MODUL AJAR DATA MINING

INSTITUT TEKNOLOGI DAN BISNIS

(ITB) STIKOM BALI

2022

LEMBAR PENGESAHAN

MODUL AJAR

NAMA MODUL	: DATA MINING
PROGRAM STUDI	: SISTEM INFORMASI
TAHUN AJARAN	: 2021/2022 GANJIL

DISAHKAN PADA :

Tanggal/Tahun : 20 Oktober 2021

DISETUJUI

K.A Program Studi Sistem Informasi

Ricky Aurelius Nurtanto Diaz, S.Kom.,M.T

NIDN. 0820128601

Denpasar, 20 Oktober 2021

Dosen Penyusun

Dr. Gede Angga Pradipta S.T.,M.Eng

NIDN. 0819078803

Satuan Acara Perkuliahan (SAP)

Program Studi : Sistem Informasi

Kode Mata kuliah : SI9379

Nama Mata kuliah : Data Mining

Semester/SKS : Ganjil/4

Deskripsi

Data mining adalah suatu proses pengumpulan informasi dan data yang penting dalam jumlah yang besar atau big data. Dalam proses ini seringkali memanfaatkan beberapa metode, seperti matematika, statistika dan pemanfaatan teknologi artificial intelligence (AI). Proses pengolahan data menjadi satu hal sangat penting di era persaingan bisnis yang mengharuskan mendapat informasi cepat. Informasi terkait data yang berguna dalam proses bisnis, hingga penentuan strategi kedepannya, karena itulah penggunaan penambangan data atau disebut dengan data mining sangat penting bagi kelangsungan bisnis yang berjalan.

Tujuan Instruksional Umum

1. Mahasiswa mampu melakukan pengolahan data dengan berbagai tipe data.
2. Mahasiswa mampu melakaukan analisis keterhubungan variable.
3. Mahasiswa mampu melakukan prapemrosesan data
4. Mahaiswa mampu memahami konsep algoritma machine learning
5. Mahaiswa mampu mengevaluasi performa metode machine learning.

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas segala rahmat-Nya sehingga modul ajar matakuliah Data Visualisasi ini bisa tersusun hingga selesai. Penulis berharap semoga modul ini bisa memenuhi kebutuhan peserta didik mata kuliah Data Mining. Pembahasan modul ini dimulai dengan menjelaskan tujuan yang akan dicapai pada mata kuliah Data Mining. Penulis sadar masih banyak kekurangan didalam penyusunan modul ini, karena keterbatasan pengetahuan serta pengalaman. Untuk itu penulis mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan modul ini.

Denpasar, Maret 2023

Penulis

DAFTAR ISI

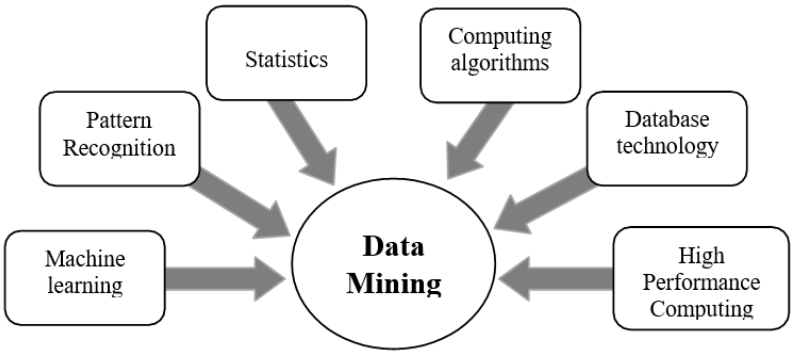
DAFTAR GAMBAR

DAFTAR TABEL

MODUL I

1.1 Pengantar Data Mining

Data mining dikenal sejak tahun 1990-an, ketika adanya suatu pekerjaan yang memanfaatkan data menjadi suatu hal yang lebih penting dalam berbagai bidang, seperti marketing dan bisnis, sains dan teknik, serta seni dan hiburan. Sebagian ahli menyatakan bahwa data mining merupakan suatu langkah untuk menganalisis pengetahuan dalam basis data atau biasa disebut Knowledge Discovery in Database (KDD). *Data mining* merupakan proses untuk menemukan pola data dan pengetahuan yang menarik dari kumpulan data yang sangat besar. Sumber data dapat mencakup *database*, *data warehouse*, *web*, *repository*, atau data yang dialirkan ke dalam sistem dinamis (Han, 2006). *Data mining*, secara sederhana merupakan suatu langkah ekstraksi untuk mendapatkan informasi penting yang sifatnya implisit dan belum diketahui. Selain itu, *data mining* mempunyai hubungan dengan berbagai bidang diantaranya statistik, *machine learning* (pembelajaran mesin), *pattern recognition*, *computing algorithms*, *database technology*, dan *high performance computing*. Diagram hubungan *data mining* disajikan pada Gambar 1.1. *Data mining*, secara sederhana merupakan suatu langkah ekstraksi untuk mendapatkan informasi penting yang sifatnya implisit dan belum diketahui. Selain itu, *data mining* mempunyai hubungan dengan berbagai bidang diantaranya statistik, *machine learning* (pembelajaran mesin), *pattern recognition*, *computing algorithms*, *database technology*, dan *high performance computing*. Diagram hubungan *data mining* disajikan pada Gambar 1.1.



Gambar 1. 1. Ilustrasi 1.1

Secara sistematis, langkah utama untuk melakukan *data mining* terdiri dari tiga tahap, yaitu sebagai berikut (Gonunescu, 2011);

a. Eksplorasi atau pemrosesan awal data

Eksplorasi atau pemrosesan awal data terdiri dari pembersihan data, normalisasi data, transformasi data, penanganan *missing value*, reduksi dimensi, pemilihan subset fitur, dan sebagainya.

b. Membangun model dan validasi

Membangun model dan validasi, yaitu melakukan analisis dari berbagai model dan memilih model sehingga menghasilkan kinerja yang terbaik.

1. Cleaning and Integration

   a. *Data Cleaning* (Pembersihan Data)

      Data cleaning (Pembersihan data) adalah proses yang dilakukan untuk menghilangkan *noise* pada data yang tidak konsisten atau bisa disebut tidak relevan. Data yang diperoleh dari *database* suatu perusahaan maupun hasil eksperimen yang sudah ada, tidak semuanya memiliki isian yang sempurna misalnya data yang hilang, data yang tidak valid, atau bisa juga hanya sekedar salah ketik. Data yang tidak relevan itu dapat ditangani dengan cara dibuang atau sering disebut dengan proses *cleaning*. Proses *cleaning* dapat berpengaruh terhadap performa dari teknik *data mining*.

   b. *Data Integration* (Integrasi data)

      Integrasi data merupakan proses penggabungan data dari berbagai *database* sehingga menjadi satu *database* baru. Data yang diperlukan pada proses *data mining* tidak hanya berasal dari satu *database* tetapi juga dapat berasal dari beberapa *database*.

2. Selection and Transformation

   a. *Data Selection* (Seleksi Data)

      Tidak semua data yang terdapat dalam *database* akan dipakai, karena hanya data yang sesuai saja yang akan dianalisis dan diambil dari *database*. Misalnya pada sebuah kasus *market basket analysis* yang akan meneliti faktor kecenderungan pelanggan, maka tidak perlu mengambil nama pelanggan, cukup dengan id

pelanggan saja.

b. *Data Transformation* (Transformasi Data)

Transformasi data merupakan proses pengubahan data dan penggabungan data ke dalam format tertentu. *Data mining* membutuhkan format data khusus sebelum diaplikasikan. Misalnya metode standar seperti analisis asosiasi dan *clustering* hanya bisa menerima input data yang bersifat kategorikal. Karenanya data yang berupa angka numerik apabila mempunyai sifat kontinyu perlu dibagi-bagi menjadi beberapa interval. Proses ini sering disebut dengan transformasi data.

3. Proses Mining

Proses *mining* dapat disebut juga sebagai proses penambangan data. Proses *mining* merupakan proses utama yang menggunakan metode untuk menemukan pengetahuan berharga yang tersembunyi dari data.

4. Evaluation and Protection

a. Evaluasi Pola (*Pattern Evaluation*)

Evaluasi pola bertugas untuk mengidentifikasi pola-pola yang menarik ke dalam *knowledge based* yang ditemukan. Pada tahap ini dihasilkan pola- pola yang khas dari model klasifikasi yang dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai dengan hipotesa, terdapat beberapa alternatif yang bisa diambil seperti menjadikannya umpan balik untuk memperbaiki proses *data mining*, atau mencoba metode *data mining* lain yang lebih sesuai.

b. Presentasi Pengetahuan (*Knowledge Presentation*)

Knowledge presentation merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan atau informasi yang telah digali oleh pengguna. Tahap terakhir dari proses *data mining* adalah memformulasikan keputusan dari hasil analisis yang didapat. Pembangunan model dilakukan menggunakan metode-metode seperti klasifikasi, regresi, analisis *cluster*, dan asosiasi.

c. Penerapan

Penerapan dilakukan dengan menerapkan model yang dipilih pada data yang baru untuk menghasilkan kinerja yang baik pada masalah yang diinvestigasi.

1.2 Dataset

Data mining tidak pernah lepas dari yang namanya *dataset*, karena dalam pengolahan *data mining*, *dataset* sangat dibutuhkan sebagai objek untuk mendapakan pengetahuan. Dalam terminologi statistik *dataset* adalah kumpulan dari suatu objek yang mempunyai atribut atau variabel tertentu, di mana untuk setiap objek merupakan individu dari data yang mempunyai sejumlah atribut atau variabel tersebut. Nama lain dari objek yang sering digunakan adalah *record, point, vector, pattern, event, observation*, dan *case*. Sementara itu, baris yang menyatakan objek-objek data dan kolom disebut atribut. Atribut juga dapat disebut dengan variabel*, field*, fitur atau dimensi.

1.2.1 Jenis jenis Dataset

Karakteristik umum *dataset* yang berpengaruh dalam proses *data mining* ada tiga, diantaranya dimensionalitas, sparsitas, dan resolusi. Sedangkan jenis *dataset* juga ada tiga macam, yaitu sebagai berikut:

1. Record Data

   Dataset yang berbentuk *record*, tidak mempunyai hubungan antara baris data yang satu dengan baris data yang lainnya. Setiap baris data berdiri sendiri sebagai sebuah data individu. Jadi, *record data* merupakan data yang terdiri dari sekumpulan *record*, yang masing-masing *record* terdiri dari satu set atribut yang tetap. Contoh *record data* CKD ditunjukkan pada Tabel 1.1.

   Tabel 1. 1. Tabel Ilustrasi 1.1

Nama Pasien	Umur	Tekanan Darah	Kepekaan Urine	Kadar Gula	Nanah	Gumpalan Nanah	Kelas
Eka	48	80	1.020	0	Normal	notpresent	ckd
Aldi	7	50	1.020	0	Normal	notpresent	ckd
April	62	80	1.010	3	Normal	notpresent	ckd
Elham	48	70	1.005	0	abnormal	present	not ckd
Hestu	51	80	1.010	0	Normal	notpresent	ckd
Winda	68	70	1.010	0	Normal	notpresent	ckd
Novi	24	?	1.015	4	abnormal	notpresent	notckd
Nerly	50	60	1.010	4	abnormal	present	notckd
Ikhsan	68	70	1.015	1	Normal	present	notckd
Hani	68	80	1.010	2	abnormal	present	ckd
Budi	40	80	1.015	0	Normal	notpresent	ckd

Tiyo	47	70	1.015	0	Normal	notpresent	notckd

2. Data Graph

   Data *graph* adalah data yang mempunyai bentuk *graph* yang terdiri dari simpul (*node*) dan rusuk (*edge*). Yang termasuk dalam data *graph* diantaranya adalah HTML *links* (dalam WWW), struktur molekul, dan sebagainya.

3. Ordered Data

   Ordered data merupakan data-data yang memperhatikan urutan nilai- nilainya. Yang termasuk dalam data terurut adalah *genomic sequence data* atau *spatio-temporal data.* Contoh data terurut *genomic sequence data* dapat dilihat pada Gambar 1.2.

GAGGATTAAT	AAATTATAAA	TGTTATTACA
TTACACTGTT	GCACGTCCAC	GTGTTCGTCC
TGATCTTGTT	ATATCATTAT	TATTATTGTT
GTGTACCATA	GTAATCTGAA	AGGAACCGCT
ATAGATTCTA	TTTTCAATTT	CTCAAATCTA
GAACGTGAGT	TATTAAGTTA	ATCTAAATAT

Gambar 1. 2. Ilustrasi 1.2

1.3 Jenis-jenis Atribut

Atribut adalah suatu simbol yang menggambarkan identitas atau karakteristik objek. Sebagai contoh atribut yang menggambarkan objek pasien rumah sakit adalah nama, umur, golongan darah, dan tekanan darah. Berikut penjelasan dari empat macam atribut berdasarkan contohnya.

1. Atribut Normal

   Atribut nominal adalah nilai atribut yang diperoleh dengan cara kategorisasi karena nilainya menggambarkan kategori, kode, atau status yang tidak memiliki urutan. Misalnya, atribut golongan darah yang mempunyai empat kemungkinan nilai yaitu A, B, AB, dan O. Contoh lainnya seperti atribut jenis kelamin yang bisa bernilai pria dan wanita.

2. Atribut Ordinal

   Atribut ordinal adalah atribut yang memiliki nilai dengan menggambarkan urutan atau peringkat. Namun, ukuran perbedaan antara dua nilai yang berurutan tidak diketahui.

Atribut ordinal sangat berguna dalam survei, yaitu untuk penilaian subjektif (kualitatif) yang tidak dapat diukur secara objektif. Misalnya, kepuasan pelanggan yang menghasilkan atribut bernilai ordinal, yaitu 0 (Tidak Puas), 1 (Cukup Puas), 2 (Puas), 3 (Sangat Puas).

3. Atribut Interval (Jarak)

   Atribut interval adalah atribut numerik yang diperoleh dengan melakukan pengukuran, di mana jarak dua titik pada skala sudah diketahui dan tidak mempunyai titik nol yang absolut. Misalnya, suhu 0°C-100°C atau tanggal 1 sampai tanggal 31.

4. Atribut Rasio (Mutlak)

   Atribut rasio adalah atribut numerik dengan titik nol absolut. Artinya, jika sistem pengukuran menggunakan rasio, dapat dihitung perkalian atau perbandingan antara suatu nilai dengan nilai yang lain. Misalnya, berat badan Doni 20 kg, berat badan Amanah 40 kg, berat badan Faiz 60 kg dan berat badan Udin 80 kg. Jika diukur dengan skala rasio maka berat badan Udin dua kali berat badan Amanah. KDD mengalami beberapa proses pengolahan. Sebelum diterapkan pada algoritma *data mining*, *dataset* dapat diolah dengan cepat dan menghasilkan kesimpulan yang tepat. Beberapa proses pengolahan awal adalah proses pengumpulan (*aggregation*), penarikan contoh (*sampling*), pengurangan dimensi (*dimensionality reduction*), pemilihan fitur (*feature selection*), pembuatan fitur (*fitur creation*), pendiskritan dan pembineran (*discretization and binarization* dan transformasi atribut (*attribute transformation*). Oleh karena itu, perlu diterapkannya pemrosesan awal pada data sebelum melakukan proses *data mining* yang akan dibahas pada bab selanjutnya.

MODUL II

2.1 Konsep Dasar Klasifikasi

Aplikasi lain yang penting dari data mining adalah kemampuannya untuk melakukan proses klasifikasi pada suatu data dalam jumlah besar. Hal ini sering disebut *mining classification rules*. Sebagai contoh, sebuah dealer mobil ingin mengkiasifikasikan pelanggannya menurut kecenderungan mereka untuk menyukai mobil jenis tertentu, sehingga para sales yang bekerja disitu akan mengetahui siapa yang harus didekati, kemana katalog mobil jenis baru harus dikirim, sehingga hal ini akan sangat membantu dalam hal promosi. Klasifikasi data adalah suatu proses yang menemukan properti-properti yang sama pada sebuah himpunan obyek di dalarn sebuah basis data, dan mengklasifikasikannya ke dalam kelas-kelas yang berbeda menurut model klasifikasi yang ditetapkan. Untuk membentuk sebuah model klasifikasi, suatu sampel basis data 'E' diperlakukan sebagai training set, dimana setiap tupel terdiri dari himpunan yang sama yang memuat atribut yang beragam seperti tupel-tupel yang terdapat dalam suatu basis data yang besar 'W'. Setiap tupel diidentifkasikan dengan sebuah label atau identitas kelas. Tujuan dari klasifikasi ini adalah pertama-tama untuk menganalisa training data dan membentuk sebuah deskripsi yang akurat atau sebuah model untuk setiap kelas berdasarkan feature-feature yang tersedia di dalam data itu. Deskripsi dari masing-masing kelas itu nantinya akan digunakan untuk mengklasifikasikan data yang hendak di test dalam basis data 'W', atau untuk membangun suatu deskripsi yang lebih baik untuk setiap kelas dalam basis data. Contoh untuk model ini adalah prediksi terhadap resiko pemberian kredit. Data terdiri dari orang-orang yang telah menerima kredit. Sebagian kreditur menjalankan kewajiban dengan baik, dan sebagian lagi tidak. Data mining, harus mampu mendefinisikan atribut-atribut apa yang paling berpengaruh.

