

BAB 2

LANDASAN TEORI

2.1. Jagung

Menurut Wikipedia Indonesia Jagung adalah salah satu bahan makanan pokok yang kaya akan karbohidrat seperti gandum dan padi. Selain sebagai karbohidrat jagung dapat dijadikan makan ternak, diambil minyaknya, dibuat tepung, dan bahan baku industri.

Karena jagung sebagai bahan makanan pokok dan banyak fungsi lainnya maka kebutuhan akan jagung terus meningkat seiring dengan laju pertumbuhan penduduk. Jadi dibutuhkan suatu gerakan intensifikasi pertanian untuk meningkatkan produksi jagung yaitu panca usaha tani. Panca usaha tani terdiri dari pengolahan tanah yang baik, pemupukan yang tepat, pengendalian hama/penyakit, pengairan/irigasi dan penggunaan varietas unggul (Dhiya, 2008).

Penggunaan varietas unggul dalam jagung adalah jagung varietas hibrida dihasilkan dari penyerbukan bunga menyilang dikarenakan tanaman jagung mempunyai komposisi genetik yang sangat dinamis (Andi et al, 2007, p74). Hasil persilangan gen yang homozigot justru akan menghasilkan tanaman kerdil dan daya hasilnya rendah. Tanaman yang subur, tumbuh cepat, dan hasilnya tinggi dihasilkan dari komposisi gen yang heterozigot.

Jagung varietas hibrida merupakan generasi pertama hasil persilangan antara tetua jagung varietas inbrida (Andi et al, 2007, p74). Jagung hibrida merupakan tanaman yang pada awalnya dikembangkan di Amerika secara komersial sejak tahun 1930.

Jagung inbrida sebagai tetua hibrida yang memiliki tingkat homozigositas yang tinggi. Oleh karena itu pengelompokan jagung varietas inbrida sangat penting untuk menghasilkan jagung hibrida unggulan.

2.2. Rancangan Acak Kelompok

Rancangan acak kelompok (RAK) atau dalam bahasa Inggris adalah *Randomized Complete Block Design (RCBD)* mempunyai tujuan melakukan pengujian hipotesis apakah terdapat perbedaan rata-rata antara kelompok dan perlakuan yang diuji dalam percobaan. .

RAK memiliki model matematis sebagai berikut (Quinn dan Keough, 2002, p269) :

$$y_{ij} = \mu + \alpha_i + \tau_j + \varepsilon_{ij}; i = 1, 2, 3, \dots, k; j = 1, 2, 3, \dots, p \quad (2.1)$$

Keterangan :

y_{ij} : Pengamatan kelompok ke-i dan Perlakuan ke-j

α_i : Pengaruh kelompok ke-i

τ_j : Pengaruh perlakuan ke-j

μ : rata-rata keseluruhan

ε_{ij} : Galat dari kelompok ke-i dan perlakuan ke-j

Data hasil percobaan dapat disusun dalam tabel pengamatan seperti pada Tabel 2.1.

Tabel 2.1 Tabel Pengamatan

Perlakuan (j)	Kelompok (i)					Total ($y_{.j}$)	Rataan ($\bar{y}_{.j}$)
	1	2	3	...	k		
1	y_{11}	y_{21}	y_{31}	...	y_{k1}	$y_{.1}$	$\bar{y}_{.1}$
2	y_{12}	y_{22}	y_{32}	...	y_{k2}	$y_{.2}$	$\bar{y}_{.2}$
3	y_{13}	y_{23}	y_{33}	...	y_{k3}	$y_{.3}$	$\bar{y}_{.3}$
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
p	y_{1p}	y_{2p}	y_{3p}	...	y_{kp}	$y_{.p}$	$\bar{y}_{.p}$
Total ($y_{i.}$)	$y_{.1}$	$y_{.2}$	$y_{.3}$...	$y_{.k}$	$y_{..}$	$\bar{y}_{..}$
Rataan ($\bar{y}_{i.}$)	$\bar{y}_{.1}$	$\bar{y}_{.2}$	$\bar{y}_{.3}$...	$\bar{y}_{.k}$	$\bar{y}_{..}$	

Pada Persamaan 2.1 dapat diterapkan dalam Tabel 2.1 nilai pengamatan dapat dijabarkan dalam komponen sebagai berikut :

$$y_{ij} = \bar{y}_{..} + (\bar{y}_{i.} - \bar{y}_{..}) + (\bar{y}_{.j} - \bar{y}_{..}) + (y_{ij} - \bar{y}_{i.} - \bar{y}_{.j} + \bar{y}_{..})$$

$$(y_{ij} - \bar{y}_{..}) = (\bar{y}_{i.} - \bar{y}_{..}) + (\bar{y}_{.j} - \bar{y}_{..}) + (y_{ij} - \bar{y}_{i.} - \bar{y}_{.j} + \bar{y}_{..}) \quad (2.2)$$

Kalau dijumlahkan dan dikuadratkan maka menjadi :

$$\sum_{i=1}^k \sum_{j=1}^p (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^k \sum_{j=1}^p (\bar{y}_{i.} - \bar{y}_{..})^2 + \sum_{i=1}^k \sum_{j=1}^p (\bar{y}_{.j} - \bar{y}_{..})^2 + \sum_{i=1}^k \sum_{j=1}^p (y_{ij} - \bar{y}_{i.} - \bar{y}_{.j} + \bar{y}_{..})^2$$

