

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem informasi merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat.

Jenis sistem informasi berbasis komputer

1. Pengolahan Data. Pengolahan data elektronik – *electronic data processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. EDP adalah aplikasi sistem informasi akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.
2. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
3. Sistem Pendukung Keputusan – *Decision Support Systems (DSS)*. DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen. Contoh adalah penggunaan *spreadsheet* untuk melakukan analisis “*what if*” dari data operasi atau anggaran.
4. Sistem pakar – Expert System (ES) adalah sistem informasi berbasis pengetahuan yang memanfaatkan pengetahuannya tentang bidang aplikasi

5. tertentu untuk bertiindak seperti seorang konsultan ahli bagi pemakainya.
(Mujilan : 2012 : 2).

II.2. Informasi Dan Keputusan

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb.

Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomik pada saat ia mendukung keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: ekstern dan intern. Pemakai ekstern mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama para manajer, kebutuhannya bervariasi tergantung pada tingkatannya dalam organisasi atau terhadap fungsi yang mereka jalankan.

Dari sudut pandang organisasi, dibedakan dua kelompok besar informasi akuntansi :

1. Mandatori : Berbagai badan pemerintah, swasta, dan perundang-undangan menetapkan Undang-Undang untuk mencatat pelaporan. Untuk memenuhi persyaratan informasi mandatori, pertimbangan utama adalah meminimalkan biaya sejalan dengan pemenuhan standar keandalan dan kemanfaatan.
2. Bebas : Pertimbangan utama adalah bahwa manfaat yang dicapai melebihi biaya untuk menghasilkannya. (Mujilan ; 2012 : 1).

II.3. Sistem Informasi

Sistem informasi dapat diartikan sebagai suatu sistem di dalam organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur, dan pengendalian yang ditujukan untuk mendapatkan jalur kombinasi yang penting.

Di dalam suatu sistem informasi terdapat beberapa komponen – komponen, yaitu :

1. Perangkat keras (*Hardware*) : mencakup piranti-piranti fisik seperti monitor, printer, scanner, keyboard dan mouse.
2. Perangkat Lunak (*Software*) atau program : sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.
3. Prosedur : sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

4. Orang : Semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan sistem informasi.
5. Basis Data (Database) : sekumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data : sistem penghubung yang memungkinkan satu sumber dipakai secara bersama atau diakses oleh sejumlah pemakai. (Sulindawati : 2010 ; 4)

II.4. Sistem Informasi Akuntansi (SIA)

Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. Sistem Informasi Akuntansi mewujudkan perubahan ini secara manual atau terkomputerisasi.

Sistem Informasi Akuntansi juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi (Xu, 2009). Saat ini, digital dan informasi *online* semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai factor yang terkait ketika mengatur sistem informasi akuntansi.

Sistem Informasi Akuntansi pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.

2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumber daya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus Mujilan : 2012 : 3).

II.5. Metode Ekuitas

Metode ekuitas adalah metode pencatatan investasi yang pada awal perolehan investor mencatat investasi sebesar biayanya, dividen maupun distribusi laba dicatat sebagai pengurang akun investasi. Nilai investasi ditambah atau dikurangi dengan bagian laba/rugi investor pada perusahaan asosiasi setelah tanggal perolehan

Contoh Studi Kasus :

PT XYZ memiliki investasi penyertaan saham di PT ABC. Total yang dibayar untuk mendapatkan saham PT ABC adalah sebesar 100.000. 000. Nilai tersebut adalah 25% kepemilikan terhadap PT ABC.

PT ABC mencatatkan laba sebesar 50.000.000. PT ABC membagikan dividen 20.000.000 untuk para pemegang saham. (Hery:2011)

Jurnal Metode Ekuitas

Saat perolehan investasi:

| | |
|-----------------------|-----------------|
| Investasi pada PT ABC | [D] 100.000.000 |
| Kas/Bank | [K] 100.000.000 |

Saat PT ABC mencatatkan laba:

| | |
|-----------------------|----------------|
| Investasi pada PT ABC | [D] 12.500.000 |
| Pendapatan Investasi | [K] 12.500.000 |

(25% x 50.000.000 = 12.500.000)

Saat PT ABC membagikan dividen:

| | |
|-----------------------|---------------|
| Kas/Bank | [D] 5.000.000 |
| Investasi pada PT ABC | [K] 5.000.000 |

(25% x 20.000.000 = 5.000.000)

II.6. Investasi

Investasi adalah suatu aktiva yang digunakan perusahaan untuk pertumbuhan kekayaan (*accretion of wealth*) melalui distribusi hasil investasi (seperti bunga, royalty, dividen, dan uang sewa) untuk apresiasi nilai investasi atau untuk manfaat lain bagi perusahaan yang berinvestasi seperti manfaat yang diperoleh melalui hubungan perdagangan yang dilakukan oleh perusahaan yang berinvestasi".(Hery:2011)

II.7. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional.