2.1.1 Algoritma Naïve Bayes

- Bayesian classification adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class.
- BC didasarkan pada teorema Bayes yg memiliki kemampuan klasifikasi serupa dengan decision tree dan neural network.
- Memiliki akurasi dan kecepatan yg tinggi saat diaplikasikan ke dalam database yg besar.
- Bentuk Umum Teorema Bayes

P(H I X) = P(X I H) P(H)

P(X)

Keterangan :

X : data dgn class yg belum diketahui

H : hipotesis data X

P(HIX) : probabilitas hipotesis H berdasar kondisi X (posteriori probability)

P(H) : probabilitas hipotesis H (prior porbability)

P(XIH) : probabilitas X berdasar kondisi pada hipotesis H

P(X) : probabilitas dari X

Contoh Soal :

Tabel 2. 1. Tabel Ilustrasi 2.1

NO	JENIS KELAMIN	STATUS MAHASISWA	STATUS PRENIKAHAN	IPK Semester 1-6	STATUS KELULUSAN
1	LAKI - LAKI	MAHASISWA	BELUM	3.17	TEPAT
2	LAKI - LAKI	BEKERJA	BELUM	3.30	TEPAT

3	PEREMPUAN	MAHASISWA	BELUM	3.01	TEPAT
4	PEREMPUAN	MAHASISWA	MENIKAH	3.25	TEPAT
5	LAKI - LAKI	BEKERJA	MENIKAH	3.20	TEPAT
6	LAKI - LAKI	BEKERJA	MENIKAH	2.50	TERLAMBAT
7	PEREMPUAN	BEKERJA	MENIKAH	3.00	TERLAMBAT
8	PEREMPUAN	BEKERJA	BELUM	2.70	TERLAMBAT
9	LAKI - LAKI	BEKERJA	BELUM	2.40	TERLAMBAT
10	PEREMPUAN	MAHASISWA	MENIKAH	2.50	TERLAMBAT
11	PEREMPUAN	MAHASISWA	BELUM	2.50	TERLAMBAT
12	PEREMPUAN	MAHASISWA	BELUM	3.50	TEPAT
13	LAKI - LAKI	BEKERJA	MENIKAH	3.30	TEPAT

Jika seorang mahasiswa dengan data sebagai berikut, prediksi kelulusannya

Jawaban :

- **Tahap 1 menghitung jumlah class/label**

  P(Y= TEPAT) = 8/15

  (*jumlah data "TEPAT" pada komom 'STATUS KELULUSAN' dibagijumlah data*)

  P(Y= TERLAMBAT) = 7/15

  (*jumlah data "TERLAMBAT" pada komom 'STATUS KELULUSAN'dibagi jumlah data*)

- **Tahap 2 menghitung jumlah kasus yang sama dengan class yang sama**

  P(JENIS KELAMIN = LAKI - LAKI | Y= TEPAT) = 5/8

  (jumlah data jenis kelamin "laki-laki" dengan keterangan "TEPAT" dibagi jumlah data TEPAT)

  P(JENIS KELAMIN = LAKI - LAKI | Y= TERLAMBAT) = 3/7

(jumlah data jenis kelamin “laki-laki” dengan keterangan ”TERLAMBAT”dibagi jumlah data TERLAMBAT)

P(STATUS MAHASISWA = MAHASISWA | Y= TEPAT) = 5/8

(jumlah data dengan status mahasiswa dengan keterangan “TEPAT” dibagi jumlah data TEPAT)

P(STATUS MAHASISWA = MAHASISWA | Y= TERLAMBAT) = 3/7

(jumlah data dengan status mahasiswa dengan keterangan“TERLAMBAT” dibagi jumlah data TERLAMBAT)

P(STATUS PRENIKAHAN = BELUM| Y= TEPAT) = 4/8

(jumlah data dengan status pernikahan “Belum menikah” dan keterangan “TEPAT” dibagijumlah data TEPAT)

Tabel 2. 2 Tabel Ilustrasi 2.2

KELAMIN	STATUS	PRENIKAHAN	IPK	KETERANGAN
LAKI - LAKI	MAHASISWA	BELUM	2.70	???

P(STATUS PRENIKAHAN = BELUM| Y= TERLAMBAT) = 4/7

(jumlah data dengan status pernikahan “Belum menikah” dan keterangan “TERLAMBAT” dibagijumlah data TERLAMBAT)

P(IPK = 2.70| Y= TEPAT) = 0/8

(jumlah data IPK “2.70” dengan keterangan “TEPAT” dibagi jumlah data TEPAT)

P(IPK = 2.70| Y= TERLAMBAT) = 1/7

(jumlah data IPK “2.70” dengan keterangan “TERLAMBAT” dibagi jumlah data TERLAMBAT)

- **Tahap 3 kalikan semua hasil variable TEPAT & TERLAMBAT**

  P (KELAMIN=LAKI – LAKI), (STATUS MHS=MAHASISWA), (PRENIKAHAN = BELUM), (IPK = 2.70) |TEPAT)

= {P(P(KELAMIN =LAKI-LAKI|Y=TEPAT). P(STATUS MHS = MAHASISWA | Y= TEPAT) . P(PRENIKAHAN =BELUM|Y=TEPAT). P(IPK = 2.70| Y= TEPAT)

= 5/8 . 5/8 . 4/8 . 0/8 . 8/15

= 0

P (KELAMIN=LAKI – LAKI), (STATUS MHS=MAHASISWA), (PRENIKAHAN = BELUM), (IPK = 2.70) |TERLAMBAT)

= {P(P(KELAMIN =LAKI-LAKI|Y= TERLAMBAT). P(STATUS MHS = MAHASISWA | Y= TERLAMBAT) . P(PRENIKAHAN = BELUM|Y= TERLAMBAT). P(IPK = 2.70| Y= TERLAMBAT)

= 3/7 . 3/7 . 4/7 . 1/7 . 7/15

= 0,0069

- **Tahap 4 Bandingkan hasil class TEPAT & TERLAMBAT**

➔ Karena hasil (P|TERLAMBAT) lebih besar dari (P|TEPAT) maka keputusanya adalah "TERLAMBAT"

Tabel 2. 3 Tabel Ilustrasi 2.3

KELAMIN	STATUS	PRENIKAHAN	IPK	KETERANGAN
LAKI - LAKI	MAHASISWA	BELUM	2.70	TERLAMBAT

2.1.2 Algoritma C.45 (Pohon Keputusan)

- Mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan
- Mengeksplorasi data untuk menemukan hubungan antara sejumlah calon variabel input dengan sebuah variabel target
- Struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan record yang lebih kecil dengan menerapkan serangkaian aturan keputusan
- Data dalam pohon keputusan biasanya dinyatakan dalam bentuk tabel dengan atribut dan record

1. Menggunakan Nilai Gain
2. Pilih atribut sebagai akar (nilai gain tertinggi)
3. Buat cabang untuk tiap nilai
4. Bagi kasus dalam cabang
5. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama

- Rumus untuk menghitung nilai Gain

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

Keterangan :

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|Si| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam S

- Rumus untuk menghitung Nilai Entropy

$$entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi$$

Keterangan :

S : himpunan kasus

A : fitur

n : jumlah partisi

P_i : proporsi dari S_i terhadap S

- **Contoh Soal : Buat Pohon keputusan dari tabel dibawah ini**

Tabel 2. 4 Tabel Ilustrasi 2.4

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Cloudy	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	Yes
Cloudy	Cool	Normal	True	yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Cloudy	Mild	High	True	Yes
Cloudy	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

- Jawab :

1. Menghitung jumlah kasus, jumlah kasus untuk keputusan yes, jumlah kasus untuk keputusan no, dan entropy dari semua kasus

Jumlah kasus : 14

Jumlah kasus untuk keputusan yes : 10

Jumlah kasus untuk keputusan no : 4

➔ **Entropy dari semua kasus :**

$$entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi$$

Entropy(S) = ((-4/14) * $\log_2$(4/14)) + ((-10/14) * $\log_2$(10/14))

= 0.52 + 0.35

= 0.87

➔ **Entropy dari setiap kasus :**

Entropy(Outlook = Cloudy) = ((-0/4) * $\log_2$(0/4)) + ((-4/4) * $\log_2$(4/4)) = 0

Entropy(Outlook = Rainy) = ((-1/5) * $\log_2$(1/5)) + ((-4/5) * $\log_2$(4/5)) = 0,721928

Entropy(Outlook = Sunny) = ((-3/5) * $\log_2$(3/5)) + ((-2/5) * $\log_2$(2/5)= 0,970951

Entropy(Temperature = Cool) = ((-0/4) * $\log_2$(0/4)) + ((-4/4) * $\log_2$(4/4)) = 0

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

Entropy(Temperature = Mild) = ((-2/6) * $\log_2$(2/6)) + ((-4/6) * $\log_2$(4/6)) = 0,918296

Entropy(Humidity = High) = ((-4/7) * $\log_2$(4/7)) + ((-3/7) * $\log_2$(3/7)) = 0,985228

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

Entropy(Windy = False) = ((-2/8) * $\log_2$(2/8)) + ((-6/8) * $\log_2$(6/8)) = 0,811278

Entropy(Windy = True) = ((-4/6) * $\log_2$(4/6)) + ((2/6) * $\log_2$(2/6)) = 0,918296

2. Menghitung nilai Gain

Gain (Total,Outlook)

= 0,863121 – ((4/14)*0)+((5/14)*0.723))+((5/14)*0,970))

= 0,258521

Gain (Total,Temperature)

= 0,863121 – ((4/14)*0)+((4/14)*1))+((6/14)*0,918))

= 0,183851

Gain (Total,humidity)

= 0,863121 – ((7/14)*0)+((7/14)*0,955228))

= 0,370506

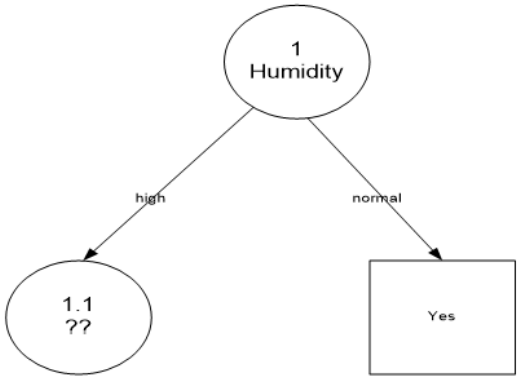
Gain (Total,windy)

= 0,863121 – ((8/14)*0,811278)+((6/14)*0,918296))

= 0,005978

3. Bentuk Pohon Keputusan

   Dari perhitungan Nilai Gain Humadity adalah nilai tertinggi sehingga menjadi akar dari pohon.



Gambar 2. 1. Ilustrasi 2.1

Menghitung jumlah kasus, jumlah kasus untuk keputusan Yes, jumlah kasus untuk keputusan No, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut OUTLOOK, TEMPERATURE dan WINDY yang dapat menjadi node akar dari nilai atribut HIGH.

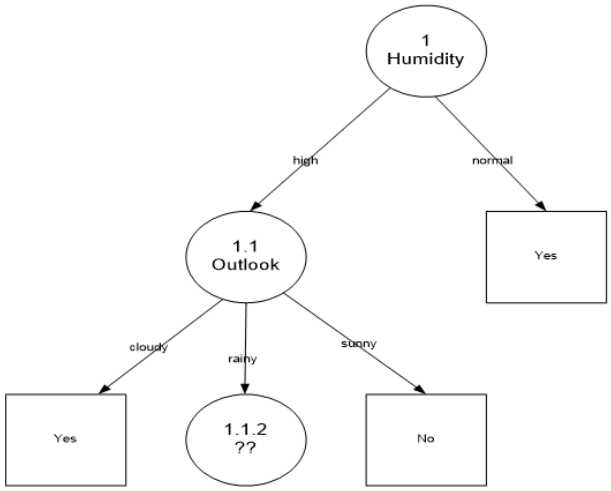
Jumlah kasus : 7

Jumlah kasus untuk keputusan yes : 4

Jumlah kasus untuk keputusan no : 3

➔ **Gain tertinggi : outlook**

➔ **Outlook menjadi akar**



Gambar 2. 2. Ilustrasi 2.2

Menghitung jumlah kasus, jumlah kasus untuk keputusan Yes, jumlah kasus untuk keputusan No, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut TEMPERATURE dan WINDY yang dapat menjadi node cabang dari nilai atribut RAINY.

2.1.3 Algoritma KKN (K-Nearest Neghbor)

Algoritma K-nearest neighbor (KNN) merupakan algoritma supervised learning di mana hasil klasifikasi data baru berdasar kepada kategori mayoritas tetangga terdekat ke-K. Tujuan dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan data training. Algoritma KNN menggunakan kalsifikasi ketetenggaan sebagai prediksi terhadap data baru

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah *k* buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai *k* yang terbaik untuk algoritma ini tergantung pada data; secara umumnya, nilai *k* yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara

setiap klasifikasi menjadi lebih kabur. Nilai *k* yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, *k* = 1) disebut algoritma *nearest neighbor*. Berikut rumus pencarian jarak menggunakan rumus *Eucledian Distence*

$$d = \sqrt{\sum_{i=1}^{p} (x_{2i} - x_{1i})^2}$$

Dengan :

x1 = sampel data

x2 = data uji

i = varibel data

d = jarak

p = dimensi data

Ketepatan algoritma KNN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik.

Langkah-langkah algoritma KNN:

- Tentukan parameter K = jumlah tetangga terdekat.
- Hitung jarak antara instance query dan semua sampel pelatihan.
- Urutkan jarak dan tentukan tetangga terdekat berdasar jarak minimum K-th.
- Kumpulkan kategori Y dari tetangga terdekat.
- Gunakan mayoritas kecil dari kategori tetangga terdekat sebagai nilai prediksi query instance.

Contoh Pertama

Kita mempunyai data dari survey kuesioner (dengan meminta pendapat masyarakat) dan testing objektif dengan dua atribut (*acid durability and strength*) untuk mengklasifikasi apakah suatu bahan pembuat kertas baik atau tidak. Dengan data yang telah ada adalah sebagai berikut:

Tabel 2. 5. Tabel Ilustrasi 2.5

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Sekarang pabrik ingin menghasilkan sebuah bahan kertas yang melewati uji laboratorium dengan **x1=3 dan x2 = 7.**

- Tentukan parameter K = jumlah tetangga terdekat, misal kita gunakan K = 3.
- Hitung jarak antara instance query dengan semua sampel pelatihan, koordinat instance query adalah (3, 7).

Tabel 2. 6. Tabel Ilustrasi 2.6

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Square Distance to query instance (3, 7)
7	**7**	$\sqrt{(7-3)^2+(7-7)^2} = \sqrt{16} = 4$
7	4	$\sqrt{(7-3)^2+(4-7)^2} = \sqrt{25} = 5$
3	4	$\sqrt{(3-3)^2+(4-7)^2} = \sqrt{9} = 3$
1	4	$\sqrt{(1-3)^2+(4-7)^2} = \sqrt{13} = 3.6$

- Urutkan jarak dan tentukan tetangga terdekat berdasar jarak minimum ke-K

Tabel 2. 7. Tabel Ilustrasi 2.7

X1=Acid Durability (seconds)	**X2 = Strength (kg/square meter)**	**Square Distance to query instance (3, 7)**	Rank minimum distance	Is it included in 3-Nearest neighbors?
7	7	4	**3**	Yes
7	4	5	4	No
3	4	3	**1**	Yes
1	4	3.6	**2**	Yes

- Kumpulkan kategori Y dari tetangga terdekat. Perhatikan, pada baris kedua kolom terakhir, kategori tetangga terdekat (Y) tidak termasuk karena peringkat datanya lebih besar dari 3 (=K)
- Gunakan mayoritas kecil dari kategori tetangga terdekat sebagai nilai prediksi dari instance query.

Tabel 2. 8. Tabel Ilustrasi 2.8

X1 = Acid Durability (seconds)	**X2 = Strength (kg/square meter)**	**Square Distance to query instance (3, 7)**	**Rank minimum distance**	**Is it included in 3-Nearest neighbors?**	Y = Category of nearest Neighbor
7	**7**	4	**3**	**Yes**	**Bad**
7	**4**	5	**4**	**No**	**-**
3	**4**	3	**1**	**Yes**	**Good**
1	**4**	3.6	**2**	**Yes**	**Good**

- Kita mempunyai 2 good dan 1 bad, karena 2 > 1 maka kita simpulkan bahwa bahan kertas yang baru yang melewati tes uji laboratorium dengan X1 = 3 dan X2 = 7 termasuk dalam kategori Good.

