

BAB II

LANDASAN TEORI

2.1 Sistem

Menurut Sukoco (2007), sistem terdiri dari subsistem yang berhubungan dengan prosedur yang membantu pencapaian tujuan. Pada saat prosedur diperlukan untuk melengkapi proses pekerjaan, maka metode berisi tentang aktivitas operasional atau teknis yang menjelaskannya.

2.2 Sistem Informasi

Sistem Informasi adalah kumpulan elemen yang saling berhubungan satu sama lain untuk membentuk suatu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi tersebut (Oetomo, 2006). Sistem informasi (SI) adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data dan teknologi. Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil,

mengubah, mengolah, dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan lainnya.

Kata “sistem” mengandung arti kumpulan dari komponen-komponen yang memiliki keterkaitan antara yang satu dengan lainnya. Dari definisi sistem, maka dapat didefinisikan bahwa “Sistem Informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yang menyajikan informasi”. Sering orang salah mengartikan antara sistem informasi dengan teknologi informasi. Dengan mengesampingkan teknologi informasi beserta produk-produknya, sistem informasi yang dihasilkan tentunya tidak lebih baik jika dibandingkan dengan sistem informasi yang menggunakan teknologi informasi untuk mendukung penyajian informasinya.

Sistem informasi juga berfungsi sebagai alat bantu kompetisi bagi organisasi dalam mengupayakan pencapaian tujuan. Sistem Informasi dituntut tidak hanya mengolah data dari dalam organisasi saja, tetapi juga dapat menyajikan data dari pihak luar yang mampu menambah nilai kompetisi bagi dalam organisasi. Dengan demikian sistem informasi harus memiliki data yang telah terpolakan dan memiliki integritas dalam hal waktu dan tempat. Hal ini dimaksudkan supaya sistem informasi tersebut dapat menyajikan informasi yang tepat bagi pengguna.

2.3 Penjualan

Secara umum definisi penjualan dapat diartikan sebagai sebuah usaha atau langkah konkrit yang dilakukan untuk memindahkan suatu produk, baik itu berupa barang ataupun jasa, dari produsen kepada konsumen sebagai sasarannya.

Tujuan utama penjualan yaitu mendatangkan keuntungan atau laba dari produk ataupun barang yang dihasilkan produsennya dengan pengelolaan yang baik. Dalam pelaksanaannya, penjualan sendiri tak akan dapat dilakukan tanpa adanya pelaku yang bekerja didalamnya seperti agen, pedagang dan tenaga pemasaran.

Penjualan adalah suatu usaha yang terpadu untuk mengembangkan rencana-rencana strategis yang diarahkan pada usaha pemuasan kebutuhan dan keinginan pembeli guna mendapatkan penjualan yang menghasilkan laba (Marwan A, 1991). Penjualan merupakan sumber hidup suatu perusahaan, karena dari perusahaan dapat diperoleh laba serta suatu usaha memikat konsumen yang diusahakan untuk mengetahui daya tarik mereka sehingga dapat mengetahui hasil produk yang dihasilkan.

Pada waktu menjual, terkadang perusahaan harus menerima pengembalian barang atau memberi potongan harga. Hal ini terjadi kalau barang yang dijual tidak sesuai dengan permintaan pembeli. Penerimaan kembali barang yang telah dijual disebut penjualan retur (sales return), sedangkan pemberian potongan harga disebut pengurangan harga (sales allowances). Pada umumnya, penjualan retur dan pengurangan harga dicatat dalam satu perkiraan (Mulyadi, 2008).

2.4 Ritel

Ritel berasal dari Bahasa Prancis *RITELLIER* yang berarti memecah sesuatu. Secara harfiah kata ritel atau retail juga berarti eceran atau perdagangan eceran, dan para peritel/retailer diartikan sebagai pengecer atau pengusaha perdagangan eceran.

