BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Kriptografi

Kriptografi secara umum merupakan ilmu dan seni untuk menjaga kerahasiaan berita. Kriptografi juga dapat diartikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Pada kriptografi tidak semua aspek keamanan informasi akan ditangani.

Ada empat tujuan mendasar dari kriptografi yang juga merupakan aspek keamanan informasi, yaitu sebagai berikut :

1. Kerahasiaan

   Merupakan layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki kunci rahasia atau otoritas untuk membuka informasi yang telah disandikan.

2. Integritas data

   Berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk dapat menjaga integritas data, suatu sistem harus memiliki kemampuan untuk mendeteksi manipulasi data yang dilakukan pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pendistribusian data lain kedalam data yang asli.

3. Autentikasi

   Berhubungan dengan identifikasi, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan harus diautentikasi keasliannya, isi datanya, waktu pengiriman dan lain sebagainya.

4. Non-repudiasi

   Merupakan usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

2.2 Dasar Matematis

Cryptographic algorithm adalah fungsi matematika yang digunakan untuk enksripsi dan dekripsi. Umumnya terdapat dua fungsi yang berhubungan yaitu satu untuk enkripsi dan satu untuk dekripsi. Komponen atau elemen kriptografi terdiri dari :

1. *Plaintext*

   Plaintext adalah sumber berita atau pesan atau teks asli.

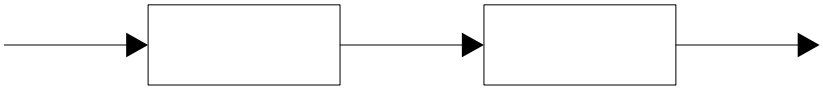
2. *Ciphertext*

   Ciphertext adalah teks asli yang sudah diproses (diacak, digantikan).

3. Algoritma & kunci

   Misal : subsitusi (algoritma) & *number of shift* (kunci) Pemisahan algoritma & kunci ditemukan oleh Auguste Kerckhofs von Niewenhof 183).

Enkripsi merupakan proses pengkodean sebuah pesan sehingga isi dari pesan tersebut tidak diketahui. Dekripsi adalah proses kebalikan dari enkripsi yaitu mentranformasi pesan yang dienkripsi kembali menjadi bentuk semula. Sebuah sistem enkripsi dan dekripsi disebut *cryptosystem*. Bentuk asli dari sebuah pesan disebut dengan *plaintext* dan bentuk asli yang dienkripsi disebut *ciphertext.*



Gambar 2.1. Enkripsi dan Dekripsi

Plaintext dinotasikan dengan **M** (Message), yang dapat berupa *bit tream*, *file text*, *digitized voice stream*, *digital video image* atau lebih singkatnya **M** adalah data binary. *Ciphertext* dinotasikan dengan **C**, juga berupa *data binary* yang kadang-kadang mempunyai ukuran yang sama dengan **M**, lebih kecil dari **M** atau lebih besar dari **M**. Fungsi enkripsi **E**, berfungsi untuk mengubah **M** menjadi **C**, dalam matematika dinotasikan dengan :

$$\mathbf{E(M) = C.}$$

Fungsi dekripsi **D,** berfungsi untuk mengubah **C** menjadi **M,** dalam matematika dinotasikan dengan :

Plaintext Encrypt Ci

$$\mathbf{D(C) = M.}$$

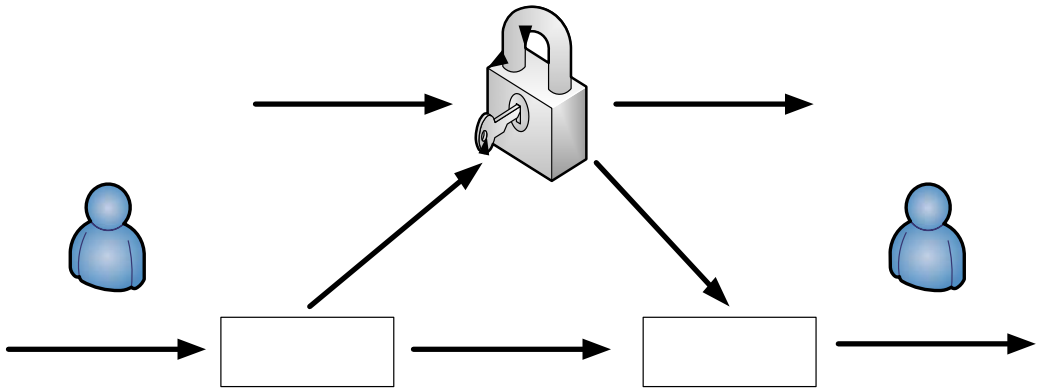
Apabila fungsi enkripsi **E** ditambahkan sebuah kunci **K** maka notasi matematikanya menjadi $\mathbf{E_K(M) = C}$ dan untuk fungsi dekripsi **D** maka notasi matematikanya menjadi $\mathbf{D_K(C) = M.}$

2.3 Jenis Kriptografi

Kriptografi terbagi menjadi dua bagian, yaitu :

1. *Private key cryptosystem* (sistem kripto kunci pribadi)

   Yang dimaksud dengan sistem kripto kunci pribadi disini bersifat simetrik, dimana kunci atau *password* yang digunakan untuk mengunci dan membuka teks yang satu/sama. Dalam penggunaan kripto kunci pribadi, pengiriman kunci akan membutuhkan saluran khusus, dimana kelebihannya akan memberikan proses yang lebih cepat dibandingkan kripto kunci publik.



Gambar 2.2 Diagram Blok Kriptografi Dengan Kunci Pribadi

2. *Public key cryptosystem* (sistem kripto kunci publik)

   Yang dimaksud dengan sistem kripto kunci publik disini bersifat asimetrik, dimana kunci atau *password* yang digunakan untuk mengunci dan membuka teks berbeda. Dalam penggunaanya kripto kunci publik membutuhkan waktu yang cukup lama (membutuhkan komputasi yang tinggi).