Dimana :

Jumlah Kuadrat Total (JKT) dapat dirumuskan (Dean dan Voss, 2000, p302) :

$$JKT = \sum_{i=1}^k \sum_{j=1}^p (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^k \sum_{j=1}^p y_{ij}^2 - \frac{(y_{..})^2}{kp} \quad (2.3)$$

Jumlah Kuadrat Kelompok (JKK) dapat dirumuskan (Dean dan Voss, 2000, p302) :

$$JKK = \sum_{i=1}^k \sum_{j=1}^p (\bar{y}_{.i} - \bar{y}_{..})^2 = \frac{1}{p} \sum_{i=1}^k y_{.i}^2 - \frac{(y_{..})^2}{kp} \quad (2.4)$$

Jumlah Kuadrat Perlakuan (JKP) dapat dirumuskan (Dean dan Voss, 2000, p302) :

$$JKP = \sum_{i=1}^k \sum_{j=1}^p (\bar{y}_{.j} - \bar{y}_{..})^2 = \frac{1}{k} \sum_{j=1}^p y_{.j}^2 - \frac{(y_{..})^2}{kp} \quad (2.5)$$

Jumlah Kuadrat Galat (JKG) dapat dirumuskan (Dean dan Voss, 2000, p302) :

$$JK \text{ Galat} = JK \text{ Total} - JK \text{ Kelompok} - JK \text{ Perlakuan} \quad (2.6)$$

Sedangkan untuk derajat bebas (db) yang dihasilkan dari Persamaan 2.2 adalah :

$$\begin{aligned} (kp - 1) &= (k - 1) + (p - 1) + (kp - k - p + 1) \\ (kp - 1) &= (k - 1) + (p - 1) + (k - 1)(p - 1) \end{aligned}$$

Jadi derajat bebas (db) dapat ditulis menjadi :

$$DB \text{ Total} = DB \text{ Kelompok} + DB \text{ Perlakuan} + DB \text{ Galat}$$

Dari persamaan diatas dapat menghasilkan tabel *analysis of variance* (ANOVA) (Dean dan Voss, 2000, p302) :

Tabel 2.2 ANOVA

Sumber Keragaman (SK)	Derajat Bebas (DB)	Jumlah Kuadrat (JK)	Keragaman Total (KT)	F-Hitung	F-Tabel α
Kelompok	(k-1)	JKK	$\frac{JKK}{(k-1)} = KTK$	$\frac{KTK}{KTG}$	$F_{\alpha, db \text{ kelompok}, db \text{ galat}}$
Perlakuan	(p-1)	JKP	$\frac{JKP}{(p-1)} = KTP$	$\frac{KTP}{KTG}$	$F_{\alpha, db \text{ perlakuan}, db \text{ galat}}$
Galat	(k-1)(p-1)	JKG	$\frac{JKG}{(k-1)(p-1)} = KTG$		
Total	(kp-1)	JKT			

Pengujian hipotesis dalam analisis rancangan acak kelompok (RAK) untuk kelompok dapat ditulis sebagai berikut :

$$H_0 : \alpha_1 = \alpha_2 = \dots = \alpha_k$$

H_1 : Terdapat perbedaan rata-rata minimal ada sepasang kelompok.

Statistika pengujian dapat diambil dari Tabel 2.2 yaitu F-Hitung dari kelompok (KTK/KTG). F-Tabel adalah $F_{\alpha, db \text{ kelompok}, db \text{ galat}}$.

Dengan daerah penolakan H_0 kelompok adalah F-Hitung > F-Tabel (Dean dan Voss, 2000, p302) yang berarti kelompok berbeda nyata.

Sedangkan untuk pengujian hipotesis dalam analisis rancangan acak kelompok (RAK) untuk perlakuan dapat ditulis sebagai berikut :

$$H_0 : \tau_1 = \tau_2 = \dots = \tau_p$$

H_1 : Terdapat perbedaan rata-rata minimal ada sepasang perlakuan.

Statistika pengujian yang diambil dari Tabel 2.2 yaitu F-hitung dari perlakuan (KTP/KTG). F-Tabel adalah $F_{\alpha, db \text{ perlakuan}, db \text{ galat}}$.

Dengan daerah penolakan H_0 perlakuan adalah F-Hitung > F-Tabel (Dean dan Voss, 2000, p302) yang berarti perlakuan berbeda nyata.

2.3. Analisis Komponen Utama

Analisis komponen utama merupakan suatu teknik analisis statistika untuk mentransformasikan variabel-variabel asli yang masih berkorelasi satu dengan yang lain menjadi satu variabel yang tidak berkorelasi. Variabel tersebut disebut sebagai komponen utama (Johnson dan Wichern, 2002, p427).

Analisis komponen utama terkait dengan struktur varian-kovarian dan korelasi dari variabel-variabel melalui sedikit kombinasi linear variabel-variabel ini untuk mengurangi data asli sehingga menjadi sebagian set kombinasi linear yang lebih sedikit akan tetapi menyerap sebagian besar jumlah varian dari data asli.

Oleh karena, analisis komponen utama mentransformasikan variabel yang berkorelasi menjadi variabel yang tidak berkorelasi maka hal yang pertama harus dilakukan adalah pengujian hipotesis yaitu apakah antara variabel saling berkorelasi ? Bila tidak saling berkorelasi maka tidak dapat dianalisis dengan analisis komponen utama. Sedangkan bila saling berkorelasi maka variabel dapat dianalisis dengan analisis komponen utama untuk menghasilkan variabel baru yang tidak saling berkorelasi dan dapat menghasilkan variabel yang lebih sedikit akan tetapi menyerap sebagian besar jumlah varian dari data asli.

