

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Program Aplikasi**

Menurut Harip Santoso, aplikasi adalah Kelompok file (form, class, report) yang memiliki tujuan untuk menjalankan aktivitas tertentu yang saling berhubungan..

Program aplikasi sangat banyak ragamnya karena disesuaikan dengan kebutuhan masing-masing pembuat dan pemesannya. Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suit* aplikasi. Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi –aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Seringkali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

Aplikasi dapat digolongkan menjadi beberapa kelas, yaitu:

- a. Perangkat lunak perusahaan (*enterprise*).
- b. Perangkat lunak infrastruktur perusahaan.
- c. Perangkat lunak informasi kerja
- d. Perangkat lunak media dan hiburan.

- e. Perangkat lunak pendidikan.
- f. Perangkat lunak pengembangan media.
- g. Perangkat lunak rekayasa produk.

### 2.1.1 *Mobile*

*Mobile* menurut ([http://Agusbarupunyablogspot.com/2010/10/Pengertian-aplikasi -mobile.html](http://Agusbarupunyablogspot.com/2010/10/Pengertian-aplikasi-mobile.html)) dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon mobile berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Karakteristik perangkat *mobile* :

- a. Ukuran yang kecil  
Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
- b. *Memory* yang terbatas  
Perangkat *mobile* juga memiliki *memory* yang kecil, yaitu *primary* (RAM) dan *secondary* (*disk*).
- c. Daya proses yang terbatas  
Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.
- d. Mengonsumsi daya yang rendah  
Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*.
- e. Kuat dan dapat diandalkan  
Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan – benturan, gerakan, dan sesekali tetesan – tetesan air.
- f. Konektivitas yang terbatas  
Perangkat *mobile* memiliki *bandwith* rendah, beberapa dari mereka bahkan tidak tersambung.
- g. Masa hidup yang pendek  
Perangkat – perangkat konsumen ini menyala dalam hitungan detik, kebanyakan dari mereka selalu menyala.

## 2.2 Pengertian Informasi

Informasi merupakan salah satu sumber yang penting dalam kehidupan sehari-hari yang bergantung kepada informasi. Informasi tidak hanya dipakai oleh pihak internal dalam organisasi, tetapi juga dipakai oleh pihak eksternal (diluar organisasi). Setiap individu memerlukan informasi yang berbeda menurut kepentingan masing-masing. Konsep atau definisi informasi yang dikemukakan oleh Azhar Susanto adalah “Hasil Pengolahan data, akan tetapi tidak semua hasil pengolahan data tersebut bisa menjadi informasi, hasil pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut” (Susanto, 2004:4). Pendapat yang telah dikemukakan diatas menjelaskan bahwa informasi merupakan hasil dari pengolahan data yang memberikan makna atau arti yang berguna atau bermanfaat bagi orang tersebut. Informasi memiliki ciri-ciri sebagai berikut:

- a. Benar atau salah, dalam hal ini, informasi berhubungan dengan kebenaran terhadap kenyataan. Jika penerima informasi yang salah dipercaya, maka efeknya seperti kalau informasi itu benar.
- b. Baru, informasi dapat diperbaharui atau memberikan perubahan terhadap informasi yang telah ada.
- c. Tambahan, informasi dapat diperbaharui atau memberikan perubahan terhadap informasi yang telah ada.
- d. Korektif, informasi dapat digunakan untuk melakukan koreksi terhadap informasi yang sebelumnya yang salah atau kurang benar.
- e. Penegas, informasi dapat mempertegas informasi yang telah ada sehingga keyakinan terhadap informasi semakin meningkat. (Kadir, 2005:36).

Berdasarkan penjelasan diatas bahwa ciri-ciri informasi ada empat meliputi ciri informasi yang pertama yaitu benar yang artinya informasi harus sesuai dengan fakta berhubungan dengan kebenaran terhadap yang diinformasikan sehingga informasi tersebut dapat dipercaya dan tidak diragukan bagi yang menerima informasi tersebut. Ciri informasi yang kedua, yaitu baru yang artinya informasi yang diinformasikan hendaknya berdasarkan hal-hal yang terbaru. Ciri

informasi yang ketiga yaitu tambahan yang artinya bahwa hendaknya sebuah informasi harus memiliki nilai tambah agar menjadi pembeda dengan informasi yang telah ada sebelumnya. Ciri informasi yang keempat yaitu korektif yang artinya bahwa informasi yang diinformasikan hendaknya dapat dijadikan bahan untuk mengoreksi informasi yang sebelumnya agar dapat meminimalisir kesalahan pada informasi yang baru. Sedangkan ciri informasi yang kelima yaitu penegas yang artinya bahwa informasi yang ada hendaknya dapat dijadikan sebagai penegas guna meyakinkan seseorang akan kebenaran sebuah informasi tersebut.

Edhy Sutanta menjelaskan fungsi informasi dalam suatu informasi antara lain:

- a. Menambah pengetahuan.
- b. Mengurangi ketidakpastian.
- c. Mengurangi resiko kegagalan.
- d. Mengurangi keanekaragaman yang tidak diperlukan.
- e. Memberi standar, aturan, ukuran, dan keputusan yang menentukan pencapaian sasaran dan tujuan. (Sutanta, 2003:11).

Berdasarkan penjelasan diatas, maka informasi seharusnya dapat berguna untuk menambah pengetahuan bagi penerimanya. Sehingga dapat digunakan sebagai bahan pertimbangan yang mendukung proses pengambil keputusan dan informasi akan mengurangi ketidakpastian, karena apa yang akan terjadi dapat diketahui sebelumnya serta menghindari keraguan pada saat pengambilan keputusan.

### **2.2.1 Pengertian Sistem Informasi**

Sistem informasi merupakan bentuk penerapan dalam sebuah organisasi atau instansi. Penerapan/penggunaan sistem informasi dalam sebuah organisasi atau instansi tersebut untuk mendukung dalam mengumpulkan dan mengolah data dan pengendalian. Azhar Susanto mendefinisikan sistem informasi sebagai berikut:

“Sistem informasi adalah kumpulan dari sub-sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna” (Susanto, 2004:55).

Secara umum sistem informasi adalah sebuah sistem manusia/mesin yang terpadu untuk menyajikan informasi guna mendukung fungsi operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi. Adapun kegiatan sistem informasi menurut Jogiyanto, sebagai berikut:

- a. Input, menggambarkan suatu kegiatan untuk menyediakan data untuk proses.
- b. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
- c. Output, suatu kegiatan untuk menghasilkan laporan dari suatu proses informasi.
- d. Kontrol, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan. (Jogiyanto, 2005:20).

Berdasarkan pendapat diatas, menjelaskan bahwa sistem informasi merupakan kumpulan dari sub sistem baik fisik maupun non fisik yang saling berhubungan dan bekerja sama antara yang satu dengan yang lainnya untuk mencapai suatu tujuan. Jadi, sistem informasi merupakan pengolahan data menjadi informasi yang berguna untuk orang banyak yang membutuhkan informasi tersebut. Sistem informasi merupakan rangkaian kegiatan yang terdiri dari input dalam hal ini dalam sebuah informasi harus menyediakan data-data yang akan dijadikan informasi yang kemudian data tersebut melalui proses data, sehingga mendapatkan output yang berupa informasi yang mempunyai nilai tambah terhadap yang membutuhkan.

### **2.3 Akademik**

Kata akademik berasal dari bahasa Yunani yakni *academos* yang berarti sebuah taman umum (plasa) di sebelah barat laut kota Ahena. Nama *academos* adalah nama seorang pahlawan yang terbunuh pada saat perang legendaris Troya.

Pada plaza inilah filosof Socrates berpidato dan membukan arena perdebatan tentang berbagai hal. Tempat ini juga menjadi tempat Plato melakukan dialog dan mengajarkan pikiran-pikiran filosofisnya kepada orang-orang yang datang. Sesudah itu, kata *academos* berubah menjadi akademik, yaitu semacam tempat perguruan. Para pengikut perguruan tersebut disebut *academis*, sedangkan perguruan semacam itu disebut *academia*. Berdasarkan hal ini, inti dari pengertian akademik adalah keadaan orang-orang bisa menyampaikan dan menerima gagasan, pemikiran, ilmu pengetahuan, dan sekaligus dapat mengujinya secara jujur, terbuka, dan leluasa (Fadjar, 2002:5).

Menurut kamus besar bahasa Indonesia edisi ke-2, akademi adalah lembaga pendidikan tinggi kurang lebih tiga tahun lamanya yang mendidik tenaga profesi. Sedangkan akademik adalah proses yang mendidik tenaga profesi yang bersifat akademik. Bidang akademik menurut Suryobroto (2004:42-45) adalah bidang administrasi yang berhubungan dengan pembelajaran yang digolongkan pada tiga bidang manajemen sebagai berikut:

- a. Pengelolaan sebelum proses pembelajaran.
  - 1) Menyusun program pembelajaran.
  - 2) Menyusun rencana pelaksanaan pembelajaran.
  - 3) Mengatur pembagian tugas mengajar.
  - 4) Menyusun jadwal pelajaran/
- b. Pengelolaan selama proses pembelajaran.
  - 1) Mengatur daftar kemajuan siswa
  - 2) Mengatur kehadiran dan absensi siswa.
  - 3) Mengadakan evaluasi siswa.
- c. Pengelolaan sesudah proses pembelajaran.
  - 1) Menyusun laporan hasil pendidikan
  - 2) Menyusun analisis hasil evaluasi.
  - 3) Mengadakan remedi siswa.

