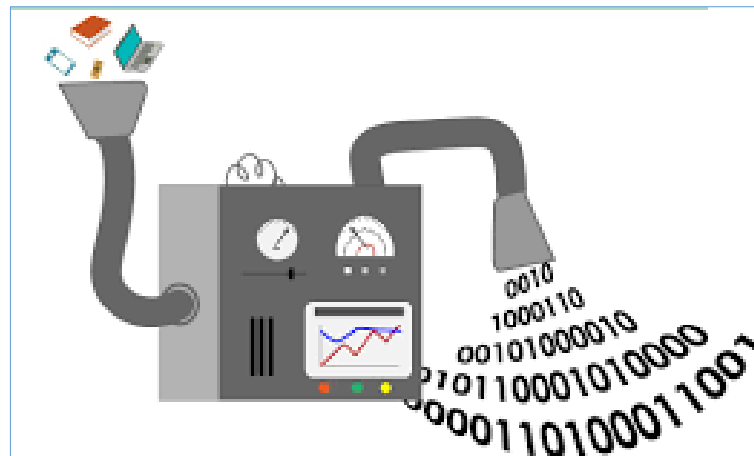


DATA MINING

Algoritme dan Implementasi Menggunakan
Bahasa Pemrograman PHP



Joko Suntoro, M.Kom.

jokosuntoro@usm.ac.id

**BUKU INI TIDAK UNTUK
DIPERJUALBELIKAN
HAK CIPTA MILIK ALLAH TA'ALA**

**Penulis,
Joko Suntoro, M.Kom.**

KATA PENGANTAR

Penulis mengucapkan Alhamdulillah Robbil'alamin kepada Allah Ta'ala karena telah diberikan kesehatan, tenaga, pikiran, dan waktu luang sehingga dapat menyelesaikan penulisan buku yang berjudul **Data Mining Algoritme dan Impelementasi Menggunakan Bahasa Pemrograman PHP**. Penulis menyadari bahwa penulisan buku ini tidak akan terwujud tanpa adanya dukungan dan bantuan dari berbagai pihak. Ucapan terima kasih penulis sampaikan kepada:

1. Ibu tercinta Solechah, dan bapak tercinta (alm.) Jamasri.
2. Istriku tercinta Rutin Fisekati, yang senantiasa mendoakan, dan mendukung penulis hingga terselesaikan penyusunan buku ini
3. Adik tercinta Riska Wibowo, beserta saudara-saudaraku.
4. Guru dan dosen yang telah membimbing saya (bu Titis Perwitasari, bu Endang Rustanti, om Luri Darmawan, pak Romi Satria Wahono, Ph.D, dan guru dan dosen lainnya).
5. Rekan kerja di Domestic Gas Region IV (pak Ringgas Hutagaol, pak W. Andi K., pak Deky Dwi Wahyu, mas Aan Chunaifi, Rendy, Furqon, mas Buyung, Illa Puspowardani, dkk lainnya), terima kasih atas semangatnya.
6. Rekan-rekan dosen di Universitas Semarang, khususnya Faktultas Teknologi Informasi dan Komunikasi (bu Dr. Titin Winarti, pak Susanto, bu Vensy Vydia, pak April Firman Daru, bu Prind Tri Ajeng, bu Nur Wakhidah, pak Bernard Very, serta rekan-rekan dosen lainnya).
7. Rekan-rekan tim penelitian Intelligent Systems Research Group (pak Dr. Adi Wijaya, pak Dr. Catur Supriyanto, Ahmad Ilham, Deden Istiawan, Rahadi Teguh, dkk lainnya), terima kasih atas diskusi hangatnya.
8. Sahabat-sahabatku: Ali Chillo, Ika Merdekawati, Erna Wahyuningsih, Fajar Sinaringtyas, Cahya Nurani Indah, Eska Sarti Kundari, dhe Surono, pak Didik Supriyadi, Mas Ochy Husain, Yudi, Ucta, Radit, Dmitri, Dwiandi, Hanif, Handini dan sahabat-sahabatku lainnya.
9. Semua pihak yang telah membantu dan tidak bisa penulis sebutkan satu-persatu, penulis ucapkan terima kasih.

Buku ini disusun secara sistematis dalam menjelaskan penghitungan manual algoritme-algoritme data mining dan disertai contoh penerapan algoritme-algoritme tersebut dalam bahasa pemrograman PHP. Algoritme yang dibahas pada buku ini khususnya adalah penerapan klasifikasi data mining, seperti: algoritme k-Nearest Neighbor, algoritme Naïve Bayes, dan algoritme Decision Trees (ID3, C4.5, dan CART).

Harapan penulis dengan buku ini dapat membantu mahasiswa, industri, dan semua pihak yang ingin belajar tentang algoritme data mining. Serta semoga buku ini bisa bermanfaat serta berkontribusi dalam ilmu, pengetahuan, dan teknologi. Kritik dan saran bisa disampaikan melalui email penulis.

Semarang, Desember 2018

Penulis,
Joko Suntoro, M.Kom.

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
Bab 1 PENGENALAN DATA MINING	6
1.1 Perkembangan Data	6
1.2 Definisi Data Mining	6
Bab 2 Instalasi PHP untuk Data Mining	9
2.1 Kebutuhan dan Persyaratan	9
2.2 Cara Penggunaan	9
Bab 3 Algoritme k-Nearest Neighbor	10
3.1 Pengenalan Algoritme k-Nearest Neighbor	10
3.2 Penghitungan Manual Algoritme k-Nearest Neighbor	11
3.3 Implementasi Algoritme k-Nearest Neighbor	13
3.4 Studi Kasus Sistem Cerdas untuk Prediksi Penyakit Kanker Payudara 14	
Bab 4 Algoritme Naïve Bayes	15
4.1 Pengenalan Algoritme Naïve Bayes	15
4.2 Penghitungan Manual Tipe Data Nominal Algoritme Naïve Bayes ..	15
4.3 Penghitungan Manual Tipe Data Numerik dan Nominal Algoritme Naïve Bayes	17
4.4 Implementasi Algoritme Naïve Bayes	20
4.5 Studi Kasus Sistem Cerdas untuk Prediksi Cacat Software	21
Bab 5 Algoritme Decision Trees	22
5.1 Pengenalan Algoritme Decision Trees	22
5.2 Penghitungan Manual Algoritme ID3	22
5.3 Penghitungan Manual Algoritme C4.5	28
5.4 Algoritme Classification and Regression Tree (CART)	35
5.5 Implementasi Algoritme Decision Trees (CART)	37
5.6 Studi Kasus Sistem Cerdas untuk Prediksi Penerimaan Karyawan	38
DAFTAR PUSTAKA	39
TENTANG PENULIS	40

DAFTAR GAMBAR

Gambar 1.1 Data Pengguna Internet di Indonesia	7
Gambar 1.2 Ilustrasi Data Mining.....	7
Gambar 2.1 Cek Versi Composer	9
Gambar 2.2 Mapping Class pada Composer	9
Gambar 3.1 Ilustrasi Algoritme k-Nearest Neighbor	10
Gambar 3.2 Algoritme k-NN dengan Nilai $k = 3$	13
Gambar 3.3 Source Code knn.php	14
Gambar 3.4 Hasil pada Browser knn.php	14
Gambar 4.1 Source Code nb-golf.php	20
Gambar 4.2 Hasil pada Browser nb-golf.php	20
Gambar 5.1 Pohon Keputusan Pembentuk Node Akar	24
Gambar 5.2 Pohon Keputusan Pembentuk Node Akar dan Node 1.1	25
Gambar 5.3 Pohon Keputusan Setelah Semua Node Terpartisi	27
Gambar 5.4 Pseudo-Code Algoritme ID3 pada Dataset Pembelian Komputer	27
Gambar 5.5 Node Akar Dataset Pengajuan Kredit.....	30
Gambar 5.6 Pembentukan Node 1.1 Dataset Pengajuan Kredit	32
Gambar 5.7 Pohon Keputusan Dataset Pengajuan Kredit	34
Gambar 5.8 Node Akar Dataset Penerimaan Karyawan	36
Gambar 5.9 Pohon Keputusan Dataset Penerimaan Karyawan.....	37
Gambar 5.10 Source Code cart.php	38
Gambar 5.11 Hasil pada Browser cart.php	38

DAFTAR TABEL

Tabel 1.1 Perbedaan Peranan Data Mining	7
Tabel 1.2 Tipe Data pada Data Mining.....	8
Tabel 3.1 Kemiripan Objek “X” dengan Singa dan Kambing	11
Tabel 3.2 Data Training	11
Tabel 3.3 Data Testing	12
Tabel 3.4 Jarak Data Training ke Data Testing	12
Tabel 3.5 Urutan Data Berdasarkan Jarak Terkecil.....	12
Tabel 3.6 Tiga Data dengan Jarak Terdekat antara Data Training dan Data Testing.....	13
Tabel 4.1 Data Training Pembelian Komputer	16
Tabel 4.2 Data Testing Pembelian Komputer.....	16
Tabel 4.3 Data Training Bermain Golf.....	18
Tabel 4.4 Data Testing Bermain Golf	18
Tabel 5.1 Hasil Penghitungan <i>Entropy</i> Semua Atribut Dataset Pembelian Komputer	23
Tabel 5.2 Data Training Bermain Golf dengan Filter Atribut Usia = tua	24
Tabel 5.3 Hasil Penghitungan <i>Entropy</i> dengan Filter Atribut Usia = Tua.....	25
Tabel 5.4 Data Training Bermain Golf dengan Filter Atribut Usia = muda	26
Tabel 5.5 Hasil Penghitungan <i>Entropy</i> dengan Filter Atribut Usia = Muda ..	26
Tabel 5.6 Dataset Pengajuan Kredit	28
Tabel 5.7 Hasil Penghitungan <i>Entropy</i> Semua Atribut Dataset Pengajuan Kredit	29
Tabel 5.8 Dataset Pengajuan Kredit dengan Filter Atribut Penghasilan = Sedang	31
Tabel 5.9 Hasil Penghitungan <i>Entropy</i> dengan Filter Atribut Penghasilan = Sedang	31
Tabel 5.10 Dataset Pengajuan Kredit dengan Filter Atribut Penghasilan = Rendah.....	33
Tabel 5.11 Hasil Penghitungan <i>Entropy</i> dengan Filter Atribut Penghasilan = Rendah.....	33
Tabel 5.12 Dataset Penerimaan Karyawan.....	35
Tabel 5.13 Hasil Penghitungan <i>IndexGini</i> Semua Atribut Penerimaan Karyawan	36

Bab 1 PENGENALAN DATA MINING

1.1 Perkembangan Data

Seiring dengan berkembangnya internet maka data-data yang tersimpan, baik dalam bentuk teks, gambar, suara dan video juga mengalami peningkatan yang sangat cepat dan signifikan. Gambar 1.1 menunjukkan bahwa di Indonesia, pengguna internet tahun 1998 hanya 500.000 pengguna sedangkan sampai dengan tahun 2015 diproyeksikan pengguna internet sudah mencapai angka 139 jutaan (<https://apjii.or.id/>, 2016).

Menurut data yang dihimpun oleh IBM (Ohlhorst, 2013), di dunia ini skala volume data yang kita buat setiap hari sebesar 2,5 exabyte ($2,5 \times 10^{18}$ byte), dimana 90% data yang berada di dunia ini, telah kita buat sejak dua tahun yang lalu. Diprediksi sampai dengan tahun 2020 skala volume data akan meningkat drastis menjadi 40 zettabyte (40×10^{21} byte).

Skala volume data yang berjumlah besar tersebut akan menjadi “sampah” pada penyimpanan apabila tidak diolah menjadi informasi yang berguna. Hal ini sesuai dengan definisi dari data, bahwa data adalah suatu fakta yang terekam, namun tidak memiliki arti (Noersasongko & Andono, 2010). Bagaimana cara untuk mengatasi “sampah” data yang tidak memiliki arti tersebut agar diolah menjadi informasi? Akan kita pelajari pada buku ini.

