

BAB II

LANDASAN TEORI

2.1 Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem informasi berbasis komputer yang interaktif, fleksibel, dan mampu beradaptasi yang secara khusus dikembangkan untuk mendukung solusi dari suatu permasalahan yang tidak terstruktur untuk meningkatkan kualitas pengambilan keputusan (Turban, 1995). Sistem pendukung keputusan merupakan suatu sistem berbasis komputer yang interaktif yang membantu pengguna dalam kegiatan penilaian dan pemilihan (Druddzel dan flynn, 2002). Terdapat 3 komponen dasar dari sistem pendukung keputusan (Druddzel dan Flynn, 2002).

1. *Database Management System (DBMS)*, menyediakan tempat penyimpanan data untuk sistem pendukung keputusan. DBMS menyimpan data dalam jumlah yang besar. DBMS memisahkan aspek fisik dengan struktur dan proses dalam *database*.
2. *Model-base Management System (MBMS)*, tujuannya adalah mengubah data dari DBMS menjadi suatu informasi yang berguna dalam pengambilan keputusan.
3. *Dialog Generation and Management System (DGMS)*, untuk menambah kemampuan atau kualitas dari *system user* menjadi bermanfaat dari suatu sistem pendukung keputusan.

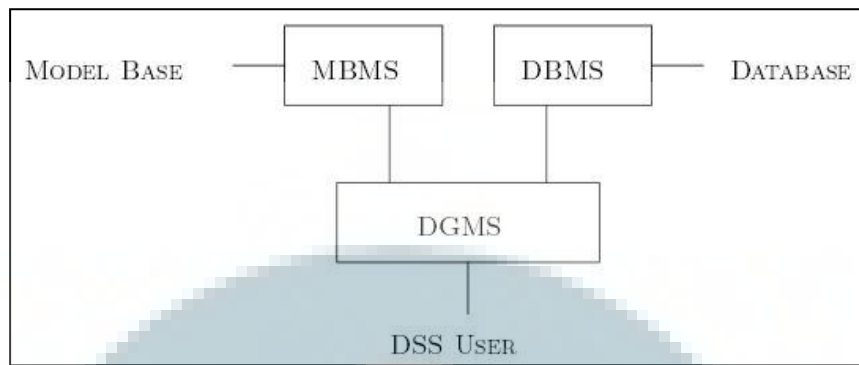


Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.



Gambar 2.1 Arsitektur DSS
(Drudzel dan Flynn, 2002)