Untuk menguji asumsi apakah antara variabel-variabel saling berkorelasi dibutuhkan beberapa tahapan-tahapan yaitu :

1. Membuat hipotesis untuk uji korelasi.

$$H_0 : r_{xy} = 0$$

$$H_1 : r_{xy} \neq 0$$

Keterangan :

r_{xy} adalah koefisien korelasi antara variabel x dan variabel y.

$$r_{xy} = \frac{\sum XY - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{n}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{n}\right)}} \quad (\text{Zar, 1999, p378})$$

2. Menghitung statistika pengujian yaitu :

$$F_{hitung} = \frac{1 + |r_{xy}|}{1 - |r_{xy}|} \quad (\text{Zar, 1999, p381}) \quad (2.7)$$

3. Melihat tabel F yaitu $F_{\alpha/2, n-2, n-2}$ dimana α adalah tingkat kepercayaan dan n adalah jumlah banyaknya data.
4. Mengambil keputusan tolak atau terima H_0 berdasarkan :

Jika $F_{hitung} \geq F_{\alpha/2, n-2, n-2}$ maka tolak H_0 (Zar, 1999, p381) selain dari itu maka terima H_0

Dimana tolak H_0 mempunyai arti variabel x dan y saling berkorelasi dengan tingkat kepercayaan α sedangkan terima H_0 mempunyai arti variabel x dan y tidak saling berkorelasi dengan tingkat kepercayaan α .

Setelah melakukan pengujian hipotesis korelasi antara variabel dalam analisis komponen utama adalah melakukan standarisasi data. Tujuan menstandarisasi data adalah agar memiliki unit/satuan yang sama dan data menyebar normal. Menstandarisasi data dapat dilakukan dengan rumus sebagai berikut :

$$z = \frac{X_i - \mu}{\sigma} \quad (\text{Zar, 1999, p72}) \quad (2.8)$$

Keterangan :

z adalah hasil data yang telah distandarisasi.

X_i adalah data yang ingin distandarisasi.

μ adalah rata-rata dari data. $\mu = \frac{\sum_{i=1}^n X_i}{N}$ (Zar, 1999, p21) ; n adalah banyaknya data.

σ adalah standart deviasi. $\sigma = \sqrt{\frac{\sum_{i=1}^n \left(X_i - \frac{\left(\sum_{i=1}^n X_i \right)}{n} \right)^2}{n-1}}$ (Zar, 1999, p39)

Tahapan ketiga dalam analisis komponen utama adalah menghitung kovarian antara variabel data yang telah distandarisasi. Kovarian antara variabel data mempunyai rumus sebagai berikut :

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (\text{Smith, 2002, p5}) \quad (2.9)$$

Keterangan :

X_i adalah variabel data x.

Y_i adalah variabel data y.

\bar{X} adalah rata-rata dari variabel x.

\bar{Y} adalah rata-rata dari variabel y.

n adalah banyak data dari satu variabel.

Tahapan keempat adalah menghitung nilai eigen dan vektor eigen dari kovarian antara variabel yang didapatkan dari tahapan ketiga. Nilai eigen bisa didapatkan dari rumus :

$$|\lambda I - A| = 0 \text{ (Kwank dan Hong, 1997, p210)} \quad (2.10)$$

Keterangan :

I adalah matrik identitas.

A adalah matrik kovarian.

λ adalah nilai eigen.

Setelah mendapatkan nilai eigen baru bisa mendapatkan vektor eigen dengan rumus :

$$(\lambda I - A)X = 0 \text{ (Kwank dan Hong, 1997, p210)} \quad (2.11)$$

Keterangan :

I adalah matrik identitas.

A adalah matrik kovarian.

λ adalah nilai eigen.

X adalah variabel dari data.

Setelah dimasukkan setiap nilai eigen maka akan didapat nilai vektor eigen setiap variabel data.

Jadi secara umum pembentukan komponen utama dapat disusun sebagai berikut :

$$\begin{aligned}
y_1 &= a_1'X = a_{11}x_1 + a_{21}x_2 + \dots + a_{p1}x_p \\
y_2 &= a_2'X = a_{12}x_1 + a_{22}x_2 + \dots + a_{p2}x_p \\
&\vdots \\
y_p &= a_p'X = a_{1p}x_1 + a_{2p}x_2 + \dots + a_{pp}x_p
\end{aligned}
\quad (\text{Johnson dan Wichern, 2002, p427}) \quad (2.12)$$

Keterangan :

y_p adalah data baru ke p.

a_p' adalah vektor eigen dari kovarian yang *ditranspose*.

x adalah data awal.

Dengan keragaman masing-masing adalah $\text{var}(y_i) = \lambda_i$ (Johnson dan Wichern, 2002, p428) dimana $i = 1, 2, \dots, p$ dan λ_i adalah akar cirri dari komponen utama ke-i.

Keragaman total yang didapat adalah $\text{Var}(Y) = \lambda_1 + \lambda_2 + \dots + \lambda_p$ (Johnson dan Wichern, 2002, p429).

