014)

Tabel 9.1: Matrik prediksi

	Positif	**Negatif**
AP	TP	FN
AN	FP	TN

Keterangan:

AP : Actual Positif

AN: Actual Negatif

FN: False Negatif

TN: True Negatif

True positive (TP) merupakan instance positif yang diklasifikasikan sebagai nilai positif. Jika diklasifikasikan sebagai negatif, maka dikatakan sebagai *false negative* (FN). *True negative* (TN) adalah instance negatif dan diklasifikasikan sebagai nilai negatif. Jika diklasifikan sebagai positif, maka dikatakan sebagai *false positive* (FP).

9.1.5 Data Processing

Data processing adalah pemprosesan data yang dilakukan dengan memanipulasi data oleh komputer. Dalam tahap ini, data yang dimasukkan ke komputer diproses untuk interpretasi. Pemprosesan dilakukakan dengan menggunakan algoritma pada mesin learning. Proses ini akan berdeda untuk tergantung pada algoritma yang digunakan. Data processing digunakan untuk mengurangi tingkat gangguan dalam suatu data, baik berupa noise, data hilang atau tidak lengkap, redudansi data, tidak konsisten. Data processing diperlukan karena apabila kualitas data awal rendah maka kualitas data mining pun menjadi rendah.

Metode data processing di antaranya adalah data cleaning, integrasi data, data reduction dan transformasi data.

1. Data cleaning digunakan untuk mengurangi atau menghilangkan noise dan memperbaiki data yang tidak konsisten.
2. Integrasi data digunakan untuk menggabungkan beberapa sumber kedalam penyimpanan data.
3. Data reduction digunakan untuk mengurangi ukuran data dan menghilangkan redudansi data atau data yang sama (duplikat).
4. Transformasi data digunakan untuk mengubah atribut data agar memiliki skala yang kecil (Sarini Vita Dewi, Adhistya Erna Permanasari, 2014).

9.2 Dataset

Dataset adalah koleksi atau kumpulan data dalam suatu media penyimpanan digital (memory) yang memiliki nilai, setiap nilai mewakili variable tertentu dan setiap variable memiliki tujuan sesuai dengan data yang dimaksud. Dataset atau

kumpulan data dapat berupa dokumen atau file. Dalam data terbuka, dataset adalah unit untuk mengukur informasi yang dapat digunakan untuk informasi public, sehingga penggunaanya menjadi lebih luas. Karena penggunaan yang luas sumber data dan waktunya menjadi tidak rasional dan ini dapat meningkatkan kesulitan untuk mencapai kesepakatan secara bersama-sama.

Format penyimpanan dataset berbentuk file.xsd. relasi antar tabel ini biasa dibuat dalam database. Cara menyimpan data mejadi data set adalah dengan membari nama file dalam format xds, dan file ini akan otomatis tersimpan dalam folder App_Code.

9.2.1 Tipe dari Himpunan Data (Dataset)

Tipe dari himpunan data yang ada dalam dataset adalah (Jimmy Nganta Ginting, 2020):

1. Record Data, merupakan data yang terdiri dari sekumpulan record atau inputan yang masing-masing data terdiri dari satu set atribut yang tetap.

   Yang termasuk dalam tipe data record adalah:

   a) Data Matrix

      Data matrix merupakan objek data yang memilki himpunan atribut numerik yang sama, objek data tersebut dapat dianggap sebagai titik-titik dalam ruang multi dimensi, di mana masing-masing dimensi menyatakan suatu atribut yang berbeda.

   b) Data Dokumen

      Tipe data dokumen ini, tiap dokumen menjadi satu vector "term". Tiap term merupakan suatu komponen (atribut) dari vector tersebut. Nilai data dari tiap komponen dinyakan dengan beberapa kali kemunculan terms dalam suatu dokumen.

   c) Data Transaksi

      Data transaksi merupakan tipe khusus dari record atau transaksi data, di mana tiap record mewakili suatu set produk. Contoh penjualan pada supermarket. Himpunan produk yang dibeli oleh pelanggan dalam satu

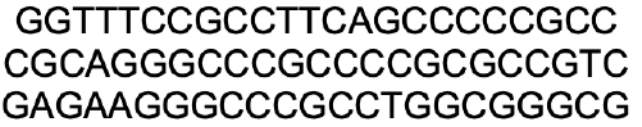
kali belanja merupakan satu transaksi, selama produk yang dibeli tersebut adalah produk yang berada dalam satu toko.

2. Data Graph

Data graph Merupakan data yang berbentuk graph, data ini terdiri dari simpul (node) dan rusuk (edge) atau tepi. Contoh data graph di antaranya adalah HTML, Link (dalam WWW) dan struktur molekul dalam jaringan.

3. Data Terurut (Ordered Data)

Data-data yang memperhatikan urutan nilai-nilainya. Yang termasuk dalam data terurut adalah dapat dilihat pada Gambar 9.1 di bawah ini.



Gambar 9.1: Contoh data terurut (Hermawati, 2013)

9.2.2 Jenis Atribut Data set

Nilai atribut merupakan angka atau simbol yang memberi nilai pada suatu atribut tersebut. Perbedaan antara atribut dan nilai atribut, yaitu (Jimmy Nganta Ginting, 2020):

1. Atribut yang sama dapat dipetakan ke nilai atribut yang berbeda.

   Contoh :

   Ketinggian dapat diukur dengan satuan kaki dan meter.

2. Atribut yang berbeda dapat dipetakan ke himpunan nilai yang berbeda.

   Contoh :

   Nilai atribut untuk ID dan umur adalah bilangan bulat, tetapi sifat dan nilai atribut dapat berbeda. ID tidak terbatas tetapi umur mempunyai nilai minimal dan nilai maksimal 21

Nilai Atribut dapat dibedakan dalam beberapa tipe yang berbeda, tergantung pada tipe domain yang digunakan, yaitu tergantung pada nilai tipe atribut yang diterima.

Adapun contoh dari tipe-tipe atribut, yaitu :

1. Atribut Katagori, atribut ini merupakan salah satu tipe yang domainnya merupakan sebuah himpunan simbol berhingga. Atribut katagori dibedakan menjadi dua tipe, yaitu:
   a) Nominal : Sebuah atribut dikatakan nominal jika nilai-nilainya tidak dapat diturunkan.

      Contoh : Jenis Kelamin, Warna Mata. Atribut nominal mempunyai sifat pembeda

   b) Ordinal : atribut ordinal memiliki nilai-nilai yang dapat diurutkan dalam beberapa cara atau tahap.

      Contoh : Skala 1-10, Grade, Tinggi (Tinggi, Sedang, Pendek).

      Pendidikan (S3, S2, S1, SMA, SMP, SD). Sifat dari atribut ordinal adalah berurutan dan berbeda-beda.

2. Tipe atribut kedua adalah numerik yang domainnya berupa bilangan rill atau integer.

   Atribut numerik juga dibedakan menjadi dua, yaitu:

   a) Interval : Untuk jenis atribut interval, tiap datanya mempunyai sifat yang berbeda atau memiliki perbedaan antara nilai- nilainya.

      Contoh : Suhu, baik dalam Celcius atau Fahrenheit, Tanggal, dan lain-lain.

      Dalam nilai interval tidak ada bedanya jika kita menyatakan bahwa suhu 20°C = dua kali dinginnya 10°C.

   b) Rasio : Dalam atribut jenis ini, perbedaan antara jenis nilai maupun rasio sangatlah berarti.

      Contoh : Suhu dalam kelvin, Panjang, Waktu, dan Jumlah.

Contoh lain, Kita dapat menyatakan orang yang berusia 40 tahun, dua kali lebih tua dari yang berusia 20 tahun.

Sedangkan atribut berdasarkan jumlah nilainya dibedakan menjadi dua, yaitu :

a Atribut Diskrit : Atribut ini hanya menggunakan sebuah himpunan yang memiliki nilai berhingga atau himpunan nilai tak berhingga yang dapat dihitung dan dijumlahkan.

Contoh : kode Zip, Jumlah atau Himpunan kata dalam suatu kumpulan dokumen. Atribut diskrit sering dinyatakan sebagai variable bilangan bulat.

b Atribut Kontinyu : Merupakan Atribut yang menggunakan bilangan rill sebagai nilai atributnya.

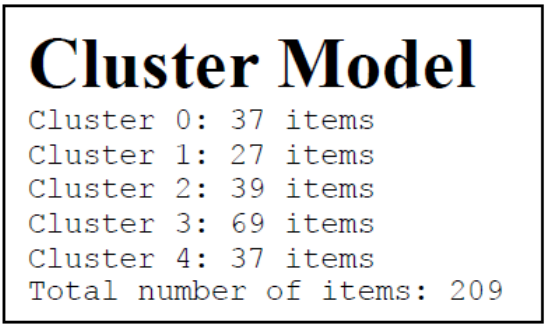
Contoh : Suhu, Ketinggian, Berat, dll.

Atribut kontinyu dinyatakan sebagai variable desimal.

9.3 Klastering

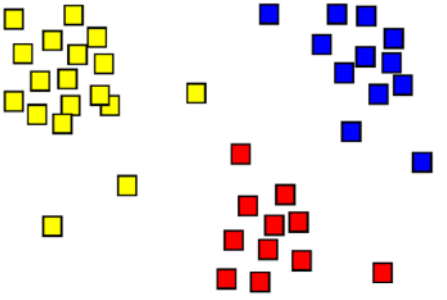
Klastering adalah metode atau cara yang digunakan untuk mengelompokkan data sesuai dengan kelasnya sehingga data yang sama akan berada dalam satu kelas data. Adanya klastering, akan memudahkan proses menemukan dokumen yang berada dalam satu klaster atau kelas berdasarkan kata kunci yang diinputkan oleh pengguna.

Proses klastering juga dapat digunakan untuk mengetahui pola kelompok data dengan menggunakan metode distance Performance. Distance performance dalam metode klastering dapat dihitung jika nilai pada setiap variable memiliki tipe numerik. Sehingga data yang digunakan dalam proses ini adalah data numerik (Nugroho, 2015)



Gambar 9.2: Klaster model (Nugroho, 2015)

Gambar 9.2 menunjukkan bahwa kelompok data berjumlah 5 klaster dengan masing-masing jumlah kelas berbeda antara satu klaster dengan klaster yang lainnya. Gambar 9.3 menunjukkan ilustrasi sederhana suatu klastering dari dataset dikelompokkan menjadi tiga kelas berdasarkan fitur-fitur tertentu tanpa adanya label atau target. Biasanya yang digunakan sebagai basis klastering adalah jarak atar kelas.