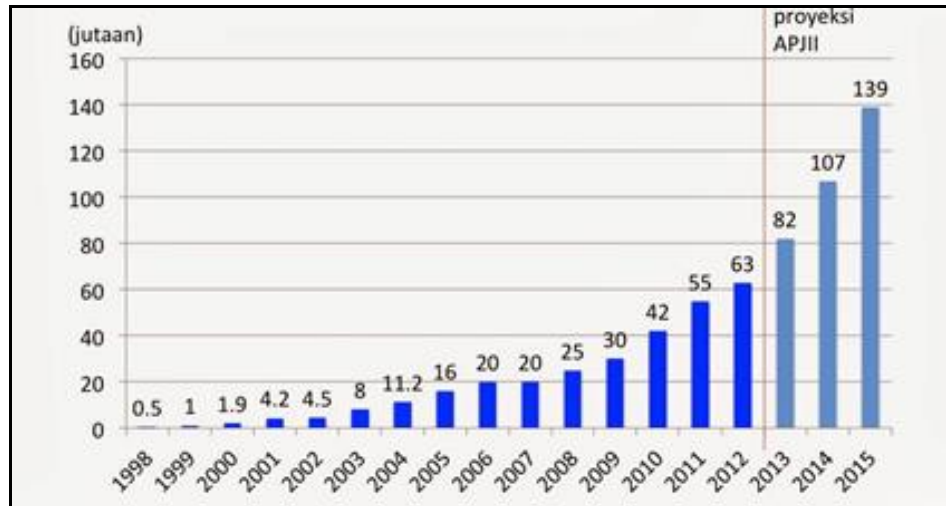
1.2 Definisi Data Mining

Menurut John Naisbitt dalam Larose (Larose, 2005) bahwa kita terbenam di dalam data, namun kita kekurangan informasi dan pengetahuan. Hal tersebut menjadi gambaran apa yang telah terjadi pada hari ini, bahwa skala volume data yang jumlahnya sangat besar tersebut hanya menjadi sampah dimemori penyimpanan saja, apabila tidak diolah menjadi informasi. Untuk mengolah data yang jumlahnya besar tersebut menjadi sebuah informasi atau pengetahuan diperlukan suatu teknik yang dinamakan dengan data mining. Definisi data mining adalah proses ekstraksi suatu data/pola (sebelumnya tidak diketahui, bersifat implisit, dianggap tidak berguna) menjadi informasi atau pengetahuan dari data yang jumlahnya besar (Witten, Ian H. Frank, 2011). Ilustrasi dari data mining dapat dilihat pada Gambar 1.2, data-data yang tidak terpola/tidak terstruktur, dianggap “sampah”, dan tidak berguna kemudian diolah (difilter) sehingga data tersebut membentuk pola/pengetahuan baru yang berguna.

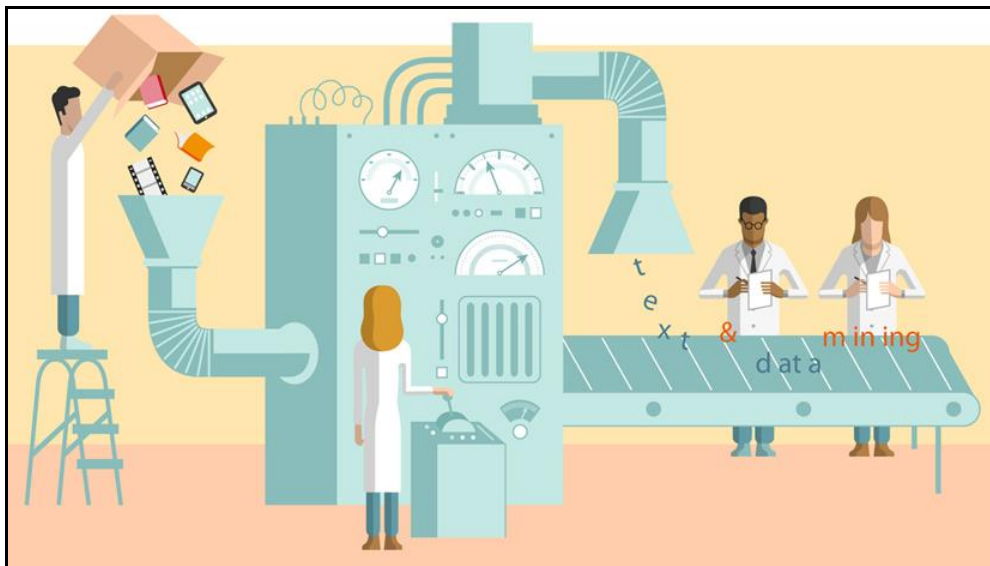
Pengolahan data menjadi informasi/pola/pengetahuan yang berguna dibutuhkan peranan data mining di dalamnya. Secara umum terdapat 5 (lima) peranan dalam data mining, yaitu estimasi, prediksi, klasifikasi, klustering, dan asosiasi. Tabel 1.1 menunjukkan perbedaan masing-masing peranan data mining. Tipe data yang digunakan pada data mining secara sederhana dibedakan menjadi 3 (tiga), yaitu tipe data numerik, tipe data kategorial, dan tipe data rentang waktu. Tipe data numerik dibagi menjadi dua bagian yaitu ratio dan interval. Tipe data kategorial juga dibagi menjadi dua bagian yaitu ordinal dan nominal. Penjelasan masing-masing tipe data dapat dilihat pada Tabel 1.2.

Proses pengolahan data dalam data mining dibutuhkan algoritme-algoritme untuk melakukan ekstraksi menjadi informasi/pola/pengetahuan. Penggunaan algoritme pada data mining diklasifikasi berdasarkan masing-masing peranan data mining. Pada peranan estimasi dan prediksi, algoritme yang banyak digunakan adalah

Linear Regression, Support Vector Machine, Neural Network, dll. Algoritme yang banyak digunakan pada peranan klasifikasi adalah k-Nearest Neighbors (k-NN), Naïve Bayes, ID3, C4.5, CART, dll. Peranan klustering digunakan algoritme K-Means, Fuzzy C-Means, K-Medoid, Self-Organization Map (SOM), dll. Sedangkan pada peranan asosiasi digunakan algoritme FP-Growth, A Priori, Chi Square, Coefficient of Correlation, dll.



Gambar 1.1 Data Pengguna Internet di Indonesia



Gambar 1.2 Ilustrasi Data Mining

Tabel 1.1 Perbedaan Peranan Data Mining

Jenis	Atribut/Feature	Kelas/Label/Target	Keterangan
Estimasi	Numerik	Numerik	
Prediksi	Numerik	Numerik	Rentang Waktu
Klasifikasi	Numerik/Kategorial	Numerik/Kategorial	
Klustering	Numerik	-	
Asosiasi	-	-	Hubungan antar atribut

Tabel 1.2 Tipe Data pada Data Mining

Tipe Data	Jenis Atribut	Deskripsi	Contoh
Numerik	Ratio	<ul style="list-style-type: none"> Data yang diperoleh dengan cara pengukuran, dimana jarak dua titik pada skala sudah diketahui Mempunyai titik nol yang absolut 	<ul style="list-style-type: none"> Umur Berat badan Tinggi badan Jumlah uang
	Interval	<ul style="list-style-type: none"> Data yang diperoleh dengan cara pengukuran, dimana jarak dua titik pada skala sudah diketahui Tidak mempunyai titik nol yang absolut 	<ul style="list-style-type: none"> Suhu 0°C-100°C, Umur 20-30 tahun
Kategorial	Ordinal	<ul style="list-style-type: none"> Data yang diperoleh dengan cara kategorisasi atau klasifikasi Tetapi diantara data tersebut terdapat hubungan atau berurutan 	<ul style="list-style-type: none"> Tingkat kepuasan pelanggan (puas, sedang, tidak puas)
	Nominal	<ul style="list-style-type: none"> Data yang diperoleh dengan cara kategorisasi atau klasifikasi Menunjukkan beberapa object yang berbeda 	<ul style="list-style-type: none"> Kode pos Jenis kelamin Nomer id karyawan Nama kota

(Sumber: <http://romisatriawahono.net/dm/>)

Bab 2

Instalasi PHP untuk Data Mining

2.1 Kebutuhan dan Persyaratan

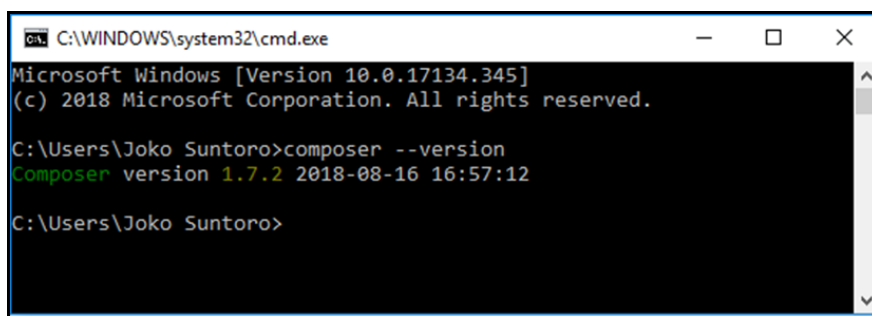
Agar bahasa pemrograman PHP bisa digunakan untuk keperluan data mining, dibutuhkan beberapa kebutuhan dan persyaratan, serta disarankan memahami dasar-dasar bahasa pemrograman PHP. Berikut adalah kebutuhan dan persyaratan yang digunakan, antara lain:

1. Browser (Mozilla Firefox, Google Chrome, dll).
2. Text Editor (Notepad++, Sublime Text, Geany, aksilIDE, dll).
3. Install PHP versi 7 ke atas (disarankan untuk install xampp versi terbaru).
4. Install composer (<https://getcomposer.org/>).

2.2 Cara Penggunaan

Berikut adalah langkah-langkah penggunaan library PHP untuk data mining:

1. Clone/download library PHP untuk data mining melalui link di bawah ini **<https://github.com/jokosuntoro/php-datamining>**
2. Ekstrak folder php-datamining-master.zip.
3. Ubah nama folder menjadi **belajar-phpdm**.
4. Simpan ke dalam folder **htdocs** (...\\xampp\\htdocs\\belajar-phpdm).
5. Cek composer apakah sudah terinstall (lihat Gambar 2.1).
6. Mapping class pada Composer (lihat Gambar 2.2).

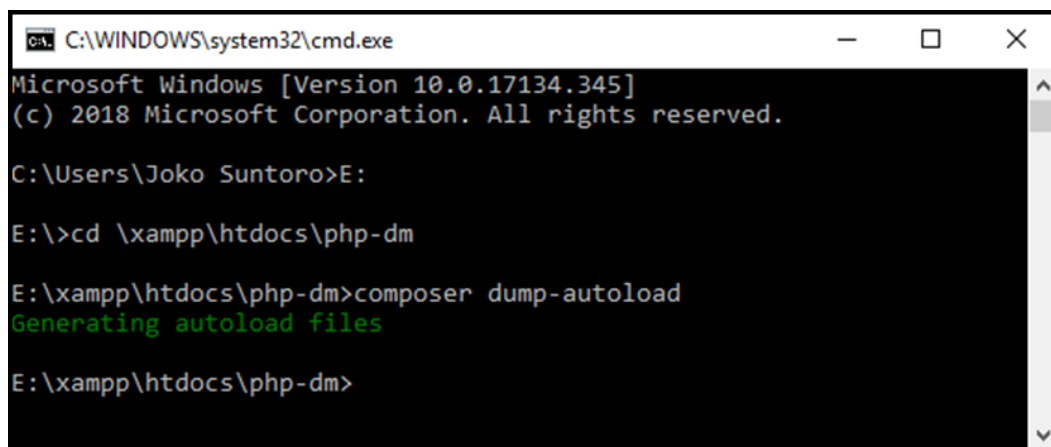


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Joko_Suntoro>composer --version
Composer version 1.7.2 2018-08-16 16:57:12

C:\Users\Joko_Suntoro>
```

Gambar 2.1 Cek Versi Composer



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Joko_Suntoro>E:
E:\>cd \xampp\htdocs\php-dm
E:\xampp\htdocs\php-dm>composer dump-autoload
Generating autoload files

E:\xampp\htdocs\php-dm>
```

Gambar 2.2 Mapping Class pada Composer

Bab 3

Algoritme k-Nearest Neighbor

3.1 Pengenalan Algoritme k-Nearest Neighbor

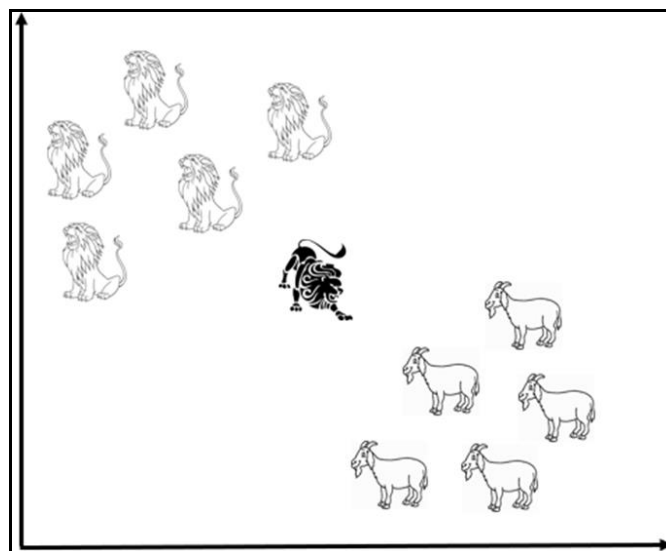
Tujuan dari algoritme klasifikasi adalah untuk memprediksi kelas baru dari *dataset* yang mempunyai kelas (Sáez, Galar, Luengo, & Herrera, 2013). Algoritme k-Nearest Neighbor (k-NN) masuk ke dalam algoritma klasifikasi (Haixiang, Yijing, Yanan, Xiao, & Jinling, 2015). Gambar 3.1 adalah ilustrasi untuk memahami konsep algoritme k-NN.