Retail adalah sekelompok kegiatan yang menjual atau menambahkan nilai barang dan jasa pada konsumen akhir untuk digunakan secara pribadi, keluarga, atau rumah tangga. Dengan demikian, peran *Retailing* di sini adalah sebagai saluran bisnis terakhir distribusi dari mata rantai pabrik kepada konsumen akhir (Utami, 2008).

Dari definisi tersebut terlihat bahwa pada hakikatnya aktivitas bisnis retail tidak sekadar penjualan barang dalam arti fisik, namun juga meliputi penjualan jasa. Dalam pengertian ini, penjualan tiket pesawat, jasa telekomunikasi (wartel) warnet, dan sebagainya juga tercakup dalam kategori bisnis retail. Penjualan tersebut merupakan penjualan *real services*.

Berkaitan dengan tempat dilakukannya aktivitas penjualan, pengertian bisnis retail tidak hanya dilakukan pada sebuah retail (*shop/store*) tetapi juga mencakup aktivitas serupa yang tidak menggunakan tempat khusus dalam proses jual-beli, misalnya *mail order* (layanan pesan barang melalui surat/telepon) dan *direct selling* (penjualan dari rumah ke rumah atau berdasarkan keanggotaan multilevel marketing). Berdasarkan pengertian bisnis retail tersebut, *mail order* dan *direct selling* juga merupakan bentuk lain dari entitas bisnis retail.

Selanjutnya, penjual partai besar (grosir atau *wholesaler* dan bahkan pabrikan atau *manufacture*) dapat pula berlaku sebagai retail jika mereka juga melakukan penjualan barang dan jasanya kepada konsumen akhir secara langsung. Contohnya adalah konsep retailing Makro dan Goro yang memosisikan diri sebagai wholesaler, namun juga melayani konsumen akhir. Kemudian, konsep *factory outlet* merupakan contoh nyata bagi industry yang melakukan penjualan

langsung kepada konsumen akhir (Bata Factory Outlet, Metro Factory Outlet, dan sebagainya).

Karakteristik dasar ritel dapat digunakan sebagai dasar mengelompokkan jenis ritel. Dalam hal ini, terdapat tiga karakteristik dasar sebagai berikut :

1. Pengelompokan berdasarkan unsur-unsur yang digunakan ritel untuk memuaskan kebutuhan konsumen.
2. Harga barang dagangan.
3. Pengelompokan berdasarkan sarana atau media yang digunakan oleh ritel.

Fungsi-fungsi yang dijalankan ritel :

- a. Menyediakan berbagai macam produk dan jasa. Peritel selalu berusaha menyediakan beraneka ragam produk dan jasa baik dari sisi variasi jenis, merek dan ukuran.
- b. Memecah (breaking bulk). Berarti memecah beberapa ukuran produk menjadi lebih kecil, yang pada akhirnya akan menguntungkan produsen dan konsumen.
- c. Menyimpan persediaan. Ritel akan berperan sebagai penyimpan stok (holding inventory) dengan ukuran lebih kecil. Dalam hal ini pelanggan akan diuntungkan karena terdapat jaminan ketersediaan barang pada saat mereka membutuhkannya.
- d. Meningkatkan nilai produk dan jasa.

Bisnis ritel pada dasarnya dapat dibedakan menjadi dua kelompok besar, yakni Ritel Tradisional dan Ritel Modern. Ritel modern sendiri merupakan pengembangan dari ritel tradisional. Seiring dengan meningkatnya kemakmuran

ekonomi yang dipicu oleh revolusi industri terjadilah pergeseran tren perilaku konsumen antara lain :

- a. Arus urbanisasi yang pesat (perpindahan penduduk dari desa ke kota) untuk mencari lapangan pekerjaan.
- b. Semakin meningkatnya pendapatan/kemakmuran memberikan banyak pilihan bagi konsumen untuk membelanjakan uangnya.
- c. Meningkatnya tuntutan terhadap kemudahan dan kenyamanan (pelayanan yang lebih baik) dalam berbelanja.
- d. Meningkatnya orientasi terhadap nilai (value) dalam berbelanja

