

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Pengenalan Sistem Keputusan**

Konsep sistem keputusan pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System*. Konsep sistem keputusan ditandai dengan sistem interaktif berbasis komputer yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tidak terstruktur (Kadarsah dkk, 2000).

Menurut Turban, dkk (2005) definisi awal sistem keputusan atau dikenal dengan DSS (Decision Support System) menunjukkan sistem keputusan sebagai sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur. Sistem keputusan dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. Sistem keputusan ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma.

Little (1970) dalam buku Turban, dkk (2005) mendefinisikan sistem keputusan sebagai “sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manager mengambil keputusan.” Dia menyatakan bahwa untuk sukses, sistem tersebut haruslah sederhana, cepat, mudah dikontrol, adaptif, lengkap dengan isu-isu penting dan mudah berkomunikasi.

Bonczek, dkk (1980) dalam buku Turban, dkk (2005) mendefinisikan sistem keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi yaitu sistem bahasa, sistem pengetahuan, dan sistem pemrosesan masalah. Komponen-komponen yang diberikan defenisi tersebut sangat penting untuk memahami hubungan antara sistem keputusan dan pengetahuan.

Sistem keputusan secara umum didefenisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semi-terstruktur. Sedangkan secara khususnya, sistem keputusan didefenisikan sebagai sebuah sistem yang mendukung kerja seorang manager dalam memecahkan masalah semi-terstruktur dengan cara memberikan informasi atau usulan menuju pada keputusan tertentu (Julius, 2005).

### **2.1.1 Karakteristik Sistem Keputusan**

1. Sistem Pendukung Keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur dengan menambahkan kebijaksanaan manusia dan informasi komputerisasi.
2. Sistem Pendukung Keputusan, dirancang sedemikian rupa sehingga dapat digunakan/dioperasikan dengan mudah.
3. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari / interogasi informasi.
4. Sistem Pendukung Keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi.

### **2.1.2 Komponen Sistem Keputusan**

Menurut kadarsah dkk, (2000) suatu sistem keputusan memiliki tiga subsistem utama yang menentukan kapabilitas teknis sistem keputusan tersebut, yaitu Subsistem pengelolaan data (*database*), Sub sistem pengelolaan model (*model base*, dan Subsistem pengelolaan dialog (*user interface*).

1. Subsistem pengelolaan data (*database*).

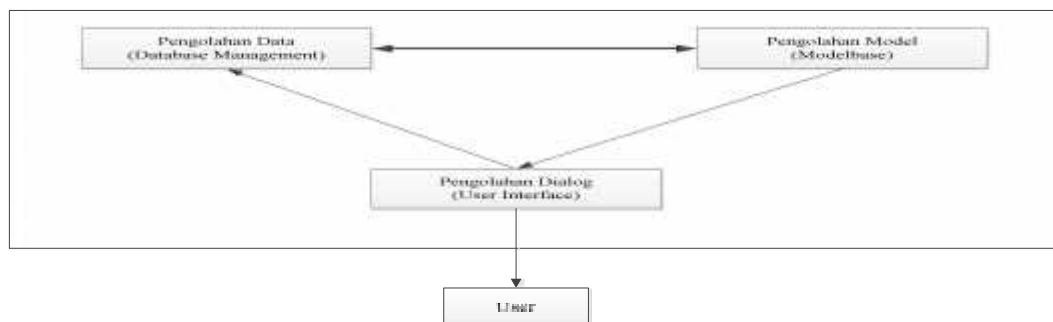
Sub sistem pengelolaan data (*database*) merupakan komponen SPK yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan dan diorganisasikan dalam sebuah basis data yang diorganisasikan oleh suatu sistem yang disebut dengan sistem manajemen basis data (*Database Management System*).