Sebagai contoh terdapat objek dengan kelas singa dan kambing. Singa mempunyai ciri-ciri yaitu: tidak bertanduk, gigi bertaring, pemakan daging, suara mengaum, kaki berjumlah empat, dan mempunyai ekor. Sedangkan kambing mempunyai ciri-ciri yaitu: bertanduk, gigi tidak bertaring, pemakan tumbuhan, suara mengembik, kaki berjumlah empat, dan mempunyai ekor. Kemudian terdapat objek "X", yang mempunyai ciri-ciri, yaitu: tidak bertanduk, gigi bertaring, pemakan tumbuhan, suara mengaum, jumlah kaki tiga, dan mempunyai ekor.

Dengan melihat *similarity* (kemiripan) pada Tabel 3.1, antara objek "X" dengan objek yang sudah diketahui kelasnya (singa dan kambing), dapat disimpulkan bahwa objek "X" tersebut masuk ke dalam kelas singa, karena nilai *similarity* pada kelas singa (4 kemiripan) lebih besar daripada nilai *similarity* pada kelas kambing (2 kemiripan).

Nilai *similarity* pada algoritme k-NN dihitung berdasarkan jarak (*distance*) antara data training dan data testing. Metode *Euclidean distance* merupakan penghitungan jarak pada algoritme k-NN yang paling banyak digunakan oleh para peneliti (Liu & Zhang, 2012). Formula penghitungan *Euclidean distance* dapat dilihat pada persamaan (3.1).

Menurut Harrington (Harrington, 2012), algoritme k-NN banyak digunakan peneliti karena mempunyai kelebihan antara lain: nilai akurasi tinggi, *insentive* terhadap *outlier* dan tidak ada asumsi terhadap data. Namun algoritme k-NN juga mempunyai kelemahan antara lain: perlu untuk menentukan nilai k optimal, komputasi yang mahal dan membutuhkan banyak memori.



Gambar 3.1 Ilustrasi Algoritme k-Nearest Neighbor

Tabel 3.1 Kemiripan Objek “X” dengan Singa dan Kambing

Ciri-Ciri Objek “X”	Singa	Kambing
Tidak bertanduk	mirip	tidak mirip
Gigi bertaring	mirip	tidak mirip
Pemakan tumbuhan	tidak mirip	mirip
Suara mengaum	mirip	tidak mirip
Kaki berjumlah tiga	tidak mirip	tidak mirip
Mempunyai ekor	mirip	mirip
Jumlah Kemiripan	4	2

Rumus Euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.1)$$

dimana:

- $d(x, y)$ adalah jarak antara data x ke data y
- x_i adalah data testing ke- i
- y_i adalah data training ke- i

Langkah-langkah algoritme k-NN adalah sebagai berikut:

1. Tentukan nilai parameter k (nilai k dipilih secara manual).
2. Hitung jarak antara data training dan data testing (metode *Euclidean distance* digunakan sebagian besar peneliti).
3. Urutkan data training berdasarkan jarak terkecil.
4. Menetapkan kelas, dimana kelas yang dipilih adalah kelas dengan jumlah nilai k terbanyak pada data testing.

3.2 Penghitungan Manual Algoritme k-Nearest Neighbor

Dataset dibagi menjadi dua bagian, yaitu data training dan data testing. Data training adalah data yang sudah mempunyai kelas, sedangkan data testing adalah data yang akan dicari kelasnya. Data training akan membentuk suatu model/pola/pengetahuan, sedangkan data testing digunakan untuk pengukuran evaluasi algoritme.

Dataset yang digunakan pada kasus ini adalah dataset *dummy* yang bisa dilihat pada tabel 3.2 (data training) dan tabel 3.3 (data testing). Kita akan menerapkan algoritme k-NN untuk mengetahui nilai prediksi (klasifikasi) kelas pada data testing, berdasarkan model pada data training.

Tabel 3.2 Data Training

X	Y	Kelas
1,2	2,3	A
2,5	4,6	A
4	1	B
5,6	1,2	B
6,0	3,5	B

Tabel 3.3 Data Testing

X	Y	Kelas
3	2	?

Berikut adalah langkah-langkah penghitungan manual algoritme k-NN:

1. Tentukan Nilai Parameter k

Nilai parameter k pada kasus ini adalah 3 (tiga). Jadi kita akan menyimpulkan data testing yang akan kita cari kelasnya tersebut berdasarkan dengan 3 (tiga) data training terdekat dengan data testing (lihat Gambar 3.2).

2. Hitung Jarak antara Data Training dan Data Testing

Penghitungan jarak yang digunakan untuk mengukur jarak antara data training dan data testing adalah *Euclidean distance* yang bisa lihat pada persamaan (3.1). Tabel 3.4 menunjukkan jarak antara data training ke data testing.

$$Ed_1 = \sqrt{(1,2 - 3)^2 + (2,3 - 2)^2} = 1,825$$

$$Ed_2 = \sqrt{(2,5 - 3)^2 + (4,6 - 2)^2} = 2,648$$

$$Ed_3 = \sqrt{(4 - 3)^2 + (1 - 2)^2} = 1,414$$

$$Ed_4 = \sqrt{(5,6 - 3)^2 + (1,2 - 2)^2} = 2,721$$

$$Ed_5 = \sqrt{(6 - 3)^2 + (3,5 - 2)^2} = 3,354$$

Tabel 3.4 Jarak Data Training ke Data Testing

Data Training		Kelas	Data Testing		Jarak Data Trainig ke Data Testing
X	Y		X	Y	
1,2	2,3	A	3	2	1,825
2,5	4,6	A	3	2	2,648
4	1	B	3	2	1,414
5,6	1,2	B	3	2	2,721
6	3,5	B	3	2	3,354

3. Urutkan Data Berdasarkan Jarak Terkecil

Berdasarkan hasil penghitungan jarak pada Tabel 3.4, data diurutkan secara asecnding berdasarkan jarak terkecil data training ke data testing. Tabel 3.5 menunjukkan urutan data training berdasarkan jarak terkecil.

Tabel 3.5 Urutan Data Berdasarkan Jarak Terkecil

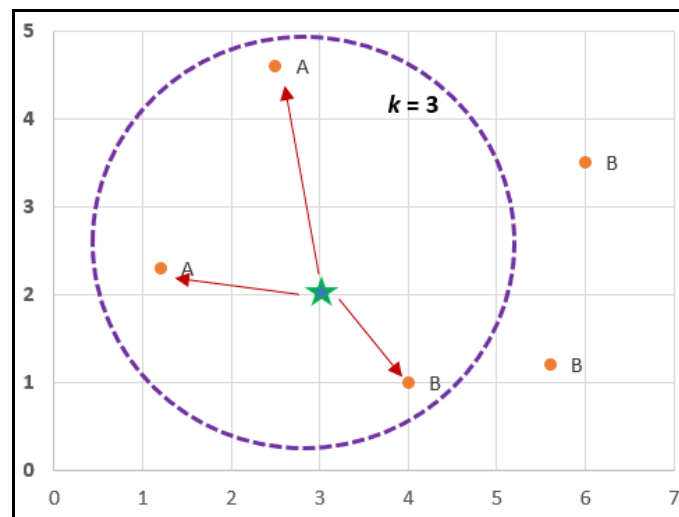
Data Training		Kelas	Data Testing		Jarak Data Trainig ke Data Testing
X	Y		X	Y	
4	1	B	3	2	1,414
1,2	2,3	A	3	2	1,825
2,5	4,6	A	3	2	2,648
5,6	1,2	B	3	2	2,721
6	3,5	B	3	2	3,354

4. Menetapkan Kelas

Setelah data training diurutkan berdasarkan jarak terkecil, langkah selanjutnya adalah menetapkan kelas. Pada Tabel 3.5 kita pilih tiga data terdekat (karena nilai $k = 3$) antara data training dan data testing (lihat Tabel 3.6). Pada Tabel 3.6 dapat kita lihat data dengan kelas A lebih banyak dari pada kelas B, dengan proporsi kelas A sebesar 67% sedangkan kelas B sebesar 33%. Sehingga dapat disimpulkan bahwa, data testing dengan nilai $X = 3$ dan $Y = 2$ masuk ke dalam kelas A.

Tabel 3.6 Tiga Data dengan Jarak Terdekat antara Data Training dan Data Testing

Data Training		Kelas	Data Testing		Jarak Data Trainig ke Data Testing
X	Y		X	Y	
4	1	B	3	2	1,414
1,2	2,3	A	3	2	1,825
2,5	4,6	A	3	2	2,648



Gambar 3.2 Algoritme k-NN dengan Nilai $k = 3$

3.3 Implementasi Algoritme k-Nearest Neighbor

Setelah memahami konsep penghitungan manual algoritme k-NN, akan kita implementasikan algoritme k-NN menggunakan bahasa pemrograman PHP. Dataset yang kita gunakan pada implementasi ini adalah data training pada Tabel 2.2 dan data testing pada Tabel 2.3. Berikut adalah langkah-langkah implementasi algoritme k-NN menggunakan pemrograman PHP:

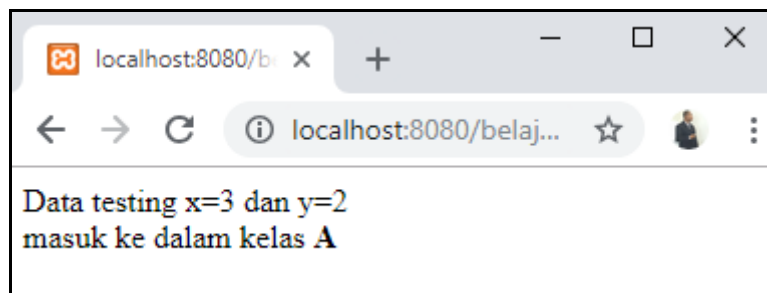
1. Buat file dengan nama **knn.php**.
2. Simpan file tersebut ke dalam folder ...\xampp\htdocs\belajar-phpdml\public.
3. Ketikkan source code knn.php, seperti terlihat pada Gambar 3.3.

```

1  <?php
2  require '../vendor/autoload.php';
3  use jokodm\Datamining\Klasifikasi\KNearestNeighbors;
4
5  // ubah data training ke dalam bentuk array
6  $dtraining = [[1.2, 2, 3], [2.5, 4.6], [4, 1], [5.6, 1.2], [6, 3.5]];
7  $labels = ['A', 'A', 'B', 'B', 'B'];
8
9  // gunakan algoritme k-NN dengan nilai k=3
10 $classifier = new KNearestNeighbors($k=3);
11 $classifier->train($dtraining, $labels);
12
13 // cari kelas data testing dimana nilai x = 3 dan y = 2
14 $pred1 = $classifier->predict([3, 2]);
15
16 // tampilkan ke dalam browser
17 echo "Data testing x=3 dan y=2" . "<br>";
18 echo "masuk ke dalam kelas <b>" . $pred1. "</b><br>";

```

Gambar 3.3 Source Code knn.php



Gambar 3.4 Hasil pada Browser knn.php

Hasil running source code knn.php pada browser dapat dilihat pada Gambar 3.4. Terlihat bahwa data testing $x = 3$, $Y = 2$ masuk ke dalam kelas A, hasil tersebut sama dengan hasil penghitungan manual pada sub bab 2.2.

3.4 Studi Kasus Sistem Cerdas untuk Prediksi Penyakit Kanker Payudara

Tersedia dalam bentuk buku cetak berbayar (on progress)

Bab 4

Algoritme Naïve Bayes

4.1 Pengenalan Algoritme Naïve Bayes

Algoritme Naïve Bayes adalah salah satu algoritme klasifikasi berdasarkan teorema Bayesian pada statistika. Algoritme Naïve Bayes dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas (Han and Kamber, 2012). Teorema Bayesian dapat dilihat pada persamaan 4.1.

Teorema Bayesian menghitung nilai *posterior probability* $P(H|X)$ menggunakan probabilitas $P(H)$, $P(X)$, dan $P(X|H)$ (Kantardzic, 2011). Dimana nilai X adalah data testing yang kelasnya belum diketahui. Nilai H adalah hipotesis data X yang merupakan suatu kelas yang lebih spesifik. Nilai $P(X|H)$ disebut juga dengan *likelihood* adalah probabilitas hipotesis X berdasarkan kondisi H . Nilai $P(H)$ disebut juga dengan *prior probability* adalah probabilitas hipotesis H . Dan nilai $P(X)$ disebut juga dengan *predictor prior probability* adalah probabilitas X .

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \quad (4.1)$$

Algoritme Naïve Bayes sangat cocok untuk melakukan klasifikasi pada dataset bertipe nominal. Untuk dataset bertipe nominal, penghitungan algoritme Naïve Bayes menggunakan persamaan 4.1. Apabila dataset bertipe numerik, maka digunakan penghitungan distribusi Gaussian. Penghitungan distribusi Gaussian dapat dilihat dari persamaan 4.2, dimana dihitung terlebih dahulu nilai rata-rata μ sesuai persamaan 4.3, dan standard deviasi σ sesuai persamaan 4.4.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp \frac{-(x-\mu)^2}{2\sigma^2} \quad (4.2)$$

$$\mu = \frac{\sum_i^n x_i}{n} \quad (4.3)$$

$$\sigma = \sqrt{\frac{\sum_i^n (x_i - \mu)^2}{n - 1}} \quad (4.4)$$

Langkah-langkah algoritme Naïve Bayes adalah sebagai berikut:

1. Siapkan dataset.
2. Hitung jumlah kelas pada data training.
3. Hitung jumlah kasus yang sama dengan kelas yang sama.
4. Kalikan semua hasil sesuai dengan data testing yang akan dicari kelasnya.
5. Bandingkan hasil per kelas, nilai tertinggi ditetapkan sebagai kelas baru.