Gambar 9.3: Klastering dengan 3 pengelompokan data (Wibowo, 2017)

Pada klasifikasi, pengelompokan suatu dataset memerlukan label atau target untuk mengawasi (supervise) proses pembelajaran algoritma yang digunakan, sementara klastering tidak memerlukan label atau target di dalam proses pembelajaran algoritmanya. Atau dengan kata lain, pembelajaran pada klastering tidak memerlukan pengawasan (supervisor). Dengan demikian,

algoritma-algoritma untuk menyelesaikan masalah klastering dikategorisasikan ke dalam unsupervised learning atau pembelajaran yang tidak diawasi

9.4 Correlation based feature selection (CFS)

Data mining memiliki banyak data yang harus diseleksi agar dapat digunakan sesuain dengan kebutuhan. Sebelum digunakan data terlebih dahulu di proses atau yang dikenal dengan pre-processing. Pemprosessan data dilakukan guna untuk mendapatkan hasil analisis yang akurat dalam pemakaian teknik-teknik machine learning dan data mining. Pre-processing juga dapat membuat ukuran data menjadi lebih kecil tanpa mengubah informasi didalamnya.

Bagian dari pre-processing di antaranya:

1. Seleksi fitur

Seleksi fitur merupakan salah satu bagian dari Pre-processing yaitu proses pemilihan fitur yang akan digunakan untuk atribut, fitur dianggap relevan bila nilainya mewakili semua katagori secara sistematis. Fungsi utama seleksi fitur adalah memilih fitur mana yang akan dipilih dan dipakai saat pengolahan data nantinya. Metode yang digunakan untuk proses seleksi fitur sangat memengaruhi hasil dari kinerja mesin learning karena apabila terjadi kesalahan pemilihan fitur akan dapat menurunkan kinerja mesin learning, hal ini terjadi jika ada beberapa fitur dihilangkan, sedangkan fitur tersebut memiliki nilai prediktif yang mungkin bernilai tinggi dalam suatu kelas.

Dengan penggunaan Seleksi fitur diharapkan dapat meningkatkan kinerja pengklasifikasian data menjadi lebih akurat, akan tetapi pada implementasinya pengurangan suatu fitur dapat berpengaruh besar terhadap hasil klasifikasi. Pengaruh ini bisa menjadi baik, tetapi juga bisa menjadi kendala dalam proses pembelajaran, karena kemungkinan fitur yang dihilangkan mungkin memiliki nilai yang berpengaruh terhadap hasil klasifikasi.

Berbeda dengan proses menginput data baru, seleksi fitur dapat memproses data secara otomasi secara komputasi. Manfaat seleksi fitur untuk mesin learning dapat mencangkup pengurangan jumlah data yang dibutuhan untuk mencapai

pembelajaran, meningkatkan akurasi prediksi, pembelajaran lebih mudah dipahami dan mengurangi waktu eksekusi.

2. Correlation based feature selection (CFS)

Correlation based feature selection (CFS) merupakan salah satu algoritma seleksi fitur dalam teknik Pre-processing untuk melakukan klasifikasi data yang dikembangkan oleh Yu dan Liu. Klasifikasi data digunaka agar fitur yang dimiliki relevan dan sesuai dengan kelasnya walau telah terjadi pengurangan fitur pada saat pengolahan data. Fitur yang tidak relevan dapat mengakibatkan makna ganda dan salah pengertian terhadap data yang digunakan. Hasilnya dari seleksi fitur adalah pengolah data menjadi lebih cepat dan meningkatkan akurasi kinerja.

Menurut penelitian yang dilakukan oleh Mark A. Hall dalah tesisnya yang berjudul Correlation-based Feature Selection forMachine Learning, dia menyatakan "pemilihan fitur untuk tugas-tugas klasifikasi dalam mesin learning dapat dilakukan berdasarkan korelasi antara fitur, dan bahwa pemilihan fitur tersebut merupak prosedur yang dapat bermanfaat untuk algoritma pembelajaran mesin umum. Dalam penelitian ini menyajikan pemilih fitur berbasis korelasi (CFS) berdasarkan hasil penelian Mark memeriksa perilaku CFS dalam berbagai kondisi dan menunjukkan bahwa CFS dapat mengidentifikasi fitur-fitur yang berguna untuk pembelajaran mesin (Hall, 1999).

Algoritma CFS menyeleksi fitur dengan prinsip suatu fitur yang baik adalah fitur yang relevan terhadap kelas tetapi tidak redundan dan tidak ambigu terhadap fitur-fitur relevan lainnya. Fitur yang relevan adalah yang memiliki tingkat korelasi tinggi dengan kelas dan korelasi antar fiturnya rendah. Karena prinsip ini CFS mengidentifikasi fitur yang relevan dengan kelas dan tidak memiliki ketergantungan dengan fitur lainya sehingga tingkat relevansinya menjadi tinggi karena fitur yang dipilih mewakili semua katagori secara sistematis.

Berdasarkan penelitian yang dilakukan oleh Lie Yu dan Huan Liu, melakukan dua pendekatan untuk menyeleksi fitur yanitu dengan mengukur korelasi antara dua variable random/acak berdasarkan nilai classical linear dan teori informasi. (Sarini Vita Dewi, Adhistya Erna Permanasari, 2014).

3. Proses seleksi fitur dengan Algoritma CFS

Dalam proses seleksi fitur hal utama yang harus perhatikan adalah relevansi, untuk Algoritma CFS tahapan seleksi fitur yang dilakukan adalah sebagai berikut

1. Tahap pertama yang dilakukan adalah mengubah dataset kedalam bentuk diskret.
2. Tahap kedua yaitu menghitung korelasi antar fitur dan kelas, dengan cara memilih nilai fitur yang memiliki tingkat korelasi paling tinggi dengan kelas dan korelasi antar fiturnya rendah. Karakteristik ini disebut dengan merit.
3. Fitur yang memiliki nilai merit paling tinggi akan dipilih
4. Nilai cfs dapat dihitung dari nilai merit yang diperoleh. Nilai merit dapat diperoleh melalui persamaan 1-1

$$r_{zc} \frac{Kr_{zi}}{\sqrt{k+k(k-1)r_{cii}}} \quad \text{.....persamaan 1-1}$$

Keterangan

r_{zc} = Nili subnet fitur 'z' dengan fitur sebanyak k terhadap hasil keputusan

Kr_{zi} = Rata-rata korelasi antar fitur dengan kelas

r_{ii} = Rata-rata korelasi antar fitur

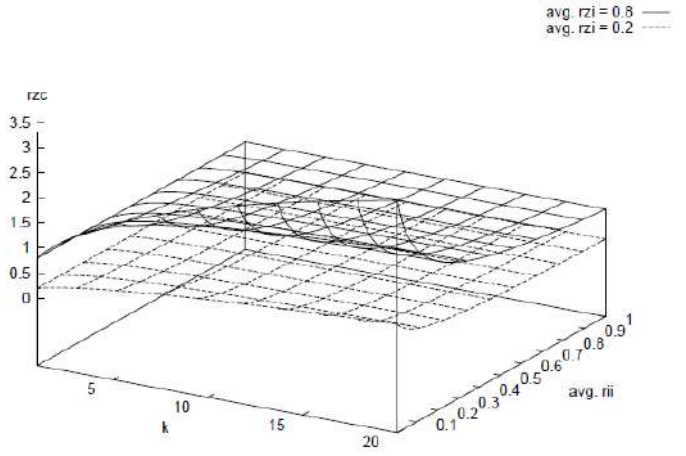
k = banyaknya fitur dalam suatu subset.

r_{zc} merupakan nilai konstribusi subet fitur dengan fitur sebanyak *k* terhadap hasil keputusan. Penggunaan metode CFS bisa saja tidak sesuai untuk semua seleksi fitur, tingkat keberhasilan sangat bergantung pada jenis dataset yang digunakan dan juga tujuan pengklasifikasiannya.

Persamaan 1-1 menjelaskan hubungan antar korelasi untuk variable yang telah relevan. Ini menunjukkan bahwa korelasi antar komposit dan variable luar merupakan fungsi dari jumlah komponen variable dengan inter-korelasi di antara keduanya. Peningkatan komponen secara subtansial meningkatkan korelasi antara komposit dan variable luar. Namun, tidak mungkin sekelompok komponen yang semuanya sangat berkorelasi dengan variable luar akan menghasilkan korelasi yang rendah satu sama lain dalam waktu bersamaan.

Menurut Hogarth, ketika penambahan kompenen tambahan dipertimbangkan, inter-korelasi yang rendah dengan komponen-komponen yang sudah dipilih kemungkinan akan lebih mendominasi.

Persamaan 1-1 digunakan sebagai "merit" dari himpunan bagian fitur dalam pengklasifikasian yang di lakukan. Dalam kondisi ini, z (variable eksternal) menjadi C(kelas). Hal ini digunakan untuk menghitung dan mengukur featureeclass koreasi antar fitur-fitur. Persamaan 1-1 dapat divisualisasikan dengan gambar 9.4 berikut ini



Gambar 9.4: Gambaran umum persamaan 1-1

Dari gambar 9.4, dapat diambil kesimpulan bahwa

- Semakin tinggi korelasi antar komponen luar, maka semakin tinggi pula komposit dan variable luar.
- Semakin rendah inter-korelasi antar komponen, maka semakin tinggi korelasi antara komposit dan variable luar.
- Ketika jumlah komponen dalam komposit bertambah (dengan asumsi penambahan komponen sama dengan komponen asli), maka korelasi antara komposit dan variable luar akan meningkat.

Tugas pembelajaran yang diawasi sering melibatkan fitur data yang berbeda, yang mana data tersebut bisa berupa data kontinu, ordinal, nominal atau biner.

Untuk menghitung suatu fitur jenis/tipe data harus seragam, contohnya adalah dengan menggunakan metode pre-processing Fayyd dan Iran untuk mengubah fitur kontinu menjadi nominal.

Contoh : Tingkat keakuratan prediksi keberhasilan seorang siswa dalam suatu bidang ilmu lebih tinggi bila dilihat dari gabungan beberapa jumlah tes yang dilakukan yaitu kemampuan untuk memahami materi tertulis, kemampuan belajar, ketangkasan dan lain sebagainya. Daripada hanya melihat dari satu uji tes yang mengukur lingkup terbatas dari satu materi.

Menurut riset yang dilakukan oleh Mark A. Hall "riset menunjukkan perbandingan algoritma CFS dengan algoritma wrapper (metode yang sering digunakan untuk seleksi fitur yang menggunakan algoritma pembelajaran target untuk mengevaluasi set fitur). Dalam banyak kasus CFS memberikan hasil yang sebanding dengan wrapper, dan secara umum, mengungguli wrapper pada dataset kecil. CFS dieksekusi berkali-kali lebih cepat daripada metode wrapper, yang memungkinkannya untuk digunakan dalam skala ke dataset yang lebih besar.