2. Sub sistem pengelolaan model (*model base*)

Keunikan dari SPK adalah kemampuannya dalam mengintegrasikan data dengan model-model keputusan. Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang suatu model adalah bahwa model yang dirancang tidak mampu mencerminkan seluruh variabel alam nyata, sehingga keputusan yang diambil tidak sesuai dengan kebutuhan oleh karena itu, dalam menyimpan berbagai model harus diperhatikan dan harus dijaga fleksibilitasnya. Hal lain yang perlu diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

3. Subsistem pengelolaan dialog (*user interface*)

Keunikan lainnya dari SPK adalah adanya fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif, yang dikenal dengan subsistem dialog. Melalui subsistem dialog, sistem diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat. Hubungan ketiga komponen diatas dapat dilihat pada gambar berikut :



**Gambar 2.1 Komponen Utama Sistem Keputusan (Turban dkk, 2005)**

### 2.1.3 Fase-Fase Sistem Keputusan

Simon (1977) mengatakan bahwa fase-fase sistem keputusan meliputi tiga fase utama : inteligensi, desain, dan kriteria. Ia kemudian menambahkan fase keempat, yakni implementasi. Monitoring dapat dianggap fase kelima. Akan tetapi turban dkk memandang mobitoring monitoring sebagai fase inteligensi yang diterapkan pada fase implementasi. Model simon merupakan karakterisasi yang paling kuat dan lengkap mengenai pengambilan keputusan rasional (Turban dkk, 2005). Berikut penjelasan dari keempat fase simon :

1. Fase Inteligensi

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. Fase Desain

Pada tahap ini dilakukan dengan melakukan perancangan seperti: perancangan fitur, menu aplikasi, perancangan data, perancangan arsitektur, perancangan interface dan perancangan prosedur.

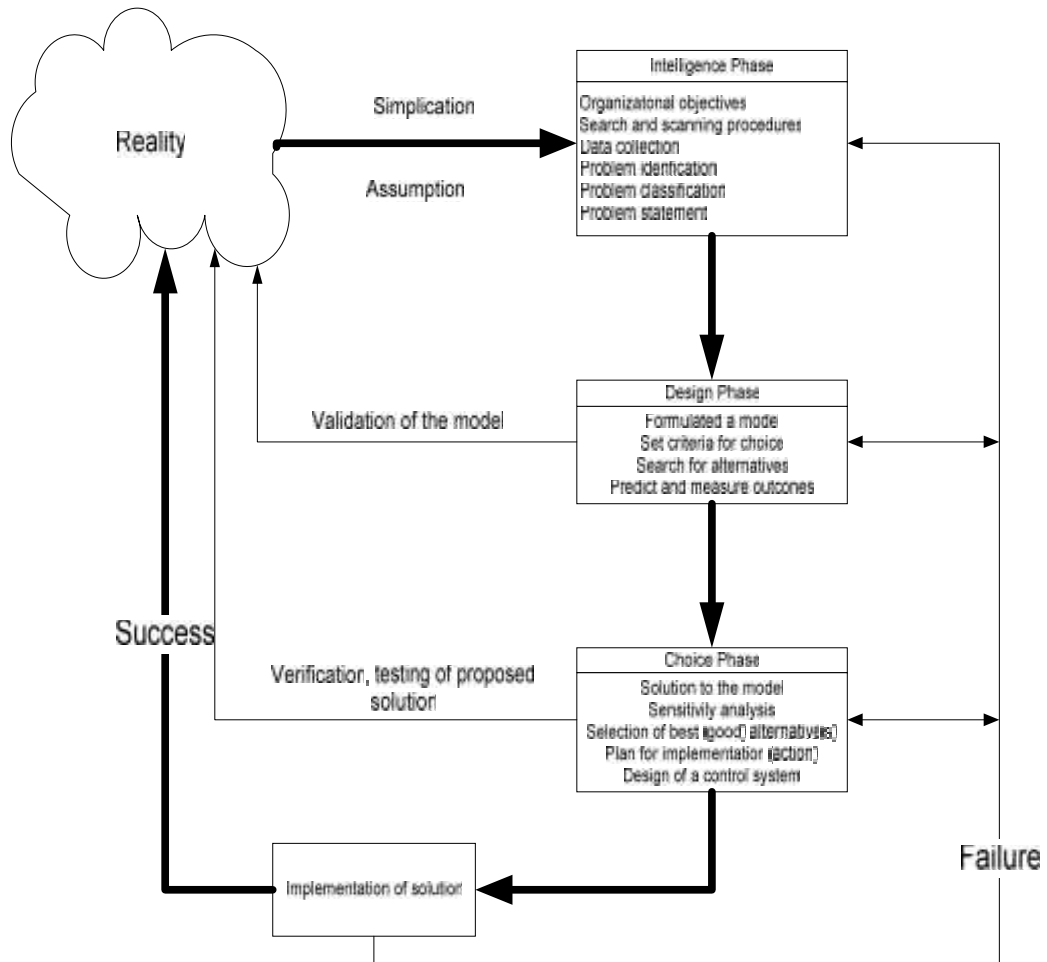
3. Fase Pilihan ( *Choice* )

