

BAB 2

LANDASAN TEORI

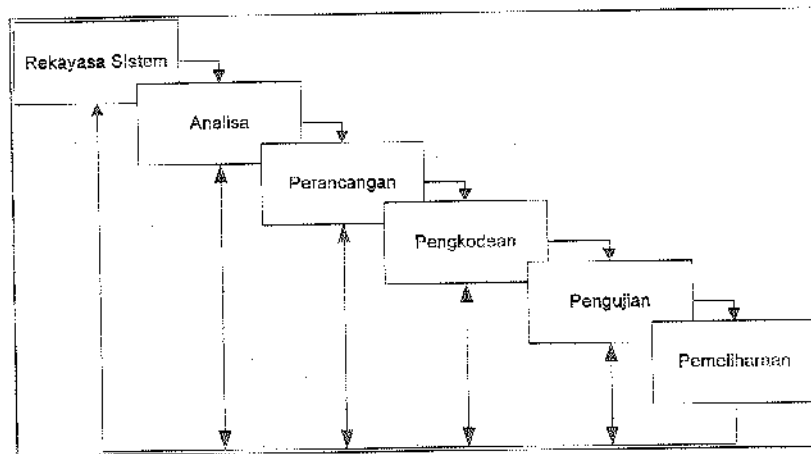
2.1 Teori Perancangan Piranti Lunak

Definisi awal dari rekayasa piranti lunak diajukan oleh Fritz Bauer pada konferensi besar pertama tentang rekayasa piranti lunak yaitu penetapan dan penggunaan prinsip – prinsip rekayasa dalam rangka mendapatkan software yang ekonomis, terpercaya dan bekerja efisien pada mesin atau komputer (Pressman, 1997, p 22).

Tujuan dari rekayasa piranti lunak adalah untuk menghasilkan produk piranti lunak yang akan diberikan kepada seorang pelanggan dengan dokumentasi yang menggambarkan bagaimana memasang dan menggunakan sistem.

Daur hidup rekayasa dibangun berdasarkan pendekatan sistematis dan berurutan. *System Development Life Cycle (SDLC)* adalah tahap operasi terstruktur yang dibutuhkan untuk menyusun, membangun dan membuat beroperasinya sebuah sistem informasi baru. SDLC dapat dipecah menjadi 4 tahap utama yaitu analisis, desain, implementasi dan perawatan.

Contoh dari penerapan SDLC adalah model linear sekuensial atau *Waterfall*. Model ini menuntut pendekatan sistematis, berurutan pada pengembangan piranti lunak. Paradigma ini terdiri dari aktivitas – aktivitas berikut ini :



Gambar 2.1 Model *Linear Sequential* (Pressman, 1997, p 31)

- Rekayasa Sistem

Karena piranti lunak adalah bagian yang besar dari suatu sistem, sehingga pekerjaan ini dimulai dengan pengusulan penetapan pada semua sistem elemen dan mengalokasikan beberapa bagiannya ke dalam usulan pada piranti lunak. Sedangkan rekayasa sistem dititikberatkan pada penggabungan semua level sistem dengan melakukan pengkajian dari level atas dalam perancangan dan analisa.

- Analisa

Pada analisa, khususnya analisa kebutuhan piranti lunak adalah suatu proses yang intensif dan memfokuskan pada spesialisasi piranti lunak. Untuk mengerti suatu kebiasaan pemrograman yang dibuat, perekayasa perlu mengerti tentang kinerja sistem dan *user interface*. Dengan demikian maka semua kebutuhan baik sistem maupun piranti lunak harus didokumentasikan dan harus dikaji ulang oleh pengguna.

- Perancangan

Pada perancangan piranti lunak, terdiri dari banyak tahap proses yang memfokuskan pada tiga hal, yaitu struktur data, arsitektur piranti lunak, dan prosedur secara detail. Pada proses perancangan, prinsipnya mengubah suatu kebutuhan menjadi piranti lunak yang layak dari segi kualitas sebelum dilakukan proses pengkodean. Seperti halnya analisa kebutuhan piranti lunak, perancangan juga diperlukan dokumentasi dan akan menjadi bagian dari konfigurasi piranti lunak.

- Pengkodean

Setelah proses perancangan, diperlukan perubahan ke dalam bentuk yang dapat dibaca oleh mesin, maka proses selanjutnya adalah proses pengkodean. Jika pada proses pendesainan dititikberatkan pada hal – hal yang lebih detail, maka pada proses pengkodean difokuskan pada hal yang bersifat mekanik.

- Pengetesan

Setelah proses pengkodean selesai, langkah selanjutnya adalah melakukan pengetesan. Pada proses pengetesan dititikberatkan pada logika internal suatu piranti lunak yang menggambarkan bahwa semua kode sudah dilakukan pengetesan. Di samping itu pengetesan dari fungsi eksternal juga perlu dilakukan, yaitu melakukan atau memberikan input, apakah sesuai dengan yang diinginkan atau belum.

- Pemeliharaan

Piranti lunak mungkin juga mengalami perubahan setelah dikirim pada pelanggan, dimana perubahan ini mungkin disebabkan karena :

- Ditemukannya kesalahan (error)
- Adaptasi lingkungan seperti perubahan sistem operasi atau perangkat lain.
- Menginginkan peningkatan fungsi atau kinerja.

Model *Linear sequential* ini merupakan paradigma yang paling tua dan paling luas digunakan. Namun banyak pula kritikan terhadap paradigma ini. Beberapa masalah jika menerapkan *Linear sequential* adalah:

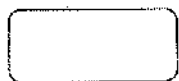
- Pada proyek sebenarnya jarang menerapkannya secara berurutan. Bahkan sering terjadi pengulangan (iterasi). Dan ini banyak menimbulkan masalah dalam menerapkan paradigma tersebut.
- Sulit menyebutkan semua kebutuhan bagi pelanggan secara eksplisit. *Linear sequential* membutuhkan kebutuhan tetapi sulit mengakomodasikan ketidakpastian yang muncul pada tahap awal proyek.
- Pelanggan harus menunggu lama untuk memperoleh versi program yang sudah siap pakai. Kesalahan bila tidak terdeteksi sampai tahap kajian ulang piranti lunak akan menjadi bencana besar.

