BAB II
TINJAUAN PUSTAKA

1.1 Konsep Sistem Informasi

1.1.1 Sistem

Sistem dapat didefinisikan sebagai kumpulan orang yang saling bekerja sama dengan ketentuan aturan yang terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan. Sistem memiliki beberapa karakteristik atau sifat yang terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem pengolahan sistem dan sasatran sistem.[3] Sehingga dapat disimpulkan bahwa sistem adalah kumpulan komponen dan elemen yang terintegasi dan bekerja bersama untuk mencapai suatu tujuan tertentu.

Suatu sistem yang baik memiliki karakteristik sebagai berikut: [4]

1. Komponen

   Suatu sistem terdiri dari sejumlah komponen komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan sistem (*boundary*)

   Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan luar sistem (*environment*)

   Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

4. Penghubung sistem (*interface*)

   Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber data mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan sistem (*input*)

   Masukan adalah energi yang dimasukkan ke dalam sistem yang dapat berupa perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer, program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran sistem (*output*)

   Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembangunan. Contoh, komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah sistem

   Suatu sistem menjadi bagian pengolah yang akan mengubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran sistem

   Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

1.1.2 Informasi

Informasi adalah hasil pengolahan data yang sudah dapat diterima oleh akal pikiran dari penerima informasi yang dapat digunakan untuk pengambilan keputusan. Bahan mentah berupa data diolah dengan metode tertentu untuk menghasilkan informasi. Informasi tersebut disampaikan, lalu digunakan oleh si penerima untuk membuat keputusan atau melakukan tindakan yang akan menghasilkan data baru lagi.[3]

Informasi yang baik untuk digunakan dibagi menjadi 3 (tiga) bagian:[5]

a. Akurat: informasi harus berdasarkan fakta yang sebenarnya, bukan isu, dugaan, atau opini yang menyesatkan.
b. Tepat waktu: penerima tidak terlambat mendapatkan informasi karena informasi yang sudah usang tidak bernilai lagi, terutama jika informasi digunakan untuk mengambil keputusan.
c. Relevan: informasi tersebut memiliki keterkaitan dan bermanfaat secara langsung bagi penerimanya. Sebuah informasi yang relevan bagi seseorang belum tentu relevan untuk

lainnya.

1.1.3 Sistem Informasi

Sistem informasi merupakan gabungan dari *hardware, software, brainware*, prosedur dan aturan yang terorganisasi secara integral untuk mengolah data menjadi informasi yang bermanfaat untuk memecahkan masalah dan pengambilan keputusan.[3] Dalam arti yang sangat luas, istilah sistem informasi sering digunakan merujuuk kepada interaksi antara orang, proses algoritma, data dan teknologi.

Komponen fisik sistem informasi dibagi menjadi 4 (empat):[3]

1. Personal (*humanware*): pelaksana manajerial, *data entry operator, computer operator, programmer, system analyst,* dan *database administrator.*
2. Prosedur (*organiware*): kebijakan formal dan petunjuk untuk mengoperasikan sistem. Terdiri dari tatalaksana, prosedur pengolahan data, dan pedoman pemakai.
3. Perangkat pengolahan data (*technoware*): *hardware, software,* perangkat pendukung seperti jaringan komputer, sistem komunikasi, dan lainnya.
4. Data (*inforware*): *database*

Berdasarkan pernyataan di atas, dapat disimpulkan bahwa sistem informasi merupakan suatu sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi.

Secara umum berdasarkan perkembangan dari teknologi dan *software* bahasa pemograman dan *software* basis data, maka bentuk sistem informasi masa kini terbagi menjadi 3 jenis, yaitu: [6]

1. Sistem Informasi Berbasis *Desktop*

Sistem informasi ini terbentuk dari bahasa pemograman yang bersifat visual dan selanjutnya di compile sehingga terbentuklah file *setup* untuk bisa diinstalasi pada *PC Server* atau *PC stand alone* tergantung kepada kebutuhan *user*. Umumnya bahasa pemograman yang dipakai adalah produk dari Microsoft seperti Visual Basic, C#, C++. Contohnya sistem informasi penggajian dan absensi kehadiran.

Kelebihan Sistem Informasi Berbasis *Desktop*:

a. Dapat berjalan dengan independen, tanpa perlu menggunakan *browser*.

b. Tidak perlu koneksi internet, karena semua file yang diperlukan untuk menjalankan

aplikasinya sudah terinstal sebelumnya.

c. Dapat dengan mudah memodifikasi settingannya.

d. Prosesnya lebih cepat.

Kekurangan Sistem Informasi Berbasis *Desktop*:

a. Apabila akan menjalankan aplikasi, harus diinstal terlebih dahulu di komputer.

b. Bermasalah dengan lisensi. Hal ini membutuhkan lisensi yang banyak pada setiap *computer*.

c. Aplikasi tidak dapat dibuka di *computer* lain, jika belum diinstall

d. Biasanya memerlukan *hardware* dengan spesifikasi tinggi.

2. Sistem Informasi Berbasis *Web*

Penggunaan sistem informasi berbasis *web* saat ini telah berkembang pesat dikarenakan penggunaannya yang *friendly* dan tetunya menjdi trendy dikalangan pengguna. Secara umum, sistem informasi ini bersifat *open source* secara *coding* dan biasanya dibuat dengan menggunkan bahasa pemograman HTML, ASP, PHP dan dapat dikombinasikan dengan CSS dan *JavaScript*. Misalnya penerapan sistem informasi *online shop*, sistem informasi akademik dan lain-lain.

Kelebihan Sistem Informasi Berbasis *Web*:

a. Kita dapat menjalankan aplikasi berbasis web dimanapun kapanpun tanpa harus melakukan penginstalan.

b. Terkait dengan isu lisensi (hak cipta), kita tidak memerlukan lisensi ketika digunakan.

c. Dapat dijalankan di system operasi manapun. Tidak perduli apakah kita menggunakan linux, windows, aplikasi berbasis web dapat dijalankan asalkan kita memiliki browser dan akses internet.

d. Dapat diakses lewat banyak media seperti: *computer, handheld* dan *handphone* yang sudah sesuai dengan standard WAP.

e. Tidak perlu spesifikasi *computer* yang tinggi untuk menggunakan aplikasi berbasis *web* ini, sebab di beberapa kasus, sebagian besar proses dilakukan di *web server* penyedia aplikasi berbasis *web* ini.