Metoda Konversi Atribut Nominal Ke Atribut Numerik

Ada 2 metode yang digunakan untuk mengkonversi atribut nominal ke atribut numerik

1. Mengkodekan dengan nilai numerik
   Contoh :
   - Yes = 1 dan no = 0
   - Biru = 1, Kuning = 2 dan merah = 3
2. Menghitung jaraknya

$$\boldsymbol{Distance} = (\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{dist}(\boldsymbol{X_i}, \boldsymbol{X_i})$$

Dist(x_1,x_2) = 0 jika x_1 dan nominal dan $x_1 = y_2$

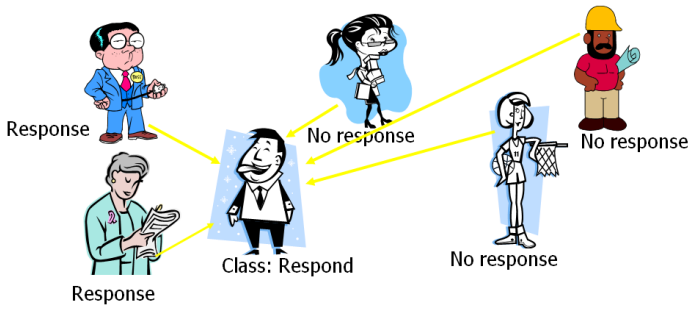
1 jika x_1 dan nominal dan $x_1 \neq y_2$

Contoh Kedua

Ada 5 data yang menunjukkan kelas apakah orang tersebut RESPONSE atau NO RESPONSE

John, Rachel, Hannah, Tom, Nellie, kemudian diinputkan data baru yaitu **David.**

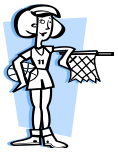
Apakah David di kelas RESPOSE atau NO RESPONSE?



Gambar 2. 3 Ilustrasi 2.3

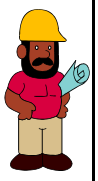
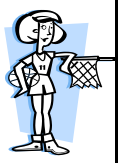
Data Lengkap masing-masing customer

Tabel 2. 9. Tabel Ilustrasi 2.9

Customer	Age	Income (K)	No. cards	RESPOSE
John 	35	35K	3	NO
 Rachel	22	50K	2	YES
Hannah 	63	200K	1	NO
Tom 	59	170K	1	NO
Nellie 	25/63 = 0.39	40K	4	YES
David 	37/63 = 0.58	50/200 = 0.25	2	??

Langkah Pertama Hitung jarak antara data baru (DAVID) dengan data yang telah ada

Tabel 2. 10. Tabel Ilustrasi 2.10

Customer	Age	Income (K)	No. cards	RESPOSE	Distence
 **John**	**35**	**35K**	**3**	**NO**	$\sqrt{(37-35)^2+(50-35)^2+(2-3)^2}$ $=\sqrt{13}\ =15.6$
 **Rachel**	**22**	**50**	**2**	**YES**	$\sqrt{(37-22)^2+(50-50)^2+(2-2)^2}$ $=\sqrt{225}\ =15$
 **Hannah**	**63**	**200**	**1**	**NO**	$\sqrt{(37-63)^2+(50-200)^2+(2-1)^2}$ $=\sqrt{23177}\ =152.24$
 **Tom**	**59**	**170**	**1**	**NO**	$\sqrt{(37-59)^2+(50-170)^2+(2-1)^2}$ $=\sqrt{14885}\ =122$
 **Nellie**	**25**	**40**	**4**	**YES**	$\sqrt{(37-25)^2+(50-40)^2+(2-4)^2}=$ $\sqrt{248}\ =15.75$

 David	37	50K	2	YES	

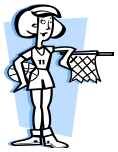
Berdasarkan perhitungan jarak dan nilai K = 3, artinya hanya diambil 3 yang memiliki jarak terpendek dari lima data yang disajikan. Ketiga data yang jaraknya terpendik yaitu 15.S6; 15; 15.75 dan kelas responsenya YES, NO, YES. Kesimpulan yang dapat ditarik David masuk dalam kelas RESPONSE YES

Normalisasi Variabel

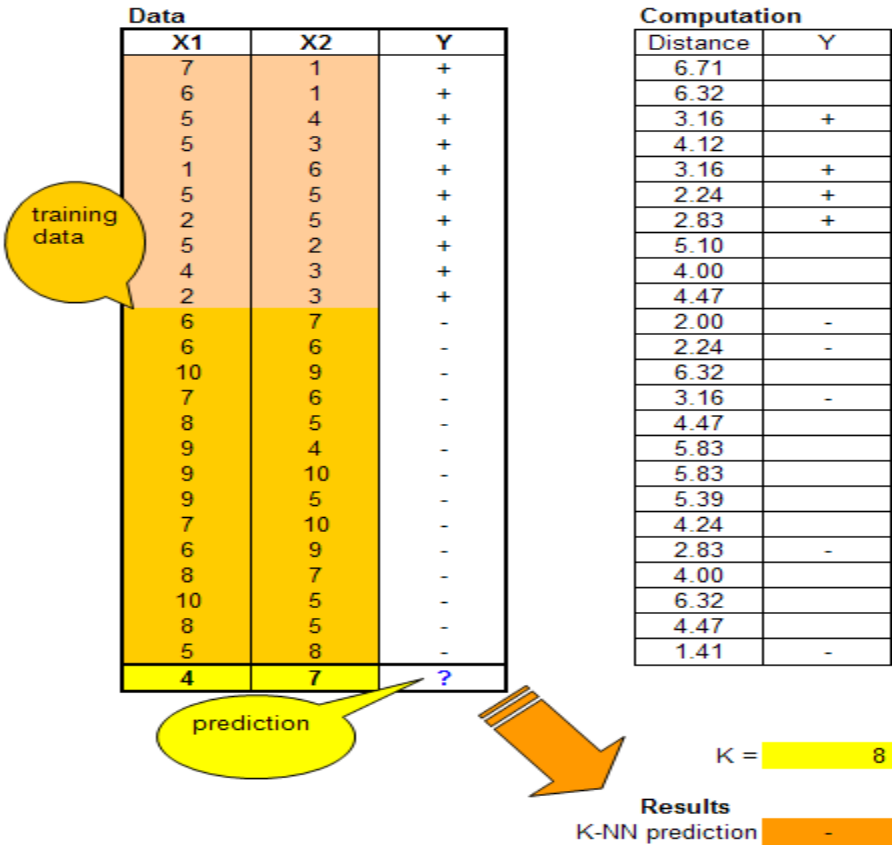
Untuk memudahkan perhitungan biasanya dilakukan normalisasi. Normalisasi yaitu membagi nilai variabel dengan nilai yang tertinggi dari data yang ada. Berikut contoh normalisasi dari contoh kedua yang disajikan di atas.

Tabel 2. 11. Tabel Ilustrasi 2.11

Customer	Age	Income (K)	No. cards
John 	55/63 = 0.55	35/200 = 0.175	¾ = 0.75
 Rachel	22/63 = 0.34	50/200 = 0.25	2/4 = 0.5
Hannah 	63/63 = 1	200/200=1	¼ = 0.25

Tom 	59/63 = 0.93	170/200 = 0.85	¼ = 0.25
Nellie 	25/63 = 0.39	40/200 = 0.2	4/4 = 1
 David	37/63 = 0.58	50/200 = 0.25	2/4 = 0.5

Contoh Ketiga



Gambar 2. 4. Ilustrasi 2.4

Dari contoh di atas ada 24 data X1 dan X2 dan Kelas yang di cari adalah Y (positif atau negatif)

Data baru X1 = 4; X2 = 7 dengan K = 8. Dari hasil perhitungan dapat ditarik kesimpulan data X1 = 4 dan X2 = 7 termasuk dalam kelas NEGATIF.

MODUL III

3.1 Teknik Clustering

Clustering sebenarnya merupakan teknik data mining yang digunakan untuk mencari pengelompokan data, yang tidak memiliki pengelompokkan alami. Pada algoritma clustering, data akan dikelompokkan menjadi cluster-cluster berdasarkan kemiripan satu data dengan data yang lain. Dalam hal ini tidak ada patokan tertentu yang digunakan pada algoritma clustering untuk mencari pengelompokkan data yang ada pada sekumpulan data tersebut. Data yang dikelompokkan dalam satu cluster memiliki similaritas tinggi, sedangkan antara satu cluster dengan cluster lainnya memiliki semilaritas rendah.

Konsep terpenting yang harus disadari adalah bahwa proses clustering yang baik akan menghasilkan cluster dengan kualitas tinggi bila memiliki:

- Tingkat kesamaan yang tinggi dalam class (*high intra-class similarity*)
- Tingkat kesamaan yang rendah antar class (*low inter-class similarity*)

Similarity yang dimaksud merupkan pengukuran secara numerik terhadap dua buah objek. Nilai similarity ini akan semakin tinggi bila dua objek yang dibandingkan tersebut memiliki kemiripan yang tinggi pula. Tentunya, perbedaan kualitas suatu hasil clustering ini bergantung pada metode yang dipakai untuk mengukur kesamaan (similaritas) tersebut serta implementasinya.

Selain itu pula, suatu metode clustering juga harus dapat diukur kemampuannya dalam usahanya untuk menemukan suatu pola tersembunyi pada data yang tersedia. Dalam mengukur nilai sililarity ini, ada beberapa metode yang dapat dipakai salah satunya adalah *Euclediean Distence.* Pada metode ini, dua buah point dapat dihitung jaraknya bila diketahui nilai dari masing-masing atribut pada kedua point tersebut. Berikut adalah rumus *distence* yang dipakai, yaitu:

$$d = \sqrt{\sum_{i=1}^{p} (x_{2i} - x_{1i})^2}$$

Dengan :

x1 = sampel data

x2 = data uji

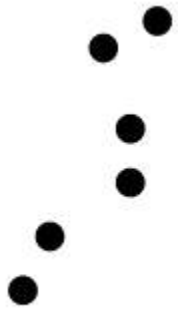
i = varibel data

d = jarak

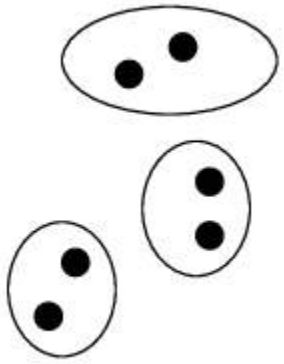
p = dimensi data

Yang dimaksud dengan bobot field (μ_k) adalah ukuran kemampuan suatu field ke-k dalam mempengaruhi jarak antara kedua point. Semakin besar nilai μ_k, akan semakin besar pula pengaruhnya terhadap jarak antara kedua point, dan sebaliknya semakin kecil nilai μ_k, akan semakin kecil pengaruhnya terhadap jarak antara ke dua point.

3.2 Tipe Clustering



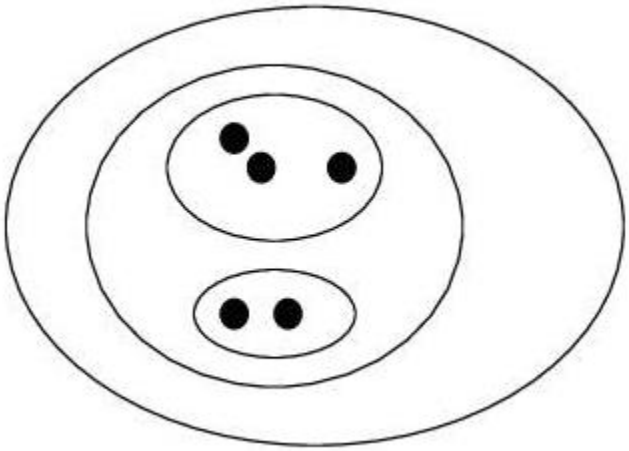
Gambar 3. 1. Ilustrasi 3.1



Gambar 3. 2. Ilustrasi 3.2

Pada dasarnya terdapat 2 tipe clustering, yaitu:

- Partitional Clustering : Tipe cluster yang benar-benar terpisah antara sekelompok objek dengan sekelompok objek lainnya.
- Hierarchical clustering : Sekelompok cluster yang terorganisasi sebagai suatu pohon hirarki (*hierarchical tree*)



Gambar 3. 3. Ilustrasi 3.3

Dalam perkembangannya, terdapat berbagai tipe cluster, yaitu:

1. Well-Separate Clusters
   Cluster adalah sekelompok point dimana tiap point dalam cluster memiliki kesamaan yang lebih (*more similar*) dengan setiap point lainnya di dalam cluster daripada tiap

point yang tidak berada dalam cluster tersebut. Dapat dikatakan bahwa setiap point yang berada dalam satu cluster akan memiliki jarak yang lebih dekat dibandingkan point-point pada cluster lain.

2. Center-besed
   Cluster adalah sekumpulan objek dimana tiap objek pada cluster memiliki kemiripan yang lebih dengan pusat (*center*) cluster lainnya.
   Pusat (*center*) dari cluster disebut dengan centroid, rata-rata dari tiap point pada cluster atau medoid merupakan point yang dapat mewakili point-point lain dari cluster tersebut (*Representative Point*)
3. Contigous Cluster (Nearest Neighbor or Transitive)
   Cluster adalah sekumpulan poin dimana tiap point dalam cluster memiliki kesamaan yang lebih (*more similar*) dengan satu point atau lebih lainnya di dalam cluster daripada tiap point yang tidak berada dalam cluster tersebut
4. Density-based
   Cluster adalah suatu are populasi point yang memisahkan antar tingkat populasi point rendah dengan tingkat populasi point yang tinggi

3.3 Penggunaan Aplikasi Clustering

Berbagai macam aplikasi penggunaan clustering, dapat meliputi sebagai berikut:

- Pengenalan pola
- Analisa data spasial (spatial data)
  Membuat Map GIS (*Geogrphic Information System*)
  Mendeteksi cluster spasial dan menjelaskannya pada data mining spasial
- Memproses image tertentu
- Ilmu Pengetahuan Ekonomi (analisa pasar)

Contoh penggunaan aplikasi cluster:

- Marketing : Membantu para pelaku pasar menemukan kelompok tertentu pada basis customer mereka dan menggunakan pengetahuan tersebut untuk mengembangkan program terget marketing mereka
- Land use : Mengidentifikasi setiap area yang ada di permukaan bumi untuk keperluan obesrvasi pada database.

- Insurance : Mengidentifikasi sekelompok pemegang polis asuransi yang memiliki tingkat biaya klaim rata-rata tertentu.
- City Planning : Mengidentifikasi sekelompok rumah berdasar tipe, nilai serta letak geografinya
- Earth-quake Studies : mengobservasi berbagai macam titik episentrum gempa bumi yang terjadi pada berbagai benua.

Salah satu Density-based clustring methods adalah metoda DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

3.4 DBSCAN (Density-Based Spatial Clustering of Applications With Noise)

DBSCAN adalah salah satu algoritma *clustering density-based*, algoritma ini pertama kali dikenalkan oleh Ester, dkk [Ester 1996] di dalam Adriano Moreira, dkk University of Minho - Portugal. Dalam satu daerah dengan kepadatan titik yang tinggi menggambarkan keberadaan cluster sedangkan daerah dengan kepadatan titik rendah disebut dengan *Noise.* Algoritma ini terutama cocok untuk menangani data yang besar, dengan noise, dan mampu mengidentifikasi cluster dengan berbagai ukuran dan bentuk.

3.4.1 Ide Utama Dari Algoritma DBSCAN

Untuk setiap titik cluster dari lingkungan radius harus berisi setidaknya jumlah minimum poin yang telah ditentukan

Algoritma ini membutuhkan 2 masukan parameter:

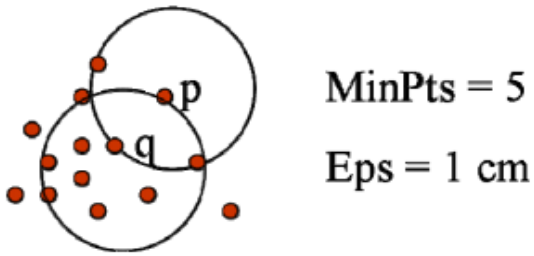
- Eps, jari-jari yang menentukan batas daerah sekitar titik (Epsneighbourhood);
- MinPts, jumlah minimum titik yang harus ada di lingkungan-Eps.