Walaupun begitu *linear sequential* tetap penting karena membantu penentuan metode manakah yang digunakan untuk analisis, desain, pengkodean, pengujian, dan pemeliharaan dapat ditempatkan.

2.2 Flowchart

Bagan alir adalah bagian yang menunjukkan alir di dalam program atas prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan dokumentasi.

→ Aliran : Menghubungkan antara simbol dan menunjukkan aliran aktivitas.



Terminal : Menunjukkan awal atau akhir bagi aktivitas



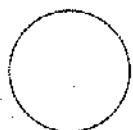
Input / Output : Menunjukkan operasi input (memasukkan data) atau output (mengeluarkan hasil).



Proses : Menunjukkan operasi aritmatik dan manipulasi data dilaksanakan.



Keputusan : Menunjukkan operasi logik atau perbandingan. Mempunyai 1 aliran masuk dan 2 aliran keluar (yaitu benar atau salah)



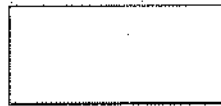
Penghubung Offpage : Menunjukkan flowchart bersambung

2.3 STD (*State Transition Diagram*)

STD adalah modul perilaku yang berdasarkan pada definisi kumpulan keadaan sistem, STD digunakan untuk menunjukkan suatu keadaan internal dari kelas tertentu (Pressman, 1997, p 217). STD menggambarkan kejadian yang menyebabkan suatu perubahan dari suatu keadaan menjadi keadaan lain dan tindakan yang dihasilkan dari suatu perubahan keadaan. Berikut ini adalah notasi yang digunakan dalam pembuatan suatu STD :

- Keadaan Sistem

Setiap kotak mewakili suatu keadaan dimana sistem mungkin berada di dalamnya. Umumnya keadaan sistem mungkin berupa menunggu user memasukkan password, menunggu perintah selanjutnya dan lain sebagainya.



Gambar 2.2 Notasi Keadaan (Notasi State)

- Perubahan Keadaan

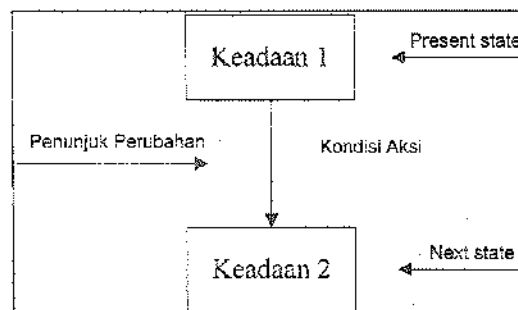
Untuk menghubungkan keadaan dengan keadaan lain, digunakan ini, jika sistem memiliki turunan dalam perilakunya, maka hanya suatu keadaan dapat berubah menjadi keadaan tertentu.



Gambar 2.3 Notasi Perubahan Keadaan

- Kondisi dan Aksi

Untuk melengkapi STD dibutuhkan hal tambahan : kondisi yang menyebabkan keadaan berubah. Dibawah ini ilustrasi dari kondisi dan aksi yang ditampilkan disebelah anak panah yang menghubungkan 2 keadaan.



Gambar 2.4 Kondisi dan Aksi

2.4 Webcam

Webcam adalah sebuah kamera video yang dipasangkan secara langsung pada sebuah komputer. *Webcam* ini dapat mengambil gambar bergerak maupun gambar diam, biasanya digunakan untuk mengawasi keadaan rumah, ataupun keadaan

lingkungan tertentu yang nantinya akan dapat dilihat pada komputer lainnya di tempat yang berbeda dan terhubung ke jaringan melalui halaman *web*, melalui *streaming* video ataupun melalui jaringan telepon. Penerapan *webcam* lainnya dapat juga sebagai pemantau keamanan suatu tempat, konferensi melalui video, informasi jalan dan lainnya.

(http://searchmobilecomputing.techtarget.com/gDefinition/0,294236,sid40_gci211738,00.html)

2.5 Telepon Selular

Telepon selular merupakan telepon yang tanpa kabel, atau lebih sering dikenal dengan istilah telepon genggam, dimana media komunikasinya tidak menggunakan kabel telepon konvensional, tetapi menggunakan jasa yang disediakan operator telekomunikasi telepon selular. Telepon selular digunakan pada daerah yang terbagi-bagi menjadi beberapa bagian (*section*) berdasarkan pada *short-range transmitter/receiver*-nya. Telepon selular merupakan suatu perangkat telepon yang bersifat *mobile*, *transportable*, *portable*, dan *personal*.

(www.cogsci.princeton.edu/cgi-bin/webwn)

2.6 Mikrokontroler (Sistem Minimum)

Mikrokontroler merupakan prosesor dengan fungsi minimum yang digunakan untuk mengontrol aplikasi tertentu atau khusus. Karena itulah, sistem dengan mikrokontroler sering dinamakan sistem minimum atau *dedicated sistem*.

Mikrokontroler adalah sebuah mikroprosesor yang dikombinasi dengan I/O dan memori (RAM/ROM) yang dikemas dalam satu level *chip*, yang menghasilkan single *chip* mikrokomputer (SCM) untuk membedakannya dengan mikrokomputer. Untuk selanjutnya SCM dapat disebut mikrokontroler.

Di antara keuntungan pemakaian mikrokontroler dibandingkan mikroprosesor adalah pada mikrokontroler sudah terdapat RAM dan peralatan I/O pendukung sehingga tidak perlu menambahkannya.