(Suryosubroto, 2004:42-45).

Pengelolaan bidang akademik tersebut akan terasa sangat kompleks bila dilakukan secara konvensional atau manual. Pengelolaan bidang akademik akan

menjadi lebih efektif dan efisien bila menggunakan suatu sistem yaitu sistem informasi akademik.

Sistem informasi akademik adalah kumpulan elemen atau bagian-bagian baik fisik maupun non fisik dan prosedur yang saling berhubungan satu sama lain menjadi sebuah informasi.

Secara umum data-data yang akan diolah dalam sistem informasi akademik yaitu data-data yang berhubungan dengan aktifitas akademis mulai dari pendaftaran hingga pengolahan nilai. Dalam pembuatan suatu sistem akademik dibuat berdasarkan kebijakan-kebijakan yang berada di sebuah sekolah sehingga proses pengolahan data akademik antar sekolah akan berbeda sesuai dengan kebijakan itu sendiri.

Berdasarkan penjelasan diatas maka dapat disimpulkan bahwa akademik adalah proses mendidik para siswa menjadi ahli dalam hal akademis dan dapat menerima gagasan dan menyampaikan gagasan sehingga gagasan tersebut dapat diuji dengan terbuka oleh semua orang.

## **2.4 Android**

Android merupakan sebuah sistem operasi yang berbasis Linux untuk perangkat portable seperti *smartphone* dan komputer tablet. Android menyediakan platform terbuka (*open source*) bagi programmer untuk mengembangkan aplikasi sendiri pada berbagai perangkat dengan sistem android.

### **2.4.1 Sejarah Android**

Awalnya, Google sebagai perusahaan raksasa membeli android inc beserta teknologinya. Google melanjutkan untuk mengembangkan android untuk bersaing pada pasar *smartphone*. Dalam usaha pengembangan android, dibentuklah *Open Handset alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Akhirnya pada awal Maret 2009 Google berhasil merilis sistem operasi android hasil pengembangan yang pertama yaitu android versi 1.1 pada perangkat

*smartphone*. Pada tahun yang sama Google kembali merilis sistem operasi versi 1.5 yang dikenal dengan nama cupcake.

Cupcake adalah nama pertama yang digunakan untuk versi sistem operasi android, yaitu dimulai dari versi 1.5. Google merilis setiap sistem operasi android bersamaan dengan perangkat lunak pengembangnya yang disebut **Android Software Development Kit (Android SDK)**.

#### **2.4.2 Versi Android**

Semenjak dirilisnya sistem operasi android versi 1.1, hingga saat ini telah tersedia berbagai versi dari sistem operasi ini.

Berikut beberapa versi sistem operasi android yang telah dirilis dari periode 2009 sampai 2012.

**a. Android 1.1 (Februari 2009)**

Android versi 1.1 dirilis oleh Google pada 9 Maret 2009. Fitur tampilan aplikasi dan jam alarm yang lebih baik, *voice search*, aplikasi Gmail dan pemberitahuan email.

**b. Android 1.5 Cupcake (2009)**

Android versi 1.5 (*Cupcake*) dirilis pada pertengahan Mei 2009 yang berbasis android dan SDK (*Software Development Kit*). Adanya pembaruan dalam hal menggunakan video dengan modus kamera, mengunggah video Player dan capture, upload video dan gambar ke internet, Bluetooth A2DP, *animated screen saver*, Qwerty layar sentuh.

**c. Android 1.6 Donut (September 2009)**

Android versi 1.6 (*Donut*) dirilis pada September 2009, memberikan keunggulan yang lebih baik dalam hal proses pencarian, kontrol applet, indikator baterai, CDMA/EVDO resolusi VWGA, Wi-fi dan teknologi *text to speech*. Integrasi penuh kamera foto dan video dengan galeri.

**d. Android 2.0/2.1 Eclair (Oktober 2009/Januari 2010)**

Android versi 2.0/2.1 (*Eclair*) dirilis pada tanggal 3 Desember 2009. Kamera sampai dengan 3,2 Mega pixel, dukungan HTML5, dan bluetooth 2.1. dukungan resolusi lebih tinggi dengan *fitur contrast ratio* yang lebih

baik. Peningkatan fitur agenda pada kalender. Penambahan fitur kamera (*flash support, digital zoom, scene mode, white balance, color effect* dan *macro focus*).

**e. Android 2.2 Froyo: Frozen Yogurth (Mei 2010)**

Android versi 2.2 (*Froyo: Frozen Yogurt*) dirilis pada tanggal 20 Mei 2010. Mendukung Flash 10.1, instal aplikasi ke SD Card, hotspot portable, dan fitur update otomatis android market. Integrasi browser Chrome javascript. Fitur USB tethering dan Wi-fi hotspot. Mendukung kualitas video 720p. Fitur zoom gesture galeri foto.

**f. Android 2.3 Gingerbread (Desember 2010)**

Android versi 2.3 (*Gingerbread*) dirilis pada 6 Desember 2010. Peningkatan performa untuk bermain game, efek audio (*reverb, virtualization, bass boost*), dan multi kamera. Fitur download manager lebih baik. Mendukung format partisi ext4. Peningkatan pada audio, grafis, input kontrol di game. Peningkatan efisiensi pada penggunaan baterai.

**g. Android 3.0/3.1 Honeycomb (Februari 2011/ Mei 2011)**

Android versi 3.0/3.1 (*Honeycomb*) dirilis pada Mei 2011. Sistem operasi android khusus untuk perangkat tablet, dioptimalkan untuk perangkat dengan multi prosesor dan performa grafis game.

**h. Android 4.0 ICS: Ice Cream Sandwich (Oktober 2011)**

Android versi 4.0 (ICS : *Ice Cream Sandwich*) diumumkan pada tanggal 19 Oktober 2011. Pengembangan dari honeycomb untuk bisa digunakan pada perangkat non-tablet. Perubahan antarmuka dan peningkatan fitur. Mendukung pemanfaatan kinerja hardware pada tampilan antarmuka, dan P2p Wi-fi (*Wi-fi Direct*). Fitur untuk menggunakan keyboard eksternal dari USB.

**i. Android 4.1 Jelly Bean (Juli 2012)**

Peningkatan pada fitur voice search dan aplikasi kamera. Multichannel audio dan mendukung perangkat audio eksternal melalui USB.

**j. Android versi 4.4 (*KitKat*)**

Android versi 4.4 (*KitKat*) dirilis pada tanggal 31 Oktober 2013 yang merupakan generasi android keluaran terbaru setelah *jelly bean*. Kelebihan dari versi ini adalah *immersive mode*, akses kontak langsung dari aplikasi telepon, *display* layar lebih baik dari sebelumnya, menggunakan *processor tri core*, mampu menjalankan aplikasi bersamaan dengan tingkat kestabilan yang lebih baik.

## 2.5 *Object Oriented Programming*

OOP (*Object Oriented Programming*) atau Pemrograman Berorientasi Objek adalah suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba-atasi dengan buatan komputer. OOP mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu.

Objek adalah konsep atau abstraksi tentang sesuatu yang memiliki arti bagi aplikasi yang akan kita kembangkan. Secara umum, dapat dikatakan bahwa objek adalah sesuatu di dunia nyata yang memiliki peranan tertentu dalam aplikasi yang akan dikembangkan (Adi Nugroho, 2003). Menurut literatur lain (Ali Bahrami, 1999), objek dapat digolongkan menjadi beberapa kategori sebagai berikut :

- a. **Objek Konsep.** Konsep adalah ide atau pemahaman yang kita miliki tentang suatu fakta yang ada di dunia nyata. Objek konsep tidak terukur tetapi sering digunakan untuk mengorganisasi aktivitas bisnis atau digunakan untuk berkomunikasi. Untuk berkomunikasi dengan seseorang, kita harus berbagi konsep yang kita pahami secara individual untuk kemudian memperoleh pemahaman bersama. Objek konsep ini sangat penting sebab tanpa konsep, pemikiran kita akan menjadi kacau. Contoh dari objek konsep adalah kinerja.
- b. **Objek Event.** Objek event adalah sebuah titik dalam waktu yang harus direkam. Sesuatu terjadi, selalu terjadi pada waktu tertentu; baik sebagai event tunggal ataupun sebagai bagian dari urutan event. Sesuatu yang berhubungan dengan event adalah pertanyaan-pertanyaan seperti kapan, siapa, apa, dimana, bagaimana, atau mengapa. Contoh dari objek event

adalah pendaratan, interupsi, permintaan, penekanan tombol serta pemesanan.