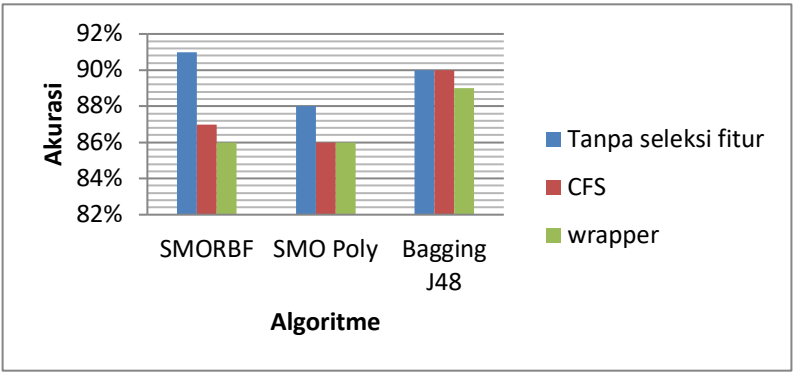
Pilihan fitur sangat memengaruhi hasil klasifikasi, dengan menggunakan Algoritma seleksi fitur CFS dapat meminimalisir terjadi kesalahan dalam proses pemilihan fitur. CFS digunakan sebagai metode pemilihan fitur dengan memperhatikan korelasi antar fitur. Metode CFS menghasilkan subset dari perhitungan heuristik berdasarkan korelasi. Metode ini merupakan metode yang sederhana, cepat dan dapat menangani dataset yang kecil walau memiliki banyak atribut yang tidak relevan.

Berdasarkan penelitian yang dilakukan oleh Sarini Vita Dewi dalam tesisnya yang berjudul "Analisa Performa Klasifikasi untuk diagnosis penyakit Parkinson" menunjukkan bahwa penggunaan CFS sebagai metode seleksi fitur cukup efektif, yaitu dengan tingkat akurasi pemilihan fitur yang sesuai menjadi lebih tinggi.

Hasil akurasi dari subset CFS menggunakan metode klasifikasi Bagging J48 sebesar 91% diperoleh dari pengklasifikasian instance TP sebanyak 139 instance, FP sebanyak 11 instance, FN sebanyak 6 instance dan TN sebanyak 36 instance. Ini menunjukkan bahwa dari 195 instance, ada 139 instance yang berhasil diklasifikasi dan positif, dan ada 36 instance yang merupakan negatif, sedangkan sisanya merupakan pengklasifikasian yang keliru yaitu sebanyak 17 instance.

- Running time yang diperoleh dari metode seleksi fitur CFS adalah 0,01 detik, dan ini merupakan running time paling rendah yang diperoleh dari penelitian ini.
- Dari hasil analisis di atas dapat dinyatakan bahwa subset yang dihasilkan dari metode seleksi fitur CFS menghasilkan nilai tertinggi dari 4 parameter evaluasi yang digunakan. Ini menunjukkan bahwa subset dengan 9 atribut yang dihasilkan oleh metode CFS dengan label P1, P2, P3, P4, P6, P13, P15, P16, P19, P20 adalah subset terbaik untuk menentukan klasifikasi pada dataset penyakit Parkinson.

Perbandingan hasil dari penelitian yang dilakukan oleh sarini vita dewi, dkk dapat dilihat pada Gambar 9.5.



Gambar 9.5: Perbandingan hasil akurasi menggunakan CFS, Wrapper dan tanpa seleksi fitur (Sarini Vita Dewi, Adhistya Erna Permanasari, 2014)

Gambar 9.5 di atas menunjukkan tingkat akurasi yang dilakukan dengan mengunakan metode CFS dan Wrapper. Dari kedua metode tersebut terlihat bahwa seleksi fitur menggunakan algoritma CFS menunjukkan nilai akurasinya lebih tinggi dibandingkan dengan menggunakan algoritma Wrapper.

Bab 10

Text Mining: Twitter Analysis

10.1 Media Sosial

Lahirnya internet di era tahun 1980-an telah membawa banyak perubahan terdapat perkembangan ilmu pengetahuan, pola bisnis, pola komunikasi, pola interaksi dan pola kehidupan bagi masyarakat yang merasakannya (Primatha, 2018). Kementrian Komunikasi dan Informasi (KOMINFO) merilis sebuah data bahwa dari 265 juta jiwa penduduk Indonesia terdapat sekitar 54% telah menggunakan teknologi internet atau sekitar 143 juta jiwa (Hutabarat, 2018). Perubahan pola komunikasi dan pola interaksi masyarakat salah satu sebabnya adalah dengan munculnya kehadiran berbagai media yang memudahkan seseorang untuk berkomunikasi dengan orang lainnya, salah satunya kehadiran media sosial. Andreas Kaplan dan Michael Haenlein mengartikan media sosial dengan definisi sebagai berikut :"A group of internet-based application that build on the ideological and technological foundation of web 2.0 and that allow the creation and exchange of user-generated content" (Ravindran & Vikram Garg, 2015).

Pengertian lainnya, istilah media sosial tersusun atas dua kata yaitu "Media" dan "Sosial". "Media" diartikan sebagai alat komunikasi. Sedangkan kata "sosial" diartikan sebagai kenyataan sosial bahwa setiap individu melakukan aksi yang memberikan kontribusi kepada masyarakat. Pernyataan ini menegaskan bahwa pada kenyataannya, media dan semua perangkat lunak merupakan "sosial" atau dalam makna bahwa keduanya merupakan produk dari proses sosial (Mulawarman, 2017).

Media sosial dengan berbasis internet mencakup banyak platform yang memfasilitasi berbagai keperluan manusia, seperti (Ravindran & Vikram Garg, 2015) :

1. *networking*, contohnya facebook, Linkedln, dan lainnya
2. *Micro Blogging*, contohnya Twitter, Tumblr, dan lainnya
3. *Photo Sharing*, contohnya Instagram, Flicker dan lainnya
4. *Video Sharing*, contohnya Youtube, Vimeo dan lainnya
5. *Stack exchanging*, contohnya stack Overflowm Github, dan lainnya
6. *Instant messaging*, contohnya Whatsapp, telegram dan lainnya.

Di era sebelum terdapat media sosial, media yang digunakan seseorang ketika ingin memberi tanggapan tentang sesuatu sangat terbatas seperti melalui korespodensi surat menyurat, atau melalui media cetak contohnya koran, tabloid atau media elektronik contohnya TV dan atau radio. Berbeda jauh dengan adanya media sosial, di mana penggunaannya dapat lebih fleksibel digunakan untuk memberikan tanggapan terhadap sesuatu kapan dan di manapun menggunakan gawai selama terkoneksi ke internet, disamping itu jenis data yang digunakanpun lebih beragam mulai dari data teks, suara, gambar ataupun video. Kita mungkin sudah pernah mendengar dengan adanya istilah big data yang mempunyai 3 ciri karakteristik yaitu mulai dari *volume*, *variety*, dan *velocity*, dan ketiga karakteristik tersebut setidaknya dapat terlihat dari data-data yang dihasilkan melalui media sosial. Media sosial saat ini mempunyai peran yang semakin meluas tidak hanya sekedar sebagai sarana komunikasi, namun juga dapat menjadi media di mana seseorang dapat memberikan ekspresi tentang apa yang menarik perhatian seseorang atau sebaliknya sharing sesuatu yang dialami atau dilihat, sharing sesuatu yang tidak disukai, atau testimoni dari suatu produk apakah produk tersebut baik atau sebaliknya.

10.2 Text Mining

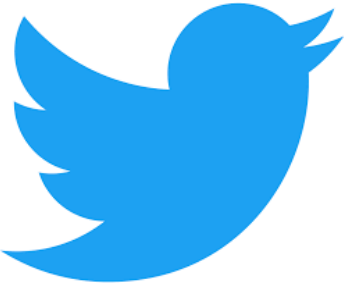
Salah satu jenis data yang banyak digali karena keberlimpahannya salah satu nya adalah data text, sehingga sering terdengar istilah text mining. Text mining dapat diartikan sebagai proses ekstraksi makna dari data teks yang tidak terstruktur yang diperoleh dari media sosial (Ravindran & Vikram Garg, 2015). Dalam referensi lainnya, text mining didefinisikan sebagai proses penggalian data berbasis text yang berkaitan dengan informasi apa yang akan kita cari serta mencari hubungan atau korelasi yang menarik (Batter, et al., 2017), sumber data yang biasa diambil untuk dilakukan text mining biasanya berasal dari dokumen

berbentuk text, word, pdf atau format lainnya, email, online chat, online news, forum-forum online, blogs, dan tentunya media sosial.

10. 3 Twitter

Dalam buku ini kita akan mencoba bagaimana cara untuk melakukan penggalian data (text mining) dari media sosial dengan sampel dari media sosial twitter. Bila digali sejarahnya, pada tahun 2006, Twitter didirikan oleh Evan Williams, Jack Dorsey, Cristopher Stone dan Noah Glass. Twitter sebagai situs microbloging dapat menuliskan pesan dengan panjang 140 karakter yang digunakan netizen untuk mengekspresikan opininya tentang apapun dan di akses dimanapun selama terkoneksi melalui internet.

Brian J Dixion menyatakan bahwa "twitter is microblogging messaging service that limits you to 140 characters per message service, including spaces and punctuation, to you update content" (Dixon, 2012)



Gambar 10.1: Logo Twitter

Perkembangan penggunaan twitter sebagai sebuah media sosial terbilang sangat cepat, hasil survey yang dilakukan oleh IDN Research Institut melansir hasil penelitiannya tentang "Indonesia Millenial Report" memberikan informasi bahwa media sosial yang paling banyak digunakan oleh masyarakat Indonesia berturut-turut di antaranya adalah facebook, Instagram, twitter, dan Path (Institut, 2019). Twitter dalam data tersebut menduduki peringkat ketiga dari media sosial yang paling banyak di akses oleh netizen. Pada tahun 2014 pengguna twitter mencapai 500 juta orang dengan pengguna aktif sebanyak 271 juta orang. Dan sekitar 23% penggunanya adalah remaja. Pada januari 2018 dan

januari 2019 twitter menjadi media sosial ke empat yang paling banyak di gunakan oleh netizen.

10.4 R Programing

Banyak tools yang dapat digunakan untuk melakukan pengambilan data dari media sosial twitter, dalam buku ini, tools yang digunakan adalah dengan menggunakan bahasa pemrograman R. R adalah bahasa dan *environment* untuk komputasi statistik dan grafik yang pada mulanya dibuat oleh Ross Ihaka dan Robert Gentleman di University of Auckland, New Zealand. R tersedia sebagai free software dibawah lisensi Free Software Foundation's GNU General Public License (Primatha, 2018).

Gambar 10.2: Logo R

R dipakai dengan menggunakan lisensi GNU GPL yang banyak digunakan untuk menangani komputasi statistik dan memudahkan dalam penyajian grafik. *Integrated development environtment* (IDE) yang digunakan adalah Rstudio yang dapat didownload dengan gratis di https://rstudio.com/ kemudian silahkan install di perangkat Personal Computer anda.