2.5 Usaha Mikro Kecil dan Menengah (UMKM)

Sesuai dengan Undang- Undang Nomor 20 Tahun 2008 tentang Usaha Mikro, Kecil dan Menengah (UMKM) :

1. Pengertian UMKM

- a. **Usaha Mikro** adalah usaha produktif milik orang perorangan dan/atau badan usaha perorangan yang memenuhi kriteria Usaha Mikro.
- b. **Usaha Kecil** adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau bukan cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dari usaha menengah atau usaha besar yang memenuhi kriteria Usaha Kecil.
- c. **Usaha Menengah** adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perseorangan atau badan usaha

yang bukan merupakan anak perusahaan atau cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dengan Usaha Kecil atau usaha besar dengan jumlah kekayaan bersih atau hasil penjualan tahunan sesuai dengan kriteria Usaha Menengah.

2. Kriteria UMKM

a) Kriteria **Usaha Mikro** adalah sebagai berikut:

- Memiliki kekayaan bersih paling banyak Rp50.000.000,00 (lima puluh juta rupiah) tidak termasuk tanah dan bangunan tempat usaha; atau
- Memiliki hasil penjualan tahunan paling banyak Rp300.000.000,00 (tiga ratus juta rupiah).

b) Kriteria **Usaha Kecil** adalah sebagai berikut:

- Memiliki kekayaan bersih lebih dari Rp50.000.000,00 (lima puluh juta rupiah) sampai dengan paling banyak Rp500.000.000,00 (lima ratus juta rupiah) tidak termasuk tanah dan bangunan tempat usaha; atau
- Memiliki hasil penjualan tahunan lebih dari Rp300.000.000,00 (tiga ratus juta rupiah) sampai dengan paling banyak Rp2.500.000.000,00 (dua milyar lima ratus juta rupiah).

c) Kriteria **Usaha Menengah** adalah sebagai berikut:

- Memiliki kekayaan bersih lebih dari Rp500.000.000,00 (lima ratus juta rupiah) sampai dengan paling banyak Rp10.000.000.000,00 (sepuluh milyar rupiah) tidak termasuk tanah dan bangunan tempat usaha; atau
- Memiliki hasil penjualan tahunan lebih dari Rp2.500.000.000,00 (dua milyar lima ratus juta rupiah) sampai dengan paling banyak Rp50.000.000.000,00 (lima puluh milyar rupiah).

3. Ciri-ciri UMKM

a) Ciri-ciri **Usaha Mikro**

- Jenis barang/komoditi usahanya tidak selalu tetap, sewaktu-waktu dapat berganti.
- Tempat usahanya tidak selalu menetap, sewaktu-waktu dapat pindah tempat.
- Belum melakukan administrasi keuangan yang sederhana sekalipun, dan tidak memisahkan keuangan keluarga dengan keuangan usaha.
- Sumber daya manusianya (pengusahanya) belum memiliki jiwa wirausaha yang memadai.
- Tingkat pendidikan rata-rata relatif sangat rendah.
- Umumnya belum akses kepada perbankan, namun sebagian dari mereka sudah akses ke lembaga keuangan non bank.