Perlengkapan dasar mikrokontroler untuk bekerja secara lancar dan sempurna, (<http://lgg-yuni.tripod.com/interest.htm>) yaitu :

1. CPU

Central Processing Unit terdiri dari dua bagian, yaitu Unit Pengendali (*Control Unit/CU*) serta Unit Aritmatika dan Logika (*Aritmatica and Logic Unit/ALU*). Fungsi utama unit pengendali adalah mengambil, mengkode dan melaksanakan urutan instruksi sebuah program yang tersimpan dalam memori.

2. Alamat

Bila menempatkan peralatan tambahan ke dalam mikrokontroler harus ditempatkan ke alamat (*address*) untuk menghindari kesalahan jika dua alat bekerja secara bersamaan. Alat penyimpan data adalah RAM atau ROM.

3. Data

CPU membutuhkan data-data yang dapat dibaca untuk dapat bekerja. Data-data ini berupa program sesuai dengan bahasa mesin CPU tersebut yang kalau dijabarkan adalah susunan logika-logika biner yang disimpan pada

alamat tertentu yang juga berupa susunan logika-logika biner pula. Data disimpan pada RAM atau ROM.

4. Pengendali

Pengendali atau *Control Bus* berguna untuk menyerempakkan operasi mikrokontroler dengan operasi rangkaian luar.

5. Memori

Memori untuk menyimpan program atau data. Pada mikrokontroler tempat menyimpan program atau data adalah ROM/EPROM. Sedangkan pada komputer pribadi disimpan pada *hard-disk* atau disket.

2.7 AT Command

AT Command adalah suatu metode standar yang dapat digunakan oleh komputer untuk berkomunikasi dengan modem dan peralatan komunikasi lainnya, misalnya : telepon selular. Tidak semua *AT Command* akan dijelaskan agar lebih memfokuskan pembahasan. Beberapa *AT Command* yang berkaitan dengan aplikasi sistem ini adalah (Khang, 2002, p 8) :

- ATE : untuk mengetahui apakah port dalam keadaan terbuka atau tertutup.
- AT+CMGR : untuk membaca isi SMS
- AT+ CMGS : untuk mengirim sebuah SMS
- AT+ CMGL : untuk memeriksa SMS
- AT+ CMGD : untuk menghapus SMS
- ATD : untuk melakukan pemanggilan (*dial*) ke suatu nomor
- ATDT : nada dering

AT Command yang lainnya dapat dilihat pada lampiran yang ada.

2.8 PDU (*Protocol Data Unit*)

PDU (*Protocol Data Unit*) merupakan bentuk data yang mengalir ke dan keluar dari SMS Center, berisi bilangan – bilangan heksadesimal (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) (Khang, pp 8-14). PDU terdiri atas beberapa *header*, dimana *header - header* untuk data yang keluar (SMS yang diterima) dari SMS Center berbeda dengan *header- header* yang masuk (SMS yang dikirimkan) ke SMS Center.

Header PDU untuk mengirimkan SMS ke SMS Center terdiri atas 8 macam, yaitu :

a. Nomor SMS Center

Terbagi menjadi 3 *subheader*, yaitu :

- Jumlah pasangan nomor SMS Center dalam bentuk bilangan heksadesimal
- Kode nasional (81) atau internasional (91)
- Nomor SMS Center-nya, dalam bentuk pasangan – pasangan yang dibalik – balik. Jika ada bilangan yang tidak ada pasangannya, maka bilangan tersebut dipasangkan dengan huruf F.

Lalu ketiga *subheader* tersebut digabungkan.

b. Tipe SMS, untuk mengirimkan SMS tipenya adalah 1, jadi bilangan heksadesimalnya adalah 01.

c. Nomor referensi SMS

Nomor referensi ini dibiarkan 0 (jadi heksadesimalnya = 00) sebab nomor ini akan diberikan secara otomatis oleh alat SMS-gatewaynya.

- d. Nomor telepon selular penerima, sama seperti header nomor SMS Center, terdiri dari 3 *subheader* dengan format penulisan yang sama, hanya berbeda pada nomor SMS Center diganti dengan nomor telepon selular yang dituju.
- e. Bentuk SMS, terdiri dari beberapa macam antara lain :
 - 0, heksadesimalnya adalah 00 → dikirim sebagai SMS
 - 1, heksadesimalnya adalah 01 → dikirim sebagai telex
 - 2, heksadesimalnya adalah 02 → dikirim sebagai fax
- f. Skema encoding data I/O, ada 2 macam, yaitu :
 - Skema 7 *bit*. Ditandai dengan bilangan 0, jadi heksadesimalnya 00. Umumnya SMS Gateway yang ada menggunakan skema 7 *bit*.
 - Skema 8 *bit*. Ditandai dengan bilangan yang lebih besar dari 0, lalu diubah ke bentuk heksadesimal.
- g. Jangka waktu validitas SMS (sebelum *expired*)

Sebaiknya jangka waktu ini dikosongkan, berarti tidak ada pembatasan jangka waktu *expired*-nya SMS, sehingga SMS yang dikirimkan pasti sampai ke nomor yang dituju.

Jika ingin memberikan pembatasan waktu, maka *header* ini diisi dengan bilangan *integer* tertentu yang kemudian akan diubah menjadi bilangan heksa. Rumus untuk menghitung jangka waktu validitas SMS dapat dilihat pada tabel berikut :

Tabel 2.1 Tabel Validitas SMS

Integer (INT)	Jangka waktu validitas SMS
0-143	$(INT+1) * 5 \text{ menit}$ (berarti : 5 menit s/d 12 jam)
144 – 167	12 jam + $((INT - 143) * 30 \text{ menit})$
168 – 196	$(INT - 166) * 1 \text{ hari}$
197 – 255	$(INT - 192) * 1 \text{ minggu}$

h. Isi SMS, terbagi menjadi 2 *subheader*, yaitu :

- Panjang isi (jumlah huruf dari isi SMS) dalam bilangan heksa.
- Isi yang diubah ke bentuk bilangan heksa. Langkah untuk mengubah isi SMS adalah mengubahnya menjadi kode 7 *bit* dahulu baru setelah itu mengubahnya menjadi bentuk 8 *bit* yang diwakili dengan bentuk pasangan heksa.