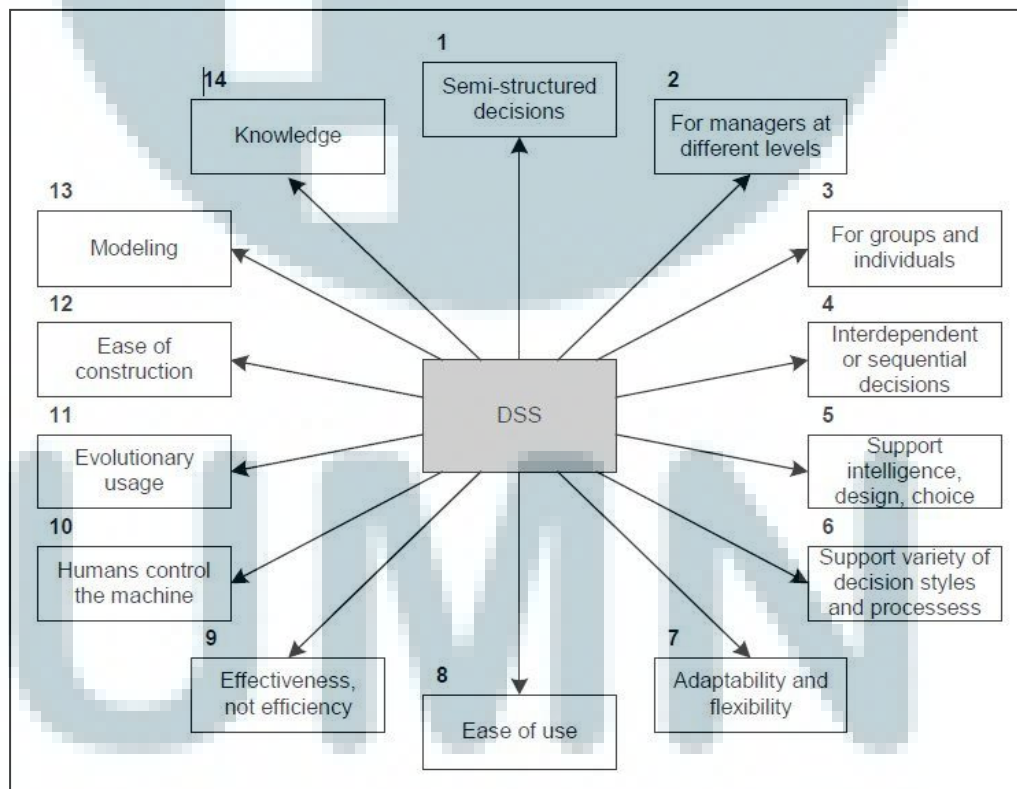
Sistem pendukung keputusan didefinisikan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, yaitu sistem bahasa yang merupakan suatu mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan yang lain, sistem pengetahuan yang merupakan sekumpulan paket pengetahuan dari suatu masalah yang ada pada sistem pendukung keputusan baik sebagai data maupun sebagai prosedur, sistem pengolahan masalah merupakan hubungan antara dua komponen lainnya yang terdiri dari satu atau lebih kemampuan dalam memanipulasi masalah umum yang diperlukan untuk pengambilan keputusan (Bonczek, 1981).

Tahap-tahap yang harus dilalui dalam proses pengambilan keputusan adalah sebagai berikut (Simon, 1955).

1. Tahap Pemahaman (*Intelligence Phase*), tahap ini merupakan proses penelusuran dan pendeteksian dari suatu lingkup masalah serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.
2. Tahap Perancangan (*Design Phase*), tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan atau solusi yang dapat diambil. Solusi tersebut merupakan representasi kejadian nyata yang

disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan (*Choice Phase*), tahap ini dilakukan pemilihan terhadap berbagai alternatif solusi yang ditampilkan pada tahap perencanaan agar dapat ditentukan dengan memperhatikan kriteria-kriteria berdasarkan tujuan yang akan dicapai.
4. Tahap Implementasi (*Implementation Phase*), tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan. Berikut karakteristik sistem pendukung keputusan yang dapat dilihat pada Gambar 2.2.



Gambar 2.2 Kerangka Karakteristik Sistem Pendukung Keputusan (Turban, 1995)

Berdasarkan Gambar 2.2, terdapat 14 karakteristik dalam sistem pendukung keputusan sebagai berikut.

1. Dukungan pengambilan keputusan untuk keadaan semi terstruktur.
2. Dukungan untuk level manajerial yang berbeda-beda dari level paling atas sampai level paling bawah.
3. Dukungan untuk individual dan kelompok.
4. Dukungan untuk semua keputusan yang saling bergantung atau sekuensial.
5. Dukungan untuk pengambilan keputusan dalam intelegensi, desain, dan proses pemilihan.
6. Dukungan terhadap berbagai macam proses dan tipe keputusan.
7. Kemampuan sistem beradaptasi terhadap masalah-masalah baru dan fleksibilitas dari sistem dalam menghadapi perubahan-perubahan tersebut.
8. Mudah digunakan dan *userfriendly*.
9. Efektivitas sistem dalam proses pengambilan keputusan daripada efisiensi.
10. *User* yang mengontrol sistem dalam proses pengambilan keputusan.
11. Sistem dapat dikembangkan lebih lanjut oleh *user*.
12. Perancangan sistem yang sederhana.
13. Menggunakan berbagai macam model dalam menganalisis proses pengambilan keputusan.
14. Kemampuan sistem dalam berintegrasi.