- Umumnya tidak memiliki izin usaha atau persyaratan legalitas lainnya termasuk NPWP.

b) Ciri-ciri **Usaha Kecil**

- Jenis barang/komoditi yang diusahakan umumnya sudah tetap tidak gampang berubah.
- Lokasi/tempat usaha umumnya sudah menetap tidak berpindah-pindah.
- Pada umumnya sudah melakukan administrasi keuangan walau masih sederhana, keuangan perusahaan sudah mulai dipisahkan dengan keuangan keluarga, sudah membuat neraca usaha.
- Sudah memiliki izin usaha dan persyaratan legalitas lainnya termasuk NPWP.
- Sumberdaya manusia (pengusaha) memiliki pengalaman dalam berwira usaha.
- Sebagian sudah akses ke perbankan dalam hal keperluan modal.
- Sebagian besar belum dapat membuat manajemen usaha dengan baik seperti *business planning*.

c) Ciri-ciri **Usaha Menengah**

- Pada umumnya telah memiliki manajemen dan organisasi yang lebih baik, lebih teratur bahkan lebih modern, dengan pembagian tugas yang jelas antara lain, bagian keuangan, bagian pemasaran dan bagian produksi.
- Telah melakukan manajemen keuangan dengan menerapkan sistem akuntansi dengan teratur, sehingga memudahkan untuk

auditing dan penilaian atau pemeriksaan termasuk oleh perbankan.

- Telah melakukan aturan atau pengelolaan dan organisasi perburuhan, telah ada Jamsostek, pemeliharaan kesehatan dll.
- Sudah memiliki segala persyaratan legalitas antara lain izin tetangga, izin usaha, izin tempat, NPWP, upaya pengelolaan lingkungan dll.
- Sudah akses kepada sumber-sumber pendanaan perbankan.
- Pada umumnya telah memiliki sumber daya manusia yang terlatih dan terdidik.

2.6 Katalog Produk

Menurut Kamus Besar Bahasa Indonesia, katalog adalah carik kartu, daftar /buku yang memuat nama benda/informasi tertentu yang ingin disampaikan, disusun secara berurutan, teratur dan alfabetis.

Definisi katalog dapat berbeda-beda sesuai dengan bidangnya. Untuk bidang bisnis (promosi), katalog berarti daftar barang-barang yang akan dijual lengkap dengan harga dan gambaran produk, biasanya dibuat dalam bentuk buku atau kadang-kadang dalam bentuk media lain seperti CD-ROM, video atau website. Dalam bidang seni, katalog dapat diartikan sebagai buku panduan dalam pameran yang berisi penjelasan mengenai benda (objek) yang dipamerkan. Dalam bidang perpustakaan, katalog berarti daftar buku-buku dan majalah yang ada diperpustakaan. Biasanya disusun berdasarkan mata pelajaran atau mata kuliah, judul atau pengarang. Dalam bidang pendidikan, katalog diartikan sebagai daftar dari semua mata kuliah yang ditawarkan oleh perguruan tinggi atau universitas,

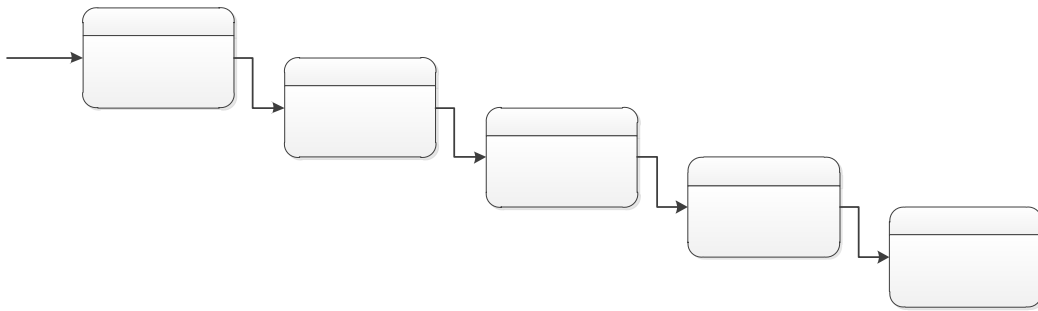
lengkap dengan informasi tentang fakultas yang ada dan informasi tentang kampus. Definisi katalog produk adalah kumpulan informasi tentang produk-produk yang dijual oleh perusahaan tertentu yang dikemas ke dalam suatu media penyampaian. Suatu katalog produk memungkinkan konsumen untuk memperoleh informasi tentang produk-produk yang dijual di dalam satu format terstruktur dengan definisi produk yang konsisten.

