

BAB II

LANDASAN TEORI

II.1. Sistem

II.1.1 Pengertian Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Contoh : Sistem Komputer terdiri dari *Software*, *Hardware*, dan *Brainware*.

- Menurut **Jerry Fith Gerald**, Sistem adalah suatu unsur jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.
- Menurut **Ludwig Von Bartalanfy**, Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi di antara unsur-unsur tersebut dengan lingkungan.
- Menurut **Anatol Rapoport**, Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.
- Menurut **L. Ackof**, Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung atau sama lainnya. (Asbon Hendra : 2012 ; 157)

II.1.2 Karakteristik Sistem

1. Komponen atau Elemen (*Components*)

Suatu sistem yang terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem, tidak peduli betapa pun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan suatu daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan yang lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu diluar batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan yang lainnya untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu dengan subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

5. Masukan (*Input*)

Masukan merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*Output*)

Keluaran merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna, contohnya informasi yang dikeluarkan oleh komputer. Dan *output* yang tidak berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

7. Pengolah (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contohnya CPU pada komputer, bagian produksi yang mengubah bahan baku menjadi barang jadi, serta bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan atau sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. (Asbon Hendra : 2012 ; 158-160)

II.1.3 Klasifikasi Sistem

1. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sebagai contoh, sistem Teologia yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan manusia.

2. Sistem Fisik (*Physical System*)

Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya. Contohnya, sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain.

3. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat oleh manusia, seperti sistem tata surya, sistem galaksi, sistem reproduksi, dan lain-lain.

4. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancamg oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system*, contohnya sistem informasi.

5. Sistem Tertentu (*Deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Contohnya sistem komputer.

6. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Contohnya sistem manusia.

7. Sistem Tertutup (*Closed System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teori, sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

8. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dengan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer. (Asbon Hendra : 2012 ; 160-161)

II.2. Sistem Informasi

Menurut Robert A. Leitch, sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporang yang diperlukan. (Asbon Hendra : 2012 ; 169)

Sistem informasi merupakan suatu tatanan yang saling terkait antara unsur data, *software*, *hardware*, sumber daya manusia dan kelembagaan serta aturan mainnya. Mengembangkan sistem informasi berarti mengembangkan seluruh unsur tersebut secara menyeluruh, tidak bisa dilakukan secara menyeluruh, tidak bisa dilakukan secara parsial atau sendiri-sendiri. (Sri Andriani : 2015 ; 25)

II.2.1 Komponen-Komponen Sistem Informasi

Sistem informasi memiliki beberapa komponen – komponen diantaranya adalah sebagai berikut :

1. Perangkat Keras (*Hardware*)

Komponen perangkat keras (*hardware*) ini mencakup piranti – piranti fisik seperti komputer dan printer.

2. Perangkat Lunak (*Software*)

Komponen perangkat lunak (*software*) atau program yaitu sekumpulan instruksi yang memungkinkan perangkat keras untuk memproses data.

3. Aturan (*Prosedure*)

Komponen aturan (*prosedure*) merupakan sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

4. Orang (*User*)

Komponen orang (*User*) yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.

5. Basis Data (*Database*)

Komponen basis data (*database*) yaitu sekumpulan tabel hubungan, dan lain – lain yang berkaitan dengan penyimpanan data.

6. Jaringan Komputer dan Komunikasi Data

Komponen jaringan komputer dan komunikasi data yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai. (Sri Andriani : 2015 ; 26)

II.3. Sistem Pendukung Keputusan / *Decision Support Sistem (DSS)*

II.3.1 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan penggabungan sumber–sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah –masalah semistruktur. (Nopita Sari : 2015)

Sistem pendukung keputusan merupakan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data

atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan anatar dua komponen lainnya, terdiri dari satu atau lebih kapasitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan. (Irsanti Merina Tamala Sari, Dita Rizki Amalia : 2015)