   Kripto kunci publik mempunyai kelebihan dalam penguncian data, yaitu kunci dalan pengenkripsian lebih sedikit dibandingkan dengan kripto kunci privat. Kripto kunci publik disini membutuhkan penyimpanan kunci publik

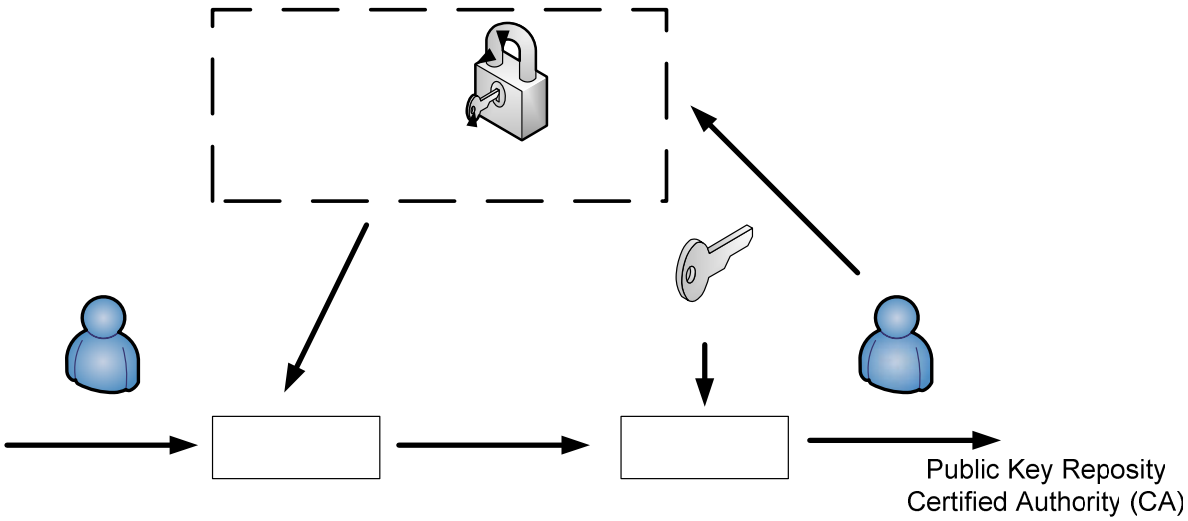
Kunci Rahasia Yang
Dipakai Bersama

yang terpercaya. Selain itu penggunaan kripto kunci publik akan komplek dikarenakan kemungkinan besar akan ada pihak ketiga.

Menurut stalling, proses enkripsi *public key* sederhana melibatkan empat tahap, yaitu sebagai berikut :

1. Setiap pengguna didalam jaringan membuat sepasang kunci untuk digunakan sebagai kunci enkripsi dan dekripsi dari pesan yang akan diterima.
2. Pengguna mempublikasikan kunci enkripsinya dengan menempatkan kunci publiknya ke tempat umum. Pasangan kunci yang lain tetap dijaga kerahasiaannya.
3. Jika pengguna A ingin mengirimkan sebuah pesan ke pengguna B, ia akan mengenkripsi pesan tersebut dengan mengunakan kunci publik user B.
4. Pada saat pengguna B ingin membuka sebuah pesan dari pengguna A, ia akan menggunakan kunci pribadinya sendiri. Tidak ada pihak lain yang bisa mendekripsi pesan itu, karena hanya B sendiri yang mengetahui kunci pribadi.

Sampai saat ini hanya ada beberapa sistem kriptografi asimetrik yang dipublikasikan. Yang paling berhasil sejauh ini adalah algoritma RSA yang memenuhi keenam persyaratan *public-key* diatas. Disamping itu, algoritma enkripsi *public key* yang lain adalah LUC, DSS, Diffie Hellman dan lain-lain.



Gambar 2.3 Diagram Blok Kriptografi Dengan Kunci Publik

2.4 Algoritma *RSA (Rivest-Shamir-Adleman)*

Algoritma RSA (*Rivest-Shamir-Adleman*) mulai diperkenalkan pada tahun 1976 oleh tiga orang peneliti dari MIT (*Massachussetes Institute of Technology*), yaitu *Ronald Rivest, Adi Shamir,* dan *Leonard Adleman*. Seperti sistem kunci publik lainnya, algoritma RSA ini berasal dari suatu permasalahan sukar (*hard problem*) teori matematika, yang dalam hal ini RSA berasal pada permasalahan pencarian faktor prima pada bilangan yang sangat besar.

Kunci enkripsi dan dekripsi pada algoritma ini keduanya merupakan bilangan bulat. Kunci untuk enkripsi merupakan kunci publik, sedangkan kunci untuk dekripsi merupakan kunci rahasia yang hanya diketahui oleh penerima pesan. Untuk membuat pasangan kunci publik dan rahasia, langkah-langkah dasarnya adalah sebagai berikut:

1. Pilih dua buah bilangan prima, kita pilih p dan q. Dari dua bilangan prima tersebut, dapat kita peroleh bilangan modulus $n = p.q$.

2. Pilih bilangan ketiga, *e*, yang bilangannya relatif prima (faktor pembagi terbesarnya =1) terhadap hasil perkalian (*p*-1) (*q*-1).
3. Hitung sebuah nilai *d* dari persamaan (*ed*-1)/[(*p*-1)(*q*-1)]. Bilangan *d* tersebut adalah kunci deskripsinya.

Keamanan algoritma *RSA* terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya, yang dalam hal ini r = $p \times q$.

Sekali *r* berhasil difaktorkan menjadi *p* dan *q*, maka $\phi(r) = (p - 1)(q - 1)$ dapat dihitung. Selanjutnya, karena kunci enkrispi *PK* diumumkan (tidak rahasia), maka kunci dekripsi *SK* dapat dihitung dari persamaan $PK \cdot SK \equiv 1 \pmod{\phi(r)}$.

Penemu algoritma *RSA* menyarankan nilai *p* dan *q* panjangnya lebih dari 100 digit. Dengan demikian hasil kali $r = p \times q$ akan berukuran lebih dari 200 digit. Menurut Rivest dan kawan-kawan, usaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun! (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik).

Algoritma yang paling mangkus untuk memfaktorkan bilangan yang besar belum ditemukan. Inilah yang membuat algoritma *RSA* tetap dipakai hingga saat ini. Selagi belum ditemukan algoritma yang mangkus untuk memfaktorkan bilangan bulat menjadi faktor primanya, maka algoritma *RSA* tetap direkomendasikan untuk menyandikan pesan.

2.5 Enkripsi dan Dekripsi

2.5.1 Enkripsi

Proses utama dalam suatu algoritma kriptografi adalah enkripsi dan dekripsi. Enkripsi merubah sebuah *plaintext* ke dalam bentuk *ciphertext*. Pada mode *ECB* (*Electronic Codebook*), sebuah blok pada *plaintext* dienkripsi ke dalam sebuah blok *ciphertext* dengan panjang blok yang sama.

Blok cipher memiliki sifat bahwa setiap blok harus memiliki panjang yang sama (Misal 128 bit). Namun apabila pesan yang dienkripsi memiliki panjang blok terakhir yang tidak tepat 128 bit, maka diperlukan mekanisme *padding*, yaitu penambahan bit-bit *dummies* untuk menggenapi menjadi panjang blok yang sesuai, biasanya *padding* dilakukan pada blok terakhir *plaintext*.