1. Bentuk Normal Pertama (1 NF)

Contoh yang kita gunakan di sini adalah sebuah perancangan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat dituliskan sebagai berikut :

PEMASOK (P#, Status, Kota, b#, qty) di mana

p# : kode pemasok (kunci utama)

status : kode status kota

Kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasok.

Sebuah tabel relasional secara defenisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomi. Ini berarti kolom-kolom tidak mempunyai nilai berulang. Tabel II.1. menunjukkan tabel pemasok dalam 1 NF.

Tabel II.1. Normalisasi Pertama Pemasok

| P# | Status | Kota | B# | Qty |
|-----------|---------------|-------------|-----------|------------|
| P1 | 20 | Yogyakarta | B1 | 300 |
| P1 | 20 | Yogyakarta | B2 | 200 |
| P1 | 20 | Yogyakarta | B3 | 400 |
| P1 | 20 | Yogyakarta | B4 | 200 |
| P1 | 20 | Yogyakarta | B5 | 100 |
| P1 | 20 | Yogyakarta | B6 | 100 |
| P2 | 10 | Medan | B1 | 300 |
| P2 | 10 | Medan | B2 | 400 |
| P3 | 10 | Medan | B2 | 200 |
| P4 | 20 | Yogyakarta | B2 | 200 |

| | | | | |
|----|----|------------|----|-----|
| P4 | 20 | Yogyakarta | B4 | 300 |
| P4 | 20 | Yogyakarta | B5 | 400 |

(Sumber : Simarmata, dkk : 2010).

2. Bentuk Normal Kedua (2 NF).

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1 NF, tetapi tidak pada 2 NF, sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1 NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Tabel pemasok berada pada 1 NF, tetapi tidak pada 2 NF karena status dan kota tergantung secara fungsional hanya pada kolom p# dari kunci gabungan (p#, b#). Ini dapat digambarkan dengan membuat daftar ketergantungan fungsional.

P# \longrightarrow Kota, Status

Kota \longrightarrow Status

(P#, B#) \longrightarrow qty

Proses mengubah tabel 1 NF ke 2 NF adalah :

- a. Tentukan sembarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukan.

- c. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru penentu akan menjadi kunci utama pada tabel baru.
- d. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
- e. Tabel asal bisa diberi nama baru.

Pada contoh, kita memindahkan kolom p#, status, dan kota ke tabel baru yang disebut pemasok2. Kolom p# menjadi kunci utama tabel ini. Tabel II.2. menunjukkan hasilnya.

Tabel II.2. Tabel Bentuk Normal Kedua (2NF).

Pemasok2
Barang

| P# | Status | Kota |
|----|--------|------------|
| P1 | 20 | Yogyakarta |
| P2 | 10 | Medan |
| P3 | 10 | Medan |
| P4 | 20 | Yogyakarta |
| P5 | 30 | Bandung |

| P# | B# | Qty |
|----|----|-----|
| P1 | B1 | 300 |
| P1 | B2 | 200 |
| P1 | B3 | 400 |
| P1 | B4 | 200 |
| P1 | B5 | 100 |
| P1 | B6 | 100 |
| P2 | B1 | 300 |
| P2 | B2 | 400 |
| P3 | B2 | 200 |
| P4 | B2 | 200 |
| P4 | B4 | 300 |
| P4 | B5 | 400 |

(Sumber : Simarmata, dkk : 2010)

3. Bentuk Normal Ketiga (3 NF).

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga (3 NF) jika tabel sudah berada pada 2 NF dan setiap kolom yang bukan

kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama. Tabel barang sudah dalam bentuk normal ketiga. Kolom bukan kunci, qty, tergantung sepenuhnya pada kunci utama (p#, b#). Pemasok masih berada pada 2 NF, tetapi belum berada pada 3 NF karena dia mengandung ketergantungan transitif. Ketergantungan transitif terjadi ketika sebuah kolom bukan kunci, yang ditentukan oleh kunci utama, menentukan kolom lainnya. Konsep ketergantungan transitif dapat digambarkan dengan menunjukkan ketergantungan fungsional pada pemasok2, yaitu :

Pemasok2. p# \longrightarrow Pemasok2, status

Pemasok2. p# \longrightarrow Pemasok2, kota

Pemasok2. kota \longrightarrow Pemasok2, status

Perlu dicatat bahwa pemasok2, status ditentukan, baik oleh kunci utama p#, maupun kolom bukan kunci, kota

Proses mengubah tabel menjadi 3 NF adalah :

- a. Tentukan semua penentu selain kunci utama dan kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom yang ditentukannya.
- c. Pindahkan kolom yang ditentukan dari tabel asal ke tabel baru. Penentu menjadi kunci utama tabel baru.
- d. Hapus kolom yang baru saja dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.

