

## BAB II

### DASAR TEORI

#### II.1. Konsep Dasar Client Server

##### II.1.1. Struktur Sistem C/S

Dalam struktur C/S komputer yang berada di atas komputer lain disebut server dan komputer-komputer pada level di bawahnya disebut client. Client meminta (request) layanan yang disediakan oleh server. Pada kenyataannya ada beberapa jenis implementasi struktur C/S<sup>1</sup>

##### 1. File Server

Client meminta record tertentu dari file, dan server mengirimkan record-record ini ke client melalui jaringan

##### 2. Database Server

Client mengirim SQL (structure Query Language) ke server melalui jaringan. Server melakukan proses, mendapatkan informasi, dan kemudian mengirimkan hasil ke client

##### 3. Transaction Server

Client mengirim request yang meminta remote procedure di server. Remote procedure ini berupa satu set SQL statement (atau bisa juga suatu fungsi). Transaksi terjadi saat hasil permintaan dikerjakan oleh remote procedure dan kemudian hasilnya dikirimkan kembali ke client.

##### 4. Groupware Server

---

<sup>1</sup> [lecturer.ukdw.ac.id/othie/bhn](http://lecturer.ukdw.ac.id/othie/bhn) 10.pdf, halaman 1.

Server menyediakan berbagai aplikasi yang memungkinkan komunikasi antar client (dan pengguna yang memakainya) dengan menggunakan teks, image, bulletin boards, video, dan cara lain.

### II.1.2. Komponen Software Pada Sistem C/S

Software untuk sistem C/S mempunyai beberapa komponen / sub sistem yang masing-masing memiliki peran yang berada dan bekerja sama satu sama lain. Sub-sub sistem ini dapat dialokasikan di client atau server atau di server, atau didistribusikan pada keduanya. Komponen/Sub sistem tersebut adalah<sup>2</sup> :

#### 1. User interaction subsystem

Subsistem yang terdiri dari semua fungsi yang berhubungan dengan user interface. Subsistem yang berfungsi sebagai data presentation bagi pengguna.

#### 2. Application subsystem

Subsistem ini mengimplementasikan requirement yang telah ditetapkan sesuai dengan konteks domainnya. Misalnya aplikasi bisnis menghasilkan laporan berdasarkan input numerik, kalkulasi dan informasi database. Aplikasi groupware menyediakan fasilitas bulletin boards, email atau calendar. Software aplikasi bisa dibagi sehingga beberapa komponen berada di client dan yang lain di server. Application subsystem juga sering disebut business logic.

#### 3. Database management subsystem

Subsistem yang melakukan manipulasi dan manajemen data yang dibutuhkan oleh aplikasi. Manipulasi data berarti seperti kirim record, atau proses transaksi SQL yang rumit.

---

<sup>2</sup> Ibid, halaman 2

#### 4. Middleware

Terdiri dari elemen software yang ada di client maupun di server.

### II.1.3. Distribusi Komponen software

Setelah komponen-komponennya sudah jelas, selanjutnya bagaimana mendistribusikan komponen-komponen tersebut. Cara pendistribusian komponen mengikuti model berikut<sup>3</sup>:

#### 1. Fat Client

User interaction/presentation, dan application pada client, sedangkan server menyediakan manajemen data. Model ini digunakan pada file server dan database server.

#### 2. Thin Client

Hanya user interaction /presentation yang ada pada client, sedangkan server menyediakan aplikasi/layanan dan data manajemen. Model ini berlaku untuk transaction dan groupware server.

Istilah Fat atau Thin didasarkan pada seberapa banyak business logic yang ada pada client, semakin sedikit, misalnya hanya untuk user interaction, maka makin Thin. Sebaliknya makin banyak business logic pada client, maka makin Fat. Misalnya browser kita tidak melakukan apa-apa kecuali menampilkan informasi atau bertindak sebagai user interaction saja. Pendekatan lain dalam mengalokasikan atau mendistribusikan komponen software C/S adalah dengan 5 jenis konfigurasi yang berbeda<sup>4</sup>:

#### 1. Distributed presentation

---

<sup>3</sup> Ibid, halaman 3

<sup>4</sup> Ibid

Database dan aplikasi/business logic dialokasikan di server, biasanya mainframe. Server juga berisi logika untuk menyiapkan informasi pada layar menggunakan software tertentu. Pada client ada software yang digunakan untuk mengubah ke karakter-karakter yang umum untuk ditampilkan di layar. Informasi dikirimkan dari server ke dalam presentasi dengan GUI di PC.