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

4. Fase Implementasi

Tahapan ini merupakan tahapan optional dalam pengembangan perangkat lunak. Bagian ini terjadi ketika sistem yang di maksud telah selesai dan mengalami perubahan ataupun permintaan

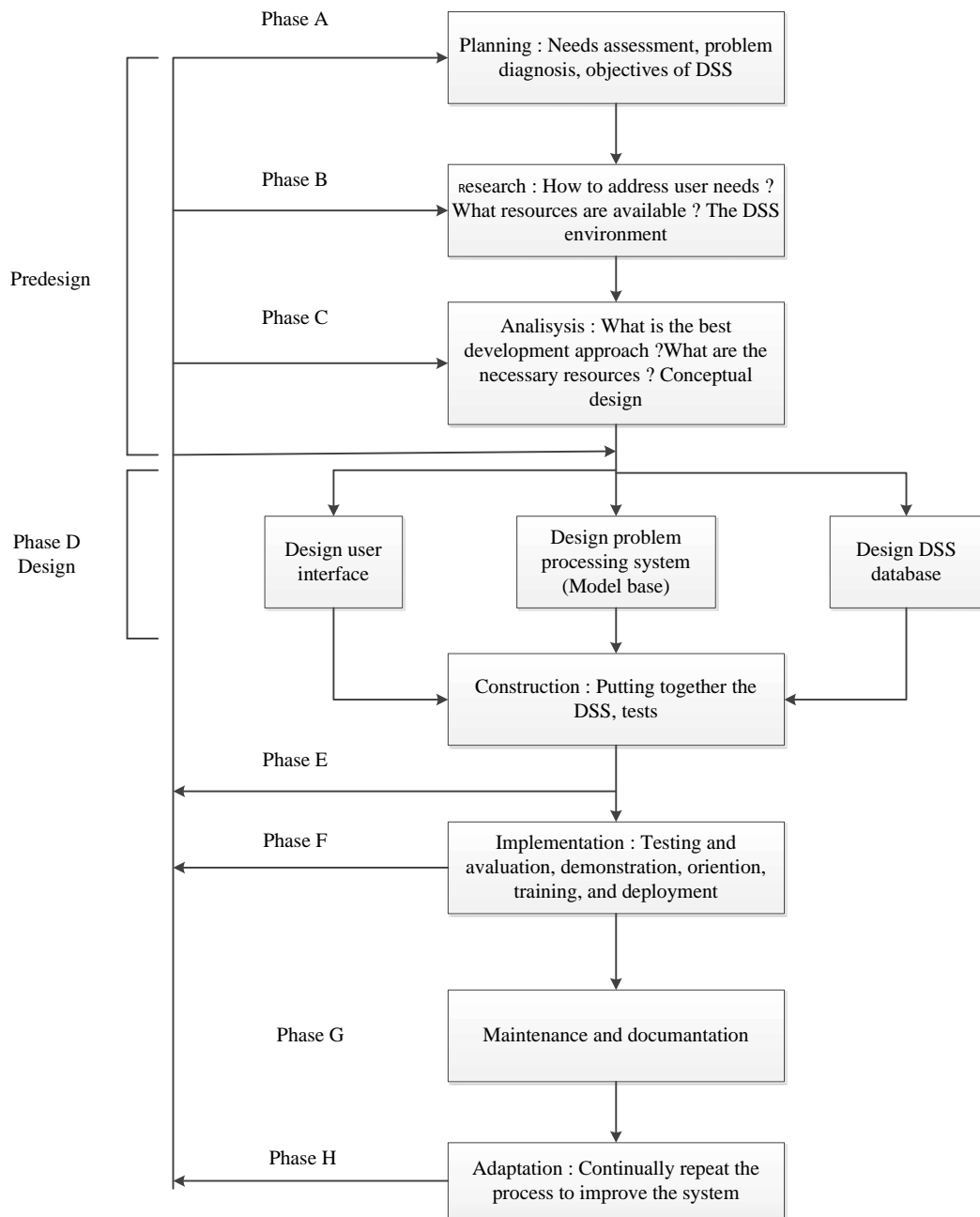
penambahan fitur dikemudian hari. Gambar konseptual mengenai proses pengambilan keputusan ditunjukkan pada gambar :