Ide dasar dari *density-based clustering* berkaitan dengan beberapa definisi baru

1. Neighborhood dengan radius Eps dari suatu obyek disebut Eps-neighborhood dari suatu obyek tersebut.
2. Jika Eps-neighborhood dari suatu obyek mengandung titik sekurang-kurangnya jumlah minimum, MinPts, maka suatu obyek tersebut dinamakan *core object*

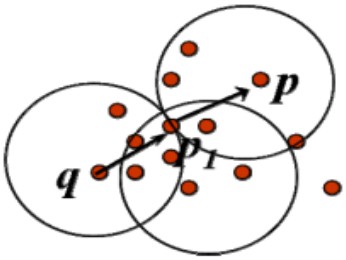
3. Diberikan set obyek D, obyek p dikatakan *directly density-reachable* (kepadatan terjangkau langsung) dari obyek q jika p termasuk dalam Eps-neighborhood dari q dan q adalah *core objek*.

Ilustrasi *directly density-reachable* (kepadatan terjangkau langsung)



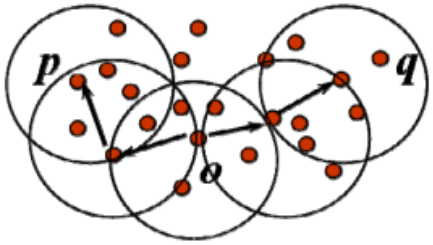
Gambar 3. 4. Ilustrasi 3.4

4. Sebuah obyek p adalah *density-reachable* dari obyek q dengan memperhatikan Eps dan MinPts dalam suatu set objek , D, jika terdapat serangkaian obyek p1,…,pn, p1=q dan pn=p dimana pi+1 adalah directly *density-reachable* dari pi dengan memperhatikan Eps dan MinPts, untuk 1 <= i <= n, pi elemen D. Konsep density-reachable di-ilustrasikan pada Gambar dibawah ini :



Gambar 3. 5. Ilustrasi 3.5

5. Sebuah obyek p adalah *density-connected* terhadap obyek q dengan memperhatikan Eps dan MinPts dalam set obyek D, jika ada sebuah obyek o elemen D sehingga p dan q keduanya density-reachable dari o dengan memperhatikan Eps dan MinPts. Gambar di bawah ini merupakan ilustrasi dari konsep *density-connected*.



Gambar 3. 6. Ilustrasi 3.6

3.4.2 Algoritma DBSCAN

- Arbitrary select a point p (memilih titik p)
- Retrieve all points density-reachable from p wrt Eps and MinPts. (Ambil semua point yang *density reachable* terhadap titik p)
- If p is a core point, a cluster is formed. (jika p adalah core point, maka cluster terbentuk)
- If p is a border point, no points are *density-reachable* from p and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed (lanjutkan proses sampai semua poin terproses)

Istilah-Istilah Yang Digunakan Dalam DBSCAN

Sebelum mendalami contoh, beberapa rangkuman istilah sebagai berikut:

- Eps-neighborhood = neighborhood dengan radius eps
- Core-Objek = Eps-neighborhood yang mengandung point >= mintpts
- Directly density-reachable = titik p dikatakan directly density-reachable dari titik q jika titik p adalah eps-neighborhood dari q dan q adalah core objek. p dan q adalah titik pusat.

  Contoh :

  Eps = 4

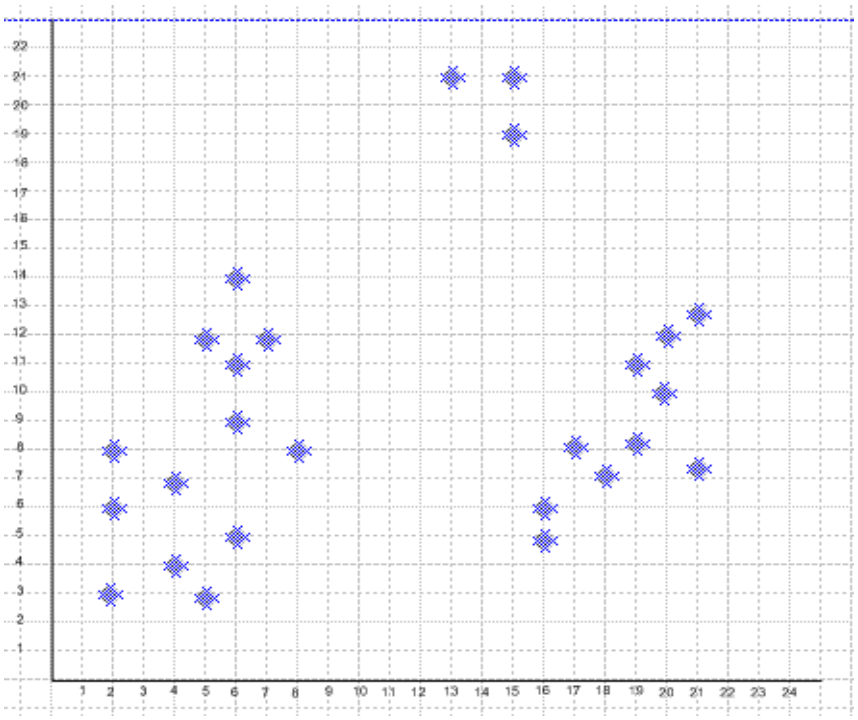
  MinPts = 4

Tabel node dengan titik (x,y)

Tabel 3. 1. Tabel Ilustrasi 3.1

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Node (x,y) dari table diatas direfresentasikan dengan gambar dibawah ini :



Gambar 3. 7. Ilustrasi 3.7

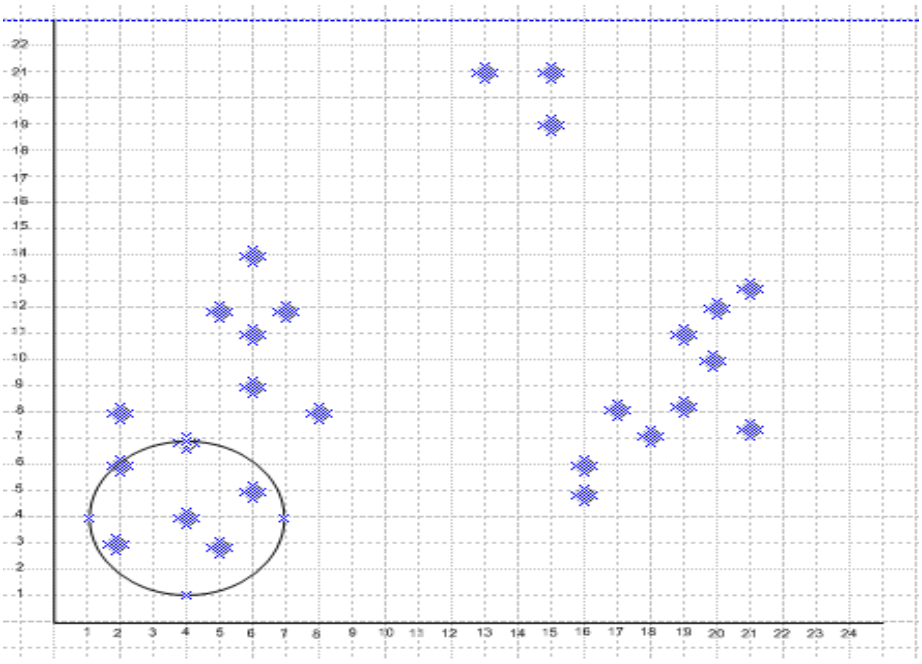
Iterasi 1

Dengan titik tengah (4,4) maka point yang menjadi anggota adalah 5 point yaitu B, C, D, F, G. Untuk lebih jelasnya dapat dilihat pada tabel berikut :

Tabel 3. 2. Tabel Ilustrasi 3.2

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah = (4,4)



Gambar 3. 8. Ilustrasi 3.8

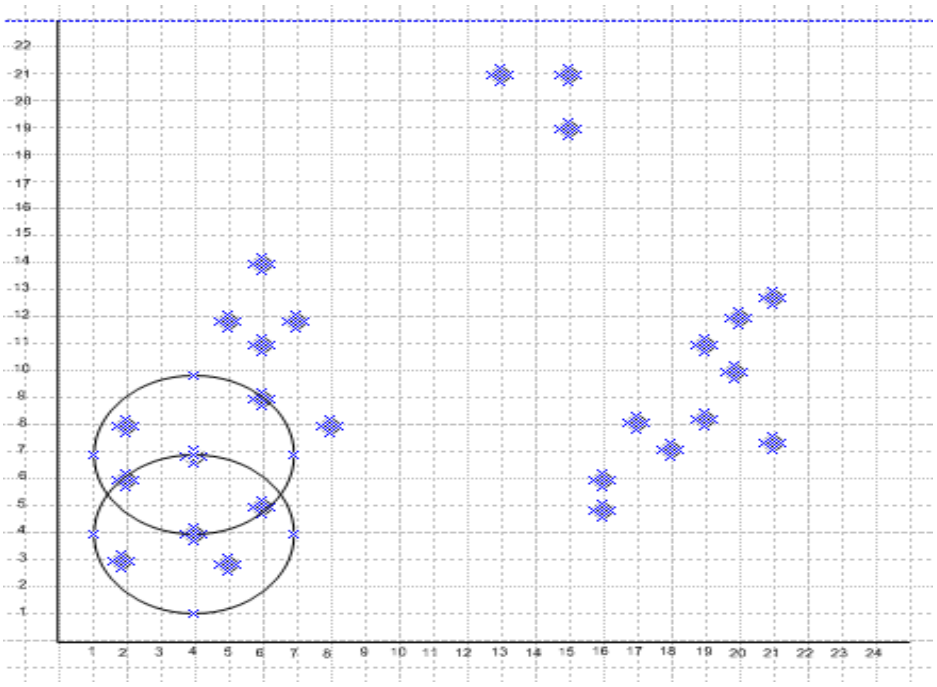
Iterasi 2

Dengan titik tengah (4,7) maka point yang menjadi anggota adalah 4 point yaitu A, D, F, K. Untuk lebih jelasnya dapat dilihat pada tabael berikut :

Tabel 3. 3. Tabel Ilustrasi 3.3

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah = (4,7)



Gambar 3. 9. Ilustrasi 3.9

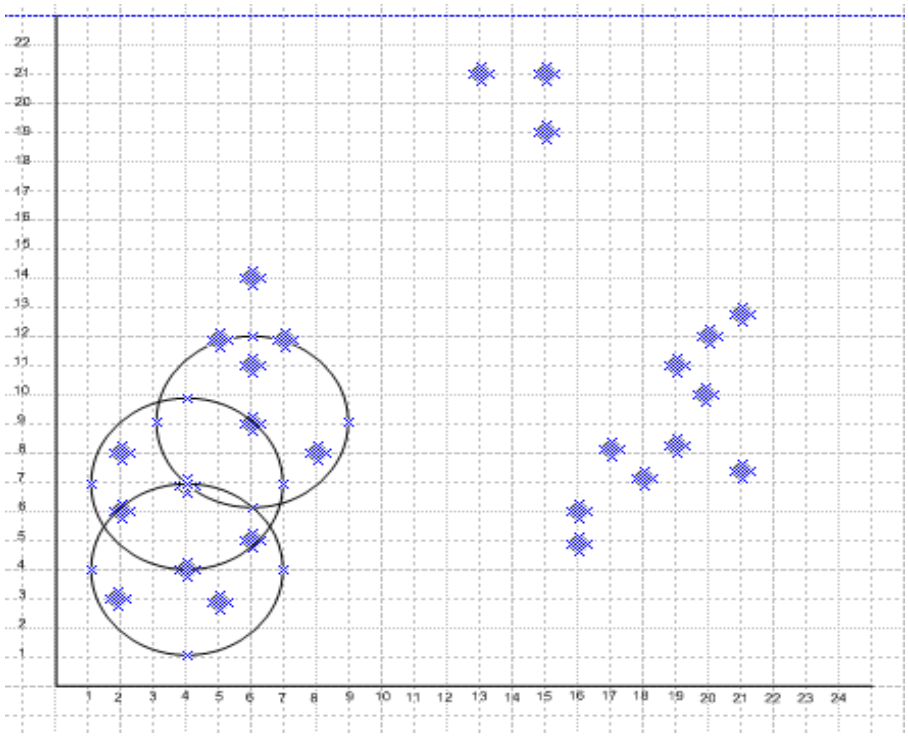
Iterasi 3

Dengan titik tengah (6,9) maka point yang menjadi anggota adalah 6 point yaitu D, G, L, Q, S dan T. Untuk lebih jelasnya dapat dilihat pada tabael berikut:

Tabel 3. 4. Tabel Ilustrasi 3.4

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah (6,9)



Gambar 3. 10. Ilustrasi 3.10

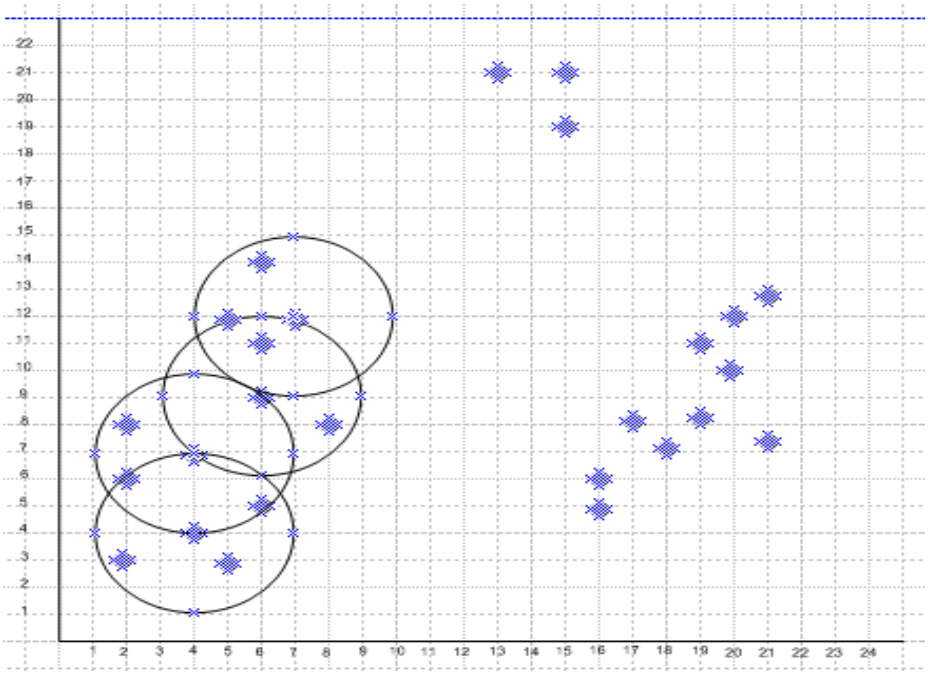
Iterasi 4

Dengan titik tengah (7,12) maka point yang menjadi anggota adalah 7 point yaitu O, P, Q, S dan W. Untuk lebih jelasnya dapat dilihat pada tabel berikut :

Tabel 3. 5. Tabel Ilustrasi 3.5

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah (7,12)



Gambar 3. 11. Ilustrasi 3.11

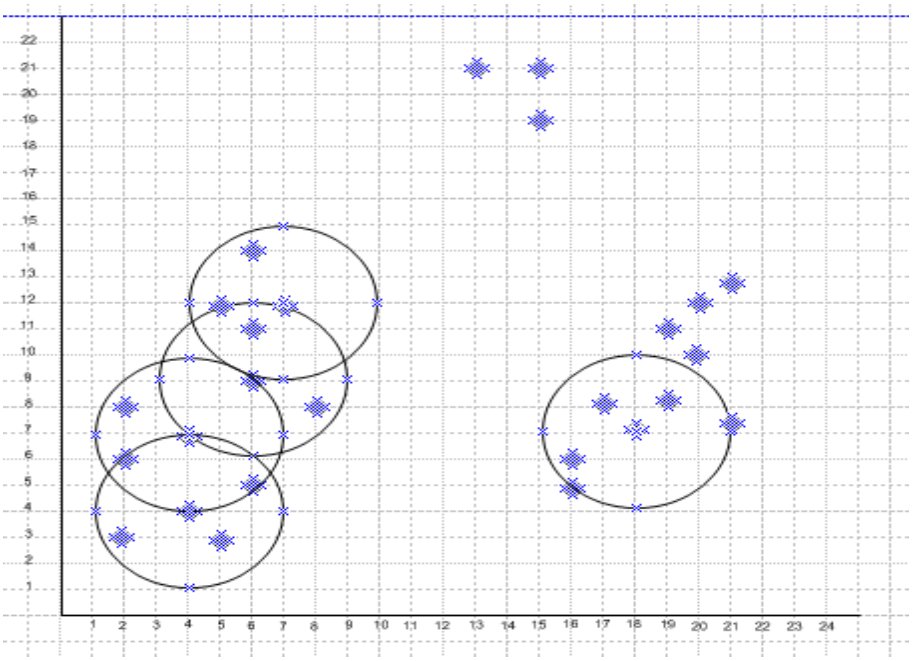
Iterasi 5

Dengan titik tengah (16,7) maka point yang menjadi anggota adalah 5 point yaitu D, I, M dan N. Untuk lebih jelasnya dapat dilihat pada tabel berikut :

