

BAB 2

LANDASAN TEORI

2.1 *Internet*

Internet adalah rangkaian komputer yang terhubung didalam beberapa rangkaian yang terhubung secara global dan menggunakan TCP/IP sebagai pertukaran paket.

Pada awalnya *internet* hanya digunakan oleh militer saja, *internet* pertama kali ditemukan pada tahun 1957 pesawat ulang – alik Sputnik diluncurkan sebagai stasiun luar angkasa pertama. Pada tahun 1968 *internet* mulai digunakan oleh beberapa universitas ternama di Amerika, dan pada tahun 1969 akhirnya jaringan *internet* dapat disambungkan melalui beberapa universitas, yaitu ; *The University of California* at Los Angeles, SRI (in Stanford), *University of California* at Santa Barbara, dan *University of Utah*. Jaringan dihubungkan dengan kabel berkecepatan 50 Kbps. Email pertama kali dibuat oleh Ray Tomlinson dari BBN. Tahun 1973 mulai menggunakan *protokol* TCP/IP yang didirikan oleh suatu group yang disebut DARPA. Tahun 1976 Dr. Robert M. Metcalfe menemukan *Ethernet*, dimana menggunakan kabel coaxial yang dapat mengirim data dengan cepat. Pada akhirnya penemuan ini yang akan mengilhami LAN. Pada tanggal 1 Januari 1981 semua mesin yang tersambung ke ARPANET pecah menjadi 2, yaitu MILNET dan ARPANET dimana MILNET menangani kebutuhan militer dan ARPANET mendukung pengembangan IPTEK, *Departement of Defence* melanjutkan dukungan pada kedua jaringan. Tahun 1992 *World Wide Web (WWW)* dikeluarkan oleh CERN

dan NSFNET ditingkatkan menjadi T3 (44,736 Mbps) . Tahun 1993 NFS menyediakan layanan *internet* dan pada tahun 1994 beberapa perusahaan menggunakan *internet* sebagai tempat usahanya, diantaranya ; Pizza Hut, First Virtual (cyberbank pertama). Tahun 1995 NFS menjual domain dengan harga \$50 per tahun diluar .edu dan .gov. Tahun 1996 lalu – lintas *internet* ditopang oleh ISPs termasuk MCI, AT&T, Sprint, UUnet, BBN planet, dan banyak lagi.

2.1.1 *World Wide Web (WWW)*

World Wide Web (WWW) adalah suatu ruang informasi yang dipakai oleh pengenalan global yang disebut URI (*Uniform Resource Identifier*) untuk mengidentifikasi sumber – sumber daya yang berguna. Desain *World Wide Web* yang baik adalah menyeimbangkan struktur dan hubungan menu dari menu atau home page dengan halaman isi atau grafik dan dokumen yang di-*link*. Tujuannya untuk membangun hierarki menu dan terstruktur dengan baik bagi pemakai, dan tidak mengganggu pemakai situs *web* atau menyesatkan mereka.

2.1.2 *Web Hosting*

Web Hosting adalah layanan penyewaan ruang simpan data (*space*) yang digunakan untuk menyimpan data – data website agar halaman website tersebut bisa diakses dari mana saja. Data web tersebut meliputi file – file html, php script, cgi script, css, image, database, dan file lain yang dibutuhkan untuk menampilkan halaman web. Selain menyimpan file – file tersebut diatas, biasanya web hosting juga memberikan fasilitas e-mail.

2.1.3 TCP/IP

TCP/IP adalah protokol yang memungkinkan semua komputer untuk menggunakan data yang disalurkan melalui jaringan internet. TCP/IP dikembangkan pada tahun 1978 oleh *U.S. Department of Defense's Advanced Research Projects Agency for Space and Computer Research*. Protokol TCP/IP ini menentukan jenis pengecekan *error build – in* yang akan digunakan, metode pengkompresan data (jika ada), bagaimana perangkat pengirim akan mengetahui bahwa ia telah selesai mengirimkan data, dan bagaimana perangkat penerima akan mengetahui bahwa ia telah menerima pesan. TCP/IP memisahkan data menjadi paket – paket yang merupakan blok data terbesar yang dapat dikirimkan melalui internet (kurang dari 1500 karakter, atau 128 Kilobyte). IP digunakan untuk mengirimkan paket – paket data melalui internet ke tujuan akhirnya. Sedangkan TCP digunakan untuk mengatur paket – paket data yang dikirimkan kedalam urutan yang sebenarnya atau memastikan agar penerima menerima paket – paket data secara benar. Jika paket data yang diterima tidak benar, maka penerima akan mengirim pesan agar paket dikirim lagi. Penerima juga akan mengirimkan pesan bahwa paket telah diterima kepada pengirim apabila paket data yang diterima telah benar. Komputer – komputer dalam jaringan internet menggunakan TCP/IP untuk semua transaksi internet yang dilakukan.

2.2 Aplikasi Web

Aplikasi *Web* adalah suatu aplikasi yang dapat membentuk halaman – halaman *web* berdasarkan permintaan pemakai. Aplikasi *Web* yang bisa mencakup permintaan interaktif ataupun kelompok diskusi. Aplikasi *Web* merupakan salah satu contoh aplikasi *client/server*. *Client* mewakili komputer yang digunakan oleh seorang pemakai yang hendak menggunakan aplikasi, sedangkan *server* mewakili komputer yang menyediakan layanan aplikasi. Dalam konteks ini, *client* dan *server* berhubungan melalui *internet* dan *intranet*. Yang menarik, model *client/server* yang menggunakan aplikasi *web* dapat melibatkan bermacam – macam *platform*.