Kelemahan Sistem Informasi Berbasis *Web*

a. Dibutuhkan koneksi intranet dan internet yang handal dan stabil, hal ini bertujuan agar

pada saat aplikasi dijalankan akan berjalan dengan baik dan lancar.

b. Dibutuhkan *system* keamanan yang baik dikarenakan aplikasi dijalankan secara terpusat, sehingga apabila *server* di pusat *down* maka *system* aplikasi tidak bias berjalan.

3. Sistem Informasi Berbasis *Mobile*

Sistem informasi ini seperti halnya sistem informasi berbasis *web* yang populer dikalangan pengguna. Salah satu contoh sistem informasi berbasis *mobile* adalah pengguna *smartphone* yang terintegrasi dengan bahasa pemograman dengan bentuk *Java* atau *Eclips*. Misalnya penerapan sistem informasi pada Gojek, Grab Bike dan lain-lain.

Kelebihan Sistem Informasi Berbasis *Mobile*:

a. Pengembangan aplikasi berbasis *mobile* adalah *multitasking*. Jadi seorang pengguna dapat menggunakan dua atau lebih dari dua aplikasi pada saat yang bersamaan.

b. Adanya fitur notifikasi yang memudahlan pengguna dalam menggunakan sistem informasi berbasis *mobile.*

c. Ketersediaan layanan *Google* yang menyediakan berbagai layanan seperti Pustaka *Google, Google Drive, Gmail*, Dokumen, yang datang sebagai bagian dari sistem operasi *Android*. Sehingga membaca *email* dan memeriksa dokumen menjadi sangat mudah.

Kekurangan Sistem Informasi Berbasis *Mobile*:

a. Dibutuhkan koneksi internet yang handal dan stabil, hal ini bertujuan agar pada saat aplikasi dijalankan akan berjalan dengan baik dan lancar.

b. Terlalu banyak iklan yang diposting di *play store* yang semakin memperlambat laju sistem operasi ini.

1.2 Konsep Dasar *Web*

1.2.1 *Website*

Halaman *web* yang saling berhubungan yang umumnya berada pada *server* yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok atau organisasi. Sebuah situs *web* biasanya ditempatkan setidaknya pada sebuah server *web* yang dapat diakses melalui jaringan seperti internet, ataupun *Local Area Network* (LAN) melalui alamat internet yang dikenali sebagai *Uniform Resource Locator* (URL).[6] *Website* merupakan sebuah media yang memiliki banyak halaman yang saling terhubung (*hyperlink*), dimana *website* memiliki fungsi dalam memberikan informasi berupa teks, gambar, video, suara dan

animasi atau penggabungan dari semuanya. Karateristik utama yang dimiliki oleh *website* adalah halaman-halaman yang saling terhubung, dan dilengkapi dengan domain sebagai alamat URL atau *World Wide Web* (www) dan juga *hosting* sebagai media yang menyimpan banyak data. *Website* dapat diakses menggunakan jaringan internet dengan *platform* yang disebut *browser*, seperi *chrome, mozilla firefox, internet explorer* (IE) dan sebagainya. *Website* dapat dibangun dalam mode *localhost*, yang artinya *website* dapat dirancang, dibangun dan dimodifikasi tanpa menggunakan jaringan internet.[7]

1.2.2 Bagian *Website*

1. *Website* Statis

Website statis adalah sebuah *web* dimana halamannya tidak berubah. Artinya untuk melakukan perubahan pada suatu halaman pada *website* harus dilakukan secara manual dengan mengedit *source code* yang menjadi struktur dari *website* tersebut. Halaman-halaman *web* tersebut tidak memliki *database*, data dan informasi yang ada pada *web* statis tidak berubah-ubah kecuali diubah sintaksnya. Dokumen *web* yang dikirim kepada *client* akan sama isinya dengan apa yang ada di *web server*. Misalnya *web* yang berisi profil perusahaan yang isinya hanya ada beberapa halaman saja dan kontennya hampir tidak pernah berubah karena konten langsung diletakan dalam file HTML.[7]

2. *Website* Dinamis

Website dinamis merupakan *website* yang secara struktur digunakan untuk *update* sesering mungkin, interaksi yang terjadi antara pengguna dan *server* sangat kompleks. Seseorang bisa mengubah konten dari halaman tertentu dengan menggunakan *browser*. *Request* (permintaan) dari pengguna dapat diproses oleh *server* yang kemudian ditampilkan dalam isi yang berbeda-beda menurut programnya. Halaman-halaman *web* tersebut memiliki *database*. *Web* dinamis memiliki data dan informasi yang berbeda-beda tergantung *input* apa yang disampaikan *client*. Dokumen yang sampai di *client* akan berbeda dengan dokumen yang ada di *web server*. *Website* dinamis pada umumnya terdiri dari halaman *front- end* yang dapat diakses oleh *user*, juga disediakan halaman *back-end* untuk mengedit *content* dari *website*.[7]

1.2.3 Front-End

Front-end atau biasa sering dikenal dengan pengembang pada sisi klien adalah orang yang fokus pekerjaanya pada tampilan *website*, atau bagian yang lagsung dilihat oleh *user*, nuansa dan pengalaman pengguna (UX) pada sebuah situs *website*.[8] Berikut adalah bagian

dari *Front-End*:

1. HTML

Hypertext Markup Language (HTML) adalah bahasa *markup* standar untuk membuat halaman *web* dan situs. HTML berisi skrip berupa tag-tag untuk membuat dan mengatur struktur *website*. Menuliskan tag-tag HTML tidak hanya sebatas memasukkan perintah-perintah tertentu agar HTML dapat di akses oleh *browser*. Tujuan dari *web browser* (*Chrome, Edge, Firefox, Safari*) adalah untuk membaca dokumen HTML dan menampilkannya dengan benar. Mendesain HTML adalah adalah sebuah seni tersendiri. *Homepage* yang merupakan *implementasi* dari HTML adalah refleksi dari orang yang membuatnya. Mendesain HTML dapat dilakukan dengan dua cara:

a. Menggunakan HTML Editor , seperti *Microsoft Front Page, Adobe Dreamweaver* dan lain-lain.
b. Dengan cara menuliskan sendiri secara manual satu persatu tag-tag HTML ke dalam dokumen HTML, seperti *notepad*.