## 2. Remote presentation

Database dan aplikasi di server, data yang dikirimkan oleh server digunakan oleh client untuk menyiapkan presentasi di layar.

## 3. Distributed logic

Client fokus pada user presentation, proses data entry. Server fokus pada manajemen database memproses query dari client, update file server, kontrol client, dan aplikasi business lain.

## 4. Remote data management

Aplikasi pada server menciptakan sumber data baru dengan memformat data hasil extract dari tempat lain. Aplikasi yang ditempatkan pada client digunakan untuk memanfaatkan data baru yang baru diformat oleh server. DSS termasuk dalam katagori ini.

## 5. Distributed databases

Data yang membentuk database tersebar di beberapa server dan client. Karena itu client harus support komponen software manajemen data, komponen software aplikasi dan komponen GUI.

## II.2. Microsoft Visual Basic 6.0

Visual basic telah melalui sejumlah versi dan jauh berbeda dengan produk aslinya. Pada setiap versi baru Visual Basic ditingkatkan untuk memasukan teknologi pengembangan perangkat lunak. Perangkat lunak Microsoft diuntungkan dari penekanan perusahaan pada integrasi produk mereka, yang juga telah ditingkatkan dengan setiap rilis terbarunya.

Visual Basic 1.0 dirilis pada pertengahan tahun 1991. Perangkat lunak ini dikembangkan dari QuickBasic yang berbasis DOS, bahasa pemrograman BASIC versi Microsoft. Visual Basic direncanakan untuk kemudahan penggunaan dan untuk memungkinkan programmer generasi baru membuat aplikasi untuk sistem operasi Windows. Versi tulang-punggung pertama pada dasarnya merupakan versi grafis dari QuickBasic, dan memasukkan tool-tool untuk pembuatan elemen antarmuka-pengguna, seperti kotak teks, sintaks pemrograman, dan kontrol kustom.

Pada awal 1992, dirilis Visual Basic Professional Toolkit. Update ini merupakan sekumpulan paker dari kontrol kustom tambahan yang dibuat oleh Microsoft dan perusahaan lain. Pada saat itu orientasi objek dan reusabilitas menjadi konsep penting dalam pengembangan perangkat-lunak. Walaupun Visual Basic sendiri jauh dari sebuah bahasa berorientasi-objek, dukungannya pada kontrol kustom menjadi sebuah langkah dalam arah yang benar. Kontrol kustom menjadi nilai plus nyata bagi para pengembang karena penggunaannya berarti potongan kode yang menjalankan suatu fungsi khusus bisa distandarisasi dalam bagian yang bisa didaur-ulang. Para pengembang bisa menggunakan kontrol kustom untuk fungsionalitas yang kompleks, yang tidak akan menghabiskan

waktu untuk membuat kode dan mengujinya sendiri. Kemampuan ini menjadi salah satu alasan utama yang membuat Visual Basic menjadi begitu populer pada masa itu.

Visual Basic 2.0 dirilis pada tahun 1992 dan memasukkan tipe data varian, konstanta True dan False, serta variabel objek. Variabel objek menempatkan Visual Basic satu langkah di depan dalam orientasi objek dengan menyediakan kemampuan untuk mendeklarasikan dan merujuk objek.

Pada saat itu VBSQL dan ODBC API menjadi satu-satunya metode akses yang tersedia yang bisa digunakan para pengembang Visual Basic. Kedua metode menyediakan akses 16-bit, dan VBSQL menerima penghargaan yang meragukan sebagai antarmuka VB-asli untuk SQL Server. ODBC API adalah sebuah antarmuka database tingkat-rendah yang rumit dengan kurva belajar yang menanjak.