4.2 Penghitungan Manual Tipe Data Nominal Algoritme Naïve Bayes

Dataset yang digunakan pada penghitungan manual ini adalah data pembelian komputer. Dataset pembelian komputer dibagi menjadi dua bagian, yaitu data training

(lihat Tabel 4.1) dan data testing (lihat Tabel 4.2). Dataset pembelian komputer bertipe data nominal, terdiri dari 4 (empat) atribut dan 1 (satu) kelas.

Berikut adalah langkah-langkah penghitungan manual algoritme Naïve Bayes:

1. Siapkan dataset

Seperti yang telah dijelaskan di atas, dataset yang digunakan pada penghitungan manual ini menggunakan dataset pembelian komputer, dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Data Training 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 4.2 Data Testing Pembelian Komputer

Usia	Pendapatan	Pelajar	Kredit	Kelas
tua	tinggi	tidak	macet	?

2. Hitung Jumlah Kelas pada Data Training

Kelas pada data training terdiri dari dua kategori, yaitu beli komputer dan tidak beli komputer, sehingga probabilitas untuk beli komputer dan tidak beli komputer adalah sebagai berikut:

$$\text{Jumlah kelas beli komputer} = 9$$

$$\text{Jumlah kelas tidak beli komputer} = 5$$

Maka:

$$P(C = \text{"beli"}) = \frac{9}{14} = 0,64$$

$$P(C = \text{"tidak beli"}) = \frac{5}{14} = 0,36$$

3. Hitung Jumlah Kasus yang Sama dengan Kelas yang Sama

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

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

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

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

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

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

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

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

4. Kalikan Semua Hasil Sesuai dengan Data Testing yang Akan Dicari Kelasnya

$$P(X | C = \text{"beli"}) = 0,33 * 0,22 * 0,33 * 0,67 = 0,02$$

$$P(X | C = \text{"tidak beli"}) = 0,40 * 0,40 * 0,80 * 0,40 = 0,05$$

$$P(C = \text{"beli"} | X) = 0,02 * 0,64 = 0,01$$

$$P(C = \text{"tidak beli"} | X) = 0,05 * 0,36 = 0,02$$

5. Bandingkan Hasil per Kelas

Dari penghitungan probabilitas beli komputer dan probabilitas tidak beli komputer pada langkah sebelumnya, maka dapat disimpulkan bahwa data usia = tua, pendapatan = tinggi, pelajar = tidak, dan kredit = macet masuk ke dalam **kelas tidak beli komputer**, karena probabilitas tidak beli komputer (0,02) lebih tinggi dibandingkan probabilitas beli komputer (0,01).

4.3 Penghitungan Manual Tipe Data Numerik dan Nominal Algoritme Naïve Bayes

Pada sub bab sebelumnya, algoritme Naïve Bayes digunakan untuk penyelesaian pada kasus dataset tipe data nominal. Sub Bab ini akan dijelaskan penyelesaian algoritme Naïve Bayes untuk penanganan dataset bertipe numerik dan nominal. Data bertipe numerik akan diselesaikan menggunakan persamaan 4.2, sedangkan pada data bertipe nominal akan diselesaikan menggunakan persamaan 4.1. Berikut adalah langkah-langkah penghitungan manual algoritme Naïve Bayes:

1. Siapkan dataset

Dataset yang akan digunakan pada contoh kasus kali ini adalah dataset bermain golf. Dataset bermain golf dibagi menjadi dua bagian, yaitu data testing (lihat Tabel 4.3) dan data testing (lihat Tabel 4.4). Dataset golf pada data training, terdiri dari 4 (empat) atribut bertipe data nominal dan numerik, serta 1 (satu) kelas bertipe data nominal.

Tabel 4.3 Data Training Bermain Golf

Cuaca	Temperature	Kelembaban	Angin	Play
cerah	85	85	tidak	tidak
cerah	80	90	ada	tidak
mendung	83	78	tidak	ya
hujan	70	96	tidak	ya
hujan	68	80	tidak	ya
hujan	65	70	ada	tidak
mendung	64	65	ada	ya
cerah	72	95	tidak	tidak
cerah	69	70	tidak	ya
hujan	75	80	tidak	ya
cerah	75	70	ada	ya
mendung	72	90	ada	ya
mendung	81	75	tidak	ya
hujan	71	80	ada	tidak

Tabel 4.4 Data Testing Bermain Golf

Cuaca	Temperature	Kelembaban	Angin	Play
cerah	73	80	tidak	?

2. Hitung Jumlah Kelas pada Data Training

Jumlah kelas bermain golf = 9

Jumlah kelas tidak bermain golf = 5

maka:

$$P(\text{play} = \text{"ya"}) = \frac{9}{14} = 0,643$$

$$P(\text{play} = \text{"tidak"}) = \frac{5}{14} = 0,357$$

3. Hitung Jumlah Kasus yang Sama dengan Kelas yang Sama

a. Atribut Cuaca

$$P(\text{cuaca} = \text{"cerah"} | \text{play} = \text{"ya"}) = \frac{2}{9} = 0,222$$

$$P(\text{cuaca} = \text{"cerah"} | \text{play} = \text{"tidak"}) = \frac{3}{5} = 0,600$$

b. Atribut Temperatur

$$\mu_{\text{play}=\text{"ya"}} = \frac{83 + 70 + 68 + 64 + 69 + 75 + 75 + 72 + 81}{9} = 73$$

$$\mu_{\text{play}=\text{"tidak"}} = \frac{85 + 80 + 65 + 72 + 71}{5} = 74,6$$

$$\sigma_{play="ya"} = \sqrt{\frac{(83 - 73)^2 + (70 - 73)^2 + \dots + (81 - 73)^2}{9 - 1}} = 6,164$$

$$\sigma_{play="tidak"} = \sqrt{\frac{(85 - 74,6)^2 + (80 - 74,6)^2 + \dots + (71 - 74,6)^2}{5 - 1}} = 7,893$$

$$P(temp = 73 | play = "ya") = \frac{1}{\sqrt{2\pi} * 6,164} \exp^{\frac{-(73-73)^2}{2*(6,164)^2}} = 0,060$$

$$P(temp = 73 | play = "tidak") = \frac{1}{\sqrt{2\pi} * 7,893} \exp^{\frac{-(73-74,6)^2}{2*(7,893)^2}} = 0,050$$

c. Atribut Kelembaban

$$\mu_{play="ya"} = \frac{78 + 96 + 80 + 65 + 70 + 80 + 70 + 90 + 75}{9} = 78,222$$

$$\mu_{play="tidak"} = \frac{85 + 90 + 70 + 95 + 80}{5} = 84$$

$$\sigma_{play="ya"} = \sqrt{\frac{(78 - 78,222)^2 + (96 - 78,222)^2 + \dots + (75 - 78,222)^2}{9 - 1}} = 9,844$$

$$\sigma_{play="tidak"} = \sqrt{\frac{(85 - 84)^2 + (90 - 84)^2 + \dots + (80 - 84)^2}{5 - 1}} = 9,618$$

$$P(lembab = 80 | play = "ya") = \frac{1}{\sqrt{2\pi} * 9,844} \exp^{\frac{-(80-78,222)^2}{2*(9,844)^2}} = 0,040$$

$$P(lembab = 80 | play = "tidak") = \frac{1}{\sqrt{2\pi} * 9,618} \exp^{\frac{-(80-84)^2}{2*(9,618)^2}} = 0,042$$

d. Atribut Angin

$$P(angin = "tidak" | play = "ya") = \frac{6}{9} = 0,667$$

$$P(angin = "tidak" | play = "tidak") = \frac{2}{5} = 0,400$$

4. Kalikan Semua Hasil Sesuai dengan Data Testing yang Akan Dicari Kelasnya

$$P(X | play = "ya") = 0,222 * 0,060 * 0,040 * 0,667 = 0,00036$$

$$P(X | play = "tidak") = 0,600 * 0,050 * 0,042 * 0,400 = 0,00050$$

$$P(play = "ya" | X) = 0,00036 * 0,643 = 0,00023148$$

$$P(play = "tidak" | X) = 0,00050 * 0,357 = 0,00017850$$

5. Bandingkan Hasil per Kelas

Dari penghitungan probabilitas bermain golf dan probabilitas tidak bermain golf pada langkah sebelumnya, maka dapat disimpulkan bahwa data cuaca = cerah, temperature = 73, kelembaban = 80, dan angin = tidak, masuk ke dalam **kelas bermain golf**, karena probabilitas bermain golf (0,00023148) lebih tinggi dibandingkan probabilitas tidak bermain golf (0,00017850).

4.4 Implementasi Algoritme Naïve Bayes

Dataset yang digunakan pada implementasi ini adalah dataset bermain golf (lihat Tabel 4.3 dan Tabel 4.4). Berikut adalah langkah-langkah implementasi algoritme Naïve Bayes menggunakan bahasa pemrograman PHP:

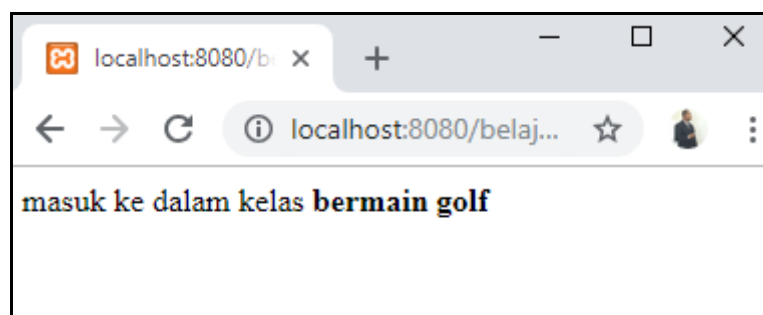
1. Buat file dengan nama **nb-golf.php**.
2. Simpan file tersebut ke dalam folder ...\\xampp\\htdocs\\belajar-phpdm\\public.
3. Ketikkan source code nb-golf.php, seperti terlihat pada Gambar 4.1.

Hasil running source code nb-golf.php pada browser dapat dilihat pada Gambar 4.2. Terlihat bahwa data testing cuaca = cerah, temperature = 73, kelembaban = 80, dan angin = tidak, masuk ke dalam kelas bermain golf, hasil tersebut sama dengan hasil penghitungan manual pada sub bab 4.3.

```

1  <?php
2  require '../vendor/autoload.php';
3  use jokodm\\Datamining\\Klasifikasi\\NaiveBayes;
4  use jokodm\\Datamining\\Dataset\\Demo\\BermainGolf;
5
6  $dataset = new BermainGolf();
7
8  $klasifikasi = new NaiveBayes();
9  $klasifikasi->train($dataset->getSamples(), $dataset->getTargets());
10
11 $pred = $klasifikasi->predict(['cerah', 73, 80, 'tidak']);
12
13 if ($pred == "ya") {
14     echo "masuk ke dalam kelas <b>bermain golf</b>";
15 } else {
16     echo "masuk ke dalam kelas <b>tidak bermain golf</b>";
17 }
    
```

Gambar 4.1 Source Code nb-golf.php



Gambar 4.2 Hasil pada Browser nb-golf.php

4.5 Studi Kasus Sistem Cerdas untuk Prediksi Cacat Software

Tersedia dalam bentuk buku cetak berbayar (on progress)

Bab 5

Algoritme Decision Trees

5.1 Pengenalan Algoritme Decision Trees

Algoritme decision trees masuk ke dalam penerapan data mining klasifikasi. Algoritme decision trees mengkonstruksi pohon keputusan dari sebuah data training yang berupa record-record dalam basis data (Larose, 2005). Algoritme decision trees banyak digunakan karena dapat secara eksplisit menggambarkan suatu pola/pengetahuan/informasi dalam bentuk pohon keputusan (Suntoro & Indah, 2017). Algoritme decision tree terdiri dari kumpulan node (simpul) yang dihubungkan oleh cabang, cabang tersebut bergerak ke bawah dari root (akar) node dan berakhir di leaf (daun) node. Leaf node adalah node yang sudah tidak dapat dipecah lagi, leaf node merepresentasikan prediksi jawaban dari masalah (data testing). Pohon keputusan decision trees berbentuk terbalik, dimana root node berada di paling atas, sedangkan leaf node berada di paling bawah.

Jenis-jenis algoritme decision tree yang banyak digunakan adalah algoritme Iterative Dichotomiser 3 (ID3), algoritme C4.5, dan algoritme Classification and Regression Tree (CART). Pemilihan atribut yang dijadikan node pada algoritme ID3 berbasis information gain (Yang, Guo, & Jin, 2018), algoritme C4.5 berbasis gain ratio, sedangkan CART berbasis gini index.