**Gambar 2.2 Fase-fase sistem keputusan (Turban dkk, 2005)**

#### 2.1.4 Proses Pengembangan Sistem Keputusan

Seperti di dalam pengembangan sistem yang lain, pengembangan DSS juga memiliki langkah-langkah yang sistematis. Biasanya seorang pengembangan sistem DSS dalam menentukan langkah-langkah disesuaikan dengan keperluan, keahlian, waktu, tenaga dan peralatan yang mendukung terhadap keberhasilan pengembangan sistem DSS. Alur pengembangannya dapat di lihat pada gambar :



**Gambar 2.3 Fase-fase pembangunan sistem keputusan (Turban and Aronson, 1998)**

Menurut Keen dan Scott Morton (1978) dan Meador et al (1984) dalam buku (Turban and Aronson, 1998) proses pengembangan DSS melalui beberapa tahap yaitu : *Planning, Research, System Analysis and Conceptual Design, Design, Construction, Implementation, Maintenance and Documentation dan*

*terakhir Adaptation*. Idealnya sebuah sistem mengalir ke bawah, untuk pengembangannya sistem dapat kembali ke fase sebelumnya.

#### **1. Tahap Perencanaan Sistem (Phase A)**

Tahap perencanaan adalah tahap awal pengembangan sistem. Perencanaan sistem adalah merencanakan pengembangan sistem secara umum. Pada tahap ini didefinisikan perkiraan kebutuhan-kebutuhan sumber daya seperti perangkat fisik, manusia, metode (teknik dan operasi), dan anggaran yang sifatnya masih umum (belum detail/rinci).

#### **2. Tahap Penelitian / Riset (Phase B)**

Tahap ini melibatkan identifikasi pendekatan yang relevan untuk mengatasi kebutuhan pengguna dan sumber daya yang tersedia (hardware, software, vendor, sistem, mempelajari atau terkait pengalaman dalam organisasi dan review penelitian yang relevan lainnya). Disini, Lingkungan DSS hubungannya sangat erat.

#### **3. Tahap Analisis Sistem (Phase C)**

Tahap analisis sistem adalah tahap penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbaharui, Artinya analisis sistem ini menganalisa permasalahan terhadap sistem yang akan dibangun, sehingga menemukan solusi dalam melakukan tahap perancangan sebuah sistem.

#### **4. Tahap Desain (Phase D)**

Tahap desain sistem adalah tahap setelah analisis sistem yang menentukan proses dan data yang dilakukan oleh sistem baru. Pada tahap ini sistem dirancang berdasarkan tahap analisis sistem, rancangan sistem terdiri atas tiga bagian yaitu rancangan user interface, rancangan model base dan rancangan database.

#### **5. Tahap Pembangunan/ Perancangan Sistem (Phase E)**

Dss dapat dibangun dalam cara yang berbeda tergantung pada filosofi desain dan perangkat yang digunakan. Pembangunan adalah teknis pelaksanaan desain. Sebagai sistem pembangun ini adalah terus menerus di

uji dan di perbaiki. Di fase ini, dss terhubung, jika diperlukan, dengan CBIS yang tepat dan jaringan.

**6. Tahap Implementasi (Phase F)**

Tahap implementasi atau penerapan adalah tahap di mana desain dan pembangunan atau perancangan sistem dibentuk menjadi suatu kode (program) yang siap untuk dioperasikan.

**7. Tahap pemeliharaan (Phase G)**

Memelihara merupakan tahap yang dilakukan setelah tahap implementasi, yang meliputi pemakaian atau penggunaan, audit sistem, penjagaan, perbaikan, dan peningkatan sistem.

**8. Tahap Adaptasi (Penyesuaian ) Sistem (Phase H)**

Adaptasi membutuhkan daur ulang melalui langkah-langkah sebelumnya secara teratur untuk menanggapi perubahan kebutuhan pengguna.