Contoh dari pelaksanaan semua langkah tersebut di atas dapat dilihat pada lampiran.

2.9 Konsep Dasar Komunikasi Data

Umumnya, manusia mengartikan komunikasi hanya sebagai tindakan saling berbicara antara dua pihak. Pada dasarnya, komunikasi merupakan suatu proses penyampaian informasi dari satu pihak ke pihak lain, dari suatu tempat ke tempat lain, dengan menggunakan suatu media transmisi tertentu. Informasi yang dimaksud

bisa berupa teks maupun gambar. Komunikasi tidak mungkin bisa terjadi jika hanya ada satu pihak, paling sedikit harus ada dua pihak baru dapat dikatakan komunikasi

Sedangkan data adalah kenyataan atau fakta penting yang sesuai dengan kenyataan atau kejadian yang ada, yang tercatat atau terekam. Dimana catatan atau rekaman ini dapat menimbulkan arti yang bermacam-macam bagi manusia ataupun komputer setelah diproses atau diinterpretasikan.

Jadi komunikasi data merupakan suatu proses untuk menyampaikan data atau fakta yang tersedia dari satu pihak kepada pihak lain (dari seorang manusia ke manusia lain ataupun dari satu komputer ke komputer lainnya) yang meminta atau mungkin membutuhkannya untuk kemudian diolah atau diinterpretasikan, dengan melalui suatu media transmisi tertentu.

Secara khusus, komunikasi data merupakan satu bagian dari ilmu komunikasi yang mengkhususkan diri pada penyampaian informasi yang berupa teks dan gambar.

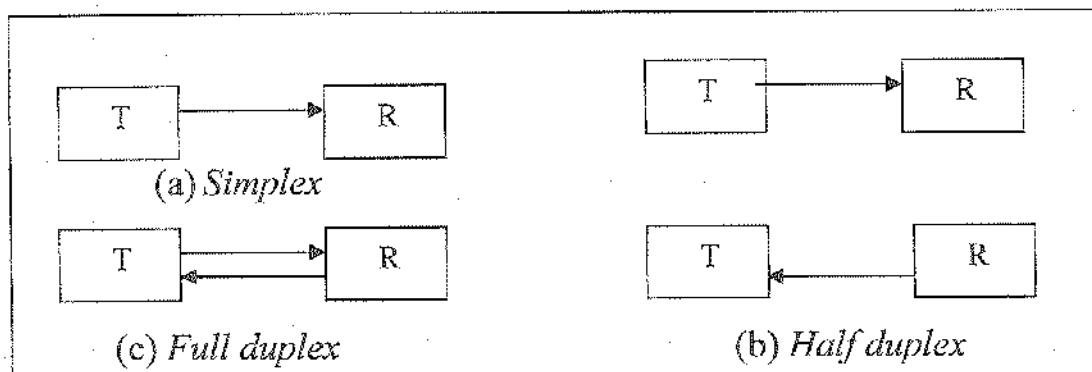
Unsur pokok komunikasi data yaitu :

- sumber (*transmitter*),
- media transmisi,
- tujuan (*receiver*),
- informasi yang berupa data

2.9.1 Arah Aliran Data

Pertukaran data pada sistem komunikasi data, dapat dibedakan menjadi 3 macam berdasarkan arah aliran datanya (Stallings, 1997, p 145), yaitu :

- a. *Simplex* : aliran data yang bersifat 1 arah, hanya dari pengirim atau pemancar ke penerima.
- b. *Half duplex* : aliran data yang bersifat 2 arah secara bergantian, artinya ada 2 buah terminal yang dapat berkomunikasi (memancarkan dan menerima) tapi tidak bisa bersamaan, hanya 1 terminal saja yang dapat mentransmisikan data pada satu waktu.
- c. *Full duplex* : aliran data yang bersifat 2 arah dan bisa dilakukan dengan bersamaan (kedua terminal secara simultan dapat mengirim dan menerima data).



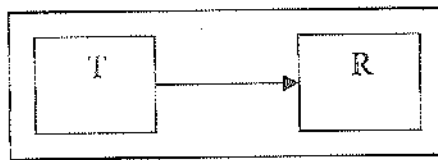
Keterangan : T = Transmitter ; R = Reciever

Gambar 2.5 Arah Aliran Data (Lukas, 1999, p 27)

2.9.2 Mode Transmisi

Ada dua jenis transmisi yang terdapat pada sistem komunikasi yaitu : transmisi paralel dan transmisi serial.

antara pengirim dan penerima, agar penerima dapat mengartikan data dengan benar (Lukas, 1999, p 28).



Gambar 2.7. Transmisi Serial

Pengiriman data secara serial dapat dibedakan menjadi 2 macam, yaitu modus transmisi asinkron dan modus transmisi sinkron

a. Modus Transmisi Asinkron

Modus ini sering disebut juga dengan *Start Stop Transmission*. Modus transmisi ini disebut asinkron karena tidak terdapat sinyal sinkronisasi dalam proses transmisi ini. Pada modus transmisi ini, pengirim (*transmitter*) dapat mengirimkan data kapan saja, sehingga penerima (*receiver*) harus selalu melakukan pemeriksaan apakah ada data yang dikirimkan untuknya dalam saluran transmisi atau tidak. Dalam pemeriksaan ini, digunakan 2 buah *bit* biner untuk dapat mengenali apakah ada data yang dikirim atau tidak, yaitu : *start bit* (selalu berlogika 0) dan *stop bit* (selalu berlogika 1).

