BAB II
LANDASAN TEORI

2.1 Sistem

Istilah sistem berasal dari bahasa Yunani, yaitu *sistema* yang berarti keseluruhan yang terdiri dari bermacam-macam bagian. Secara umum sistem didefinisikan sebagai sekumpulan elemen-elemen yang saling berinteraksi untuk mencapai tujuan tertentu di dalam lingkungan yang kompleks.

Di dalam definisi sistem juga terdapat dua kelompok yang mendefinisikannya, yaitu menekankan pada prosedur dan yang menekankan pada komponen atau elemen. Menurut Hartono, sistem merupakan jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan sesuatu kegiatan atau menyelesaikan suatu sasaran tertentu" [2].

Pendekatan sistem yang merupakan jaringan kerja di prosedur lebih ditekankan pada urutan-urutan operasi di dalam suatu sistem. "Suatu prosedur adalah suatu urut-urutan operasi klerikal (tulis menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang ditetapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi"[2].

Sedangkan pendekatan sistem yang menekankan pada elemen atau komponen, adalah adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu. Pendekatan sistem yang merupakan kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang lebih luas dan lebih banyak diterima karena pada kenyataannya

suatu sistem terdiri dari beberapa subsitem atau sistem-sistem bagian. Komponen-komponen atau subsistem-subsistem dalam suatu sistem tidak dapat berdiri sendiri, semuanya saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga sasaran sistem dapat tercapai.

Sistem juga merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sana untuk memroses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) seperti yang diinginkan[8].

2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen (*components*), batas (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*), dan sasaran (*objectives*) atau tujuan (*goal*).[2]

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari subsistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar disebut dengan supra sistem.

b. Batas sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem

ini memungkinkan suatu sistem dipasang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan luar sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran (*output*) dari satu subsistem akan menjadi satu masukan (*input*) bagi subsistem yang lain dan akan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Masukan sistem

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan peralatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintanance in*put adalah energi yang diproses agar didapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputer, sedangkan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

g. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objektif*). Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali, masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Karakteristik suatu sistem dapat dilihat dari gambar 2.1.

2.1.2 Klasifikasi Sistem

Dari berbagai sudut pandang, sistem dapat diklasifikasikan menjadi beberapa bagian, yaitu: [2,8]

1. Sistem abstrak dan sistem fisik

   Sistem abstrak merupakan sistem yang tidak bisa dilihat secara mata biasa dan biasanya sistem ini berupa pemikiran atau ide-ide. Contohnya adalah filsafat. Sistem fisik merupakan sistem yang bisa dilihat secara mata biasa dan biasanya sering digunakan oleh manusia. Contohnya adalah sistem akuntansi dan sistem komputer.

2. Sistem alamiah dan sistem buatan

   Sistem alamiah merupakan sistem yang terjadi karena pengaruh alam. Misalnya sistem rotasi bumi dan sistem gravitasi. Sistem buatan merupakan

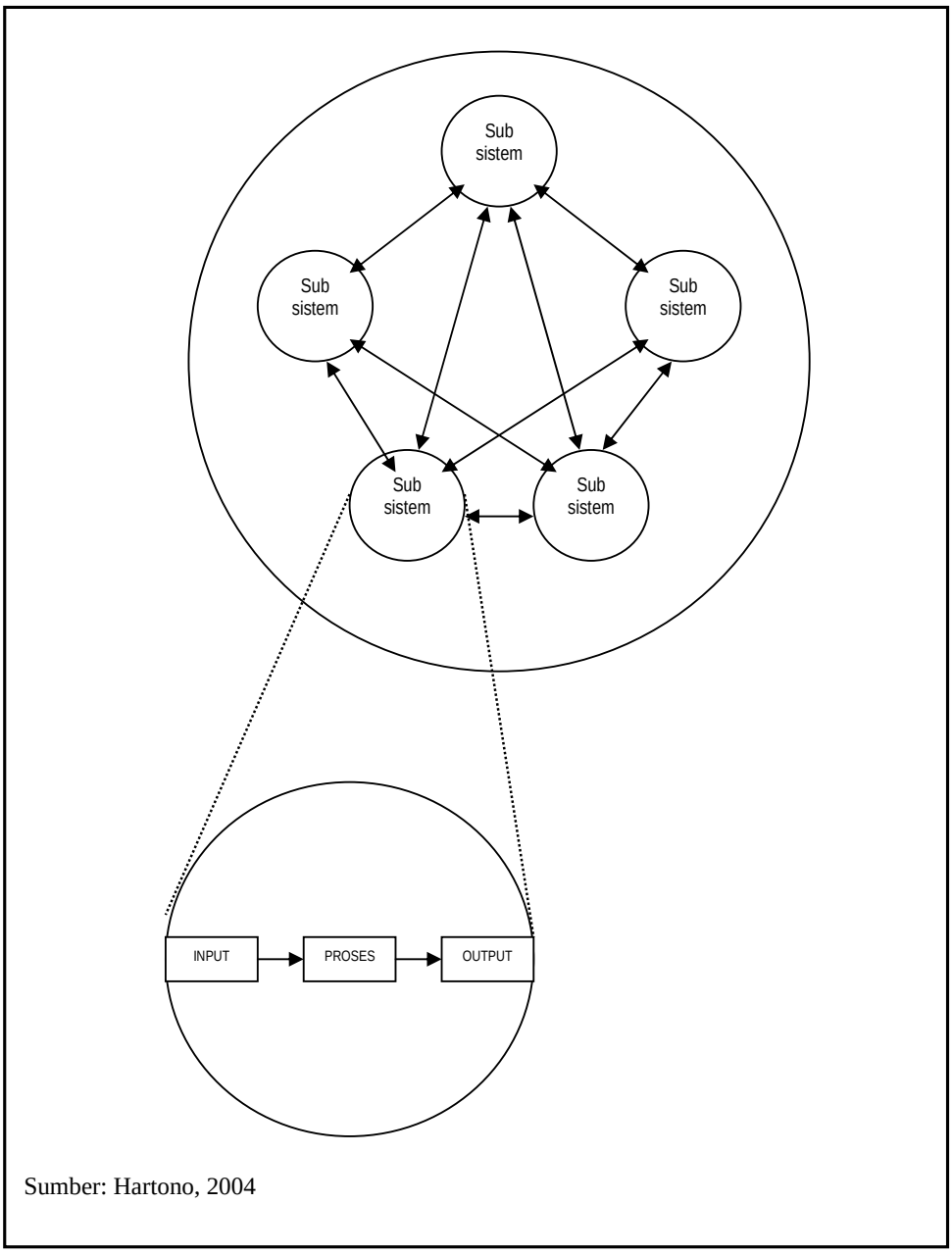
sistem yang dirancang dan dibuat oleh manusia, misalnya sistem pengolahan gaji.

3. Sistem tertutup dan sistem terbuka

   Sistem tertutup merupakan sistem yang tidak berhubungan dengan bagian luar sistem dan biasanya tidak terpengaruh oleh kondisi di luar sistem. Sedangkan sistem terbuka merupakan sistem yang berhubungan dengan bagian luar sistem.

4. Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*).

Sistem tertentu beroperasi tertentu dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.



Sumber: Hartono, 2004

Gambar 2.1. Karakteristik Suatu Sistem

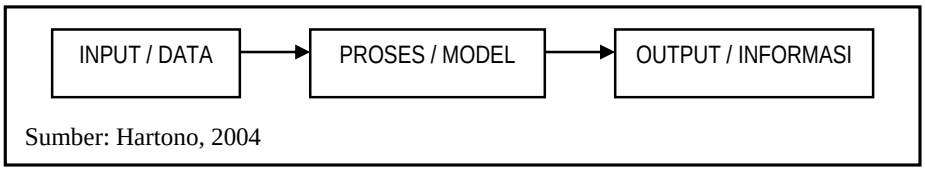
2.2 Informasi

Informasi pada saat ini sangatlah penting dan dibutuhkan sekali oleh manusia. Menurut Hartono, informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya [2].

Biasanya dalam pemakaian istilah data dan informasi sering tertukar, namun terdapat perbedaan yang mendasar antar keduanya, yaitu data adalah bahan baku yang diolah untuk mendapatkan informasi sedangkan informasi merupakan acuan dalam pengambilan suatu keputusan yang berarti derajat informasi lebih tinggi dari data.

Masih menurut Hartono, informasi dapat didefinisikan sebagai hasil dari pengolahan suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata yang digunakan untuk pengambilan keputusan.

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal atau data-item. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata.