   Beberapa tugas utama HTML dalam membangun *website* yaitu menentukan *layout website*, memformat teks dasar, membuat *list*, membuat tabel, menyisipkan gambar, video dan audio dan membuat link.[8]

2. CSS

Cascading Style Sheet (CSS) biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa *markup*, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs. CSS merupakan *features* baru dari HTML 4.0, hal ini diperlukan setelah melihat perkembangan HTML menjadi kurang praktis karena halaman *web* (*web pages*) terlalu banyak dibebani hal-hal yang berkaitan dengan faktor tampilan seperti ukuran huruf (*font*) dan lain-lain. Untuk itu jika kumpulan pengaturan gaya (*style*) tersebut dikelola secara terpisah maka manajemen halaman (*pages*) menjadi lebih mudah dan efisen.[9]

3. JavaScript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang fungsional digunakan untuk menambahkan interaksi antara halaman *web* dengan pengunjung halaman *web*. *JavaScript* dijalankan pada sisi klien yang akan memberikan kemampuan fitur-fitur tambahan halama *web* yang lebih baik dibandingkan fitur fitur yang terdapat pada HTML.[9]

Berdasarkan penjelasan diatas maka dapat disimpulkan bahwa fungsi utamanya yaitu:

HTML : Memungkinkan untuk menambahkan konten ke halaman *web*.

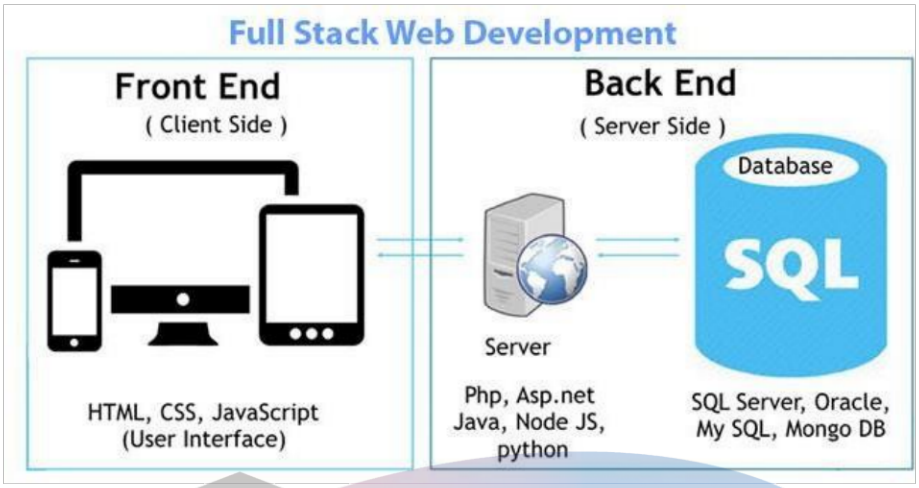
CSS : Menentukan *layout, style*, serta keselarasan halaman *website*.
JavaScript : Menyempurnakan tampilan dan sistem halaman *web*.

1.2.4 *Back-End*

Back-end dikenal sebagai orang yang khusus melakukan pengembangan *website* pada sisi server, *back-end* juga biasanya menjadi arsitek dari teknologi yang digunakan pada sebuah *platform*, pada back-end data ditambahkan, diubah atau dihapus. *Back-end* mencakup segala sesuatu yang biasanya tidak dilihat atau berinteraksi langsung kepada *user*, seperti *database* dan *server*.[8] *Back-end* yang seringkali digunakan adalah PHP dan MySQL sebagai *database*.

PHP merupakan singkatan dari *Hypertext Preprocessor* adalah bahasa pemograman *web server side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Ketika *web server* mendapat permintaan suatu halaman yang didalamnya terdapat perintah atau *script* PHP, maka tidak akan langsung dikirimkan ke *client*. *Web server* akan meminta bantuan PHP *Engine* untuk menerjemahkan setiap perintah PHP menjadi perintah HTML, CSS dan *Javascript* yang dapat dimengerti oleh *web browser client*. Jadi dalam hal ini, *client* selalu menerima halaman sudah dalam bentuk *client- side script.*[10]

MySQL adalah sebuah *Database Management System* (Manajemen Basis Data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. MySQL masuk ke dalam jenis RDBMS (*Relational Database Management System*). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah *database* terdapat satu atau beberapa tabel. SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*. Jika aplikasi *web* mengandung data yang berasal dari *server database* MySQL. Data akan diambil oleh PHP dari *database*, selanjutnya diterjemahkan dalam format HTML dan dikembalikan ke *web server*. *Web server* akan mengirimkannya ke *client* dalam bentuk yang "siap saji" di *web browser*.[10]



Gambar 1.1 *Front-End* dan *Back-End*

Pada gambar di atas, *client side* adalah program yang ada di *client side* dan *developer* yang membuat kode untuk *client* tersebut. Walaupun kode dibuat hanya di *server side*, namun *server* yang akan memproses kode tersebut untuk dikirimkan ke *client*. Program di *server side* akan dikonversi menjadi bahasa pemrograman *client side* seperti HTML, tergantung dengan bagaimana membuatnya di *server side*. Jadi, *user* yang berada di *client side* tidak akan bisa melihat *script* PHP sama sekali. Yang bisa mereka lihat hanyalah kode HTML, CSS, Javascript, dan kode pemrograman *client* lainnya.[11]

1.3 *E-Commerce*

Electronic Commerce (e-commerce) adalah proses pembelian, penjualan atau pertukaran produk, jasa dan informasi melalui jaringan komputer. *E-Commerce* merupakan bagian dari *e-business*, di mana cakupan *e-business* lebih luas. *E-business* melibatkan pemasaran, perancangan produk, evaluasi layanan konsumen dan lain-lain. Sedangkan *e-commerce* hanya mencakup transaksi bisnis seperti pembelian dan penjualan barang serta jasa melalui internet. Selain teknologi jaringan *www*, *e-commerce* juga memerlukan teknologi basis data atau pangkalan data (*database*), surat elektronik (*e-mail*) dan bentuk teknologi non komputer yang lain seperti halnya sistem pengiriman barang dan alat pembayaran untuk *e-commerce*.[12]

E-commerce merupakan salah satu sektor bisnis yang justru mampu meningkatkan pasar di era pandemi Covid-19. Pandemi ini justru menjadi faktor utama penguatan bisnis *e-commerce* di seluruh dunia terutama bidang retail dan grosir. Hal ini terbukti pada laporan yang dibuat oleh *Analytic Data Advertising* (ADA) yang mengatakan bahwa terjadi lonjakan penggunaan aplikasi *e-commerce* sebesar 300% dalam 1 bulan pertama Pembatasan Sosial