- c. **Objek Organisasi.** Objek organisasi adalah kumpulan orang-orang, sumberdaya, fasilitas, atau kelompok yang dimiliki pengguna. Contohnya adalah departemen akuntansi, departemen riset dan pengembangan, dan sebagainya.
- d. **Objek Manusia.** Objek manusia menampilkan peran-peran yang berbeda dalam hubungannya dengan sistem/perangkat lunak yang kita kembangkan. Contoh objek manusia adalah karyawan, klien, direktur, dosen, serta manager.
- e. **Objek Tempat.** Objek tempat meliputi lokasi fisik yang harus dicatat dalam sistem, misalnya gedung, toko, kantor, dan sebagainya.
- f. **Sesuatu Yang Terukur Serta Objek Peralatan.** Objek ini meliputi objek fisik serta kelompok objek yang terukur serta peralatan-peralatan yang berinteraksi dengan sistem yang kita kembangkan. Contohnya, mobil adalah sesuatu yang terukur sedangkan sensor tekanan adalah peralatan.

## 2.6 Pengertian Internet

Pengertian internet menurut seorang pakar internet asal Indonesia, Onno W. Purbo menjelaskan bahwa Internet dengan berbagai aplikasinya seperti *Web*, *VolP*, *E-mail* pada dasarnya merupakan media yang digunakan untuk mengefisiensikan proses komunikasi.

Menurut tim penelitian dan pengembangan wahana komputer (2005,p4), internet adalah metode untuk menghubungkan berbagai komputer ke dalam satu jaringan global, melalui protokol yang disebut *Transmission Control Protocol / Internet Protoco* (TCP/IP).

### 2.6.1 Cara kerja internet

Jumlah pengguna internet yang besar dan semakin berkembang, telah mewujudkan budaya internet. Internet juga mempunyai pengaruh besar atas ilmu dan pandangan dunia. Dengan hanya berpanduan mesin seperti *Google*, pengguna

diseluruh dunia mempunyai akses internet yang mudah atas bermacam-macam informasi, internet melambangkan penyebaran atau pengetahuan informasi dan data secara ekstrim.

Hal – hal yang berperan penting dalam sistem kerja internet dijelaskan sebagai berikut :

a. *Server Client*

Untuk terhubung dengan halaman web tertentu atau komputer dalam keadaan online, terlebih dulu kita harus terhubung dengan *server*, yang bekerja sepanjang hari kecuali jika ada gangguan atau sedang down. Server adalah komputer yang mengirimkan data dari dan ke dalam internet. Pengguna internet yang meminta informasi dari *server* disebut *client*, Komputer yang kita gunakan adalah **client** yang meminta informasi dari ISP (*Internet Provider Service*), yang berfungsi sebagai *server*.

b. Antena Pemancar dan *Satelit*

Server terhubung dengan antena pemancar yang selanjutnya juga terhubung dengan satelit, sehingga proses transmisi data dapat berjalan dengan lancar.

c. *Modem* dan *Router*

Antara *server* dengan *client*, menggunakan penghubung yang dinamakan modem. Jika koneksi yang diperlukan hanya untuk satu komputer, modem dapat langsung terhubung ke komputer (*client*) setelah terhubung dengan *server*. Namun, jika koneksi ditunjukkan untuk beberapa komputer, modem setidaknya dihubungkan ke *router* terlebih dahulu sebelum tersambung ke komputer-komputer *client*.

d. Protokol

Semua *server* dan *client* berkomunikasi dengan bahasa komputer yang disebut protocol. Jika terhubung dengan internet, maka akan melihat setiap alamat situs web diawali dengan `http://` yang merupakan cara kerja *World Wide Web*. Ini merupakan *default address* bagi kebanyakan situs di dunia dengan tujuan agar tidak perlu lagi mencari-cari protokol untuk mengakses sebuah situs. Cukup dengan satu protokol meski sebenarnya

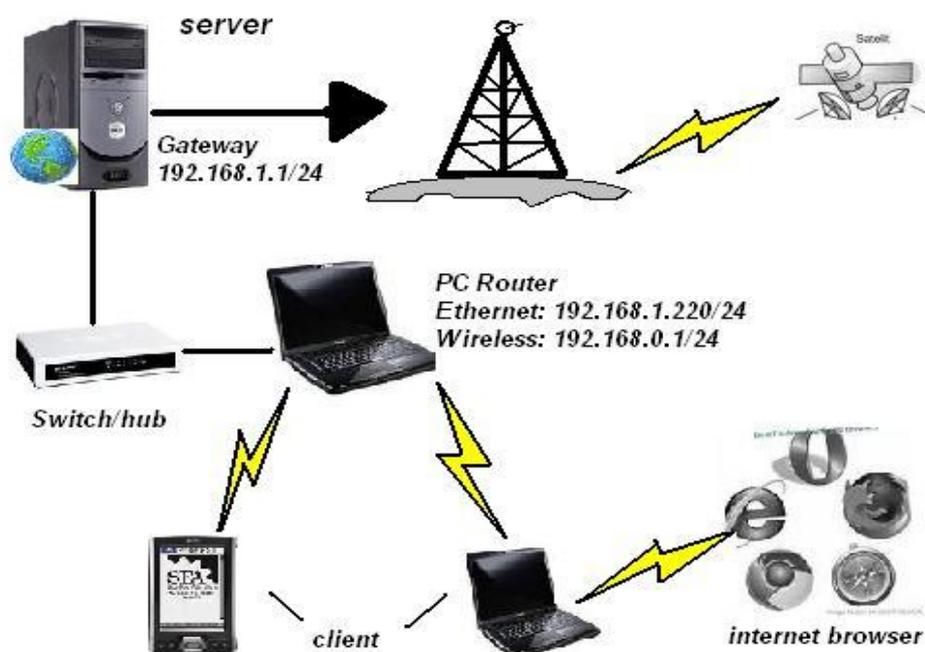
ada beberapa protokol lain untuk *electronic mail*. *Transfer* data komputer, atau yang lainnya.

e. IP

Paket Informasi komputer dikirim dari komputer ke komputer menggunakan alamat numerik yang disebut *IP Address*. *IP Address* memberikan informasi dari mana paket berasal dan akan ditunjukkan ke mana. Terdiri dari 4 deret angka dipisahkan oleh titik. Misal, 123.567.341.2.

f. Internet *Browser*

Agar anda dapat melihat informasi yang dikirim oleh komputer *server*, membutuhkan sebuah *software browser/internet browser*, Contoh *software* ini adalah *Internet Explorer, Firefox, Netscape, Opera* atau *Google Chrome*, *Software-software* tersebut menerjemahkan informasi yang dikirim server kepada *client*, sehingga memudahkan pengguna dalam mendapatkan informasi yang diinginkan.



**Gambar 2.1 Skema Cara Kerja Internet**

## 2.6.2 Cara-Cara Memperoleh Sambungan Internet

Ada beberapa macam koneksi yang dapat dilakukan agar dapat terkoneksi dengan internet, dan melakukan aktivitas online sepuasnya. Jenis-jenis koneksi juga menentukan kecepatan akses internet.

### a. Dial up

Melalui jalur PSTN (public switched telephone network) adalah koneksi ke internet melalui jaringan telephon reguler / telephon rumah. Koneksi ini paling murah dan sederhana dan pada umumnya di gunakan oleh orang-orang yang mengakses internet dari rumah. Koneksi ini memiliki keterbatasan bandwidth / lebar saluran yang di lalui transmisi data dengan kecepatan akses internet 56 kbps. Contoh: telkomnet instan.

### b. Jaringan

Adalah salah satu cara untuk terhubung ke internet dengan menghubungkan komputer kita ke jaringan komputer yang terhubung ke internet. Cara ini banyak di gunakan di perusahaan, kampus-kampus dan warnet. Komputer yang di jadikan server di hubungkan ke internet komputer lain yang di jaringan tersebut di hubungkan ke server tersebut. Komputer yang berfungsi sebagai server di hubungkan ke ISP melalui kabel telepon / antenna. Untuk menghubungkan komputer ke komputer server menggunakan kartu LAN dan KABEL UTP.