Sumber: Hartono, 2004

Gambar 2.2. Model Pengolahan Data

Informasi yang baik harus memiliki karakteristik sebagai berikut[9]:

1. *Pertinent* atau relevan. Informasi harus sesuai dengan apa yang dibutuhkan, dengan kata lain harus mempunyai manfaat bagi pemakainya.
2. *Timely* atau tepat waktu. Informasi harus tersedia ketika dibutuhkan.
3. *Accurate* atau akurat. Informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan.

Selain ketiga hal di atas, Kroenke juga menambahkan bahwa informasi yang baik juga harus mampu mengurangi ketidakpastian (*reduces uncertainty*) dan informasi harus berisi *element of surprise.*

2.3 Pariwisata

Menurut *World Tourism Organization*, pariwisata didefinisikan sebagai rangkaian kegiatan perjalanan seseorang dan tinggal di luar daerah asalnya untuk kepentingan *leisure*, bisnis ataupun tujuan lainnya. Sedangkan menurut Undang-undang No. 9 tahun 1990 tentang kepariwisataan, pariwisata adalah segala sesuatu yang berhubungan dengan wisata termasuk pengusahaan objek dan daya tarik wisata, usaha sarana pariwisata, dan usaha lain di bidang tersebut.

Pariwisata dapat didefinisikan sebagai jumlah dari sebuah fenomena dan hubungan yang muncul dari interaksi antara wisatawan, usaha pariwisata, pemerintah, dan masyarakat lokal dalam proses untuk menarik dan melayani wisatawan tersebut serta pengunjung lainnya. Pariwisata merupakan gabungan dari aktifitas, pelayanan, dan industri yang memberikan pengalaman berwisata [12].

Pariwisata memiliki komponen-komponen sebagai berikut[5]:

1. *Tourist attraction and activities*. Segala sesuatu seperti alam, budaya, dan bentukan khusus dan berhubungan dengan aktifitas pada suatu area yang dapat menarik wisatawan untuk mengunjunginya.
2. *Accommodation* atau akomodasi. Sarana akomodasi seperti hotel dan jenis fasilitas akomodasi lainnya.
3. *Other tourists' facilities and services*. Merupakan fasilitas dan jasa yang diperlukan atau yang berhubungan degan pariwisata, termasuk di dalamnya adalah operator perjalanan, restoran, tempat cinderamata, pusat kerajinan, bank, dan fasilitas lainnya yang berhubungan.
4. Fasilitas transportasi dan jasa. Menggambarkan akses transportasi menuju suatu Negara, daerah, atau area pengembangan lainnya, jalur transportasi menuju dan ke daya tarik wisata. Termasuk di dalamnya adalah transportasi darat, laut, dan udara.
5. Infrastruktur. Termasuk persediaan air (*water supply*), jaringan listrik, jaringan komunikasi, drainase, dan yang lainnya.

2.4 Sistem Informasi Pariwisata

Sistem informasi dapat difenisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputuasan yang cerdik.[2]

Suatu sistem informasi juga dapat didefinisikan sebagai berikut[5]:

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi
3. Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Jadi, sistem informasi pariwisata merupakan sebuah sistem yang menyajikan informasi mengenai pariwisata dan sumber dayanya, termasuk di dalamnya adalah daya tarik wisata, sarana pendukung, dan *human actor* yang berhubungan dalam industri pariwisata.

2.5 Teknologi Informasi Berbasis Internet

Sistem Informasi berbasis web ini memiliki hubungan yang sangat erat dengan pengolahan informasi. Oleh karena itu, sistem ini perlu mendapat dukungan teknologi informasi yang memadai. Berikut akan dibahas mengenai beberapa macam teknologi informasi sebagai teknologi pendukung dalam sistem yang dirancang.

2.5.1 Internet dan TCP/IP

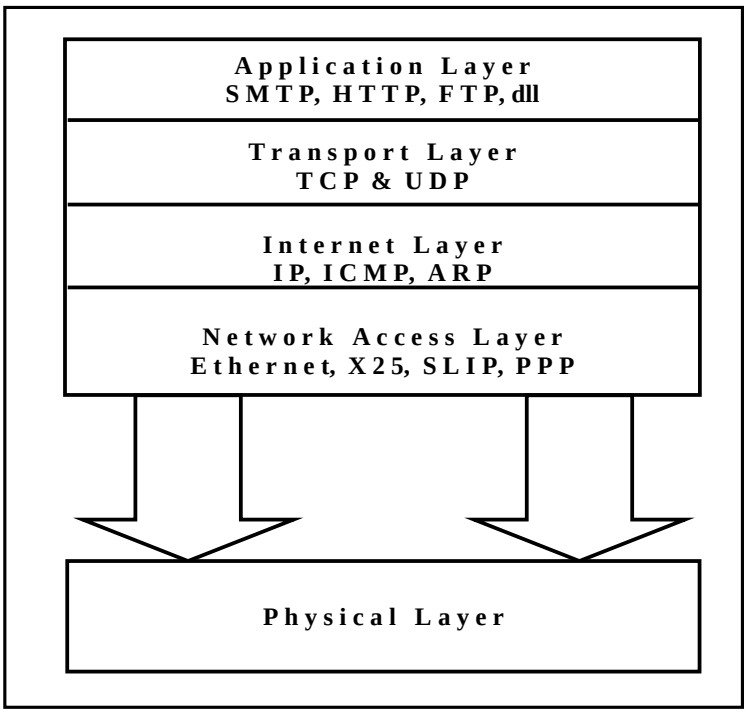
Internet adalah himpunan informasi dan sumber daya komputer yang paling besar di dunia. Internet berasal dari kata *Interconnection Networking* yang

mempunyai arti hubungan komputer dengan berbagai tipe dan sistem yang berbeda yang membentuk sistem jaringan yang mencakup seluruh dunia (jaringan komputer global) dengan menggunakan jalur telekomunikasi.

Pembahasan mengenai internet tidak dapat dipisahkan dari pembahasan tentang komunikasi data. Pada dasarnya komunikasi data merupakan proses memindahkan data dari satu komputer ke komputer yang lain. Komunikasi yang dibangun pada jaringan internet menggunakan mekanisme tertentu yang disebut dengan protokol.

Protokol yang digunakan untuk menjalankan komunikasi antar jaringan komputer pada internet ini adalah *Transmission Control Protocol*/Internet *Protocol* (TCP/IP). Dengan menggunakan protokol yang sama yaitu TCP/IP pada awal perkembangannya di gunakan oleh *Defense Advanced Research Project Agency* (DARPA) untuk membangun jaringan untuk keperluan militer. Nama TCP/IP itu sendiri diambil dari nama protokol pada *layer network* yaitu *Internet Protokol* (IP) dan protokol pada *layer transport* yaitu *Transmission Control Protokol* (TCP).

Dengan mengacu pada nama ini TCP/IP Internet kemudian sering disebut sebagai internet. Arsitektur protokol TCP/IP ini dapat dimodelkan dengan empat *layer* sebagaimana yang dapat dilihat pada Gambar 2.3



Gambar 2.3. Arsitektur protokol TCP/IP

TCP/IP terdiri atas empat lapis kumpulan protokol yang bertingkat. Empat tingkat itu adalah:

1. *Aplication Layer,* yaitu *layer* yang bertanggung jawab atas aplikasi - aplikasi yang menggunakan TCP/IP.
2. *Transport Layer*, yaitu *layer* yang bertanggung jawab memberikan layanan pengiriman data dari satu kode ke kode yang lain.
3. *Internet Layer*, yaitu *layer* yang menentukan jenis diagram dan menangani perutean data yang dikirim.
4. *Network Interface Layer*, yaitu *layer* yang bertanggung jawab mengirim dan menerima data pada media fisik transmisi yang digunakan.

2.5.2 Hypertext Transfer Protokol (HTTP)

HTTP merupakan sebuah protokol yang didesain untuk mentransfer informasi dalam bentuk hypermedia antara *server* dengan sebuah *client.* HTTP juga mentransfer data suatu informasi melalui *heade*r-nya. Header HTTP ini merupakan bentuk pengembangan dari *Multiproses Internet Mail Extention* (MIMEs). Pengembangan ini memungkinkan HTTP untuk mentransfer informasi dalam bentuk biner dan informasi dalam bentuk yang tidak standar yang berhasil dinegosiasi antara *server* dan *client*. Secara umum, *delay* akan terjadi pada saat *server* dan *client* melakukan suatu negosiasi sebelum proses transfer data. Karena lama dan *delay* yang di sebabkan oleh overhead akan menjadi relatif lebih besar pada transfer data yang sebetulnya singkat. HTTP merupakan protokol yang sederhana yang didesain untuk mereduksi *delay* yang bersifat *stateless*, sehingga *server* akan memproses setiap *request* dan *user* secara terpisah dan *request* yang lain. Independen terhadap *request* sebelumnya. HTTP menggunakan 8 bit untuk mentransfer semua tipe data yang mungkin.