Berskala Besar (PSBB) diterapkan oleh pemerintah. Masyarakat banyak beralih menggunakan *e-commerce* dalam berbelanja kebutuhan pokok untuk menghindari kontak fisik dan mencegah penyebaran virus Covid-19 pada masyarakat lain. Selain itu menurut ADA, peningkatan penggunaan *e- commerce* juga terjadi pada aplikasi jual beli barang bekas yang mengindikasikan bahwa banyak masyarakat kecil dan juga pekerja yang terdampak pandemi ini sehingga terpaksa harus menjual barang-barang demi menyambung hidup.[1]

Sektor Usaha Mikro Kecil Menengah (UMKM) menjadi salah satu sektor yang terdampak dari bertumbuhnya *e-commerce* di Indonesia. UMKM memiliki kontribusi yang besar bagi perekonomian seuatu negara. Menurut *World Bank*, UMKM berkontribusi 60% terhadap *Gross Domestic Product* (GDP) Indonesia. UMKM bahkan menjadi salah satu sektor yang membantu pemerintah dalam membuka lapangan pekerjaan.[1]

E-commerce akan merubah semua kegiatan marketing dan juga sekaligus memangkas biaya-biaya operasional untuk kegiatan *trading* (perdagangan). Proses yang ada dalam *E-commerce* adalah sebagai berikut:

a. Presentasi elektronis (pembuatan *website*) untuk produk dan layanan.
b. Pemesanan secara langsung dan tersedianya tagihan.
c. Otomatisasi akun pelanggan secara aman (baik nomor rekening maupun nomor Kartu Kredit).
d. Pembayaran yang dilakukan secara langsung (*online*) dan penanganan transaksi

1.3.1 Ruang Lingkup *E-Commerce*

Ruang lingkup *e-commerce* diuraikan sebagai berikut:[1]

1. *Electronic Business*: terdiri dari lingkup perdagangan atau transaksi jual beli dilakukan melalui media internet. Cara untuk memahami batasan *e-commerce* adalah dengan melihat fenomena dari berbagai dimensi:
2. Teknologi: kontributor terbesar yang memungkinkan *e-commerce* adalah teknologi informasi, teknologi komputer dan telekomunikasi.
3. Pemasaran dan proses konsumen baru: dari segi pemasaran, jangkauan sebuah transaksi menjadi semakin luas karena yang dapat memasarkan produk dan jasa ke seluruh dunia tanpa memperhatikan batasan-batasan geografis.
4. Hubungan Elektronik: dengan adanya *e-commerce*, maka dua buah divisi dapat bekerja

secara efisien melalui pertukaran data elektronis/*Information Value Adding*. Di dalam *e-commerce*, bahan baku yang utama adalah informasi.

5. Pembuatan Pasar: keberadaan perdagangan aunsaue yang mempertemukan berjuta- juta penjual dan pembeli di sebuah pasar digital maya (*e-market*).
6. *Service Infrastructure*: konsep *e-commerce* membuahkan mekanisme transaksi jual beli dan jasa-jasa yang diperlukan sebagai sarana pendukung aktivitas jual beli produk tersebut.
7. *Electronic Commerce*: dilakukan secara elektronik di mana di dalamnya termasuk: Perdagangan via internet (*internet commerce*), Perdagangan dengan fasilitas *web* internet (*web e-commerce*), dan perdagangan dengan sistem pertukaran data terstruktur secara elektronik (*Elektronik Data Interchange*/EDI).

1.3.2 Jenis-Jenis *E*-Commerce

Jenis-jenis *e-commerce* dibagi menjadi empat bagian yaitu:[1]

1. *Business to business* (B2B) - Jenis di mana sebuah perusahaan menjual produk atau jasa kepada perusahaan lainnya. Dalam model *e-commerce* ini, biasanya pembeli memesan barang dalam jumlah besar. Contohnya adalah sebuah perusahaan yang membeli perlengkapan kantor dari sebuah produsen.
2. *Business to consumer* (B2C) - Dalam jenis *ecommerce* ini, sebuah perusahaan menjual produk atau jasa kepada konsumen. Pada umumnya, pelanggan dalam *e- commerce* B2C hanya mengecer. Jika anda pernah membeli dari suatu toko *online*, aktivitas tersebut termasuk dalam golongan ini.
3. *Consumer to consumer* (C2C) - C2C adalah transaksi *online* antara dua individu. Misalnya 2 orang yang ingin menjual barang bekas di internet. *Consumer to business* (C2B) - Berkebalikan dengan B2C, *e-commerce* C2B adalah skenario di mana seseorang menjual produk atau layanan kepada sebuah perusahaan. Seorang desainer grafis, misalnya, menawarkan dan menjual logo buatannya kepada sebuah bisnis makanan.

1.3.3 Tujuan & Manfaat *E-Commerce*

Dengan menggunakan *e-commerce*, produsen dapat merubah daftar harga atau melakukan kustomisasi produk atau jasa yang ditawarkan dan terinformasikan secara cepat melalui *website*. Sesuatu yang biasanya memerlukan waktu yang lama untuk dilaksanakan atau diintegrasikan, dengan *e-commerce* menjadi lebih cepat. Melakukan model usaha yang inovatif atau melakukan *reengineering*, melaksanakan spesialisasi dengan derajat yang tinggi

atau meningkatkan produktivitas dan perhatian terhadap pelanggan, bukan sesuatu yang tidak mungkin dengan *e-commerce*.[1]

E-commerce juga bermanfaat dalam membangun *database* pelanggan yang *komprehensif*. Produsen dapat mempunyai informasi tentang pola pemesanan yang dilakukan pelanggan dan mengelolanya sebagai informasi yang berharga. *Database* tersebut akan membantu produsen saat melakukan pemasaran dan strategi promosi agar dapat tepat sasaran.[1]

Dalam konteks hubungan dengan mitra bisnis, *e-commerce* membantu dalam mengurangi inefisiensi yang mungkin terjadi dalam rantai penawaran, mengurangi kebutuhan untuk membuat *inventory* dan menghindari keterlambatan pengiriman. Sehingga produsen mempunyai kepercayaan diri tentang usaha yang dijalankan dalam melakukan kerjasama dengan pemasok dan perusahaan jasa.[1]

E-commerce akan menyederhanakan dan mengotomatisasi proses bisnis yang mendukung, menggabungkan dengan kecepatan dan efisiensi dalam kegiatan usaha. Dalam hubungannya dengan pelanggan, *e-commerce* membantu dalam menfasilitasi kegiatan pembelian yang nyaman. *E-commerce* dapat menghemat waktu pelanggan dibandingkan jika pelanggan tersebut melakukan pembelian secara *offline*. Seringkali pelanggan membayar lebih murah untuk harga produk tertentu dibandingkan jika pelanggan membelinya secara *offline*. Meskipun memiliki beberapa keuntungan, penggunaan *e-commerce* juga menghadapi kendala. Melakukan kegiatan transaksi secara *online* berarti pelanggan akan terpaksa menyediakan sejumlah informasi pribadi yang dipersyaratkan oleh penjual.[1]

1.4 Metode *Prototyping*

Model *prototyping* adalah metode pengembangan sistem di mana *prototype* dibangun, diuji dan kemudian dikerjakan ulang seperlunya sampai hasil yang dapat diterima dicapai dari mana sistem atau produk yang lengkap dapat dikembangkan. Dengan metode *prototyping* ini akan dihasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi.