10.5 Studi Kasus

Pada buku ini, selanjutnya akan membahas bagaimana langkah-langkah untuk melakukan pengambilan data dari media sosial twitter serta langkah-langkah bagaimana melakukan text analysis, besar kemungkinan bahwa langkah-langkah text mining yang ditunjukan dalam buku ini adalah bukan satu-satunya

cara yang dapat dilakukan, tentunya masih banyak cara-cara lainnya yang dapat digunakan.

Dalam pembahasan selanjutnya, penulis akan mengambil suatu tema yang akan dijadikan sebagai bentuk contoh kasus. Tema yang akan dijadikan sebagai contoh kasus adalah berkaitan dengan tema wabah covid19 yang terjadi akhir 2019 yang bermula di Cina atau awal 2020 yang mulai menyebar ke berbagai negara.

Langkah-langkah analisisnya dimulai dari (Zhao, 2015):

1. Extracting tweets,
2. Text cleaning,
3. Mencari dan menghitung banyaknya frekuensi setiap kata
4. Membuat frekuensi kata dalam wordcloud,
5. Melihat keterhubungan setiap kata dan
6. Melakukan analisis sentimen.

10.6 Twitter API

Pengambilan data melalui twitter dapat dilakukan setelah sebelumnya mendapatkan API dari media sosial twitter sehingga kita akan memperoleh

1. consumer_key,
2. consumer_secret_key,
3. access_token,
4. access_secret_key

yang selanjutnya akan digunakan untuk mengakses dan mengambil data dari twitter. Bagaimana cara mencari twitter API dalam buku ini tidak dibahas, silahkan pembaca dapat mencari referensi lainnya yang berkaitan dengan bagaimana cara mendapatkan twitter API.

Sekarang silahkan buka Rstudio yang telah anda install di *personal computer* anda kemudian ketikan code berikut pada window Rscript: Silahkan install beberapa package yang ada dibawah ini, kemudian panggil dalam bentuk

library seperti yang ada pada sintaks di bawah ini, yang bertujuan agar dapat mengambil data tweet dari twitter.

```
library(twitteR)
library(ROAuth)
```

Setelah kita menginstall dan memasang package yang diperlukan maka, langkah selanjutnya adalah mengambil data dari media sosial twitter, tentu ini bisa dilakukan jika kita sudah mempunyai consumer_key, consumer_secret_key, access_token, dan access_secret_key.

Jika kita sudah memilikinya maka kita bisa menuliskan sintaks berikut yang berfungsi agar kita dapat terhubung ke twitter.

```
api_key <- "mXIT1jyOyt1jru4WXByE3Uum"
api_secret <- "31PGu6lnrscfrQCRBagEYeEtsRNggw2XC4ZK8j3Wt12lVnq
um"
api_token <- "77423691-1T3JDJMQ4ZpsDCFYUqzWdNhA1lElHMisbYbw25j
O"
api_token_secret <- "Bl6xMFV9mN0pIdFbGTQdK8vVpMhbsK0nSlCa9qZPD
YAJ"
setup_twitter_oauth(api_key, api_secret, api_token, api_token_
secret)

## [1] "Using direct authentication"
```

Setelah terhubung ke twitter, maka kita dapat mengambil data berkaitan dengan tema yang sedang kita cari. Dalam contoh ini penulis menggunakan tema covid19 dengan kata kunci covid19Tracking, covid19Tracking merupakan sebuah project yang berusaha untuk mengumpulkan informasi dari 50 negara bagian AS, Distrik Columbia, dan 5 wilayah AS lainnya untuk memberikan data pengujian paling komprehensif. Lebih lanjut dapat dilihat di laman https://covidtracking.com/ Dalam latihan di bawah ini, kita mencoba untuk mengambil data sebanyak 3000 tweet, berikut sintaks yang digunakan :

```
covid19 <- userTimeline("COVID19Tracking",n=3000)
length(covid19)

## [1] 426
```

dengan perintah length(covid19) di atas, kita akan mengetahui bahwa jumlah data yang bisa diambil dari twitter adalah sebanyak 426 data.

```
#mengkonversi data yang diperoleh menjadi dataframe
covid19.df <- twListToDF(covid19)
```

dari data yang kita peroleh, kita dapat melihat secara sekilas 6 data teratas dengan sintaks berikut:

```
head(covid19.df$text)
```

```
## [1] "When this project began, we did not know we'd still be
doing this in April. Now we expect to be doing it for many m…
https://t.co/hm8aVbmadl"
## [2] "OK, the new site is up:\nhttps://t.co/PZrmH4bl5Y\n\nHe
re's a look at the visual evolution. And there's at least this
m… https://t.co/YiS9hZBDDV"
## [3] "Huge thanks go to @beep, @tealtan, @jasonsantamaria @k
aicurry, Kevin Miller, and Andrew Schwartz. And, of course, t…
https://t.co/NtBHX8PzfR"
## [4] "We'll be launching an updated site around 12pm ET. Pre
pare for some instability for a few minutes, and then a new,…
https://t.co/55E1GBzLNr"
## [5] "It’s truly unbelievable. Our volunteers are amazing, b
ut we cannot force standardized state reporting. https://t.co/
c6cNtac91e"
## [6] "Very good story on the challenges of modeling this out
break.\n\nhttps://t.co/RLMVevhcys"
```

Kita dapat mengecek atribut dari data yang telah kita peroleh, dengan cara berikut:

```
#fungsi names() digunakan untuk melihat nama-nama atribute yan
g dimiliki oleh data yang diperoleh dari twitter
names(covid19.df)
```

```
##  [1] "text"          "favorited"     "favoriteCount" "reply
ToSN"
##  [5] "created"       "truncated"     "replyToSID"    "id"
##  [9] "replyToUID"    "statusSource"  "screenName"    "retwe
etCount"
## [13] "isRetweet"     "retweeted"     "longitude"     "latit
ude"
```

agar data yang kita peroleh dapat diambil kembali sewaktu-waktu, maka kita dapat menyimpan data yang kita peroleh ke dalam lokal drive dengan nama covid19Tracking.csv,

```
write.csv(covid19.df, "D:/covid19/covid19Tracking.csv")
```

untuk mengecek rangkuman data covid19.df terlihat ada 426 objek yang diperoleh, dengan 16 variabel, seperti yang terlihat dibawah ini.

```
str(covid19.df)

## 'data.frame':    426 obs. of  16 variables:...
#kita bisa melihat data dalam bentuk data frame dengan perinta
h dibawah ini
View(covid19.df)
```

kitapun dapat mencoba memanggil dan menampilkan data ke 174 dengan beberapa atribut terpilih. Dengan sintaks berikut:

```
covid19.df [174, c("id", "created", "screenName", "replyToSN",
"favoriteCount", "retweetCount", "longitude",  "latitude","tex
t")]

writeLines(strwrap(covid19.df$text[174], 60))

## @nbashaw The California data is all over the place. They do
## have a lot of pending tests (~12k) —not included in our…
## https://t.co/OfUEOa0uGK
```

selanjutnya kita akan mencoba untuk melakukan pra processing atau membersihkan data yang telah kita ambil dari twitter dengan menggunakan package tm, silahkan install package tersebut dan panggil dalam bentuk library(tm)

```
library(tm)

## Loading required package: NLP
```

langkah berikutnya adalah kita akan membersihkan data, membuat corpus dan menjadikan text kedalam vector karakter

```
corpus_ku <- Corpus(VectorSource(covid19.df$text))
```

mengubah data menjadi *lowercase*

```
corpus_ku <- tm_map(corpus_ku, content_transformer(tolower))

## Warning in tm_map.SimpleCorpus(corpus_ku, content_transform
er(tolower)):
## transformation drops documents
```

menghapus URLs

```
removeURL <- function(x) gsub ("http[^[:space:]]*", "", x)
corpus_ku <- tm_map(corpus_ku, content_transformer(removeURL))

## Warning in tm_map.SimpleCorpus(corpus_ku, content_transform
er(removeURL)):
## transformation drops documents
```

Menghapus tanda baca atau spasi

```
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "
", x)
corpus_ku <- tm_map(corpus_ku, content_transformer(removeNumPu
nct))

## Warning in tm_map.SimpleCorpus(corpus_ku, content_transform
er(removeNumPunct)):
## transformation drops documents

corpus_ku <- tm_map(corpus_ku, stripWhitespace)

## Warning in tm_map.SimpleCorpus(corpus_ku, stripWhitespace):
transformation drops
## documents

# menghilangkan stopwords secara manual
myStopwords <- c(setdiff(stopwords('english'),c("covid19", "co
rona")),"be", "for", "sorri", "t", "way", "to", "im", "i", "on
", "my", "you", "it", "that", "out", "whitwyatt", "also","weve
","can", "now", "alexismadrig")
corpus_ku <- tm_map(corpus_ku, removeWords, myStopwords)

## Warning in tm_map.SimpleCorpus(corpus_ku, removeWords, mySt
opwords):
## transformation drops documents
```

```
#menghilangkan number
corpus_ku <- tm_map(corpus_ku, removeNumbers)

## Warning in tm_map.SimpleCorpus(corpus_ku, removeNumbers): t
ransformation drops
## documents
```

pada saat kita menghilangkan *stopword* atau kata-kata yang sering berulang tapi tidak mempunyai makna apa-apa, kita bisa menambahkannya kata-kata yang lain pada sintaks diatas,

selanjutnya kita akan mengcopy data yang telah dibersihkan kedalam object lainnya, hal ini penting dilakukan sebagai data cadangan

```
corpus_ku2 <- corpus_ku
# stem words
corpus_ku <- tm_map(corpus_ku, stemDocument)

## Warning in tm_map.SimpleCorpus(corpus_ku, stemDocument): tr
ansformation drops
## documents
```

kita bisa menguji coba memanggil data setelah proses pembersihan data

```
writeLines(strwrap(corpus_ku[[174]]$content, 60))

## nbashaw california data place lot pend test k includ
```

hasil di atas terlihat sebuah tweet yang bersih dari URL, *number*, terdiri dari *lower case*,

berikutnya kita akan Membuat Term document matrix

```
tdm <- TermDocumentMatrix(corpus_ku,
  control = list(wordLengths = c(1, Inf)))
tdm

## <<TermDocumentMatrix (terms: 1366, documents: 426)>>
## Non-/sparse entries: 3853/578063
## Sparsity           : 99%
## Maximal term length: 15
## Weighting          : term frequency (tf)
```

dalam analisis text dari sekumpulan data kita dapat mencari kata yang paling banyak muncul dari data yang kita miliki, dengan cara berikut,

```
(freq.terms <- findFreqTerms(tdm, lowfreq = 20))
```