Beberapa keuntungan dari sistem pendukung keputusan yaitu, kemampuan dalam mencari suatu solusi terbaik dari suatu masalah yang kompleks, mampu menerapkan strategi yang berbeda-beda pada konfigurasi yang berbeda pula secara cepat dan tepat, meningkatkan sistem *monitoring* manajemen dan kinerja,

menghemat biaya serta meningkatkan produktivitas dari suatu analisis (Subakti, 2002).

2.2 Makanan dan Pola Makan

Makanan merupakan suatu kebutuhan pokok dalam menunjang kehidupan sehari – hari sebagai penambah energi dan semangat dalam melakukan suatu pekerjaan. Berdasarkan Menkes RI (2014), pola makan merupakan perilaku paling penting yang dapat mempengaruhi keadaan gizi. Hal ini disebabkan karena kuantitas dan kualitas makanan ataupun minuman yang dikonsumsi akan mempengaruhi kesehatan individu dan masyarakat. Pola makan yang tidak baik dapat menyebabkan berbagai masalah kesehatan pada tubuh. Pola makan dapat diartikan sebagai suatu kebiasaan menetap dalam hubungan dengan konsumsi makanan (Almatsier, 2002). Salah satu cara hidup sehat adalah mengatur pola makan yang disesuaikan dengan kebutuhan tubuh. Pengaturan pola makan yang disesuaikan dengan kebutuhan tubuh harus memperhatikan kalori yang boleh dikonsumsi oleh seseorang (Stephanie, 2008).

Kebutuhan kalori yang dibutuhkan dapat dihitung berdasarkan jenis kelamin, usia, tinggi dan berat badan, serta tingkat aktivitas yang dilakukan.

Langkah pertama adalah menghitung kebutuhan kalori dengan rumus BMR (Basal Metabolic Rate) berdasarkan persamaan Harris Benedict yang dapat dijabarkan sebagai berikut (Harris dan Benedict, 1919).

$$\text{Laki-laki} = 66 + (13,7 \times \text{berat (kg)}) + (5 \times \text{tinggi (cm)}) - (6,8 \times \text{usia})$$

$$\text{Perempuan} = 655 + (9,5 \times \text{berat (kg)}) + (1,8 \times \text{tinggi (cm)}) - (4,7 \times \text{usia}) \dots(2.1)$$

Langkah kedua adalah dengan menentukan tingkat aktivitas seseorang. Salah satu cara untuk mengukur intensitas dari aktivitas adalah dengan MET (Metabolic Equivalent Task) (Stephanie, 2008). Tingkat aktivitas dan contohnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Aktivitas dan Tingkat Aktivitas
(www.topendsports.com)

Tingkat Aktivitas	MET /Jam	Aktivitas
Ringan	1,3	Berdiri
	1,5	Membaca, berkomunikasi
	1,8	Belajar, duduk, mencatat
	2	Berjalan santai, bermain musik, memasak, pekerjaan rumah tangga, menjaga toko.
Sedang	3	Berkendara, naik turun tangga, mengangkat barang ringan, pekerjaan kantor.
	4	Bersepeda santai (kurang dari 2 km/jam), mengecat, pekerjaan tukang
	4,5	Berenang
	5	Berjalan cepat, bersepeda (2-4 km/jam)
Berat	6	Olahraga (tenis, basket, voli, dll), mendaki gunung
	7	Jogging
	8	Kegiatan aerobik (push up, sit up, dll)
	10	Berlari (8-10 km/jam)
	13,5	Berlari (10-12 km/jam)

Setiap tingkat aktivitas memiliki konstanta tersendiri dalam perhitungan kebutuhan kalori harian (Stephanie, 2008). Tabel konstanta dapat dilihat pada Tabel 2.2.