Mekanisme yang terjadi pada HTTP bila suatu *client* menginginkan layanan dari *server* di bagi dalam 4 langkah yaitu:

1. *Connection Setup*.

   Client mengakses sebuah *server* dengan menggunakan internet *address* dan *port number*. Default dan port number adalah 8.0.

2. *Request.*

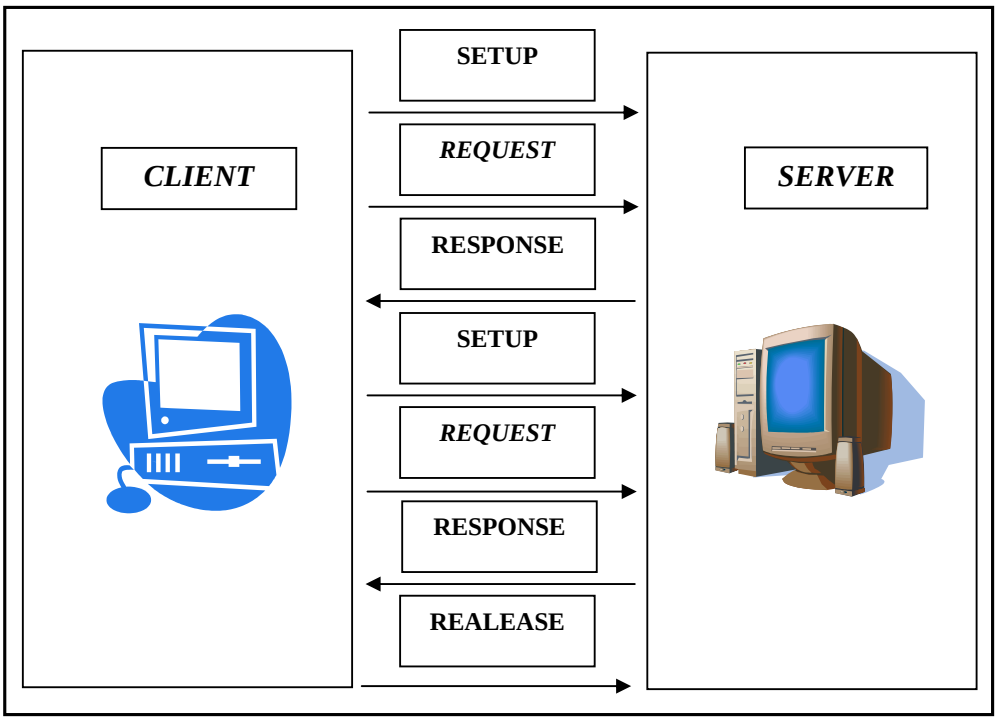
   Client mengirimkan *message* beserta informasi dan metode transaksi dan kapabilitas *client*.

3. *Response*.

   Server mengirimkan *response* kepada *client* sesudah *client* itu menyelesaikan *request*-nya. *Response message* meliputi informasi dan transaksi dan data yang diminta.

4. *Connection Release*

   Client mengakhiri koneksi dengan *server*.

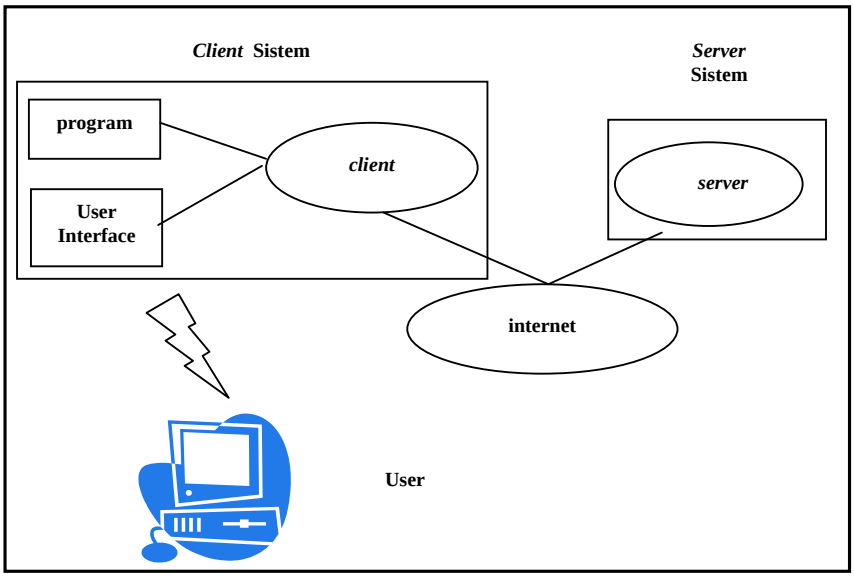


Gambar 2.4. Proses Koneksi protokol HTTP antara *client* dan *server*

2.5.3 Model *Client-Server*

Karakteristik yang mendasar dan proses yang terjadi di Internet yang telah diatur berdasarkan protokol standar adalah data satu objek yang meminta kepada objek yang lain untuk memberikan sebuah layanan. Dua objek tersebut dapat terletak dalam satu jaringan (*subnet*) ataupun terletak pada jaringan yang berbeda.

Objek yang meminta satu layanan disebut sebagai *client*. sedangkan objek yang memberikan layanan disebut *server*.



Gambar 2.5. Model *Client – Server* – Internet

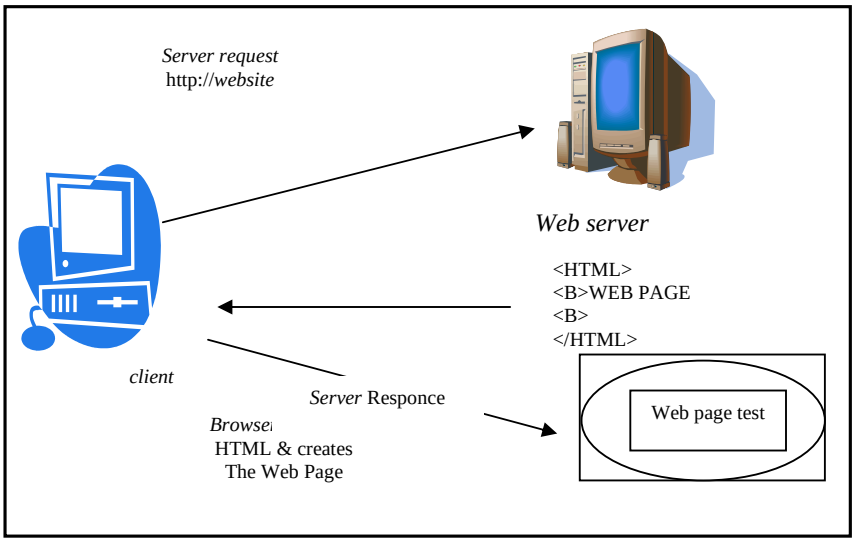
Untuk mengatur dan menampilkan halaman isi dapat diperindah dengan ditambahkan script yang sifatnya *client-side* seperti Java Script, Jscript ataupun VBScript. Dan untuk mengembangkan situs yang dinamis diperlukan teknologi *server-side*, seperti PHP. Dengan teknologi *server-side* kita dapat mengembangkan satu aplikasi berbasis internet yang dapat menghasilkan dan mendapatkan content secara dinamis.

Secara kerja situs dengan *content* yang statis dapat dijelaskan dengan Gambar 2.6.