2.5.2 Dekripsi

Dekripsi merupakan proses kebalikan dari proses enkripsi, merubah *ciphertext* ke dalam bentuk *plaintext.*

2.6 Konsep Dasar Perhitungan Matematis

2.6.1 Fungsi Phi-Euler

Fungsi Phi-Euler (φ)*n* merupakan fungsi terhadap bilangan bulat positif *n* yang menyatakan banyaknya elemen $\mathbb{Z}n$ yang mempunyai invers terhadap operasi pergandaan. Ingat bahwa $\mathbb{Z}n$ belum tentu merupakan grup terhadap operasi pergandaan. Dengan kata lain, (φ)*n* adalah banyaknya elemen {x, $0 \leq x <$ *n* | gcd(x , *n* = 1 }. Jika $n = pq$ dengan *p* dan *q* adalah bilangan prima, maka (φ)*n* = $(p - 1)(q - 1)$. Jika *n* adalah bilangan prima, maka (φ)*n* = *n* -1̃.

2.6.2 Algoritma Euclide

Algoritma ini digunakan untuk mencari nilai pembagi persekutuan terbesar dari dua bilangan bulat. Algoritma ini didasarkan pada pernyataan berikut ini. Diberikan bilangan bulat r_0 dan r_1 , dengan $r_0 \geq r_1$, kemudian dihitung menggunakan algoritma pembagian :

$r_0 = q_1 r_1 + r_2, \quad 0 < r_2 < r_1$

$r_1 = q_2 r_2 + r_3, \quad 0 < r_3 < r_2$

$r_{n-2} = q_{n-1} r_{n-1} + r_n, \quad 0 < r_n < r_{n-1}$

$r_{n-1} - q_n r_n$

Dari pernyataan di atas, dapat ditunjukkan bahwa :

$\gcd(r_0, r_1) = \gcd(r_1, r_2) = \ldots = \gcd(r_{n-1}, r_n) = (r_n, 0) = r_n.$

Contoh 1. Akan dihitung gcd(40,24).

Jawab : 40 = 1.24 + 16

24 = 1.16 + 8

16 = 2.8

Jadi, gcd(40,24)=8.

2.6.2.1 Algoritma Euclide Diperluas

Algoritma ini merupakan perluasan dari algoritma Euclide (*Extended Euclidean Algorithm*), digunakan untuk mencari invers terhadap operasi pergandaan. Algoritma ini didasarkan pada pernyataan berikut. Diberikan bilangan bulat positif r_0 dan r_1, $r_0 > r_1$. Jika $\gcd(r_0, r_1)$, maka $r_1^{-1} \bmod r_0$ ada. Jika $\gcd(r_0, r_1) \neq 1$, $r_1^{-1} \bmod r_0$ tidak ada. Gunakan rumus rekurensi berikut ini :

$r_0 = 0, t_1 = 1$

$t_j = t_{j-2} - q_{j-1} t_{j-1}, j \geq 2$ …… (1)

Ket: q_j diperoleh dari perhitungan gcd(r_0, r_1) menggunakan algoritma Euclide. Jika gcd(r_0, r_1) = 1, $r_1^{-1} = t_n$, sehingga $r_n = t_n r_1$ atau $1 = t_n r_1$

Contoh 2. Akan dihitung 28^{-1} mod 75 .

Diketahui $r_0 = 75$ dan $r_1 = 28$ *r*. Hitung gcd(75,28) menggunakan algoritma Euclide, yaitu:

75 = 2.28 + 19 $n = 1$, $r_1 = 28$, $q_1 = 2$

28 = 2.19 + 9 $n = 2$, $r2 = 19$, $q_2 = 2$

19 = 2.9 + 1 $n = 3$, $r_3 = 9$, $q_3 = 2$

9 = 9.1 $n = 4$, $r4 = 1$, $q4 = 9$

Jadi, gcd(75,28) = 1, dengan kata lain 28^{-1} mod 75 ada.

Dari penyelesaian algoritma Euclide di atas diperoleh $n = 4$. Selanjutnya, menggunakan rumus rekurensi (1) di atas, diperoleh : (semua perhitungan dilakukan dalam mod 75)

$t_2 = t_0 - q_1 t_1 = 0 - 2.1 = -2 = 73$

$t_3 = t_1 - q_2 t_2 = 1\text{-}1.(\text{-}2) = 3$

$t_4 = t_2 - q_3 t_3 = \text{-}2\text{-}2.3 = \text{-}8 = 67$

$r_4 = t_4 r_1 = 67.28 = 1 \leftrightarrow 67 = 28^{-1}$

Dari hasil terakhir di atas, diperoleh 28^{-1} mod 75= 67.

2.6.3 Metode Fast Exponentiation

Metode ini digunakan untuk menghitung operasi pemangkatan besar bilangan bulat modulo dengan cepat. Metode ini memanfaatkan ekspansi biner dari eksponennya dan didasarkan pada pernyataan berikut ini:

$$g^{2i+1} = (g^{2i})^{2}$$

Untuk lebih jelasnya mengenai langkah-langkah metode *fast exponentiation*, perhatikan contoh berikut ini.

Contoh 3. Akan dihitung 6^{73} mod100 .

Jawab :

$73 = 1.2^6 + 1.2^3 + 1.2^0$ atau $73 = (1001001)_2$

(semua perhitungan dilakukan dalam mod 100)

$6^{2^0} = 6$, $6^{2^1} = 36$, $6^{2^2} = 36^2 = 96$, $6^{2^3} = 16$,

$6^{2^4} = 16^2 = 56$, $6^{2^5} = 56^2 = 36$, $6^{2^6}\ 56^2 = 96$

Sehingga diperoleh:

$6^{73} = 6.6^{2^3}.6^{2^6} = 6.16.96 = 16$

Jadi 6^{73} mod100 = 16

2.7 Proses Algoritma RSA

Proses-proses yang digunakan oleh RSA terdiri dari pembentukan kunci, enkripsi, dan dekripsi, yaitu sebagai berikut :

2.7.1 Pembentukan Kunci

Berikut ini adalah proses pembentukan kunci. Proses ini dilakukan oleh pihak penerima, dalam hal ini adalah B.

1. Pilih bilangan prima *p* dan *q*.
2. Hitung $n = pq$.

3. Hitung j (n) = (p -1)(q -1) .

4. Pilih sebarang bilangan b, 1 < b < j (n) , dengan gcd(b,j (n)) = 1.

5. Hitung invers dari b, yaitu a = b-1 modj (n) .

6. Kunci publik: (n, b) dan kunci rahasia: a.

Untuk mempermudah pemahaman dalam sandi RSA, maka *plaintext* yang digunakan hanya berupa bilangan 0 s/d 25 yang berkorespondensi dengan huruf a s/d z. tetapi pada penggunaan yang sebenarnya, digunakan korespondensi khusus seperti kode ASCII, serta bilangan-bilangan yang sangat besar. Dalam pemilihan p dan q harus memenuhi $n = pq$ lebih dari atau sama dengan nilai plainteks yang mungkin. Dalam hal ini $n = pq \geq 25$.

