BAB II

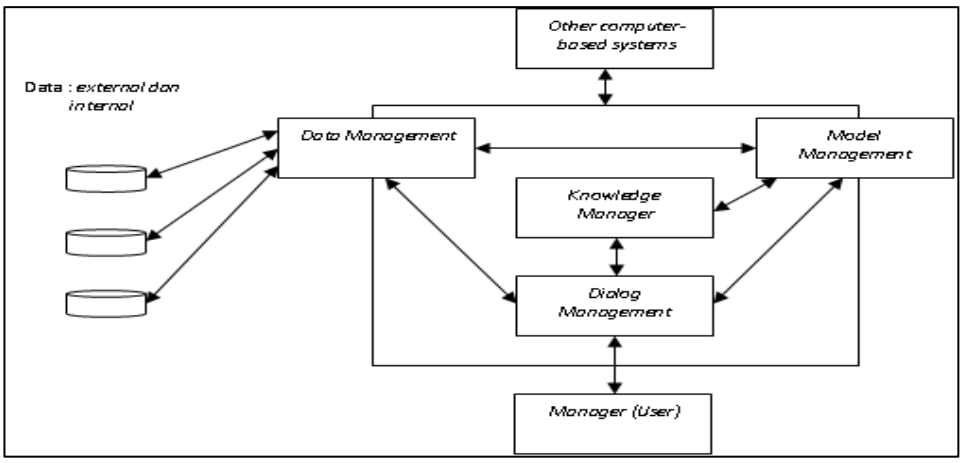
LANDASAN TEORI

2.1 Beasiswa Jalur Tes Reguler UMN

Menurut Bapak Lukman sebagai *Staff Marketing* di UMN (wawancara pribadi, Lukman, 29 April 2019), beasiswa jalur tes regular telah diberikan kepada mahasiswa UMN yang memiliki nilai tes psikotest, nilai tes matematika, dan nilai tes Bahasa Inggris yang memenuhi standar nilai yang dimiliki oleh UMN.

2.2 Sistem Pendukung Keputusan

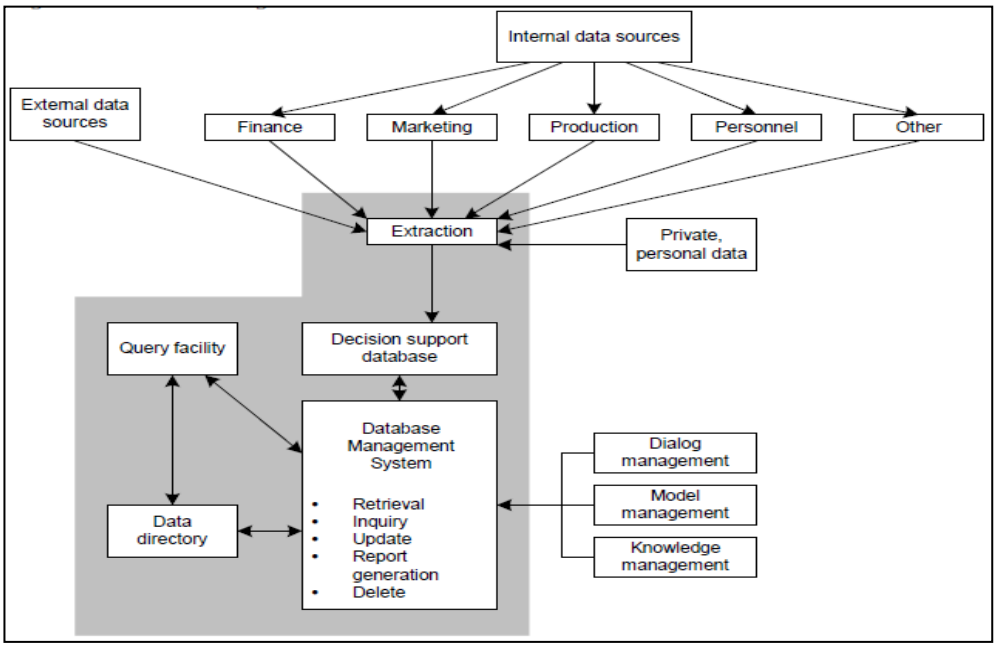
Sistem Pendukung Keputusan (SPK) atau *Decision System Support*, secara umum didefinisikan sebagai sebuah sistem untuk mendukung keputusan semi-terstruktur . SPK dimaksudkan menjadi alat bantu bagi para pengambil keputusan agar kapabilitas mereka dapat diperluas, dengan perangkat yang interaktif namun penilaian para pengambil keputusan tetap tidak digantikan oleh SPK (Agus, 2016). Adapun komponen-komponen dari SPK yaitu (Subakti, 2010):