Pada saat ada suatu *request* dan *browser*, *web server* akan melakukan langkah:

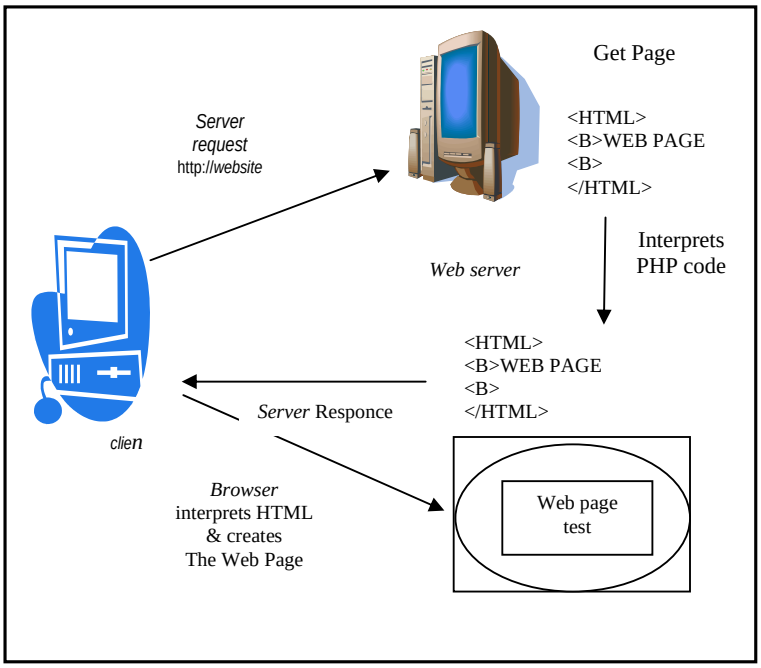
1. Membaca *request* yang dikirim oleh *browser*.
2. Mencari dan menemukan halaman dari *server*

3. Mengirim halaman yang diminta melalui internet ke *browser*



Gambar 2.6. Web yang statis

Cara kerja situs dengan *content* yang dinamis dapat ditunjukkan dengan Gambar 2.7 dibawah ini



Gambar 2.7. Web yang dinamis

Pada saat ada suatu *request* dan *browser*, *server* web akan melakukan langkah- langkah:

1. Membaca *request* yang dikirim dari *browser*.
2. Mencari dan menemukan halaman di *server*.
3. Menterjemahkan perintah yang diberikan oleh bahasa pro *server-side* menjadi halaman HTML.
4. Mengirimkan halaman yang diminta melalui internet ke *browser*.

Perbedaan mendasar antara HTML dengan bahasa *server-side* adalah HTML diterjemahkan oleh *client browser*, tidak dieksekusi ke *server*. Dengan membuat kode yang dapat dieksekusi pada *server*, kita dapat menciptakan banyak sekali aplikasi yang bersifat dinamis dan dapat dikendalikan oleh user melalui *browser*. Beberapa kelebihan teknologi *server-side* dibandingkan HTML:

1. Memberikan kemudahan untuk mengedit *content* untuk satu halaman web. pengeditan dapat dilakukan dengan, *meng-update content* dalam satu basis data dan tidak lagi pada kode HTML-nya.
2. Dapat membuat halaman yang dapat dikostumisasi penampilan sesuai dengan keinginan *user*.
3. Dapat menampilkan dan melakukan perubahan data pada basis data yang dapat dilakukan melalui web itu sendiri.
4. Memperoleh feedback dari user yang mengembalikan informasi berdasarkan isian yang disediakan untuk user.

2.6 WWW (World Wide Web)

World Wide Web (WWW) lebih dikenal dan sering disebut web, merupakan suatu sistem penyediaan informasi dengan skala yang besar yang mengelola informasi tersebut secara terdistribusi dalam internet dengan menggunakan teknologi *hypermedia*.

WWW menggunakan model *client-server*. WWW juga menggunakan ekspresi informasi, transfer informasi. Metode penamaan informasi yang standar sehingga dapat memproses dan mentransfer informasi secara terdistribusi yang dilakukan dengan sistematik. Informasi yang terdistribusi disimpan dalam *server* WWW dan user mengakses informasi tersebut menggunakan suatu *software* yang disebut dengan *browser*. WWW menggunakan standar untuk mencapai konsistensi dalam menghasilkan dan melakukan transfer informasi.

WWW menyediakan suatu *interface* yang terintegrasi terhadap protokol, format data, sistem pengalamatan, dan lain-lain sehingga sebagai macam layanan dan basis data yang ada di internet dan dapat di akses secara langsung.

WWW mempunyai beberapa karakteristik sebagai berikut:

1. Dukungan *Interface* yang terintegrasi

   WWW menyediakan suatu *interface* yang terintegrasi terhadap protokol TCP/IP, format data, sistem pengalamatan. dan lain-lain sehingga berbagai macam layanan dan berbasis data yang ada di internet dapat diakses secara langsung.

2. Kemudahan pada sisi penggunaan

   WWW mendukung secara transparan sebagian besar aplikasi-aplikasi yang terdapat di internet seperti Iayanan *telnet, gopher*, *anonymous ftp*, *finger*,

dan lain-lain. Kombinasi layanan ini memberikan user *interface* yang terintegrai dengan aplikasi-aplikasi internet yang lain.

3. Kemudahan dalam perkembangan

   Kapabilitas *server* WWW dapat dikembangkan secara mudah dengan menggunakan *Script Server Programming* antara *server* WWW dengan aplikasi yang lain. *Script Server Programming* juga memungkinkan suatu perubahan informasi secara dinamis yang dapat diperoleh secara *real time*.

4. Tidak tergantung pada suatu *platform* tertentu

   WWW memungkinkan seseorang dapat membangun *server* WWW di berbagai sistem yang berlainan dan memberikan informasi dalam bentuk *hypermedia*. WWW tidak didesain untuk mendukung sistem tertentu.

5. HTML

   HTML adalah singkatan dari *Hyper Text Markup Language*, suatu sistem penulisan perintah dan *formating hypertext* sederhana yang dituliskan ke dalam dokumen teks ASCII normal agar dapat menghasilkan tampilan visual yang terintegrasi. Dengan kata lain, dokumen yang dapat dibuat dalam aplikasi pengolahan kata yang di simpan kedalam format ASCII normal sehingga menjadi *website* dengan tambahan perintah-perintah HTML. Dengan menggunakan perintah - perintah HTML kemungkinan user untuk melakukan fungsi-fungsi berikut ini:

   - Menentukan ukuran dan alur teks.
   - Mengintegrasikan gambar dengan teks (*in link*)
   - Membuat links
   - Mengintegrasikan *file* audio dan video

- Membuat *form* interaksi.

HTML lebih menekankan pada penggambaran komponen - komponen dan formating di dalam halaman web dari pada menentukan penampilan. Sedangkan web *browser* digunakan untuk *menginvite* perintah-perintah HTML yang disisipkan kedalam teks dan menampilkan susunan halaman ke *style built - in browser* dengan menggunakan *font*, tab, warna, garis dan teks yang dikehendaki ke komputer yang menampilkan ke halaman web.

2.7 Web Design

Web design merupakan suatu rancangan dari *web pages*, *website*, dan aplikasi web yang menggunakan HTML, CSS, dan *image*. *Web design* merupakan bagian dari upaya membuat sebuah *website* (*web development*) yang dapat menyertakan pula konfigurasi *web server*, penulisan aplikasi web, dan keamanan server.[26]

Web design merupakan pengaturan dan pembuatan halaman web dalam upaya membangun sebuah *website*. Ada beberapa aspek yang mempengaruhi kegiatan ini, dan berhubungan dengan pengembangan yang sangat cepat, aspek-aspek baru terus bermunculan. Selama perkembangan *website* berorientasi bisnis terus berlangsung, ada beberapa hal mendasar yang sangat mempengaruhi yaitu:

a. Visabilitas dari sebuah *site* di internet
b. Nilai informasi dari sebuah *site*, terutama dari pandangan para pencari informasi
c. Nilai keindahan/ tampilan secara profesional dari halaman web.

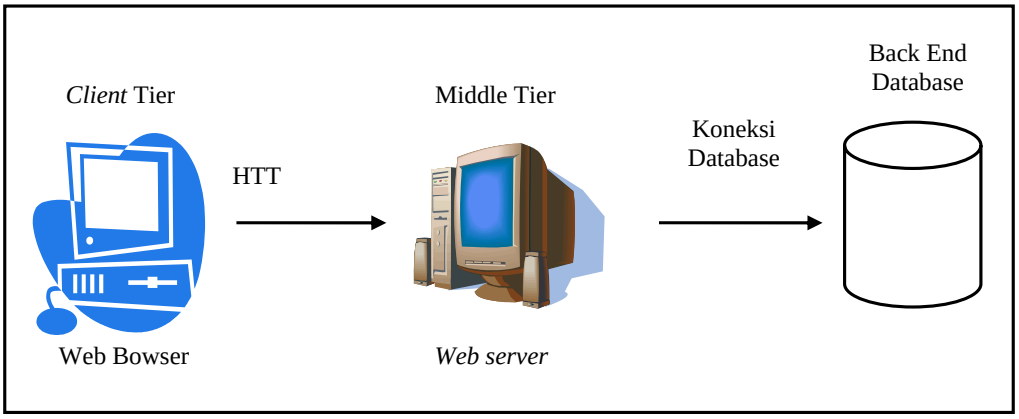
"Web Design is the art and process of creating a single Web page or entire Web sites and may involve both the aesthetics and the mechanics of a Web site's operation although primarily it focuses on the look and feel of the Web site. Some

of the aspects that may be included in Web design or Web production are graphics and animation creation, color selection, font selection, navigation design, content creation, HTML/XML authoring, JavaScript programming, and ecommerce development. Web design is a form of electronic publishing." [25]

2.8 Teknologi *Server* Basis Data

Web yang dinamis harus didukung oleh ketersediaan *content*. *Content* ini harus mudah untuk dimodifikasi dan diakses. Untuk memudahkan proses tersebut *content* dan suatu web yang dinamis disimpan dalam suatu basis data.