Tabel 2.1 Korespondensi

a	b	c	d	e	f	g	h	i	j	k	l	m
1	2	3	4	5	6	7	8	9	10	11	12	13

n	o	p	q	r	s	t	u	v	w	x	y	z
14	15	16	17	18	19	20	21	22	23	24	25	26

2.7.2 Enkripsi RSA

Berikut ini adalah proses enkripsi RSA. Dilakukan oleh pihak pengirim, dalam hal ini adalah A. Seluruh perhitungan pemangkatan bilangan modulo dilakukan menggunakan metode *fast exponentiation*.

1. Ambil kunci publik (n,b).

2. Pilih *plaintext* m, dengan $0 \leq m \leq n$ -1 .

3. Hitung $c = m^b \bmod n$.

4. Diperoleh *ciphertext c*, dan kirimkan kepada B.

2.7.3 Dekripsi RSA

Berikut ini adalah proses dekripsi RSA. Dilakukan oleh pihak penerima *ciphertext*, yaitu B.

1. Ambil kunci publik (*n*,*b*) dan kunci rahasia *a*.

2. Hitung $m = c^a \bmod n$.

2.8 Pengertian Jaringan Komputer

Jaringan komputer merupakan sekelompok komputer otonom yang saling berhubungan antara satu dengan yang lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program-program, penggunaan perangkat keras bersama seperti *printer, harddisk*, dan sebagainya.

2.8.1 Jenis – Jenis Jaringan Komputer

Jenis-jenis jaringan komputer terdiri dari :

1. *Local Area Network (LAN)*

Local Area Network (LAN) merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. *LAN* seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *Workstation* dalam kantor perusahaan atau pabrik-pabrik untuk memakai bersama *resource* (misalnya *printer, scanner*) dan saling bertukar informasi.

LAN dapat dibedakan dari jenis jaringan lainnya berdasarkan tiga karakteristik yaitu ukuran, teknologi transmisi dan topologinya. *LAN* mempunyai ukuran yang terbatas, yaitu berarti bahwa waktu transmisi pada keadaan terburuknya terbatas dan dapat diketahui sebelumnya. Dengan mengetahui keterbatasannya menyebabkan adanya kemungkinan untuk menggunakan jenis desain tertentu. Hal ini juga memudahkan manajemen jaringan.

2. Metropolitan Area Network (MAN)

Metropolitan Area Network (MAN) pada dasarnya merupakan versi *LAN* yang berukuran lebih besar dan biasanya memakai teknologi yang sama dengan *LAN*. *MAN* dapat mencakup kantor-kantor perusahaan yang berdekatan dan dapat dimanfaatkan untuk keperluan pribadi swasta atau umum. *MAN* biasanya mampu menunjang data dan suara, dan bahkan dapat berhubungan dengan jaringan televisi kabel.

MAN hanya memiliki sebuah kabel atau dua buah kabel dan tidak mempunyai elemen *switching*. *Switching* dapat digunakan atau berfungsi untuk mengatur paket melalui beberapa *output* kabel. Adanya elemen *switching* membuat rancangan menjadi lebih sederhana.

3. Wide Area Network (WAN)

Wide Area Network (WAN) adalah sebuah jaringan yang memiliki jarak yang sangat luas, karena radiusnya mencakup sebuah Negara dan benua. Pada sebagian besar *WAN*, komponen yang dipakai dalam berkomunikasi biasanya terdiri dari dua komponen, yaitu :

kabel transmisi dan elemen *switching*. Kabel transmisi berfungsi untuk memindahkan bit-bit dari satu komputer ke komputer lainnya, sedangkan elemen *switching* disini adalah sebuah komputer khusus yang digunakan untuk menghubungkan dua buah kabel transmisi atau lebih. Saat data yang dikirimkan sampai ke kabel penerima, elemen *switching* harus memilih kabel pengirim untuk meneruskan pesan-pesan tersebut.

4. Jaringan tanpa kabel

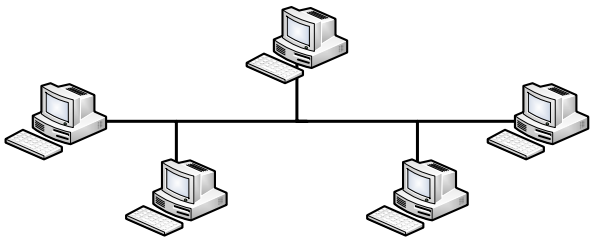
komputer *mobile* seperti komputer *notebook* dan *Personal Digital Assitant (PDA),* merupakan cabang industri komputer yang paling cepat pertumbuhannya. Banyak pemilik jenis komputer tersebut yang sebenarnya telah memiliki mesin-mesin desktop yang terpasang pada *LAN* atau *WAN* tetapi karena koneksi kabel tidaklah mungkin dibuat di dalam mobil atau pesawat terbang, maka banyak yang tertarik untuk memiliki komputer dengan jaringan tanpa kabel. Jaringan tanpa kabel mempunyai berbagai manfaat, yang telah umum dikenal adalah kantor *portable*. Orang yang sedang dalam perjalanan seringkali ingin menggunakan peralatan elektronik *portable*nya untuk mengirim atau menerima telepon, *fax*, *e-mail*, membaca file jarak jauh, *login* ke mesin jarak jauh dan sebagainya, juga ingin melakukan hal-hal tersebut di mana saja, darat, laut, udara. Jaringan tanpa kabel sangat berguna untuk mengatasi permasalahan-permasalahan tersebut.

2.8.2 Topologi Jaringan Komputer

Topologi jaringan komputer adalah pola hubungan antar terminal dalam suatu jaringan komputer. Bentuk topologi terbagi dua menjadi topologi secara logik dan topologi secara fisik. Bentuk topologi yang paling utama terdiri dari topologi *bus*, *ring* dan *star*, yaitu sebagai berikut :

1. Bus

Pada topologi *bus* semua terminal terhubung ke jalur komunikasi. Informasi yang dikirim akan melewati semua terminal pada jalur tersebut. Jika alamat yang tercantum dalam data atau informasi yang dikirim sesuai dengan alamat terminal yang dilewati, maka data atau informasi tersebut akan diterima dan diproses, jika alamat tersebut tidak sesuai, maka informasi tersebut akan diabaikan oleh terminal yang dilewati.