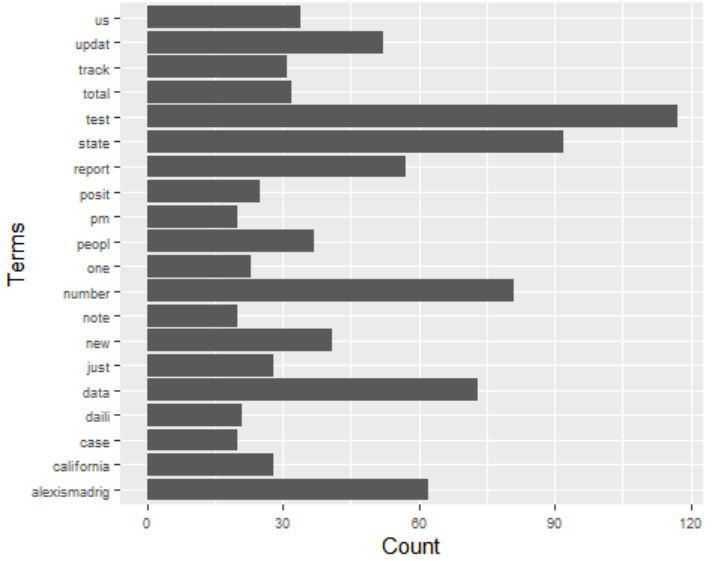
```
##  [1] "new"          "pm"           "updat"        "report"
"state"
##  [6] "california"   "track"        "one"          "peopl"
"test"
## [11] "us"           "daili"        "alexismadrig" "just"
"number"
## [16] "total"        "case"         "data"         "posit"
"note"
```

hasil di atas menunjukan kata-kata yang banyak muncul dari data yang kita miliki, selanjutnya kita akan coba menghitungnya lalu kemudian akan dibuatkan kedalam sebuah grafik,

```
term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 20)
df <- data.frame(term = names(term.freq), freq = term.freq)

library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

```
ggplot(df, aes(x=term, y=freq)) + geom_bar(stat="identity") +
xlab("Terms") + ylab("Count") + coord_flip() +
theme(axis.text=element_text(size=7))
```



kemudian kita pun dapat menampilkan plot dalam bentuk wordcloud

```
m <- as.matrix(tdm)
# menghitung frekuensi kata dan mengurutkannya berdasark
an frekuensi tersebut
word.freq <- sort(rowSums(m), decreasing = T)

# plot word cloud
library(wordcloud)

## Loading required package: RColorBrewer

pal <- brewer.pal(9, "BuGn")[-(1:4)]
wordcloud(words = names(word.freq), freq = word.freq, mi
n.freq = 3,
random.order = F, colors = pal)
```

Gambar 10.1: Wordcloud

jika kita gali lebih dalam kita bisa mencari, kata apa yang selalu mendampingi suatu kata dengan menerapkan aturan asosiasi

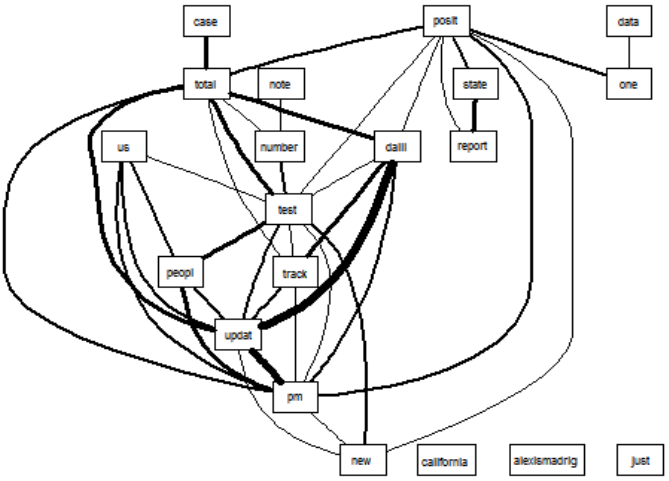
```
findAssocs(tdm, "test", 0.2)

## $test
##    ramp   peopl complet   total    penc      vp
##    0.28    0.27    0.27    0.27    0.23    0.23
```

hasil di atas menunjukan ada beberapa kata yang banyak mendampingi kata ‘test’.

selanjutnya kita bisa menggali lebih lanjut dengan melihat jaringan yang terbentuk dari kata-kata yang sering muncul dengan bantuan package graphTweets dan Rgraphviz

```
library(graphTweets)

## Warning: package 'graphTweets' was built under R version 3.
6.3
```

```
## help('graphTweets') for examples

if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("Rgraphviz")

library(Rgraphviz)

plot(tdm, term = freq.terms, corThreshold = 0.1, weighting = T
)
```



Gambar 10.2: Jaringan Kata Yang Banyak Muncul

10. 7 Analisis Sentimen

Apa yang ditulis dalam media sosial bisa saja berupa tulisan yang sangat ekspresif, sehingga kita dapat melihat bagaimana sebuah perasaan di ekspresikan, apakah ekspresi tersebut negatif atau juga ekspresi yang bersifat positif, jika dikaitkan dalam dunia bisnis, ekspresi seseorang dalam media sosial dapat membantu perusahaan sebagai bahan analisis untuk memahami pelanggannya, seperti bagaimana respon seseorang terhadap suatu produk yang

baru dikeluarkan, selain itu dari media sosial juga dapat diketahui bagaimana posisi kompetitor perusahaan (Sagar, 2018).

Pada kasus analisis sentimen terdapat suatu metode yang bernama lexicon analysis atau lexicon based. metode berbasis lexicon umumnya menggunakan kamus yang berisi kata-kata opini untuk menentukan suatu sentimen atau polaritas seperti positif atau negatif dari suatu data teks (dikutip dari imam syafei dan hendri mufti). metode text mining yang berbasis lexicon mempunyai beberapa model, di antaranya :

NRC LEXICON

NRC emotion lexicon adalah susunan kata berbahasa Inggris yang dikelompokan dalam delapan emosi dasar seperti angry (marah), fear (ketakutan), anticipation (antisipasi), disgust (jijik), joy (gembira), sadness (sedih), surprise (heran), trust (percaya) serta dua sentimen seperti negatif dan positif. Keterangan lengkap NRC Emotion Lexicon biasanya ditulis NRC Word-Emotion Association Lexicon atau biasa di tulis EmoLex. Jumlah kata dalam lexicon NRC terdapat 13.901 kata dibuat dibawah pengawasan dari National Research Council Canada. NRC Lexicon dapat diakses melalui laman https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm. Secara lengkap jumlah setiap kata dari masing-masing kategori adalah sebagai berikut:

Tabel 10.1: Jumlah Kata dalam NRC Lexicon

Kategori	Jumlah Kata
Anger	1247
Anticipation	839
Disgust	1058
Fear	1476
Joy	689
Sadness	1191
Surprise	534
Trust	1231

Positive	2312
Negative	3324

BING LEXICON

Bing lexicon di buat oleh Minqing Hu dan Bing Liu sebagai kamus untuk menentukan opini. Jumlah kata yang terdapat dalam Bing Lexicon adalah 6789 kata. Semua kata-kata tersebut dapat ditemui dalam sumber utamanya di

https://www.cs.edu/~lib/FBS/sentiment-analysis.html#lexicon. Jumlah kata setiap kategori ada pada tabel berikut :

Tabel 10.2: Jumlah Kata dalam Bing Lexicon

Kategori	Jumlah Kata
Positive	2006
Negative	4783

untuk melakukan analisis lexicon kita memerlukan sebuah package yang bernama syuzet. sehingga kita perlu install kemudian memanggil library-nya.

```
library(syuzhet)
```

dengan menggunakan data yang sebelumnya telah kita peroleh, kita dapat melanjutkannya sebagai bahan untuk di analisis,

```
covid19.df2 <-  gsub("http.*","",covid19.df$text)
covid19.df2 <- gsub("https.*","",covid19.df2)
covid19.df2 <- gsub("#.*","",covid19.df2)
covid19.df2 <- gsub("@.*","",covid19.df2)

head(covid19.df2)

## [1] "When this project began, we did not know we'd still be
doing this in April. Now we expect to be doing it for many m…
"
## [2] "OK, the new site is up:\n"
## [3] "Huge thanks go to "
## [4] "We'll be launching an updated site around 12pm ET. Pre
pare for some instability for a few minutes, and then a new,…
"
```

```
## [5] "It’s truly unbelievable. Our volunteers are amazing, b
ut we cannot force standardized state reporting. "
## [6] "Very good story on the challenges of modeling this out
break.\n\n"
```

agar kita dapat menggunakan NRC lexicon, maka data yang digunakan harus dalam bentuk vektor, maka kita perlu membuat konversinya

```
word.df <- as.vector(covid19.df2)

emotion.df <- get_nrc_sentiment(word.df)

emotion.df2 <- cbind(covid19.df2, emotion.df)
```

kita dapat melihat data teratas dari penerapan NRC lexicon

```
head(emotion.df2)

##
covid19.df2
## 1 When this project began, we did not know we'd still be do
ing this in April. Now we expect to be doing it for many m…
## 2
OK, the new site is up:\n
## 3
Huge thanks go to
## 4  We'll be launching an updated site around 12pm ET. Prepa
re for some instability for a few minutes, and then a new,…
## 5              It’s truly unbelievable. Our volunteers are
amazing, but we cannot force standardized state reporting.
## 6                                                        Very
good story on the challenges of modeling this outbreak.\n\n
##   anger anticipation disgust fear joy sadness surprise trus
t negative positive
## 1     0            1       0    0   0       0        1
1        0        1
## 2     0            0       0    0   0       0        0
0        0        0
## 3     0            0       0    0   0       0        0
0        0        0
## 4     0            1       1    1   0       0        0
0        1        1
```

```
## 5     1           0       0    1   0       0        0
1        2        0
## 6     0           1       0    0   1       0        1
1        0        1
```

proses ekstraksi nilai sentimen dapat dilakukan dengan langkah-langkah dibawah ini,

```
sent.value <- get_sentiment(word.df)
most.positive <- word.df[sent.value == max(sent.value)]
```

kita dapat mencari tweet mana yang mempunyai nilai paling positif,

```
most.positive

## [1] "We want to thank some states for excellent new dashboa
rds that now include all negative tests, including those from…
"
```

dan begitupun dengan nilai sentimen tweet yang paling negatif

```
most.negative <- word.df[sent.value <= min(sent.value)]
most.negative

## [1] "Note: we cannot capture all negative or pending tests
because of the way that states report. Consider our number a…
"
```

jika kita ingin melihat nilai sentimen dari keseluruhan data yang dimiliki, kita dapat mencarinya dengan langkah seperti yang ada dibawah

```
sent.value

##   [1]  0.40  0.80  0.60  0.40 -0.85 -0.05  0.00 -0.10  0.00
0.70 -0.40  0
```

langkah dibawah adalah untuk mengumpulkan seluruh tweet positif, negatif dan netral dalam satu variabel yang bernama positive.tweets, negative.tweets, netral.tweets

```
positive.tweets <- word.df[sent.value > 0]
head(positive.tweets)

## [1] "When this project began, we did not know we'd still be
doing this in April. Now we expect to be doing it for many m…
```