Gambar 2.1 Model Konseptual Sistem Pendukung Keputusan (Subakti, 2010)

Gambar 2.1 adalah gambaran dari komponen-komponen sebuah sistem pendukung keputusan yang meliputi :

1. *Data Management*

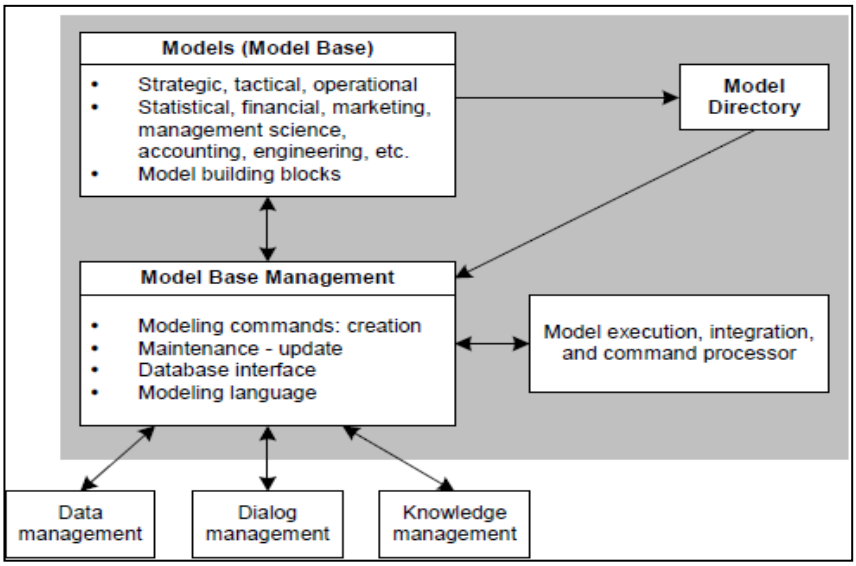


Gambar 2.2 Subsistem *Data Management* (Turban, 2015)

Gambar 2.2 adalah merupakan gambar dari konseptual sistem pendukung keputusan, termasuk *database*, yang mengandung data relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).