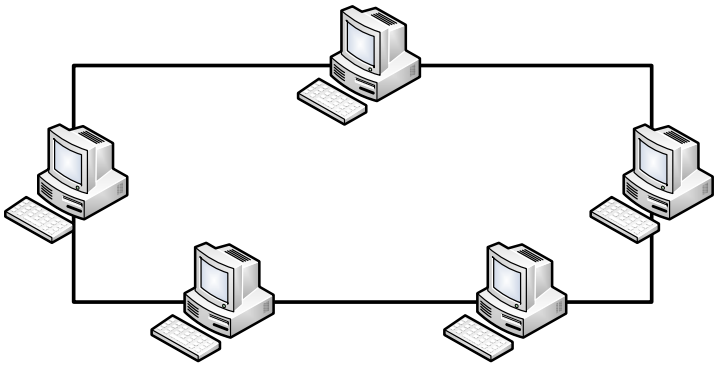


Gambar 2.4 Topologi *Bus*

2. Ring

Sistem *ring* menggunakan koneksi antar node berbentuk melingkar, sistem ini dikembangkan oleh IBM. Topologi ini mirip dengan topologi *bus*, tetapi kedua terminal yang berada di ujung saling dihubungkan. Setiap informasi yang diperoleh diperiksa alamatnya oleh terminal yang

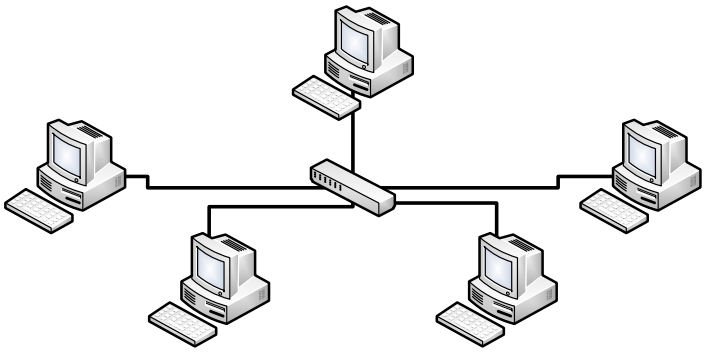
dilewatinya. Jika bukan untuknya, informasi dilewatkan sampai menemukan alamat yang benar. Setiap terminal dalam topologi ini sangat tergantung, sehingga jika terjadi kerusakan pada satu terminal maka seluruh jaringan akan terganggu.



Gambar 2.5 Topologi *Ring*

3. Star

Sistem *star* menggunakan konsentrator untuk koneksi semua node, konsentrator ini bisa berupa *hub* ataupun *switch*. Pada topologi *star*, semua terminal pusat bertindak sebagai pengatur dan pengendali semua komunikasi data yang terjadi. Terminal-terminal lain terhubung padanya dan pengiriman data dari satu terminal ke terminal lainnya melalui terminal pusat. Terminal pusat akan menyediakan jalur komunikasi khusus untuk dua terminal yang akan berkomunikasi. Sebagai salah satu contoh penggunaan topologi ini adalah jaringan telepon.



Gambar 2.6 Topologi *Star*

2.8.3 Manfaat Jaringan Komputer

Manfaat komputer bagi manusia dapat dikelompokkan pada jaringan untuk perusahaan, jaringan untuk umum, dan masalah sosial jaringan.

1. Jaringan untuk perusahaan atau organisasi

Dalam membangun jaringan komputer di perusahaan atau organisasi, ada beberapa *resource sharing*, reliabilitas tinggi, lebih ekonomis, skabilitas, dan media komunikasi. *Resource sharing* bertujuan agar seluruh program, peralatan, khususnya data dapat digunakan oleh setiap orang yang ada pada jaringan tanpa terpengaruh oleh lokasi *resource* dan *user* (pemakai). Reliabilitas tinggi yaitu adanya sumber-sumber alternatif pengganti jika terjadi masalah pada salah satu perangkat dalam jaringan. Komputer yang lebih kecil memiliki rasio harga atau kinerja yang lebih baik dibanding dengan komputer besar. komputer *mainframe* memiliki kekuatan kurang lebih sepuluh kali lipat kecepatan komputer pribadi, akan tetapi harga *mainframe* seribu kalinya lebih mahal. Dengan selisih rasio harga/kinerja yang cukup besar ini menyebabkan perancang sistem

memilih membangun sistem yang terdiri dari komputer-komputer pribadi dibandingkan dengan *mainframe*.

Skalabilitas yaitu kemampuan untuk meningkatkan kinerja sistem secara berangsur-angsur sesuai dengan beban pekerjaan dengan hanya menambahkan sejumlah prosesor. pada komputer *mainframe* yang tersentralisasi, jika sistem sudah jenuh, maka komputer harus diganti dengan komputer yang mempunyai kemampuan lebih besar. hal ini membutuhkan biaya yang sangat besar dan dapat menyebabkan gangguan terhadap kontinyuitas kerja para pemakai. Sebuah jaringan komputer mampu bertindak sebagai media komunikasi yang baik bagi para pegawai yang terpisah secara jauh.

2. Jaringan untuk umum

Jaringan komputer akan memberikan layanan yang berbeda kepada perorangan di rumah-rumah dibandingkan dengan layanan yang diberikan pada perusahaan. Terdapat tiga hal pokok yang menjadi daya tarik jaringan komputer pada jaringan yaitu :

a. Akses ke informasi yang berada ditempat yang lebih jauh.

b. Komunikasi orang ke orang.

c. Hiburan interaktif.

3. Masalah sosial jaringan

Penggunaan jaringan oleh masyarakat luas akan menyebabkan masalah-masalah sosial, etika, dan politik. Internet telah masuk ke segala penjuru kehidupan masyarakat, semua orang dapat memanfaatkannya tanpa memandang status sosial, usia dan jenis kelamin. Penggunaan internet tidak akan menimbulkan masalah selama subyeknya terbatas pada topik-topik teknis, pendidikan atau hobi, hal-hal dalam batas norma-norma kehidupan.

4. Layanan-layanan dalam jaringan komputer

a. Layanan kirim data

Layanan ini berfungsi untuk melakukan proses kirim (*send*) ataupun menerima data (*receive*).

b. Layanan *chatting* (percakapan dalam jaringan lokal atau internet)

Chatting adalah percakapan dalam bentuk tulisan yang diselenggarakan melalui jaringan internet. Pelaksanaanya tidak berbeda dengan percakapan telepon yang dapat dilaksanakan dengan siapapun/dimanapun dalam jangkauan jaringan telepon baik melalui saluran maupun dengan transmisi. Perbedaanya terletak pada bentuknya yang berupa komunikasi tulisan. Terdapat beberapa *web site* yang juga menyediakan fasilitas *chatting* bagi penggunanya dan ada pula *web* yang khusus untuk *chatting*. Pengguna hanya perlu memonitor nama-nama pengguna yang sedang *online* dalam *chatting*, selanjutnya menawarkan diri untuk untuk

bergabung dalam *chatting,* pengguna dapat pula terlebih dahulu mengadakan perjanjian untuk *chatting* dengan pengguna lain.