Ciri khas lain pada penggunaan aplikasi *web*, pemakai menggunakan perangkat lunak yang dinamakan *web browser* atau yang biasa disebut *browser* saja (misalnya *Netscape communication*, *Internet Explorer*, *Mozilla*) untuk mengakses aplikasi *web*. Komputer yang bertindak sebagai server umumnya menyediakan *database server*, selain *web server* yang ditujukan untuk melayani permintaan pemakai yang hendak mengakses aplikasi *web*. *Database server* adalah *server* yang melayani akses terhadap *database*. Oracle dan MySQL merupakan contoh dari sekian *database server*. Adapun contoh *web server* yaitu Apache (sangat terkenal di lingkungan linux) dan IIS (*Internet Information Server*), yang merupakan andalan microsoft.

2.2.1 Web Sever

Server web adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya

berbentuk dokumen HTML. Server web yang terkenal diantaranya adalah Apache dan Microsoft Internet Information Service (IIS). Apache merupakan server web antar-platform, sedangkan IIS hanya dapat beroperasi di sistem operasi Windows.

2.2.2 HTML

Untuk membuat kode PHP dengan mudah maka kita harus mengenal sejumlah tag pada HTML. HTML (*Hypertext Markup Language*) adalah bahasa yang digunakan untuk mengkonstruksi halaman web. Berdasarkan standard inilah web browser bisa memahami isi suatu dokumen yang berasal dari web server. HTML bekerja dengan menggunakan HTTP (*Hypertext Transfer Protocol*), yaitu protokol komunikasi yang memungkinkan web server berkomunikasi dengan web browser. Kode HTML berupa sebuah berkas teks dengan akhiran berupa .HTML, .html, .HTM atau .htm.

2.2.3 PHP

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Selanjutnya rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C dan disertakan juga modul – modul ekstensi yang meningkatkan kemampuan besar PHP/FI secara signifikan. Ditahun ini juga 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 tingkatan PHP dirubah menjadi akronim berulang PHP.2 Hypertext preprocessing. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0 yang mana versi ini adalah yang paling banyak dipakai pada awal abad ke-21 dikarenakan kemampuannya untuk membangun aplikasi web kompleks yang tetapi tetap 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 ke dalam PHP untuk menjawab perkembangan bahasa pemrograman kearah paradigma berorientasi objek.

2.2.4 JavaScript

JavaScript adalah bahasa pemrograman berbasis prototype yang berjalan disisi klien. Jika kita berbicara dalam konteks web, sederhananya, kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk di browser atau halaman web agar halaman web menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu Java dan Script. Java adalah bahasa pemrograman berorientasi objek, sedangkan Script adalah serangkaian intruksi program.

Secara fungsional, JavaScript digunakan untuk menyediakan akses script pada objek yang dibenamkan (*embedded*). Contoh sederhana dari penggunaan JavaScript adalah membuka halaman pop up, fungsi validasi pada form sebelum data dikirimkan pada server, merubah image kursor ketika melewati objek tertentu, dan lain lain.

2.2.5 Perancangan Menu

Perancangan Menu diperlukan untuk navigasi pemakai dalam memudahkan penggunaan perangkat lunak. *Item – item* menu dibuat berdasarkan deskripsi kejadian dan keadaan. Menu tersebut disusun dengan hierarki tertentu dimana setiap menu memiliki submenu dan submenu dapat memiliki submenu lagi.

2.2.6 Perancangan Layar

Perancangan Layar digunakan untuk memudahkan pengguna dalam memakai perangkat lunak. Dalam setiap halamannya perancangan dilakukan dengan format yang tetap. Keuntungan menggunakan perancangan layar dengan format tetap adalah pemakai dapat mengidentifikasi informasi dengan cepat dan dapat mengatur navigasi dengan perintah – perintah yang konsisten.

2.3 Interaksi Manusia dan Komputer (IMK)

Interaksi manusia dan komputer merupakan disiplin ilmu yang berhubungan dengan perancangan, evaluasi, dan implementasi sistem komputer interaktif untuk digunakan oleh manusia serta studi fenomena-fenomena besar yang berhubungan dengannya. Tujuan dari ilmu ini adalah untuk membangun atau merancang suatu

system yang aman sesuai dengan kebutuhan, efektif, efisien, dan dapat dipakai. Konsep utama dari interaksi manusia dan komputer adalah bagaimana membuat sistem yang mudah dipelajari dan digunakan.

2.3.1 Delapan Aturan Emas Perancangan

Menurut Schneiderman (1998, pp 72-73), terdapat delapan aturan emas dalam merancang antar muka pengguna, yaitu :

- a. Berusaha untuk konsisten Antarmuka pengguna harus memperhatikan konsistensi misalnya konsistensi dalam aksi untuk melakukan tugas yang serupa dan konsisten tampilan (penggunaan istilah, menu, bantuan, warna, dan jenis huruf).
- b. Memungkinkan pemakai yang sudah biasa untuk menggunakan *shortcut*. *Shortcut* akan memudahkan pemakai yang sudah biasa dan menginginkan interaksi yang singkat dan cepat.
- c. Mementingkan umpan balik balik (*feedback*) yang informative untuk aksi yang sering digunakan, diberi *feedback* yang sederhana dan mencukupi, sedangkan untuk aksi yang jarang dilakukan berikan *feedback* yang lebih rumit dan informatif.
- d. Memberikan tanda untuk suatu penutupan (keadaan akhir). Harus terdapat tanda kapan di mulai, pertengahan, dan akhir dari prosesnya, dan juga terdapat penjelasan yang informatif pada setiap perpindahan proses.
- e. Memberikan pencegahan kesalahan dan penanganan kesalahan sederhana. Sistem harus dirancang sedemikian rupa agar pemakai tidak dapat membuat kesalahan yang serius. Jika kesalahan itu terjadi sistem harus dapat

mendeteksi dan menawarkan instruksi yang sederhana untuk memperbaikinya.