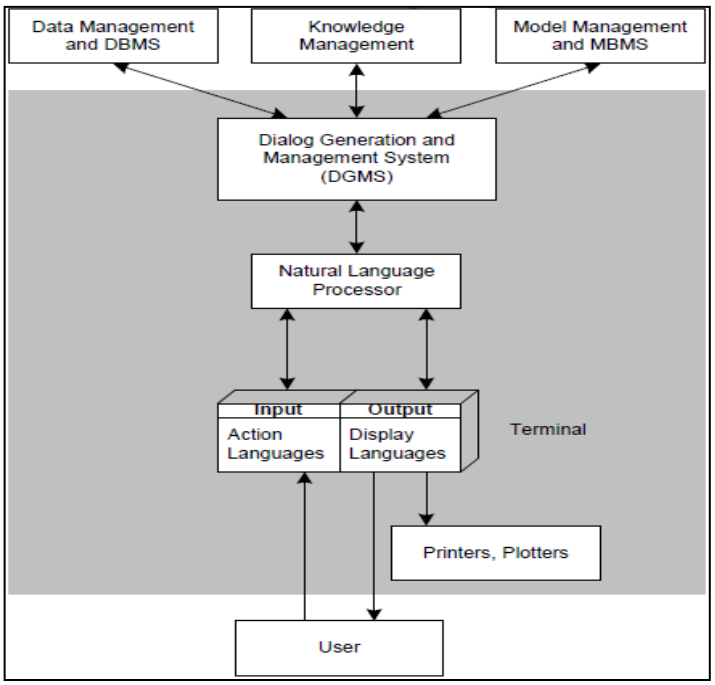
2. *Model Management*

Gambar 2.3 merupakan gambar dari Subsistem *Data Management* yang melibatkan model finansial, statistical, *management science*, atau berbagai model kualitatif lainnya, sehingga dapat memberikan kemampuan analitis dan manajemen perangkat lunak yang dibutuhkan bagi sistem.



Gambar 2.3 Subsistem *Model Management* (Turban, 2015)

3. *Communication*



Gambar 2.4 Skema *Communication* (Turban, 2015)

Gambar 2.4 merupakan gambar dari skema *Communication*, dimana *User* dapat berkomunikasi dan memberikan perintah pada SPK melalui subsitem ini (antarmuka).

4. *Knowledge Management*

Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

2.3 Data Mining

Data Mining adalah sebuah proses yang dilakukan untuk menemukan hubungan, pola, dan tren baru yang bermakna dengan cara menyaring data yang sangat besar dan tersimpan dalam penyimpanan menggunakan teknik pengenalan pola seperti Statistika dan Matematika (Amelia, 2017).

Menurut para ahli, *data mining* adalah suatu analisa dari observasi data dalam jumlah besar untuk menemukan hubungan yang tidak diketahui sebelumnya dan dua metode baru untuk meringkas data agar mudah dipahami serta kegunaannya untuk pemilihan data (Sherekar, 2013).

2.3.1 Pengolahan Data Mining

Pengolahan *data mining* terdiri dari beberapa tahap metode pengolahan, yaitu (Amelia, 2017):

a. *Predictive* Modeling yang merupakan pengolahan data mining dengan melakukan prediksi atau peramalan. Tujuan metode ini untuk membangun model prediksi suatu nilai yang mempunyai ciri-ciri tertentu. Contoh algoritmanya Linear Regression, Neural Network, Support Vector Machine.
b. *Association* (Asosiasi) merupakan teknik dalam data mining yang mempelajari hubungan antar data. Contoh penggunaannya seperti untuk menganalisis perilaku mahasiswa yang datang terlambat. Contohnya jika

mahasiswa memiliki jadwal dengan dosen A dan dosen B, maka mahasiswa akan dating terlambat. Contoh algoritmanya FP-Growth, A-Priori.

c. *Clustering* (Klastering) atau pengelompokan merupakan teknik untuk mengelompokan data ke dalam suatu kelompok tertentu. Contoh untuk klastering: Terdapat lima pulau di Indonesia: Sumatera, Kalimantan, Jawa, Sulawesi, dan Papua. Maka lima pulau tersebut diajdikan tiga klaster berdasarkan zona waktunya: Waktu Indonesia Barat(Sumatera, Kalimantan, dan Jawa), Waktu Indonesia Tengah(Sulawesi), Waktu Indonesia Timur(Papua). Contoh algoritmanya K-means, K-Medoids, Self-Organization, Map(SOM), Fuzzy C-Means, dan lain-lain.
d. *Classification* (Klasifikasi) merupakan teknik mengklasifikasikan data. Perbedaannya dengan klastering terletak pada data, dimana pada klastering variable dependen tidak ada, sedangkan pada klasifikasi diharuskan ada variable dependen. Contoh algoritmanya ID3 dan K-Nearest Neighbors.