2.9 *Visual Basic*

Microsoft Visual Basic adalah bahasa pemrograman yang digunakan untuk membuat aplikasi *Windows* yang berbasis grafis (GUI – *Graphical User Interface*). Visual Basic merupakan *event-driven programming* (pemrograman terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa *event*/kejadian tertentu (tombol diklik, menu dipilih, dan lain-lain). Ketika *event* terdeteksi, kode yang berhubungan dengan *event* (prosedur *event*) akan dijalankan.

2.9.1 Sejarah Singkat Visual Basic

Berikut ini adalah point-point penting dalam sejarah perkembangan Visual Basic, sebagai berikut :

1. Visual Basic pertama kali diperkenalkan tahun 1991.
2. Berikutnya dirilis Visual Basic 3.0 tahun 1993.
3. Visual Basic 4.0 dirilis pada akhir tahun 1995 (tambahan dukungan untuk aplikasi 32 bit).
4. Visual Basic terbaru adalah versi 6.0 yang dirilis pada akhir tahun 1998.

Microsoft umumnya membuat tiga edisi Visual Basic yaitu :

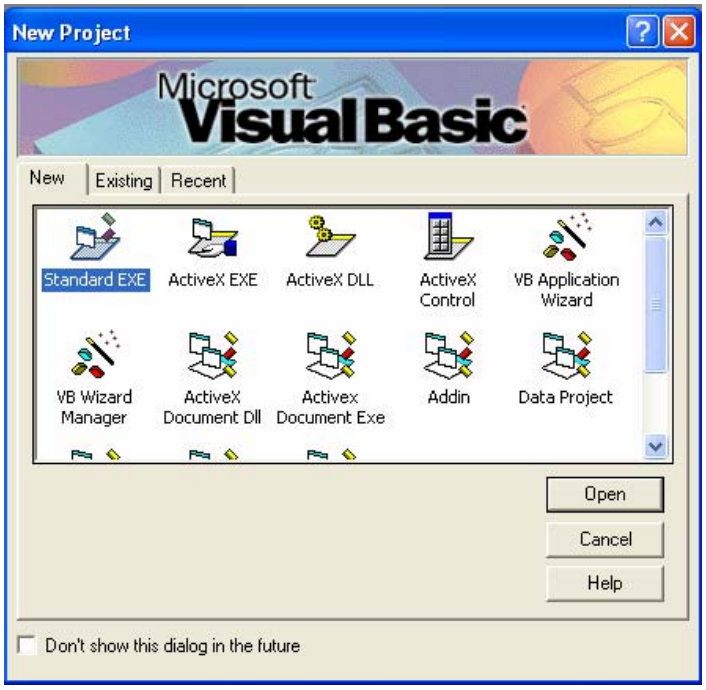
a. *Standard Edition* merupakan produk dasar.
b. *Profesional Edition* berisi tambahan *Microsoft Jet Data Access Engine* (*database*) dan pembuatan *server OLE automation*.

c. *Enterprise Edition* adalah edisi *client-server*.

2.9.2 IDE Visual Basic

Langkah awal dari belajar Visual Basic adalah mengenal IDE (Integrated Developement Environment) Visual Basic yang merupakan Lingkungan Pengembangan Terpadu bagi programmer dalam mengembangkan aplikasinya. Dengan menggunakan IDE programmer dapat membuat user interface, melakukan koding, melakukan testing dan debuging serta menkompilasi program menjadi executable. Penguasaan yang baik akan IDE akan sangat membantu programmer dalam mengefektifkan tugas-tugasnya sehingga dapat bekerja dengan efisien

Sesaat kita aktif di IDE Visual Basic, maka kita akan dihadapkan kepada suatu pilihan terhadap jenis Project yang ingin kita buat sebagaimana yang ditunjukkan oleh Gambar 2.7.
Sebagai langkah awal dari proses belajar adalah memilih project Standard EXE.



Gambar 2.7 *Dialog box New Project*

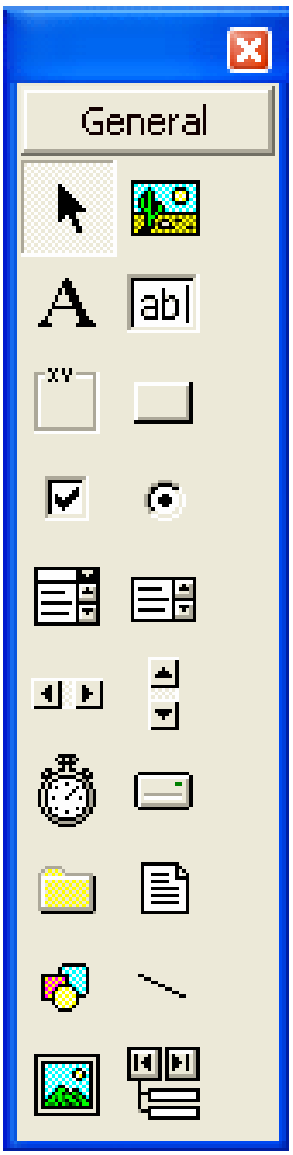
2.9.3 Jendela IDE

IDE Visual Basic 6 menggunakan model MDI (Multiple Document Interface). Berikut ini adalah gambar yang menunjukan bagian-bagian dan nama-nama jendela yang dapat tampil pada IDE Visual Basic. Mungkin pada IDE anda hanya ditampilkan sebagian jendela , karena penampilan dari jendela-jendela tersebut dapat diatur dengan perintah pada menu View. Sebagai langkah awal dari proses belajar, tidak semua jendela akan kita gunakan, tetapi hanya beberapa yang penting, sedangkan yang lainnya bersifat khusus.

Adapun jendela-jendela yang perlu kita perhatikan adalah sebagai berikut:

1. **Menu Bar**, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll
2. **Main Toolbar**, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
3. **Jendela Project**, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan icon Toggle Folders untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama. Anda dapat menggunakan Ctrl+R untuk menampilkan jendela project, ataupun menggunakan icon Project Explorer.
4. **Jendela Form Designer**, jendela ini merupakan tempat anda untuk merancang user interface dari aplikasi anda. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.