Menurut Keen dan Scoot Morton : “Sistem Pendukung Keputusan merupakan penggabungan sumber – sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah – masalah semi struktur.” (Muhammad Reza Okaviana, Rani Susanto : 2014)

II.3.2 Tujuan Sistem Pendukung Keputusan

Tujuan dari DSS adalah:

1. Membantu dalam pengambilan keputusan atas masalah yang terstruktur.
2. Memberikan dukungan atas pertimbangan manager dan bukan dimaksudkan untuk mengganti fungsi manager.
3. Meningkatkan efektifitas keputusan yang diambil lebih dari pada perbaikan efesiensinya.
4. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat. (Irsanti Merina, Dita Rizki : 2015)

II.3.3 Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan terdiri atas 3 (tiga) komponen yaitu :

1. Subsistem (*Database*)

Subsistem data (*database*) merupakan komponen sistem pendukung keputusan yang berguna sebagai penyedia data bagi sistem.

2. Subsistem Model (*ModelBase*)

Model adalah suatu tiruan dari alam nyata.

3. Subsistem Dialog (*User System Interface*)

Subsistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif, yang dikenal dengan subsistem dialog. (Irsanti Merina, Dita Rizki : 2015)

II.3.4 Keuntungan Sistem Pendukung Keputusan

Dengan berbagai karakter khusus yang dimiliki sistem pendukung keputusan, SPK dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah :

1. SPK memperluas kemampuan pengambilan keputusan dalam memproses data atau informasi bagi pemakainya.
2. SPK membantu pengambil keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu SPK, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun ia dapat

menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan. (Shanti Wirana : 2015)

II.4. Multifactor Evaluation Process (MFEP)

Metode *Multifactor Evaluation Process* adalah suatu metode yang memecah-mecah suatu situasi yang kompleks, tidak terstruktur, ke dalam bagian-bagian komponennya. Metode *Multifactor Evaluation Process* adalah penspesifikasian dimensi dari permasalahan, dimana pembuat keputusan harus mengevaluasi setiap alternatif kriteria yang majemuk secara spesifik. (Nopita Sari : 2015)

Multifactor Evaluation Process (MFEP) adalah metode kuantitatif yang menggunakan *Weighting System*. Dalam pengambilan keputusan multi faktor, pengambil keputusan secara subyektif dan intuitif menimbang berbagai faktor atau kriteria yang mempunyai pengaruh penting terhadap alternatif pilihannya. Untuk keputusan yang berpengaruh secara strategis, lebih dianjurkan menggunakan sebuah pendekatan kuantitatif seperti MFEP. (Muhammad Reza Oktaviana : 2014)

II.4.1 Konsep Dasar Penggunaan Metode MFEP

Dibawah ini merupakan langkah-langkah proses perhitungan menggunakan metode MFEP, yaitu :

1. Menentukan faktor / kriteria dan bobot faktor / kriteria dimana total pembobotan harus sama dengan 1 atau 100 (pembobotan = 1), yaitu *factor weight*.

2. Mengisikan nilai untuk setiap faktor (kriteria) yang mempengaruhi dalam pengambilan keputusan dari data-data yang akan diproses, nilai yang dimasukkan dalam proses pengambilan keputusan merupakan nilai objektif, yaitu sudah pasti yaitu *factor evaluation* yang nilainya antara 0 - 1 (0-100).
3. Proses perhitungan *weight evaluation* yang merupakan proses perhitungan bobot antara *factor weight* dan *factor evaluation* dengan serta penjumlahan seluruh hasil *weight evaluations* untuk memperoleh total hasil evaluasi.