```
"
## [2] "OK, the new site is up:\n"
## [3] "Huge thanks go to "
## [4] "We'll be launching an updated site around 12pm ET. Pre
pare for some instability for a few minutes, and then a new,…
"
## [5] "Also: testing doesn't only help people track the outbr
eak. It also matters for patient care, and one of the more tr…
"
## [6] "One reason is supply constraints, which are hard to pi
n down, but definitely a real factor. Also, many labs are tak…
"

negative.tweets <- word.df[sent.value < 0]
head(negative.tweets)

## [1] "It’s truly unbelievable. Our volunteers are amazing, b
ut we cannot force standardized state reporting. "
## [2] "Very good story on the challenges of modeling this out
break.\n\n"
## [3] "We finally have some answers on what's been going on i
n California. The short version: as demand exploded in the we…
"
## [4] "There are also clear *demand* constraints as testing c
riteria remain quite strict. We are still getting reports fro…
"
## [5] "Our daily update is published. We’ve tracked a total o
f 831,351 tests, up 95,647 from yesterday, the lowest number…
"
## [6] "While we finalize, let me explain the problem. Our num
bers lock at ~4pm ET each day. California's updates tend to f…
"

neutral.tweets <- word.df[sent.value == 0]
head(neutral.tweets)

## [1] "In "
## [2] "If you've been following this account, you know we've
been tracking The California Situation. The state is completi…
"
## [3] "The US has now completed tests on over 1 million peopl
e: 1,048,971 to be exact. \n\nIt's a milestone. \n\nBut/and ou
```

```
r d… "
## [4] ""
## [5] ""
## [6] ""
```

Langkah terakhir adalah mencoba untuk melhat berapa nilai keseluruhan dari sentimen yang kita miliki, dengan cara seperi yang ada di bawah

```
category_senti <- ifelse(sent.value < 0, "Negative", ifelse(se
nt.value > 0, "Positive", "Neutral"))

head(category_senti)

## [1] "Positive" "Positive" "Positive" "Positive" "Negative"
"Negative"

table(category_senti)