5. **Jendela Toolbox**, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan user interface.
6. **Jendela Code**, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi Shift-F7.
7. **Jendela Properties**, merupakan daftar properti-properti object yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (foreground) dan warna latarbelakang (background). Anda dapat menggunakan F4 untuk menampilkan jendela properti.
8. **Jendela Color Palette**, adalah fasilitas cepat untuk mengubah warna suatu object.
9. **Jendela Form Layout**, akan menunjukan bagaimana form bersangkutan ditampilkan ketika runtime.
10. **Jendela Toolbox** merupakan jendela yang sangat penting bagi anda. Dari jendela ini anda dapat mengambil komponen-komponen (object) yang akan ditanamkan pada form untuk membentuk user interface.
11. Adapun secara garis besar fungsi dari masing-masing intrinsic kontrol tersebut adalah sebagai berikut :



Gambar 2.8 *Toolbox*

12. **Pointer** bukan merupakan suatu kontrol; gunakan icon ini ketika anda ingin memilih kontrol yang sudah berada pada form.
13. **PictureBox** adalah kontrol yang digunakan untuk menampilkan image dengan format: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, dan JPEG.
14. **Label** adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.
15. **TextBox** adalah kontrol yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.
16. **Frame** adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.
17. **CommandButton** merupakan kontrol hampir ditemukan pada setiap form, dan digunakan untuk membangkitkan event proses tertentu ketika pemakai melakukan klik padanya.
18. **CheckBox** digunakan untuk pilihan yang isinya bernilai yes/no, true/false.

19. **OptionButton** sering digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.
20. **ListBox** mengandung sejumlah item, dan user dapat memilih lebih dari satu (bergantung pada property MultiSelect).
21. **ComboBox** merupakan konbinasi dari TextBox dan suatu ListBox dimana pemasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.
22. **HScrollBar dan VScrollBar** digunakan untuk membentuk scrollbar berdiri sendiri.
23. **Timer** digunakan untuk proses background yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.
24. **DriveListBox, DirListBox, dan FileListBox** sering digunakan untuk membentuk dialog box yang berkaitan dengan file.
25. **Shape dan Line** digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.
26. **Image** berfungsi menyerupai image box, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol image menggunakan resource yang lebih kecil dibandingkan dengan PictureBox
27. **Data** digunakan untuk data binding
28. **OLE** dapat digunakan sebagai tempat bagi program eksternal seperti Microsoft Excel, Word, dll.

2.9.4 Jenis *File* Pada Visual Basic

Proyek merupakan cikal bakal dari aplikasi dan terdiri dari beberapa *file* yang saling terkait satu sama lain, *file-file* tersebut mempunyai ekstensi yang berbeda-beda sesuai dengan isi dan fungsinya. Sebuah *Proyek* dapat terdiri atas:

a. File .VBF untuk tiap proyek
b. File .FRM untuk tiap form aplikasi pada sebuah proyek.
c. File .VBX atau OCX untuk tiap custom pada sebuah proyek.
d. File .BAS untuk tiap modul standar pada proyek.
e. File .CLS untuk tiap modul class pada sebuah proyek.
f. File .FRX untuk tiap form pada sebuah proyek yang memiliki control berproperti menggunakan data *binary*.
g. File .RES yaitu sebuah *resource file* pada sebuah proyek.

Dalam hal ini tidak perlu membuat seluruh *file* di atas. Karena apa saja yang dibutuhkan tergantung dari proyek yang sedang dikerjakan.

2.9.5 Komponen Pada Visual Basic

Komponen-komponen yang sering dipakai diantaranya:

a. Modul *Form* (File berekstensi .FRM).

Merupakan kumpulan *form* yang berisi deskripsi secara grafis mengenai tampilan serta kontrol yang digunakan dalam sebuah proyek. Termasuk di dalamnya pengaturan properti yang digunakan oleh masing-masing kontrol. Disampimg itu rutin yang ditulis untuk masing-masing *obyek* pada *form* juga tersimpan pada *file* ini. Untuk aplikasi yang terdiri dari

banyak *form*, masing-masing *form* tersimpan dalam sebuah *file* berekstensi .FRM.

b. Modul *Class* (File Berekstensi .CLS)

Fungsinya sama dengan modul *form* tetapi terlihat dalam bentuk tampilan. Melalui modul ini bisa dicoba membuat sendiri obyek yang ingin ditampilkan. Masing-masing obyek yang dibuat harus mempunyai properti dan modul tersendiri.

c. Modul Standar (*File* berekstensi .BAS).

Modul ini berisi deklarasi tipe, konstanta, variabel eksternal *procedure* (prosedur yang menangani kejadian) dan *public procedure*. Biasanya *fil*e ini berisikan perhitungan atau logika yang kompleks.

d. *File Resource* (File berekstensi .RES).

File ini berisi *bitmap*, text string, atau data lainnya yang dapat diubah tanpa perlu mengedit kembali rutin. Biasanya *file* ini digunakan untuk mengubah bahasa pada user *interface* alokasi untuk pemakai berbahasa asing (selain yang digunakan oleh si pembuat aplikasi).

e. *OLE Custom Control* dan *Insertable Objects* (File berekstensi .OCX)

File ini berisi *custom* dan *Insertable Objects* yang ditambahkan pada proyek anda.

f. VBX *Custom Controls* (File berekstensi .VBX).

Sama dengan *file* berekstensi .OCX namun hanya digunakan pada Visual Basic 4.0 versi 16 *bit*.

2.10 Pendekatan Rekayasa Perangkat Lunak

2.10.1 Pengertian Rekayasa Perangkat Lunak

Pengertian rekayasa perangkat lunak menurut Pressman (1992) : Pembentukan dan penggunaan dari prinsip-prinsip rekayasa yang baik untuk mendapatkan perangkat lunak ekonomis yang handal dan bekerja efisien pada mesin nyata.

Rekayasa perangkat lunak adalah :

1. Sekumpulan aktifitas-aktifitas kerja yang berkaitan erat dengan perancangan dan implementasi produk-produk dan prosedur-prosedur yang dimaksudkan untuk merasionalisasikan produksi perangkat lunak berikut pengawasannya.
2. Pendekatan sistematik dalam pengembangan, operasi, pemeliharaan perangkat lunak.

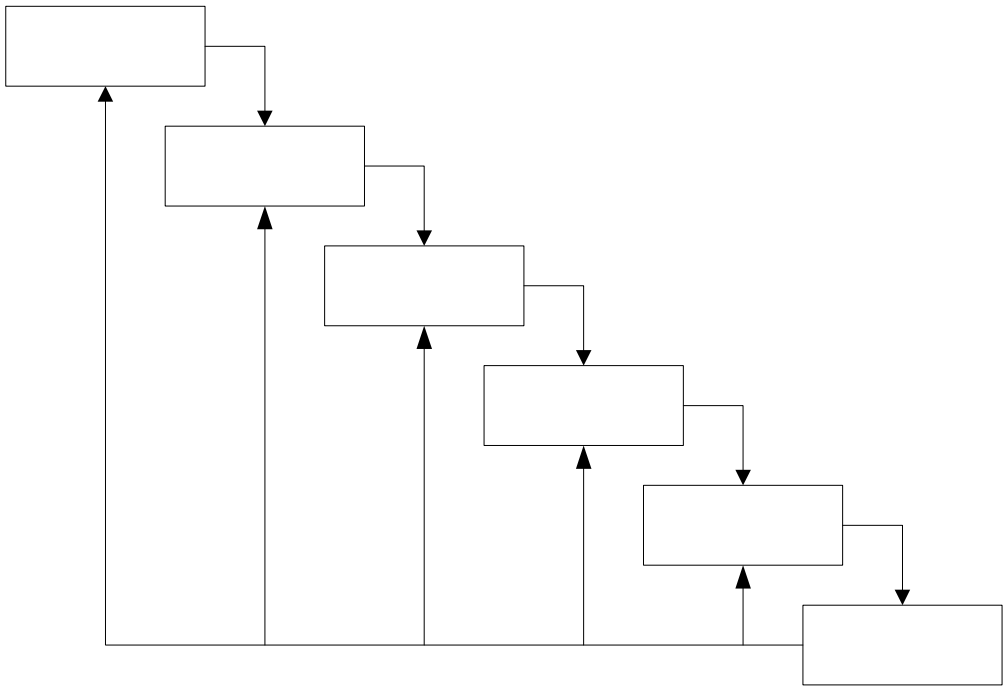
Rekayasa perangkat lunak merupakan teknologi yang terdiri dari beberapa elemen (Pressman, 1992) yaitu :