Pada tahun 1993 Visual Basic 3.0 dirilis dan memasukkan kontrol data standar. Kontrol kustom ini menyediakan akses database dalam aplikasi-aplikasi dengan pengkodean minimal. Versi ini juga mendukung kontrol kustom VBX (16-bit).

Versi Visual Basic ini disertai dengan mesin database Jet versi 1.1. Jet dipakai untuk konektivitas database melalui **DAO (Data Access Objects)** atau kontrol data. Walaupun Jet diteruskan hingga sekarang dengan versi 4.0-nya, **ADO (ActiveX Data Objects)** adalah metode akses data terbaru yang disukai karena merupakan salah satu komponen utama dalam strategi Universal Data Access Microsoft.

Visual Basic 4.0 diluncurkan pada tahun 1995 dan menjadi sebuah revisi yang menentukan. Perangkat lunak ini mengakomodasi kecanggihan pengembangan perangkat lunak dengan memasukkan teknologi OLE dan kemampuan untuk membuat objek. Versi ini juga mendukung kontrol kustom 32-bit yang disebut OCX. Pada saat ini, perubahan telah dibuat pada aplikasi lain, seperti Word dan Excel agar mereka bisa diakses dari dalam aplikasi Visual Basic menggunakan **VBA (*Visual Basic for Applications*)**.

Pada front akses data, versi Visual Basic ini disertai dengan **RDO (*Remote Data Objects*)** dan **Remote Data Control**. RDO adalah sebuah metode akses data yang dirancang dari bawah-ke-atas untuk menggantikan DAO. Pustaka ActiveX 32-bit ini lebih cepat dan lebih kecil daripada DAO, dan dirancang bersama sebuah hierarki objek yang sama dengan ODBC API. Sayangnya RemoteData Control dihindangi berbagai masalah dan tidak pernah dipakai.

Pada tahun 1997 Visual Basic 5.0 dirilis. Perangkat lunak ini mendukung standar **COM Microsoft** dan memungkinkan pembuatan kontrol ActiveX. Versi ini menjadi sebuah terobosan nyata karena sekarang para pengembang bisa membuat kontrol dan DLL mereka sendiri menggunakan Visual Basic, yang sebelumnya tidak mungkin.

Visual Basic 6.0 dirilis pada tahun 1998. Karena kebutuhan para pengembang, versi Visual Basic ini memuat metode baru yang telah ditingkatkan untuk berinteraksi dengan SQL Server. Ini termasuk akses data yang telah ditingkatkan, tool dan kontrol baru untuk dipakai bersama database (seperti ADO

Data Control), fitur Internet yang telah ditingkatkan (seperti kelas Web), dan sejumlah wizard baru.

ADO 2.0, model akses data terhebat dan terbaru pada saat ini, diluncurkan bersama Visual Basic 6.0. Antarmuka ke OLE DB ini serupa dengan RDO, namun dengan struktur yang kurang hierarkis dan lebih kecil. ADO dirancang bagi kinerja dalam perusahaan besar dan aplikasi Internet serta merupakan salah satu komponen utama dalam *Universal Data Access (UDA)* Microsoft.

Visual Basic adalah bagian dari *suite tool* pengembangan Microsoft yang disebut Visual Studio, yang juga memasukkan Visual C++, Visual FoxPro, Visual InterDev, Visual J++, dan Visual SourceSafe.

### **II.3. Winsock (Windows Socket)**

#### **II.3.1. Pengenalan Winsock**

Windows Sockets (disingkat “Winsock” atau “WinSock”) merupakan antarmuka pemrograman jaringan untuk Microsoft Windows. Dengan winsock maka memungkinkan beberapa komputer saling berhubungan atau saling berkomunikasi. Dalam organisasi internasional untuk standarisasi *Open System Interconnection (ISO/OSI)*, winsock beroperasi pada interface session layer sampai transport.

#### **II.3.2. Properti Winsock**

Winsock mempunyai beberapa properti, diantaranya :

1. BytesReceived



Fungsi properti ini untuk mengembalikan nilai/jumlah ukuran data yang diterima. Satuannya adalah byte dan jenis datanya adalah long.

## 2. LocalHostName

Fungsi properti ini untuk mengembalikan nama komputer lokal. Jenis datanya adalah string, sifatnya adalah Read-only.