Secara umum, arsitektur sistem terdiri dan *client tier, middle tier dan back end database,* seperti terlihat pada Gambar 2.8 berikut ini:



Gambar 2.8. Arsitektur Sistem

Client Tier adalah komputer *client* yang akan mengakses aplikasi berbasis web pada sistem. Komputer ini dilengkapi dengan *web browser* yang mampu menerjemahkan format HTML kedalam tampilan yang sesuai.

Middle tier berfungsi sebagai tempat untuk memproses aplikasi yang diminta oleh *browser*. Tingkatan ini berupa sebuah *web server* dilengkapi dengan *script* yang akan dieksekusi oleh *server*.

Back end database merupakan tempat penyimpanan database yang berupa transaksi transaksi yang dilakukan oleh pemakai.

Secara sederhana cara ke sistem ini adalah sebagai berikut:

Komputer *client* melalui web *browser* akan mengirim permintaan HTTP ke *middle tier*. *Middle tier* akan mengambil data dari *backend database*, kemudian melakukan beberapa pemrosessan data, dan mengirim kembali hasil pemrosessan kepada komputer *client*.

2.9 *Web server*

Web server adalah sebuah *server* yang melayani *request*/permintaan yang datang dari komputer lain ke komputer tujuan tempat dimana *file* yang diinginkan berada. Terdapat beberapa modul yang diletakkan pada *client* atau *server* untuk keperluan tertentu, seperti modul PHP pada *web server* agar *web server* dapat menjalankan program yang ditulis dalam *file* PHP. *File-file* tersebut diletakkan pada direktoni yang telah teregistrasi pada *web server*. Tanpa *web server* kita tidak dapat menjalankan *file* yang dapat dijalankan pada *server* seperti menjalankan *file* PHP, ASP dan sebagainya. *Web server* yang dapat digunakan dalam sistem informasi ini adalah Apache *Web server* dan bisa digunakan di semua sistem operasi.

2.10 PHP (*Personal Home Page*)

Server side Scripting adalah bahasa scripting yang dieksekusi ke *server* pada saat ada permintaan layanan pada *Web server*. Ada beberapa bahasa scripting pada *server* saat mi seperti PHP, ASP dan JSP. Dan seringkali dalam implementasinya melibatkan aplikasi *database*, manipulasi *image*, mengirim email, enkripsi, LDAP, XML, aplikasi WAP dan banyak lagi.

PHP merupakan sebuah model pemrograman web yang bersifat *server side scripting*. PHP adalah sebuah bahasa *scripting* yang bisa digunakan dengan

bahasa HTML, atau dokumennya secara bersamaan dan bisa menyatu dengan tag-tag HTML, yang dieksekusi di *server*, dan digunakan untuk membuat halaman WEB yang dinamis. PHP dalam perkembangannya kini sudah ada PHP versi 4.0 yang merupakan penyempurnaan dan *scripting engine* dan *build in* HTTP *session* dan PHP versi sebelumnya.

Alasan penulisan memilih menggunakan PHP karena:

1. *Life Cycle* yang singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi Internet.
2. *Cross Platform*, sehingga dapat dipakai dihampir semua *web server* yang ada di pasaran dan di yang dijalankan pada berbagai sistem informasi (Windows, Linux, Unix. *Free*BSD, Solaris dan lain - lain).
3. PHP meningkatkan kecepatan dan proses *script*nya dan cukup stabil serta mendukung banyak paket database baik yang komersil maupun non komersil.

2.11 SQL dan MySQL

SQL adalah singkatan dari *Structured Query Language* dan sering disebut sequel saja. SQL merupakan bahasa standar untuk pengolahan *database.*

Structured Query Language yang biasa disingkat dengan SQL merupakan bahasa standar untuk mengakses *database* yang bersifat *non-procedure* karena bahasa ini tidak memiliki perintah perulangan (*looping*) maupun percabangan (*kondisi*).

Sebagai sebuah standar, sejalan dengan perkembangan teknologi basis data dan teknologi komputer pada umumnya, SQL sendiri mengalami beberapa perubahan (atau tepatnya penyempurnaan). SQL mula-mula dibuat pada tahun

1970 dan disebut dengan *sequel*. Standarisasi yang pertama pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan ISO (*International Standarts Organization*), yang disebut sebagai SQL-86. SQL-86 ini diperbaharui pada tahun 1989 menjadi SQL-89. Standar terakhir yang dibuat dan digunakan hingga saat ini adalah SQL-92 yang dikeluarkan pada tahun 1992.

Sebuah ekspresi SQL dasar sebenarnya hanya terdiri atas 3 klausa, yaitu: ***select,from*** dan ***where.***

a. Klausa ***select*** digunakan untuk menetapkan daftar atribut (*field*) yang diinginkan sebagai hasil *query*.
b. Klausa ***from*** digunakan untuk menetapkan tabel (atau gabungan tabel) yang akan ditelusuri selama *query* data dilakukan.
c. Klausa ***where*** yang sifatnya opsional, digunakan sebagai predikat (kriteria) yang harus di penuhi dalam memperoleh hasil *query*.

Sintak (cara penulisan) dari ekspresi SQL dasar tersebut adalah:

select a1 [,a2,..,an]

from t1 [,t2,…,tn]

[***where*** P]

di mana:

a. a1, a2,…,an merupakan daftar atribut (field)
b. t1, t2,…,tn merupakan daftar tabel
c. [ ] merupakan tanda opsional.

Aplikasi *database* dirasakan adalah sebuah komponen yang sangat penting dan harus ada dalam sebuah aplikasi *website* karena posisinya sebagai penyimpanan data, *update* dan hal-hal yang berhubungan dengan pengolahan data.

Ada beberapa Aplikasi *Database server* saat ini yang sering digunakan orang dalam membangun dan mendukung aplikasi berbasis *web* seperti MySQL, Postge SQL, mSQL, Microsoft SQL *Server*, Oracle, Informix,dan banyak lagi.

Dalam sistem ini sistem informasi berbasis web ini digunakan *Database* MySQL. MySQL adalah *database server* buatan T.C.X Data Konsult AB (http://www.tcx.se) Swedia MySQL didesain agar dapat berjalan cepat sampai 100 akses secara simultan yang membuat MySQL menghilangkan kemampuan seperti transaktional processing yang bisa diakses secara *multi user* (banyak user) dan *multi – threaded* (banyak urutan).

MySQL sangat cepat dan *fleksible* saat diakses dan juga menyediakan logs penyimpanan dan gambar yang saat dimpleimentasikan pada aplikasi *client server* terdiri dari *server deamon* MySQL dan beberapa program *client* dan *libraries*.

Beberapa keunggulan dari MySQL adalah sebagai berikut:

1. MySQL adalah open source *software* yang dapat diperoleh secara gratis, dapat di download di (http://www.mySQL.com).
2. Kemampuan mengatasi *unlimited number* dan *simultaneous users*.
3. Mempunyai kapasitas untuk menangani 50000000 record data.
4. Kemampuan perintah eksekusi yang cepat, bahkan paling cepat yang ada di pasaran.

2.12 Pemodelan Data

Pemodelan data atau data *modelling* merupakan sebuah teknik untuk mendefiniskan persyaratan bisnis untuk sebuah *database*. Pemodelan data kadang disebut sebagai pemodelan *database* karena model data kadang-kadang diimplementasikan sebagai sebuah *database*.

Ada beberapa catatan mengenai pemodelan data. Model aktual disebut *entity relationship diagram* (ERD) karena model ini menjelaskan data dalam konteks entitas dan hubungan yang digambarkan oleh data tersebut.