5.2 Penghitungan Manual Algoritme ID3

Algoritme ID3 menggunakan penghitungan *entropy* dan *information gain* untuk pemilihan atribut menjadi node. Formula penghitungan *entropy* dapat dilihat pada persamaan 5.1, dimana nilai S adalah himpunan kasus, n adalah jumlah partisi S , dan p_i adalah proporsi himpunan kasus ke- i terhadap himpunan kasus. Sedangkan formula penghitungan *information gain* dapat dilihat pada persamaan 5.2, dimana S adalah himpunan kasus, A adalah atribut, n adalah jumlah partisi atribut A , $|S_i|$ adalah proporsi S_i terhadap S , $|S|$ adalah jumlah kasus dalam S , dan $entropy(S_i)$ adalah entropy untuk sampel yang memiliki nilai ke- i .

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i \quad (5.1)$$

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (5.2)$$

Berikut adalah langkah-langkah penghitungan manual algoritme ID3:

1. Siapkan Dataset

Dataset yang digunakan adalah dataset pembelian komputer (lihat Tabel 4.1 sebagai data training dan Tabel 4.2 sebagai data testing).

2. Hitung Nilai Entropy

$$Entropy(kelas) = \left(-\left(\frac{9}{14}\right) * \log_2 \left(\frac{9}{14}\right) \right) + \left(-\left(\frac{5}{14}\right) * \log_2 \left(\frac{5}{14}\right) \right) = 0,940$$

$$Entropy(usia muda) = \left(-\left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right) \right) + \left(-\left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right) \right) = 0,971$$

$$Entropy(usia tengah baya) = \left(-\left(\frac{4}{4}\right) * \log_2\left(\frac{4}{4}\right) \right) + \left(-\left(\frac{0}{4}\right) * \log_2\left(\frac{0}{4}\right) \right) = 0$$

$$Entropy(usia tua) = \left(-\left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right) \right) + \left(-\left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right) \right) = 0,971$$

Jika semua atribut dihitung nilai *entropy*, maka didapatkan hasil seperti yang terlihat pada Tabel 5.1.

Tabel 5.1 Hasil Penghitungan *Entropy* Semua Atribut Dataset Pembelian Komputer

Atribut	Kategori	Jumlah Kasus	Beli	Tidak Beli	Entropy
Kelas		14	9	5	0,940
Usia					
	muda	5	2	3	0,971
	tengah baya	4	4	0	0
	tua	5	3	2	0,971
Pendapatan					
	rendah	4	3	1	0,811
	sedang	6	4	2	0,918
	tinggi	4	2	2	1
Pelajar					
	ya	7	6	1	0,592
	tidak	7	3	4	0,985
Kredit					
	macet	8	6	2	0,811
	lancar	6	3	3	1

3. Hitung Nilai *Gain*

$$Gain(Usia) = 0,940 - \left(\left(\frac{5}{14} * 0,971 \right) + \left(\frac{4}{14} * 0 \right) + \left(\frac{5}{14} * 0,971 \right) \right) = 0,247$$

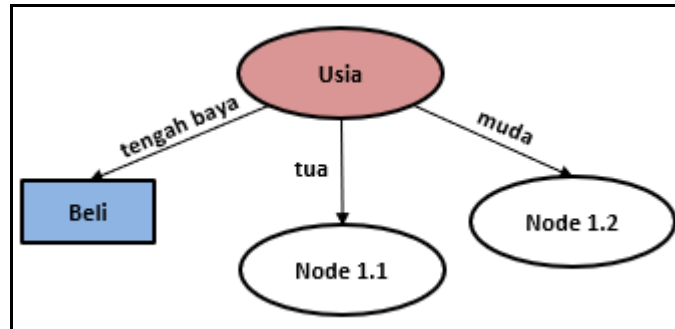
$$Gain(Pendapatan) = 0,940 - \left(\left(\frac{4}{14} * 0,811 \right) + \left(\frac{6}{14} * 0,918 \right) + \left(\frac{4}{14} * 1 \right) \right) = 0,029$$

$$Gain(Pelajar) = 0,940 - \left(\left(\frac{7}{14} * 0,592 \right) + \left(\frac{7}{14} * 0,985 \right) \right) = 0,152$$

$$Gain(Kredit) = 0,940 - \left(\left(\frac{8}{14} * 0,811 \right) + \left(\frac{6}{14} * 1 \right) \right) = 0,048$$

4. Membuat Node dan Cabang dari Nilai Gain Maksimal

Pada langkah 3 di atas, kita telah menghitung nilai gain. Dari nilai gain tersebut, kita pilih nilai gain maksimal di antara masing-masing atribut. Nilai gain maksimal adalah *Gain(Usia)* dengan nilai 0,247, sehingga atribut **Usia** menjadi **node akar**.



Gambar 5.1 Pohon Keputusan Pembentuk Node Akar

Gambar 5.1, menunjukkan pembentukan node akar, yaitu atribut usia. Cabang yang terbentuk berisi data kategori pada atribut usia, yaitu tengah baya, tua, dan muda. Pada kategori tengah baya, nilai *entropy* nya adalah 0 (nol), sehingga pada cabang tengah baya terbentuk **leaf node**. Lihat Tabel 5.1, pada atribut usia dan kategori tengah baya, frekuensi beli (sejumlah 4) lebih tinggi dibandingkan tidak beli (sejumlah 0), maka dapat disimpulkan bahwa nilai kelas pada kategori tengah baya adalah beli. Untuk kategori tua dan muda pada atribut usia, belum terbentuk leaf node, karena nilai *entropy* tidak sama dengan 0 (nol), sehingga akan dilakukan penghitungan kembali pada langkah selanjutnya.

5. Ulangi Langkah (2) Sampai Langkah (4) Hingga Semua Node Terpartisi

a. Penghitungan untuk Node 1.1

Berdasarkan pembentukan node akar pada Gambar 5.1, node 1.1 akan dilakukan penghitungan lanjut. Untuk mempermudah penghitungan, data pada Tabel 4.1 difilter dengan mengambil data atribut Usia = tua, lebih jelasnya lihat Tabel 5.2.

Tabel 5.2 Data Training Bermain Golf dengan Filter Atribut Usia = tua

Usia	Pendapatan	Pelajar	Kredit	Kelas
tua	sedang	tidak	macet	beli
tua	rendah	ya	macet	beli
tua	rendah	ya	lancar	tidak beli
tua	sedang	ya	macet	beli
tua	sedang	tidak	lancar	tidak beli

Hitung nilai *entropy* dengan filter atribut usia = tua

$$Entropy(Usia = tua) = \left(-\left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right) \right) + \left(-\left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right) \right) = 0,971$$

$$Entropy(Pendapatan = rendah) = \left(-\left(\frac{1}{2}\right) * \log_2\left(\frac{1}{2}\right) \right) + \left(-\left(\frac{1}{2}\right) * \log_2\left(\frac{1}{2}\right) \right) = 1$$

$$Entropy(Pendapatan = sedang) = \left(-\left(\frac{2}{3}\right) * \log_2\left(\frac{2}{3}\right) \right) + \left(-\left(\frac{1}{3}\right) * \log_2\left(\frac{1}{3}\right) \right) = 0,918$$

$$Entropy(Pendapatan = tinggi) = \left(-\left(\frac{0}{0}\right) * \log_2\left(\frac{0}{0}\right) \right) + \left(-\left(\frac{0}{0}\right) * \log_2\left(\frac{0}{0}\right) \right) = 0$$

Jika semua atribut dengan filter usia = tua dihitung nilai *entropy*, maka didapatkan hasil seperti yang terlihat pada Tabel 5.3.

Tabel 5.3 Hasil Penghitungan *Entropy* dengan Filter Atribut Usia = Tua

Atribut	Kategori	Jumlah Kasus	Beli	Tidak Beli	Entropy
Usia	tua	5	3	2	0,971
Pendapatan	rendah	2	1	1	1
	sedang	3	2	1	0,918
	tinggi	0	0	0	0
Pelajar	ya	3	2	1	0,918
	tidak	2	1	1	1
Kredit	macet	3	3	0	0
	lancar	2	0	2	0

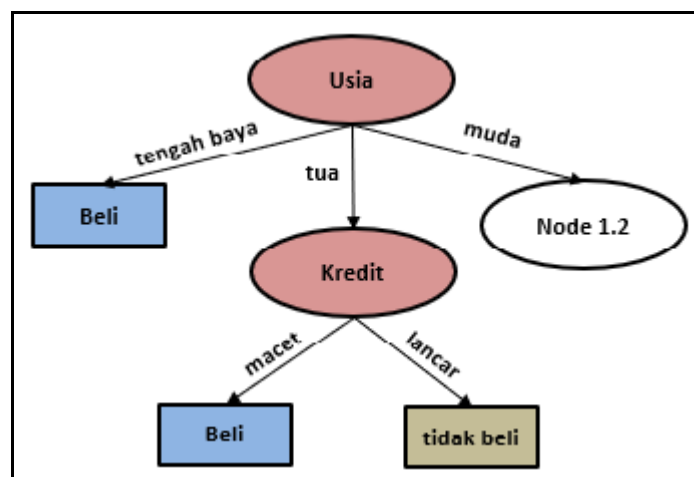
Setelah dihitung nilai *entropy* pada masing-masing atribut, langkah selanjutnya adalah menghitung nilai gain dengan filter atribut usia = tua, didapatkan nilai gain sebagai berikut:

$$Gain(Pendapatan) = 0,971 - \left(\left(\frac{2}{5} * 1 \right) + \left(\frac{3}{5} * 0,918 \right) + \left(\frac{0}{5} * 0 \right) \right) = 0,020$$

$$Gain(Pelajar) = 0,971 - \left(\left(\frac{3}{5} * 0,918 \right) + \left(\frac{2}{5} * 1 \right) \right) = 0,020$$

$$Gain(Kredit) = 0,971 - \left(\left(\frac{3}{5} * 0 \right) + \left(\frac{2}{5} * 0 \right) \right) = 0,971$$

Dari penghitungan nilai gain dengan filter atribut usia = tua, didapatkan gain maksimal adalah atribut kredit, sehingga disimpulkan bahwa node 1.1 berisi atribut kredit. Gambar 5.2 menunjukkan pohon keputusan yang terbentuk setelah penghitungan node akar dan node 1.1.



Gambar 5.2 Pohon Keputusan Pembentuk Node Akar dan Node 1.1

b. Penghitungan untuk Node 1.2

Penghitungan node 1.2 dihitung berdasarkan filter atribut usia = muda. Tabel 5.4 menunjukkan dataset bermain golf dengan filter atribut usia = muda.

Tabel 5.4 Data Training Bermain Golf dengan Filter Atribut Usia = muda

Usia	Pendapatan	Pelajar	Kredit	Kelas
muda	tinggi	tidak	macet	tidak beli
muda	tinggi	tidak	lancar	tidak beli
muda	sedang	tidak	macet	tidak beli
muda	rendah	tidak	macet	beli
muda	sedang	ya	lancar	beli

Hitung nilai *entropy* dengan filter atribut usia = muda

$$Entropy (Usia = muda) = \left(-\left(\frac{2}{5}\right) * \log_2 \left(\frac{2}{5}\right) \right) + \left(-\left(\frac{3}{5}\right) * \log_2 \left(\frac{3}{5}\right) \right) = 0,971$$

$$Entropy (kredit = macet) = \left(-\left(\frac{1}{3}\right) * \log_2 \left(\frac{1}{3}\right) \right) + \left(-\left(\frac{2}{3}\right) * \log_2 \left(\frac{2}{3}\right) \right) = 0,918$$

$$Entropy (kredit = lancar) = \left(-\left(\frac{1}{2}\right) * \log_2 \left(\frac{1}{2}\right) \right) + \left(-\left(\frac{1}{2}\right) * \log_2 \left(\frac{1}{2}\right) \right) = 1$$

Jika semua atribut dengan filter usia = tua dihitung nilai *entropy*, maka didapatkan hasil seperti yang terlihat pada Tabel 5.5.