Pengiriman data dilakukan dengan mentransmisikan 1 karakter dalam 1 waktu. Dimana setiap karakter panjangnya 8 *bit*. Untuk mentransmisikan suatu karakter, maka awal dari suatu karakter akan ditandai dengan *start bit* dan akhir dari suatu karakter ditandai dengan *stop bit*. Jadi untuk mentransmisikan sebuah karakter 8 *bit*, akan dikirimkan dalam sebuah paket sebesar 10 *bit*, yang terdiri dari 1 *start*

bit, 8 *bit* data, dan 1 *bit stop bit*. Satu-satunya proses sinkronisasi yang terjadi adalah ketika penerima menterjemahkan awal dan akhir karakter.

Ketika tidak ada karakter yang dikirimkan, maka hubungan antara pengirim dan penerima dikatakan dalam keadaan diam (*idle*), dan ditandai dengan pemberian elemen signal bernilai biner 1. Pengirim akan selalu mengirimkan sinyal ini, sampai pengirim siap untuk mengirimkan suatu karakter yang baru.

Keuntungan modus asinkron adalah dari segi teknis modus ini lebih mudah. Namun, modus ini memiliki beberapa kelemahan (Lukas, 1999, p 29), yaitu :

- Tidak efisien : karena waktu pengiriman tidak teratur sehingga media transmisi tidak dipakai secara terus menerus.
- Lambatnya pengiriman data : Setiap karakter yang akan dikirim selalu ditambah dengan *start* dan *stop bit*. Untuk mode ini dianjurkan maksimum kecepatan transmisi adalah 2400 bps (karena perkembangan teknologi kecepatan 9600 bps seringkali cukup baik tergantung media yang dipakai).

b. Modus Transmisi Sinkron

Pada modus transmisi sinkron, tidak ada *start bit* dan *stop bit*. Dan pengiriman data pun dilakukan dengan mengirimkan blok per blok, bukan per karakter. Dimana satu blok data dapat berisi antara

256 karakter sampai 2048 karakter, yang antar karakternya tidak memiliki pemisah. Sehingga membutuhkan sinkronisasi yang sama antara pengirim dan penerima agar data yang diterima dapat diartikan dengan benar.

Untuk proses sinkronisasi pada modus ini, digunakan *clock*. Terdapat 2 metode dalam transmisi sinkron, dimana metode-metode ini dapat digunakan secara terpisah ataupun digabungkan. Pada metode pertama, untuk menandakan awal transmisi data, pengirim akan mengirimkan karakter unik yang disebut *syn character*. Setelah pihak penerima menerima karakter ini, *clock* penerima dan *clock* pengirim akan disamakan.

Sedangkan pada metode kedua. Digunakan *transmit clock* dan *receive clock*. Sehingga dibutuhkan sebuah saluran transmisi berupa kabel yang terpisah, yang digunakan untuk mengirimkan timing dari transmisi data yang dikirimkan.

Keuntungan yang terdapat pada modus sinkron (Lukas, 1999, p 30) :

- Menaikkan efisiensi pengiriman untuk data yang banyak, alasannya :
 1. tidak ada *start* dan *stop bit* untuk setiap karakter, karena pengiriman data dilakukan per paket
 2. tidak ada selang waktu antara karakter satu dengan lainnya
- Kecepatan transmisi tinggi

Kelemahan yang terdapat pada modus sinkron (Lukas, 1999, p 30) :

- Dari segi teknis lebih sukar yang berarti lebih mahal
- Bila pada waktu pengiriman terdapat data yang salah, maka seluruh data (1 blok) akan dikirimkan ulang
- Memerlukan saluran transmisi yang berkualitas baik, agar mengurangi pengiriman kembali data

2.9.3 Protokol Komunikasi Data

Di dalam jaringan komputer saat ini, perangkat lunak jaringan yang ada semakin terstruktur. Untuk dapat mengurangi kerumitan rancangannya, sebagian besar jaringan diorganisasikan ke dalam bentuk tumpukan lapisan (layer-layer), dimana setiap layer yang berada di bawahnya memberikan layanan bagi layer yang berada di atasnya.

Alasan dibuatnya konsep layering adalah:

1. menyediakan dasar baku yang sama dalam koordinasi jaringan komunikasi komputer
2. menyediakan referensi dan pengertian yang sama pada konsep komunikasi data
3. mengurangi kesalahan interpretasi atau pengertian

Layanan yang diberikan layer dibawahnya, bisa dibedakan menjadi 2 macam (Tanenbaum, 1997, p 17), yaitu :

1. *Connection Oriented*, yaitu layanan yang mirip dengan sistem telepon. Pertama-tama pengguna layanan harus membuat koneksi, lalu menggunakan koneksi tersebut, dan memberhentikan koneksi tersebut.
2. *Connectionless Service*, yaitu layanan yang mirip dengan sistem surat pos. Dimana setiap pesan memiliki alamat yang dituju, dan memiliki route masing-masing.

Layanan dikatakan berkualitas baik jika tidak ada kehilangan data. Kualitas ini dapat ditingkatkan dengan pemakaian *acknowledgement* (pemberitahuan) dari penerima kepada pengirim bahwa pesan sudah diterima. Dan agar antar suatu layer pada suatu sistem dapat berkomunikasi dengan layer pada sistem lain, diperlukan adanya protokol yang berperan sebagai prosedur yang mengatur bagaimana komunikasi tersebut harus dilakukan. Protokol dapat didefinisikan sebagai persetujuan semua pihak mengenai tata cara yang mengatur komunikasi data antar sistem, yang harus dipatuhi oleh sistem yang hendak berkomunikasi, dan harus mempunyai kehandalan yang tinggi.