### c. Jaringan GPRS ( General Packet Radio Service)

Jaringan GPRS ( General Packet Radio Service) adalah koneksi yang menggunakan perangkat ponsel atau pda (personal data asistan) yang berfungsi sebagai modem dan di hubungkan ke komputer melalui kabel data, infrared atau bluetooth dengan menggunakan operator ponsel sebagai ISP. Komunikasi data dan suara di lakukan dengan menggunakan gelombang radio. GPRS menjanjikan kecepatan 56 kbps sampai 114 kbps. Keunggulan dari GPRS adalah dapat mengakses internet dari mana saja yang penting memiliki sebuah komputer, telephon seluler yang di lengkapi fasilitas GPRS dan sebuah simcard. Kelemahannya untuk mendapatkan

koneksi internet secara mobile kita harus melakukan setting tertentu pada telepon seluler kita.

d. *Leased Line*

*Leased Line* adalah fasilitas lain yang di sewa untuk melakukan koneksi ini kita perlu memakai TCP / IP software pada suatu jaringan biasanya di rancang oleh perusahaan untuk di jual pada perusahaan lainnya. Dalam membuat fasilitas ini di perlukan waktu, biaya yang infrastruktur yang lebih.

e. *Frame Relay*

*Frame Relay* adalah layanan koneksi yang memungkinkan beberapa pengguna menggunakan satu jalur transmisi pada waktu bersamaan. Pada saat maktu jalur lalu lintas padat, jaringan ini lebih efisien di bandingkan dengan leased line khusus untuk satu pelanggan yang umumnya menggunakan 10% -20% dari kapasitas bandwidth yang di sediakan.

f. *ADSL (Asymmetric Digital Subscriber Line)*

*ADSL (Asymmetric Digital Subscriber Line)* adalah koneksi internet melalui kabel telepon rumah dengan menggunakan modem ADSL sirkuit DSL dikonfigurasi untuk menghubungkan 2 tempat yang spesifik seperti pada *leased line*. Koneksi DSL jauh lebih cepat di bandingkan koneksi dial up jalur PSTN./ rata kecepatan download antara 256 kbps-8mbps pada jarak 1,5 km. Untuk kecepatan upload paling lambat yaitu 64kbps-256kbps. Contohnya speedy.

g. *WiFi (Wiereles Fedelity)*

*WiFi (Wiereles Fedelity)* adalah teknologi jaringan tanpa kabel yang menggunakan frekuensi tinggi frekuensi itu berada pada spectrum 2,4 GHz. WiFi mempunyai kemampuan akses internet samapi kecepatan hingga 11 MBps. Daerah yang mempunyai sinyal WiFi adalah daerah yang berada pada radius 100 meter dari titik akses.

h. *ISDN (Integarated Service Digital Network)*

*ISDN (Integarated Service Digital Network)* adalah istilah untuk saluran telepon digital yang dapat di manfaat kan untuk menyaklurkan data berformat digital dengan kecepatan transfer data yang lebih tinggi dengan

menggunakan saluran telepon reguler. Saluran ISDN memungkinkan akses internet dengan kecepatan 128 Kbps.

i. VSAT

VSAT adalah solusi terhadap pengguna internet yang berada di tempat terpencil karena tidak tersedianya jaringan telepon, *leased line*, maupun ISDN. Koneksi ini membutuhkan perangkat antenna VSAT yang berbentuk piringan berukuran besar dan di arahkan ke langit.

j. Tv Kabel

Tv Kabel adalah untuk menghubungkan komputer ke internet telah banyak di lakukan. Kelebihan mengakses internet dengan TV KABEL adalah kita dapat mmengakses setiap saat dan bebas dari gangguan telepon sibuk. Jaringan TV KABEL adalah jaringan dengan jalurb komunikasi data yang lebar.

k. *Handphone*

*Handphone* adalah anda harus dapat mengoprasikan HP tersebut. Komputer anda dapat di hubungkan dengan HP yang anda gunakan melalui Infra merah.atau kabel USB.

l. *Wiereles Broadband*

*Wiereles Broadband* adalah saat ini di kembangkan sebuah teknologi dengan nir kabel baru yang di sebut dengan wireless broadband. Dibandingkan dengan WiFi, Wibro dapat di akses dari jarak 1 KM dari titik akses. Kecepatan nya 512 KBps.

m. Kabel Data Modem Hp

Kabel Data Modem Hp adalah koneksi internet menggunakan kabel data ke hp yang sudah memiliki fasilitas GPRS.

n. Bluetooth Modem Hp

Bluetooth Modem Hp adalah koneksi internet melalui bluetooth yang tersambung ke HP yang sudah memiliki fasilitas GPRS.

o. Broadband Dengan GSM

Broadband Dengan GSM adalah salah 1 sistem koneksi ke internet yang banyak di pakai. Dengan banyak operator yang telah mendukung maka

sistem ini yang salah satu di minati. Kelemahan sistem ini adalah kecepatan akses internet sangat tergantung dari jarak ke BTS terdekat operator yang kita pakai.

p. Wireless LAN

Wireless LAN adalah koneksi internet menggunakan antenna khusus tanpa kabel atau wave LAN yang di pakai ke ISP .

q. PCMCIA

PCMCIA adalah koneksi internet menggunakan perangkat PCMCIA yang digunakan pada laptop.

r. Modem UHF

Modem UHF adalah koneksi internet melalui antenna UHF.

## **2.7 UML (Unified Modelling Language)**

Menurut Hend, UML adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML merupakan salah satu alat bantu yang handal di dunia pengembangan sistem berorientasi obyek. Hal ini karena UML menyediakan bahasa pemodelan visual yang mendukung bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan kepada calon pengguna sistem dan pemrograman.

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch*, *metodologi coad*, *metodologi OOSE*, *metodologi OMT*, *metodologi shlaer-mellor*, *metodologi wirfs-brock* dan sebagainya. Masa itu terkenal dengan perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerja sama dengan grup/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai Pada Oktober 1994, Dr. James Rumbaugh bergabung dengan Perusahaan *Rational software*, dimana Grady Booch sudah bekerja disana sebelumnya. Grady Booch mengembangkan *Object Oriented Design* (OOD) dan Dr. James Rumbaugh mengembangkan *Object Modeling Technique* (OMT). Duet Mereka pada Oktober 1995 menghasilkan *Unified Method* versi 0.8.

Musim gugur 1995 Dr. Ivar Jacobson ikut pula bergabung dengan duet Rumbaugh-Booch, dengan memperkenalkan *tool use case*. Trio tersebut pada bulan Juni 1996 menghasilkan *Unified Modeling Language* (UML) versi 0.9. Sebelumnya Dr. Ivar Jacobson mengembangkan *Object Oriented Software Engineering* (OOSE).

Trio ini mengembangkan *Ratinal Unified Process* (RUP). Dari konsorsium tersebut pada bulan Januari 1997 lahirlah UML versi 1.0. Pada bulan September 1997 lahirlah UML versi 1.1, dengan 8 buah diagram, yaitu:

- a. *Use case diagram*
- b. *Activity diagram*
- c. *Sequence diagram*
- d. *Collaboration diagram*
- e. *Class diagram*
- f. *Statechart diagram*
- g. *Component diagram*
- h. *Deployment diagram*

Pada bulan November 1997 sebuah organisasi *non profit* standarisasi *Object Management Group* (OMG) mengakui UML sebagai sebuah bahasa pemodelan standar untuk aplikasi *object oriented*. OMG didirikan pada bulan April 1989 oleh sebelas perusahaan *software*, dengan kantor pusat di Needham, MA, USA. ([www.omg.org](http://www.omg.org)).

Pada tahun 1999 lahirlah UML versi 1.3, menjadi 9 buah diagram, dengan penambahan *Business use case diagram*. Pada May 2001 lahirlah UML versi 1.4, menjadi 10 buah diagram, dengan penambahan *Object Diagram*.

Pada tahun 2002 lahirlah UML versi 2.0, menjadi 13 buah diagram, dengan penambahan dan penggantian yaitu:

- a. *Use case diagram*
- b. *Activity diagram*
- c. *Sequence diagram*
- d. *Communication Diagram (Collaboration diagram in versi 1.x)*
- e. *Class diagram*
- f. *State Machine Diagram (Statechart diagram in versi 1.x)*
- g. *Component diagram*
- h. *Deployment diagram*
- i. *Composite Structure Diagram*
- j. *Interaction Overview Diagram*
- k. *Object Diagram*
- l. *Package Diagram*
- m. *Timing Diagram*

Alasan mengapa memilih UML ialah sebagai berikut :

- a. Mempunyai Kombinasi 3 konsep :
  - 1) OOD (*Object Oriented Design*) from Grady Booch
  - 2) OMT (*Object Modelling Technique*) from James Rumbaugh
  - 3) OOSE (*Object Oriented Software Engineering*) from Ivar Jacobson
- b. Pengakuan dari pemimpin industri seperti Microsoft, IBM, Oracle, HP.
- c. Pengakuan dari organisasi non profit standarisasi OMG (*Object Management Group*) [www.omg.org](http://www.omg.org).

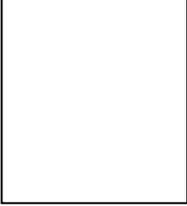
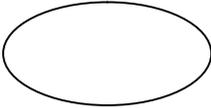
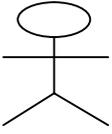
Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. Adapun tujuan dari UML adalah sebagai berikut:

- a. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model yang mudah dan dimengerti secara umum.
- b. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- c. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

### 2.7.1 Use Case Diagram

*Use case* diagram merupakan suatu diagram yang berisi *use case*, aktor, serta *relationship* diantaranya. *Usecase* diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Usecase* diagram dapat digunakan untuk menentukan kebutuhan apa saja yang diperlukan dari suatu sistem. Jadi dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan sesuatu, bagaimana cara aktor akan menggunakan sistem, serta apa saja yang dapat dilakukan terhadap suatu sistem. Notasi yang digunakan dalam *usecase* adalah persegi panjang yang merupakan *system boundary*, oval yang merupakan suatu proses, dan gambar orang yang berinteraksi dalam proses tersebut.