2.3.2 Pendekatan Teknik Data Mining

Dalam dunia *data mining* atau *data science*, terdapat 3 pendekatan yaitu (Norfiansyah, 2015) :

a. *Supervised Learning* adalah pendekatan *data mining* dengan data yang telah dilatih, dan terdapat variabel yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengkelompokan suatu data ke data yang sudah ada.
b. *Unsupervised Learning* adalah pendekatan *data mining* dengan data yang tidak dilatih, sehingga dari data yang ada, pengelompokan dilakukan menjadi 2 atau 3 bagian dan seterusnya.

c. *Reinforcement Learning* adalah pendekatan *data mining* dengan adanya agen yang mempelajari sesuatu dengan aksi tertentu dan menerima hasil dari aksi tersebut (belajar berdasarkan pengalaman yang dialami oleh agen tersebut).

2.4. Algoritma Naïve Bayesian Classifier

Algoritma *Naïve Bayesian Classifier* adalah sebuah algoritma pengklasifikasi probalistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan (Maclennan, 2009). Definisi lain mengatakan algoritma *Naïve Bayesian Classifier* merupakan pengklasifikasian dengan metode probabilistic dan statistic yang dikemukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya (Kamber, 2009).

Langkah – langkah dari persamaan dari algoritma *Naïve Bayesian Classifier* adalah dapat dilihat pada gambar 2.5 dengan rincian langkah-langkah sebagai berikut (Han dkk, 2012):

1. Jika ada *D* menjadi seperangkat pelatihan *tuples* dan label kelas yang terkait. Masing-masing *tuples* diwakili oleh sebuah n-dimensional atribut *vector, X* = $(x_1, x_2, \ldots, x_n)$, yang menggambarkan n. Pengukuran dilakukan pada *tuples* dari *n* atribut dengan masing-masing nilai $A_1, A_2, \ldots, A_n$.
2. Misalkan terdapat kelas *m*, $C_1, C_2, \ldots, C_m$. Diberikan *tuple X*, *Classifier* akan meramalkan bahwa *X* termasuk dalam kelas yang memiliki probabiliats posterior tertinggi, dikondisikan pada *X*. Artinya, *Naïve Bayesian Classifier* akan memperkirakan bahwa *tuple X* adalah milik kelas *Ci* jika

$$P(C_i|X) > P(C_j|X) \quad \text{untuk} \quad 1 \leq j \leq m, j \neq i$$

Dengan demikian, Dilakukan pemaksimalan *P(Ci|X)* kelas *Ci* yang disebut hipotesis posteriori maksimum.

$$P\ (C|X) = \frac{P(X|C)_{*\,P(C)}}{P(X)} \qquad \text{atau} \quad Posterior = \frac{prior\ x\ likelihood}{evidence} \qquad \ldots(2.1)$$

Dimana :

X: Data dengan class yang belum diketahui.

C: Hipotesis data X merupakan suatu class spesifik.

$P(C|X)$: Probabilitas hipotesis C berdasar kondisi X (posteriori probabilitas).

$P(C)$: Probabilitas hipotesis C (prior probabilitas).

$P(X|C)$: Probabilitas X berdasarkan kondisi pada hipotesis C.

$P(X)$: Probabilitas X.

3. *Data set* dengan banyak atribut akan sangat sulit untuk menghitung $P(X|C_i)$. Untuk mengurangi perhitungan dalam evaluasi $P(X|C_i)$, dibuat asumsi *naïve* terhadap *class-conditional-independence*. Ini menganggap nilai atribut secara kondisional tidak bergantung satu sama lain.