- f. Memungkinkan pembalikan aksi dengan mudah. Setiap aksi sedapat mungkin harus dapat dibatalkan, sehingga pemakai tidak takut untuk melakukan kesalahan.
- g. Mendukung pusat kendali internal (*internal focus control*). Tanggapan sistem yang tidak jelas, urutan memasukkan data yang tidak jelas, kesulitan mendapatkan informasi yang diperlukan, dan ketidakmampuan sistem menghasilkan aksi yang diinginkan akan membuat *user* merasa tidak memegang kendali.
- h. Mengurangi beban ingatan jangka pendek. Ingatan manusia adalah terbatas, sehingga diusahakan agar tampilan dibuat sesederhana mungkin dan mengurangi pergerakan antara jendela sehingga mengurangi beban ingatan jangka pendek.

2.4 Database

Menurut Carolyn Begg (2000, p13), *database* adalah sistem terkomputerisasi yang tujuan utama adalah memelihara informasi dan membuat informasi tersedia saat dibutuhkan. Sebuah system *database* dapat memiliki beberapa *database*. Setiap *database* dapat berisi / memiliki sejumlah objek *database*, yang antara lain yaitu :

a. *Field*

Field adalah sekumpulan kecil dari kata atau sebuah deretan angka – angka.

b. *Record*

Record adalah kumpulan dari *file* yang berelasi secara logis. Contoh : nama, alamat, nomor telepon, dan sebagainya.

c. *File*

File atau berkas adalah kumpulan dari *record* yang berelasi secara logis. Contoh : berkas transaksi toko A yang mempunyai *record* tanggal, kode barang, dan harga.

d. *Entity*

Entity adalah orang, tempat, benda, atau kejadian yang berkaitan dengan informasi yang disimpan. Contoh : pelanggan, pekerja, dan sebagainya.

e. *Attribute*

Attribute adalah karakteristik yang menjelaskan suatu *entity*. Contoh : nama pelanggan, umur pekerja, dan sebagainya.

f. *Primary key*

Primary key adalah sebuah *field* yang nilainya unik yang tidak sama antara satu *record* dan *record* lain. *Primary key* digunakan sebagai tanda pengenal suatu tanda pengenal dari suatu *field*.

g. *Foreign key*

Foreign key adalah sebuah *field* yang nilainya berguna untuk menghubungkan *primary key* lain yang berada pada table yang berbeda.

2.4.1 Relational Database

Relational *database* adalah representasi logical dari data. Data tersebut dapat diakses tanpa ada ketergantungan dengan struktur fisik dari *database* tersebut. Relational *database* merupakan sistem *database* yang paling banyak dipakai saat ini.

Salah satu bahasa yang sering dipakai untuk memanipulasi data adalah SQL. Data dalam relational *database* disimpan didalam table dimana terdapat kolom dan baris.

2.4.2 MySQL

MySQL adalah salah satu *database server* yang sangat terkenal, disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database*. *Software server* MySQL pertama dibuat oleh Michael “Monthy” Wildenius dan timnya pada tahun 1994. Pemakaian *database* MySQL yang dimaksud penulis adalah pengembang aplikasi *database* yang ingin menggunakan MySQL sebagai *database*-nya atau “*backend*” dari aplikasi yang dibangun. Sedangkan MySQL mempunyai kelebihan dapat diakses oleh banyak bahasa pemrograman. MySQL merupakan *database server* yang ideal untuk data segala ukuran dengan kemampuan mempunyai kecepatan yang sangat tinggi dalam pemrosesan data, *multi-thread*, *multi-user*, dan *query*. Ukuran database MySQL lebih kecil file *database* yang lain.

Beberapa pertimbangan programmer memilih MySQL sebagai “*backend*” dalam mengolah *database* adalah :

- Kecepatan
- Mudah digunakan
- *Open Source*
- Kapabilitas
- Biaya murah
- Keamanan

- *Lintas platform*

2.4.3 Navicat MySQL

Dalam merancang database, kami menggunakan software Navicat guna merancang database yang lebih cepat, mudah dan efisien. Navicat merupakan software database management berbasis MySQL. Software ini dilengkapi dengan GUI (Graphic User Interface). Navicat tersedia untuk tiga platform yang berbeda, yaitu Windows, Linux, Mac OS X.

Tampilan software ini cukup enak dilihat karena dilengkapi dengan berbagai icon menarik di jendela utamanya. Berbagai feature MySQL terbaru seperti Trigger, Stored Procedure, Function, View, dan Manage User juga sudah didukung oleh software ini. Namun, meskipun sudah dilengkapi GUI, bukan berarti console MySQL dihilangkan.

Adanya Tree dan table grid akan mempermudah dalam menggambarkan data – data yang dimiliki. Wizard yang ada didalamnya juga akan membimbing dalam mengimpor atau mengekspor file dari atau ke format yang berbeda. Software ini juga menyediakan fasilitas backup dan restore data untuk menjaga keamanan data dari hal – hal yang dapat merusak data.

2.5 Rekayasa Perangkat Lunak (Software Engineering)

Subbab ini menjelaskan lebih rinci tentang rekayasa perangkat lunak, karakteristik – karakteristik yang dimiliki oleh rekayasa perangkat lunak, dan model *sekuensial linear*. Model *sekuensial linear* adalah metode yang digunakan dalam rancangan *web* kami ini. Menurut Pressman (1997, pp10-13) *software* adalah

perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan.

2.5.1 Pengenalan Rekayasa Perangkat Lunak

Struktur data yang memungkinkan program memanipulasi informasi secara proporsional, dan dokumen yang menggambarkan operasi dan kegunaan program.

Software memiliki dua peran, disatu sisi berfungsi sebagai sebuah produk dan disisi lain sebagai kendaraan yang mengantarkan sebuah produk. Sebagai produk, *software* mengantarkan potensi perhitungan yang dibangun oleh *software* komputer. Baik didalam sebuah telepon seluler, atau beroperasi disebuah *mainframe* komputer, *software* merupakan *transformer* informasi yang memproduksi, mengatur, memperoleh, memodifikasi, menampilkan, atau menancarkan informasi, dimana pekerjaan ini dapat sesederhana suatu bit tunggal atau sekompleks sebuah simulasi media. Sedangkan peran sebagai dasar untuk kontrol komputer (system operasi), komunikasi informasi (jaringan), dan penciptaan serta kontrol dari program – program lain (peranti dan lingkungan *software*).