2.7 Referensi Pawoon.com

Pawoon.com merupakan sebuah aplikasi kasir berbasis cloud yang dapat digunakan untuk memantau penjualan dalam suatu perusahaan hanya dengan memanfaatkan koneksi internet. Pawoon.com dapat digunakan untuk melakukan pencatatan penjualan, mengelola stok barang, dan lain sebagainya.

2.8 Siklus Hidup Pengembangan Sistem

Menurut Pressman (2015), *System Development Life Cycle (SDLC)* atau Siklus Hidup Pengembangan Sistem ini biasanya disebut juga dengan model waterfall. Menurut Pressman (2015), nama lain dari Model *Waterfall* adalah Model Air Terjun kadang dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak. Pengembangan perangkat lunak dimulai dari spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan.



Gambar 2.1 Pengembangan menggunakan Model *Waterfall*

Gambar 2.1 menunjukkan tahapan umum dari model proses *waterfall*. Model ini disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Akan tetapi, Pressman (2015) memecah model ini meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *Communication, Planning, Modeling, Construction, dan Deployment*.

Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman (2015):

a. Communication

Langkah pertama diawali dengan komunikasi kepada konsumen/pengguna. Langkah awal ini merupakan langkah penting karena menyangkut pengumpulan informasi tentang kebutuhan konsumen/pengguna.

b. Planning

Setelah proses *communication* ini, kemudian menetapkan rencana untuk pengerjaan software yang meliputi tugas-tugas teknis yang akan dilakukan, risiko yang mungkin terjadi, sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal pengerjaan.

c. *Modeling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural.

d. *Construction*

Construction merupakan proses membuat kode (*code generation*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

e. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan user. Kemudian software yang telah dibuat harus dilakukan pemeliharaan secara berkala.

2.9 Diagram *Unified Modeling Language (UML)*

2.9.1 Diagram *Use Case* Bisnis

Diagram *use case* bisnis digunakan untuk mempresentasikan bisnis yang dilakukan organisasi. Diagram ini menjawab pertanyaan: “apa yang bisnis lakukan?” dan “mengapa kita membangun sistem untuk itu?”. Diagram *use case* bisnis digunakan untuk memodelkan aktivitas bisnis organisasi sebagai landasan pembuatan *use case* sistem. Diagram *use case* bisnis juga digunakan untuk mendefinisikan bisnis apa saja yang dilakukan organisasi dalam rangka menjalankan visi organisasi (Sholiq, 2010).

Diagram *use case* bisnis digambarkan menurut perspektif organisasi. Ia tidak membedakan apakah aktivitas tersebut dilakukan secara manual atau otomatis menggunakan perangkat lunak.

Komponen pembentuk diagram *use case* bisnis :

1. Aktor Bisnis

Aktor bisnis adalah seseorang atau sesuatu yang ada di luar organisasi. Ia berinteraksi dengan organisasi dan terlibat dalam kegiatan bisnis organisasi. Contoh aktor bisnis, antara lain : pelanggan, kreditor, investor atau pemasok. Jadi posisi mereka di luar organisasi yang sedang dimodelkan, tetapi terlibat dalam kegiatan organisasi. Aktor bisnis dimodelkan dengan menggunakan ikon berikut :



Gambar 2.2 Notasi Aktor Bisnis

2. Pekerja Bisnis

Pekerja bisnis adalah suatu peran (*role*) di dalam organisasi, bukan posisi atau jabatan. Seseorang bias memainkan banyak peran tetapi memegang hanya satu posisi.