Tabel 2.2 Konstanta Aktivitas
(Stephanie, 2008)

Aktivitas	Laki-laki	Perempuan
Ringan	1,3	1,3
Sedang	1,65	1,55
Berat	1,76	1,7

Langkah ketiga adalah menentukan BMI (Body Mass Index) yang dapat dihitung dengan rumus (2.2).

$$\text{BMI} = \frac{\text{Berat badan(kg)}}{(\text{Tinggi badan (m)})^2} \quad \dots(2.2)$$

Dari hasil perhitungan, maka dapat diketahui proporsi tubuh yang dikategorikan sebagai berikut (World Health Organization (WHO)).

- Underweight = kurang dari 18,5
- Normal = 18,5 – 24,9
- Overweight = lebih dari atau sama dengan 25

Berdasarkan kategori tersebut, perlu dilakukan penyesuaian kebutuhan kalori sebagai berikut (Stephanie, 2008).

- Underweight = kalori + 500
- Normal = tidak perlu penyesuaian
- Overweight = kalori – 500

Langkah terakhir adalah menghitung kebutuhan kalori yang dibutuhkan perhari berdasarkan tingkat aktivitas dan BMI. Contoh perhitungan kalori dari seorang laki-laki 25 tahun dengan berat 70 kg dan tinggi 175 cm dan tingkat aktivitas sedang.

$$\text{BMR} = 66 + (13,7 \times 70) + (5 \times 175) - (6,8 \times 25) = 1730 \text{ kkal}$$

$$\text{Kebutuhan Kalori} = 1730 \times 1,65 = 2854,5$$

$$\text{BMI} = 70 / (1,75)^2 = 22,8 \text{ (Normal)}$$

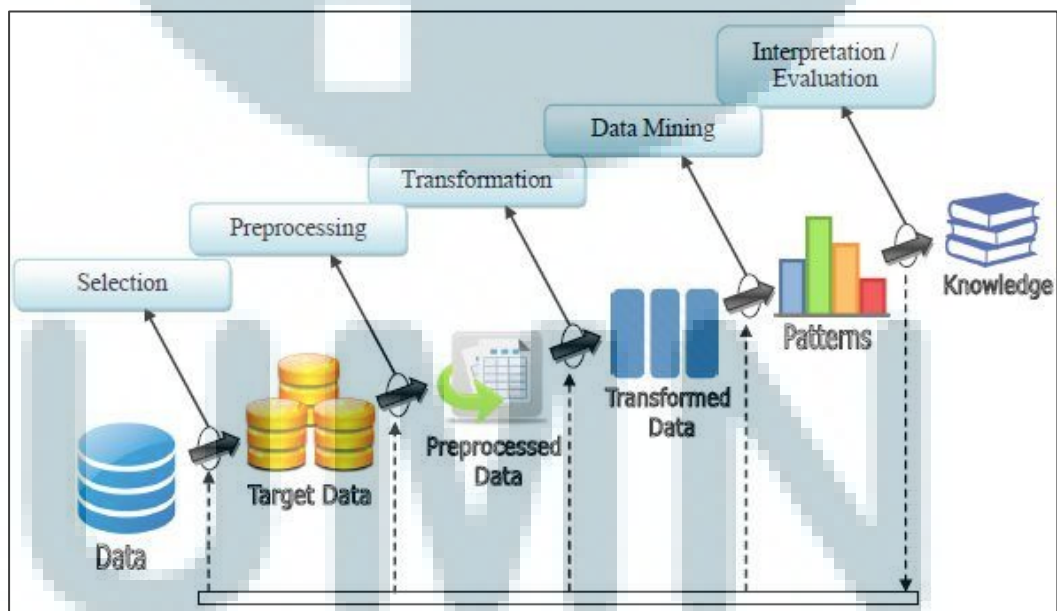
Kebutuhan kalori perhari yang diperlukan adalah 2854,5.