2.5.2 Karakteristik Rekayasa Perangkat Lunak

Software secara umum dapat dibagi dua yaitu *software* sistem dan *software* aplikasi. *Software* sistem dapat dibagi lagi menjadi tiga macam, yaitu :

1. Bahasa pemrograman

Bertugas mengkonversikan arsitektur dan algoritma yang dirancang manusia ke dalam format yang dapat dijalankan komputer, contoh bahasa pemrograman diantaranya : BASIC, COBOL, Pascal, C++, FORTRAN.

2. Sistem Operasi

Saat komputer pertama kali dihidupkan, sistem operasi yang pertama kali dijalankan, sistem operasi yang mengatur seluruh proses menterjemahkan masukan, mengatur proses internal, manajemen penggunaan memori dan memberikan keluaran ke peralatan yang bersesuaian, contoh sistem operasi : DOS, Unix, Windows 95, IMB OS/2, Apple's System 7.

3. *Utility*

Software system dengan fungsi tertentu, misalnya pemeriksaan perangkat keras (*hardware troubleshooting*), memeriksa disklet yang rusak (bukan rusak fisik), mengatur ulang isi *harddisk* (partisi, defrag), contoh *Utility* adalah *Norton Utility*. *Software* aplikasi merupakan bagian *software* yang sangat banyak dijumpai dan terus berkembang.

Software lebih merupakan elemen logika dan bukan merupakan elemen sistem fisik. *Software* memiliki cirri yang berbeda dari *hardware*, yaitu :

1. *Software* dibangun dan dikembangkan, tidak dibuat dalam bentuk yang klasik.
2. *Software* tidak pernah usang.
3. Sebagian besar *software* dibuat secara *custom-built*, serta tidak dapat dirakit komponen yang sudah ada.

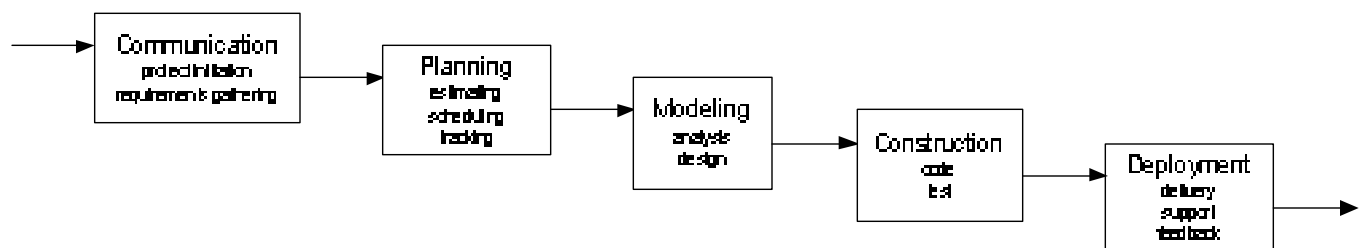
Karakteristik dari produk *software* :

1. *Maintainability*

2. *Dependability*
3. *Efficiency*
4. *Usability*

2.5.3 Model Proses Software

Menurut Pressman (2005, p29-47), ada beberapa model proses *software* yang umum digunakan, salah satunya adalah Model Sekuensial Linear. Sekuensial Linear ini juga dikenal dengan nama “*Classic Life Cycle*” atau “*Waterfall Model*”. Model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Model ini melingkupi aktifitas - aktifitas seperti ditunjukkan pada gambar 2.1 berikut ini.



Gambar 2.1 Waterfall Model

a) Rekayasa dan Pemodelan Sistem Informasi

Rekayasa dan pemodelan sistem informasi diperlukan karena perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen system dan

mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut.

Perangkat lunak harus berhubungan dengan elemen – elemen yang lain seperti perangkat lunak, manusia, dan *database*.

b) Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, seorang perancang perangkat lunak harus memahami kebutuhan informasi, fungsi-fungsi, unjuk kerja, dan *interface* yang diperlukan.

c) Perancangan

Design perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat *attribute* sebuah program, yaitu struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail algoritma.

d) Pengkodean

Design harus diterjemahkan menjadi bentuk yang dapat dibaca atau dimengerti oleh komputer, biasanya dalam bentuk pemrograman.

e) Pengujian

Sekali kode dibuat, pengujian program juga dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil actual yang sesuai dengan hasil yang dibutuhkan.

f) Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya, atau pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasi lagi setiap fase sebelumnya, lalu memperbaiki program sebelumnya dan tidak membuat yang baru lagi.