Tabel 5.5 Hasil Penghitungan *Entropy* dengan Filter Atribut Usia = Muda

Atribut	Kategori	Jumlah Kasus	Beli	Tidak Beli	Entropy
Usia	muda	5	2	3	0,971
Pendapatan	rendah	1	1	0	0
	sedang	2	1	1	1
	tinggi	2	0	2	0
Pelajar	ya	2	2	0	0
	tidak	3	0	3	0
Kredit	macet				
	lancar	3	1	2	0,918

Setelah dihitung nilai *entropy* pada masing-masing atribut, langkah selanjutnya adalah menghitung nilai gain dengan filter atribut usia = tua, didapatkan nilai gain sebagai berikut:

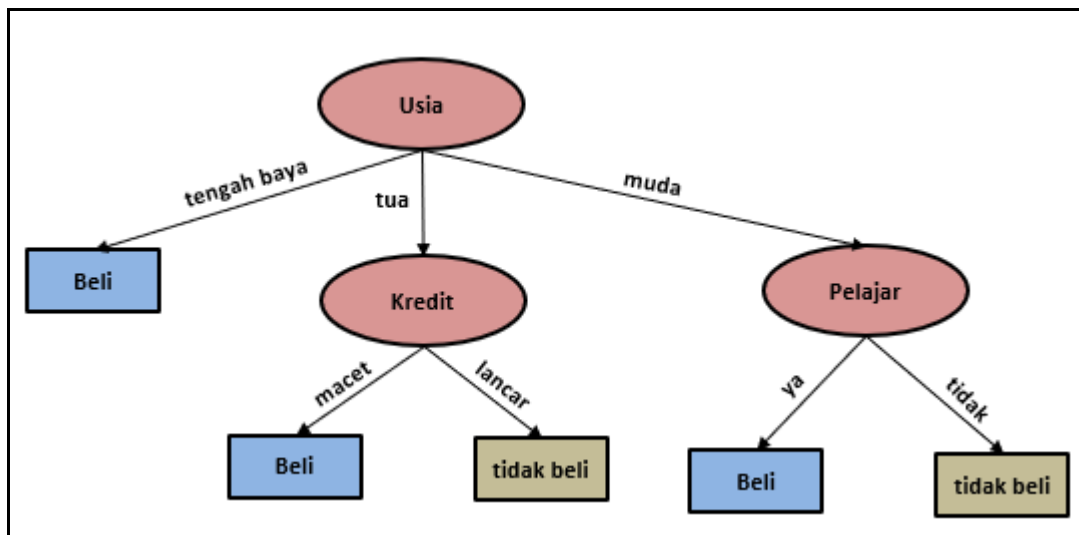
$$Gain(Pendapatan) = 0,971 - \left(\left(\frac{1}{5} * 0 \right) + \left(\frac{2}{5} * 1 \right) + \left(\frac{2}{5} * 0 \right) \right) = 0,571$$

$$Gain(Pelajar) = 0,971 - \left(\left(\frac{2}{5} * 0 \right) + \left(\frac{3}{5} * 0 \right) \right) = 0,971$$

$$Gain(Kredit) = 0,971 - \left(\left(\frac{3}{5} * 0,918 \right) + \left(\frac{2}{5} * 1 \right) \right) = 0,020$$

Dari penghitungan nilai gain dengan filter atribut usia = muda, didapatkan gain maksimal adalah atribut pelajar, sehingga disimpulkan bahwa node 1.2 berisi atribut pelajar. Gambar 5.3 menunjukkan pohon keputusan dengan node telah terpartisi semua (terbentuk setelah penghitungan node akar, node 1.1, dan node 1.2).

Pola yang terbentuk dari algoritme decision trees berupa pohon keputusan. Pohon keputusan pada Gambar 5.3 apabila diterapkan dalam bentuk pseudo-code akan berupa pencabangan (if-elseif), seperti terlihat pada Gambar 5.4.



Gambar 5.3 Pohon Keputusan Setelah Semua Node Terpartisi

```

if (usia = tengah baya)
    echo "beli komputer";
elseif (usia = tua and kredit = macet)
    echo "beli komputer";
elseif (usia = tua and kredit = lancar)
    echo "tidak beli komputer";
elseif (usia = muda and pelajar = ya)
    echo "beli komputer";
elseif (usia = muda and pelajar = tidak)
    echo "tidak beli komputer";
    
```

Gambar 5.4 Pseudo-Code Algoritme ID3 pada Dataset Pembelian Komputer

5.3 Penghitungan Manual Algoritme C4.5

Algoritme C4.5 menggunakan penghitungan *entropy*, *information gain*, *split info* dan *gain ratio* untuk pemilihan atribut menjadi node. Formula penghitungan *entropy* dan *information gain* dapat dilihat pada persamaan 5.1 dan 5.2. Sedangkan untuk penghitungan *split info* dan *gain ratio* dapat dilihat pada persamaan 5.3 dan 5.4. Pada *SplitInfo*, nilai D adalah ruang data sampel yang digunakan untuk training, nilai D_j adalah jumlah sampel pada atribut j .

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right) \quad (5.3)$$

$$GainRatio = \frac{Gain(S, A)}{SplitInfo_A(D)} \quad (5.4)$$

Berikut adalah langkah-langkah penghitungan manual algoritme C4.5:

1. Siapkan dataset

Dataset yang akan digunakan pada penghitungan manual algoritme C4.5 adalah dataset pengajuan kredit yang dapat dilihat pada Tabel 5.6. Dataset persetujuan kredit terdiri dari empat atribut biasa dan satu atribut spesial (kelas). Tipe data pada dataset persetujuan kredit adalah nominal. Jumlah data pada dataset persetujuan kredit adalah 14 (empat belas) data.

Tabel 5.6 Dataset Pengajuan Kredit

Penghasilan	Pekerjaan	Hub. Sosial	Status Rumah	Layak Kredit
rendah	ASN	buruk	kontrak	tidak layak
tinggi	swasta	baik	HM	layak
rendah	pengusaha	buruk	HM	tidak layak
rendah	pengusaha	baik	kontrak	layak
sedang	swasta	baik	kontrak	tidak layak
rendah	swasta	baik	HM	layak
rendah	ASN	buruk	HM	tidak layak
sedang	pengusaha	buruk	HM	layak
sedang	swasta	baik	HM	layak
sedang	pengusaha	buruk	kontrak	tidak layak
sedang	pengusaha	baik	kontrak	layak
tinggi	ASN	buruk	HM	layak
tinggi	pengusaha	buruk	kontrak	layak
tinggi	ASN	baik	HM	layak

2. Proses Pembentukan Node Akar

A. Hitung Nilai Entropy

$$Entropy(Layak Kredit) = \left(- \left(\frac{9}{14} \right) * \log_2 \left(\frac{9}{14} \right) \right) + \left(- \left(\frac{5}{14} \right) * \log_2 \left(\frac{5}{14} \right) \right) = 0,940$$

$$Entropy(Penghasilan tinggi) = \left(- \left(\frac{4}{4} \right) * \log_2 \left(\frac{4}{4} \right) \right) + \left(- \left(\frac{0}{4} \right) * \log_2 \left(\frac{0}{4} \right) \right) = 0$$

$$Entropy(Penghasilan\ sedang) = \left(-\left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right) \right) + \left(-\left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right) \right) = 0,971$$

$$Entropy(Penghasilan\ sedang) = \left(-\left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right) \right) + \left(-\left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right) \right) = 0,971$$

Jika semua atribut pada dataset pengajuan kredit dihitung nilai *entropy*, maka didapatkan hasil seperti yang terlihat pada Tabel 5.7.

B. Hitung Nilai *Gain*

$$Gain(Penghasilan) = 0,940 - \left(\left(\frac{4}{14} * 0\right) + \left(\frac{5}{14} * 0,971\right) + \left(\frac{5}{14} * 0,971\right) \right) = 0,247$$

$$Gain(Pekerjaan) = 0,940 - \left(\left(\frac{6}{14} * 0,918\right) + \left(\frac{4}{14} * 0,811\right) + \left(\frac{4}{14} * 1\right) \right) = 0,029$$

$$Gain(Hub. Sosial) = 0,940 - \left(\left(\frac{7}{14} * 0,592\right) + \left(\frac{7}{14} * 0,985\right) \right) = 0,152$$

$$Gain(Status Rumah) = 0,940 - \left(\left(\frac{8}{14} * 0,811\right) + \left(\frac{6}{14} * 1\right) \right) = 0,048$$

Tabel 5.7 Hasil Penghitungan *Entropy* Semua Atribut Dataset Pengajuan Kredit

Atribut	Kategori	Jumlah Kasus	layak	tidak layak	Entropy
Layak Kredit		14	9	5	0,940
Penghasilan					
	tinggi	4	4	0	0
	sedang	5	3	2	0,971
	rendah	5	2	3	0,971
Pekerjaan					
	pengusaha	6	4	2	0,918
	swasta	4	3	1	0,811
	ASN	4	2	2	1
Hub. Sosial					
	baik	7	6	1	0,592
	buruk	7	3	4	0,985
Status Rumah					
	HM	8	6	2	0,811
	kontrak	6	3	3	1

C. Hitung Nilai *SplitInfo*

$$\text{SplitInfo}(\text{Penghasilan}) = \left(-\left(\frac{4}{14}\right) * \log_2 \left(\frac{4}{14}\right) \right) + \left(-\left(\frac{5}{14}\right) * \log_2 \left(\frac{5}{14}\right) \right) + \left(-\left(\frac{5}{14}\right) * \log_2 \left(\frac{5}{14}\right) \right) = 1,577$$

$$\text{SplitInfo}(\text{Pekerjaan}) = \left(-\left(\frac{6}{14}\right) * \log_2 \left(\frac{6}{14}\right) \right) + \left(-\left(\frac{4}{14}\right) * \log_2 \left(\frac{4}{14}\right) \right) + \left(-\left(\frac{4}{14}\right) * \log_2 \left(\frac{4}{14}\right) \right) = 1,577$$

$$\text{SplitInfo}(\text{Hub. Sosial}) = \left(-\left(\frac{7}{14}\right) * \log_2 \left(\frac{7}{14}\right) \right) + \left(-\left(\frac{7}{14}\right) * \log_2 \left(\frac{7}{14}\right) \right) = 1$$

$$\text{SplitInfo}(\text{Status Rumah}) = \left(-\left(\frac{8}{14}\right) * \log_2 \left(\frac{8}{14}\right) \right) + \left(-\left(\frac{6}{14}\right) * \log_2 \left(\frac{6}{14}\right) \right) = 0,985$$

D. Hitung Nilai *GainRatio*

$$\text{GainRatio}(\text{Penghasilan}) = \frac{0,247}{1,577} = \mathbf{0,156}$$

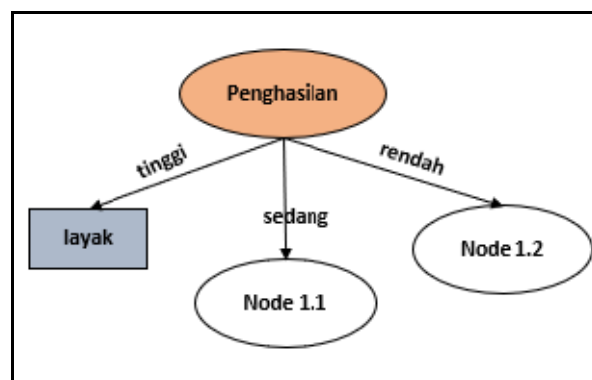
$$\text{GainRatio}(\text{Pekerjaan}) = \frac{0,029}{1,577} = 0,019$$

$$\text{GainRatio}(\text{Hub. Sosial}) = \frac{0,152}{1} = 0,152$$

$$\text{GainRatio}(\text{Status Rumah}) = \frac{0,048}{0,985} = 0,049$$

E. Pembentukan Node Akar

Dari penghitungan nilai *GainRation* pada langkah sebelumnya, didapatkan hasil bahwa nilai *GainRation* atribut penghasilan lebih besar daripada atribut-atribut lainnya. Sehingga atribut penghasilan dijadikan node akar, dan mempunyai tiga cabang, yaitu tinggi, sedang, dan rendah. Gambar 5.5 menunjukkan node akar pada dataset pengajuan kredit menggunakan algoritme C4.5. Pada cabang penghasilan tinggi telah didapatkan kelas yaitu layak, sedangkan pada cabang penghasilan sedang dan rendah terbentuk node baru, yaitu node 1.1 dan node 1.2. Node 1.1, yang akan dihitung pada langkah berikutnya.



Gambar 5.5 Node Akar Dataset Pengajuan Kredit

Tabel 5.8 Dataset Pengajuan Kredit dengan Filter Atribut Penghasilan = Sedang

Penghasilan	Pekerjaan	Hub. Sosial	Status Rumah	Layak Kredit
sedang	swasta	baik	kontrak	tidak layak
sedang	pengusaha	buruk	HM	layak
sedang	swasta	baik	HM	layak
sedang	pengusaha	buruk	kontrak	tidak layak
sedang	pengusaha	baik	kontrak	layak

3. Proses Pembentukan Node 1.1

Berdasarkan pembentukan node akar pada Gambar 5.5, node 1.1 akan dilakukan penghitungan dengan mengambil data atribut penghasilan = sedang pada dataset pengajuan kredit, untuk lebih jelasnya lihat Tabel 5.8.

A. Hitung Nilai Entropy

$$Entropy(Penghasilan\ sedang) = \left(-\left(\frac{3}{5}\right) * \log_2 \left(\frac{3}{5}\right) \right) + \left(-\left(\frac{2}{5}\right) * \log_2 \left(\frac{2}{5}\right) \right) = 0,971$$

$$Entropy(Pekerjaan\ pengusaha) = \left(-\left(\frac{2}{3}\right) * \log_2 \left(\frac{2}{3}\right) \right) + \left(-\left(\frac{1}{3}\right) * \log_2 \left(\frac{1}{3}\right) \right) = 0,918$$

$$Entropy(Pekerjaan\ swasta) = \left(-\left(\frac{1}{2}\right) * \log_2 \left(\frac{1}{2}\right) \right) + \left(-\left(\frac{1}{2}\right) * \log_2 \left(\frac{1}{2}\right) \right) = 1$$

$$Entropy(Pekerjaan\ ASN) = \left(-\left(\frac{0}{0}\right) * \log_2 \left(\frac{0}{0}\right) \right) + \left(-\left(\frac{0}{0}\right) * \log_2 \left(\frac{0}{0}\right) \right) = 0$$

Jika semua atribut pada dataset pengajuan kredit dengan filter atribut penghasilan sedang dihitung nilai *entropy*, maka didapatkan hasil seperti yang terlihat pada Tabel 5.9.