Proses ini memiliki beberapa variasi (karena ada banyak variasi dari DSS). Contohnya Belardo and Pazer (1995) Mengembangkan kerangka kerja tertentu untuk membangun dss manajemen krisis.

**2.2. *Principial Component Analysis (PCA)***

*Principal Components Analysis (PCA)* ditemukan pada tahun 1901 oleh Karl Pearson. Sebutan lain untuk *Principal Components Analysis (PCA)* adalah Transformasi Karhunen-Loeve (*singular value decomposition*) pada matriks, Transformasi Hotelling atau *Proper Orthogonal Decomposition (POD)* ataupun *Empirical orthogonal Function (EOF)*.

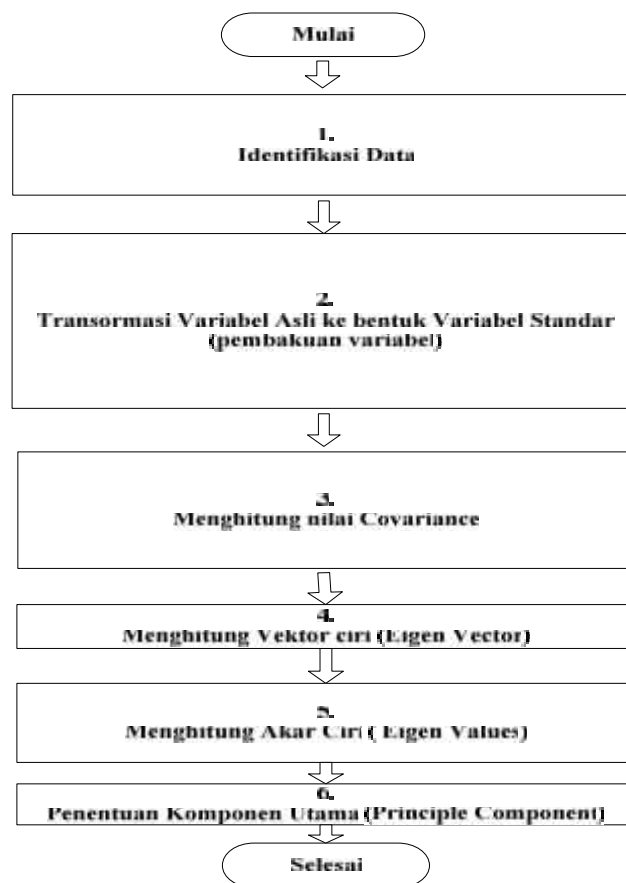
*Principal Components Analysis (PCA)* merupakan suatu teknik yang digunakan untuk mereduksi data multidimensi menjadi data dengan dimensi yang lebih kecil, dengan tetap mempertahankan karakteristik dari data tersebut, untuk menganalisis data dan membuat model peramalan, dalam perhitungannya melibatkan nilai eigen dari matriks kovarians (*singular value decomposition*).



Secara matematis *Principal Components Analysis* (PCA) adalah sebuah transformasi linear orthogonal, yaitu transformasi yang memberikan nilai optimal yang didapat dengan metode kuadrat terkecil. Prosedur PCA pada dasarnya adalah bertujuan untuk menyederhanakan variabel yang diamati dengan menyusutkan (mereduksi) dimensinya. Hal ini dilakukan dengan cara menghilangkan korelasi di antara variabel bebas melalui transformasi variabel bebas asal ke variabel baru yang tidak berkorelasi sama sekali atau yang biasa disebut dengan principal component.

### 2.3.1 Flowchart PCA

Berikut adalah flowchat atau diagram alir dari *Principal Components Analysis* (PCA) (Supranto, 2004) :



**Gambar 2.4 Flowchat PCA**

Dari gambar 2.4 diatas, dapat dilihat bahwa proses perhitungan PCA terdiri dari 6 tahap. Setiap tahap perhitungan akan dijelaskan lebih rinci pada bagian dibawah ini :

1. Identifikasi data

Awalnya adalah identifikasi data yaitu:

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ X_{21} & X_{22} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ X_{n1} & \dots & \dots & X_{np} \end{bmatrix}$$