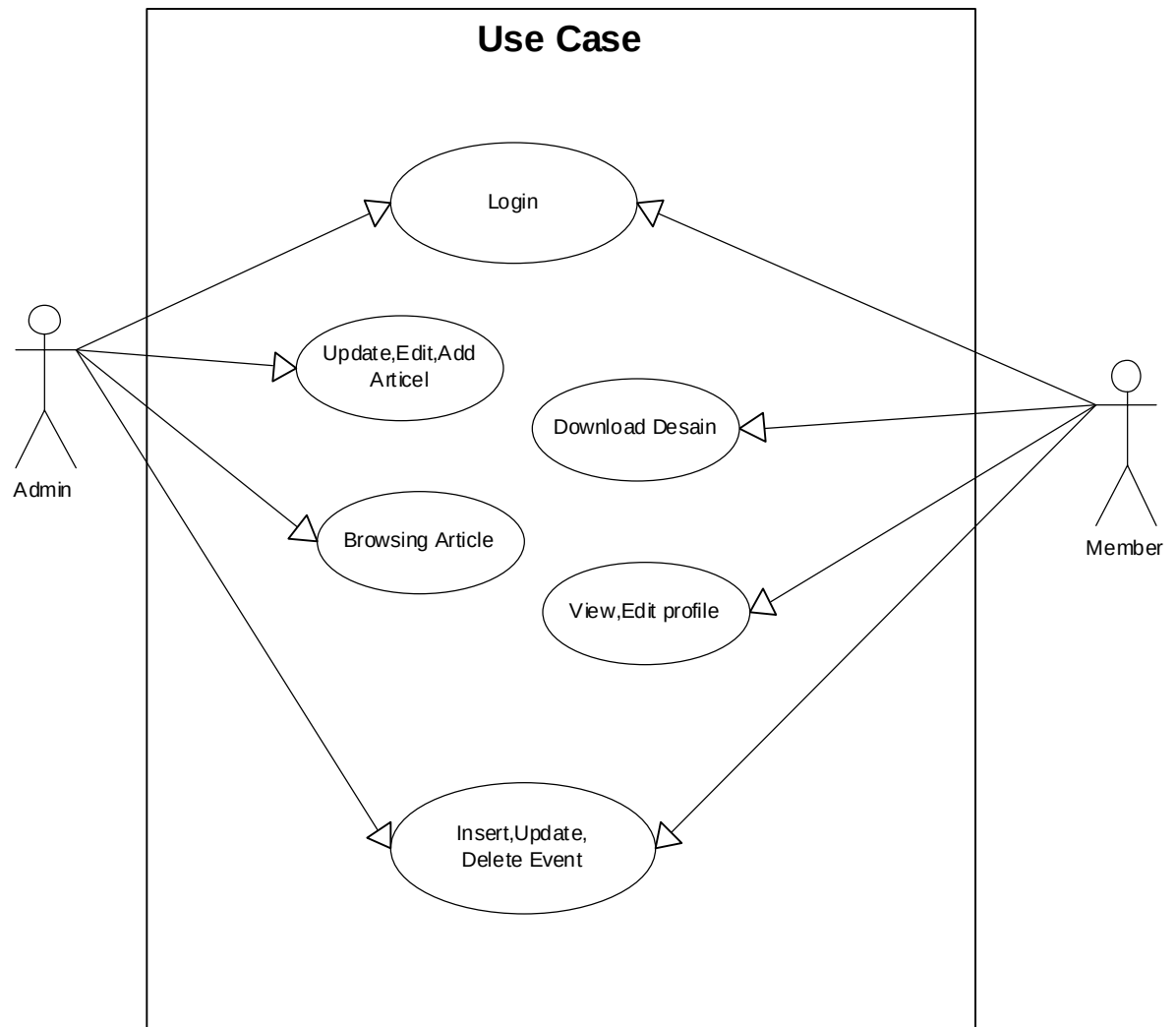
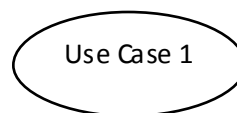
2.6 Unified Modelling Language (UML)

Pada tahun 1994, Grady Booch dan James Rumbaugh sepakat bergabung untuk menggunakan metode pengembangan berorientasi objek dengan tujuan membuat proses standart tunggal untuk pengembangan sistem berorientasi objek. Ivar Jacobson bergabung pada tahun 1995, dan mereka bertiga fokus membuat sebuah bahasa pemodelan objek standart sebagai ganti dari pendekatan atau metode berorientasi onjek standart (pada saat menuliskan, mereka bertiga memasarkan metodologi pemodelan objek yang disebut *Unified Process*). Berdasarkan kerja mereka dan hasil kerja lainnya pada industry, *Unified Modeling Process* (UML) versi 1.0 dirilis pada tahun 1997.

UML tidak menentukan metode untuk sistem – sistem pengembangan, hanya catatan yang saat ini telah diterima luas sebagai standart untuk pemodelan objek. *Object Management Group* (OMG), badan standart industry, mengadopsi UML pada bulan November 1997 dan terus bekerja untuk meningkatkannya berdasarkan kebutuhan industri.

2.6.1 *Use-case diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Sebuah contoh diagram *use-case* ditunjukkan pada Gambar 2.2

Gambar 2.2 Contoh Diagram Model *Use-case*Gambar 2.3 Simbol *Use-case*

Use-case merupakan urutan langkah yang secara tindakan saling terkait (skenario), baik termotifasi maupun secara manual, untuk melengkapi satu tugas bisnis tunggal.

2.6.2 Deskripsi Use-case

Bagian umum kejadian use-case adalah deskripsi use-case, yaitu langkah demi langkah mulai dengan pelaku menganalisis use-case dan melanjutkan hingga akhir kejadian bisnis. Dimana di dalam deskripsi use-case terdapat:

- Nama use-case, merupakan nama dari suatu use-case.
- Aktor (actor), pelaku yang berpartisipasi di dalam suatu use-case.
- Tujuan (goal), tujuan dari use-case.
- Prakondisi (precondition), batasan pada keadaan system sebelum use-case dapat di eksekusi.
- Ringkasan (summary), deskripsi ringkasan pendek yang berisi sejumlah kalimat yang menunjukan secara garis besar tujuan use-case dan berbagai kegiatannya.
- Related use-case, use-case yang saling berhubungan.
- Langkah-langkah (steps), menjelaskan setiap langkah dari use-case dengan menggunakan dua kolom, dimana kolom sebelah kiri menunjukan aksi dari pelaku dan kolom sebelah kanan merupakan respon yang di berikan oleh sistem.

2.6.3 Activity Diagram

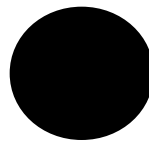
Activity diagram / diagram aktifitas adalah sebuah diagram yang dapat digunakan untuk menggambarkan secara logika behavior grafis aliran proses bisnis, langkah-langkah sebuah use-case atau (metode) object. Diagram ini serupa dengan flowchart, dimana secara grafis diagram ini menggambarkan aliran sekuebsial dari kegiatan baik itu

proses bisnis atau sebuah use-case. Diagram ini berbeda dari flowchart dimana diagram ini menyediakan sebuah mekanisme untuk menggambarkan kegiatan yang tampak secara paralel.