Kebutuhan kalori perhari seseorang tidak kurang dari 10 persen total kebutuhan kalori perhari. Misalnya, kebutuhan kalori seseorang adalah 2500 kkal, maka kebutuhan kalorinya antara 2250 – 2500 kkal perhari (Stephanie, 2008).

2.3 Data Mining

Data sekarang ini menjadi bagian yang sangat penting bagi kehidupan sehingga diperlukan suatu cara untuk mengelola data-data dalam jumlah yang besar tersebut. *Data mining* dapat didefinisikan sebagai proses dalam menemukan pola-pola dalam suatu data. Pola-pola yang ditemukan harus berarti dan dapat memberikan beberapa keuntungan. *Data mining* menyelesaikan masalah dengan menganalisis data yang telah tersedia dalam suatu *database* (Witten, 2011).

Data mining merupakan tahap analisis dalam proses *Knowledge Discovery in Databases* (KDD) yang mencari suatu pola baru dari suatu dataset yang besar. KDD telah berkembang dan terus berkembang dari penelitian-penelitian seperti, *machine learning*, *pattern recognition*, *databases*, *statistics*, *Artificial Intelligent*, *data visualization* dan lainnya (Fayyad, 1996). Berikut proses *Knowledge Discovery in Databases* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Proses KDD
(Fayyad, 1996).

Tahapan-tahapan dalam proses KDD dapat didefinisikan sebagai berikut (Fayyad, 1996).

1. *Selection*, melakukan seleksi terhadap suatu data set yang bertujuan untuk menentukan data set mana yang akan digunakan. Selanjutnya, menentukan variabel atau data sampel utama yang akan digunakan.
2. *Preprocessing*, melakukan operasi dasar yang meliputi penghilangan *noise* jika diperlukan, mengumpulkan informasi yang dibutuhkan untuk mengatasi *noise*, menentukan cara-cara untuk menangani data yang hilang, dan mengelola informasi serta perubahan-perubahan yang terjadi secara berkala.
3. *Transformation*, melakukan pengurangan (reduksi) data yang berguna untuk mengurangi variabel yang masih diperhitungkan atau variabel yang tidak berhubungan dapat ditemukan.
4. *Data mining*, mencari pola-pola yang menarik dari suatu data set berdasarkan metode-metode seperti aturan klasifikasi atau *tree*, regresi dan *clustering*.
5. *Interpretation / Evaluation*, melakukan evaluasi secara keseluruhan dari tahap-tahap sebelumnya.

2.4 Naive Bayes

Naive Bayes menyediakan algoritma pembelajaran praktis, pengetahuan sebelumnya (*prior knowledge*) dan data yang diamati dapat dikombinasikan. Naive Bayes memberikan perspektif yang berguna untuk memahami dan mengevaluasi banyak algoritma pembelajaran, menghitung probabilitas eksplisit untuk hipotesis dan bagus pada data yang memiliki *noise* yang tinggi. Metode ini terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam *database* dengan data yang besar (Susanto, 2013). Naive Bayes melakukan suatu

pendekatan dengan membuat asumsi sederhana bahwa semua atribut adalah independen. Pendekatan ini membawa dampak yang lebih sederhana dan efektif dalam pengklasifikasiannya (Zaki, 2014).

Naive Bayes mengklasifikasikan suatu data dengan metode probabilitas dan statistik dengan memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya (teorema *Bayes*). Metode ini dapat diasumsikan bahwa ada atau tidaknya suatu ciri tertentu dari suatu *class* tidak ada hubungannya dengan ciri dari *class* lainnya. Berikut persamaan umum teorema *Bayes* (Bustami, 2014).

$$P(H|X) = \frac{P(X|H) P(H)}{P(X)} \quad \dots(2.3)$$

Dimana :

X = suatu kejadian atau *event*.

H = suatu *class* data atau kondisi.