Identifikasi masing-masing data variabel dapat dihitung nilai rata-ratanya dengan menggunakan persamaan:

$$\bar{X}_i = \frac{\sum_{j=1}^n x_{ij}}{n} \tag{2.1}$$

Dan menghitung nilai simpangan baku (deviasi) dengan menggunakan persamaan:

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{(n-1)}} \tag{2.2}$$

Ket :

$\bar{X}$  = rata-rata

S = simp. Baku

n = jumlah variabel

x<sub>i</sub> = kriteria ke i

2. Transformasi variabel asli ke bentuk variabel standar (pembakuan variabel)

Oleh karena satuan pengukuran dari variabel-variabel berbeda, maka satuan pengukuran itu perlu dibakukan dengan jalan melakukan transformasi variabel asli ke dalam variabel baku, yakni :

$$Z = \begin{bmatrix} Z_{11} & Z_{12} & \dots & Z_{1p} \\ Z_{21} & Z_{22} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ Z_{n1} & \dots & \dots & Z_{np} \end{bmatrix}$$

Caranya yaitu dengan mengurangkan masing-masing data dengan nilai rata-rata dari masing-masing variabel, kemudian dibagi dengan simpangan baku masing-masing variabel. Persamaan yang digunakan adalah :

$$Z_i = \left[ \frac{X_i - \bar{X}}{s_{X_i}} \right] \quad (2.3)$$

Ket :

Z = variabel baku

### 3. Menghitung Nilai *Covariance*

Semua variasi (ragam) yang mungkin diperoleh dari perpasangan vektor kolom dinyatakan sebagai *covariance*. Sebagai contoh, data yang digunakan adalah data dengan variabel 4, maka *covariance* matriks yang diperoleh akan berdimensi 4x4. Nilai *covariance* diperoleh dengan menggunakan persamaan :

$$\left. \begin{aligned} \text{covariance}(X, X) &= \frac{\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})}{n-1} \\ \text{covariance}(X, Y) &= \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n-1} \end{aligned} \right\} \quad (2.4)$$

Nilai *covariance* tersebut merupakan nilai dari matriks korelasi R :

$$R = \begin{bmatrix} r_{Z_1 Z_1} & r_{Z_1 Z_2} & \dots & r_{Z_1 Z_p} \\ r_{Z_2 Z_1} & r_{Z_2 Z_2} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ r_{Z_{ni}} & \dots & \dots & r_{Z_{np}} \end{bmatrix}$$

Ket :

R = nilai *covariance*

Setelah matriks R diperoleh, kemudian dilakukan penggandaan matriks R

### 4. Menghitung Nilai Vektor Ciri (*Eigen Vektor*)

Eigen Vektor dicari dengan melakukan iterasi dari matriks korelasi, kemudian dinormalkan agar berlaku  $a'_1 \cdot a_1 = 1$  dengan menggunakan persamaan :

$$a_{ij} = \frac{a_i}{\sqrt{(a_1)^2 + \dots + (a_n)^2}} \quad (2.5)$$

Ket :

a = *Eigen Vektor*

Iterasi dihentikan jika hasil iterasi yang terakhir berdekatan dengan hasil iterasi sebelumnya. Terlebih dahulu harus ditentukan vektor awal ( $\alpha'_0$ ) dengan mempertimbangan struktur matriks R, yaitu bernilai positif atau negatif. Kemudian dilakukan perkalian vektor  $\alpha'_1$  dengan matriks  $R^2$  ( $\alpha'_1 \cdot R^2$ ). Dari hasil yang  $\alpha'_1 \cdot R^2$ , dipilih nilai (elemen) terbesar yang akan dijadikan sebagai pembagi untuk masing-masing elemen  $\alpha'_1 \cdot R^2$ . Pada perhitungan PCA ini, selisih yang digunakan adalah 0.05. Didalam buku statistik dijelaskan bahwa 0.05 adalah kemungkinan error. Jadi untuk tingkat kepercayaannya bisa dihitung dengan persamaan  $(1 - \alpha) \cdot 100\%$  dimana  $\alpha = 0.05$  berarti  $(1 - 0.05) \cdot 100\% = 95\%$ .