**Tabel 2.1** Komponen *Use Case Diagram*

	<i>System Boundary</i> menggambarkan batasan antara sistem dengan <i>actor</i>
	Simbol ini menggambarkan interaksi antara <i>actor</i> dengan <i>software</i> aplikasi tersebut
	<i>Actor</i> menggambarkan pengguna dari sistem, dapat berupa manusia atau sistem terotomatisasi lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi
	Menggambarkan hubungan antar <i>actor</i> dan <i>use case</i>

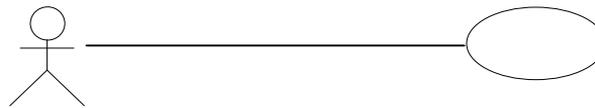
(Rosa, A. S. dan M. Shalahudin: 2013)

a. Relasi / *Associations*

*Association* menggambarkan bagaimana aktor terlibat dalam *usecase* dan bukan menggambarkan aliran data atau informasi. Relasi (*relationship*) digambarkan sebagai bentuk garis antara dua simbol dalam *usecase diagram*.

1) *Association* antara aktor dan *usecase*

Ujung panah pada *association* antara aktor dan *usecase* mengindikasikan siapa/apa yang meminta interaksi dan bukannya mengindikasikan aliran data. Sebaiknya menggunakan garis tanpa panah untuk *association* antara aktor dan *usecase*. *Association* antara aktor dan *usecase* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.



**Gambar 2.2** Assosiasi antara aktor dan *usecase* dalam *UML*

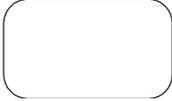
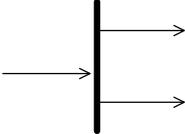
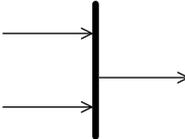
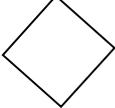
### 2.7.2 *Activity Diagram*

*Activity Diagram* adalah teknik untuk menggambarkan *logika procedural*, proses bisnis dan alur kerja. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku parallel sedangkan *flowchart* tidak bisa.

*Activity diagram* menggambarkan berbagai alur kerja dari satu aktivitas ke aktivitas lainnya dalam sistem yang sedang dirancang, bagaimana masing-masing alur aktivitas berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem dan interaksi antar *subsistem* secara *eksak*, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *usecase* pada *usecase diagram*, atau bahkan tanpa menggunakan *usecase*.

Beberapa simbol pada *Activity Diagram* :

**Tabel 2.2** Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Start Point</i>
	<i>End Point</i>
	<i>Activities</i>
	<i>Fork</i> (Percabangan)
	<i>Join</i> (Penggabungan)
	<i>Decision</i>
Swimlane	Sebuah cara untuk mengelompokkan <i>activity</i> berdasarkan Aktor (mengelompokkan <i>activity</i> dalam sebuah urutan yang sama)

(Rosa, A. S. dan M. Shalahudin: 2013)

Sebuah *activity diagram* mempunyai :

- a. *Start point (initial node)*, menggambarkan awal dari aktivitas.



**Gambar 2.3** Titik Awal *Activity*

- b. *End point (activity final node)*, menggambarkan akhir dari aktivitas.



**Gambar 2.4** Titik Akhir *Activity*

- c. *Activities*, menggambarkan proses bisnis dan dikenal sebagai *activity state*.

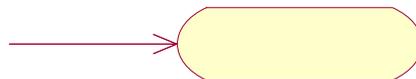


**Gambar 2.5** Simbol *Activity*

Jenis-jenis *activities*, yaitu :

- 1) *Black hole activities*

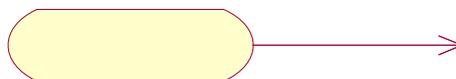
Ada masukan dan tidak ada keluaran, biasanya digunakan jika dikehendaki ada 1 atau lebih transisi.



**Gambar 2.6** Simbol *Black Hole Activities*

- 2) *Miracle activities*

Tidak ada masukan dan keluaran, biasanya dipakai pada waktu *start point* dan dikehendaki ada 1 atau lebih transisi.



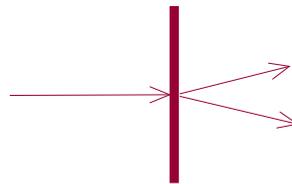
### Gambar 2.7 Simbol *Miracle Activities*

#### 3) *Parallel activities*

Suatu *activity* yang berjalan secara bersamaan yang terdiri dari:

##### (a) *Fork* (Percabangan)

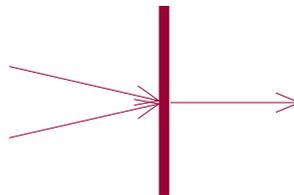
Mempunyai 1 transisi masuk dan 2 atau lebih transisi keluar. Ketika ada  $> 1$  transisi masuk ke *fork* yang sama, gabungkan dengan sebuah *decision point*.



Gambar 2.8 Simbol *Fork* dalam UML

##### (b) *Join* (Gabungan)

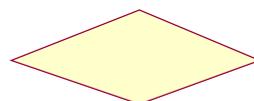
Mempunyai 2 atau lebih transisi masuk dan hanya 1 transisi keluar, dan *fork* harus berhubungan dengan *join*.



Gambar 2.9 Simbol *Join* dalam UML

##### (c) *Decision point*

Digambarkan dengan lambang wajik atau belah ketupat. Mempunyai transisi (sebuah garis dari/ke *decision point*). Setiap transisi yang ada harus mempunyai *GUARD* (kunci). Tidak ada sebuah keterangan (pertanyaan) pada tengah belah ketupat seperti pada *flowchart*.



### Gambar 2.10 Simbol *Decision Point*

(d) *Guards* (Kunci)

*Guards* (Kunci) adalah sebuah kondisi benar sewaktu melewati sebuah transisi. Digambarkan dengan diletakkan diantara tanda [ ]. Tanda [*otherwise*] *guard* untuk menangkap suatu kondisi yang belum terdeteksi. Setiap transisi dari/ke *decision points* harus mempunyai *guard* yang harus konsisten dan lengkap serta tidak *overlap*.

Contoh :

$x < 0, x = 0$ dan $x > 0$	konsisten
$x \leq 0$ dan $x \geq 0$	tidak konsisten
$x < 0$ dan $x > 0$	bagaimana jika $x = 0$

(e) *Swimlane*

Sebuah cara untuk mengelompokkan *activity* berdasarkan *actor* (mengelompokkan *activity* dalam sebuah urutan yang sama). Aktor bisa ditulis nama aktor ataupun sekaligus dengan lambang aktor (*stick figure*) pada *usecase diagram*. *Swimlane* digambarkan secara vertikal, walaupun kadang-kadang digambarkan secara horizontal.

(f) *Swimarea*

*Swimarea* mengelompokkan *activity* berdasarkan kegiatan didalam *usecase*.

## 2.8 Database (Basis Data)

Basis Data terdiri dari 2 kata yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data dan arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data atau arsip.

Satu hal yang juga harus diperhatikan, bahwa basis data bukan hanya sekedar penyimpanan data secara elektronik (dengan bantuan komputer). Artinya, tidak semua bentuk penyimpanan data secara elektronik bisa disebut basis data.

Basis Data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa *mengatap* satu sama lain atau tidak perlu suatu kerangkapan data (kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol [*controlled redundancy*]), data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali; data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal; data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya; data disimpan sedemikian rupa sehingga proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

### **2.8.1 ERD (Entity Relationship Diagram)**

Menurut Jogiyanto, *ERD (Entity Relationship Diagram)* adalah suatu komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang mempresentasikan seluruh fakta dari dunia nyata yang ditinjau. Model data sendiri merupakan sekumpulan cara, peralatan untuk

mendiskripsikan data, yang hubungan satu sama lain serta batasan konsistensi. Model data terdiri dari model hubungan entitas dan model relasional.

*Entity Relationship Diagram* disebut juga dengan diagram hubungan entitas dan digunakan untuk menganalisa data dan menggambarkan hubungan antara data yang ada dalam diagram arus data. *ERD* menggambarkan hubungan antara data yang ada dan tidak menggambarkan proses-proses yang terjadi.

Komponen-komponen *ERD* adalah

a. *Entity* atau Entitas

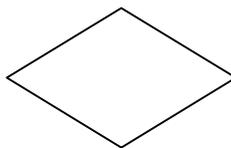
*Entity* atau entitas itu sendiri adalah objek yang dapat dibedakan dalam dunia nyata. Entitas biasanya didokumentasikan dengan simbol persegi panjang.