Beberapa konsep dasar yang mendasari semua model data, antara lain:

1. Entitas, adalah sesuatu yang diperlukan bisnis untuk menyimpan data. Entitas adalah sekelompok orang, tempat, objek, kejadian atau konsep tentang apa yang kita perlukan untuk men-*capture* dan menyimpan data. Biasanya entitas digambarkan sebagai persegi dengan sudut tumpul.
2. Atribut, yaitu sifat atau karakteristik deskriptif suatu entitas. Sinonimnya antara lain elemet, sifat, dan bidang. Beberapa atribut dikelompokkan secara logis menjadi superatribut yang disebut *compound attribute*/atribut gabungan, yaitu atribut yang terdiri lain. Sinonimnya dalam berbagai bahasa pemodelan data sangat banyak, antara lain *concatenated attribute, composite attribute*, dan struktur data.
3. Hubungan atau *relationship*, adalah hubungan bisnis alami yang ada di antara satu atau lebih entitas. Hubungan tersebut dapat menyatakan kejadian yang menghubungkan entitas atau hanya persamaan logika yang ada di antara entitas.[23]

2.13 Pemodelan, Analisis, dan Perancangan Berorientasi Objek

Pemodelan berorientasi objek adalah pemodelan data yang menerapkan teknik permrograman berorientasi objek.[7] Pemodelan objek atau o*bject modelling* adalah teknik untuk mengidentifikasi objek di dalam lingkungan sistem dan mengidentifikasi hubungan antara objek-objek tersebut. [23]

Analisis sistem informasi berorientasi objek adalah teknik yang digunakan untuk baik dalam melakukan studi atas objek yang telah ada untuk melihat apakah mereka dapat digunakan ulang atau diadaptasikan sebagai manfaat baru, dan juga untuk mendefinisikan objek baru atau yang dimodifikasi yang akan dikombinasikan dengan objek yang telah ada dalam aplikasi komputasi bisnis yang bermanfaat.

Perancangan berorientasi objek adalah kelanjutan dari proses analisis berorientasi objek. Kelanjutan yang dimaksud adalah melanjutkan pemodelan atas hasil analisis dengan maksud memfokuskan pada pengembangan sistem informasi.

Analisis dan perancangan sistem informasi berorientasi objek teleh mengalami perkembangan dalam metodologinya, sebagai berikut:

1. Metode Shalre/Mellor
2. Metode Wirfs Brock
3. Metode Coad/Yourdan
4. Metode Booch
5. Metode Teknik Memodel Objek
6. Metode *Unified Modelling Language* (UML) yang diperkenalkan pada tahun 1997.

2.13.1 UML (*Unified Modelling Language*)

Sri Dharwiyanti dan Romi Satria Wahono mendefinisikan: "*Unified Modeling Language (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."

A. Suhendar S.Si dan Hariman Gunadi S.Si, MT mendefinisikan:

"*Unified Modeling Language (UML)* adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi, dan mendokumentasikan *artifacts* dari sistem *software*, untuk memodelkan bisnis, dan sistem *nonsoftware* lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks."

Kemudian Ali Bahrami juga mengemukakan:

"The unified modeling language is a language for specifying, constructing, visualizing, and documenting the software system and its components. A UML is a graphical language with sets of rules and semantics. The rules and semantics are expressed in English, in a form known as object constraint language (OCL)."

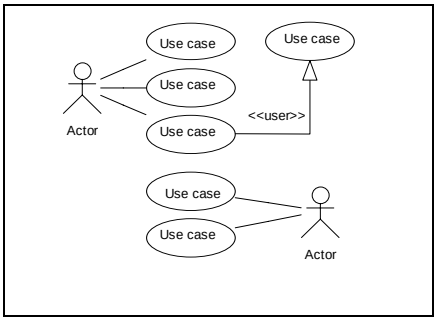
Secara umum UML merupakan 'bahasa' untuk visualisasi, spesifikasi, konstruksi, serta dokumentasi. Dalam kerangka visualisasi, para pengembang menggunakan UML sebagai suatu cara untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna sistem/perangkat lunak. Dengan adanya 'bahasa' yang bersifat standar, komunikasi perancangan dengan pemrogram (lebih tepat lagi: komunikasi antaranggota kelompok pengembang) serta calon pengguna diharapkan menjadi mulus. Dalam kerangka spesifikasi, UML menyediakan model-model yang tepat, tidak mendua-arti (*ambigu*), serta lengkap. Secara khusus, UML menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan, serta implementasi dalam sistem yang sangat bernuansa perangkat lunak (*software intensive system*).

Pendekatan yang dilakukan dengan metode UML (*Unified Modeling Language*) penulis membuat beberapa abstrak konsep dasar dari UML, yaitu terdiri dari *structural classification*, *dynamic behavior*, dan *model management*. Untuk penentuan pertama dalam pembuatan UML penulis mendefinisikan dalam diagram-diagram yang nantinya diperlukan, yaitu:

1. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan pada "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case diagram* mempresentasikan sebuah interaksi antara *actor* dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, menciptakan sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Contoh penggunaan *use case diagram*:



Gambar 2.9. Contoh Penggunaan *Use Case Diagram*

2. *Class Diagram*

Class diagram merupakan sebuah spesifikasi yang jika diinisialisasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem,

sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi).

Class diagram meggambarkan struktur dari deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan asosiasi, dan lain-lain.

Class memiliki tiga area pokok:

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut:

a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
c. *Public*, dapat dipanggil oleh siapa saja.

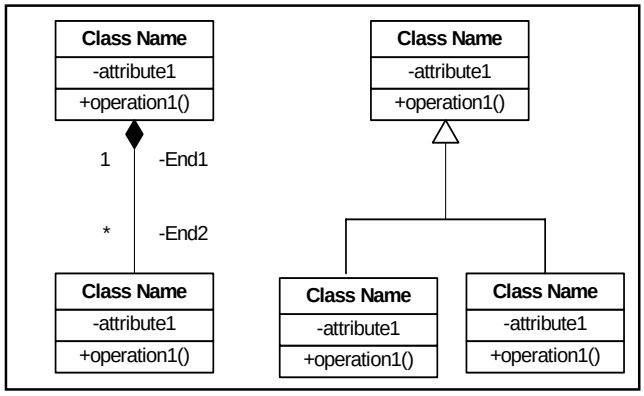
Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstanisasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Terdapat hubungan antarkelas dalam *class* diagram ini, antara lain:

1. Asosiasi, yaitu hubungan statis antarkelas. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.

2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram*yang akan dijelaskan kemudian.

   Contoh penggunaan *class diagram:*



Gambar 2.10. Contoh Penggunaan *Class Diagram*

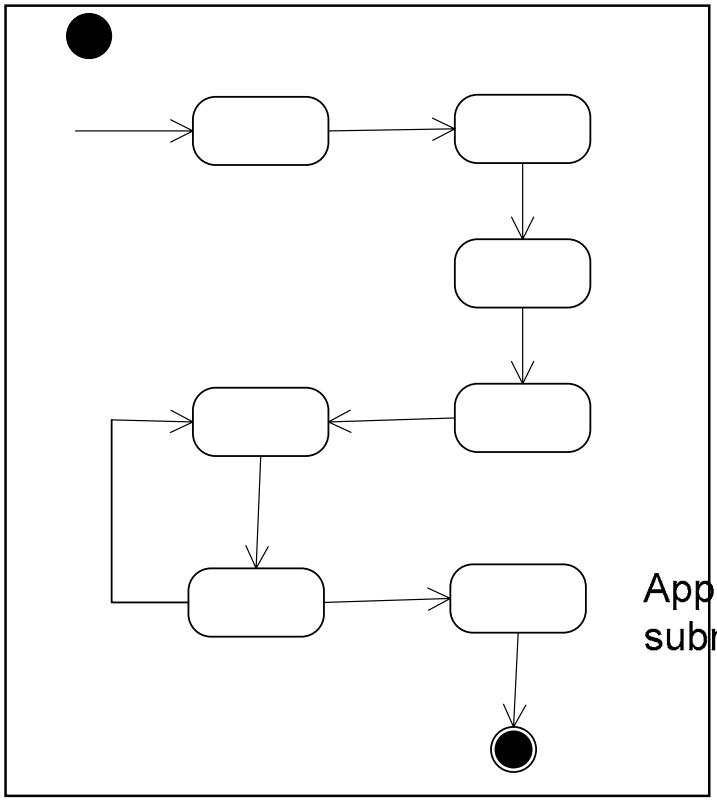
3. *Statechart Diagram*

Statechart Diagram menggambarakan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart* diagram).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang

bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring.

Contoh penggunaan *statechart diagram*:



Gambar 2.11. Contoh Penggunaan *Statechart Diagram*

4. *Activity Diagram*

Menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka terakhir. *Activity diagram* juga dapat menggambarkan proses *parallel* yang mungkin terjadi pada beberapa eksekusi.