Tabel 3. 6. Tabel Ilustrasi 3.6

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (12,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah (16,7)



Gambar 3. 12. Ilustrasi 3.12

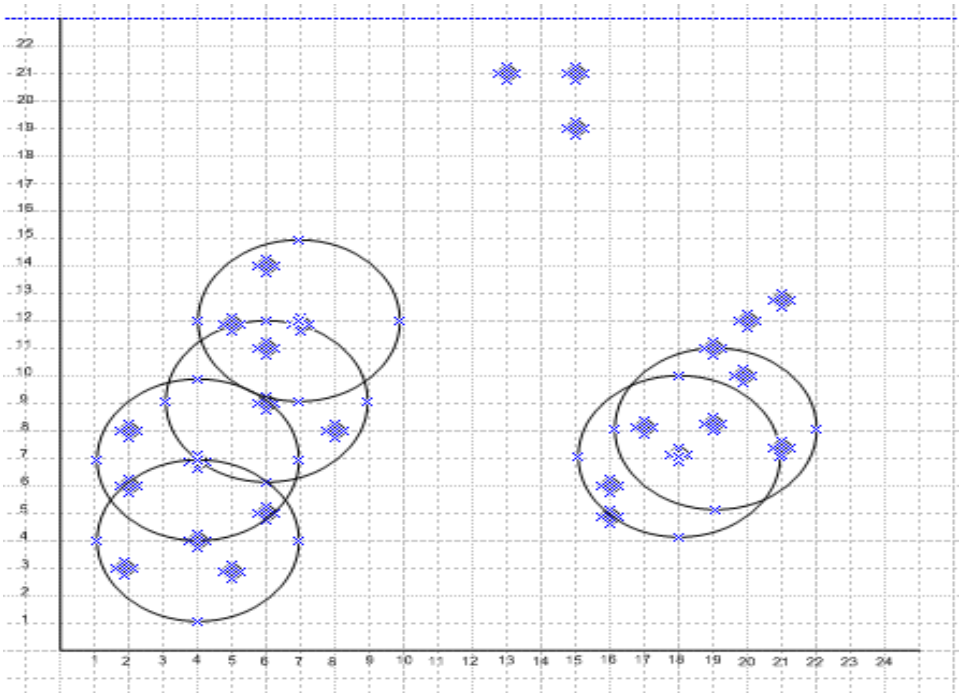
Iterasi 6

Dengan titik tengah (19,8) maka point yang menjadi anggota adalah 5 point yaitu H,I, J, M, P dan R. Untuk lebih jelasnya dapat dilihat pada tabael berikut :

Tabel 3. 7. Tabel Ilustrasi 3.7

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah (19,8)



Gambar 3. 13. Ilustrasi 3.13

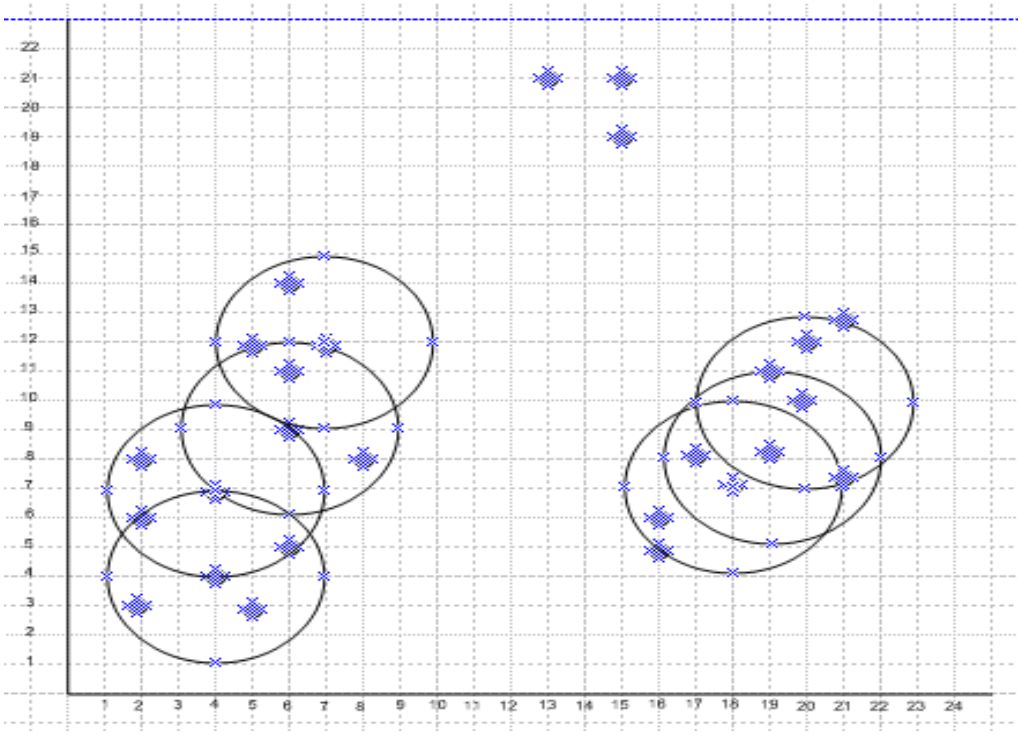
Iterasi 7

Dengan titik tengah (20,10) maka point yang menjadi anggota adalah 4 point yaitu I, J, M, N. Untuk lebih jelasnya dapat dilihat pada tabael berikut :

Tabel 3. 8. Tabel Ilustrasi 3.8

A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah (20,10)



Gambar 3. 14. Ilustrasi 3.14

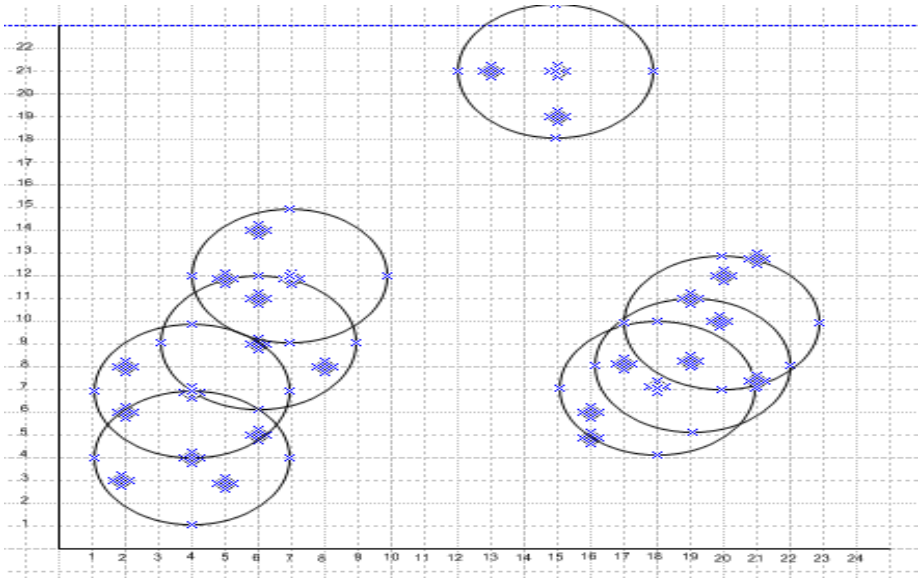
Iterasi 8

Dengan titik tengah (15,21)maka point yang menjadi anggota adalah 2 point yaitu. Untuk lebih jelasnya dapat dilihat pada tabael berikut :

Tabel 3. 9. Tabel Ilustrasi 3.9

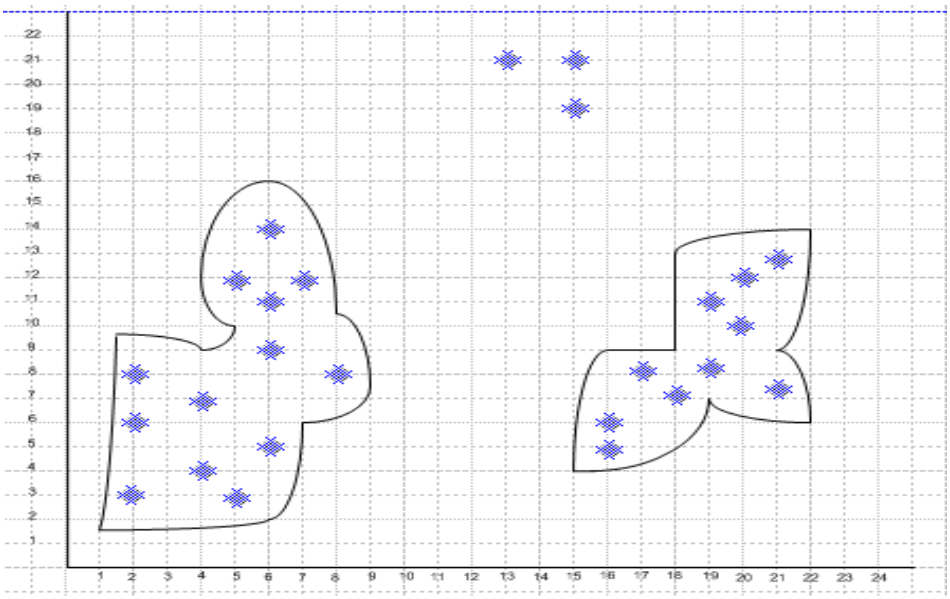
A (4,4)	H (16,7)	O (6,9)	V(21,13)
B (2,3)	I (18,7)	P (20,10)	W(6,14)
C (5,3)	J (121,7)	Q (6,11)	X(15,19)
D (6,5)	K (2,8)	R (19,11)	Y(13,21)
E (16,5)	L (8,8)	S (5,12)	Z(15,21)
F(2,6)	M (17,8)	T (7,12)	
G (4,7)	N (19,8)	U (20,12)	

Titik Tengah (15,12)



Gambar 3. 15. Ilustrasi 3.15

Hasil



Gambar 3. 16. Ilustrasi 3.16

3.5 K-MEANS

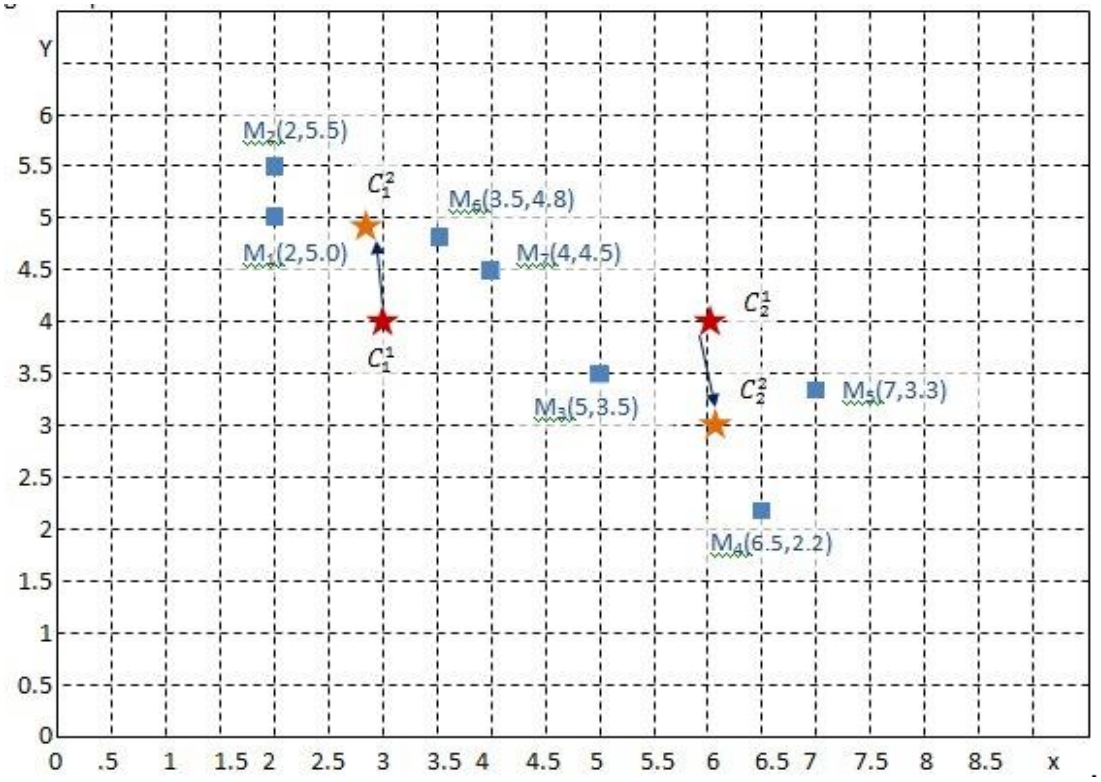
K-Means Termasuk partitioning clustering yang memisahkan data ke k daerah bagian yang terpisah. K-means algorithm sangat terkenal karena kemudahan dan kemampuannya untuk mengklaster data besar dan data outlier dengan sangat cepat. Sesuai dengan karakteristik partitioning clustering, Setiap data harus termasuk ke cluster tertentu, dan Memungkinkan bagi setiap data yang termasuk cluster tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke cluster yang lain.

3.5.1 Algoritma K-Means :

- Menentukan k sebagai jumlah cluster yang ingin dibentuk
- Membangkitkan k centroids (titik pusat cluster) awal secara random
- Menghitung jarak setiap data ke masing-masing centroids Setiap data memilih centroids yang terdekat
- Menentukan posisi centroids baru dengan cara menghitung nilai rata-rata dari data-data yang memilih pada centroidyangsama.
- Kembali ke langkah 3 jika posisi centroids baru dengan centroids lama tidak sama.

Contoh Kasus K-Means :

Using K-means algorithm find the best groupings and means of two clusters of the 2D data below. Show all your work, assumptions, and regulations. M1 = (2, 5.0), M2 = (2, 5.5), M3 = (5, 3.5), M4 = (6.5, 2.2), M5 = (7, 3.3), M6 = (3.5, 4.8), M7 = (4, 4.5)



Gambar 3. 17. Ilustrasi 3.17

Asumsi:

Semua data akan dikelompokkan ke dalam dua kelas

Center points of both clusters are $C_1(3,4)$, $C_2(6,4)$

M1 = (2, 5.0), M2 = (2, 5.5), M3 = (5, 3.5), M4 = (6.5, 2.2), M5 = (7, 3.3), M6 = (3.5, 4.8), M7 = (4, 4.5)

Iterasi 1

a. Menghitung Euclidean distance dari semua date ke tiap titik pusat pertama

$$D_{11} = \sqrt{(M_{1x} - C_{1x})^2 + (M_{1y} - C_{1y})^2} = \sqrt{(2-3)^2 + (5-4)^2} = \sqrt{2} = 1.41$$
$$D_{12} = \sqrt{(M_{2x} - C_{1x})^2 + (M_{2y} - C_{1y})^2} = \sqrt{(2-3)^2 + (5.5-4)^2} = \sqrt{3.25} = 1.80$$
$$D_{13} = \sqrt{(M_{3x} - C_{1x})^2 + (M_{3y} - C_{1y})^2} = \sqrt{(5-3)^2 + (3.5-4)^2} = \sqrt{4.25} = 2.06$$
$$D_{14} = \sqrt{(M_{4x} - C_{1x})^2 + (M_{1y} - C_{1y})^2} = \sqrt{(6.5-3)^2 + (2.2-4)^2} = \sqrt{2} = 3.94$$
$$D_{15} = \sqrt{(M_{5x} - C_{1x})^2 + (M_{5y} - C_{1y})^2} = \sqrt{(7-3)^2 + (3.3-4)^2} = \sqrt{2} = 4.06$$
$$D_{16} = \sqrt{(M_{6x} - C_{1x})^2 + (M_{6y} - C_{1y})^2} = \sqrt{(3.5-3)^2 + (4.8-4)^2} = \sqrt{2} = 0.94$$
$$D_{17} = \sqrt{(M_{7x} - C_{1x})^2 + (M_{7y} - C_{1y})^2} = \sqrt{(4-3)^2 + (4.5-4)^2} = \sqrt{2} = 1.12$$

Dengan cara yang sama hitung jara tiap titik ke titik pusat ke dan kita akan mendapatkan D_{21}= 4.12, D_{22}=4.27, D_{23}= 1.18, D_{24}= 1.86, D_{25}=1.22, D_{26}=2.62, D_{27}=2.06

b. Dari penghitungan Euclidean distance, kita dapat membandingkan:

	M_1	M_2	M_3	M_4	M_5	M_6	M_7
jarak ke							
C_1	**1.41**	**1.80**	2.06	3.94	4.06	**0.94**	**1.12**
C_2	4.12	4.27	**1.18**	**1.86**	**1.22**	2.62	2.06

➔ {M_1, M_2, M_6, M_7} anggota C_1 and {M_3, M_4, M_5} anggota C_2

c. Hitung titik pusat baru

$$C_1 = \left(\frac{2+2+3+4}{4}, \frac{5+5.5+4.8+4.5}{4}\right) = (2.75, 4.9)$$
$$C_2 = \left(\frac{5+6.5+7}{3}, \frac{3.5+2.2+3.3}{3}\right) = (6.17, 3)$$

Iterasi 2

a. Hitung Euclidean distance dari tiap data ke titik pusat yang baru Dengan cara yang sama dengan iterasi pertama kita akan mendapatkan perbandingan sebagai berikut:

	M_1	M_2	M_3	M_4	M_5	M_6	M_7
Jarak							
C_1	**0.76**	**0.96**	2.65	4.62	4.54	**0.76**	**1.31**
C_2	4.62	4.86	**1.27**	**0.86**	**0.88**	3.22	2.63

b. Dari perbandingan tersebut kira tahu bahwa {M_1, M_2, M_6, M_7} anggota C_1 dan {M_3, M_4, M_5} anggota C_2

c. Karena anggota kelompok tidak ada yang berubah maka titik pusat pun tidak akan berubah.