**Gambar 2.11** Simbol *Entitas (entity)*

b. Relasi

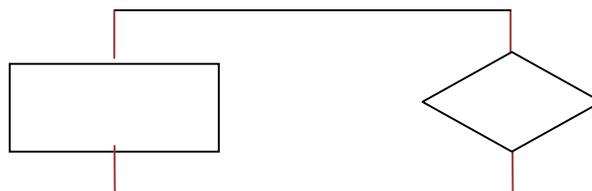
Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas. Relasi digambarkan dengan bentuk belah ketupat. Tiap belah ketupat diberi label kata kerja.



**Gambar 2.12** Simbol Relasi

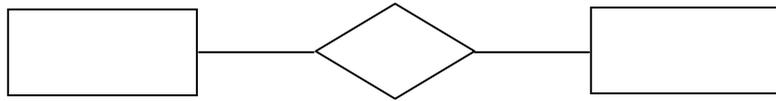
Adapun jenis atau varian dari Relasi dalam *ERD* adalah sebagai berikut :

- 1) *Unary Relationship* maksudnya adalah satu relasi dihubungkan oleh satu entitas.



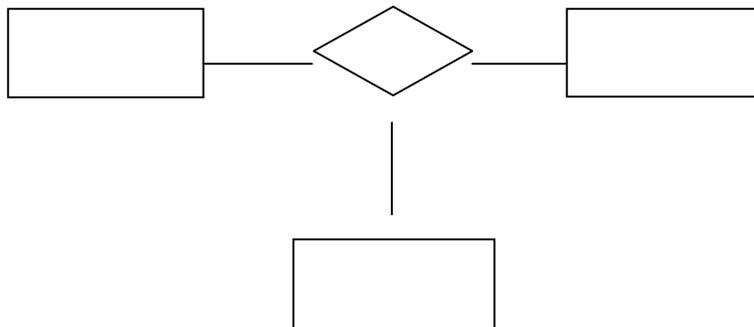
**Gambar 2.13** contoh *ERD* dalam bentuk *Unary Relationship*

- 2) *Binary Relationship* maksudnya adalah satu relasi dihubungkan oleh dua entitas.



**Gambar 2.14** Contoh ERD dalam bentuk *Binary Relationship*

- 3) *N-ary Relationship* maksudnya adalah satu relasi dihubungkan oleh tiga entitas



**Gambar 2.15** Contoh ERD dalam bentuk *N-ary Relationship*

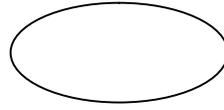
c. Atribut

Atribut adalah karakteristik dari *entity* atau relasi, yang menyediakan penjelasan detail tentang *entity* atau relasi tersebut, nilainya jarang berubah atau nilai tunggal, yang disebut nilai atribut digambarkan dalam bentuk *elips*.

Nilai atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut dalam suatu *entity* atau relasi. Terdapat 2 jenis atribut yaitu :

- 1) *Identifier (Key)*, menentukan *entity* secara unik.
- 2) *Descriptor (Non Key Atribut)*, menentukan *entity* yang tidak unik.

Sebutan lain dari atribut adalah properti, elemen data dan *field*. *Foreign Key* (Kunci Tamu). Suatu atribut dalam suatu *entity* yang menunjukkan ke atribut *primary key* dari entitas lain.



**Gambar 2.16** Simbol Atribut

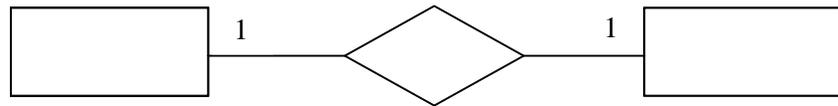
3) *Cardinality*

*Cardinality* adalah batasan yang menjelaskan jumlah keterhubungan satu *entity* dengan *entity* lainnya.

3Tingkat hubungan atau derajat relasi pada *cardinality* yaitu :

(a) *One to One* [ 1 : 1 ]

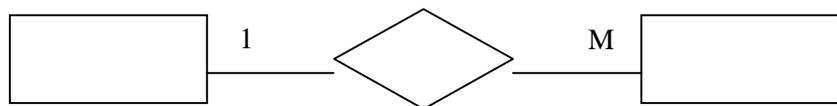
Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu.



**Gambar 2.17** Simbol *One to One*

(b) *One to Many* [ 1 : M ] / [ M : 1 ]

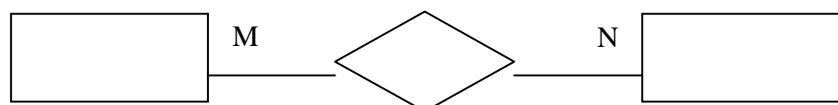
Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu.



**Gambar 2.18** Simbol *One to Many*

(c) *Many to Many* [ M : N ]

Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak.



**Gambar 2.19** Simbol *Many to Many*

### **2.8.2 Transformasi ERD ke LRS**

Transformasi diagram hubungan entitas ke dalam *Logical Record Structure* merupakan kegiatan untuk membentuk data-data ER diagram ke dalam LRS. Pada sebuah ER diagram nama *field* ditulis di luar kotak (diluar *entity*), sedangkan pada sebuah LRS nama *field* ditulis di dalam kotak dan memiliki sebuah nama unit. Aturan di atas berlaku, saat dipengaruhi elemen yang menjadi titik perhatian

### **2.8.3 Logical Record Structure (LRS)**

*Logical Record Structure* (LRS) digambarkan oleh kotak persegi panjang dan dengan nama yang unik. *File record* pada LRS ditempatkan dalam kotak. LRS terdiri dari *link-link* diantara tipe *record* lainnya, banyaknya *link* dari LRS yang diberi nama oleh *field-field* yang kelihatan pada kedua *link* tipe *record*.

Penggambaran LRS dimulai dengan bagaimana memperoleh model yang dimengerti. Metode yang digunakan dimulai dengan hubungan model data yang dapat dikonversikan ke LRS. Tiap satu LRS akan menjadi satu tabel, nama LRS akan menjadi nama tabel, tiap satu atribut akan menjadi satu kolom dan nama atribut akan menjadi nama kolom.

### **2.8.4 Tabel/Relasi**

Tabel merupakan hasil transformasi *Logical Record Structure* ke relasi. Tiap satu LRS akan menjadi satu tabel. Nama LRS dapat menjadi nama tabel. Tiap satu atribut akan menjadi satu kolom dan nama atribut akan menjadi nama kolom.

### **2.8.5 Spesifikasi Basis Data**

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lain dan disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, serta digunakan perangkat lunak (*software*) tertentu untuk memanipulasi untuk kegunaan tertentu.

Sedangkan spesifikasi basis data adalah suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer dengan tujuan untuk

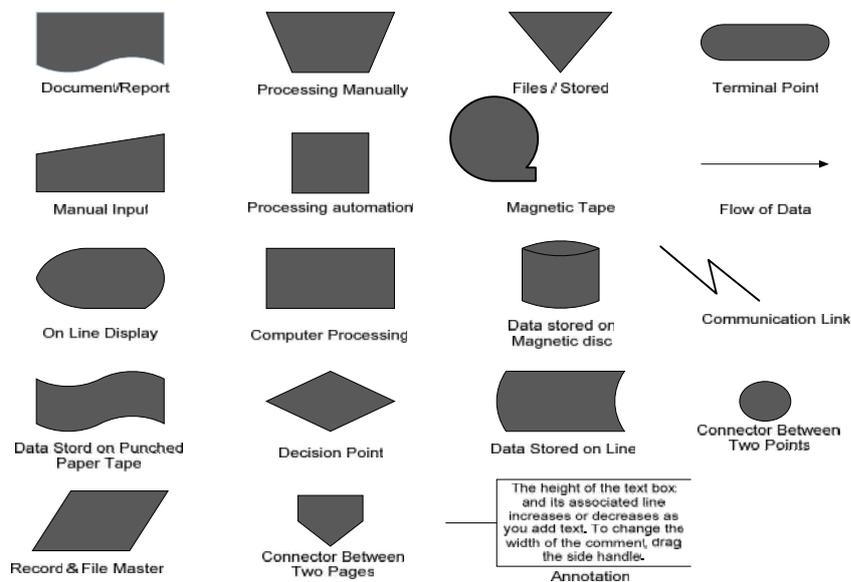
menyimpan atau merekam serta melihat data operasional lengkap pada sebuah organisasi, sehingga mampu menyediakan informasi yang diperlukan untuk kepentingan proses pengambilan keputusan.

## 2.9 Flowchart

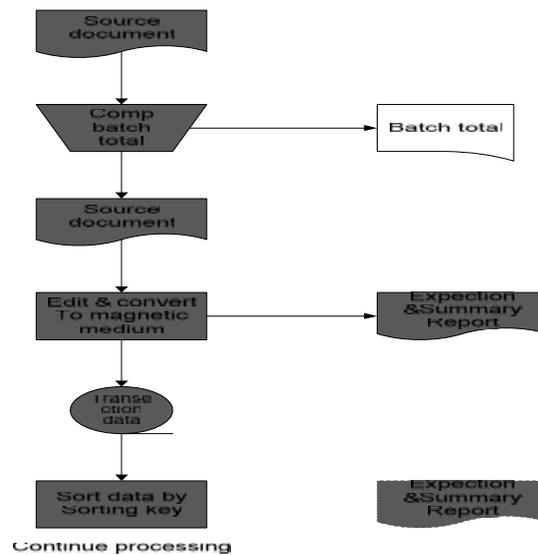
Menurut Jogiyanto, *Flowchart* adalah bagan (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur sistem secara logika. *Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah, khususnya yang perlu dipelajari dan dievaluasi lebih lanjut.