Tahapan kelima dalam analisis komponen utama adalah menghitung besarnya proporsi dari keragaman total populasi yang dapat diterangkan oleh komponen utama ke-k adalah (Adiningsih et al, 2004, p218)

$$\text{Proporsi ke-k} = \frac{\lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_p} \times 100\%; k = 1, 2, \dots, p; \quad (2.13)$$

Keterangan :

λ_i adalah nilai vektor ke-i.

Dari Persamaan 2.13 didapatkan nilai proporsi dari keragaman total yang dapat diterangkan oleh komponen utama, kedua atau sampai jumlah komponen utama secara bersama-sama dengan semaksimal mungkin dan meminimalisasi informasi yang hilang.

Proporsi keragaman yang dianggap cukup mewakili total keragaman data yang paling umum adalah 80 %.

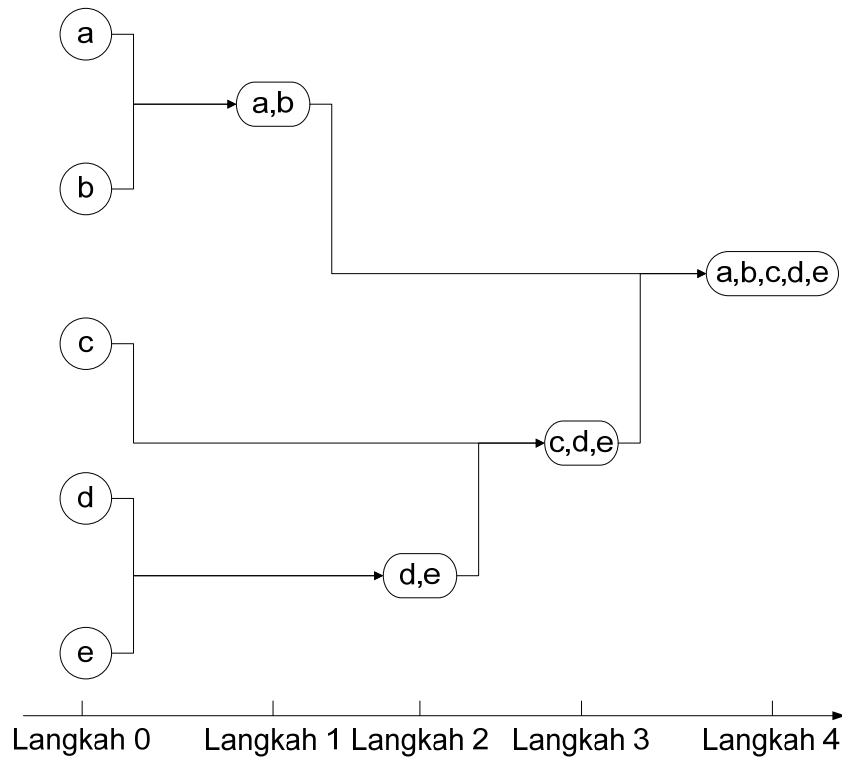
Tahapan keenam adalah tahapan terakhir yaitu membentuk data baru berdasarkan jumlah keragaman total yang diinginkan. Pembentukan data baru menggunakan model 2.12 yaitu $y_i = a_i'X$.

Demikian terdapat enam tahapan dalam melakukan analisis komponen utama sehingga dapat menghasilkan data baru yang dapat menerangkan keragaman total semaksimal mungkin dan meminimalisasi informasi yang hilang.

2.4. Analisis Cluster

Analisis cluster adalah upaya menemukan sekelompok objek yang mewakili suatu karakter yang sama atau hampir sama antara satu objek dengan objek lainnya pada suatu kelompok dan memiliki perbedaan dengan objek-objek pada kelompok lain (Budhi et al, 2008, pp25). Menurut Kaufman dan Rousseeuw (2005, p38) terdapat dua metode algoritma yaitu metode *partitioning* dan metode *hierarchical*. Metode *partitioning* adalah metode membangun k kluster yang mana data diklasifikasikan dalam k group (Kaufman dan Rousseeuw, 2005, p38). Metode *hierarchical* adalah metode yang tidak dibangun dengan partisi dengan k group tetapi dibangun dengan semua nilai dari k dalam waktu bersamaan (Kaufman dan Rousseeuw, 2005, p44). Di dalam metode *hierarchical* terdapat metode *agglomerative*.

Metode *agglomerative* adalah metode yang mulai dengan semua bagian objek untuk melihat lebih jelasnya dapat dilihat pada Gambar 2.1 (dimulai dari langkah 0 dengan n objek) (Kaufman dan Rousseeuw, 2005, p44).



Gambar 2.1 Metode Agglomerative

Jarak yang akan digunakan dalam analisis metode agglomerative adalah jarak Euclidean. Jarak Euclidean mempunyai rumus sebagai berikut (Kaufman dan Rousseeuw, 2005, p11) :

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + \dots + (x_{ip} - x_{jp})^2} \quad (2.14)$$

Setelah mendapatkan jarak Euclidean antara objek maka lakukan tahapan dalam melakukan analisis dengan menggunakan metode agglomerative.

Tahapan-tahapan dalam melakukan metode agglomerative untuk mengelompokan N objek (Johnson dan Wichern, 2002, p681):

1. Mulai dengan N kelompok, setiap kelompok mengandung satu kesatuan dan

$N \times N$ simetrik matrik dari jarak $D = \{d_{ik}\}$.