Kesimpulan

M_1, M_2, M_6, M_7} anggota C_1 dan {M_3, M_4, M_5} anggota C_2

3.6 E-Mediods

Algoritma k-medoids adalah algoritma clustering yang berkaitan dengan algoritma k-means dan algoritma medoidshift. Baik k-means dan algoritma k-medoids adalah teknik partisi (memecah dataset ke dalam kelompok) dan keduanya berusaha untuk meminimalkan square error (jarak antara titik berlabel berada dalam cluster dan titik yang ditunjuk sebagai pusat cluster tersebut). Berbeda dengan algoritma k-means, k-medoids memilih data points sebagai pusat (medoids atau eksemplar).

k-medoid adalah teknik partisi klasik untuk *clustering* yang melakukan klasterisasi data dari n objek ke dalam k cluster yang dikenal dengan *a priori*. Sebuah alat yang berguna untuk menentukan k adalah *silhouette*.

k-medoid lebih kuat terhadap *noise* dan *outliner* dibandingkan dengan k-means karena meminimalkan jumlah dari ketidaksamaan bukannya meminimalkan jumlah kuadrat jarak Euclidean.

medoid dapat didefinisikan sebagai objek cluster, yang rata-rata perbedaan untuk semua objek dalam suatu cluster minimal yaitu merupakan titik paling pusat dari data yang diberikan.

Realisasi yang paling umum dari clustering k-medoid adalah Partition Around Medoids (PAM) dan algoritma adalah sebagai berikut :

- Inisialisasi : pilih secara acak k dari n data point sebagai medoids
- Asosiasikan setiap data point ke medoid yang terdekat (terdekat berarti menggunakan perhitungan jarak yang biasa digunakan adalah Euclidean distance, Manhattan distance atau Minkowski distance)
- Untuk setiap medoid m
    - Untuk setiap data non medoid o
    - Tukarkan m and o dan hitung berapa total *cost* dari setiap konfigurasi (penukaran m dan o)
- Pilih konfigurasi dengan *cost* paling sedikit
- Ulangi langkah 2 sampai 5 dan hentikan jika sudah tidak terdapat perubahan medoids.

Contoh PAM

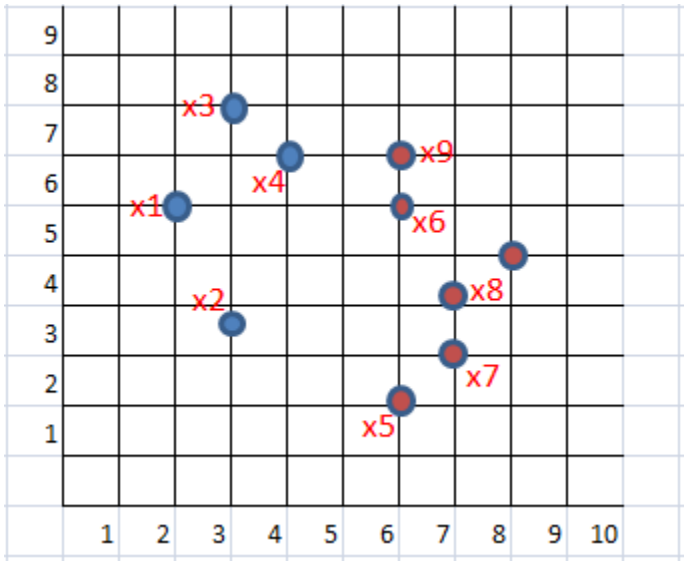
Klasterisasi data set yang terdiri dari sepuluh objek menjadi 2 cluster, anggaplah data set nya adalah dalam 2 dimensi sebagai berikut :

Tabel 3. 10. Tabel Ilustrasi 3.10

X_1	**2**	**6**
X_2	**3**	**4**
X_3	**3**	**8**
X_4	**4**	**7**
X_5	**6**	**2**
X_6	**6**	**6**

X_7	7	3
X_8	7	4
X_9	8	5
X_{10}	7	6

Berikut adalah gambar dataset yang akan diklasterisasi



Gambar 3. 18. Ilustrasi 3.18

Langkah Pertama

- Inisialisasi pusat *k*

  Mari kita asumsikan c_1 = (3,4) dan c_2 = (7,4)

  Jadi kita sekarang memiliki c_1 dan c_2 sebagai medoid, berikutnya kalkulasi jarak menggunakan rumus jarak minkowski dengan p = 1

$$\sum_{i=1}^{n}(|x_i - y_i|^p)^{\frac{1}{p}}$$

Untuk point pertama data objek (2,6) sedangkan data pusat adalah (3,4) jadi total jarak yang didapat adalah

jarak = $(|3-2|^1)^{1/1} + (|4-6|^1)^{1/1}$

jarak = (1+2) ; jarak = 3

Dan untuk titik-titik yang lain maka akan mendapatkan jarak sebagai berikut

Tabel 3. 11. Tabel Ilustrasi 3.11

c_1		Data objects (X_i)		Cost (distance)
3	4	2	6	3
3	4	3	8	4
3	4	4	7	4
3	4	6	2	5
3	4	6	6	5
3	4	7	3	5
3	4	8	5	6
3	4	7	6	6

Tabel 3. 12. Tabel Ilustrasi 3.12

c_2		Data objects (X_i)		Cost (distance)
7	4	2	6	7
7	4	3	8	8
7	4	4	7	6
7	4	6	2	3
7	4	6	4	1
7	4	7	3	1
7	4	8	5	2
7	4	7	6	2

Tugasi setiap objek ke objek perwakilan terdekat Menggunakan L1 Metric (Manhattan), kami membentuk Cluster berikut:

Sehingga sekarang cluster menjadi

Cluster1 = {(3,4)(2,6)(3,8)(4,7)}

Cluster2 = {(7,4)(6,2)(6,4)(7,3)(8,5)(7,6)}

Cost dari dua titik dapat dicari dengan rumus :

$$cost(x, c) = \sum_{i=1}^{d} |x - c|$$

Dimana

x = adalah data objek

c = adalah medoid

d = adalah dimensi dari objek (sekarang menggunakan dimensi 2)

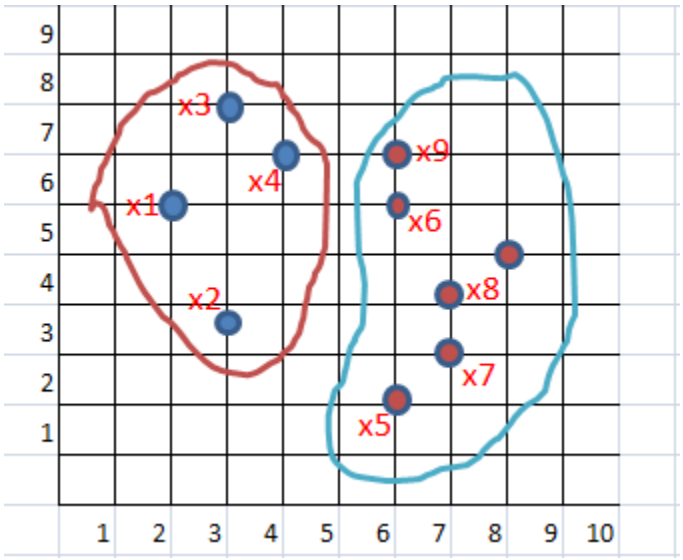
Total cost adalah perhitungan total dari setiap cost yang dimiliki dalam sebuah cluster dimana untuk c_1 dan c_2 total cost adalah

Total cost = {cost((3,4),(2,6)) + cost((3,4),(3,8)) + cost((3,4),(4,7))} + {cost((7,4),(6,2)) + cost((7,4),(6,4)) + cost((7,4),(7,3)) + cost((7,4),(8,5)) + cost((7,4),(7,6))}

= (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2)

= 20

Berikut adalah hasil klasterisasi sementara dari langkah pertama



Gambar 3. 19. Ilustrasi 3.19

Langkah Kedua

Pemilihan nonmedoid O' secara acak

Mari kita asumsikan O'= (7,3) (ada di c_2 maka kita akan tukar c_2 dengan O') Maka sekarang medoids yang ada adalah c1 (3,4) dan O'(7,3)

Jika c1 dan O' adalah medoids baru, hitung biaya total yang terlibat

Dengan menggunakan rumus pada langkah 1 maka didapatkan

Tabel 3. 13. Tabel Ilustrasi 3.1

c_1		Data objects (X_i)		Cost (distance)
3	4	2	6	3
3	4	3	8	4
3	4	4	7	4
3	4	6	2	5
3	4	6	4	3
3	4	7	4	4

3	4	8	5	6
3	4	7	6	6
O′		Data objects (X_i)		Cost (distance)
7	3	2	6	8
7	3	3	8	9
7	3	4	7	7
7	3	6	2	2
7	3	6	4	2
7	3	7	4	1
7	3	8	5	3
7	3	7	6	3

Hitung total *cost* yang terjadi setalah memindahkan medoid dari c_2 ke O’

Dengan menggunakan rumus yang sama dengan langkah pertama maka didapatkat

Total cost’ = 3 + 4 + 4 + 2 + 2 + 1 + 3 + 3

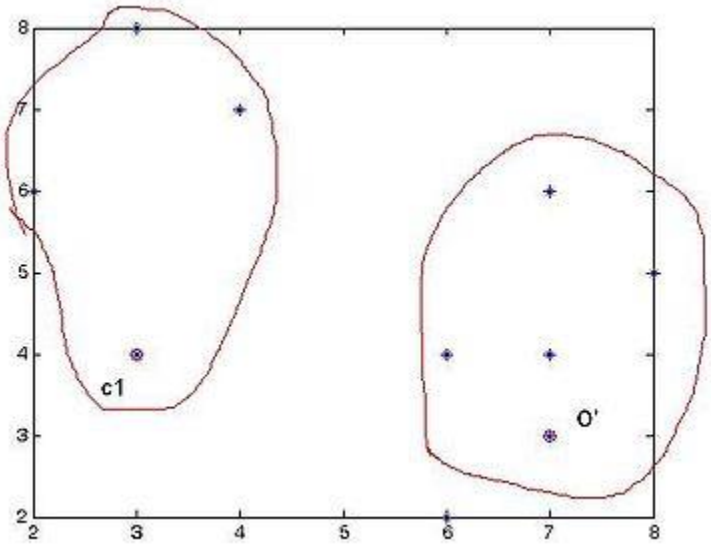
Total cost’ = 22

Dan bandingkanlah total cost dengan total cost’

S = total cost’ – total cost

S = 22 – 20 = 2

Karena 2 > 0 maka didapatkan kesimpulan bahwa medoid yang baru tidak lebih baik dari medoid yang lama, berikut adalah hasil klasterisasi setelah langkah kedua dilakukan



Gambar 3. 20. Ilustrasi 3.20

Lakukanlah langkah 1 dan 2 terus menerus (mengulang langkah 2 sampai 5 pada k-medoid) sampai akhirnya mendapatkan medoid yang paling bagus (medoid tidak dirubah lagi) dan algoritma ini pun akan dihentikan.

MODUL IV

4.1 Tools Data Mining

Weka merupakan sebuah *tools data mining opensource* yang berbasis java dan berisi kumpulan algoritma *machine learning* dan data *pre-processing*. *Waikato Environment for Knowledge Analysis* adalah kepanjangan dari Weka yang dibuat di Universitas Waikato, New Zealand yang digunakan untuk pendidikan, penelitian, dan aplikasi eksperimen *data mining*. Weka mampu menyelesaikan masalah-masalah *data mining* yang ada di dunia, khususnya klasifikasi yang mendasari pendekatan *machine learning*. Weka dapat digunakan dan diterapkan pada beberapa tingkatan yang berbeda. Weka menyediakan pengimplementasian algoritma pembelajaran *state of the art* yang dapat diterapkan pada *dataset* dari *command line*. Dalam Weka terdapat *tools* untuk *preprocessing data*, *clustering*, aturan asosiasi, klasifikasi, regresi, dan visualisasi. Adanya *tools* yang sudah tersedia pengguna dapat melakukan *preprocess* pada data, kemudian memasukkannya dalam sebuah skema pembelajaran, dan menganalisis *classifier* yang dihasilkan serta performanya. Semua itu dilakukan tanpa menulis kode program sama sekali. Contoh penggunaan weka adalah menerapkan sebuah metode pembelajaran ke *dataset* dan menganalisis hasilnya untuk memperoleh informasi tentang data, atau menerapkan beberapa metode dan membandingkan performa untuk dipilih.

4.2 Isntalasi Weka

Instalasi weka pada buku ini menggunakan versi 3.9.2. Langkah Langkah instalasi weka, yaitu sebagai berikut.

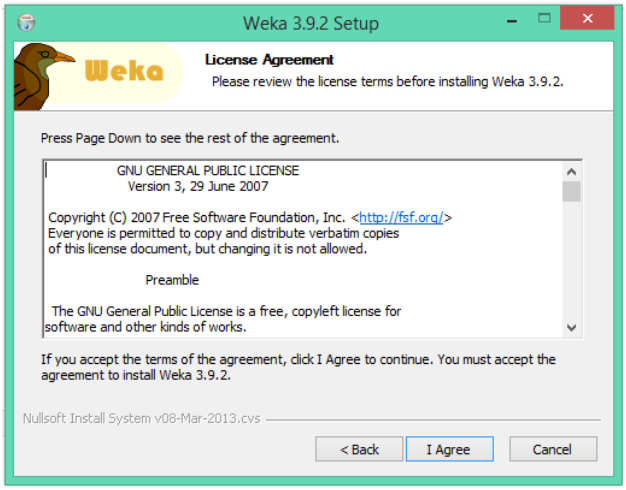
1. Klik dua kali file *executable* dari weka 3.9.2 (.exe) seperti pada Gambar 4.1

Gambar 4. 1. Ilustrasi 4.1

2. Kemudian akan muncul jendela seperti pada Gambar 4.2, selanjutnya pilih *next.*

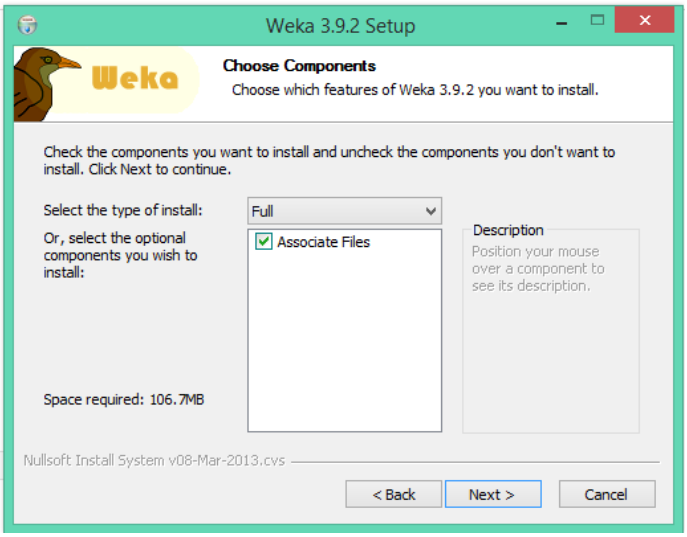
Gambar 4. 2. Ilustrasi 4.2

3. Kemudian akan muncul jendela *Licene Agreement* seperti pada Gambar 4.3, lalu pilih *I Agree.*



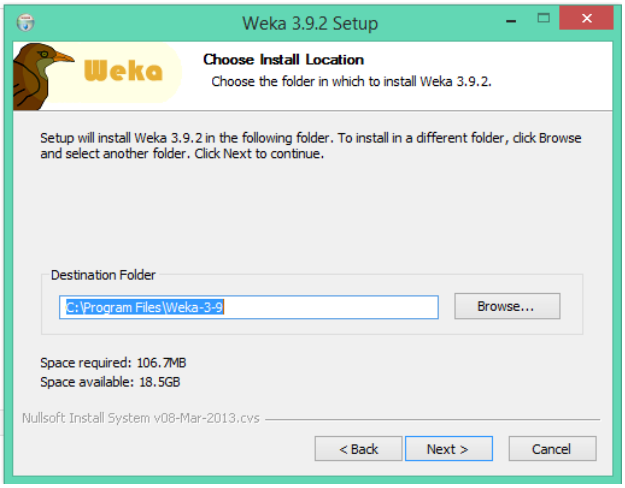
Gambar 4. 3. Ilustrasi 4.3

Selanjutnya akan muncul jendela seperti pada Gambar 4.4, pada bagian *select type of install* pilih *full*, untuk menginstall seluruh komponen yang diperlukan untuk menjalankan apilkasi, kemudian pilih *next.*