Jenis-jenis *flowchart* terdiri atas :

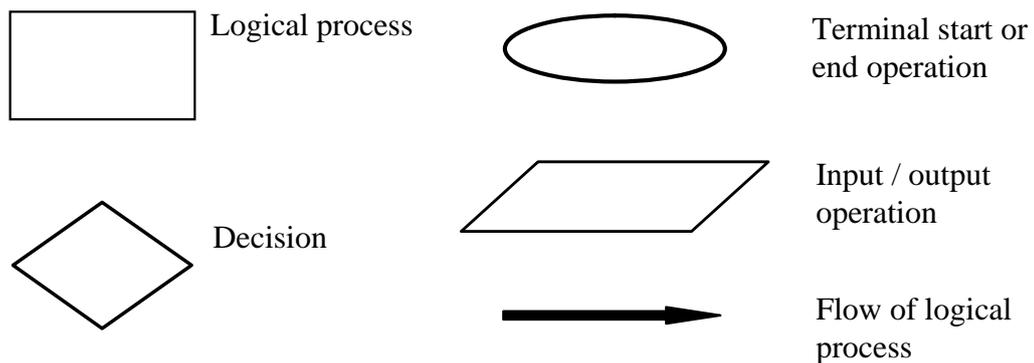
- a. *System flowchart*
- b. *Document flowchart*
- c. *Schematic flowchart*
- d. *Program flowchart*
- e. *Proses flowchart*



**Gambar 2.20** Notasi *System Flowchart*



**Gambar 2.21** Guide line for System Flowchart



**Gambar 2.22** Notasi Program Flowchart

### 2.10 My Structure Query Language (MySQL)

MySQL adalah suatu sistem manajemen database. Suatu *database* adalah suatu koleksi data terstruktur. Data dapat berupa apa saja. Untuk menambah, mengakses dan memproses data yang tersimpan dalam sebuah *database* dibutuhkan suatu sistem manajemen *database* seperti MySQL.

MySQL adalah suatu sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam tabel yang terpisah. Tabel-tabel tersebut terhubung oleh suatu relasi yang memungkinkan memperoleh kombinasi data dari beberapa tabel dalam suatu permintaan.

*SQL* adalah bahasa *standart* yang digunakan untuk mengakses *database* yang bersifat *open source*. *Open source* berarti dapat digunakan dan dimodifikasi oleh siapapun. Semua orang dapat *download* dari internet dan menggunakan secara gratis.

a. Kelebihan *MySQL*

Kelebihan *database* ini adalah :

- 1) Digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
- 2) Memiliki kecepatan yang bagus dalam menangani *query* sederhana.
- 3) Memiliki operator dan fungsi secara penuh dan mendukung perintah *select* dan *where* dalam perintah *query*.
- 4) Memiliki keamanan yang bagus karena beberapa lapisan sekuritis dan seperti level *subnetmaks*, nama *host*, dan izin akses *user* dengan sistem perijinan yang *mendetail* serta sandi terenkripsi.
- 5) Mampu menangani basisdata dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60000 tabel serta kurang lebih 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- 6) Melakukan koneksi dengan *client* menggunakan *Protocol TCP/IP*, *unix socet (UNIX)*, atau *named tipes (ENTE)*.
- 7) Dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.
- 8) Dapat berjalan stabil pada berbagai sistem operasi seperti *windows*, *LINUX*, *freeBSD*, *Mac OS X Server*, *Solaris*, *Amiga*, dan sebagainya.
- 9) Didistribusikan secara *open source*, dibawah lisensi *GPL* sehingga dapat digunakan secara gratis.

b. Fungsi *MySQL*

Beberapa contoh fungsi *MySQL* yang digunakan adalah sebagai berikut:

- 1) *MySQL\_connect()* untuk membuka koneksi dengan *server MySQL*.

- 2) *MySQL\_pconnect()* untuk membuka koneksi *persisten* dengan *server MySQL*.
- 3) *MySQL\_select\_db()* untuk memilih *database MySQL*.
- 4) *MySQL\_query()* untuk mengirim *query* ke *server MySQL*.
- 5) *MySQL\_db\_query()* untuk mengirim *query* ke *server MySQL*.
- 6) *MySQL\_num\_rows()* untuk mengambil banyak baris yang diperoleh operasi *SELECT*.
- 7) *MySQL\_fetch\_array()* untuk mengambil hasil perbaris sebagai *array* asosiatif, *numeric*, atau keduanya.
- 8) *MySQL\_fetch\_assoc()* untuk mengambil hasil terbaris sebagai *array* asosiatif saja.
- 9) *MySQL\_insert\_id()* untuk mengambil nomor *auto\_increment* perintah *INSERT* terakhir.
- 10) *MySQL\_affected\_rows()* untuk mengambil jumlah baris yang terpengaruh operasi *MySQL* sebelumnya.
- 11) *MySQL\_errno()* untuk mengambil nomor eror dari operasi *MySQL* sebelumnya.

c. Tipe data yang diterima *MySQL*

*MySQL* menyediakan berbagai tipe data dengan spesifikasi dan panjang masing-masing. Tipe data untuk dalam *MySQL* diantaranya ditampilkan pada tabel berikut :

**Tabel 2.3** Tipe data yang diterima *MySQL*

Jenis Tipe	Tipe	Keterangan
	TINYINT	-128 s/d 127 SIGNED 0 s/d 255 UNSIGNED
	SMALLINT	-32768 s/d 32767 SIGNED 0 s/d 65525 UNSIGNED
	MEDIUMINT	-8388608 s/d 8388607 SIGNED 0 s/d 16777215 UNSIGNED
	INT	-2147483648 s/d 2147483647

		SIGNED 0 s/d 4294967295 UNSIGNED
	BIGINT	9223372036854775808 s/d 9223372036854775807 SIGNED 0 s/d 18446744073709551615 UNSIGNED
	FLOAT	Bilangan pecahan presisi TUNGGAL
	DOUBLE	Bilangan pecahan presisi ganda
	DECIMAL	Bilangan dengan decimal
DATE/TIME	DATE	Tanggal dengan format YYYY-MM-DD
	DATETIME	Tanggal dan waktu dengan format YYYY-MM-DD HH : MM : SS
	TIMESTAMP	Tanggal dan waktu dengan format YYYYMMDDHHMMSS
	TIME	Waktu dengan format HH : MM : SS
	YEAR	Tahun dengan format YYYY
STRING	CHAR	0-255 karakter
	VARCHAR	0-255 karakter
	TINYTEXT	String dengan panjang maksimum 255 karakter
	TEXT	String dengan panjang maksimum 65535 karakter
	BLOB	String dengan panjang maksimum 65535 karakter
	MEDIUMTEXT	String dengan panjang maksimum 16777215 karakter
	MEDIUMBLOB	String dengan panjang maksimum 16777215 karakter

	LONGTEXT	String dengan panjang maksimum 4294967295 karakter
	LONGBLOB	String dengan panjang maksimum 4294967295 karakter
KHUSUS	ENUM	Tipe data dengan isi tertentu
	SET	Tipe data dengan isi tertentu

## 2.11 Hypertext Preprocessor (PHP)

*Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memrogram situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS maupun *framework*.

### a. Sejarah PHP

Pada awalnya *PHP* merupakan kependekan dari *personal home page* atau (situs personal). *PHP* pertama kali dibuat oleh *Rasmus Lerdorf* pada tahun 1995. Pada waktu itu *PHP* masih bernama *Form Interpreted (FI)*, yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya *Rasmus* menulis kode sumber tersebut untuk umum dan menamakannya *PHP/FI*. Dengan perilisannya kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan *PHP*.

Pada november 1997, dirilis *PHP/FI 2.0*. pada rilis ini, *interpreter PHP* sudah diimplementasikan pada program *c*. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan *PHP/FI* secara signifikan.

Pada november 1997, sebuah perusahaan bernama *ZEND* menulis ulang *interpreter PHP* menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada juni 1998, perusahaan tersebut merilis *interpreter* baru untuk *PHP* dan meresmikan rilis tersebut sebagai *PHP 3.0* dan singkatan *PHP* diubah menjadi Akronim berulang *PHP: Hypertext Preprocessing*.

Pada pertengahan tahun 1999, ZEND merilis *interpreter PHP* baru dan rilis tersebut dikenal dengan *PHP 4.0*. *PHP 4.0* adalah versi *PHP* yang paling banyak dipakai pada awal abad ke 21. Versi ini banyak dipakai disebabkan oleh kemampuannya untuk membangun aplikasi *web* kompleks tetapi memiliki kecepatan dan stabilitas yang tinggi.