2. Mencari jarak matrik untuk terdekat pasangan dari kelompok dengan mendapatkan paling jarak minimum antara kelompok.
3. Gabungkan kelompok U dan V. Namakan bentuk kelompok baru (UV). Perbaruhi semua masukan dalam jarak matrik dengan (a) menghapus baris dan kolom sesuai pada kelompok U dan V dan (b) menambahkan sebuah baris dan kolom yang diberikan jarak antara kelompok (UV) dan sisa kelompok.
4. Ulangi cara kedua dan 3 sebanyak N-1 kali.

Analisis cluster dalam metode agglomerative menurut Johnson dan Wichern (2002, p680) terdapat tiga metode yaitu yaitu *single linkage*, *complete linkage*, dan *average linkage*.

Single linkage adalah metode agglomerative yang mencari jarak minimum atau terdekat dari tetangga. Penggunaan *single linkage* pada tahap 3 dalam tahapan melakukan analisis metode agglomerative, jarak antara (UV) dan kelompok W dapat dihitung dengan (Johnson dan Wichern, 2002, p681) :

$$d_{(UV)W} = \min \{d_{UW}, d_{VW}\} \quad (2.15)$$

Keterangan :

d_{UW} adalah jarak antara U dan W.

d_{VW} adalah jarak antara V dan W.

Complete linkage adalah metode agglomerative yang mencari jarak maksimum atau terjauh dari tetangga. Penggunaan *complete linkage* pada tahap 3 dalam tahapan melakukan analisis metode agglomerative, jarak antara (UV) dan kelompok W dapat dihitung dengan (Johnson dan Wichern, 2002, p685) :

$$d_{(UV)W} = \max \{d_{UW}, d_{VW}\} \quad (2.16)$$

Keterangan :

d_{UW} adalah jarak antara U dan W.

d_{VW} adalah jarak antara V dan W.

Average linkage adalah metode agglomerative yang memperlakukan dua kelompok sebagai jarak rata-rata antara semua pasangan dimana satu anggota dari pasangan dipunyai setiap kelompok (Johnson dan Wichern, 2002, p689). Penggunaan *complete linkage* pada tahap 3 dalam tahapan melakukan analisis metode agglomerative, jarak antara (UV) dan kelompok W dapat dihitung dengan (Johnson dan Wichern, 2002, p689) :

$$d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)}N_W} \quad (2.17)$$

Keterangan :

d_{ik} adalah jarak antara objek i dalam kelompok (UV) dan objek k dalam kelompok W.

$N_{(UV)}$ adalah banyaknya isi dari kelompok (UV).

N_w adalah banyaknya isi dari kelompok W.

Setelah mendapatkan masing-masing jarak dari *linkage*, untuk mengetahui persamaan antara objek U dan V dapat digunakan persamaan sebagai berikut :

$$s_{(UV)} = 100\% \times \left(1 - \frac{d_{UV}}{d_{\max}} \right) \quad (2.18)$$

Keterangan:

$s_{(UV)}$ adalah persamaan antara objek U dan V.

d_{UV} adalah jarak minimum antara objek U dan V.

d_{MAX} adalah jarak maksimum pada jarak Euclidean yang dihasilkan.