Penggunaan model MFEP dapat direalisasikan dengan contoh berikut :

$$\mathbf{WE = FW \times E}$$

$$\mathbf{WE = WE1 + WE2 + WE3.....WEn}$$

Tabel II.1. Rumus MFEP

(Sumber : Diwanda, Isnawaty dan Sagala : 2016 ; 343)

Keterangan :

WE = Weighted Evaluation

FW = Factor Weight

E = Evaluation

WE = Total Weighted Evaluation

II.5. Sales Promotion Girl (SPG)

II.5.1 Pengertian Sales Promotion Girl (SPG)

Menurut Poerwodarminto, *Sales Promotion Girl* merupakan suatu profesi yang bergerak dalam pemasaran atau promosi suatu produk. Profesi ini biasanya menggunakan wanita yang mempunyai karakter fisik yang menarik sebagai usaha untuk menarik perhatian konsumen.

Pengertian SPG ditinjau dari sistem pemasaran, Nitisemito berpendapat bahwa sebagai salah satu pendukung pemasaran suatu produk maka diperlukan tenaga promosi suatu produk sehingga mampu menarik konsumen. Selanjutnya, dengan kemampuan berpromosi yang dimiliki seorang sales promotion girl akan mampu memberikan berbagai informasi yang berkaitan dengan produk.

Retnasih menyatakan: "*Sales Promotion Girl* adalah seorang perempuan yang direkrut oleh perusahaan untuk mempromosikan produk." Pendapat ini melihat keberadaan *Sales Promotion Girl* dari fungsinya yaitu sebagai pihak presenter dari suatu produk. Lebih lanjut pendapat ini mengilustrasikan bahwa tugas utama dari *Sales Promotion Girl* adalah promosi produk, pada umumnya status *Sales Promotion Girl* adalah karyawan kontrak. Mereka dikontrak dalam kurun waktu tertentu untuk mempromosikan produk.

Raharti menyatakan bahwa terdapat beberapa persyaratan yang harus dipenuhi oleh *Sales Promotion Girls*, yaitu :

1. Performance

Performance ini merupakan tampilan fisik yang dapat diindera dengan menggunakan penglihatan. Dalam perspektif ini, *performance* juga mengilustrasikan tentang pembawaan seseorang. Pembawaan ini diukur dari

penampilan outlook (penampilan fisik) dan desain dress code (desain pakaian), ukuran dari pembawaan ini subyektif (setiap orang dimungkinkan berbeda).

2. *Communicating Style*

Komunikasi mutlak harus terpenuhi oleh sales promotion girl karena melalui komunikasi ini akan mampu tercipta interaksi antara konsumen dan *Sales Promotion Girl*. Komunikasi ini diukur dari gaya bicara dan cara berkomunikasi. Pengukuran atas communicating style ini dikembalikan kepada konsumen karena bisa bersifat subyektif.

3. *Body Language*

Body language ini lebih mengarah pada gerakan fisik (lemah lembut, lemah gemulai, dan lainnya). Gerak tubuh ketika menawarkan produk dan sentuhan fisik (*body touch*) adalah deskripsi dari *body language* ini. Pengukuran atas *body language* dikembalikan kepada konsumen karena bisa bersifat subyektif. (Syukron Ma'Mun : 2015)

II.5.2 Jenis-jenis *Sales Promotion Girl* (SPG)

Sales Promotion Girl dikategorikan menjadi dua jenis, yaitu:

1. SPG Tetap (SPG *Reguler*)

SPG *reguler* yang merupakan pegawai tetap perusahaan tersebut biasanya memiliki pengetahuan produk yang lebih rinci karena memang merupakan bagian dari perusahaan tersebut. SPG ini banyak di temui di *mall* dan *departement store*. SPG *reguler* ini memiliki sistem upah bulanan atau gaji bulanan yang tetap untuk gaji pokok nya. Sesuai dengan namanya profesi SPG *reguler* ini bertugas menjual atau mempromosikan

suatu produk. Sistem kerja SPG *regular* sangat teratur dengan intensitas rutin yang tinggi, mulai dari penghitungan stok barang/produk hingga pembukuan atau jurnal penjualan harian yang harus di laporkan tiap bulannya. SPG *regular* terikat kontrak dan bekerja dalam jangka waktu sesuai dengan kontrak tersebut. Dalam SPG *regular* ini, ada yang disebut sebagai SPG *mobile*. SPG *mobile* ini biasanya di kontrak oleh suatu brand. Kebanyakan memang brand rokok. Namun sekarang SPG *mobile* bukan hanya SPG rokok namun brand provider ponsel pun sekarang banyak yang memakai jasa SPG *mobile*. Para SPG *mobile* ditemani *Team Leader* (TL) yang sudah ditentukan areanya masing-masing. Tugas para SPG *mobile* yaitu berjualan yang diawasi oleh TL.