Karena itu, diagram ini sangat berguna untuk memodelkan kegiatan yang akan dilakukan saat sebuah operasi dieksekusi dan untuk memodelkan hasil – hasil dari kegiatan- kegiatan ini – seperti memodelkan event yang menyebabkan window akan di tampilkan atau ditutup.

Dibawah ini merupakan symbol-simbol yang ada pada diagram aktivitas:

- a. *Initial state* dengan menggambarkan awal dari sebuah proses



Gambar 2.4 *Initial State*

- b. *State*, segi empat bersudut tumpul menggambarkan sebuah kegiatan atau tugas yang perlu di lakukan.



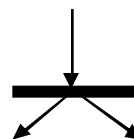
Gambar 2.5 *State*

- c. *Control flow*, panah menggambarkan sasaran yang mengawali kegiatan.



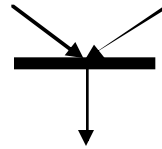
Gambar 2.6 *Control flow*

- d. *Transition (fork)*, merupakan bar sinkronisasi dimana kegiatan dapat dilakukan secara paralel.



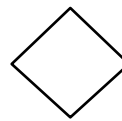
Gambar 2.7 *Transition (fork)*

- e. *Transition (join)*, merupakan bar sinkronisasi dimana dua aktifitas yang mempunyai tujuan yang sama dan tujuan dari keduanya digabung menjadi satu.



Gambar 2.8 *Transition (join)*

- f. *Decision*, menggambarkan sebuah kegiatan keputusan



Gambar 2.9 *Decision*

- g. *Final State*, menggambarkan akhir dari sebuah proses



Gambar 2.10 *Final State*

2.6.4 *Class Diagram*

Class diagram menggambarkan struktur objek sistem. Diagram ini menunjukkan kelas objek yang menyusun sistem dan juga hubungan antara kelas objek tersebut. Di dalam diagram kelas terdapat *class*, *attribute* dan *behavior*. Gambar 2.11 merupakan contoh dari *class diagram*. Symbol panah menunjukkan asosiasi.

Gambar 2.11 Contoh dari *class diagram*

Pada contoh diatas, satu mahasiswa bisa memiliki banyak buku.

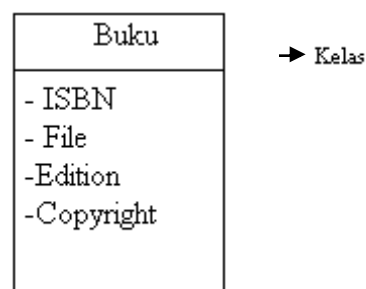
a. Object

Object adalah sesuatu yang nyata atau dapat dilihat, disentuh, atau dirasakan.

User menyimpan data serta mencatat perilaku mengenai sesuatu itu.

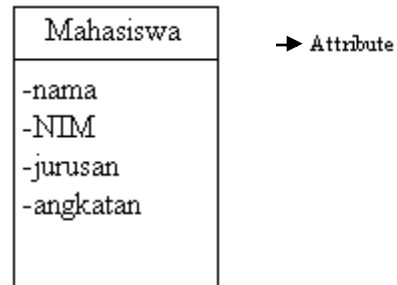
b. *Class* / kelas

Class merupakan satu set object yang memiliki *attribute* dan *behavior* yang sama, yang biasa disebut dengan object *class*. Di bawah ini merupakan contoh dari sebuah *class* / kelas.

Gambar 2.12 *Class* dalam UML

c. *Attribute*

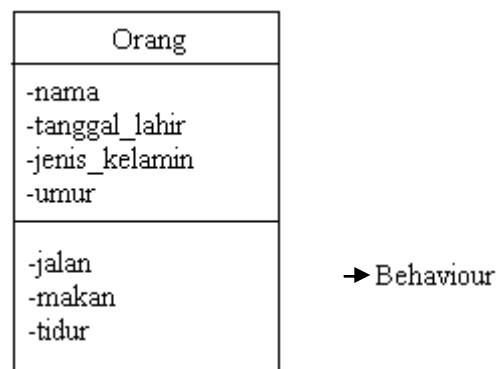
Data yang mewakili karakteristik tentang suatu objek. Perhatikan contoh dari *attribute* yang ada pada kelas orang.



Gambar 2.13 *Attribute* dari kelas orang

d. *Behavior*

Behavior merupakan kumpulan dari sesuatu yang dapat dilakukan oleh obyek dan terkait dengan fungsi-fungsi yang bertindak pada data objek. Pada siklus berorientasi objek, perilaku objek merujuk kepada metode, operasi, atau fungsi. Perhatikan Gambar 2.14



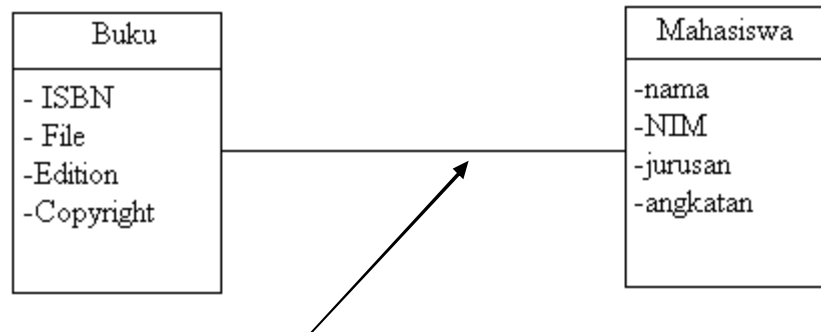
Gambar 2.14 *Behaviour* dari kelas orang

Beberapa symbol lain yang ada pada *class* diagram adalah:

a. *Associations*

Menunjukkan hubungan antara kelas yang satu dengan kelas yang lain.