- e. Tabel asal bisa diberi nama baru.

Untuk mengubah PEMASOK2 menjadi 3 NF, kita membuat tabel baru yang disebut KOTA_STATUS dan memindahkan kolom kota dan status ke tabel baru. Status dihapus dari tabel diberi nama baru PEMASOK_KOTA. Tabel II.4 menunjukkan hasilnya.

Tabel II.3. Tabel Bentuk Normal Ketiga (3 NF)

| PEMASOK_KOTA | | KOTA_STATUS | |
|--------------|------------|-------------|--------|
| P# | Kota | Kota | Status |
| P1 | Yogyakarta | Yogyakarta | 20 |
| P2 | Medan | Medan | 10 |
| P3 | Medan | Bandung | 30 |
| P4 | Yogyakarta | Semarang | 40 |
| P5 | Bandung | | |

(Sumber : Simarmata, dkk : 2010).

4. Bentuk Normal Boyce Code (BCNF)

Setelah 3 NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3 NF sudah cukup karena sangat jarang entitas yang berada pada 3 NF bukan merupakan 4 NF dan 5 NF. Lebih lanjut, mereka berpendapat bahwa keuntungan yang didapat mengubah entitas ke 4 NF dan 5 NF sangat kecil sehingga tidak perlu dikerjakan. Bentuk Normal Boyce- Code (BCNF) adalah versi 3 NF lebih teliti dan berhubungan dengan tabel relasional yang mempunyai (a) banyak kunci kandidat (b) kunci kandidat gabungan, dan (c) kunci kandidat yang saling tumpang tindih.

BCNF didasarkan pada konsep penentu. Sebuah kolom penentu adalah kolom di mana kolom-kolom lain sepenuhnya tergantung secara fungsional. Sebuah tabel relasional berada pada BCNF jika dan hanya setiap penentu adalah kunci kandidat.

5. Bentuk Normal Keempat (4 NF)

Sebuah tabel relasional berada pada bentuk normal keempat (4 NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional.

Bentuk normal keempat (4 NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

Defenisi secara formal diberikan oleh CJ. Date, yaitu :

Misalnya, ada sebuah tabel relasional R dengan kolom A, B dan C, Maka $R.A \twoheadrightarrow R.B$ (kolom A menentukan kolom B).

Adalah benar jika dan hanya jika himpunan nilai B yang cocok dengan pasangan nilai A dan nilai C pada R hanya tergantung pada nilai A dan tidak tergantung pada nilai C.

MVD selalu terjadi dalam pasangan, yaitu $R.A \twoheadrightarrow R.B$ dipenuhi jika dan hanya jika $R.A \twoheadrightarrow R.C$ dipenuhi pula.

6. Bentuk Normal Kelima (5 NF).

Sebuah tabel berada pada bentuk normal kelima jika dia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil.

Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*). Ketergantungan gabungan berarti sebuah tabel, setelah deskomposisi menjadi tiga atau lebih tabel yang lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain 5 NF menunjukkan ketika sebuah tabel tidak dapat dideskomposisi lagi (Janner Simarmata, 2012).

II.8. Basis Data (*Database*)

Secara sederhana database (basis data/ pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat (Kadir, 2004). Pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media penguinat yang disebut *hard disk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk database.

Pengaplikasian database dapat kita lihat dan rasakan dalam keseharian kita. Database ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (anjungan tunai mandiri/ *automatic teller machine*) bank karena bank telah mempunyai database tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks

database sebenarnya kita sudah melakukan perubahan (*update*) data pada database di bank.

Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep database. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir (*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya.

Pemahaman tentang database ini dapat didekatkan pada konsep akuntansi. Kita bisa umpamakan bahwa ketika kita melakukan proses akuntansi secara manual, kita menuliskan suatu catatan ke dalam lajur dan kolom buku. Mulai dari jurnal, buku besar, buku pembantu kita memasukkan catatan satu demi satu. Melihat buku akuntansi tersebut, sebenarnya kita sudah melihat konsep database, yang jika dikelola dengan komputer masih diperlukan penyesuaian dalam membentuk kolom-kolomnya. (Mujilan : 2012 : 23)

II.8.1. Model Database

Model database yang saat ini banyak digunakan adalah model database relational. Imam (2008) menyebutkan “Model database ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*). Tabel-tabel yang ada dihubungkan (*relationship*) sedemikian rupa menggunakan *field-field* kunci (*key field*) sehingga dapat meminimalkan duplikasi data.”