Gambar 2.3 Notasi Pekerja Bisnis

Memodelkan pekerja bisnis digunakan untuk memahami peran di dalam aktivitas bisnis organisasi dan bagaimana peran tersebut berinteraksi dengan proses bisnis organisasi. Sangat jelas perbedaan antara aktor bisnis dan pekerja bisnis, jika aktor bisnis berada di luar ruang lingkup bisnis yang sedang didefinisikan, sedangkan pekerja bisnis berada di dalam bisnis yang didefinisikan.

3. Use Case Bisnis

Sebuah *use case* bisnis adalah model yang digunakan untuk menggambarkan sebuah proses bisnis organisasi. Dengan kata lain, *use case* bisnis menginformasikan tentang aktivitas bisnis utama yang organisasi lakukan.



Gambar 2.4 Notasi Use Case

Use case bisnis memiliki format penulisan nama kata kerja atau frase kata kerja, seperti pada kata "Memberi harga produk". Tentu saja, *Use Case*

Bisnis ‘Memberi harga produk’ tidak memberikan banyak informasi tanpa adanya bberaoa penjelasan tambahan. Setiap *use case* bisnis, dapat dibuat penjelasan tambahan untuk menjelaskan secara rinci apa yang terjadi di dalam *use case* bisnis. Untuk hal tersebut didokumentasikan secara spesifik di dalam sebuah *workflow*.

4. Relasi

Untuk membuat diagram *use case* bisnis digunakan penghubung (relasi) antara aktor bisnis dan atau pekerja bisnis dengan *use case* bisnis. Ada dua jenis relasi yang mungkin terjadi, pertama adalah relasi antara aktor bisnis atau pekerja bisnis dengan use case bisnis, relasi jenis ini disebut asosiasi. Relasi asosiasi adalah relasi regular yang sering atau hampir terjadi pada kegiatan pemodelan bisnis. Kedua, relasi pewarisan struktur antara elemen-elemen pemodelan bisnis sendiri yang disebut generalisasi.

5. Entitas Bisnis

Entitas bisnis adalah objek digunakan atau yang dihasilkan oleh organisasi saat melakukan aktivitas bisnis. Entitas bisnis meliputi sesuatu yang pekerja bisnis hadapi sehari-hari. Setiap entitas bisnis harus diberi nama yang unik yang menggambarkan tanggung jawabnya. Nama berbentuk kata benda atau frase kata benda.



Gambar 2.5 Notasi Entitas Bisnis

2.9.2 Diagram Use Case

Diagram use case menyajikan interaksi antara use case dan actor dalam sistem yang akan dikembangkan. Use case sendiri adalah fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut menurut pandangan pemakai sistem. Sedangkan actor bias berupa orang, peralatan, atau sistem lain yang berinteraksi terhadap sistem yang akan dibangun (Sholiq, 2010).

2.9.3 Diagram Aktivitas

Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Ada dua kegunaan diagram aktivitas dalam pemodelan dengan UML, yaitu :

1. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan alur kerja bisnis (*business workflow*).
2. Pada tahap pemodelan sistem, diagram aktivitas dapat digunakan untuk menjelaskan aktivitas yang terjadi didalam sebuah *use case*.

Diagram aktivitas mendefinisikan darimana *workflow* dimulai, dimana *workflow* berakhir, aktivitas apa saja yang terjadi di dalam *workflow*, dan apa saja yang dilakukan saat aktivitas terjadi. Aktivitas adalah tugas yang dilakukan selama dalam *workflow*.

2.9.4 Diagram Sekuensial

Diagram sekuensial digunakan untuk menunjukkan alur (*flow*) fungsionalitas yang melalui sebuah use case yang disusun dalam urutan waktu.

2.9.5 Diagram Kelas

Diagram kelas menunjukkan interaksi antar kelas-kelas dalam sistem. Kelas juga dapat dianggap sebagai cetak biru dari objek-objek di dalam sistem.