$P(H|X)$ = peluang suatu *class* H berdasarkan kejadian X.

$P(H)$ = peluang suatu *class* data atau kondisi.

$P(X|H)$ = peluang kejadian X berdasarkan kondisi H.

$P(X)$ = peluang suatu kejadian X.

Algoritma Naive Bayes akan memprediksi peluang terjadinya suatu kejadian x jika kondisi h terpenuhi ($P(h_i|x_j)$), dengan perkiraan suatu *class* h jika diberikan nilai dari atribut $h_1, h_2, \dots, h_n$ (Wiratama, 2012). Berikut cara kerja algoritma Naive Bayes.

Hitung semua peluang dari setiap kondisi h yang diinginkan terhadap semua pilihan kejadian x yang ada berdasarkan rumus 2.4 (Meisner, 2003).

$$P(h_i|x_j) = \frac{n_{c+a.p}}{n+a} \quad \dots(2.4)$$

dimana :

n = jumlah data yang memenuhi suatu kejadian x .

n_c = jumlah data yang memenuhi suatu kejadian x dan dengan kondisi h .

p = 1 / jumlah pilihan kejadian x .

a = bernilai bebas dan konsisten untuk setiap perhitungan.

Untuk setiap pilihan kejadian x , hitung nilai peluang terjadinya kejadian x_1 , x_2 , ..., x_n dengan persyaratan kondisi $h_1, h_2, \dots, h_n$ dengan rumus 2.5 berikut.

$$P(x_n) * P(h_1|x_n) * P(h_2|x_n) * \dots * P(h_n|x_n) \quad \dots(2.5)$$

Nilai Peluang suatu kejadian x adalah prediksi yang dihasilkan oleh algoritma Naive Bayes dengan persyaratan kondisi $h_1, h_2, \dots, h_n$ yang telah diberikan sebelumnya.

Berikut contoh tahapan-tahapan penggunaan Naive Bayes untuk memprediksi kelulusan mahasiswa.

1. Membuat suatu data set dalam bentuk tabel yang berisi data-data yang mendeskripsikan hasil yang ingin diperoleh.

Tabel 2.3 Contoh Data Set
(Rahman, 2013)

No	Jenis Kelamin	Status Mahasiswa	Status Pernikahan	IPK Semester 1-6	Status Kelulusan
1	Pria	Mahasiswa	Belum	3.17	Tepat
2	Pria	Bekerja	Belum	3.30	Tepat
3	Wanita	Mahasiswa	Belum	3.01	Tepat
4	Wanita	Mahasiswa	Menikah	3.25	Tepat
5	Pria	Bekerja	Menikah	3.20	Tepat
6	Pria	Bekerja	Menikah	2.50	Terlambat
7	Wanita	Bekerja	Menikah	3.00	Terlambat
8	Wanita	Bekerja	Belum	2.70	Terlambat
9	Pria	Bekerja	Belum	2.40	Terlambat
10	Wanita	Mahasiswa	Menikah	2.50	Terlambat
11	Wanita	Mahasiswa	Belum	2.50	Terlambat

12	Wanita	Mahasiswa	Belum	3.50	Tepat
13	Pria	Bekerja	Menikah	3.30	Tepat
14	Pria	Mahasiswa	Menikah	3.25	Tepat
15	Pria	Mahasiswa	Belum	2.30	Terlambat

Berdasarkan Tabel 2.3, akan dilakukan prediksi kelulusan seorang mahasiswa dengan menggunakan metode Naive Bayes. Parameter yang digunakan adalah jenis kelamin wanita, status mahasiswa bekerja, status pernikahan belum menikah dan ipk 3.00.

2. Menghitung peluang dari setiap nilai atau kondisi h yang diinginkan terhadap semua pilihan kejadian x yang ada.

h = wanita, bekerja, belum menikah, ipk 3.00.

x = tepat atau terlambat.