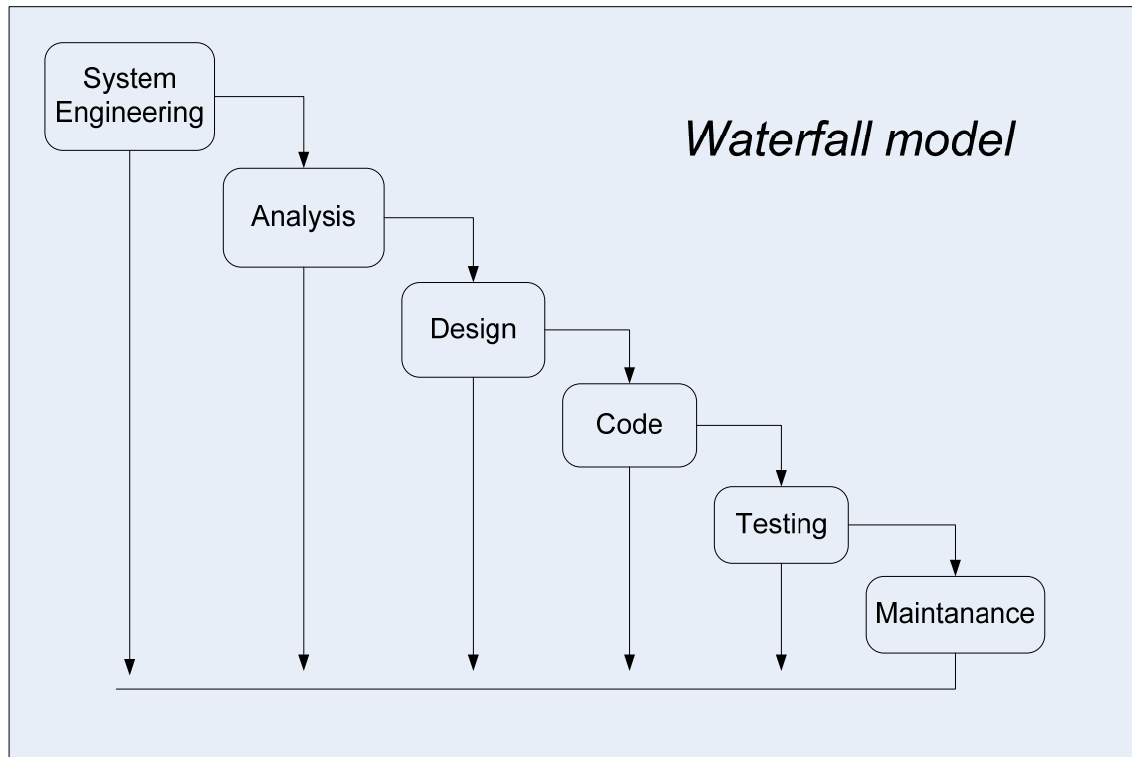
2.5. Rekayasa Perangkat Lunak

Menurut Pressman (1992, p10) perangkat lunak adalah (1) instruksi (komputer program) yang ketika dieksekusi menyediakan fungsi dan kemampuan yang diinginkan, (2) data struktur yang dapat digunakan oleh program untuk memanipulasi informasi, (3) dokument-dokument yang menggambarkan operasi-operasi dan penggunaan program.

Menurut Abran et al (2004, p1) rekayasa perangkat lunak adalah aplikasi dari sistematis, disiplin, pendekatan yang dapat dihitung dalam pengembangan, pengoperasian, dan pemeliharaan dari perangkat lunak.

2.6. Daur Hidup Perangkat Lunak

Daur hidup perangkat lunak (*software life cycle*) digunakan untuk mengidentifikasi proses-proses yang terjadi pada saat pengembangan perangkat lunak. Menurut Pressman (1992,p24) salah satu model daur hidup perangkat lunak adalah model air terjun (*waterfall model*). Seperti terlihat pada Gambar 2.2.



Gambar 2.2 Model Air Terjun

- a) *System engineering and analysis*. Perangkat lunak merupakan bagian dari suatu sistem yang lebih besar sehingga langkah pertama yang harus dilakukan adalah menetapkan kebutuhan untuk semua elemen sistem.
- b) *Software requirements analysis*. Proses pengumpulan kebutuhan dilakukan secara intensif dan terfokus, khususnya pada perangkat lunaknya. Untuk mengerti sistem yang akan dibangun, seorang pembuat sistem harus memahami informasi yang dibutuhkan oleh perangkat lunak itu nantinya, fungsi-fungsi, *performance* dan antarmuka yang akan digunakan.
- c) *Design*. Perangkat lunak sebenarnya di fokuskan pada beberapa langkah proses yaitu struktur data, arsitektur perangkat lunak, prosedur secara detil dan karakteristik antarmuka. Proses perancangan ini diperkirakan kualitas sebelum proses pengkodean dimulai.

- d) *Coding*. Design yang telah dibuat harus diterjemahkan kedalam bentuk yang dapat dibaca oleh mesin yaitu berupa *coding* komputer.
- e) *Testing*. Setelah pengkodean tahap selanjutnya pengujian program. Pengujian ini bertujuan untuk menemukan kesalahan dan memastikan keluaran yang diharapkan.
- f) *Maintenance*. Pemeliharaan ini dilakukan dengan mengadakan modifikasi perangkat lunak setelah diserahkan kepada pemakai. Perubahan ini terjadi jika terdapat kesalahan di dalam sistem atau adanya perubahan lingkungan perangkat lunak seperti perubahan perangkat keras atau sistem operasi, atau untuk meningkatkan fungsi dan kinerja dari perangkat lunak itu sendiri.

2.7. Alat Bantu Perancangan

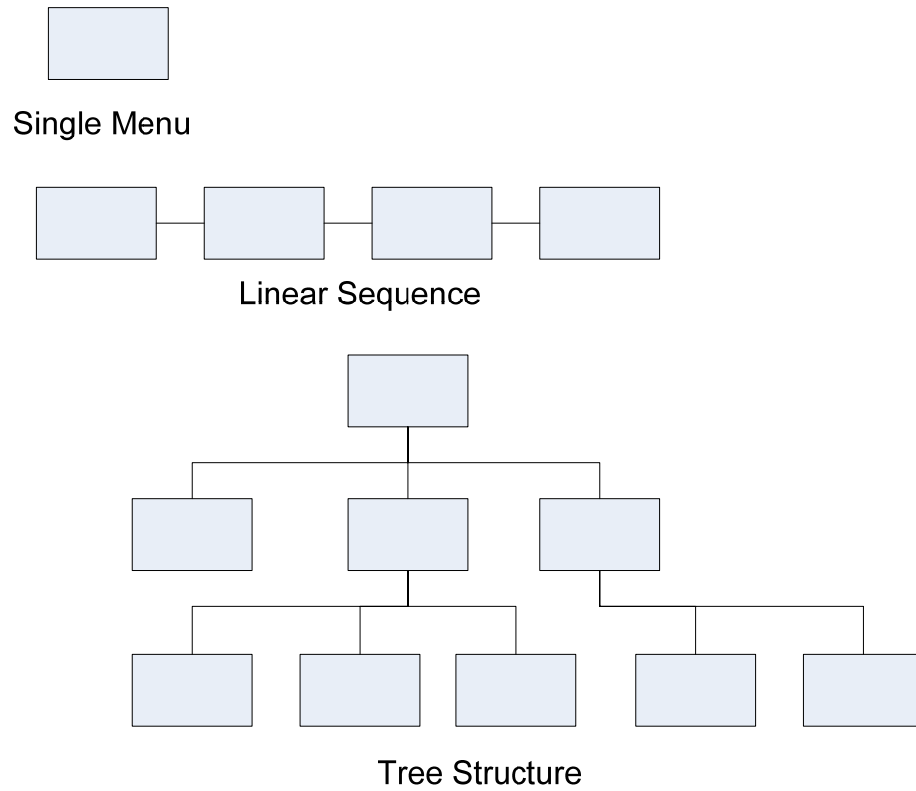
Dalam merancang program yang bersifat prosedural maka dibutuhkan diagram-diagram yaitu struktur menu, *structure chart*, *state transition diagram (STD)* dan pseudocode.