2.10 Website

Menurut Shelly dan Vermaat (2010), web adalah koleksi dokumen elektronik milik semua orang di dunia yang mengaksesnya melalui internet menggunakan web *browser*. Menurut Simamarta (2010), aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web. Fitur-fitur aplikasi web biasanya berupa data *persistence*, mendukung transaksi dan komposisi halaman web dinamis yang dapat dipertimbangkan sebagai hibridasi, antara hypermedia dan sistem informasi. Aplikasi web adalah bagian dari *client-side* yang dapat dijalankan oleh *browser* web. *Client-side* mempunyai tanggung jawab untuk pengeksekusian proses bisnis.

Interaksi web menurut Simamarta (2010), dibagi dalam tiga langkah utama, yaitu:

1. Permintaan

Pengguna mengirimkan permintaan ke server web, biasanya via halaman web yang ditampilkan pada browser web.

2. Pemrosesan

Server web menerima permintaan yang dikirimkan oleh pengguna, kemudian memproses permintaan tersebut.

3. Jawaban

Browser menampilkan hasil dari permintaan pada jendela browser.

2.11 MySQL

MySQL adalah sebuah perangkat lunak *database* (basis data) sistem terbuka yang sangat terkenal di kalangan pengembang sistem *database* dunia yang digunakan untuk berbagai aplikasi terutama untuk aplikasi berbasis web. MySQL mempunyai fungsi sebagai SQL (*Structured Query Language*) telah diperluas. MySQL umumnya digunakan bersama dengan PHP untuk membuat aplikasi yang dinamis dan *powerful*.

2.12 PHP (*Hypertext Preprocessor*)

Menurut Saputra (2011, p.1) PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka layout web, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, web akan sangat mudah di-maintenance. PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap menjalankan PHP, wajib adanya web server.

2.13 *Framework CodeIgniter*

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.

Ada beberapa alasan mengapa menggunakan *Framework*:

- Mempercepat dan mempermudah pembangunan sebuah aplikasi web.

- Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada)
- Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, *pagination*, *multiple database*, *scaffolding*, pengaturan *session*, *error handling*, dll)
- Lebih bebas dalam pengembangan jika dibandingkan CMS

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *Small Talk*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu :

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (*insert*, *update*, *delete*, *search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu programmer yang menangani bagian model dan *controller*, sedangkan designer yang menangani bagian *view*, sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan.

2.14 Testing

Menurut Romeo (2003), testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan. *Testing Software* adalah proses pengoperasian *software* dalam suatu kondisi yang dikendalikan untuk verifikasi, mendeteksi *error* dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem apakah proses yang telah dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna. Deteksi *error* adalah testing yang berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut tidak terjadi. *Test case* merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun

hasil yang telah ditentukan sebelumnya. Adapun kegunaan dari *test case* ini, adalah sebagai berikut:

1. Untuk melakukan testing kesesuaian suatu komponen terhadap desain *White Box Testing*.
2. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi *Black Box Testing*.

2.14.1 White Box Testing

Menurut Romeo (2003), *white box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural. Seringkali *white box testing* diasosiasikan dengan pengukuran cakupan tes, yang mengukur persentase jalur-jalur dari tipe yang dipilih untuk dieksekusi oleh *test cases*. *White box testing* dapat menjamin semua struktur *internal* data dapat dites untuk memastikan validitasnya.

Cakupan pernyataan, cabang dan jalur adalah suatu teknik *white box testing* yang menggunakan alur logika dari program untuk membuat *test cases* alur logika adalah cara dimana suatu bagian dari program tertentu dieksekusi saat menjalankan program. Alur logika suatu program dapat direpresentasikan dengan *flow graph*.

2.14.2 Black Box Testing

Menurut Romeo (2003), *Black box testing* dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal dari sistem atau komponen yang dites, juga disebut sebagai *functional testing*. *Black box testing* bergfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.

Dengan adanya *black box testing*, perancang *software* dapat menggunakan kebutuhan fungsional pada suatu program. *Black box testing* dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari *error* dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan.