## category_senti
## Negative  Neutral Positive
##       33      274      119
```

hasilnya dari 422 tweet yang kita miliki, maka jika diurai terdapat 32 bernilai negatif, 276 neutral dan 114 bernilai positif.

Daftar Pustaka

Aco, A. and Endang, A. H. (2017) 'Analisis Bisnis E-Commerce pada Mahasiswa Universitas Islam Negeri Alauddin Makassar', Jurnal Insypro.

Aditya, Marisa, F. and Purnomo, D. (2016) 'Penerapan Algoritma A Priori Terhadap Data Penjualan di Toko Gudang BM', JOINTECS (Journal of Information Technology and Computer Science), 1(1), pp. 1–5. doi: 10.31328/jointecs.v1i1.408.

Alpkan, L. ütfiha., Şanal, M. and Ayden, Y. ükse. (2012) 'Market Orientation, Ambidexterity and Performance Outcomes', Procedia - Social and Behavioral Sciences, 41, pp. 461–468. doi: 10.1016/j.sbspro.2012.04.056.

Alter, Steven. 2002. Information System: Foundation of E-Business. Prentice Hall, New Jersey.

Amroh (2010) Aplikasi Penentuan Status Kesuburan Tanah Berdasarkan Kandungan Sifat Kimia Tanah Menggunakan Metode K Nearest Neighbor (KNN).

Aprilla, D. et al. (2013) Belajar Data Mining Dengan Rapid Minner.

Aqra, I. et al. (2018) A novel association rule mining approach using TID intermediate itemset, PLoS ONE. doi: 10.1371/journal.pone.0179703.

Aribowo, A. S. (2015) 'Analisa Asosiatif Data Mining Untuk Mengetahui Pola Kecelakaan Lalu Lintas', Telematika, 8(2), pp. 2–7. doi: 10.31315/telematika.v8i2.458.

Arija Fachriyan, H. and Wijaya, I. P. E. (2018) 'Aplikasi Model E-Marketplace Dalam E-Agribusiness', Journal of Chemical Information and Modeling, 14(1), pp. 1–13. doi: 10.1017/CBO9781107415324.004.

Aspian Nur, A. (2019) 'Perhitungan Regresi Berganda (Multiple Regression) secara manual', https://www.researchgate.net/publication/335234153, (August). doi: 10.13140/RG.2.2.18009.47205.

Asriningtias, Y. et al. (2014) 'Aplikasi Data Mining Untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa', 8(1), pp. 837–848. doi: 10.12928/jifo.v8i1.a2082.

Bagus, P. et al. (2018) 'Analysis of A Priori Algorithm on Sales Transactions to Arrange Placement of Goods on Minimarket', International Journal of Engineering and Emerging Technology, 3(1), pp. 13–17.

Batter, M., Pradeepta, M., Nathan, D. & Richard, H., (2017). R: Mining Spatial, Text, Web, and Social Media Data. Birmingham: Packt.

Beritasatu (2016) 2020, 50 Miliar Perangkat Terhubung Internet - BeritaSatu.com. Available at: https://www.beritasatu.com/iptek/398265-2020-50-miliar-perangkat-terhubung-internet (Accessed: 27 April 2020).

Chang, C. I. and Ren, H. (2000) 'An experiment-based quantitative and comparative analysis of target detection and image classification algorithms for hyperspectral imagery', IEEE Transactions on Geoscience and Remote Sensing, 38(2 II), pp. 1044–1063. doi: 10.1109/36.841984.

Cynthia, E. P. and Ismanto, E. (2018) 'Metode Decision Tree Algoritma C.45 Dalam Mengklasifikasi Data Penjualan', Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASI), (3) Juli(July), pp. 1–13. doi: 10.30645/jurasik.v3i0.60.

Dixon, B. J., (2012). Social Media For School Leaders. San Fracisco: John Wiley & Sons.

Estivill-Castro, V. (2002) 'Why so many clustering algorithms- A position paper', SIGKDD Explorations.

Fadli, A. (2003) 'Konsep Data Minning', Konsep Data Mining, pp. 1–9. Available at: http://ilmukomputer.com.

Fayyad, U. and Stolorz, P. (1997) 'Data mining and KDD: Promise and challenges', Future Generation Computer Systems, 13(2–3), pp. 99–115.

Gunawan, R. (2019) 'Implementasi Data Mining Menggunakan Regresi Linier Berganda dalam Memprediksi Jumlah Nasabah Kredit Macet Pada BPR Tanjung Morawa', Sains dan Komputer (SAINTIKOM), 18(1), pp. 87–91.

Hadiri, A. D. (2016) 'PENERAPAN PARTICLE SWARM OPTIMIZATION UNTUK SELEKSI ATIRBUT PADA METODE DECISION TREE C

4 . 5 UNTUK PERSETUJUAN ABSTRACT : Bad credit is one of the credit risk faced by the financial and banking industry . Improved accuracy of credit ratings can be done b', in KNIT-2 Nusa Mandiri.

Hall, M. A. (1999) "Correlation-based Feature Selection for Machine Learning," (April).

Han, J. and Kamber, M. (2006) Data Mining : Concepts and Techniques Second Edition. San Francisco: Elsevier.

Han, J., Kamber, M. and Pei, J. (2012) Data Mining (Third Edition), Data Mining (Third Edition). doi: https://doi.org/10.1016/B978-0-12-381479-1.00016-2.

Han, J., Kamber, M. and Pei, J. (2012) Data mining: concepts and techniques.

Hartama, D. (2011) Model Aturan Keterhubungan Data Mahasiswa Menggunakan Algoritma C 4.5 Untuk Meningkatkan Indeks Prestasi. Universitas Sumatera Utara.

Hasibuan, A. et al. (2020) E-Business: Implementasi, Strategi dan Inovasinya. Medan: Yayasan Kita Menulis.

He, H. & Tan, Y., 2012. A two-stage genetic algorithm for automatic clustering. Neurocomputing, Volume 81, pp. 49-59.

Hemeida, A. M. et al. (2019) 'Implementation of nature-inspired optimization algorithms in some data mining tasks', Ain Shams Engineering Journal, pp. 1–10.

Hermawati, F. A. (2013) Data Mining, Andi: Yogyakarta. Jogyakarta: Penerbit: Andi.

Hofmann, M. & Klinkenberg, R., 2016. RapidMiner: Data mining use cases and business analytics applications. s.l.:CRC Press.

Hutabarat, D., (2018). Kementerian Kominfo Sebut Pengguna Internet Indonesia Capai 54 Persen, Jakarta: Kementrian Kominfo RI.

Informatikalogi (2017) Algoritma K-Nearest Neighbor (K-NN). Available at: https://informatikalogi.com/algoritma-k-nn-k-nearest-neighbor/.

Institut, I. R., (2019). Indonesia Millenial Report. s.l.:IDN Research Institut.

Jimmy Nganta Ginting (2020) Performance improvement and accuracy of artificial neural network using particle swarm optimization for breast cancer prediction.

Jin, X. et al. (2011) 'K-Medoids Clustering', in Encyclopedia of Machine Learning. doi: 10.1007/978-0-387-30164-8_426.

Kantardzic, M., 2011. Data mining: concepts, models, methods, and algorithms. s.l.:John Wiley \& Sons.

Kaufman, L. and Rousseeuw, P. J. (1987) 'Clustering by means of medoids', Statistical Data Analysis Based on the L1-Norm and Related Methods.

Kaufman, L. and Rousseeuw, P. J. (1990) Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics), Eepe.Ethz.Ch. doi: 10.1007/s13398-014-0173-7.2.

Khotimah, T. (2014) 'Pengelompokan Surat dalam Al Qur'an menggunakan Algortima K-Means', Jurnal Simetris, 5(1), pp. 83–88.

Kohavi.Provost (1998) 'Glossary of terms. Machine Learning—Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. Machine Learning', 30, pp. 271–274. doi: https://doi.org/10.1023/A:1017181826899.

Kurniawan, M. Y. and Rosadi, M. E. (2017) 'Optimasi Decision Tree Menggunakan Particle Swarm Optimization Pada Data Siswa Putus Sekolah', Jtiulm, 2(1), pp. 15–22.

Lamiaa, Fattouh Ibrahim ; Manal, H. A. H. (2012) 'Using Modified Partitioning Around Medoids Clustering Technique in Mobile Network Planning', International Journal of Computer Science Issues.

Larose, D. T. (2005) An Introduction to Data Mining The CRISP-DM.

Larose, D. T. (2005) Discovering Knowledge in Data: An Introduction to Data Mining: Second Edition. New Jersey: John Wiley & Sons.

Mastel (2018) Tren dan Perkembangan IoT di Tahun 2018 | Website Masyarakat Telematika Indonesia. Available at: https://mastel.id/tren-dan-perkembangan-iot-di-tahun-2018/.

Mulawarman, A. D. N., (2017). Perilaku Penggunaan Media Sosial Beserta Implikasinya Ditinjau dari prespektif Psikologi Sosial Terapan. vol. 25, no. 1, pp. 36 - 44, 2017 penyunt. s.l.:Buletin Psikologi.

Murnawan, Sinaga, A. and Nughraha, U. (2018) 'Implementation of A Priori Algorithm for determining purchase patterns in one transaction', International Journal of Engineering and Technology(UAE), 7(4), pp. 204–207. doi: 10.14419/ijet.v7i4.33.23560.

Mustafa, M. S., Ramadhan, M. R. and Thenata, A. P. (2018) 'Implementasi Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier', Creative Information Technology Journal, 4(2), p. 151. doi: 10.24076/citec.2017v4i2.106.

Ng, R. T. and Han, J. (1994) 'Efficient and Effective Clustering Methods for Spatial Data Mining', Proceedings of the 20th International Conference on Very Large Data Bases.

Ng, R. T. and Han, J. (2002) 'CLARANS: A method for clustering objects for spatial data mining', IEEE Transactions on Knowledge and Data Engineering. doi: 10.1109/TKDE.2002.1033770.

Nikulin, V., 2008. Classification of imbalanced data with random sets and mean-variance filtering. International Journal of Data Warehousing and Mining (IJDWM), 4(2), pp. 63-78.

Ningsih, S. R. et al. (2019) 'Analisis K-Medoids Dalam Pengelompokkan Penduduk Buta Huruf Menurut Provinsi', Prosiding Seminar Nasional Riset Information Science (SENARIS). doi: 10.30645/senaris.v1i0.78.

Nofriansyah, D. (2015) Modul: Data Mining, Algoritma Data Mining Dan Pengujian. Available at: http://sajeegm301.blogspot.com/2015/11/data-mining.html.

Nofriansyah, D. and Nurcahyo, G. W. (2009) Algoritma Data Mining dan Pengujian.

Nofriansyah, D. and Nurcahyo, G. W. (2015) Algoritma Data Mining Dan Pengujian. Yogyakarta: Deepublish.

Nugroho, Y. S. (2015) "Klasifikasi dan klastering mahasiswa informatika universitas muhammadiyah surakarta," University Research Colloquium 2015, (February), hal. 89–98. doi: 10.13140/RG.2.1.3783.4002.

Nurhachita, N. & Negara, E. S., 2020. A Comparison Between Naïve Bayes and The K-Means Clustering Algorithm for The Application of Data Mining

on The Admission of New Students. Jurnal Intelektualita: Keislaman, Sosial dan Sains, 9(1), pp. 51-62.

Nurjoko and Kurniawan, H. (2016) 'Aplikasi Data Mining Untuk Memprediksi Tingkat Kelulusan Mahasiswa Menggunakan Algoritma A Priori di IBI Darmajaya, Bandar Lampung', Jurnal TIM Darmajaya, 02(01), pp. 79–93.

Olson, D. L. & Delen, D., 2008. Advanced data mining techniques. s.l.:Springer Science \& Business Media.

Pahlevi, O., Sugandi, A. and Sintawati, I. D. (2018) 'Penerapan Algoritma A Priori Dalam Pengendalian Kualitas Produk', Sinkron, 3(1), pp. 272–278.

Panggabean, D. S. O., Buulolo, E. and Silalahi, N. (2020) 'Penerapan Data Mining Untuk Memprediksi Pemesanan Bibit Pohon Dengan Regresi Linear Berganda', JURIKOM (Jurnal Riset Komputer), 7(1), p. 56. doi: 10.30865/jurikom.v7i1.1947.

Panjaitan, S. et al. (2019) 'Implementation of A Priori Algorithm for Analysis of Consumer Purchase Patterns', Journal of Physics: Conference Series, 1255(1), pp. 1–8. doi: 10.1088/1742-6596/1255/1/012057.

Parlina, I. et al. (2018) 'Memanfaatkan Algoritma K-Means dalam Menentukan Pegawai yang Layak Mengikuti Asessment Center untuk Clustering Program SDP', CESS (Journal of Computer Engineering System and Science), 3(1), pp. 87–93.

Perkulihan, M. et al. (2018) 'Oleh BAMBANG HERMANTO, M.Cs 19790912 2008121 002'.

Pramesti, D. F. et al. (2017) 'Implementasi Metode K-Medoids Clustering Untuk Pengelompokan Data Potensi Kebakaran Hutan/Lahan Berdasarkan Persebaran Titik Panas (Hotspot)', Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya. doi: 10.1109/EUMC.2008.4751704.

Prasetyo, E. (2012) Data Mining: Konsep dan Aplikasi menggunakan Matlab. Yogyakarta: Andi Offset.

Primartha, R. (2018) Belajar Machine learning Teori dan Praktik. Bandung: Informatika Bandung.

Primatha, R., (2018). Belajar Machine Learning Teori dan Praktik. Bandung: Penerbit Informatika.

Purnia, D. S. and Warnilah, A. I. (2017) 'Implementasi Data Mining Pada Penjualan Sepatu Dengan Menggunakan Algoritma A Priori', IJCIT (Indonesian Journal on Computer and Information Technology), 2(2), pp. 31–39.

Ramdhani, L. S. (2016) 'Penerapan Particle Swarm Optimization (Pso) Untuk Seleksi Atribut Dalam Meningkatkan Akurasi Prediksi Diagnosis Penyakit Hepatitis Dengan Metode Algoritma C4 . 5', Swabumi, IV(1), pp. 1–15.

Ravindran, S. K. & Vikram Garg, (2015. Mastering Social Media Mining With R. s.l.:PACKT Publishing.

Reilly, E. D. / (2004) Concise encyclopedia of computer science. Hoboken, New Jersey: Wiley.

Riszky, A. R. and Sadikin, M. (2019) 'Data Mining Menggunakan Algoritma A Priori untuk Rekomendasi Produk bagi Pelanggan', Jurnal Teknologi dan Sistem Komputer, 7(3), p. 103. doi: 10.14710/jtsiskom.7.3.2019.103-108.

Rokach, L. and Maimon, O. (2012) Data Mining with Decision Trees Theory and Applications 2nd edition.

Ryu, D. & Baik, J., 2016. Effective multi-objective naive Bayes learning for cross-project defect prediction. Applied Soft Computing, 49(Elsevier), pp. 1062-1077.

Sagar, C., (2018). Twitter Sentiment Analysis using R. s.l.:s.n.

Sahu, H., Shrma, S. and Gondhalakar, S. (2008) 'A Brief Overview on Data Mining Survey', Ijctee, 1(3), pp. 114–121.