Pada juni 2004, ZEND merilis *PHP 5.0*. dalam versi ini, inti dari *interpreter PHP* mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek kedalam *PHP* untuk menjawab perkembangan bahasa pemrograman kearah paradigma berorientasi objek.

b. Tipe Data *PHP*

Tipe data adalah jenis-jenis data yang dikelompokkan berdasarkan kriteria dan watak tertentu. *PHP* memiliki 8 tipe data :

1) *Integer*

Tipe ini merupakan tipe bilangan bulat. Contoh bilangan bulat: 1,0,-1,2,100. Untuk menyatakan suatu nilai bilangan bulat kita bisa menggunakan format desimal, maupun hexsa desimal.

2) *Float*

Tipe ini merupakan tipe bilangan *real*. Pada bahasa pemrograman seperti *C/C++*, tipe *float* terdiri dari *single*, *double*. Pada *PHP* jangkauan *float* juga tergantung pada *platform* yang digunakan, dimana maksimum bisa mencapai  $1.8 \times 10^{308}$  dengan presisi 14 digit(64 *bits*).

3) *Boolean*

Tipe ini memiliki dua nilai yaitu *True* dan *False*. Kedua nilai ini dapat ditulis tanpa memperhatikan huruf besar maupun kecil. Tipe ini biasanya digunakan untuk menguji suatu ekspresi bersyarat. Adapun nilai *TRUE* dan *FALSE* dapat dinyatakan dengan suatu nilai pada tipe data lainnya.

4) *String*

*String* merupakan kumpulan karakter. Pada *PHP* karakter sama dengan tipe bilangan *Byte* yang mana memiliki 256 karakter berbeda. Panjang dari *string* yang dapat dibuat dibatasi oleh ukuran memori komputer

kita. Jadi panjang *string* tak terbatas sepanjang memori kita mencukupi. Penulisan pada string *PHP* dapat diaktif dengan tanda petik tunggal (') atau petik ganda("").

5) *Object*

*Object* adalah suatu tipe data yang dapat berupa sebuah bilangan, variabel atau bahkan sebuah fungsi.

6) *Array*

*Array* dalam *PHP* dapat berupa *indeksed array (vector)* dan *hash tables (assosiative array)*. *PHP* juga mendukung *array* multidimensi. *Array* adalah sebuah data yang mengandung satu atau lebih data, dan dapat diindeks berdasarkan numerik maupun *string*. Pada *PHP* tipe masing-masing data variabel *array* tidak harus sama dan variabel *array* juga bisa menyimpan suatu variabel *array* didalamnya. *Array* dapat dibuat dengan menggunakan fungsi *array () list()*.

7) *Null*

Menyatakan bahwa suatu variabel tidak memiliki nilai. *NULL* hanya merupakan nilai mungkin dari tipe *NULL* yang telah diperkenalkan pada *PHP* 4 dan *keyword NULL* adalah *case* sensitif.

8) *Resource*

Tipe data baru yang diperkenalkan mulai *PHP* 4. Tipe ini memiliki nilai yang dihasilkan dari pemanggilan fungsi-fungsi yang menggunakan *resource* sistem, seperti *mysql\_connect*. Variabelnya secara otomatis akan menggunakan tipe data *resource* ini.

c. Variabel dalam *PHP*

Variabel adalah tempat untuk menyimpan data. Data yang disimpan sewaktu-waktu bisa saja dipanggil atau diganti dengan data lain. Mendefinisikan variabel cukup dengan kata (akan menjadi nama variabel) yang diawali dengan tanda \$, kemudian isikan nilai variabel tersebut.

Contoh :

\$bulan = "Mei"; atau \$nomor = 2011;

Penulisan variabel ada aturannya, yaitu : gunakan apa saja yang didahului oleh huruf dan selalu diberi tanda \$ di depan kata tersebut. Contoh : \$bulan.

## **2.12 Metode Waterfall**

Rosa A.S dan M. Shalahuddin (2013:28) menyatakan, bahwa model *Waterfall* sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (support).

### **2.12.1 Pengumpulan Data**

Rosa A.S dan M. Shalahuddin (2013:19) menyatakan, bahwa hal pertama yang dilakukan dalam analisis sitem adalah melakukan pengumpulan data. Ada beberapa teknik pengumpulan data yang sering dilakukan yaitu sebagai berikut.

#### **a. Teknik Wawancara**

Pengumpulan data dengan menggunakan wawancara mempunyai beberapa keuntungan sebagai berikut.

- 1) Lebih mudah dalam menggali bagian sistem mana yang dianggap baik dan bagian mana yang dianggap kurang baik.
- 2) Jika ada bagian tertentu yang menurut anda perlu untuk digali lebih dalam, anda dapat langsung menanyakan kepada narasumber.
- 3) Dapat menggali kebutuhan user secara lebih bebas.
- 4) User dapat mengungkapkan kebutuhannya secara lebih bebas.

#### **b. Teknik Observasi**

Pengumpulan data dengan menggunakan observasi mempunyai keuntungan yaitu:

- 1) Analis dapat melihat langsung bagaimana sistem lama berjalan
- 2) Mampu menghasilkan gambaran lebih baik jika dibanding dengan teknik lainnya.

c. Dokumentasi

Dokumentasi adalah dengan cara mencari data yang berupa catatan, transkrip, buku, surat kabar, majalah prasasti, notulen rapat, legger, agenda dan sebagainya. Hadari Nawawi menyatakan bahwa studi dokumentasi adalah cara pengumpulan data melalui peninggalan tertulis terutama berupa arsiparsip dan termasuk juga buku mengenai pendapat, dalil yang berhubungan dengan masalah penyelidikan.

d. Studi Literatur/Pustaka

Studi literatur dalam sebuah penelitian untuk mendapatkan gambaran yang menyeluruh tentang apa yang sudah dikerjakan orang lain dan bagaimana orang mengerjakannya, kemudian seberapa berbeda penelitian yang akan kita lakukan. Penting karena untuk menghindari usaha yang sebenarnya sudah pernah dilakukan orang lain dan bisa digunakan pada penelitian kita untuk menghemat waktu, tenaga dan biaya. Penting juga untuk memberi arah penelitian selanjutnya yang perlu dilakukan untuk melanjutkan misi penelitian. Kemampuan peneliti untuk menyusun kerangka teoritis akan sangat terkait dengan upaya penelusuran studi kepustakaan, sebagai upaya memperoleh sejumlah referensi yang mendukung dan tepat untuk membahas lingkup kajian penelitian yang dilakukan. Selanjutnya kerangka teoritis yang disusun akan bermanfaat pada saat peneliti menentukan hipotesis penelitian. Setelah seorang peneliti telah menetapkan topik penelitian, langkah selanjutnya adalah melakukan kajian yang berkaitan dengan: teori yang berkaitan dengan topik penelitian. Dalam pencarian teori, peneliti akan mengumpulkan informasi sebanyak-banyaknya dari kepustakaan yang berhubungan. Sumber-sumber kepustakaan dapat diperoleh dari buku, jurnal, majalah, hasil-hasil penelitian (tesis dan disertasi), dan sumber-sumber lainnya yang sesuai (internet, koran dll). Keseluruhan upaya tersebut, dikatakan sebagai upaya Studi Kepustakaan untuk penelitian.

### 2.13 *Web Service*

*Web service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

*Web Service* dapat diartikan juga sebuah metode pertukaran data tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di platform apa sebuah data itu dikonsumsi.

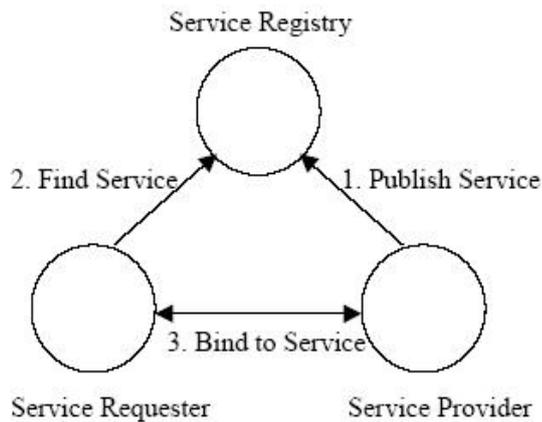
*Web service* bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

- a. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
- b. *Web service* memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
- c. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi firewall.

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

- a. Service Requester (peminta layanan)
- b. Service Provider (penyedia layanan)
- c. Service Registry (daftar layanan)



- Service Provider: Berfungsi untuk menyediakan layanan/service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
- Service Registry: Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.
- Service Requestor: Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

Secara umum, web service memiliki tiga operasi yang terlibat di dalamnya, yaitu:

- a. Publish/Unpublish: Menerbitkan/menghapus layanan ke dalam atau dari registry.
- b. Find: Service requestor mencari dan menemukan layanan yang dibutuhkan.
- c. Bind: Service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke service provider untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh service provider.