2.7.1. Struktur Menu

Menurut Shneiderman (1998, pp239) design struktur menu dapat dibagi menjadi tiga yaitu :

1. *Single menu*.
2. *Linear sequence menu*.
3. *Tree structure menu*.

Untuk melihat lebih jelasnya dapat dilihat pada Gambar 2.3.

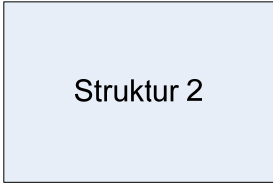
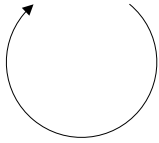
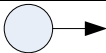



Gambar 2.3 Design Struktur Menu

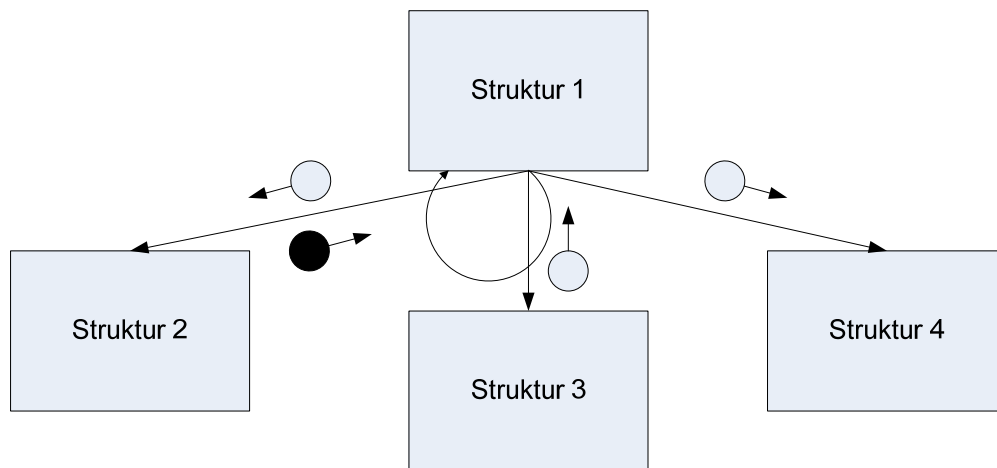
2.7.2. *Structure Chart*

Menurut Pressman (1992, p 379) *structure chart* adalah representasi dari struktur program. Dasar dari isi struktur chart dapat dilihat pada Tabel 2.3.

Tabel 2.3 Isi dari *Struktur Chart*

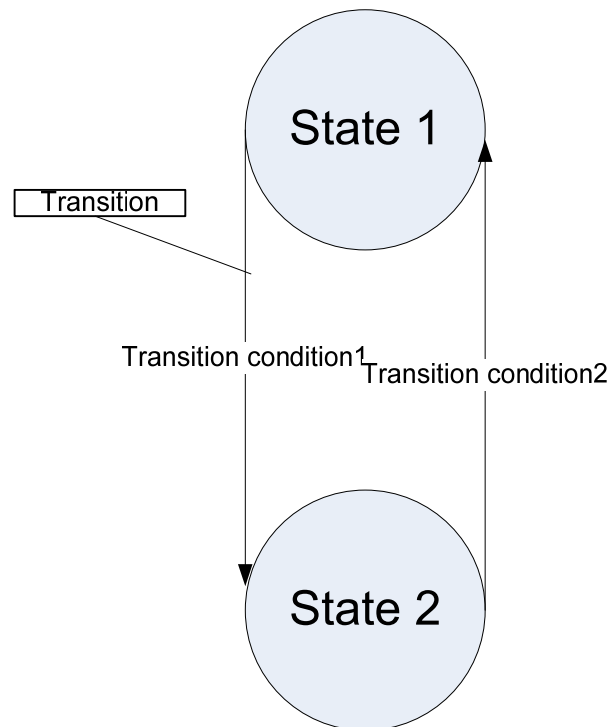
Simbol	Keterangan
	Modul-Modul Program
	Pengulangan
	Komunikasi data antara modul
	Komunikasi pesan antara modul

Sedangkan Contoh gambar *structure chart* dapat dilihat pada Gambar 2.4.

Gambar 2.4 *Structure Chart*

2.7.3. State Transition Diagram

Menurut Pressman (1992, p217) *State Transition Diagram* (STD) adalah diagram yang menggambarkan bagaimana *state* dihubungkan dengan *state* yang lain pada satu waktu. *State transition diagram* menunjukkan bagaimana sistem bekerja sebagai akibat dari kejadian eksternal. STD dapat digambarkan pada Gambar 2.5.



Gambar 2.5 State Chart Diagram

State menunjukkan keadaan atau kegiatan yang menjelaskan bagian tertentu dari proses.

Transition menunjukkan perubahan kondisi dari suatu sistem.