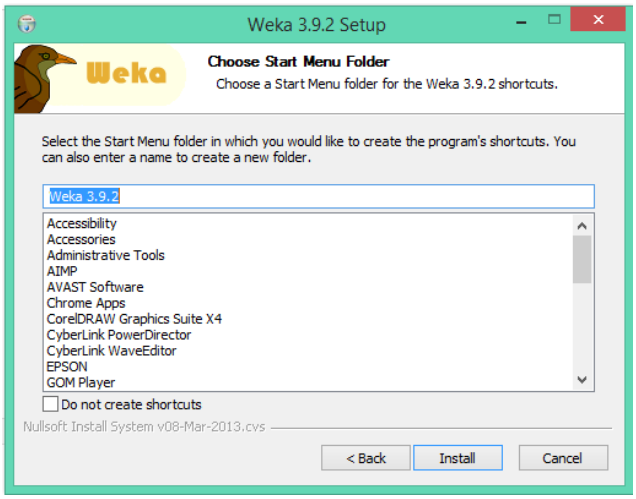
Gambar 4. 4. Ilustrasi 4.4

4. Selanjutnya seperti pada Gambar 4.5, tentukan dimana ingin menyimpan *file* hasil proses instalasinya, setelah selesai menentukan direktori kemudian pilih *next*



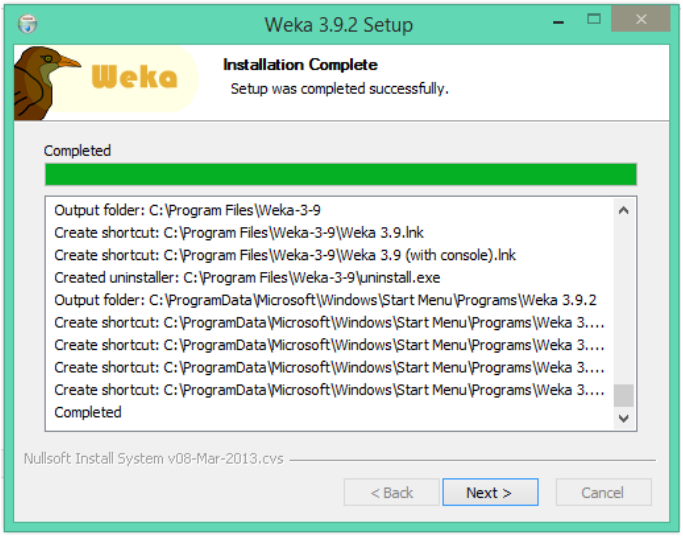
Gambar 4. 5. Ilustrasi 4.5

5. Kemudian tentukan apakah ingin membuat *shortcut* untuk menjalankan aplikasinya pada *start* menu atau tidak dan tentukan nama dari *shortcut*nya, selanjutnya pilih *install* seperti pada Gambar 4.6 berikut ini.



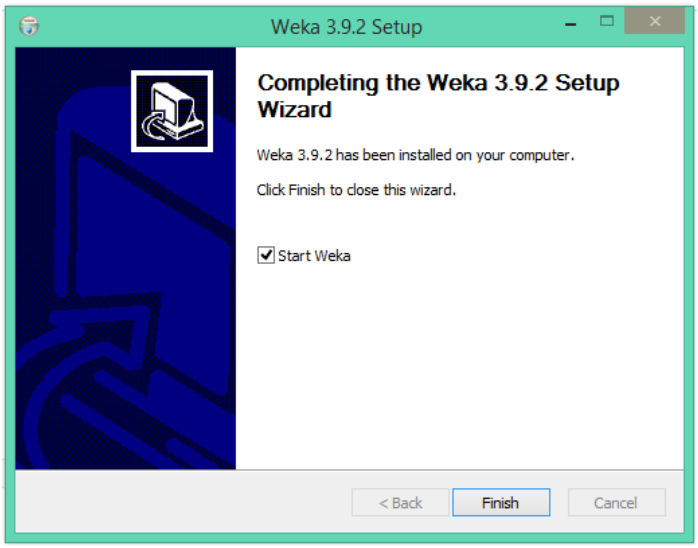
Gambar 4. 6. Ilustrasi 4.6

6. Maka proses instalasi akan dilakukan. Setelah proses instalasi selesai kemudian klik *next* seperti pada Gambar 4.7.



Gambar 4. 7. Ilustrasi 4.7

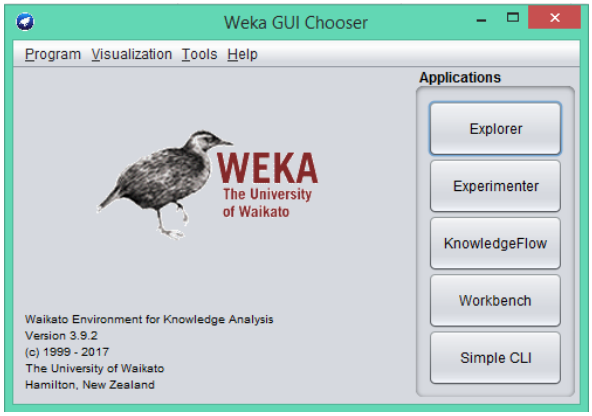
7. Selanjutnya klik *finish*, apabila ingin menjalankan aplikasi weka maka tandai pilih *Start Weka* seperti yang ditunjukkan pada Gambar 4.8.



Gambar 4. 8. Ilustrasi 4.8

4.2.1 Menjalankan Weka

Setelah proses instalasi selesai, pada subbab ini akan membahas tentang GUI Weka**.** Gambar 4.9 merupakan halaman awal dari weka 3.9.2.



Gambar 4. 9. Ilustrasi 4.9

Tampilan awal ketika apliaksi weka dijalankan maka terlihat seperti pada Gambar 4.9, pada tampilan awal weka terdapat empat menu utama diantaranya *program, visualisation, tools,* dan *help* serta lima tombol diantaranya *explorer, experimenter, knowledgeflow, woekbench,* dan *simple CLI.*

1. Program

   Menu program mempunyai empat sub menu, diantaranya:

a. *LogWindow* (*Shortcut* CTRL+L)

   Sub menu *LogWindow* digunakan untuk menampilkan log yang merekap semua yang tercetak untuk stdout dan stderr.

b. *Memory usage* (*Shortcut* CTRL+M)

   Sub *menu memory usage* digunakan untuk menampilkan penggunaan memori pada saat aplikasi weka digunakan.

c. Setting

   Sub menu *setting* digunkaan untuk mengatur tampilan pada *user interface.*

d. *Exit* (*Shortcut* CTRL+E)

   Submenu *exit* digunakan untuk keluar dari aplikasi weka.

2. Visualisation

Menu *visualisation* merupakan sarana yang digunakan untuk memvisualisasikan data dengan aplikasi weka. Menu ini mempunyai lima submenu, diantaranya:

a. *Plot* (*Shortcut* CTRL+P)

   Sub menu plot digunakan untuk menampilkan plot 2D dari sebuah *dataset.*

b. ROC (*Shortcut* CTRL+R)

   Sub menu ROC digunakan untuk menampilkan kurva ROC yang telah disimpan sebelumnya.

c. *TreeVisualizer* (*Shortcut* CTRL+T)

   Sub menu *TreeVisualizer* digunakan untuk menampilkan graf berarah, contohnya: *decision tree.*

d. *Graph Visualizer* (*Shortcut* CTRL+G)

   Sub menu *graph visualizer* digunakan untuk memvisualisasikan format grafik XML, BIF, atau DOT, contohnay sebuah jaringan bayesian.

e. *Boundary Visualizer* (Shortcut CTRL+B)

   Sub menu *boundary visualizer* bertugas untuk mengizinkan visualisasi dari batas keputusan *classifier* dalam plot 2D.

3. Tools

Menu *tools* menampilkan aplikasi lainnya yang berguna bagi pengguna. Pada menu ini terdapat empat sub menu, yaitu:

a. *Package Manager* (*Shortcut* CTRL+U)

b. *ArffViewer* (*Shortcut* CTRL+A)

Sebuah aplikasi MDI yang menampilkan *file* Arff dalam format *spreadsheet.*

c. *SqlViewer* (*Shortcut* CTRL+S)

Merepresentasikan sebuah lembar kerja SQL, untuk melakukan *query database* via JDBC.

d. *Bayes net editor* (*Shortcut* CTRL+N)

Sebuah apliksi untuk mengedit, memvisualisasikan dan mempelajari *bayes net.*

4. Help

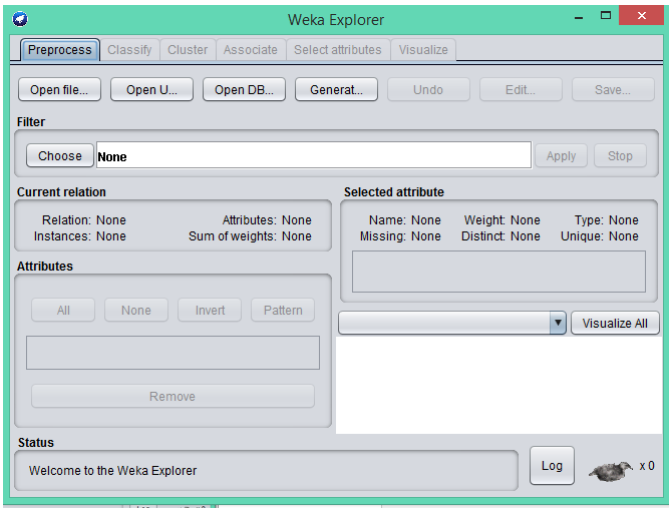
Menu *help* mempunyai empat sub menu, diantaranya:

1. *Weka Homepage* (*Shortcut* CTRL+H)
2. *HOWTOs, code snippets, etc.* (*Shortcut* CTRL+W)
3. *Weka on sourcefouge* (*Shortcut* CTRL+F)
4. *System info* (*Shortcut* CTRL+I)

Tools yang dapat digunakan untuk *preprocessing dataset* membuat pengguna dapat berfokus pada algoritma yang digunakan tanpa terlalu memperhatikan detail seperti pembacaan data dari *file-file*, penyediaan kode untuk evaluasi hasil, dan implementasi algoritma *filtering*. Untuk melakukan pengujian maka perlu memahami beberapa tombol GUI yang ada di weka, diantaranya:

a. *GUI Explorer*

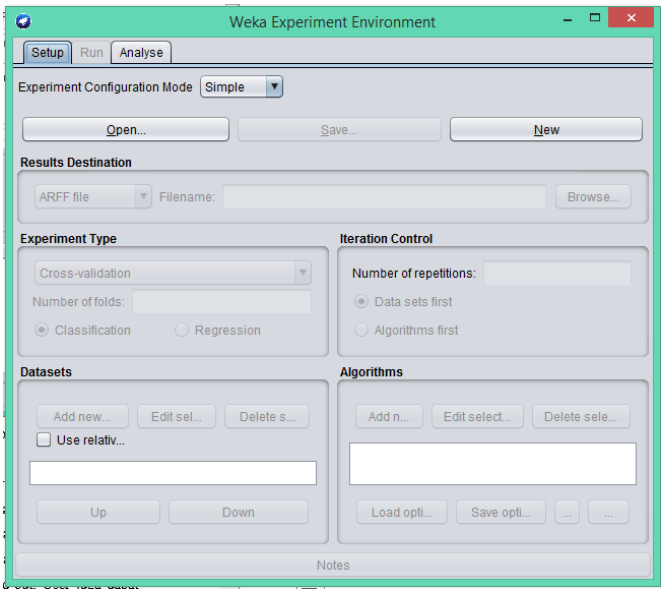
GUI Explorer merupakan GUI yang menyediakan semua fitur weka dalam bentuk tombol dan tampilan visualisasi yang menarik dan lengkap. *GUI Explorer* adalah GUI yang paling mudah digunakan. *Preprocess*, asosiasi,klasifikasi, *clustering, select atribut,* dan *visualize* dapat dilakukan dengan mudah. Tampilan *weka expoler* dapat dilihat pada Gambar 4.10.



Gambar 4. 10. Ilustrasi 4.10

b. *GUI Experimenter*

GUI experimenter biasanya digunakan untuk klasifikasi dan regresi. *GUI experimenter* dapat memudahkan perbandingan performansi skema-skema pembelajaran yang berbeda. Hasil dari perbandingan tersebut dapat dituliskan dalam *database* atau file. Dalam Weka tersedia pilihan evaluasi yaitu *learning curve*, *cross-validation*, *hold-out*. Pengguna juga dapat melakukan iterasi menurut beberapa *setting parameter* yang berbeda. *GUI experimenter* dapat dilihat pada Gambar 4.11.



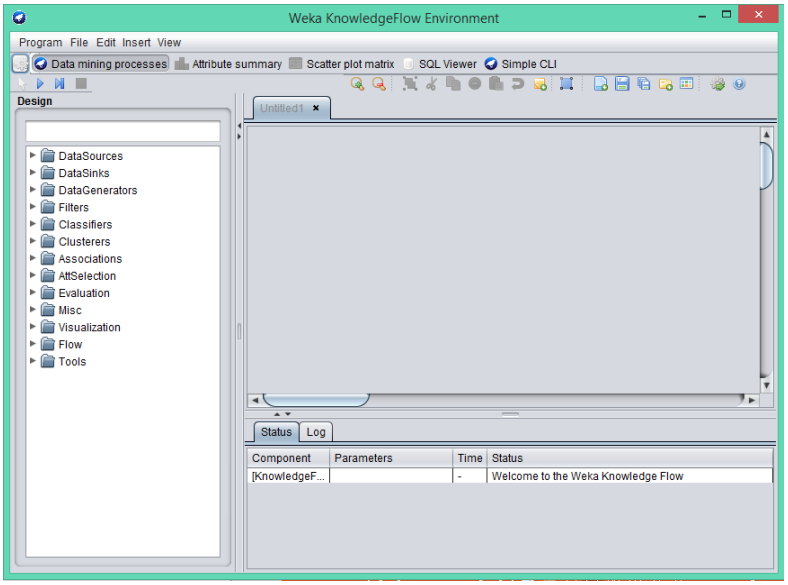
Gambar 4. 11. Ilustrasi 4.11

Tab setup yang muncul saat dibuka *experimenter* memungkinkan *user* memilih dan mengkonfigurasi eksperimen yang dilakukan. Setelah menyimpan definisi eksperimen yang dilakukan pengguna dapat memulai eksperimen dari *tab run* dan mengklik tombol *start*. Hasilnya akan disimpan dalam format CSV dan dapat dibuka dalam bentuk *spreadsheet*.

Tab Analize, dapat digunakan untuk menganalisa hasil eksperimen yang dikirim ke weka, Jumlah baris hasil ditunjukkan pada *panel source*. Hasilnya dapat di-*load* dalam format .arff maupun dari basis data

c. *GUI Knowledge Flow*

GUI knowledge flow merupakan GUI yang ada dalam weka yang merupakan antarmuka *Java-Beans-Based* untuk melakukan *setting* dan menjalankan percobaan-percobaan *macine learning*. Tampilan GUI *Weka knoledge flow* dapat dilihat pada Gambar 4.12..