Salmiah, S. et al. (2020) Online Marketing. Medan: Yayasan Kita Menulis.

Santoso, F. et al. (2018) 'Algoritma C4 . 5 Dengan Particle Swarm Optimization Untuk Klasifikasi Lama Menghafal Al-Quran Pada Santri', Jurnal Teknologi Informasi, 14(2), pp. 92–103.

Saputra, D. H. et al. (2020) Digital Marketing: Komunikasi Bisnis Menjadi Lebih Mudah. Medan: Yayasan Kita Menulis.

Sarini Vita Dewi, Adhistya Erna Permanasari, H. A. N. (2014) Analisis Performa Klasifikasi untuk Diagnosis Penyakit Parkinson.

Schubert, E. and Rousseeuw, P. J. (2019) 'Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms', in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). doi: 10.1007/978-3-030-32047-8_16.

Sihombing, E. G. (2017) 'Klasifikasi Data Mining Pada Rumah Tangga Menurut Provinsi Dan Status Kepemilikan Rumah Kontrak / Sewa Menggunakan K-Means Clustering Method', CESS (Journal of Computer Engineering System and Science), 2(2), pp. 74–82.

Simarmata, J. (2006). Pengenalan Teknologi Komputer dan Informasi. Yogyakarta: Andi.

Statistica (2016) Jumlah Perangkat yang Saling Terhubung oleh Internet of Things (IoT) di Seluruh Dunia dari Tahun 2015-2025. Available at: https://teknologi.id/tekno/jumlah-perangkat-yang-saling-terhubung-oleh-internet-of-things-iot-di-seluruh-dunia-dari-tahun-2015-2025/.

Sudirman, S., Windarto, A. P. and Wanto, A. (2018) 'Data Mining Tools | RapidMiner : K-Means Method on Clustering of Rice Crops by Province as Efforts to Stabilize Food Crops In Indonesia', IOP Conference Series: Materials Science and Engineering, 420(012089), pp. 1–8.

Sulistiyanto (2018) 'Penerapan C4 . 5 Berbasis Particle Swarm Optimization (PSO) dalam Memprediksi Siswa Lolos Seleksi Perguruan Tinggi', in Seminar Nasional Teknologi Dan Bisnis, pp. 162–170.

Suntoro, J., Christanto, F. & Indriyawati, H., 2018. Software Defect Prediction Using AWEIG+ADACOST Bayesian Algorithm for Handling High Dimensional Data and Class Imbalance Problem. International Journal of Information Techonology and Business , 1(1), pp. 36-41.

Supriyadi, B. et al. (2018) 'Classification of natural disaster prone areas in Indonesia using K-means', International Journal of Grid and Distributed Computing, 11(8), pp. 87–98.

Suyanto (2019) Data Mining Untuk Klasifikasi dan Klasterisasi Data. Bandung: Informatika Bandung.

Syahputra, T., Halim, J. and Perangin-angin, K. (2018) 'Penerapan Data Mining Dalam Memprediksi Tingkat Kelulusan Uji Kompetensi (UKOM) Bidan Pada STIKes Senior Medan Dengan Menggunakan Metode Regresi Linier Berganda', jurnal Sains dan Komputer (SAINTIKOM), 17(1), pp. 1–07.

Tan, P. N., Steinbach, M. and Kumar, V. (2006) Introduction to Data Mining, Pearson New International Edition. London: Pearson Education.

Vulandari, R. T. (2017) 'Data Mining Teori dan Aplikasi Rapidminer', p. viii+124. Available at: www.infogavamedia@yahoo.com.

Wibowo, A. (2017) Klastering – MTI. Tersedia pada: https://mti.binus.ac.id/2017/11/24/klastering/#8-640x429 (Diakses: 18 April 2020).

Wisiastuti, D. K., (2014). Twitter sebagai Media Alternatif Informasi Publik. Yogyakarta: UIN Yogyakarta.

Witten, I. H., Frank, E. and Hall, M. A. (2011) Data Mining : Practical Machine learning Tools and Techniques Third Edition. Burlington: Elsevier.

Xhemali, D., J HINDE, C. & G STONE, R., 2009. Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. International Journal of Computer Science Issues, 4(1), pp. 16-23.

Yuli, M. (2017) 'Data Mining : Klasifikasi Menggunakan Algoritma C4 . 5', Jurnal Edik Informatika, 2(2), pp. 213–219.

Zaki, M. J. and Jr, W. M. (2014) DATA MINING AND ANALYSIS Fundamental Concepts and Algorithms, Procedia Computer Science. Cambridge University Press.

Zhao, Y., (2015). R and Data Mining: Examples and Case Studies. s.l.:Elsevier.

Biodata Penulis

Anjar Wanto, M.Kom

Lahir di Bah Jambi, 14 Februari 1985. Menyelesaikan Pendidikan D3 Manajemen Informatika di AMIK Tunas Bangsa Pematangsiantar, S1 Teknik Informatika di STT Poliprofesi Medan, S2 Teknik Informatika di Universitas Sumatera Utara (USU), dan saat ini sedang melanjutkan studi S3 Teknologi Informasi di UPI YPTK Padang, Sumatera Barat. Konsentrasi keilmuan dibidang Kecerdasan Buatan (Jaringan Saraf, Sistem Pendukung Keputusan, Data Mining). Penulis merupakan peneliti dan Dosen di STIKOM Tunas Bangsa Pematangsiantar hingga saat ini. Penulis juga merupakan Reviewer di 21 Jurnal terakreditasi yang ada di Perguruan Tinggi di Indonesia, diantara nya Terakreditasi SINTA 2 (JTIIK - Universitas Brawijaya, JSINBIS - Universitas Diponegoro, JEPIN - Universitas Tanjungpura). Informasi WA : 0822-9436-5929, Email : anjarwanto@amiktunasbangsa.ac.id / anjarwanto@gmail.com

Muhammad Noor Hasan Siregar

Lulusan S1. Teknik Industri (ST) dari Universitas Andalas Padang dan melanjutkan pendidikan di Teknik Informatika Universitas Putra Indonesia "YPTK" Padang (M.Kom). Bekerja Sebagai Dosen di Program Studi Manajemen, Universitas Graha Nusantara Padangsidimpuan dari tahun 2011. Informasi lebih lanjut dapat dilihat melalui blog di alamat http://hasan.dosen.ugn.ac.id dan bisa dihubungi melalui email noor.siregar@gmail.com

Agus Perdana Windarto

Lahir di Pematangsiantar pada tanggal 30 Agustus 1986. Penulis menyelesaikan pendidikan Magister (S2) bidang Ilmu Komputer pada tahun 2014 di Universitas Putra Indonesia ''YPTK'' Padang. Saat ini penulis sedang melanjutkan program Doktor (S3) di Universitas Putra Indonesia ''YPTK'' Padang. Penulis aktif mengajar di STIKOM Tunas Bangsa sejak tahun 2012 sebagai Dosen Tetap pada program studi Sistem Informasi. Fokus penelitian yang dilakukan adalah bidang Artificial Intelligence (Sistem Pendukung Keputusan, Sistem Pakar, Data Mining, Jaringan Saraf Tiruan, Fuzzy Logic, Deep Learning dan Algortima Genetika). Penulis juga aktif menjadi reviewer di berbagai Jurnal Nasional Akreditasi (SINTA 2 – SINTA 6). Selain itu penulis mengelola Komunitas yang dineri nama Pemburu Jurnal (2017-now) dilingkungan STIKOM Tunas Bangsa. Komunitas ini terdiri dari 20 Mahasiswa/i angkatan pertama (2017-2018) dimana semua memiliki Scopus ID dan menghasilkan publikasi sebanyak kurang lebih 135 artikel ilmiah baik seminar, conference dan jurnal nasional). Angkatan kedua terdiri dari 12 Mahasiswa/i yang menghasilkan 52 publikasi artikel ilmiah baik seminar baik seminar, conference dan jurnal nasional. Penulis juga pemenang beberapa Proposal Hibah Penelitian DIKTI (2x) tahun 2018-2019, pemenang Proposal Hibah Pengabdian DIKTI (1x) tahun 2019, pemenang proposal Hibah PKM-P sebagai pembimbing mahasiswa (2018) dan pemenang proposal Hibah PKM-AI sebagai pembimbing mahasiswa (2019). Sekarang ini penulis juga tergabung dalam komunitas Relawan Jurnal Indonesia (RJI) wilayah sumatera utara dan Forum Komunitas Perguruan Tinggi (FKPT).

Dedy Hartama

Lahir di Pematangsiantar pada tanggal 11 Oktober 1973. Penulis menyelesaikan pendidikan Magister (S2) bidang Ilmu Komputer pada tahun 2011 di Universitas Sumatera Utara. Selanjutnya Penulis menyelesaikan pendidikan Doktor (S3) bidang Ilmu Komputer pada tahun 2018 di Universitas Sumatera Utara. Penulis aktif mengajar di AMIK dan STIKOM Tunas Bangsa sejak tahun 2003 sebagai Dosen Tetap pada program studi Teknik Informatika Fokus penelitian yang dilakukan adalah bidang Data Mining dan Pemodelan Integer Linier Programming (Operation Research). Penulis juga aktif menjadi reviewer di berbagai Jurnal Nasional Akreditasi (SINTA 4 – SINTA 6). Selain itu penulis juga aktif mengisi Seminar Nasional di bidang Pemodelan dalam bidang Computer Science sebagai Nara Sumber. Saat ini Penulis menjabat sebagai Ketua STIKOM Tunas Bangsa Pematangsiantar.

Ni Luh Wiwik Sri Rahayu Ginantra, M.Kom.

Lahir dan besar di Bali. Pendidikan TK hingga SD diselesaikan di Kabupaten Bangli dan Pendidikan SMP hingga SMA di Kota Denpasar. Menyelesaikan Pendidikan S1 Teknik Informatika di Institut Teknologi Adhi Tama Surabaya dan S2 Ilmu Komputer Universitas Pendidikan Ganesha Singaraja. Dosen tetap di STMIK STIKOM Indonesia (STIKI Indonesia) di kota Denpasar. Mengajar mata kuliah Analisa Desain Sistem Informasi, Kecerdasan Buatan, Enterprise Information System, Software Engineering, Human Computer Interaction, Database.

Dr. Darmawan Napitupulu, S.T., M.Kom.

Sarjana Teknik dari Fakultas Teknik Elektro, Universitas Kristen Satya Wacana (UKSW) Salatiga, Magister Ilmu Komputer dari Universitas Indonesia dan Doktor Ilmu Komputer dari Universitas Indonesia diraih penulis pada tahun 2016 lalu. Saat ini bekerja sebagai peneliti di Lembaga Ilmu Pengetahuan Indonesia (LIPI) dengan fokus riset dalam bidang Sistem Informasi/Teknologi Informasi, E-Government, Smart City dan E-Business. Penulis aktif melakukan riset dengan mendapatkan berbagai hibah penelitian di level nasional. Hasil penelitiannya telah dipublikasikan di berbagai jurnal nasional terakreditasi dan jurnal internasional terindeks Scopus dan WoS. Sejak 2017, penulis adalah dosen di program Magister Ilmu Komputer, Universitas Budi Luhur. Penulis juga aktif diundang sebagai narasumber pada konferensi internasional dan mengisi berbagai workshop terkait E-Government (SPBE), Industri 4.0 serta Kiat/Teknik penulisan artikel ilmiah. Beberapa buku yang ditulis yakni Pengantar E-Government dan Sistem Informasi Bisnis, yang diterbitkan oleh penerbit Andi Yogyakarta. Saat ini penulis juga mengajar sebagai dosen di program Magister Ilmu Komputer di Fakultas Teknologi Informasi, Universitas Budi Luhur. Penulis dapat dihubungi melalui WA: 081314060258 atau melalui Surel: darwan.na70@gmail.com.

Dr. Edi Surya Negara, M.Kom

Dr. Edi Surya Negara, M.Kom dilahirkan di Padangsidimpuan, Sumatera Utara pada tanggal 5 Maret 1988 dari bapak Drs. Sahmiran Hrp, MPd. dan ibu Syamsinar Tanjung. Saat ini bekerja sebagai dosen tatap pada Program Pascasarjana, Program Studi Teknik Informatika Universitas Bina Darma. Menyelesaikan pendidikan S-3 Doktor Ilmu Komputer dan Teknologi Informasi dari Universitas Gunadarma Jakarta tahun 2017. Merupakan lulusan terbaik S-2 Magister Teknik Informasi Universita Bina Darma tahun 2012, dan lulusan terbaik S-1 Teknik Informatika Univesitas Bina Darma tahun 2011. Bidang keahlian dan konsentrasi adalah Data Science, Social Network Analytics, Social Media Analytics, Computer Network, dan Computer Network Security.

Muhammad Ridwan Lubis

ridwanlubis@amiktunasbangsa.ac.id

Lahir di Pematangsiantar, 26 Nopember 1986. Mulai aktif mengajar pada tahun 2010 di AMIK Tunas Bangsa dan melanjutkan kuliah program Pasca Sarjana (S2) di Universitas Sumatera Utara dan lulus pada tahun 2016. Mulai aktif melakukan penelitian dan memenangkan hibah Penelitian Dosen Pemula pada tahun pelaksanaan 2018 dan 2019 dan lulus Sertifikasi Dosen tahun 2019. Terus menulis dan mengembangkan diri dengan membiasakan melakukan kegiatan Kolaborasi Penelitian antar mahasiswa dan Dosen di bidang Ilmu Komputer seperti kecerdasan Buatan, Aplikasi Basis Data dan Bahasa Pemrograman. Mulai aktif menulis buku pada tahun 2020 dengan berkolaborasi dengan Dosen di luar Institusi dengan Buku pertama berjudul "**Biometrika: Teknologi Identifikasi**".

Sarini Vita Dewi

Lulusan S1 Teknik ELektro, bidang Teknologi Informasi di Universitas Syiah Kuala Banda Aceh dengan gelar S.T. Melanjutkan pendidikan S2 di Universitas Gadjah Mada pada Prodi Teknik Elektro Magister Teknologi Informasi (MTI). Saat ini menetap di banda aceh dan berprofesi sebagai dosen di Fakultas Ilmu Komputer universitas Ubudiyah Indonesia, Banda Aceh

Cahyo Prianto, S.Pd, MT

Lahir di Bandung 27 Juli 1984, memperoleh gelar sarjana Pendidikan Fisika dari Universitas Pendidikan Indonesia (UPI) dan Magister Teknik dari Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung (ITB). Pada tahun 2015 bergabung di program studi Teknik Informatika politeknik Pos Indonesia sebagai dosen tetap dan mengampu mata kuliah data mining. Saat ini fokus penelitian pada bidang data science. Penulis dapat dihubungi melalui cahyoprianto@poltekpos.ac.id

Data Mining

ALGORITMA & IMPLEMENTASI

Data mining dapat diterapkan untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Terdapat beberapa teknik yang digunakan dalam data mining, salah satu teknik data mining adalah clustering. Terdapat dua jenis metode clustering yang digunakan dalam pengelompokan data, yaitu hierarchical clustering dan non-hierarchical clustering.

Buku ini terdiri dari 10 (sepuluh) bab, yaitu :

YAYASAN KITA MENULIS
press@kitamenulis.id
www.kitamenulis.id

ISBN 978-623-7645-79-5