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$
$$= P(x_1|C_i)\ \text{x}\ P(x_2|C_i)\text{x}\ \ldots\ \text{x}\ P(x_n|C_i) \qquad \ldots(2.2)$$

Kita dapat dengan mudah memperkirakan probabilitas $P(x_1|C_i)$, $P(x_2|C_i)$, ... , $P(x_n|C_i)$ dari *training tuple*s. Mengingat bahwa x_k mengacu pada nilai atribut A_k untuk *tuples X*. Untuk tiap atribut, kita melihat apakah atribut itu kategorial atau bernilai kontinu.

a. Jika *Ak* bersifat kategorial, maka $P(x_k|\ C_i)$ adalah jumlah *tuples* kelas C_i dalam *D* yang memiliki nilai *xk* untuk A_k, dibagi dengan $|\ C_i\ ,\ D\ |$, jumlah *tuples* kelas C_i dalam *D*.

b. Jika *Ak* bernilai kontinu, maka biasanya akan diasumsikan memiliki distribusi Gaussian dengan *mean* μ dan standar deviasi σ yang didefinisikan sebagai

$$g(x, \mu, \sigma) \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \ldots(2.3)$$

c. Jika terdapat hasil probabilitas 0 pada suatu kriteria, maka akan digunakan laplace correction dengan persamaan berikut.

$$\rho i = \frac{m_i+1}{n+k} \quad \ldots(2.4)$$

Dimana

$\rho i = probabilitas\ dari\ atribut\ m_i$

$m_i = jumlah\ sample\ dalam\ kelas\ dari\ atribut\ m_i$

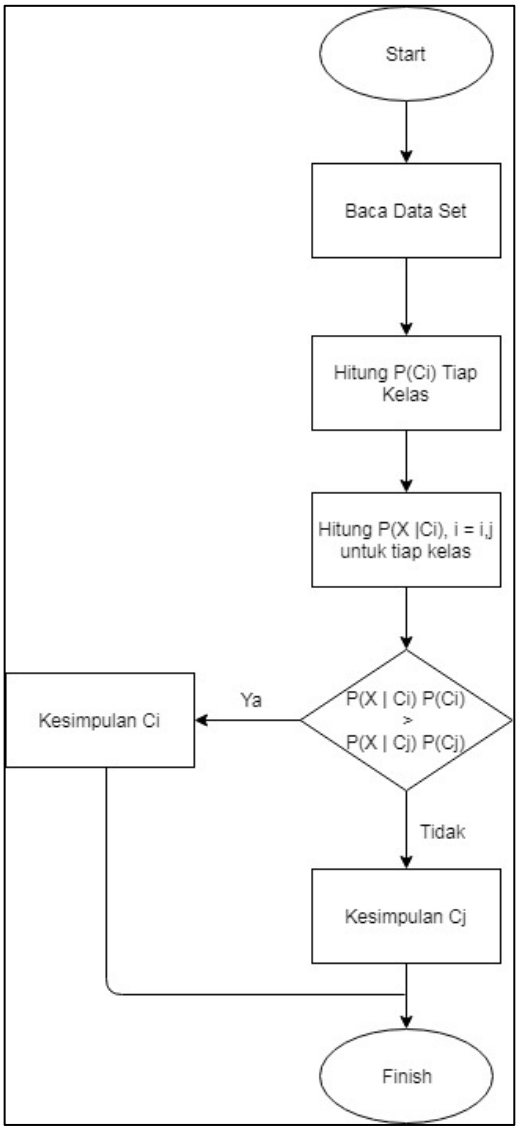
$k = jumlah\ kelas\ dari\ atribut\ m_i$

$n : jumlah\ sample$

4. Untuk memprediksi label kelas *X*, $P(X \mid C_i)\, P(C_i)$ dievaluasi untuk tiap kelas C_i. Naïve Bayesian Classifier memprediksi bahwa kelas *tuples X* adalah kelas C_i jika

$$P(X|C_i)P(C_i) > P\left(X\middle|C_j\right)P(C_j) \quad \text{untuk} \quad 1 \leq j \leq m, j \neq i$$

Dengan kata lain, label kelas yang diprediksi adalah kelas $C_i$dengan $P(X \mid C_i)\, P(C_i)$ telah maksimum.