2. SPG Kontrak (SPG Event)

SPG *event* bekerja hanya pada *event* yang digelar saja. Artinya, SPG *event* hanya bekerja di waktu tertentu saja. SPG *event* adalah pekerjaan dengan tugas utama yaitu mempromosikan suatu produk. SPG *event* seringkali di temui di *mall-mall* yang bersanding dengan sebuah *booth* atau stand pameran untuk suatu produk. Pameran besar hingga kelas menengah sudah pasti membutuhkan jasa SPG *event* ini yang anda bisa dapat kan melalui para agency SPG *event*. Berbeda dengan SPG *regular*, SPG *event* tidak memiliki dan menguasai pengetahuan mengenai produk sepenuhnya.

(Syukron Ma'Mun : 2015)

II.6. *Microsoft Visual Studio 2010*

Microsoft Visual Studio adalah sebuah *Integrated Development Environment* buatan Microsoft Corporation. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). *Visual Studio* mencakup sebuah kode editor yang didukung oleh fitur *intellisense* atau yang disebut dengan *code refactoring*. *Debugger* telah terintegrasi bekerja pada level *source*, level *debugger*, level *debugger* mesin. *Tool built in* mencakup *form designer* untuk membangun sebuah aplikasi GUI, *Web Designer*, *Class Designer* dan *Database Schema Designer*. (Wahana Komputer : 2013)

Microsoft Visual Basic 2010 merupakan program aplikasi untuk membentuk program dari bahasa pemrograman klasik, legendaris dan tiada duanya yang paling banyak dipakai oleh jutaan *programmer* dan tercatat sebagai program yang paling disukai oleh mayoritas orang. (Sri Andriani : 2015 : 54)

Visual Basic 6.0 adalah salah satu aplikasi pemrograman under *Windows* yang berbasis pada visual atau grafis. Aplikasi ini dikeluarkan oleh *Microsoft Cooperation* yang juga pemilik dari sistem operasi *Microsoft Windows*. Pada awalnya BASIC (*Beginner's Allpurpose Symbolic Instruction Code*) adalah bahasa pemrograman yang merupakan awal dari bahasa pemrograman tingkat tinggi sesudahnya, yang berbasis DOS (*Diskette Operating System*). BASIC memiliki struktur bahasa yang sulit dan memiliki tampilan yang tidak menarik,

dengan kemajuan teknologi maka diperlukan suatu aplikasi pemrograman yang bukan hanya cepat tapi juga menarik dan *user friendly* atau mudah digunakan. Maka *Microsoft* mengembangkan *Visual Basic* sebagai salah satu bahasa pemrograman tingkat tinggi berdasarkan dari bahasa pemrograman BASIC. (Indra Kanedi : 2013)

II.7. *Microsoft SQL Server 2008*

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

Microsoft merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan *single* prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
2. Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan *system* operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini:

1. Versi *Compact*, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi desktop pada *SQL Server 2000*. Versi ini juga digunakan pada *handled divice* seperti Pocket PC, PDA, *SmartPhone*, Tablet PC.
2. Versi *Express*, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini berbeda dengan versi *compact*) dan paling cocok untuk latihan para pengembang aplikasi. Versi ini memuat *Express Manager Standar*, integrasi dengan CLR dan XML. (Wenny Widya : 2014)

II.8. Unified Modelling Language (UML)