Langkah-langkah model *prototyping* yaitu:[13]

1. Persyaratan sistem baru didefinisikan sedetail mungkin. Ini biasanya melibatkan wawancara dengan sejumlah pengguna yang mewakili semua departemen atau aspek dari sistem yang ada dengan cara berinteraksi secara langsung dengan calon pengguna untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian yang akan

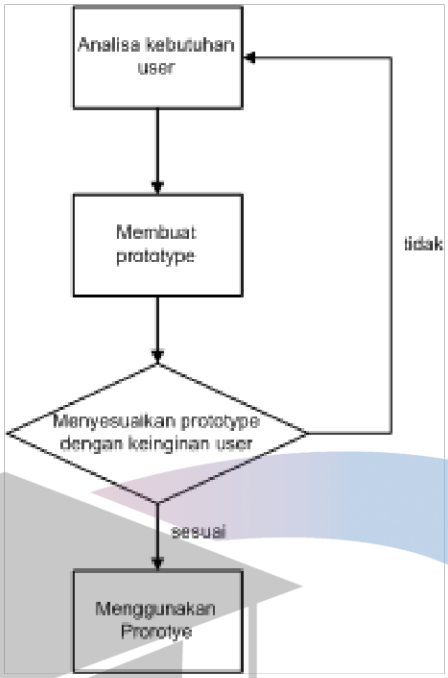
dibutuhkan.

2. Desain awal dan sederhana dibuat untuk sistem baru.
3. *Prototype* pertama dari sistem baru dibangun dari desain awal. Ini biasanya merupakan sistem yang diperkecil, dan mewakili perkiraan karakteristik produk akhir.
4. Pengguna secara menyeluruh mengevaluasi *prototype* pertama dan mencatat kekuatan dan kelemahannya, apa yang perlu ditambahkan dan apa yang harus dihapus. Pengembang mengumpulkan dan menganalisis komentar dari pengguna.
5. *Prototype* pertama dimodifikasi, berdasarkan komentar yang diberikan oleh pengguna, dan *prototype* kedua dari sistem baru dibangun.
6. *Prototype* kedua dievaluasi dengan cara yang sama seperti *prototype* pertama.
7. Langkah-langkah sebelumnya diulang sebanyak yang diperlukan, sampai pengguna yakin bahwa *prototype* tersebut mewakili produk akhir yang diinginkan.
8. Sistem akhir dibangun, berdasarkan *prototype* akhir.
9. Sistem akhir dievaluasi dan diuji secara menyeluruh. Pemeliharaan rutin dilakukan secara terus menerus untuk mencegah kegagalan skala besar dan meminimalkan waktu henti.

McLeod dan Schell menefinisikan dua tipe dari *prototype*, yaitu :

1. *Evolutionary Prototype*

Evolutionary Prototype yaitu, *prototype* yang secara terus menerus dikembangkan hingga *prototype* tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sitem. Berikut ini adalah gambaran dari tahapan *evolutionary prototype*.

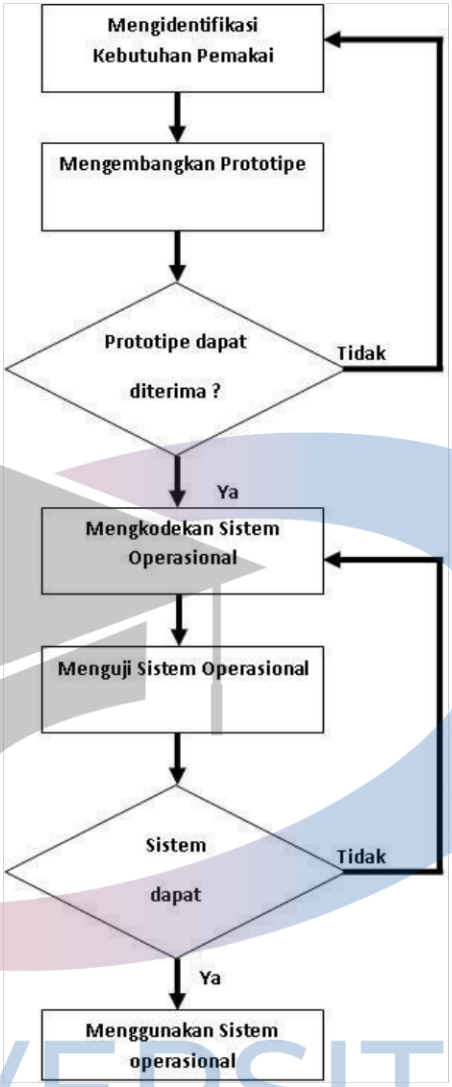


Gambar 1.2 Tahapan *Evolutionary Prototype*

a. Analisis kebutuhan *user*, pengembang dan pengguna atau pemilik sistem melakukan diskusi dimana pengguna atau pemilik sistem menjelaskan kepada pengembang tentang kebutuhan sistem yang mereka inginkan.
b. Membuat *prototype*, pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh pengguna atau pemilik sistem.
c. Menyesuaikan *prototype* dengan keinginan *user*, pengembang menanyankan kepada pengguna atau pemilik sistem tentang *prototype* yang sudah dibuat, apakah sesuai dengan kebutuhan sistem atau tidak.
d. Menggunakan *prototype*, sistem mulai dikembangkan dengan *prototype* yang sudah dibuat.

2. Requirement prototype

Requirement Prototype merupakan *prototype* yang dibuat oleh pengembang dengan mendefinisikan fungsi dan prosedur sistem dimana pengguna atau pemilik sistem tidak bisa mendefinisikan sistem tersebut. Berikut ini langkah-langkah dari *requirement prototype*.