Model database relational ini dapat kita kenal konsepnya mulai dari yang paling sederhana misalnya dengan penerapan program aplikasi *excel*. Meskipun untuk pengelolaan database secara luas *excel* jarang digunakan dan kurang mencukupi, namun untuk melihat konsep database dan konsep membangunnya program ini dapat dimanfaatkan. *Excel* mempunyai baris yang disebut *raw* dan mempunyai kolom. Kemudian item data merupakan sel atau pertemuan antara baris dan kolom. Tabel-tabel dapat diumpakan apabila kita menggunakan tabel dalam suatu *sheet* tertentu. Data dari berbagai tabel dapat diambil dari tabel lain menggunakan perintah *look up* yang berdasarkan kode kunci tertentu. Kode kunci tersebut berada pada suatu kolom tertentu, yang dalam konsep database relational disebut sebagai *key field* tadi. (Mujilan : 2012 : 24)

II.9. Unified Modeling Language (UML)

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. (Chonoles, 2003) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*. (Widodo Dan Herlawati:2011).

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

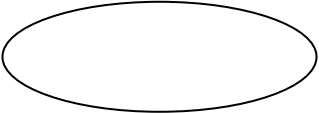
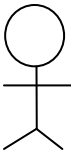
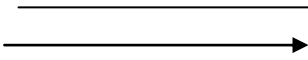


UML telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. (Widodo Dan Herlawati:2011).

II.9.1. Use Case Diagram

Menurut Yuni Sugiarti (2013) *Use Case Diagram* merupakan permodelan untuk menggambarkan kelakuan (Behavior) sistem yang akan dibuat. *Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat . *Use Case Diagram* digunakan untuk mengetahui fungsi-fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case*, yaitu:

Tabel II.4. Diagram Use Case

| Gambar | Keterangan |
|---|--|
|  | Use Case menggambarkan fungsionalitas yang disediakan system sebagai unit yang bertukar pesan antar unit dengan actor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case |
|  | Actor atau actor adalah abstraction dari orang atau system yang lain dengan mengaktifkan fungsi dari target system. Untuk mendefenisikan actor, harus ditentukan pembagian tenaga kerja dan tugas tugas yang berkaitan dengan peran para konteks target system. Orang atau system biasa muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan use case, tetapi tidak memiliki control terhadap use case |
|  | Asosiasi antara actor dengan use case digambarkan dengan garis tanda panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data |
|  | Include, Merupakan didalam use case oleh use case orang lain. Contohnya adalah pemanggilan sebuah fungsi program |
|  | Extends, merupakan perluasan dari use case jika kondisi atau syarat terpenuhi |

(Sumber : Sugiarti ; 2013)

II.9.2 Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas

system bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh system. Diagram aktivitas mendukung perilaku paralel.




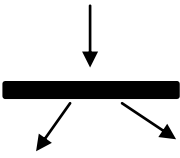
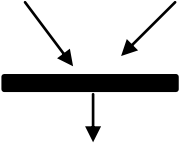
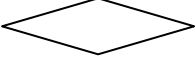
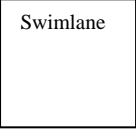
Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis system yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari system/ *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya.

Activity diagram menggambarkan berbagai alir aktivitas dalam system 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 transisi di trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena *Activity diagram* tidak menggambarkan *behavior* internal sebuah system (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. (Sugiarti: 2013)

Tabel II.5. Diagram Aktivitas

| Gambar | Keterangan |
|---|---|
|  | <i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas |
|  | End Point, akhir aktivitas |
|  | <i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis. |
|  | Fork (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu |
|  | <i>Join</i> atau penggabungan atau raje, digunakan untuk menunjukkan adanya dekomposisi |
|  | <i>Decision</i> , menggambarkan pilihan untuk menampilkan keputusan , True Atau false |
|  | <i>Swimlane</i> , pembagian activity diagram untuk menunjukkan siapa melakukan apa |

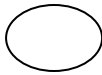
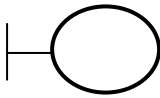




(Sumber : Sugiarti; 2013)

II.9.3. Sequence Diagram

Diagram Sequence menggambarkan kelakuan/ perilaku objek pada *use* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sequences maka harus diketahui objek-objek yang terlibat dalam sebuah *use* beserta metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek itu.