Hitung peluang dari :

$P(\text{Wanita}|\text{Tepat})$, $P(\text{Bekerja}|\text{Tepat})$, $P(\text{Belum Menikah}|\text{Tepat})$, $P(\text{IPK 3.00}|\text{Tepat})$

$P(\text{Wanita}|\text{Terlambat})$, $P(\text{Bekerja}|\text{Terlambat})$, $P(\text{Belum Menikah}|\text{Terlambat})$, $P(\text{IPK 3.00}|\text{Terlambat})$

dengan ketentuan, $a = 4$ dan $p = 1 / \text{jumlah } x$.

$$(P | \text{Jenis Kelamin}) = 1 / 2 = 0,5$$

$$(P | \text{Status Mahasiswa}) = 1 / 2 = 0,5$$

$$(P | \text{Status Pernikahan}) = 1 / 2 = 0,5$$

$$(P | \text{IPK}) = 1 / 11 = 0,09$$

$$(P | \text{Status kelulusan}) = 1 / 2 = 0,5$$

$$(\text{Wanita} | \text{Tepat}) : n = 7, n_c = 3$$

$$(\text{Wanita} | \text{Terlambat}) : n = 7, n_c = 4$$

$$(\text{Bekerja} | \text{Tepat}) : n = 7, n_c = 3$$

(Bekerja | Terlambat) : $n = 7, n_c = 4$

(Belum Menikah | Tepat) : $n = 8, n_c = 4$

(Belum Menikah | Terlambat) : $n = 8, n_c = 4$

(IPK 3.00 | Tepat) : $n = 1, n_c = 0$

(IPK 3.00 | Terlambat) : $n = 1, n_c = 1$

Maka :

$$P(\text{Wanita} | \text{Tepat}) = \frac{3 + 4 * 0.5}{7 + 4} = 0.45$$

$$P(\text{Wanita} | \text{Terlambat}) = \frac{4 + 4 * 0.5}{7 + 4} = 0.54$$

$$P(\text{Bekerja} | \text{Tepat}) = \frac{3 + 4 * 0.5}{7 + 4} = 0.45$$

$$P(\text{Bekerja} | \text{Terlambat}) = \frac{4 + 4 * 0.5}{7 + 4} = 0.54$$

$$P(\text{Belum Menikah} | \text{Tepat}) = \frac{4 + 4 * 0.5}{8 + 4} = 0.5$$

$$P(\text{Belum Menikah} | \text{Terlambat}) = \frac{4 + 4 * 0.5}{8 + 4} = 0.5$$

$$P(\text{IPK 3.00} | \text{Tepat}) = \frac{0 + 4 * 0.09}{1 + 4} = 0.072$$

$$P(\text{IPK 3.00} | \text{Terlambat}) = \frac{1 + 4 * 0.09}{1 + 4} = 0.27$$

3. Menghitung peluang terjadinya kejadian $x_1, x_2, \dots, x_n$ dengan persyaratan kondisi $h_1, h_2, \dots, h_n$. Pilihan kejadian x hanyalah x_1 dan x_2 yaitu tepat dan terlambat sehingga

$$P(\text{Tepat}) = 8 / 15 = 0,53$$

$$P(\text{Terlambat}) = 7 / 15 = 0,46$$

Untuk $x = \text{Tepat}$

$$P(\text{Tepat}) * P(\text{Wanita} | \text{Tepat}) * P(\text{Bekerja} | \text{Tepat}) * P(\text{Belum Menikah} | \text{Tepat}) \\ * P(\text{IPK 3.00} | \text{Tepat})$$

$$= 0.53 * 0.45 * 0.45 * 0.5 * 0.072$$

$$= 0.003$$

Untuk $x = \text{Terlambat}$

$$P(\text{Terlambat}) * P(\text{Wanita} | \text{Terlambat}) * P(\text{Bekerja} | \text{Terlambat}) * P(\text{Belum Menikah} | \text{Terlambat}) * P(\text{IPK 3.00} | \text{Terlambat})$$

$$= 0.46 * 0.54 * 0.54 * 0.5 * 0.27$$

$$= 0.018$$

Berdasarkan hasil perhitungan data dengan menggunakan algoritma Naive Bayes, karena $0.018 > 0.003$, maka prediksi kelulusan dari contoh kasus adalah terlambat.