Gambar 1.3 Tahapan *Requirement Prototype*

a. Analisis kebutuhan *user*, pengembang dan pengguna atau pemilik sistem melakukan diskusi dimana pengguna atau pemilik sistem menjelaskan kepada pengembang tentang kebutuhan sistem yang mereka inginkan.
b. Membuat *prototype*, pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh pengguna atau pemilik sistem.
c. Menyesuaikan *prototype* dengan keinginan *user*, pengembang menanyakan kepada pengguna atau pemilik sistem tentang *prototype* yang sudah dibuat, apakah sesuai dengan kebutuhan sistem atau tidak.
d. Membuat sistem baru, pengembang menggunakan *prototype* yang sudah dibuat untuk membuat sistem baru.
e. Melakukan *testing* sistem, pengguna atau pemilik sistem melakukan uji coba terhadap

sistem yang dikembangkan.

f. Menyesuaikan dengan keinginan *user*, sistem disesuaikan dengan keinginan *user* dan kebutuhan sistem, jika sudah sesuai sistem siap digunakan.
g. Menggunakan sistem.

Keuntungan dari model *prototyping*. [13]

1. Pelanggan mendapatkan suara dalam produk sejak dini, meningkatkan kepuasan pelanggan
2. Fungsionalitas yang hilang dan kesalahan terdeteksi dengan mudah.
3. *Prototype* dapat digunakan kembali di masa depan, proyek yang lebih rumit.
4. Ini menekankan komunikasi tim dan praktik desain yang fleksibel.
5. Pengguna memiliki pemahaman yang lebih baik tentang cara kerja produk.
6. Umpan balik pelanggan yang lebih cepat memberikan gambaran yang lebih baik tentang kebutuhan pelanggan.

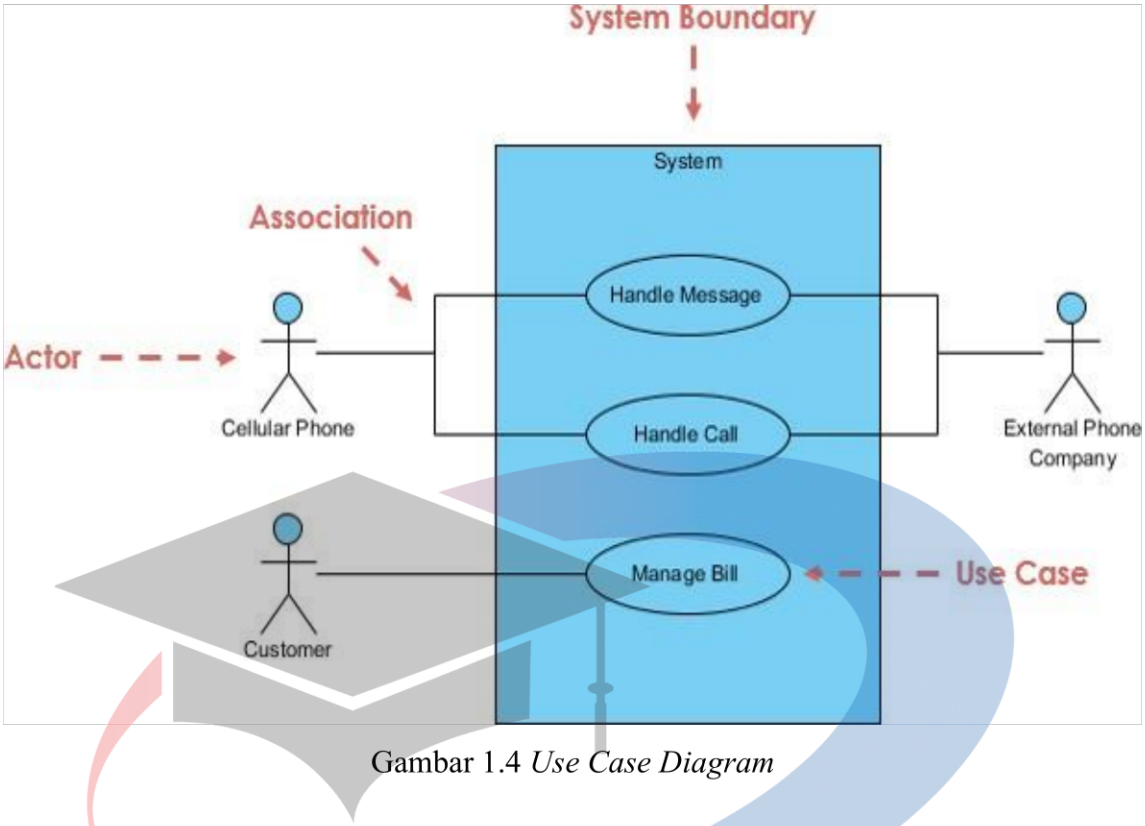
Kelemahan dari teknik pengembangan *prototyping* adalah mengundang umpan balik pelanggan sejak awal dalam siklus hidup pengembangan dapat menyebabkan masalah. Satu masalah adalah mungkin ada terlalu banyak permintaan perubahan yang mungkin sulit untuk dipenuhi. Masalah lain dapat muncul jika setelah melihat *prototype*, pelanggan menuntut rilis akhir yang lebih cepat atau menjadi tidak tertarik pada produk.[13]

1.5 Teknik Perancangan Kebutuhan

1.5.1 *Use Case Diagram*

Use Case Diagram adalah gambaran grafis dari beberapa atau semua *actor, use case* dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Di dalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat.[14]

Bentuk standar diagram *use case* didefinisikan dalam *Unified Modeling Language* seperti yang ditunjukkan pada contoh *Use Case Diagram* di bawah ini:



Gambar 1.4 *Use Case Diagram*

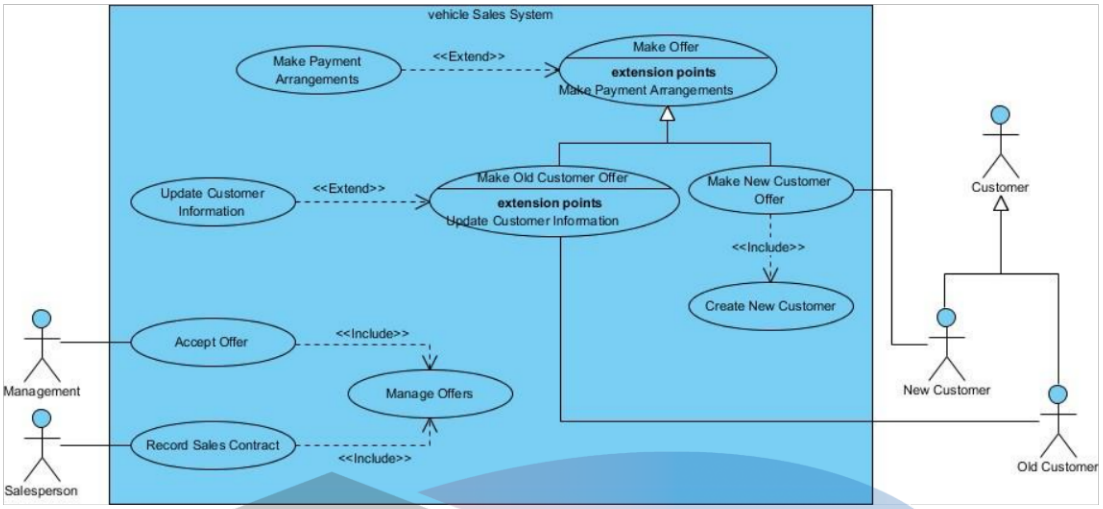
Elemen-elemen pada *Use Case Diagram*:[15]