Protokol memiliki beberapa elemen penting (Lukas, 1999, p 83), yaitu :

1. *syntax* : meliputi format data, sinyal dan level yang masuk
2. *semantics* : meliputi pengendalian arus informasi dan menangani kesalahan
3. *timing* : meliputi kecepatan dan urutan arus informasi yang datang

2.9.3.1 Protokol Standar Jaringan Komunikasi

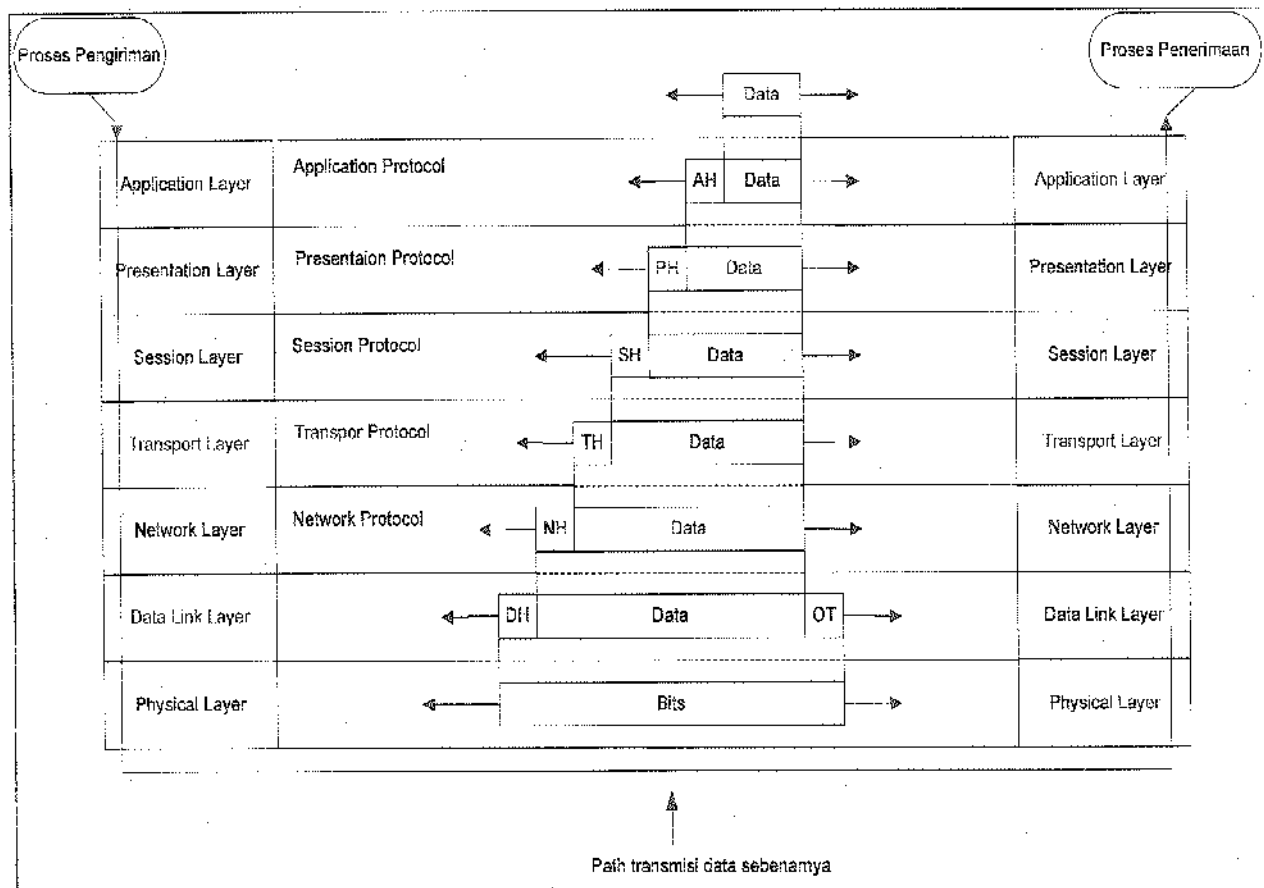
Banyak sekali protokol yang memiliki banyak perbedaan, misalnya perbedaan dalam karakter sinkronisasi, jumlah karakter sinkronisasi, dan lainnya. Hal ini tentunya bisa mengakibatkan kekacauan dalam komunikasi jaringan, tiap protokol tidak dapat berkomunikasi dengan protokol lainnya karena perbedaan tersebut. Untuk itu perlu adanya usaha untuk menyamakan persepsi protokol antar komputer, yang disebut standarisasi. Standarisasi dapat dilakukan dengan menggabungkan beberapa perusahaan atau organisasi, bahkan secara internasional, agar hasilnya dapat diterima oleh semua pihak.

International Standard Organization (ISO) merupakan suatu badan internasional yang membuat standar untuk berbagai hal, termasuk protokol komunikasi. Pada tahun 1977, ISO membentuk sebuah sub komite untuk mengembangkan suatu standar bagi dunia komputer dan komunikasi, yang sebelumnya menggunakan standar komunikasi yang berdasarkan persetujuan saja dan belum ada standar yang jelas. Hasilnya adalah model *Open Sistem Interconnection (OSI)* yang menjanjikan sistem berlapis (*layering*) yang dapat diatur dan memiliki fungsi tertentu yang spesifik dalam komunikasi data.

Dalam model OSI, protokol komunikasi data dibagi menjadi 7 layer (Stallings, 1997, pp 517-520). Dimana setiap layer memiliki fungsinya masing-masing untuk berkomunikasi dengan sistem lain. Operasi dari suatu layer menentukan layer yang lain. Layer yang berada di bawah mendukung layer yang di atasnya. Tingkat yang lebih bawah memiliki fungsi yang lebih

primitif. Tingkat yang lebih tinggi memiliki fungsi yang bermanfaat bagi pengguna

dengan memanfaatkan operasi dari layer di bawahnya.



Gambar 2.8. OSI Layer (Tanenbaum, 1997, p 25)

1. Physical Layer

Layer ini bertugas untuk menangani transmisi dari *bit-bit* yang tidak terstruktur antara media fisik, menangani antarmuka fisik antara peralatan (*devices*) dan peraturan tentang *bit* yang mana yang dikirimkan antara satu dengan lainnya.