Gambar 4. 12. Ilustrasi 4.12

Knowledge flow dapat menangani data secara *incremental* maupun dalam *batches* (*Explorer* hanya menangani data *batch*). Tentunya diperlukan sebuah *classifier* yang dapat *diupdate instance per instance* untuk pembelajaran dari data secara *incremental*

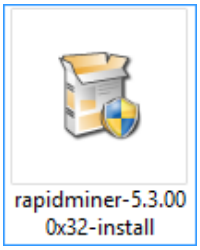
4.3 Rapid Miner

RapidMiner merupakan perangkat lunak yang bersifat *open source* untuk melakukan analisis *data mining, text mining*, dan analisis prediksi. Operator data mining yang terdapat pada *RapidMiner* diantaranya operator untuk *input, output*, visualisasi, dan data *preprocessing*. *RapidMiner* ditulis menggunakan bahasa java sehingga bisa bekerja di semua sistem operasi. Sebelumnya *RapidMiner* bernama *Yet Another Learning Environment (YALE)*, dimana versi awalnya 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 GNU *Affero General Public License (AGPL)* versi 3. Hingga saat ini ribuan aplikasi telah dikembangkan menggunakan *RapidMiner* di lebih dari 40 negara. Sebagai *software open source* untuk *data mining, RapidMiner* tidak perlu diragukan lagi karena *software* ini sudah terkemuka di dunia. Peringkat pertama *software data mining* pada *polling* oleh KDnuggets, sebuah portal *data mining* pada 2010-2011 ditempati oleh *RapidMiner.*

4.3.1 Instalasi Rapid Miner

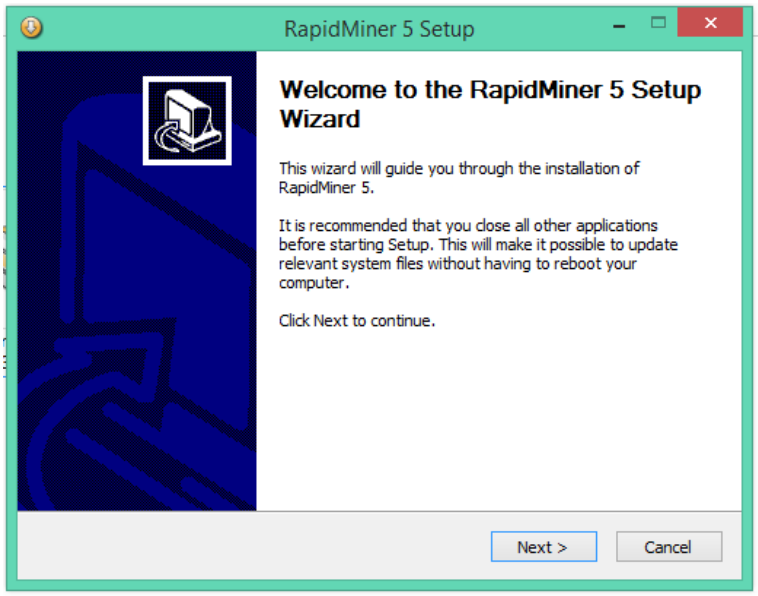
Instalasi RapidMiner pada buku ini menggunakan versi 5.3. Langkh-lanagkah instalasi rapid miner, yaitu sebagai berikut.

1. Klik dua kali file executable dari RapidMiner 5.3.000x32-install(.exe) seperti pada Gambar 4.13.



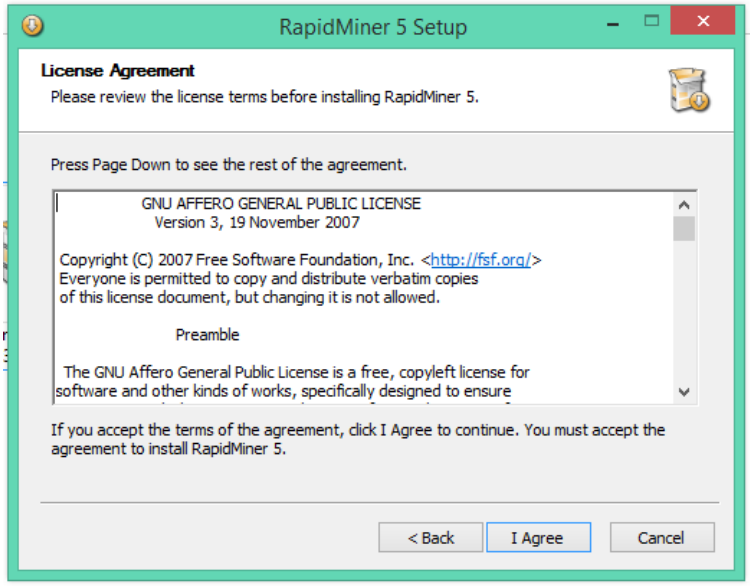
Gambar 4. 13. Ilustrasi 4.13

2. Jendela yang muncul pada Gambar 4.14, pilih next.



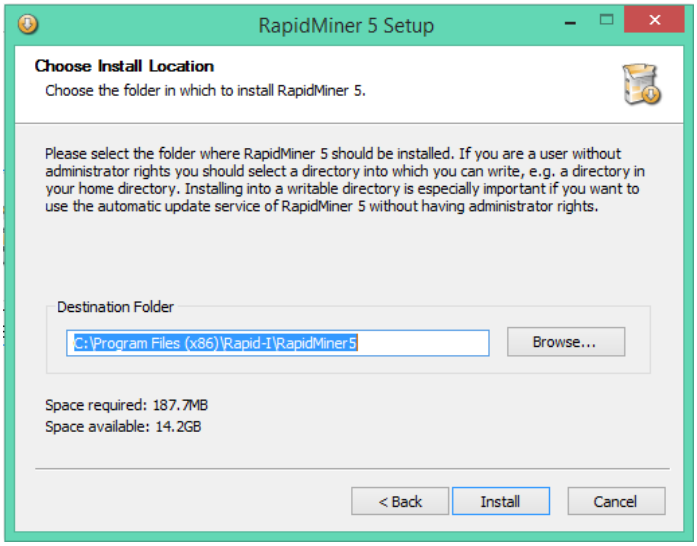
Gambar 4. 14. Ilustrasi 4.14

3. Jendela Licene Agreement pada Gambar 4.15, pilih I Agree



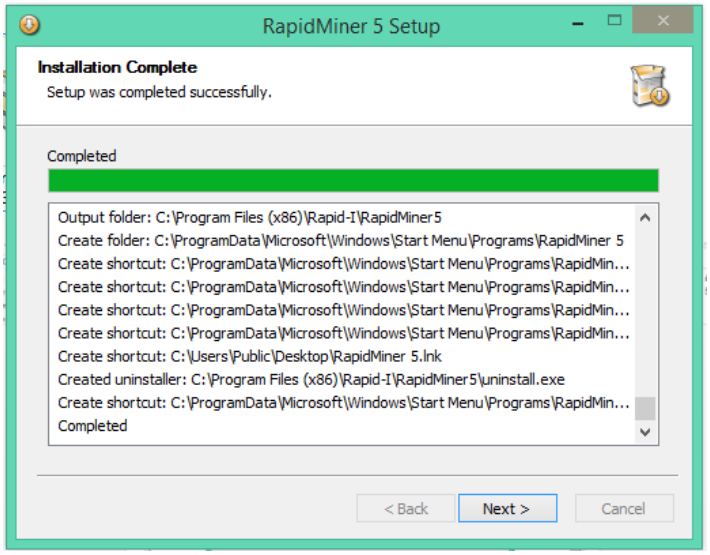
Gambar 4. 15. Ilustrasi 4.15

4. Selanjutnya akan menampilkan form seperti pada Gambar 4.16 untuk menentukan tempat penyimpanan file hasil proses instalasi, setelah selesai menentukan direktori kemudian pilih install



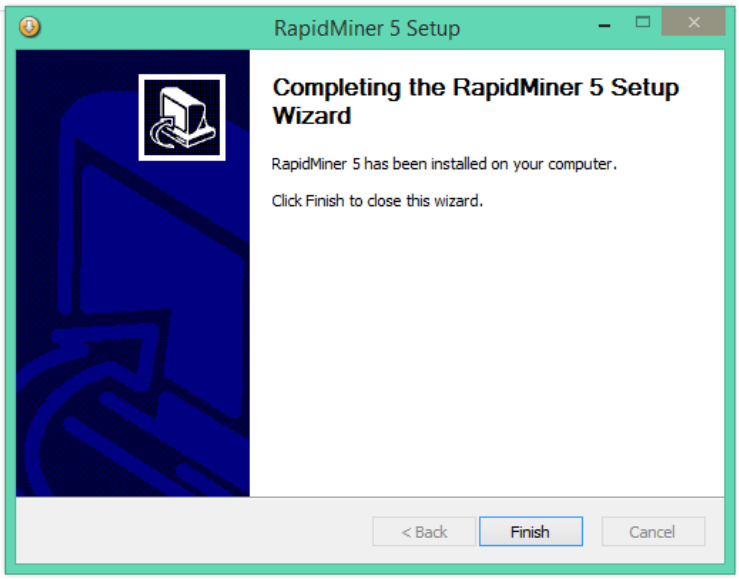
Gambar 4. 16. Ilustrasi 4.16

5. Setelah proses instalasi selesai kemudian klik next seperti pada Gambar 4.17.



Gambar 4. 17. Ilustrasi 4.17

6. Selanjutnya klik finish pada Gambar 4.18 untuk mengakhiri proses instalasi.



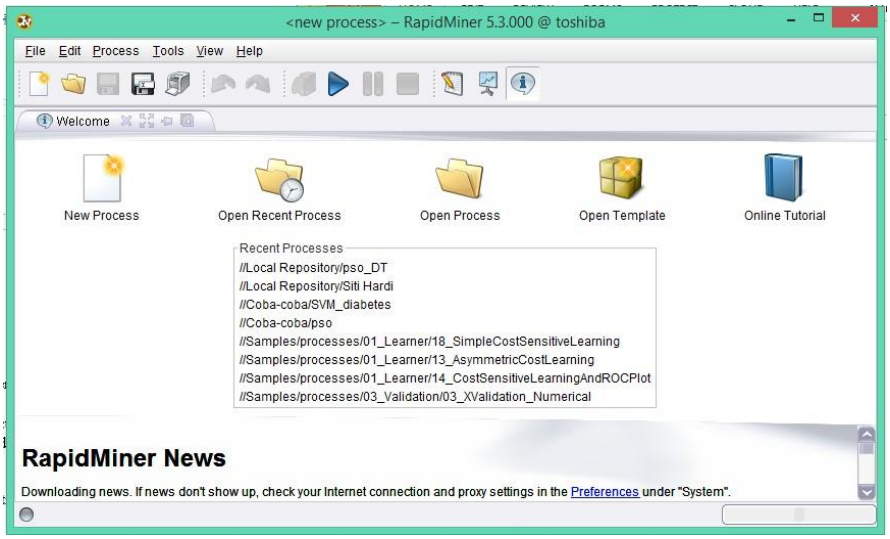
Gambar 4. 18. Ilustrasi 4.18

4.3.2 Pengenalan Interface Rapid Miner

RapidMiner menyediakan tampilan yang *user friendly* untuk memudahkan penggunanya ketika menjalankan aplikasi. Tampilan pada *RapidMiner* dikenal dengan istilah *perspective* dan terdapat 3 *perspective* dalam *RapidMiner* diantaranya *welcome perspective*, *design perspective*, dan *result perspective*.

1. Welcome Perspective

   Ketika membuka aplikasi maka akan tampil seperti pada Gambar 4.19.



Gambar 4. 19. Ilustrasi 4.19

Pada gambar di atas menampilkan beberapa daftar aksi, diantaranya New, Open Recent Process, Open Process, Open Template, dan Online Tutorial. Berikut ini rincian lengkap daftar aksi tersebut:

a. New

   New digunakan untuk memulai proses analisis baru. Untuk memulai proses analisis, pertama-tama harus menentukan nama dan lokasi proses serta data repository.

b. Open Recent Process

   Open recent process digunakan untuk membuka proses yang baru saja ditutup. Selain itu, dapat digunakan untuk membuka proses yang baru ditutup dengan mengeklik dua kali dari salah satu daftar yang ada pada recent process maka tampilan welcome persective akan otomatis beralih ke design perspective.

c. Open Process

   Open process digunakan untuk membuka repository browser yang berisi daftar proses.

d. Open Template

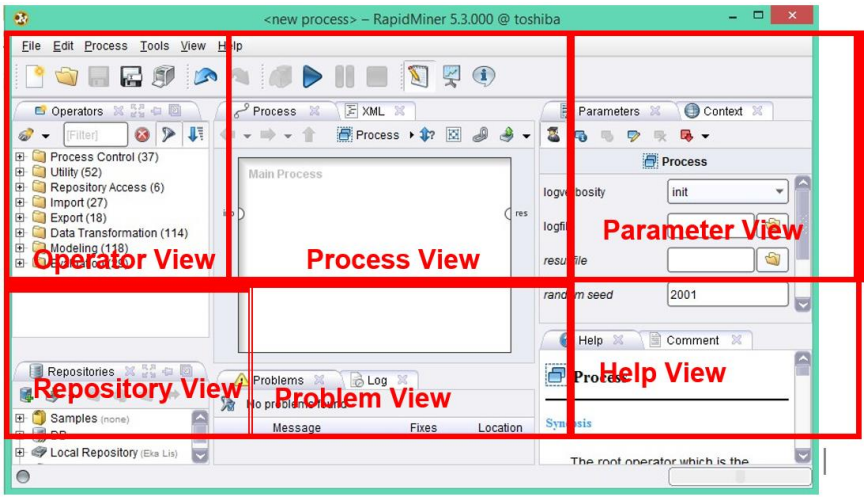
   Open template digunakan untuk menunjukkan pilihan lain yang sudah ditentukan oleh proses analisis.

e. Online Tutorial

Online tutorial digunakan untuk memulai tutorial secara online dengan catatan komputer harus terhubung dengan internet. Tutorial yang didapat secara langsung dari rapid miner berupa perkenalan tentang konsep data mining.

2. Design Perspective

Design perspective merupakan lembar kerja pada RapidMiner yang digunakan untuk membuat dan mengelola proses analisis dari konsep data mining. Seperti yang ditunjukkan pada Gambar 4.20 yang mempunyai beberapa view.



Gambar 4. 20. Ilustrasi 4.20

Berikut rincian dari beberapa view yang ditampilkan pada design perspective:

a. Operator View

Operator view merupakan salah satu view yang paling penting karena semua operator dari RapidMiner disajikan dalam bentuk hierarki sehingga operator-operator tersebut dapat digunakan pada proses analisis data mining.

b. Process View

Process view merupakan halaman yang digunakan untuk menunjukkan langkah-langkah dalam proses analisis data mining dengan menggunakan komponen-komponen yang ada pada operator view.

c. Parameter View

Beberapa operator dalam RapidMiner membutuhkan satu atau lebih parameter agar dapat didefinisikan sebagai fungsionalitas yang benar. Namun terkadang parameter tidak mutlak dibutuhkan, meskipun eksekusi operator dapat dikendalikan dengan menunjukkan nilai parameter tertentu.

d. Repository View

Repository view merupakan komponen utama dalam design perspective. Repository view digunakan untuk mengelola dan menata proses analisis data mining.

e. Problem View

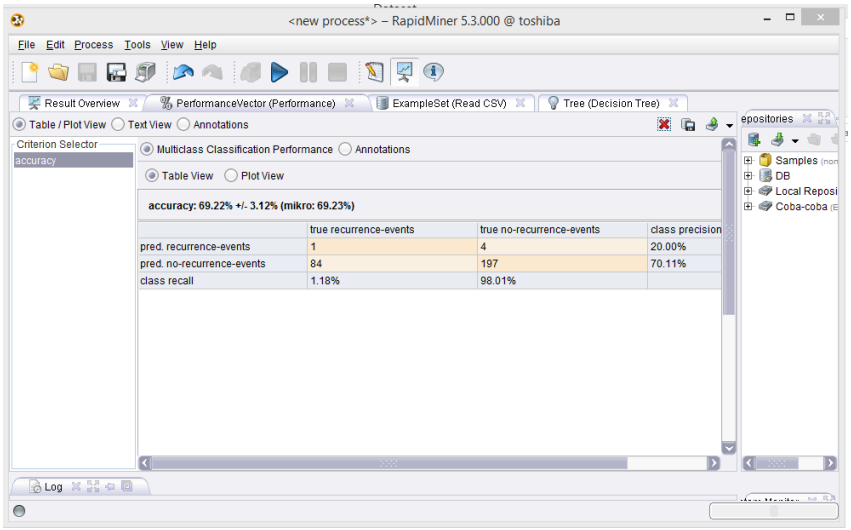
Problem view merupakan komponen yang sangat berharga dan merupkan sumber bantuan bagi pengguna selama merancang proses analisis, karena apabila ada kesalahan dalam proses analisis maka akan ada pemberitahuan pada halaman problem view.

f. Help dan Comment View

Help view digunakan untuk memberi penjelasan singkat mengenai fungsi operator dalam satu atau beberapa kalimat. Sedangkan comment view merupakan area bagi pengguna menuliskan komentar pada proses analisis data mining.

3. Result Perspective

Result perspective merupakan tampilan yang digunakan untuk menampilkan hasil dari proses analisis data mining, seperti yang ditunjukkan pada Gambar 4.21.



Gambar 4. 21. Ilustrasi 4.21

DAFTAR PUSTAKA