II.8.1 Pengertian UML

Menurut Whitten (2004) dalam Rumambi (2005), UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat *software* berorientasi objek, karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma *object oriented*. UML juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. UML terdiri dari beberapa diagram, yaitu *usecase diagram*, *class diagram*, *state diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, dan *deployment diagram*. (Muhammad Ilham Alfarisyi, Rispianda, Khuria Amila : 2014)

Unified Modelling Language (UML) adalah sistem arsitektur yang bekerja dalam *OOAD (Object-Oriented Analysis Design)* dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi dan mendokumentasi *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*, dapat berupa model, deskripsi, atau *software*) yang terdapat dalam sistem *software*. (Sri Andriani : 2015 ; 37)

Adapun pengertian UML menurut Yasin “*Unifield Modeling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. (Henny Destiana : 2014)

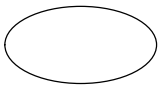
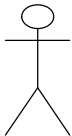

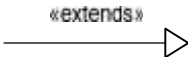

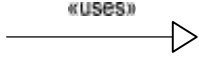
Adapun beberapa jenis diagram pada UML yang dapat membantu perancangan sistem menurut Yasin adalah sebagai berikut :

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* mempresentasikan sebuah interaksi antar *actor* dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien dan merancang *testcase* untuk semua *feature* yang ada pada sistem. (Henny Destiana : 2014)

Berikut ini adalah simbol-simbol yang ada pada *use case diagram* :

Tabel II.2. Simbol-Simbol Use Case Diagram

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor / <i>Actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komuniikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>Extends</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / <i>Generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / <i>Include / Uses</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p>

(Sumber : Muhammad Hambali : 2016 ; 32-33)

2. Class Diagram

Class Diagram membantu kita dalam visualisasi *structure* kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram* memperlihatkan hubungan antarkelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem. (Sri Andriani : 2015 ; 53)

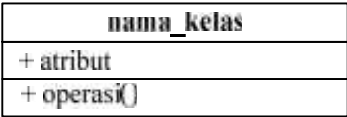
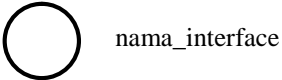
Class Diagram juga menunjukkan atribut- atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, *Attribute*, Operasi, *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. (Sri Andriani : 2015 ; 53)


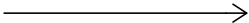
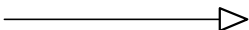

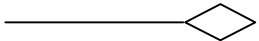
Diagram kelas atau *Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut ini adalah simbol-simbol yang ada pada *class diagram* :

Tabel II.3. Simbol-Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi / <i>Generalization</i> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>)






(Sumber : Muhammad Hambali : 2016 ; 33-34)

3. Activity Diagram

Activity diagram menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). (Henny Destiana : 2014)

Berikut ini adalah simbol-simbol yang ada pada *activity diagram* :

Tabel II.4. Simbol-simbol *Activity Diagram* (Diagram Aktivitas)

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	<i>Decision</i> , atau pilihan untuk mengambil keputusan.
Penggabungan / <i>Join</i> 	Arah tanda panah alur proses.
Status Akhir 	Titik akhir atau akhir dari aktivitas.

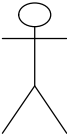
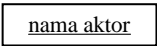

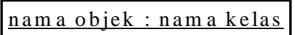
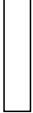
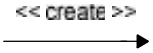
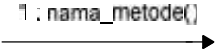
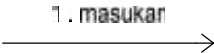
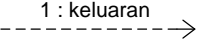

(Sumber : Muhammad Hambali : 2016 ; 35)

4. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Berikut adalah simbol-simbol yang ada pada *Sequence Diagram*:

Tabel II.5. Simbol-Simbol *Sequence Diagram*

Simbol	Keterangan
<p>Aktor / <i>Actor</i></p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada</p>

×	objek yang di akhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>
---	--

(Sumber : Muhammad Hambali : 2016 ; 36-37)