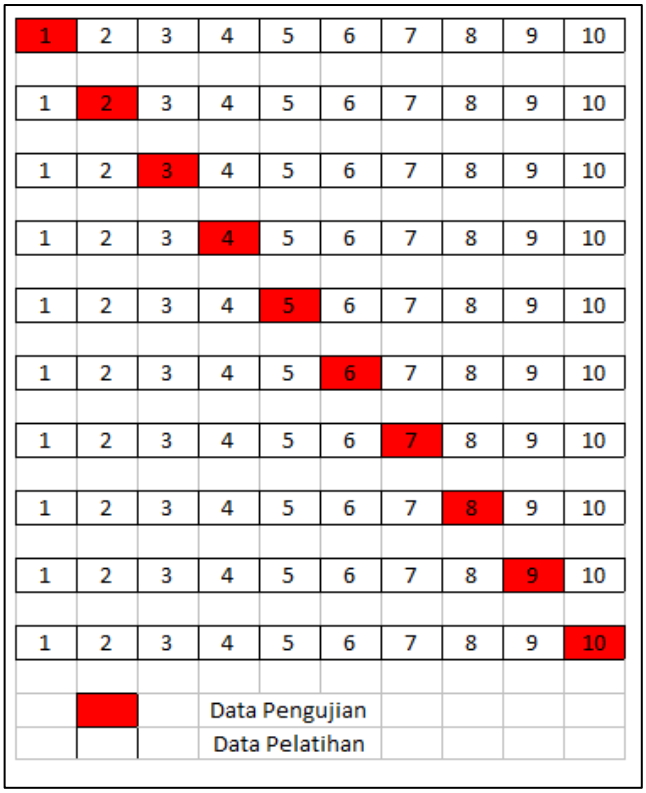


Gambar 2.5 *Flowchart* algortima *Naïve Bayesian Classifier* (Han dkk, 2012)

2.5. 10-Fold-Cross-Validation

Metode *K-Fold Cross* dengan *K* = 10 untuk pembagian *data training* dan *data testing*, dimana data dibagi menjadi 10 *fold* berukuran kira-kira sama, sehingga menghasilkan 10 *subset* data untuk melakukan proses evaluasi kinerja model atau algoritma. Untuk masing-masing dari 10 *subset* tersebut, *cross-validation* akan menggunakan 9 *fold* untuk *data training* dan 1 *fold* untuk *data testing.* 10 *fold-cross validation* direkomendasikan dalam penelitian ini karena teknik cenderung

memberikan estimasi akurasi yang biasnya rendah dibanding dengan *cross validation* lainnya .



Gambar 2.6 Skema 10-*fold-cross-validation* (Wijaya, 2014)

2.6. Confusion Matrix

Confusion Matrix merupakan salah satu metode yang dapast digunakan untuk mengukur kinerja suatu algoritma klasifikasi. Pada dasarnya Confusion Matrix mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusanya (Sokolova, 2009).

Pada pengukuran kinerja menggunakan Confusion Matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi, yaitu *True Postive (*TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).*True Positive* (TP) adalah data positif yang terdeteksi benar. *True Negative* (TN) adalah jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* adalah data

negative yang terdeteksi sebagai data positif. *False Negative* adalah kebalikan dari *True Positive*, sehingga data positif, namun terdeteksi sebagai data negative (Han dkk, 2012).

Tabel 2.1 kebenaran Confusion Matrix (Han dkk, 2012)

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	P
	no	*FP*	*TN*	N
	Total	*P'*	*N'*	P+N

Tabel 2.1 adalah tabel kebenaran dari Confusion Matrix. Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP), dapat diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasi data secara benar.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots(2.4)$$

Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasi secara benar dibagi dengan total data yang diklasifikasi positif.

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad \ldots(2.5)$$

Sementara nilai *recall* menunjukan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem.

$$Recall = \frac{TP}{FN+TP} * 100\% \quad \ldots(2.6)$$