2.4.1 Equal Frequency Discretization (EFD)

Diskritisasi merupakan suatu pendekatan yang cukup dikenal dalam mengatasi atribut-atribut yang bersifat numerik dalam *machine learning*. Proses klasifikasi sering kali melibatkan atribut-atribut yang bersifat numerik. Atribut-atribut numerik dalam Naive Bayes biasanya diproses dengan metode diskritisasi sebagai acuan dalam meningkatkan performa klasifikasi saat atribut tersebut didiskritisasi (Yang and Webb, 2002). Salah satu metode yang dapat digunakan dalam proses diskritisasi adalah EFD. EFD membagi nilai yang telah disusun ke dalam suatu k (parameter yang telah didefinisikan) interval sehingga setiap interval mengandung rata-rata angka yang sama dari contoh data *training*. Metode EFD berpotensi kehilangan beberapa informasi karena k ditentukan tanpa referensi terhadap properti dalam data *training*. Meskipun metode ini terlihat sederhana, EFD sering digunakan dan secara tidak langsung dapat bekerja dengan baik untuk Naive Bayes (Yang dan Webb, 2002).

Metode EFD menentukan minimum dan maksimum nilai dari atribut diskritisasi, kemudian disusun dan dibagi dalam suatu interval k dengan setiap interval berisi kira-kira n/k contoh data dengan nilai yang berdekatan (Dash dkk, 2011). Berikut contoh proses diskritisasi dengan menggunakan metode EFD.

Tabel 2.4 Contoh Penyelesaian EFD
(Dash dkk, 2011)

Data values	V1	V2	V3	V4	V5	V6	V7	V8
Original Values	10	50	15	20	12	25	40	30
Sorted Values	10	12	15	20	25	30	40	50
Description	$k = 2, n = 8$ $n/k = 8/2 = 4$							
Discrete Values	1	2	1	1	1	2	2	2

Berdasarkan Tabel 2.4, terdapat 8 nilai awal yang kemudian nilai-nilai tersebut disusun dan selanjutnya dilakukan proses diskritisasi dengan metode EFD. Pada contoh tersebut, variabel k diberi nilai 2 dan variabel n merupakan nilai awal dari data sehingga terdapat 2 kelompok interval yang masing-masing berisi 4 data (n/k).

Dalam jurnal yang berjudul *Heart Disease Prediction System using Naive Bayes and Jelinek-mercer Smoothing* yang dilakukan oleh Patil (2014), pengukuran performa metode Naive Bayes terhadap sistem dilakukan dengan beberapa analisis klasifikasi sebagai berikut (Patil, 2014).

1. Sensitivity, pengukuran tingkat kebenaran yang bersifat positif dengan rumus berikut.

$$\frac{\text{True_Pos}}{\text{Pos}} \quad \dots(2.6)$$

2. Specificity, pengukuran tingkat kebenaran yang bersifat negatif dengan rumus berikut.

$$\frac{\text{True_Neg}}{\text{Neg}}$$

...(2.7)

3. Accuracy, pengukuran tingkat akurasi dengan rumus berikut.

$$\text{Sensitivity} \frac{\text{Pos}}{\text{Pos+Neg}} + \text{Specificity} \frac{\text{Neg}}{\text{Pos+Neg}}$$

...(2.8)

Dimana :

True_Pos = total kebenaran nilai positif dengan kriteria positif.

Pos = total kebenaran positif.

True_Neg = total kebenaran nilai negatif dengan kriteria negatif.

Neg = total kebenaran negatif.

UMN