#### 5. Menghitung Nilai Akar Ciri (*Eigen Value*)

Jika matriks yang dicari *eigen value*nya berdimensi  $p \times p$ , maka akan ada  $p$  *eigen value* (akar ciri) yang diperoleh dari hasil *eigen vector* (vector ciri) yang telah diperoleh sebelumnya dengan memenuhi persamaan linier :

$$\alpha'_i (1 - \lambda_i) + \alpha'_{ni} + r_{12} = 0 \quad (2.6)$$

Ket :

$$\lambda_i = \text{Eigen Value}$$

#### 6. Penentuan Komponen Utama (*Principal Component*)

Persamaan untuk komponent utama pertama:

$$Y_1 = \alpha'_{11}Z_1 + \alpha'_{21}Z_2 + \dots + \alpha'_{p1}Z_p \quad (2.7)$$

Dari nilai *eigen value*, maka persamaan :

$$Y_1 = \frac{\lambda_i}{\sigma^2} \quad (2.8)$$

Dimana  $Y = \text{Principal Component}$ ,  $I = 1, 2, \dots, n$  dan  $p = \text{jumlah variable}$

Jika nilai PC1 memiliki persentase nilai yang besar ( $\geq 75\%$ ), berarti telah mampu menerangkan keragaman total secara baik.

## **2.3 *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)***

### **2.3.1 Sejarah TOPSIS**

*Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)* adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yonn dan Hwang pada tahun 1981. Ide dasar dari metode ini adalah bahwa alternatif yang dipilih memiliki jarak terdekat dengan solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Dengan melakukan perbandingan pada keduanya, urutan pilihan dapat ditentukan. Pilihan akan diurutkan berdasarkan nilai sehingga alternatif yang memiliki jarak terpendek dengan solusi ideal positif adalah alternatif yang terbaik. Dengan kata lain, alternatif yang memiliki nilai yang lebih besar itulah yang lebih baik untuk dipilih. TOPSIS banyak digunakan dengan alasan:

- konsepnya sederhana dan mudah dipahami
- komputasinya efisien
- memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana

### **2.3.2 Flowchart TOPSIS**

Berikut adalah flowchat atau diagram alir dari TOPSIS :



**Gambar 2.5 Flowchat TOPSIS**

### 2.3.3 Deskripsi Teknik Pemodelan

TOPSIS menggunakan prinsip bahwa alternatif yang terpilih tidak hanya mempunyai jarak terpendek dari solusi ideal positif, namun juga memiliki jarak terpanjang dari solusi ideal negatif. Konsep ini banyak digunakan untuk menyelesaikan masalah keputusan secara praktis. Konsepnya sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan kedalam bentuk matematis yang sederhana (kusumadewi dkk. , 2006). Konsep fundamental dari metode ini adalah penentuan jarak Euclide terpendek dari solusi ideal positif dan jarak. Berikut penjelasan dari gambar 2.5 :

Langkah-langkah:

- 1) TOPSIS membutuhkan rating kinerja setiap alternatif  $A_i$  pada setiap kriteria  $C_j$  yang ternormalisasi, yaitu :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Ket =

$r_{ij}$  = rating kinerja

$x_{ij}$  = kriteria ke i pada alternatif ke j

2) Membuat matriks keputusan yang ternormalisasi terbobot :

$$y_{ij} = w_i r_{ij}$$

Ket =

$y$  = rating terbobot yang ternormalisasi

$w$  = bobot

3) Solusi ideal positif  $A^+$  dan solusi ideal negatif  $A^-$  dapat ditentukan berdasarkan rating bobot ternormalisasi ( $y^{ij}$ ) sebagai :

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+)$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-)$$

Ket =

$A^+$  = Solusi ideal positif

$A^-$  = Solusi ideal negatif

Dengan :

$$y_j^+ = \begin{cases} \max_i y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \min_i y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

$$y_j^- = \begin{cases} \min_i y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \max_i y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

4) a. Jarak antara alternatif  $A_i$  dengan *solusi ideal positif* dirumuskan sebagai :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}$$

b. Jarak antara alternatif  $A_i$  dengan *solusi ideal negatif* dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2}$$

5) Nilai *preferensi* untuk setiap alternatif ( $V_i$ ) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

Ket =

$V_i$  = Nilai *preferensi*

6) Nilai  $V_i$  yang lebih besar menunjukkan bahwa alternatif lebih dipilih