2. *Data link layer*

Data link layer berfungsi untuk membuat *physical link* terjamin saat mengaktifkan, mengelola, dan men-nonaktifkan *link*. Layanan yang disediakan oleh *data link layer* ini bagi layer-layer yang ada di atasnya adalah deteksi kesalahan (*error detection*) dan pengendalian (*control*). Layer ini mengatur pengiriman data agar kesalahan yang terjadi sekecil mungkin. Oleh karena itu, dengan sebuah protokol *data link layer* yang sepenuhnya berfungsi, level berikutnya bisa menganggap transmisi yang bebas dari kesalahan.

Pada layer ini juga mengatur *flow control* yang berfungsi sebagai:

- a. Cara untuk mengatur aliran data agar *transmitter* tidak membebani *receiver* dengan data yang dikirim berlebihan
- b. *Receiver* biasanya dilengkapi dengan *huffer*
- c. *Receiver* harus melakukan proses lanjutan terhadap data yang diterima sebelum ia dapat menampung data berikutnya

3. *Network layer*

Layer ini bertanggungjawab untuk membuat, mengelola, dan memberhentikan koneksi atau hubungan. Menyediakan transfer informasi antar sistem akhir melalui suatu jaringan komunikasi. Dengan adanya layer ini, layer-layer yang lebih tinggi tidak perlu tahu tentang transmisi data dan teknologi *switching* yang digunakan untuk menghubungkan sistem.

Pada layer ini, sistem komputer berdialog dengan jaringan untuk menspesifikasikan alamat tujuan dan untuk meminta suatu fasilitas jaringan, misalnya prioritas. Layer ini mengatur rute aliran data dalam jaringan komunikasi komputer, sinkronisasi paket data.

4. *Transport layer*

Menyediakan mekanisme untuk pertukaran data dan mengatur perpindahan data dari satu sistem ke sistem lain. Layanan transport yang berbasis koneksi memastikan bahwa data terkirim dengan bebas *error* atau tanpa *error*, sesuai dengan urutannya, dan tidak ada data yang hilang ataupun terduplikasi. Layer ini juga bertugas untuk mengoptimalkan penggunaan layanan jaringan dan menyediakan permintaan layanan yang berkualitas bagi *session layer*. Contohnya, *session layer* dapat menentukan tingkat kesalahan, penundaan maksimum, prioritas dan keamanan yang dapat diterima.

5. *Session layer*

Session layer berfungsi untuk pengendalian komunikasi antar aplikasi seperti, membuat, mengelola transfer data, dan memberhentikan koneksi (*session*) antara aplikasi yang saling berkomunikasi. Layer ini mengatur koordinasi dialog dan sinkronisasi.

Session layer menyediakan mekanisme untuk mengendalikan dialog antar aplikasi dalam sistem akhir. Layanan yang diberikan *session layer* adalah :

a. *Dialogue discipline*

Bisa dua arah secara simultan (*Full duplex*) atau dua arah bergantian (*Half duplex*)

b. *Grouping*

Aliran data dapat ditandai untuk mendefinisikan group dari data.

c. *Recovery*

Session layer menyediakan mekanisme *checkpointing*, sehingga jika ada kegagalan antar *checkpoint*, layer ini dapat mengirimkan ulang semua data setelah *checkpoint* terakhir.

6. *Presentation layer*

Menghasilkan kebebasan atau ketidaktergantungan proses aplikasi terhadap perbedaan representasi data (sintaks dan *semantics*). Mendefinisikan format dari data yang akan dipertukarkan antara aplikasi dan memberikan layanan untuk transformasi data kepada program aplikasi. Layer ini memastikan agar tidak terjadi kesalahan interpretasi data.

Salah satu layanan dari layer ini adalah *encoding data*. *Presentation layer* mengatur struktur data yang ditransmisikan dan mengkonversikannya dari representasi yang digunakan pada suatu

komputer menjadi representasi standar jaringan, dan sebaliknya.
(Tanenbaum, 1997, p 24)

7. *Application layer*

Pada layer inilah terdapat aplikasi program dari masing-masing pengguna. Layer ini menyediakan sejumlah fungsi bagi program aplikasi untuk mengakses lingkungan OSI dan juga menghasilkan layanan informasi yang terdistribusi. Layer ini memiliki fungsi-fungsi manajemen dan berguna untuk mekanisme yang mendukung aplikasi yang terdistribusi. Sebagai tambahan, aplikasi umum seperti transfer file, email, akses terminal ke komputer di lain tempat juga terdapat di layer ini.

2.9.3.2 *Error Correction*

Salah satu metode untuk mendeteksi kesalahan pengiriman data adalah *Automatic Repeat Request (ARQ)*. Bila saat terjadi komunikasi data antara pengirim dan penerima, maka penerima akan memeriksa data yang dikirimkan benar atau tidak. Jika ada kesalahan, maka penerima akan meminta pengirim untuk kirim lagi data yang hilang atau salah itu.

Dalam metode ARQ, terdapat beberapa teknik pengontrolan kesalahan (Stallings, 1997, pp 171-175), yaitu :

2. *Error detection* : Terdapat 3 jenis pengecekan *parity* yaitu :

a. *Vertical Redudancy Checking*

Sering dikenal sebagai *parity check*, jenisnya yaitu :

i. *Even Parity*

Menggunakan jumlah *bit* yang berbobot logik "1"

ii. *Odd Parity*

Menggunakan jumlah *bit* yang berbobot logik "1"

b. *Horizontal Redundancy Checking*

Merupakan penggunaan *vertical redundancy checking* secara vertikal dan horizontal.

c. *Cyclic Redundancy Checking*

Teknik CRC menggunakan prinsip pembagian *bit* data dengan suatu fungsi matematik yang disebut polinomial. Dan hasil baginya dikirim ke penerima untuk dicocokkan dengan cara yang sama.