1. Proses : Bertindak sebagai perekat yang mempertahankan teknologi dan memungkinkan pengembangan perangkat lunak menjadi rasional dan tepat waktu. Selain itu, proses juga mendefinisikan kerangka yang harus didirikan agar penyerahan teknologi rekayasa perangkat lunak menjadi efektif.
2. Metode : Menyediakan teknik bagaimana membangun perangkat lunak, meliputi tugas-tugas yang mencakup analisis, perancangan, implementasi (pemrograman), pengujian, dan pemeliharaan.
3. Alat bantu (tools) : Alat bantu yang digunakan saling terintegrasi sehingga informasi yang dihasilkan oleh suatu alat bantu dapat digunakan alat bantu

lainnya, maka sistemnya sering disebut *Computer Aided Software Engineering* (CASE).

2.10.2 **Model Proses Waterfall** (Pressman 1992)

Model ini memerlukan pendekatan yang sistematis dan sekuensial di dalam pengembangan sistem perangkat lunaknya, pengembangannya dimulai dari analisis dan definisi kebutuhan, perancangan sistem dan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem, operasi dan perawatan. Model Proses *Waterfall* dapat dilihat dalam gambar 2.9.



Gambar 2.9 Model Proses *Waterfall*

Dengan demikian, pada model ini terdapat aktifitas-aktifitas sebagai berikut :

1. Rekayasa sistem.

Rekayasa Sistem

Analisis

Karena perangkat lunak merupakan bagian dari sistem yang lebih besar maka pengembangannya dimulai dari pengumpulan semua kebutuhan elemen-elemen sistem. Hal ini menjadi sangat penting karena perangkat lunak akan berkomunikasi dengan perangkat keras, data, manusia, dan bahkan dengan perangkat lunak lainnya.

2. Analisis.

   Pada tahap ini dilakukan pengumpulan kebutuhan elemen-elemen di tingkat perangkat lunak. (requirement analysis). Dengan analisis, harus dapat ditentukan domain-domain data atau informasi, fungsi, proses, atau prosedur yang diperlukan beserta unjuk kerjanya. Hasil akhir dari tahap ini adalah spesifikasi kebutuhan perangkat lunak.

3. Perancangan.

   Pada tahap perancangan, kebutuhan-kebutuhan atau spesifikasi perangkat lunak, yang dihasilkan pada tahap analisis, ditransformasikan ke dalam bentuk arsitektur perangkat lunak yang memiliki karateristik mudah dimengerti dan tidak sulit untuk diimplementasikan.

4. Pemrograman.

   Pada tahap ini dilakukan implementasi hasil rancangan ke dalam baris-baris kode program yang dapat dimengerti oleh mesin (komputer).

5. Pengujian.

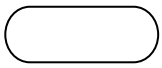
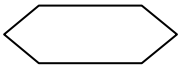
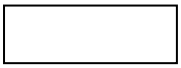
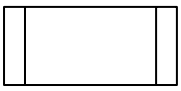
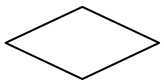
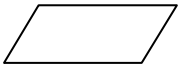
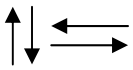
   Pada tahap ini pengujian dilakukan pada setiap modul. Jika setiap modul selesai diuji dan tidak bermasalah, modul-modul tersebut segera diintegrasikan (dan dikompilasi) hingga membentuk suatu perangkat lunak yang utuh.

6. Pengoperasian dan Pemeliharaan.

   Tahap ini ditandai oleh penyerahan perangkat lunak kepada pemesanannya yang kemudian dioperasikan oleh pemiliknya. Dalam masa operasional sehari-hari, suatu perangkat lunak mungkin saja mengalami kesalahan atau kegagalan dalam menjalankan fungsi-fungsinya. Dengan demikian, kedua faktor ini menyebabkan perlunya perangkat lunak dipelihara dari waktu ke waktu.

2.11 Flowchart

Flowchart adalah suatu alat yang cukup efisien untuk menggambarkan suatu algoritma agar dapat dengan mudah diimplementasikan pada suatu aplikasi. Tujuan utama penggunaan flowchart adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana. Metode ini menggunakan simbol-simbol tertentu yang mudah dimengerti dan telah distandarisasi dalam menyelesaikan masalah yang dihadapi. Adapun simbol-simbol program flowchart dapat dilihat pada gambar 2.10.

SIMBOL	KETERANGAN
	**Terminal Point Symbol.** digunakan untuk menunjukkan awal dan akhir dari suatu proses.
	**Preparation Symbol.** Simbol **Persiapan** digunakan untuk memberi nilai awal Suatu besaran atau *variabel* (harga awal).
	**Process Symbol.** Simbol **Proses** atau **Pengolahan** digunakan untuk mewakili suatu proses, seperti pengolahan aritmatika atau pemindahan data.
	**Predefined Process Symbol.** Simbol **Proses Terdefenisi** digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain atau untuk proses yang detilnya dijelaskan terpisah, misalnya berbentuk subroutine.
	**Decision Symbol.** Simbol **Keputusan** digunakan untuk mewakili operasi perbandingan logika atau suatu penyeleksian kondisi di dalam program.
	**Input/Output Symbol.** Simbol **Input/Output** digunakan untuk menyatakan dan mewakili data masukan atau keluaran.
	**Connector Symbol.** Simbol **Penghubung** digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama.
	**Off-page Connector.** Simbol **Penghubung Halaman lain** digunakan untuk menunjukkan hubungan arus proses yang terputus dengan sambungannya ada di halaman yang lain.
	**Flow Lines Symbol.** Simbol **Garis Alir** digunakan untuk menunjukkan aliran atau arus dari proses.

Gambar 2.10 Simbol-Simbol Program *Flowchart*