Perhatikan Gambar 2.15

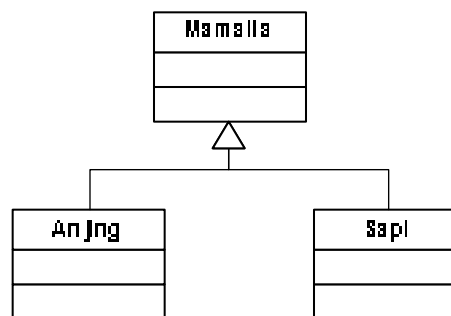


Garis mengindikasikan hubungan / *Associations*

Gambar 2.15 Hubungan antara *class* Buku dan Mahasiswa

b. *Generalization*

Generalization merupakan sebuah teknik antara *attribute* dan *behavior* yang umum pada beberapa tipe kelas objek, dikelompokkan ke dalam kelasnya sendiri. Perhatikan Gambar 2.16



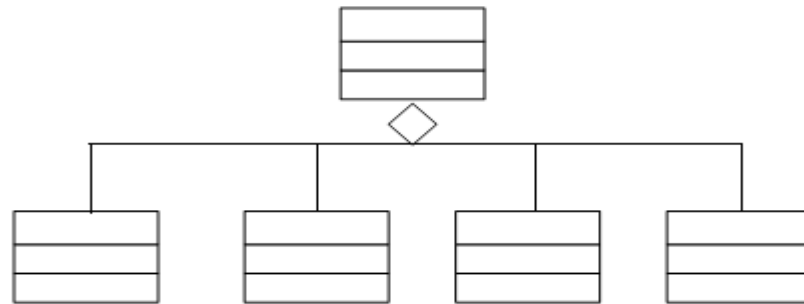
Gambar 2.16 Hubungan generalisasi

c. *Aggregation*

Terkadang sebuah kelas dapat mempunyai beberapa kelas komponen.

Dan hubungan antara kelas dengan kelas komponennya itu bernama

Aggregation.

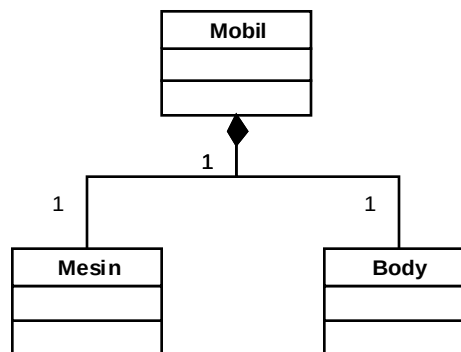


Gambar 2.17 Hubungan *Aggregation*

d. *Composition*

Composition adalah sebuah tipe *aggregation* yang kuat. Artinya setiap

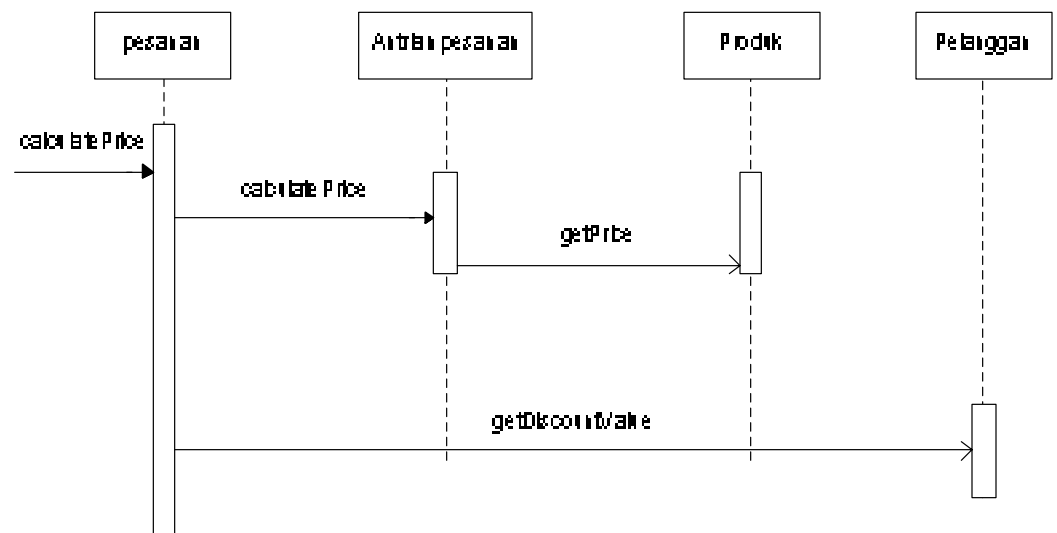
kelas komponen hanya dimiliki oleh satu kelas saja.



Gambar 2.18 *Composition Relationship* dalam UML

2.6.5 Sequence Diagram

Diagram rangkaian menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah use-case atau operasi, diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi apa. Perhatikan Gambar 2.19 merupakan contoh dari *sequence diagram*.

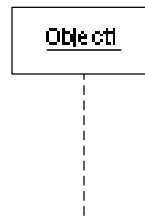


Gambar 2.19 Sequence Diagram

Di bawah ini merupakan symbol-simbol yang ada pada *sequence diagram*:

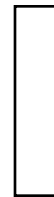
a. *Object lifeline*

Menggambarkan panjang kehidupan suatu objek selama scenario sedang dibuat.

Gambar 2.20 *Object lifeline*

b. *Activation*

Dimana proses sedang di lakukan oleh *object* / *class* untuk memnuhi *pesan*/ perintah

Gambar 2.21 *Activation symbol*

c. *Message*

Sebuah anak panah yang mengidentifikasi pesan di antara objek. Dan objek dapat mengirimkan pesan ke dirinya sendiri.

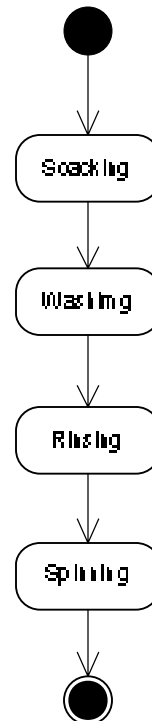
Gamabar 2.22 *Message symbol*

2.6.6 *Statechart Diagram*

Sebuah *state diagram* (Iethbridge abd Laganiere, 2002, p276), merupakan cara lain untuk mengekspresikan informasi dinamis tentang sebuah system, diagram ini digunakan untuk menggambarkan fungsi eksternal yang terlihat dari

sebuah system atau dari sebuah objek secara individu (*a state diagram is another way of expressing dynamic information about system; it us used to describe the extemally visible behavior of a system or of an individual object*).