Transition condition menunjukkan kondisi atau syarat pada lingkungan eksternal yang dapat menyebabkan perubahan dari satu *state* ke *state* lainnya.

2.7.4. *Pseudocode*

Menurut Robertson (2000, pp6) *pseudocode* mempunyai arti sebagai suatu pernyataan yang ditulis dalam bahasa Inggris, setiap intruksi ditulis dalam garis terpisah, dan kata-kata dan pemberian spasi (*indentation*) digunakan untuk struktur control tertentu.

Pseudocode dapat membantu dalam perancangan perangkat lunak. Di dalam penulisan *pseudocode* juga tidak ada standarisasi. *Pseudocode* mempunyai tujuan agar mudah dibaca manusia daripada dibaca oleh komputer.

2.8. Interaksi Manusia dan Komputer

Suatu sistem yang baik harus memiliki suatu design antarmuka pengguna yang baik sehingga membuat pengguna tertarik untuk menggunakannya. Menurut Shneiderman (1998, pp74-75) dalam menentukan desain antarmuka pengguna yang baik terdapat “delapan aturan emas” yaitu :

a) Berusaha untuk konsisten.

Konsisten ini adalah konsisten dalam penggunaan bentuk dan ukuran font, pemberian warna pada latar belakang dan tulisan, pembuatan *layout*. Konsisten ini memberikan kemudahan bagi pengguna dalam menjalankan aplikasi sehingga pengguna tidak perlu mengingat dan mempelajari perbedaan-perbedaan dalam interaksi.

b) Memungkinkan pengguna menggunakan *shortcut* sesering mungkin.

Pengurangan jumlah interaksi melalui fasilitas *shortcuts* memberikan manfaat bagi pengguna dalam memberikan waktu respon dan waktu tampilan yang cepat. Dan fasilitas ini disukai oleh pengguna.

- c) Menawarkan umpan balik yang informatif.

Setiap tindakan dari pengguna, sistem harus merespon atau menyediakan umpan balik. Umpan balik bisa berupa tampilan ataupun suara sehingga pengguna mengetahui bahwa perangkat lunak tersebut memberikan respon.

- d) Merancangan dialog untuk hasil akhir.

Susunan dari tindakan yang harus diatur ke dalam group-group memiliki sebuah permulaan, pertengahan dan akhir. Umpan balik yang informatif dalam penyelesaian tindakan-tindakan memberikan kepuasan hasil akhir kepada pemakai.

- e) Menawarkan pencegahan error dan kemudahan untuk mengatasinya.

Sistem didesign sedemikian rupa agar pengguna tidak membuat kesalahan serius. Jika pengguna membuat kesalahan, sistem harus mampu menemukan kesalahan dan menawarkan mekanisme penanganan yang mudah, membangun dan spesifik untuk memperbaiki kesalahannya.

- f) Mengizinkan pembalikan tindakan dengan mudah.

Sebisa mungkin harus terdapat pembalikan tindakan. Karena hal ini dapat mengurangi, menghilangkan kecemasan karena pengguna tahu bahwa terjadi kesalahan maka pengguna dapat membalik ke keadaan sebelumnya. Jadi mendorong pengguna untuk mengeksplorasi lebih mendalam terhadap sistem tersebut.

- g) Mendukung tempat pengendalian internal.

Pengguna yang berpengalaman sangat menginginkan bahwa mereka dapat mengendalikan sistem tersebut dan juga dapat merespon tindakan mereka.

h) Mengurangi muatan *memory* jangka pendek.

Manusia mempunyai keterbatasan dalam mengingat sehingga memerlukan tampilan sederhana, tampilan halaman-halaman dapat digabungkan, dan pergerakan Windows dapat dikurangi.

Proses desain antarmuka pengguna bukan merupakan proses yang rumit. Berikut adalah langkah-langkah pokok yang dapat dilakukan adalah (Whitten et al., 2004, p635):

- a) Petakan dialog antarmuka pengguna
- b) Buat prototype dialog dan antarmuka pengguna
- c) Dapatkan *feedback* dari pengguna
- d) Jika perlu, kembali ke langkah 1 atau 2.

Suatu program yang interaktif dan baik harus bersifat *user friendly*. Nielsen (2003) menjelaskan 5 kriteria yang harus dipenuhi oleh suatu program yang *user friendly* yaitu (Galitz, 2007, p64) :

- a) Dapat dipelajari (*Learnability*): seberapa mudah pengguna pertama kali mengerjakan tugas-tugas dasar?
- b) Efisien: Seberapa cepat pengguna dapat melakukan tugas?
- c) Kesalahan: Berapa banyak kesalahan yang pengguna buat, Seberapa parah pengguna salah, dan seberapa mudah pengguna dapat memperbaiki kesalahan?
- d) Dapat diingat (*Memorability*) ketika pengguna kembali menggunakan design setelah beberapa waktu tidak menggunakan, seberapa mudah pengguna dapat menggunakan kembali design tersebut?
- e) Kepuasan: Bagaimana kepuasan pengguna dalam menggunakan design?

Saat ini sebagian besar antarmuka pengguna didesain menggunakan prototipe yang dapat dibuat dengan cepat. Prototipe ini dihasilkan dengan menggunakan pengembang aplikasi cepat seperti Microsoft Visual Basic, Inpire's JBuilder (for java), atau IBM Visual Age (untuk berbagai bahasa).