Tabel 5.9 Hasil Penghitungan Entropy dengan Filter Atribut Penghasilan = Sedang

Atribut	Kategori	Jumlah Kasus	layak	tidak layak	Entropy
Penghasilan	sedang	5	3	2	0,971
Pekerjaan					
	pengusaha	3	2	1	0,918
	swasta	2	1	1	1
	ASN	0	0	0	0
Hub. Sosial					
	baik	3	2	1	0,918
	buruk	2	1	1	1
Status Rumah					
	HM	2	2	0	0
	kontrak	3	1	2	0,918

B. Hitung Nilai *Gain*

$$Gain(Pekerjaan) = 0,971 - \left(\left(\frac{3}{5} * 0,918 \right) + \left(\frac{2}{5} * 1 \right) + \left(\frac{0}{5} * 0 \right) \right) = 0,020$$

$$Gain(Hub. Sosial) = 0,971 - \left(\left(\frac{3}{5} * 0,918 \right) + \left(\frac{2}{5} * 1 \right) \right) = 0,020$$

$$Gain(Status Rumah) = 0,971 - \left(\left(\frac{2}{5} * 0 \right) + \left(\frac{3}{5} * 0,918 \right) \right) = 0,420$$

C. Hitung Nilai *SplitInfo*

$$SplitInfo(Pekerjaan) = \left(-\left(\frac{3}{5} \right) * \log_2 \left(\frac{3}{5} \right) \right) + \left(-\left(\frac{2}{5} \right) * \log_2 \left(\frac{2}{5} \right) \right) + \left(-\left(\frac{0}{5} \right) * \log_2 \left(\frac{0}{5} \right) \right) = 0$$

$$SplitInfo(Hub. Sosial) = \left(-\left(\frac{3}{5} \right) * \log_2 \left(\frac{3}{5} \right) \right) + \left(-\left(\frac{2}{5} \right) * \log_2 \left(\frac{2}{5} \right) \right) = 0,971$$

$$SplitInfo(Status Rumah) = \left(-\left(\frac{2}{5} \right) * \log_2 \left(\frac{2}{5} \right) \right) + \left(-\left(\frac{3}{5} \right) * \log_2 \left(\frac{3}{5} \right) \right) = 0,971$$

D. Hitung Nilai *GainRatio*

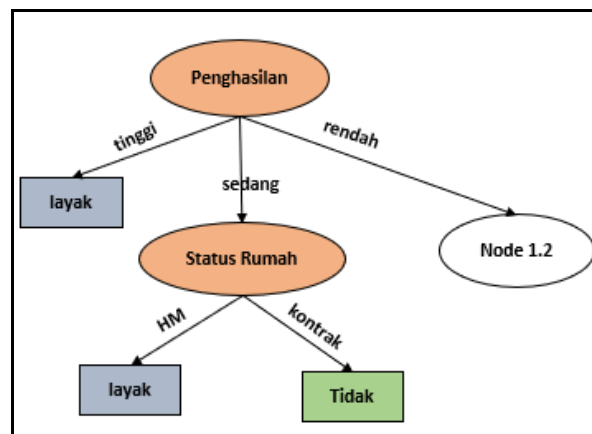
$$GainRatio(Pekerjaan) = \frac{0,020}{0} = 0$$

$$GainRatio(Hub. Sosial) = \frac{0,020}{0,971} = 0,021$$

$$GainRatio(Status Rumah) = \frac{0,420}{0,971} = \mathbf{0,433}$$

E. Pembentukan Node 1.1

Dari penghitungan sebelumnya nilai *GainRation* atribut status rumah lebih besar daripada atribut-atribut lainnya. Sehingga atribut status rumah dijadikan node 1.1, dan mempunyai dua cabang, yaitu HM dan kontrak. Gambar 5.6 menunjukkan node 1.1 pada dataset pengajuan kredit menggunakan algoritme C4.5. Pada cabang status rumah HM dihasilkan kelas yaitu layak, sedangkan pada cabang status rumah kontrak dihasilkan kelas yaitu tidak (diterapkan metode pruning, dengan frekuensi layak = 1; dan frekuensi tidak layak = 2; sehingga dihasilkan kelas tidak).



Gambar 5.6 Pembentukan Node 1.1 Dataset Pengajuan Kredit

Tabel 5.10 Dataset Pengajuan Kredit dengan Filter Atribut Penghasilan = Rendah

Penghasilan	Pekerjaan	Hub. Sosial	Status Rumah	Layak Kredit
rendah	ASN	buruk	kontrak	tidak layak
rendah	pengusaha	buruk	HM	tidak layak
rendah	pengusaha	baik	kontrak	layak
rendah	swasta	baik	HM	layak
rendah	ASN	buruk	HM	tidak layak

4. Proses Pembentukan Node 1.2

Berdasarkan pembentukan node akar pada Gambar 5.5, node 1.2 akan dilakukan penghitungan dengan mengambil data atribut penghasilan = rendah pada dataset pengajuan kredit, untuk lebih jelasnya lihat Tabel 5.10.

A. Hitung Nilai Entropy

$$Entropy(Penghasilan\ rendah) = \left(-\left(\frac{2}{5}\right) * \log_2 \left(\frac{2}{5}\right) \right) + \left(-\left(\frac{3}{5}\right) * \log_2 \left(\frac{3}{5}\right) \right) = 0,971$$

$$Entropy(Pekerjaan\ pengusaha) = \left(-\left(\frac{1}{2}\right) * \log_2 \left(\frac{1}{2}\right) \right) + \left(-\left(\frac{1}{2}\right) * \log_2 \left(\frac{1}{2}\right) \right) = 1$$

$$Entropy(Pekerjaan\ swasta) = \left(-\left(\frac{1}{1}\right) * \log_2 \left(\frac{1}{1}\right) \right) + \left(-\left(\frac{0}{1}\right) * \log_2 \left(\frac{0}{1}\right) \right) = 0$$

$$Entropy(Pekerjaan\ ASN) = \left(-\left(\frac{0}{2}\right) * \log_2 \left(\frac{0}{2}\right) \right) + \left(-\left(\frac{2}{2}\right) * \log_2 \left(\frac{2}{2}\right) \right) = 0$$

Jika semua atribut pada dataset pengajuan kredit dengan filter atribut penghasilan rendah dihitung nilai *entropy*, maka didapatkan hasil seperti yang terlihat pada Tabel 5.11.

Tabel 5.11 Hasil Penghitungan *Entropy* dengan Filter Atribut Penghasilan = Rendah

Atribut	Kategori	Jumlah Kasus	layak	tidak layak	Entropy
Penghasilan	rendah	5	2	3	0,971
Pekerjaan					
	pengusaha	2	1	1	1
	swasta	1	1	0	0
	ASN	2	0	2	0
Hub. Sosial					
	baik	2	2	0	0
	buruk	3	0	3	0
Status Rumah					
	HM	3	1	2	0,918
	kontrak	2	1	1	1

B. Hitung Nilai *Gain*

$$Gain(Pekerjaan) = 0,971 - \left(\left(\frac{2}{5} * 1 \right) + \left(\frac{1}{5} * 0 \right) + \left(\frac{2}{5} * 0 \right) \right) = 0,571$$

$$Gain(Hub. Sosial) = 0,971 - \left(\left(\frac{2}{5} * 0 \right) + \left(\frac{3}{5} * 0 \right) \right) = 0,971$$

$$Gain(Status Rumah) = 0,971 - \left(\left(\frac{3}{5} * 0,918 \right) + \left(\frac{2}{5} * 1 \right) \right) = 0,020$$

C. Hitung Nilai *SplitInfo*

$$SplitInfo(Pekerjaan) = \left(-\left(\frac{2}{5} \right) * \log_2 \left(\frac{2}{5} \right) \right) + \left(-\left(\frac{1}{5} \right) * \log_2 \left(\frac{1}{5} \right) \right) + \left(-\left(\frac{2}{5} \right) * \log_2 \left(\frac{2}{5} \right) \right) = 1,522$$

$$SplitInfo(Hub. Sosial) = \left(-\left(\frac{2}{5} \right) * \log_2 \left(\frac{2}{5} \right) \right) + \left(-\left(\frac{3}{5} \right) * \log_2 \left(\frac{3}{5} \right) \right) = 0,971$$

$$SplitInfo(Status Rumah) = \left(-\left(\frac{3}{5} \right) * \log_2 \left(\frac{3}{5} \right) \right) + \left(-\left(\frac{2}{5} \right) * \log_2 \left(\frac{2}{5} \right) \right) = 0,971$$

D. Hitung Nilai *GainRatio*

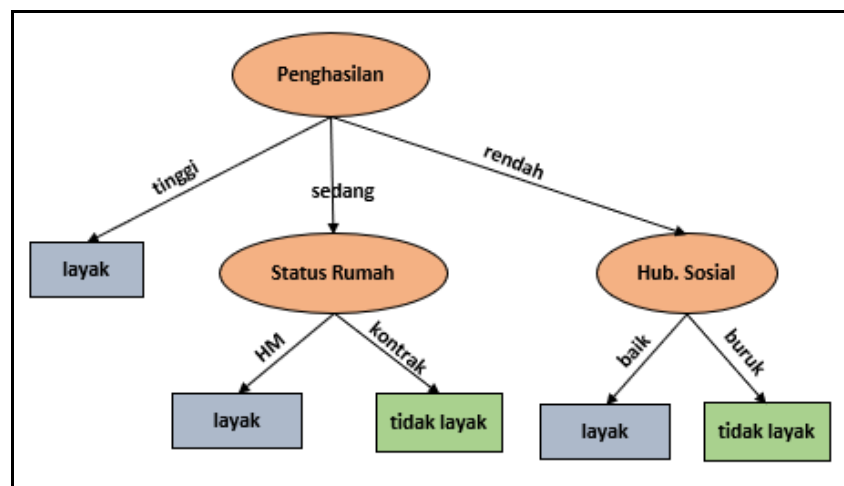
$$GainRatio(Pekerjaan) = \frac{0,571}{1,522} = 0,375$$

$$GainRatio(Hub. Sosial) = \frac{0,971}{0,971} = 1$$

$$GainRatio(Status Rumah) = \frac{0,020}{0,971} = 0,021$$

E. Pembentukan Node 1.2

Dari penghitungan sebelumnya nilai *GainRation* atribut hubungan sosial lebih besar daripada atribut-atribut lainnya. Sehingga atribut hubungan sosial dijadikan node 1.2. Atribut hubungan sosial dan mempunyai dua cabang, yaitu baik dan buruk. Gambar 5.7 menunjukkan node 1.2 pada dataset pengajuan kredit menggunakan algoritme C4.5. Pada cabang hubungan sosial baik dihasilkan kelas yaitu layak, sedangkan pada cabang hubungan sosial buruk dihasilkan kelas yaitu tidak layak.



Gambar 5.7 Pohon Keputusan Dataset Pengajuan Kredit

5.4 Algoritme Classification and Regression Tree (CART)

Pada sub-bab sebelumnya telah dibahas algoritme ID3 dan C4.5. Pembentukan cabang pada algoritme ID3 dan C4.5 berbasis pada penghitungan nilai *entropy*, berbeda degan algoritme Classification and Regression Tree (CART), algoritme CART menggunakan penghitungan *IndexGini* untuk pembentukan cabang. Sedangkan untuk pembentukan node, pada algoritme CART digunakan penghitungan *GiniGain*. Formula penghitungan *IndexGini* dan *Gini Gain* dapat dilihat pada persamaan 5.5 dan 5.6

$$IndexGini = 1 - \sum_{i=1}^k p_i^2 \quad (5.5)$$

$$Gini\ Gain = Gini(A, S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Gini(S_i) \quad (5.6)$$

Berikut adalah langkah-langkah penghitungan manual algoritme CART:

1. Siapkan Dataset

Dataset yang digunakan adalah dataset penerimaan karyawan, yang dapat dilihat pada Tabel 5.12. Dataset penerimaan karyawan terdiri dari empat atribut dan satu kelas, dan tipe data semua atribut adalah nominal.

2. Hitung Nilai *IndexGini*

$$IndexGini\ (diterima) = 1 - ((5/12)^2 + (7/12)^2) = 0,486$$

$$IndexGini\ (status\ menikah) = 1 - ((4/7)^2 + (3/7)^2) = 0,490$$

$$IndexGini\ (status\ lajang) = 1 - ((1/5)^2 + (4/5)^2) = 0,320$$

Jika semua atribut dihitung nilai *IndexGini*, maka didapatkan hasil seperti yang terlihat pada Tabel 5.13.