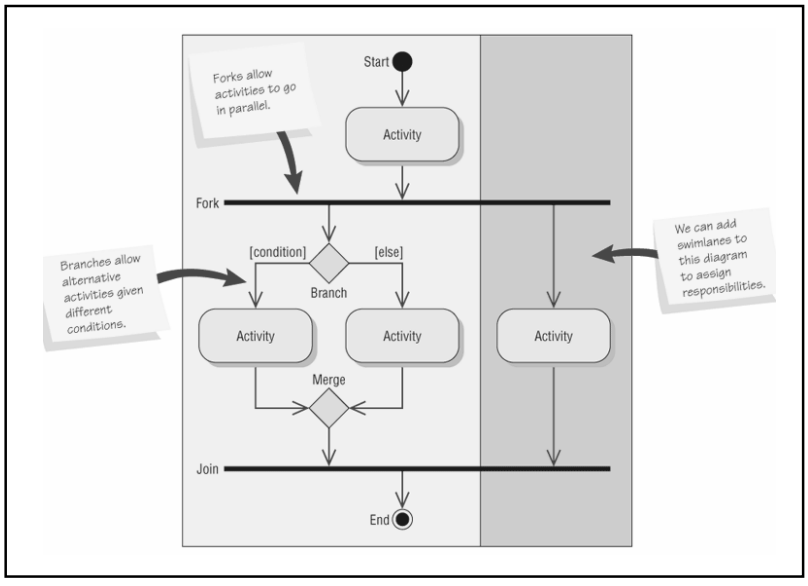
Activity diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagai besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak

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.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *joint*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Contoh penggunaan *Activity diagram:*



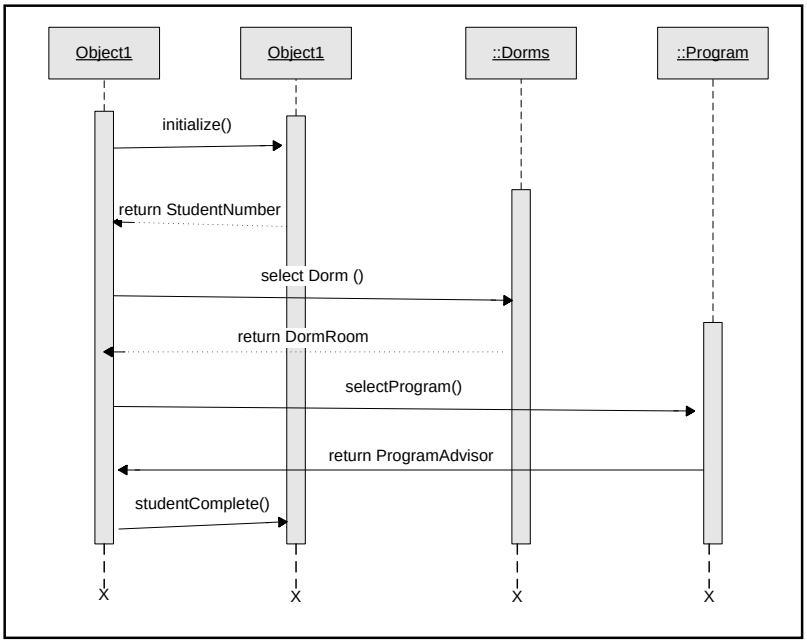
Gambar 2.12. Contoh Penggunaan *Activity diagram*

5. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari antar dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Contoh Penggunaan *Sequence diagram:*

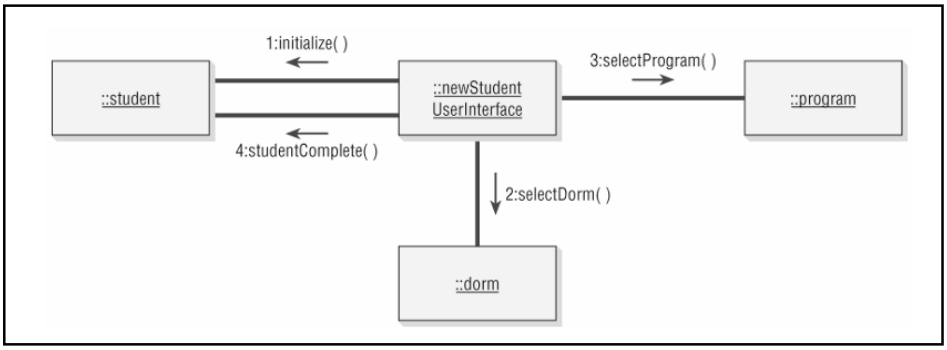


Gambar 2.13. Contoh Penggunaan *Sequence Diagram*

6. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message.* Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1. *Message*s dari level yang sama memiliki prefiks yang sama.

Contoh penggunaan *Collaboration diagram*:



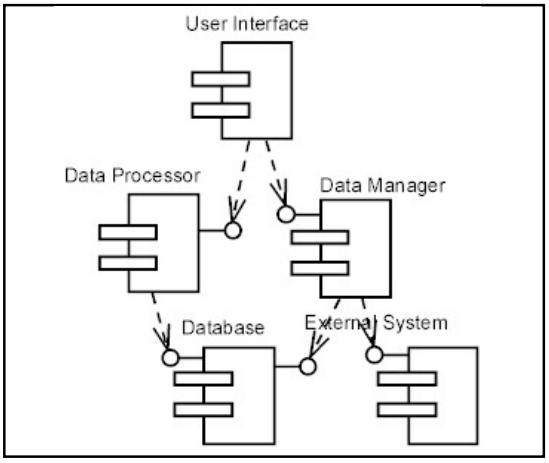
Gambar 2.14. Contoh Penggunaan *Collaboration diagram*

7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tetapi dapat juga dari komponen-komponen yang lebih kecil.

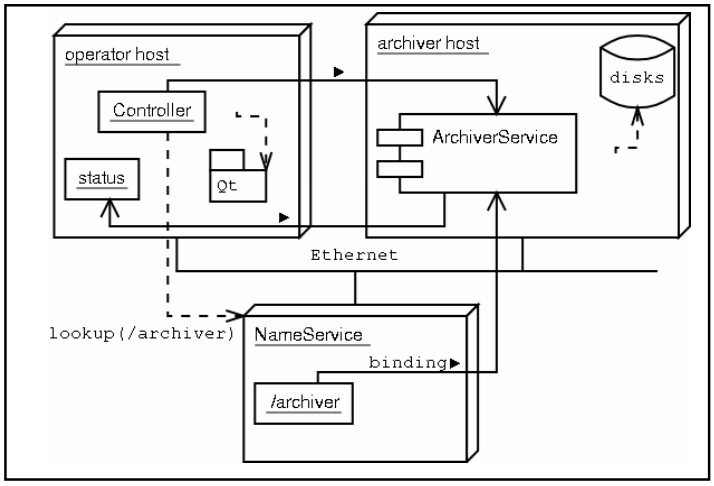
Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Contoh penggunaan *Component diagram*:



Gambar 2.15. Contoh Penggunaan *Component diagram*

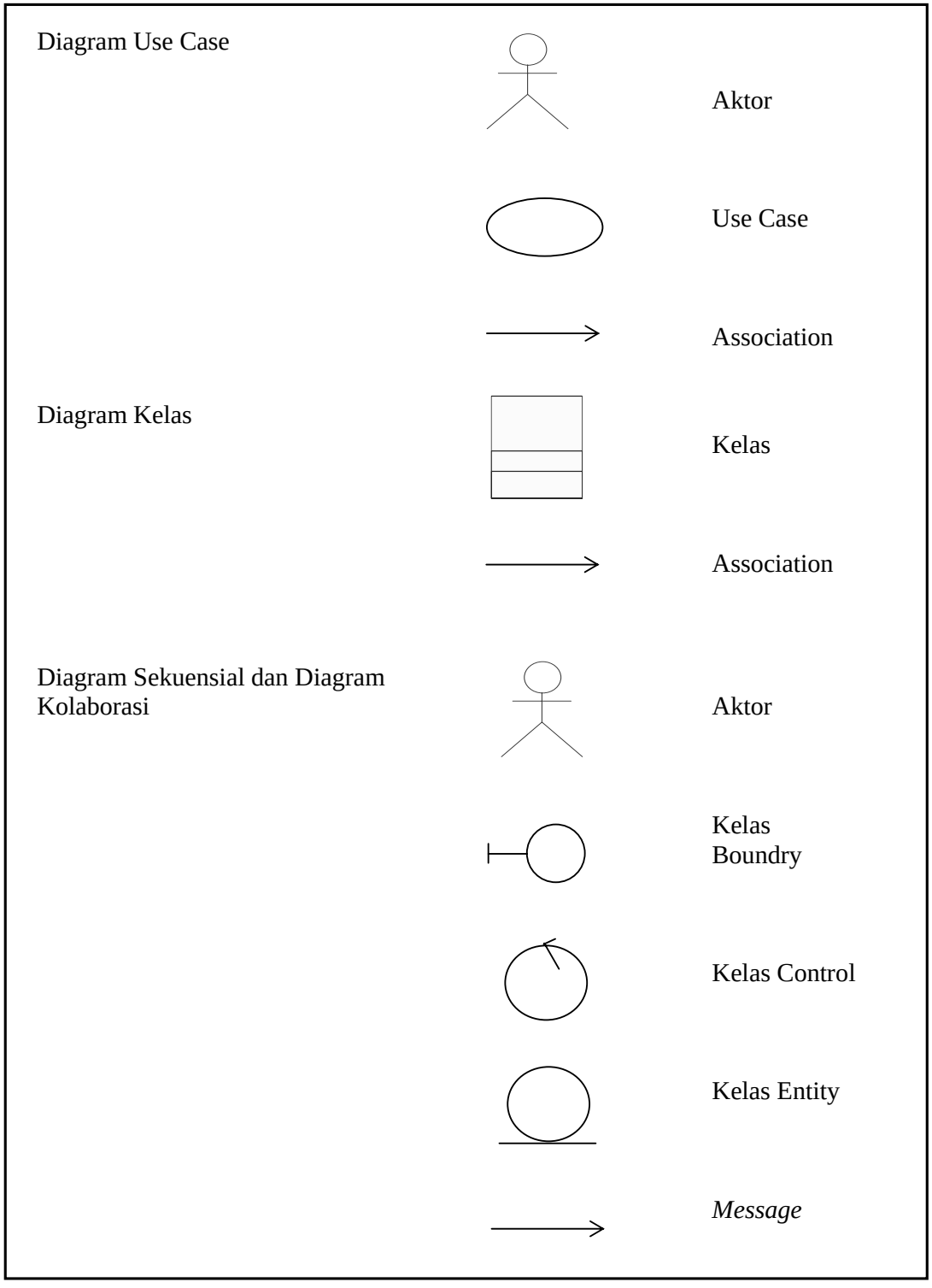
8. *Deployment Diagram*



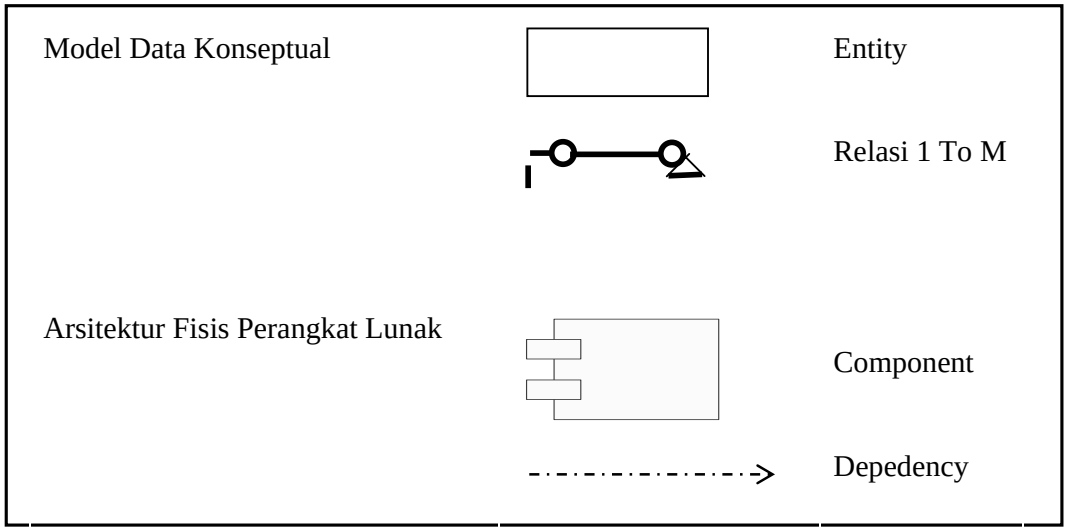
Gambar 2.16. Contoh Penggunaan *Deployment diagram*

Dari uraian di atas, maka pelaksanaan dari analisis dan perancangan sistem informasi yang berorientasi objek dengan pemanfaatan UML dapat dilakukan dengan memodelkan semua subsistem dari sistem informasi ke dalam diagram-diagram. Diagram tersebut menjadi alat komunikasi yang terdiri dari berbagai pihak yang terkait dengan sistem informasi tersebut.

Berikut adalah daftar simbol dari diagram-diagram tersebut antara lain:



Gambar 2.17. Simbol-simbol Diagram UML (1)



Gambar 2.18. Simbol-simbol Diagram UML (2)

2.15 Metode Pengembangan Sistem Informasi

Untuk mengembangkan sistem informasi yang akan dihasilkan, metode yang digunakan adalah dengan memanfaatkan model *waterfall*. Model ini juga dikenal dengan nama model tradisional atau model klasik, dan model ini menawarkan cara pembuatan sistem informasi secara lebih nyata.

Langkah-langkah yang penting dalam model ini adalah:

1. Penentuan dan analisis spesifikasi

   Jasa, kendala, dan tujuan dihasilkan dari konsultasi dengan pengguna sistem. Kemudian semuanya itu dibuat dalam bentuk yang dapat dimengerti oleh *user* dan staf pengembang.

2. Desain sistem dan sistem informasi

   Proses desain sistem membagi kebutuhan-kebutuhan menjadi sistem sistem informasi atau perangkat keras. Proses tersebut menghasilkan sebuah arsitektur sistem keseluhan. Desain sistem informasi termasuk menghasilkan fungsi sistem sistem informasi dalam bentuk yang mungkin ditransformasi ke dalam satu atau lebih program yang dapat dijalankan.

3. Implementasi dan ujicoba unit

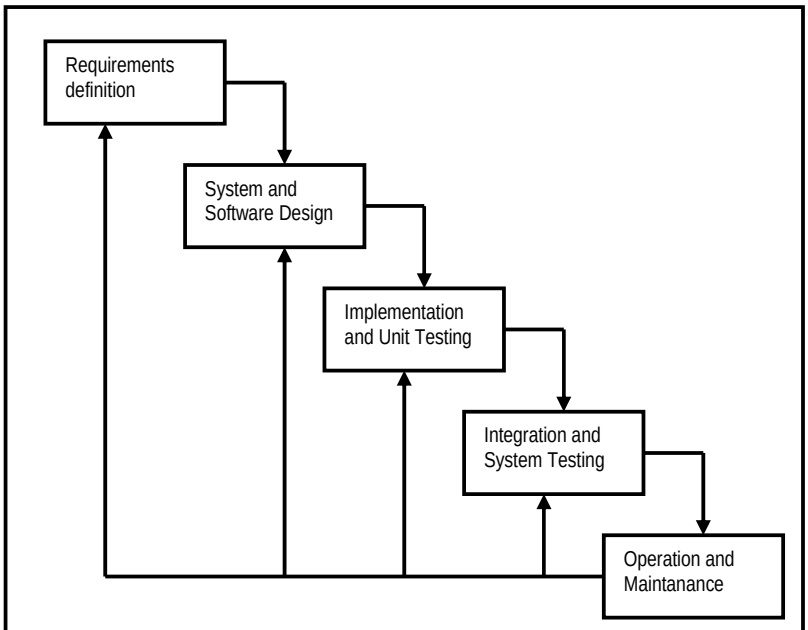
   Selama tahap ini desain sistem informasi disadari sebagai sebuah program lengkap atau unit program. Uji unit termasuk pengujian bahwa setiap unit sesuai spesifikasi.

4. Integrasi dan ujicoba sistem

   Unit program diintegrasikan dan diuji menjadi sistem yang lengkap untuk menyakinkan bahwa persyaratan sistem informasi telah dipenuhi. Setelah ujicoba, sistem disampaikan ke pelanggan.

5. Operasi dan pemeliharaan

   Normalnya, ini adalah fase yang terpanjang. Sistem dipasang, digunakan, dan dilakukan pemeliharaan termasuk pembetulan kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru ditemukan.



Sumber: Herlambang dan Tanuwijaya, 2005

Gambar 2.19. Model Waterfall