## 3. LocalIP

Fungsi properti ini untuk mengembalikan IP address komputer lokal. Jenis datanya adalah string, sifatnya adalah Read-only.

## 4. LocalPort

Fungsi properti ini untuk mengembalikan port komputer lokal. Jenis datanya adalah string, sifatnya adalah *Read/Write*.

## 5. Protokol

Fungsi properti ini untuk memilih jenis protokol yang akan digunakan, ada 2 jenis protokol, yaitu :

- TCP (0-sck TCPProtokol) yang merupakan default.
- UDP (1-sck UDPProtokol)

## 6. RemoteHost

Fungsi properti ini untuk menentukan IP address atau nama komputer tujuan. Jenis datanya adalah string.

## 7. RemoteIP

Fungsi properti ini untuk mengembalikan IP address komputer *remote*. Jenis datanya adalah string.

## 8. RemotePort

Fungsi properti ini untuk menentukan atau mengembalikan port yang dituju pada komputer *remote*. Jenis datanya adalah long.

#### 9. State

Fungsi properti ini untuk mengembalikan nilai integer yang berkaitan dengan kondisi koneksi maupun error yang terjadi selama winsock beroperasi.

### II.3.3. Metode Winsock<sup>5</sup>

Metode yang digunakan pada winsock adalah sebagai berikut:

#### 1. Accept

Metode ini berfungsi untuk menerima sebuah koneksi baru. Setiap koneksi memiliki ID yang disebut RequestID yang tipe datanya long.

#### 2. Close

Metode ini berfungsi untuk menutup koneksi yang sedang aktif

#### 3. GetData

Metode ini berfungsi untuk menerima data yang dikirim oleh client

#### 4. Listen

Metode ini berfungsi agar winsock siap menerima koneksi baru pada port yang telah ditentukan

#### 5. PeekData

Hampir sama dengan fungsi getdata namun tidak menghapus data pada buffer. Metode ini jarang digunakan karena data pada buffer akan terus bertambah

#### 6. SendData

---

<sup>5</sup> Ibid

Metode ini berfungsi untuk mengirim data. Metode ini digunakan oleh server dan client

## II.4. Microsoft Access

### II.4.1. Memulai/Membuat Database di Microsoft Access

Adapun langkah-langkah untuk memulai/membuat database di Microsoft Access adalah sebagai berikut :

1. Klik tombol **Start** menu
2. Pilih Program → **Microsoft Access**
3. Maka akan tampil jendela **Microsoft Access**
4. Kemudian pilih blank database untuk membuat database baru
5. Pilih lokasi penyimpanan kemudian ketik nama database yang akan dibuat kemudian klik tombol create

### II.4.2. Membuat Tabel

Tabel merupakan bagian database yang digunakan untuk menampung data. Jika dalam database tidak terdapat tabel maka database tersebut tidak ada gunanya. Untuk membuat tabel di Microsoft Access, pada jendela database pilih *object table* maka akan tampil 3 (tiga) pilihan yaitu :

1. *Create table in Design view*

Digunakan untuk membuat tabel yang mana kolom dan tipe datanya ditentukan sendiri.

2. *Create table by using Wizard*

Digunakan untuk membuat tabel berdasarkan contoh tabel bawaan microsoft access.

### 3. *Create table by entering data*

Digunakan untuk membuat tabel dengan cara mengisi langsung nama kolom dan datanya.

## II.5. Flowchart

Flowchart (bagan alir) merupakan representasi secara grafik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah. Dengan menggunakan flowchart akan memudahkan kita untuk melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah. Di samping itu flowchart juga berguna sebagai fasilitas untuk berkomunikasi antara pemrograman yang bekerja dalam tim suatu proyek. Flowchart ada dua macam:

### II.5.1. Flowchart Sistem

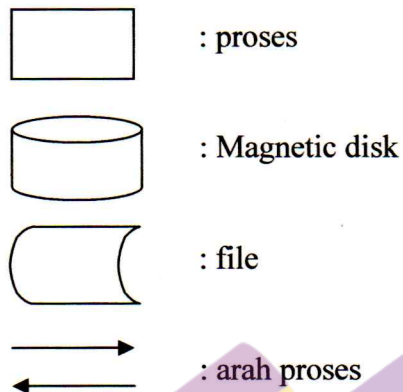
Flowchart sistem yaitu diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data dan perhubungan antar peralatan tersebut. Flowchart sistem digunakan untuk menggambarkan urutan langkah dalam memecahkan masalah tetapi hanya berisi prosedur dalam sistem yang dibentuk. Simbol yang digunakan adalah sebagai berikut



: input output berupa dokument

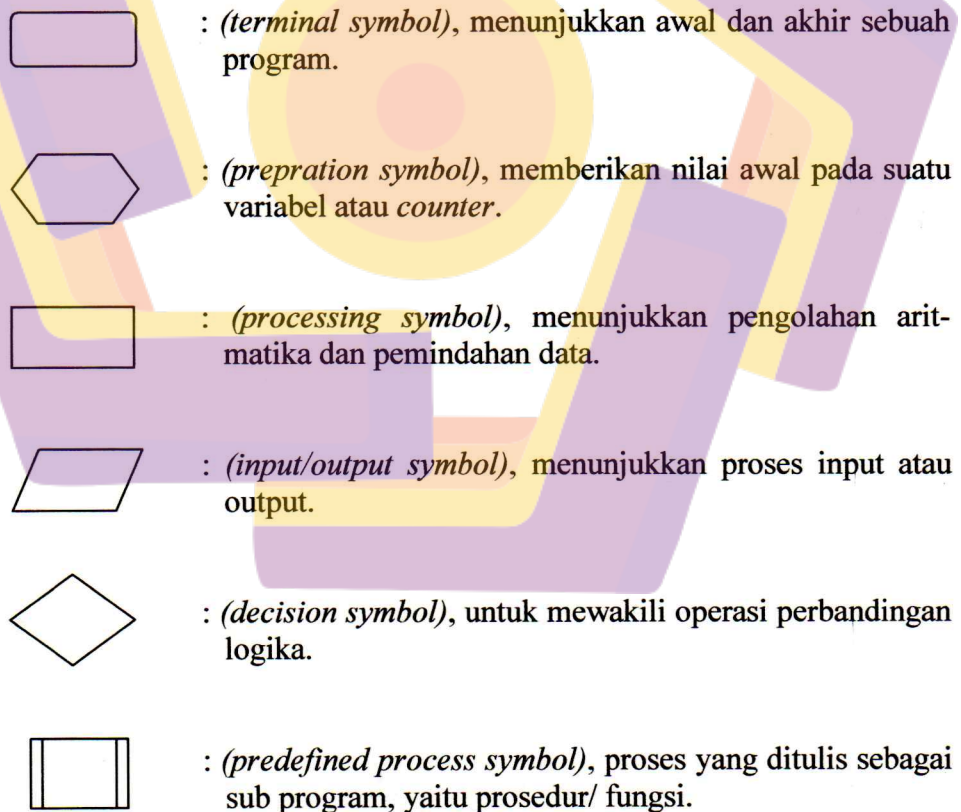


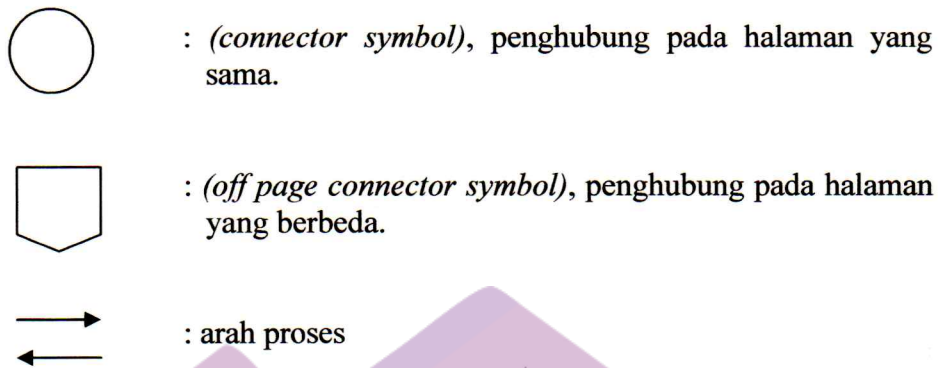
: manual input



### II.5.2. Flowchart Program

Flowchart Program yaitu bagan yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Simbol yang digunakan adalah *American National Standard Inc.*