3. Hitung Nilai *Gini Gain*

$$Gini\ Gain(Status) = 0,486 - ((7/12 * 0,490) + (5/12 * 0,320)) = 0,067$$

$$Gini\ Gain(Test) = 0,486 - ((3/12 * 0,444) + (8/12 * 0,500) + (1/12 * 0)) = 0,067$$

$$Gini\ Gain(Hub.\ Sosial) = 0,486 - ((7/12 * 0,490) + (5/12 * 0,480)) = 0$$

$$Gini\ Gain(Kinerja) = 0,486 - ((8/12 * 0,469) + (4/12 * 0)) = 0,174$$

Tabel 5.12 Dataset Penerimaan Karyawan

Status	Test Tertulis	Hubungan Sosial	Kinerja	Diterima
Menikah	cukup	Buruk	Baik	Ya
Menikah	Baik	Buruk	Buruk	Tidak
Menikah	cukup	Baik	Baik	Ya
Menikah	Baik	Baik	Baik	Ya
Menikah	Kurang	Baik	Baik	Tidak
Menikah	cukup	Buruk	Baik	Ya
Menikah	Cukup	Baik	Buruk	Tidak
Lajang	cukup	Baik	Baik	Ya
Lajang	Baik	Baik	Buruk	Tidak
Lajang	Cukup	Baik	Buruk	Tidak
Lajang	cukup	Buruk	Baik	Tidak
Lajang	cukup	Buruk	Baik	Tidak

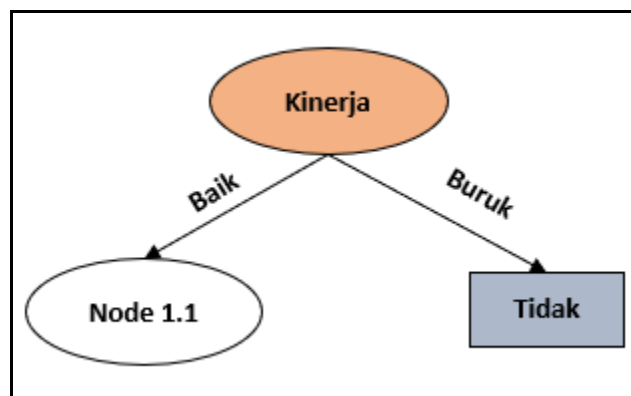
Tabel 5.13 Hasil Penghitungan *IndexGini* Semua Atribut Penerimaan Karyawan

Atribut	Kategori	Jumlah Kasus	Ya	Tidak	Index Gini
Diterima		12	5	7	0,486
Status					
	Menikah	7	4	3	0,490
	Lajang	5	1	4	0,320
Test Tertulis					
	Baik	3	1	2	0,444
	Cukup	8	4	4	0,500
	Kurang	1	0	1	-
Hubungan Sosial					
	Baik	7	3	4	0,490
	Buruk	5	2	3	0,480
Kinerja					
	Baik	8	5	3	0,469
	Buruk	4	0	4	0

4. Membuat Node dan Cabang dari Nilai *Gini Gain* Maksimal

Pada langkah 3 di atas, kita telah menghitung nilai *Gini Gain*. Dari nilai *Gini Gain* tersebut dipilih nilai *Gini Gain* maksimal di antara masing-masing atribut. Nilai *Gini Gain* maksimal adalah *Gini Gain*(Kinerja) dengan nilai 0,174, sehingga atribut **Kinerja** menjadi **node akar**, lihat Gambar 5.8.

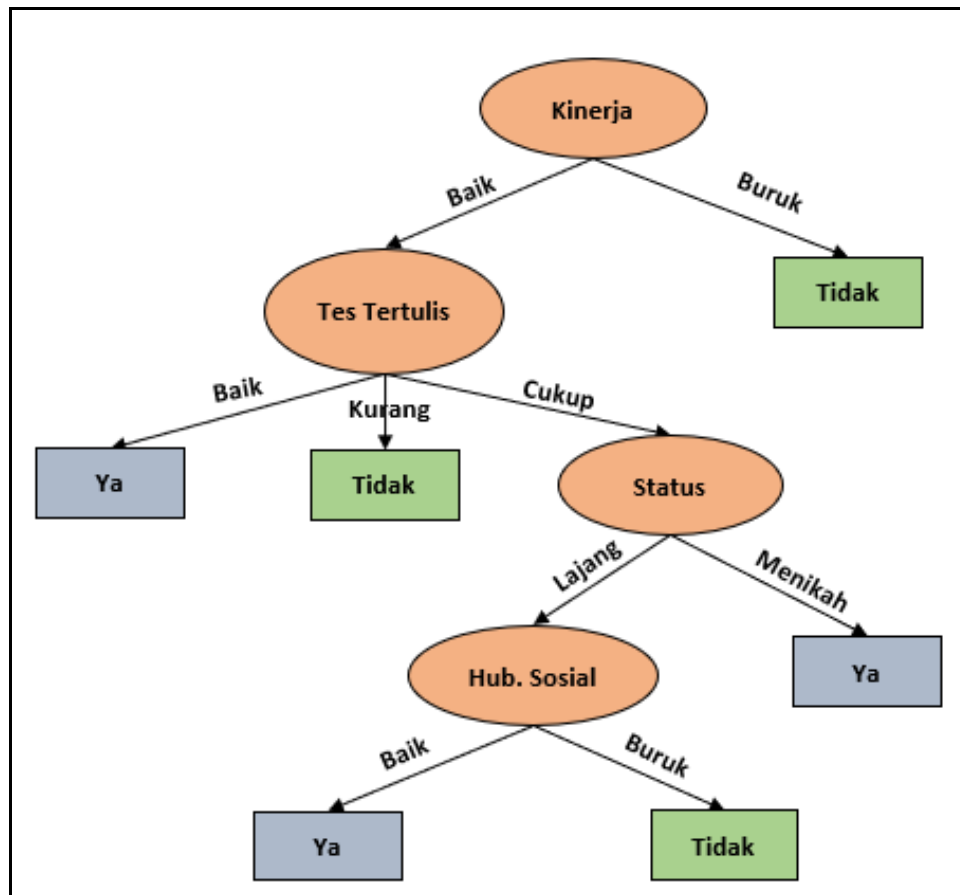
Gambar 5.8, menunjukkan pembentukan node akar, yaitu atribut Kinerja. Cabang yang terbentuk berisi data baik, dan buruk. Pada kategori kinerja buruk, nilai *IndexGini* adalah 0 (nol), sehingga pada cabang kinerja buruk terbentuk **leaf node**. Atribut kinerja buruk, nilai frekuensi tidak (sejumlah 4) lebih tinggi dibandingkan frekuensi ya (sejumlah 4), sehingga dapat disimpulkan bahwa nilai kelas pada kinerja buruk adalah tidak. Untuk kategori kinerja baik belum terbentuk leaf node, karena nilai *IndexGini* tidak sama dengan 0 (nol), sehingga akan dilakukan penghitungan kembali.



Gambar 5.8 Node Akar Dataset Penerimaan Karyawan

5. Ulangi Langkah (2) Sampai Langkah (4) Hingga Semua Node Terpartisi

Pohon keputusan yang terbentuk setelah mengulangi langkah (2) sampai langkah (4) maka akan terbentuk pohon keputusan seperti yang terlihat pada Gambar 5.9.



Gambar 5.9 Pohon Keputusan Dataset Penerimaan Karyawan

5.5 Implementasi Algoritme Decision Trees (CART)

Dataset yang digunakan pada implementasi ini adalah dataset bermain GolfNominal. Berikut adalah langkah-langkah impelentasi algoritme CART menggunakan bahasa pemrograman PHP:

1. Buat file dengan nama **cart.php**.
2. Simpan file tersebut ke dalam folder ...\xampp\htdocs\belajar-phpdm\public.
3. Ketikkan source code cart.php, seperti terlihat pada Gambar 5.10.

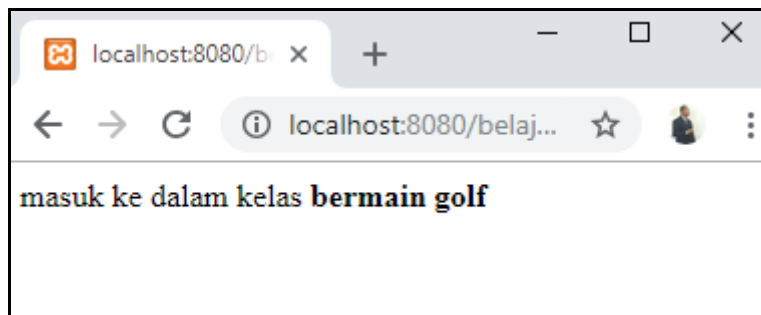
Hasil running source code cart.php pada browser dapat dilihat pada Gambar 5.11. Terlihat bahwa data testing cuaca = cerah, temperature = normal, kelembaban = normal, dan angin = tidak, masuk ke dalam kelas bermain golf.

```

1  <?php
2  require '../vendor/autoload.php';
3  use jokodm\Datamining\Klasifikasi\DecisionTree;
4  use jokodm\Datamining\Dataset\Demo\GolfNominal;
5
6  $dataset = new GolfNominal();
7
8  $klasifikasi = new DecisionTree();
9  $klasifikasi->train($dataset->getSamples(), $dataset->getTargets());
10
11 $pred = $klasifikasi->predict(['hujan', 'normal', 'normal', 'tidak']);
12
13 if ($pred == "Main") {
14     echo "masuk ke dalam kelas <b>bermain golf</b>";
15 } else {
16     echo "masuk ke dalam kelas <b>tidak bermain golf</b>";
17 }

```

Gambar 5.10 Source Code cart.php



Gambar 5.11 Hasil pada Browser cart.php

5.6 Studi Kasus Sistem Cerdas untuk Prediksi Penerimaan Karyawan

Tersedia dalam bentuk buku cetak berbayar (on progress)

DAFTAR PUSTAKA

- Haixiang, G., Yijing, L., Yanan, L., Xiao, L., & Jinling, L. (2015). BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence*, 1–18. <https://doi.org/10.1016/j.engappai.2015.09.011>
- Han and Kamber. (2012). *Data Mining Concepts and Techniques Third Edition*. Elsevier and Morgan Kaufmann (Vol. 1). <https://doi.org/10.1017/CBO9781107415324.004>
- Harrington, P. (2012). *Machine Learning in Action*. United States of America: Manning Publications Co.
- <https://apjii.or.id/>. (2016). Indonesia Internet Users. Retrieved from <https://apjii.or.id/>
- Kantardzic, M. (2011). *Data Mining: Concepts, Models, Methods, and Algorithms* (Second Edi). Canada: A John Willey & Sons, Inc., Publication. Retrieved from <http://ieeexplore.ieee.org/xpl/bkabstractplus.jsp?bkn=5265979>
- Larose, D. T. (2005). *Discovering knowledge in data*. Canada: A John Willey & Sons, Inc., Publication. <https://doi.org/10.1017/CBO9781107415324.004>
- Liu, H., & Zhang, S. (2012). Noisy data elimination using mutual k-nearest neighbor for classification mining. *Journal of Systems and Software*, 85(5), 1067–1074. <https://doi.org/10.1016/j.jss.2011.12.019>
- Noersasongko, E., & Andono, P. N. (2010). *Mengenal Dunia Komputer*. Jakarta: Elex Media Komputindo.
- Ohlhorst, F. (2013). *Big Data Analytics. Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Sáez, J. a., Galar, M., Luengo, J., & Herrera, F. (2013). Tackling the problem of classification with noisy data using Multiple Classifier Systems: Analysis of the performance and robustness. *Information Sciences*, 247(June), 1–20. <https://doi.org/10.1016/j.ins.2013.06.002>
- Suntoro, J., & Indah, C. N. (2017). Average Weight Information Gain Untuk Menangani Data Berdimensi. *Jurnal Buana Informatika*, 8, 131–140.
- Witten, Ian H. Frank, E. H. (2011). *Data Mining Practical Machine Learning Tools and Techniques*. Elsevier and Morgan Kaufmann (Vol. 1). <https://doi.org/10.1017/CBO9781107415324.004>
- Yang, S., Guo, J. Z., & Jin, J. W. (2018). An improved Id3 algorithm for medical data classification. *Computers and Electrical Engineering*, 65, 474–487. <https://doi.org/10.1016/j.compeleceng.2017.08.005>

TENTANG PENULIS



Joko Suntoro. Lahir pada tanggal 31 Juli 1989 di Kota Semarang, merupakan putra pertama dari pasangan ibu Solechah dan bapak (alm.) Jamasri. Menyelesaikan pendidikan SD tahun 2001 di SDN Pandean Lamper 03 Semarang, kemudian melanjutkan pendidikan di SMPN 32 Semarang lulus tahun 2004. Setelah itu melanjutkan pendidikan di SMA Institut Indonesia, lulus tahun 2007. Memperoleh gelar S.Kom pada jurusan Teknik Informatika di Universitas Semarang pada tahun 2015, dan gelar M.Kom pada jurusan Magister Teknik Informatika di Universitas Dian Nuswantoro, Semarang tahun 2016. Sejak tahun 2008 tergabung dalam tim operasi Domestic Gas Region IV. Selain itu, saat ini aktif sebagai dosen teknik informatika di Universitas Semarang dan tergabung dalam tim penelitian Intelligent Systems Research Group yang dipimpin oleh Romi Satria Wahono, Ph.D. Bidang penelitian penulis adalah data mining, software engineering, dan machine learning.