Banyak Diagram Sekuence yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram sekuence sehingga banyak *use case* yang didefinisikan maka diagram sekuence yang harus dibuat juga semakin banyak. (Sugiarti, 2013)

Tabel II.6. Diagram Aktivitas

| Gambar | Keterangan |
|---|---|
|  | Entity class, merupakan bagian dari system yang berisi kumpulan kelas beruoa entitas yang membentuk gambaran awal system dan menjadi landasan untuk menyusun basis data |
|  | Boundary Class, berisis kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih actor dengan system |
|  | Controll Class, Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas |
|  | Messages, symbol mengirim pesan antar class |
|  | Activation, Activation meewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi |
|  | Lifeline, garis titik-titik yang terhubung dengan objek sepanjang lifeline terhadap activation. |

(Sumber : Sugiarti; 2013)

II.9.4. Diagram Kelas

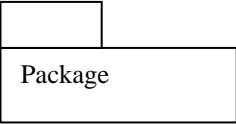
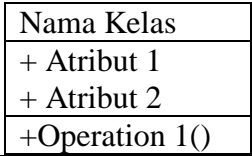

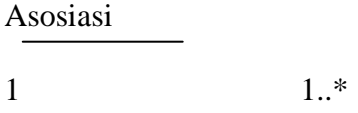
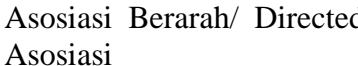
Diagram class menggambarkan struktur system dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system. Kelas memiliki apa yang disebut atribut dan metode atau operasi yaitu:

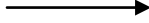
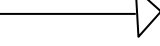
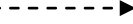
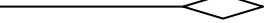
1. atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
2. atribut Mendeskripsikan property dengan sebaris teks didalam kotak kelas tersebut.
3. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Diagram kelas mendeskripsikan jenis-jenis objek dalam system dan berbagai hubungan statis yang terdapat diantara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

Diagram kelas menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti *Containment*, pewarisan, asosiasi, dan lain-lain.

Tabel II.7. Diagram Aktivitas

| Simbol | Deskripsi |
|--|--|
| Package  | Package merupakan sebuah bungkus dari satu atau lebih kelas |
| Operasi  | Kelas pada struktur sistem |
| Antarmuka Interface  | Sama dengan konsep Interface dalam pemrograman berorientasi objek |
| Asosiasi  | Relasi antar kelas dengan makna umum yang, asosiasi biasanya juga disertai dengan multiplicity |
| Asosiasi Berarah/ Directed Asosiasi  | Relasi antar kelas dengan makna kelas yang satu |

| | |
|---|--|
|  | digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multiplicity |
| Generalisasi  | Relasi antar kelas dengan makna generalisasi – spesialisasi (Umum Khusus) |
| Kebergantungan/ Defedency  | Relasi antar kelas dengan makna kebergantungan antar kelas |
| Agregasi  | Relasi antar kelas dengan makna semua bagian (Whole-Part) |

(Sumber : Sugiarti; 2013)

II.10. Bahasa Pemograman *Microsoft Visual Studio 2010*

Microsoft Visual Studio 2008 merupakan kelanjutan dari *Microsoft Visual Studio* sebelumnya, yaitu *Visual Studio. Net 2003* yang diproduksi oleh Microsoft. Pada bulan Februari 2002 *Microsoft* memproduksi teknologi. *Net Framework* versi 1.0, teknologi. *Net* ini didasarkan atas susunan berupa *Net Framework*, sehingga setiap produk baru yang terkait dengan teknologi. *Net* akan selalu berkembang mengikuti perkembangan. *Net Frameworknya*. Pada perkembangannya nantinya mungkin untuk membuat program dengan teknologi. *Net* memungkinkan para pengembang perangkat lunak akan dapat menggunakan lintas sistem operasi, yaitu dapat dikembangkan di sistem operasi windows juga dapat dijalankan pada sistem operasi lain, misalkan pada sistem operasi *Linux*, seperti yang telah dilakukan pada pemograman *Java* oleh *Sun Microsystem*. Pada

saat ini perusahaan-perusahaan sudah banyak mengupdate aplikasi lama yang dibuat *Microsoft Visual Basic 6.0* ke teknologi *.Net* karena kelebihan-kelebihan yang ditawarkan, terutama memungkinkan pengembang perangkat lunak secara cepat mampu membuat program *robust*, serta berbasiskan integrasi ke internet yang dikenal dengan *XML Web Service* (.Hirin : 2010)

II.11. MYSQL

MYSQL adalah perangkat lunak basis data server yang terkenal dan bersifat open source dengan dukungan driver yang luas dari berbagai vendor. *MySql* adalah seakuntansi implementasi dari system manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. (Supardi:2007).