## II.6. Data Flow Diagram (DFD)

Data flow diagram adalah diagram yang menggambarkan aliran data yang difokuskan pada fungsi sistem tanpa mengabaikan karakteristik fungsional. Komponen DFD terdiri dari sejumlah komponen yang sederhana: proses (**process**), aliran (**flows**), penyimpanan (**stores**) dan terminator. Secara garis besar DFD dikelompokkan menjadi dua model yaitu:

### II.6.1. Diagram Konteks

Aliran dalam diagram konteks memodelkan masukan ke sistem dan keluaran dari sistem, seperti halnya sinyal kontrol yang diterima atau dibuat sistem. Aliran data hanya digambarkan jika diperlukan untuk mendeteksi kejadian dalam lingkungan dimana sistem harus memberikan respon atau membutuhkan data untuk menghasilkan respon. Selain itu aliran data dibutuhkan untuk menggambarkan transportasi antara sistem dan *terminator*. Dengan kata lain aliran data digambarkan jika data tersebut diperlukan untuk menghasilkan respon pada kejadian tertentu.

## II.6.2. Diagram Level

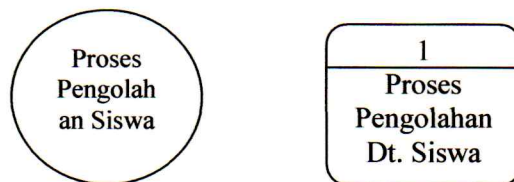
Diagram level menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data. Sebagai perangkat analisis, model ini hanya mampu memodelkan sistem dari satu sudut pandang yaitu sudut pandang fungsi.

Pertama kali digunakan pada rekayasa perangkat lunak sebagai notasi untuk mempelajari desain sistem, dengan menggunakan notasi *graph theory* yang selanjutnya menjadi notasi yang mengimplementasikan model kebutuhan pemakai sistem. Karena itu, model ini tidak hanya dapat digunakan untuk memodelkan sistem pemrosesan informasi tetapi bisa juga sebagai jalan untuk memodelkan keseluruhan organisasi, sebagai rencana kerja dan rencana strategi. Pada sejumlah kasus, model ini digunakan juga untuk memodelkan **real-time system** dan menggunakan notasi tambahan yang tidak diperlukan pada sistem yang *business-oriented*, tetapi lebih kepada **scientific** dan **engineering system**.

Ada empat komponen dalam model ini, yaitu:

### a. Proses

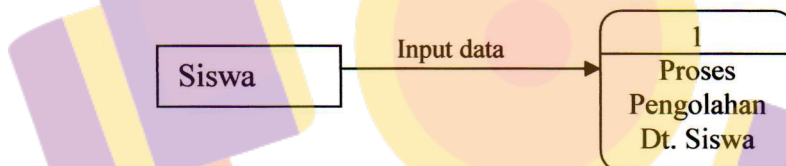
Proses kadang-kadang dinamakan juga **buble** (gelembung). Proses menunjukkan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran ataupun sebaliknya. Proses direpresentasikan dalam bentuk lingkaran atau bujursangkar dengan sudut melengkung. Pada umumnya didefinisikan dengan kata tunggal, atau kalimat sederhana.



**Gambar II.1.** Contoh Proses

b. Aliran

Komponen ini direpresentasikan dengan menggunakan panah yang menuju ke/dari proses. Digunakan untuk menggambarkan gerakan paket data atau informasi dari satu bagian ke bagian lain dari sistem dimana penyimpanan mewakili lokasi penyimpanan data. Ujung panah menggambarkan kemana data bergerak ke/ dari proses, penyimpanan ataupun terminator atau keduanya.