a. *Actor*: mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Aktor hanya berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case.*

b. *Use Case*: gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

c. *Association*: menghubungkan link antar element

d. *Boundary*: batas sistem berpotensi menjadi keseluruhan sistem seperti yang ditentukan dalam dokumen persyaratan. Untuk sistem yang besar dan kompleks, setiap modul mungkin merupakan batas sistem. Seluruh sistem dapat menjangkau semua modul ini yang menggambarkan batas sistem secara keseluruhan.



Gambar 1.5 Relasi dalam *Use Case Diagram*

Ada beberapa relasi yang terdapat pada *Use Case Diagram*: [15]

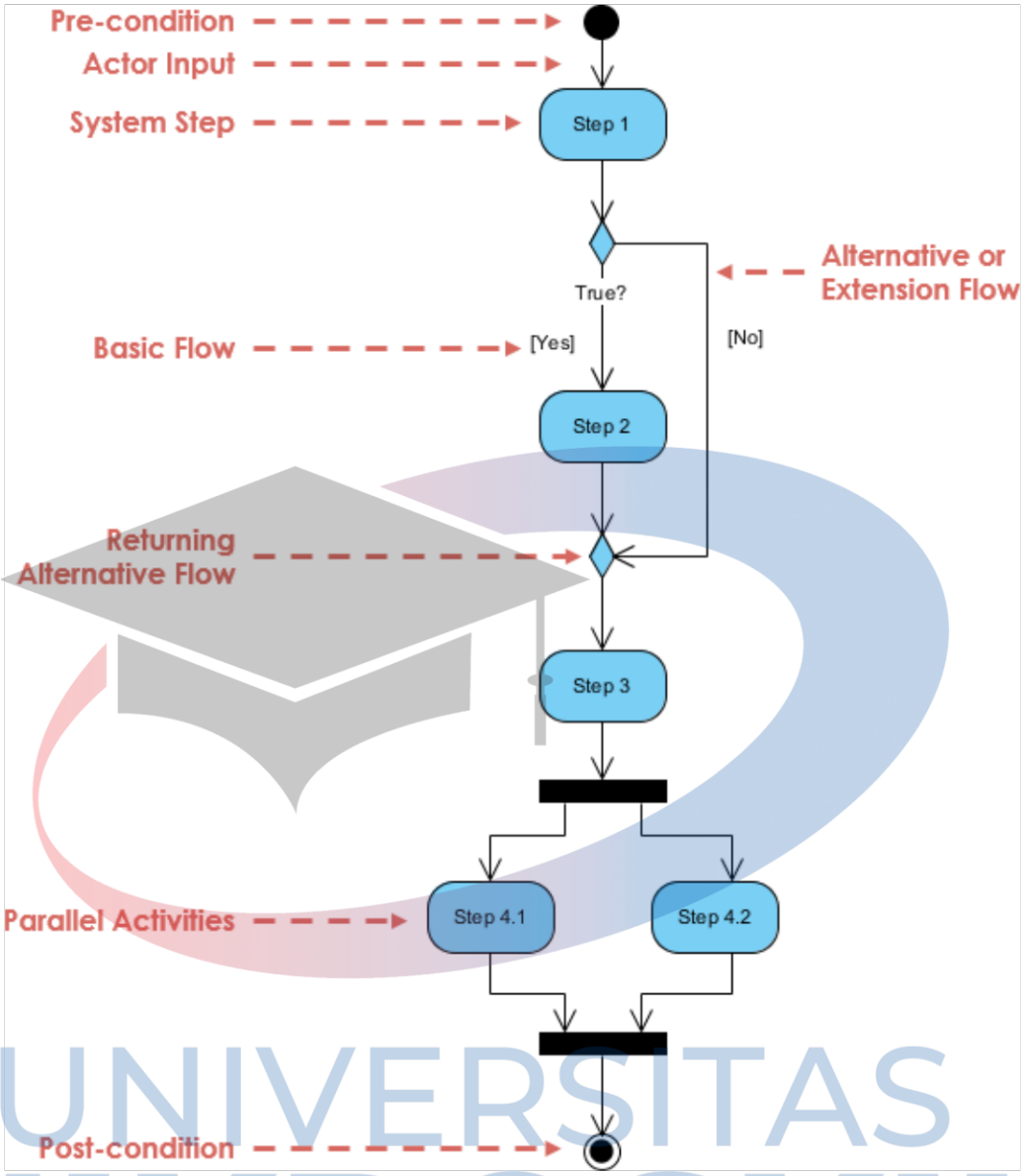
a. *Relasi extend* menunjukkan bahwa *use case "Invalid Password"* dapat mencakup perilaku yang ditentukan oleh *use case "Login Account"*. Relasi ini menunjukkan suatu hubungan yang diperluas.
b. *Include* yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
c. *Generalization* menggambarkan hubungan induk-anak antara *use case. Use case* anak merupakan tambahan dari *use case* induk.

1.5.2 *Activity Diagram*

Activity diagram adalah diagram alur yang menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah-langkah dalam proses kerja sistem yang dibuat. Struktur diagram ini juga mirip dengan *flowchart*.[16]

Fungsi *activity diagram* yaitu:

1. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.
2. Memperlihatkan urutan aktifitas proses pada sistem.
3. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.



Gambar 1.6 *Activity Diagram*

Komponen yang ada pada *activity diagram* yaitu:[16]

1. *Start Point/Initial State* (Titik Mulai)

   Start Point merupakan lingkaran hitam kecil, yang menandakan tindakan awal atau titik awal aktivitas untuk setiap diagram aktivitas.