Contoh *statechart diagram* dapat dilihat pada gambar 2.23



Gambar 2.23 *State diagram*

2.7 E - Commerce

Menurut Minoli (1998, p.5), *electronic commerce* yang biasa disingkat *e – commerce* merupakan suatu penggabungan simbiotik dari komunikasi, manajemen data, dan keamanan untuk memungkinkan aplikasi bisnis dalam perusahaan yang berbeda secara otomatis melakukan pertukaran informasi yang kaitannya dengan penjualan barang atau jasa. Atau secara garis besar adalah kegiatan transaksi jual –

beli yang dilakukan secara elektronik melalui suatu jaringan internet. *E – commerce* mencakup beberapa metode yang menghubungkan penjual dan pembeli, antara lain periklanan, demo produk, catalog produk, dan petunjuk – petunjuk. Berikut ini adalah tujuan dari *E-Commerce* :

- Segala kegiatan jual – beli dapat dilakukan dimana saja dengan akses internet.
- Membuat wadah tempat berkumpulnya komunitas dengan membangun basis komunitas, membangun konsep pasar menjadi bukan hanya sekedar tempat jual beli, dan sebagai pusat informasi.
- Informasi yang up to date, komunikasi multi – arah yang dinamis.

Adapun kelebihan yang dapat dirasakan dari sistem *E-Commerce* adalah :

- Otomasisasi, menggantikan proses manual.
- Integrasi, meningkatkan efisiensi & efektifitas proses.
- Publikasi, memberikan jasa promosi & komunikasi atas produk dan jasa yang dipasarkan.
- Interaksi, pertukaran data atau informasi antar berbagai pihak yang akan meminimalkan “*human error*”.
- Transaksi, kesepakatan antara dua pihak untuk melakukan transaksi yang melibatkan institusi lain.

2.7.1 *Business to Business (B2B)*

Business to Business atau yang biasa disingkat B2B merupakan jenis transaksi antar dua perusahaan (bisnis). Ini dikarenakan transaksi antar dua perusahaan biasanya dilakukan berulang – ulang dan dalam ukuran yang besar, seperti antara produsen dan grosir, atau antara grosir dan pengecer. B2B berfokus pada system distribusi suatu produk hingga sampai kepada konsumen. Proses ini melalui beberapa tahapan, sehingga harga suatu produk akan terus meningkat seiring bertambahnya proses distribusi.

Sebagai contoh, sebuah produsen mobil membuat beberapa transaksi B2B seperti membeli ban, kaca untuk kaca depan, dan selang karet untuk kendaraan. Sebagai rangkaian penutup, kendaraan yang sudah selesai dijual kepada konsumen, adalah satu (B2C) transaksi.

2.7.2 *Business to Customer (B2C)*

Pengertian *Business to Customer* adalah konsep bisnis dimana penjual berhubungan langsung dengan para pembelinya dan antar para penjual dan pembeli belum tentu kenal dengan penjualnya. Sifat dari bisnis tersebut adalah terbuka dan bebas dimanfaatkan oleh khayalak umum.

2.8 Bauran Pemasaran (*Marketing mix*)

Menurut Kotler dan Armstrong (2008. P62), bauran pemasaran adalah kumpulan alat pemasaran taktis terkendali yang dipadukan perusahaan untuk menghasilkan respon yang diinginkannya di pasar sasaran. Bauran pemasaran terdiri dari beberapa komponen (4P), yaitu:

- *Product* (produk), kombinasi produk dan jasa yang ditawarkan perusahaan kepada pasar sasaran.
- *Price* (Harga), jumlah uang yang harus dibayarkan pelanggan untuk memperoleh produk.
- *Place* (Tempat), kegiatan perusahaan yang membuat produk bisa tersedia bagi pelanggan sasaran.
- *Promotion* (Promosi), aktivitas yang menyampaikan manfaat produk dan membujuk pelanggan untuk membelinya.

Program pemasaran efektif memadukan semua elemen bauran pemasaran ke dalam suatu program pemasaran terintegrasi yang dirancang untuk mencapai tujuan pemasaran perusahaan dengan menghantarkan nilai bagi konsumen. Bauran pemasaran merupakan sarana taktis perusahaan untuk menentukan *positioning* yang kuat dalam pasar sasaran.

2.8.1 Pengaruh Internet Dalam Bauran Pemasaran

Internet menyediakan kesempatan untuk meragamkan elemen dari bauran pemasaran pada:

- Produk – fitur dari produk dapat divariasikan: secara umum, layanan konsumen dan nilai merek dapat ditingkatkan. Informasi baru berbasis produk dapat disediakan oleh internet seperti informasi pasar khusus dalam *subscription*. Beragam merek dapat dihasilkan untuk beberapa pasar.

- Harga – menggunakan internet sebagai saluran retail baru memungkinkan pengurangan harga produk sejak jumlah produk dan biaya distribusi melalui jaringan belanja tradisional berkurang.
- Promosi – internet menawarkan komunikasi pemasaran tambahan baru dimana menginformasikan konsumen mengenai keuntungan produk dan mempengaruhi konsumen dalam keputusan pembelian.
- Tempat – internet juga menyediakan saluran penjualan baru untuk mendistribusikan produk melalui elektronik.
- Orang – internet dapat dipikirkan sebagai marginalisasi peran dari kontak *customer* langsung dari organisasi, namun digunakan secara luas untuk membantu merekrut staf yang berkualitas.
- Proses – internet harus diintegrasikan dengan proses pemasaran lain seperti *telemarketing* dan pemasaran langsung dan proses berbeda dari rantai nilai agar sukses.