**Gambar II.2.** Contoh Aliran dan Proses

c. Penyimpanan

Komponen ini digunakan untuk memodelkan kumpulan data atau paket data. Notasi yang digunakan adalah garis sejajar, segiempat dengan sudut melengkung, atau persegi panjang.

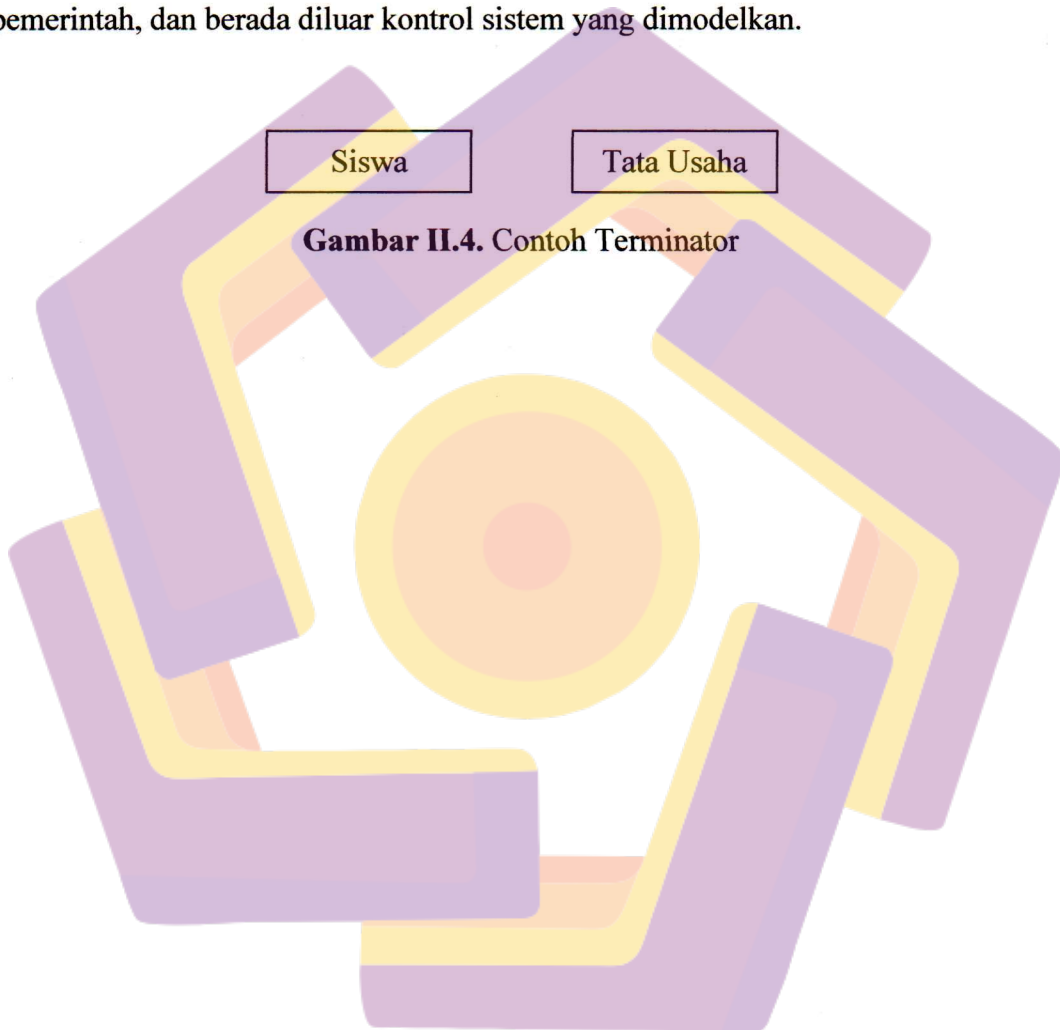


**Gambar II.3.** Contoh Penyimpanan



d. Terminator

Komponen ini dinotasikan dengan persegi panjang, yang mewakili entiti luar dimana sistem berkomunikasi. Biasanya notasi ini melambangkan orang atau kelompok orang misalnya organisasi di luar sistem, grup, departemen, perusahaan pemerintah, dan berada diluar kontrol sistem yang dimodelkan.



**Gambar II.4.** Contoh Terminator