3. *Positive Acknowledgment* : penerima kirim sinyal kembali pada pengirim bahwa data yang didapat adalah bebas dari kesalahan.
4. *Negative Acknowledgment and retransmission* : penerima mengembalikan sinyal kepada pengirim bahwa data ada kesalahan.
5. *Retransmission after timeout* : pengirim kirim ulang data sebelumnya setelah beberapa waktu tidak dapat balasan dari penerima.

Terdapat tiga jenis ARQ, yaitu :

1. *Stop and Wait ARQ*

Dalam melakukan transmisi data, setelah mengirimkan suatu *frame* data, maka pengirim akan menunggu balasan dari penerima sebelum mengirimkan *frame* berikutnya. Balasan dari penerima bisa berupa ACK (*acknowledge*) atau NACK (*not acknowledge*). Setelah menerima *frame* data, penerima akan melakukan pengecekan apakah ada kesalahan atau tidak pada *frame* data. ACK dikirimkan jika ternyata tidak ada kesalahan pada *frame* data, tapi NACK akan dikirimkan jika ada kesalahan dalam *frame* data dan pengirim akan mengirimkan ulang *frame* data tersebut.

2. Go - Back - NARQ

Data dikirimkan beberapa *frame* sekaligus menjadi satu kesatuan paket, sesuai dengan ukuran *window*-nya. Dan pengirim akan menunggu balasan dari penerima sebelum mengirimkan *window* selanjutnya. Namun jika ternyata ada kesalahan pada salah satu *frame* dalam *window* yang dikirimkan tersebut maka *frame* berikutnya akan diabaikan, dan penerima akan mengirimkan NACK. Lalu pengirim akan mengirimkan kembali *frame* tersebut dan *frame* selanjutnya sesuai ukuran *window*, meskipun ternyata *frame* berikutnya tersebut sudah dikirimkan pada *window* sebelumnya.

Untuk lebih jelasnya, misalkan ukuran *window*-nya adalah sebesar 3 *frame*. Pengirim akan mengirimkan *window* pertama, yang berisi *frame* 1, 2, dan 3. Tapi ternyata penerima mengirimkan NACK untuk *frame* 2. Maka pengirim akan mengirimkan *window* baru yang

berisi *frame* 2, 3, dan 4, meskipun *frame* 3 sudah dikirimkan pada *window* awal.

3. *Selective - Reject ARQ*

Metode ini mirip dengan metode Go-Back-N, perbedaannya adalah jika terjadi kesalahan pada salah satu *frame* dalam *window*, maka yang dikirimkan ulang hanya *frame* tersebut dan *frame* berikutnya yang merupakan kelanjutan dari *window* sebelumnya.

Misalkan ukuran *window* sebesar 3 *frame*. Pengirim mengirimkan *frame* 1, 2, dan 3. Tapi kesalahan ada pada *frame* 2, maka pengirim akan mengirimkan *window* baru yang berisi *frame* 2, 4, dan 5.

2.10 *Remote Access*

Pengertian dari *Remote Access* adalah kemampuan untuk mengakses jaringan dari lokasi jarak jauh. Secara umum, hal ini berarti penggunaan sebuah komputer, modem dan beberapa perangkat lunak *remote access* untuk mengakses jaringan. Sementara *remote access* berarti mengendalikan komputer lainnya, *remote access* juga berarti bahwa komputer yang berada di jarak jauh sebenarnya berperan sebagai *host* dalam jaringan. Perangkat lunak *remote access* menghubungkan *host* jaringan secara langsung. Satu-satunya perbedaan antara komputer *host* dengan komputer klien yang terhubung langsung pada jaringan adalah kecepatan transfer data yang lebih rendah.

2.11 Standar Komunikasi Serial

Standar komunikasi serial yang paling terkenal di dunia komputer dan industri adalah RS-232 dan variannya. Standar ini dikeluarkan oleh EIA (*Electronic Industries Association*) yang berpusat di Amerika. Standar RS-232 ditetapkan dengan metode *single ended* dengan transmisi tak berimbang (*unbalanced transmission*). Pada penggunaannya standar ini menjadi salah satu cara komunikasi serial *point to point* yang paling populer. Namun kelemahan dari metode ini adalah dalam hal kecepatan (maksimum 20 Kbps) dan jarak transmisi (maksimum 50 feet dengan kecepatan 20 Kbps).

Mengatasi kelemahan-kelemahan itu timbul perbaikan pada standar ini yang kemudian melahirkan varian-varian yang mempunyai kemampuan lebih dari standar RS-232. Salah satunya adalah *differential data transmission* (*balance differential signal*) yang menawarkan keunggulan dalam banyak aplikasi.

RS-232 merupakan standar komunikasi serial *single ended* yang dikeluarkan oleh EIA sekitar tahun 1962. Sinyal RS-232 menggunakan transmisi tak berimbang (*unbalanced transmission*) yang mempunyai karakteristik sebagai berikut, untuk tegangan diatas +3 volt maka diterjemahkan sebagai logic '0' (*low*) sedangkan untuk tegangan yang lebih kecil dari -3 volt diterjemahkan sebagai logic '1' (*high*). Umumnya tegangan yang dipakai oleh komputer pada port serial adalah +12 volt (*low*) dan -12 volt (*high*). RS-232 pada mulanya mempunyai kemampuan efektif pada sinyal rate (kecepatan transfer) maksimum 20 Kbps dan jarak media transmisi maksimum 15 meter. Pada pengembangannya dikeluarkanlah versi lain dari RS-232 yaitu RS-232-C yang paling populer pada tahun 1969 dan pada tahun 1987

dikeluarkan versi ke empat, yang dikenal dengan EIA-232-D yang masih kompatibel dengan RS-232-C dengan kemampuan yang lebih baik. Sebuah standar *single ended* yang lain adalah RS-449 yang merupakan perkembangan dari RS-232-C dengan kemampuan operasi (kemampuan transfer data maksimum 10MBps).