2. *Activitiy* (Aktivitas)

   Activity menunjukkan aktivitas yang dilakukan atau yang sedang terjadi.

3. Action Flow (Arah)

   Action Flow digunakan untuk transisi dari suatu tindakan ke tindakan yang lain atau menunjukan aktivitas selanjutnya setelah aktivitas sebelumnya.

4. *Decision* (Keputusan)

   Decision adalah suatu titik atau point pada *activity diagram* yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.

5. *Synchornization*

   Synchornization dibagi menjadi 2 yaitu *fork* dan *join*.

a. *Fork* (percabangan) digunakan untuk memecah behaviour menjadi activity atau *action* yang paralel.

b. *Join* (penggabungan) untuk menggabungkan kembali *activity* atau *action* yang paralel.

6. *Merge Event* (Menggabungkan)

   Merge Event berfungsi untuk menggabungkan *flow* yang dipecah oleh *decission*.

7. *Swimlanes*

   Swimlanes berfungsi untuk memecah *activity diagram* menjadi baris dan kolom untuk membagi tangung jawab objek-objek yang melakukan aktivitas.

8. *Final State/End Point* (Titik Akhir)

   Final State menunjukan bagian akhir dari aktivitas.

1.5.3 *Sequence Diagram*

Sequence Diagram adalah diagram interaksi yang merinci bagaimana sebuah operasi dilakukan. *Sequence diagram* digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi selama periode waktu tertentu.[17]

Manfaat *sequence diagram*:[18]

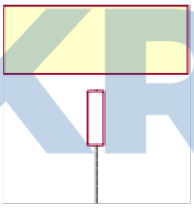
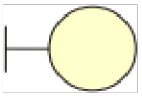
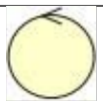
1. *Sequence diagram* adalah jenis diagram UML yang banyak dimanfaatkan sebagai referensi bisnis dan kegiatan lainya.
2. Mewakili detail kasus pengguna UML.
3. Memodelkan logika dan fungsi, operasi atau prosedur yang canggih.
4. Memperlihatkan bagaimana objek dan komponen saling berinteraksi satu sama lain untuk menyelesaikan suatu proses.
5. Merencanakan dan memahami fungsionalitas terperinci dari skenario yang ada saat ini atau di masa depan.

Tujuan *sequence diagram*:[18]

1. Model interaksi tingkat tinggi antara objek aktif dalam sistem.
2. Membuat model interaksi antara contoh objek dalam kolaborasi yang mewujudkan *use case.*
3. Membuat model interaksi antara objek dalam kolaborasi yang mewujudkan sebuah operasi.
4. Salah satu model interaksi umum (menampilkan semua kemungkinan jalur melalui interaksi) atau contoh spesifik dari interaksi (hanya menampilkan satu jalur melalui interaksi).

Berikut adalah komponen-komponen yang terdapat di dalam *sequence diagram:*[18]

Tabel 1.1 Komponen *Sequence Diagram*

Nama	Simbol	Deskripsi
Aktor		Aktor merepresentasikan entitas yang berada di luar sistem dan berinteraksi degan sistem, bisa berupa perangkat keras ataupun sistem yang lain.
Lifeline		Fungsi dari simbol ini adalah mengeksekusi objek selama *sequence* (pesan dikirim atau diterima dan aktifasinya).
General		Fungsinya adalah merepresentasikan entitas tunggal dalam *sequence diagram*. Entitas memiliki nama, *stereotype* atau berupa *instance (class).*
Boundary		*Boundary* biasanya berupa tepi dari sistem, seperti user interface atau suatu alat yang berinteraksi dengan sistem yang lain.
Control		*Control* elemen mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya megatur perilaku dan perilaku bisnis.

Entity		Entitas biasanya elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa *beans* atau model *object*.
Activation		Suatu titik dimana sebuah objek mulai berpartisipasi di dalam sebuah *sequence* yang menunjukan kapan sebuah objek mengirim atau menerima objek.
Message		Berfungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan. *Message* terjadi antara dua buah objek dimana satu objek (*client*) dan meminta objek (*supplier*) untuk melakukan sesuatu.
Message Entry		Simbol ini berfungsi untuk menggambarkan pesan/hubungan antar objek yang menunjukan urutan kejadian yang terjadi.
Message to Self		Simbol ini menggambarkan pesan/hubungan objek itu sendiri, yang menunjukan urutan kejadian yang terjadi.
Message Return		Simbol ini menggambarkan hasil dari pengiriman *message* dan digambarkan dengan arah dari kanan ke kiri.

1.5.4 *Class Diagram*

Class diagram adalah suatu jenis diagram yang paling berguna di UML, hal ini karena dapat dengan jelas memetakan struktur sistem tertentu dengan memodelkan kelas, atribut, operasi serta hubungan antar objek. *Class diagram* adalah model logis, yang berkembang menjadi model fisik dan akhirnya menjadi sistem informasi yang berfungsi. Analisis struktur, enritik, data penyimpanan dan proses diubah menjadi struktur data dan kode program. . *Class diagram* berkembang menjadi modul kode, objek baca, dan komponen sistem lainnya. Dalam diagram kelas, setiap kelas muncul sebagai susunan ulang, dengan nama kelas di atas, diikuti dengan atribut dan metode kelas. Garis dan bagaimana hubungan antar kelas memiliki label yang mengidentifikasi tindakan yang menghubungkan kedua kelas. Untuk membuat *class diagram*, tinjau kasus penggunaan dan identifikasi kelas yang berpartisipasi dalam proses

bisnis yang mendasari. Diagram kelas juga mencakup konsep yang disebut *cardinality*, yang menjelaskan bagaimana *instance* dari satu kelas berhubungan dengan *instarnce* kelas lain.[17]

Tujuan *class diagram:*[19]

1. Menunjukkan struktur statis pengklasifikasi dalam suatu sistem.
2. Memberikan notasi dasar untuk diagram struktur lain yang ditentukan oleh UML.
3. Bermanfaat untuk pengembang dan anggota tim lainnya.
4. Analis bisnis dapat menggunakan diagram kelas untuk memodelkan sistem dari perspektif bisnis.

Komponen *class diagram*: [19]

1. *Class name*, merupakan sesuatu yang mewakili dari nama kelas.
2. Atribut, merupakan properti dari sebuah kelas, atribut melambangkan batas nilai kelas yang mungkin terdapat dalam objek kelas.
3. Proses atau *method*, sesuatu yang dapat dilakukan tau diproses oleh sebuh kelas.