

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi Akuntansi

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. 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.1.1. Kualitas Data Sistem Informasi Akuntansi

Seperti telah disebutkan di atas bahwa kualitas informasi merupakan hal yang sangat penting dalam sistem informasi. Informasi yang baik akan tergantung pula dengan data yang baik. Untuk membentuk sistem yang mampu menyediakan data dan informasi yang baik dan dapat dipercaya membutuhkan suatu usaha yang tidak mudah. Digarisbawahi hal penting dalam kaitannya dengan kualitas data.

1. Personil yang kompeten (*competent personnel*) agar sistem sesuai atau cocok digunakan.
2. Kontrol input (*input control*) merupakan hal bagian dari kontrol yang penting, apalagi dalam kasus transaksi *online*.
3. Jika perlu sediakan manajer kualitas data (DQ manager) yang bertanggung jawab pada kualitas data dalam sistem informasi akuntansi.

Berikut ini merupakan gambaran tentang pentingnya kualitas data dalam sistem informasi akuntansi dari hasil wawancara yang dilakukan oleh pada para manajer.

We have to monitor our cash balances fairly closely and it [data quality] is definitely one of the highest priorities. We have forecasts that need to be met, so we need to give ourselves early warning signals if a part of the business looks like it is not performing. The numbers will tell us that hopefully, so we can address the issue.

Kami harus memonitor saldo kas secara jelas dan kualitas data merupakan salah satu prioritas penting. Kami memperkirakan kebutuhan itu untuk dipenuhi, sehingga kami perlu mendapat tanda peringatan awal jika salah satu bagian dari bagian bisnis tidak berjalan sebagaimana mestinya. Suatu nomor aka memberitahukannya, jadi kita dapat mengenali permasalahannya. (Agustinus Mujilan, 2012:4).

II.2. Anggaran

Anggaran merupakan rencana keuangan masa datang yang mencakup harapan manajemen terhadap pendapatan, biaya, dan transaksi keuangan lain dalam masa satu tahun. Tahap penyusunan anggaran merupakan tahap yang sangat penting karena anggaran yang tidak efektif dan tidak berorientasi pada kinerja justru bisa menggagalkan program yang telah disusun sebelumnya. Tujuan penelitian ini untuk mengetahui proses penyusunan anggaran dan laporan realisasi anggaran pada BPM-PD Provinsi Sulawesi Utara. Penyusunan anggaran pada organisasi sektor publik dapat membantu mewujudkan akuntabilitas. Laporan Realisasi anggaran menyajikan ikhtisar sumber, alokasi, dan pemakaian sumber daya ekonomi yang dikelola oleh pemerintah daerah yang menggambarkan perbandingan antara anggaran dan realisasinya dalam satu periode pelaporan (Ramlah Basri : 2013).

II.3. Biaya Overhead Pabrik

Biaya overhead pabrik adalah elemen biaya produksi selain biaya bahan baku dan biaya tenaga kerja langsung, yang terdiri dari biaya bahan tidak langsung, biaya tenaga kerja tidak langsung dan biaya pabrik lainnya. Biaya overhead pabrik dapat digolongkan menjadi tiga bagian, yaitu:

1. Penggolongan Biaya Overhead Pabrik Menurut Sifatnya :
 - a. Biaya bahan penolong.
 - b. Biaya reparasi dan pemeliharaan.
 - c. Biaya tenaga kerja tidak langsung.
 - d. Biaya yang timbul sebagai akibat
 - e. Biaya yang timbul sebagai akibat berlalunya waktu.
 - f. Biaya overhead pabrik lainnya yang memerlukan pengeluaran uang tunai.
2. Penggolongan biaya overhead pabrik menurut perilakunya dalam hubungannya dengan perubahan volume produksi
 - a. Biaya Overhead Pabrik Tetap.
 - b. Biaya Overhead Pabrik Variabel.
 - c. Biaya Overhead Pabrik Semi variabel.
3. Penggolongan biaya overhead pabrik menurut hubungannya dengan departemen
 - a. Biaya Overhead Pabrik Langsung Departemen (*Direct Departmental Overhead Expenses*).
 - b. Biaya Overhead Pabrik Tidak Langsung Departemen (*Indirect Departmental Overhead Expenses*).

Penentuan tarif biaya overhead pabrik dapat dilakukan melalui tiga tahap, yaitu:

1. Menyusun Anggaran Biaya Overhead Pabrik.
2. Memilih Dasar Pembebanan Biaya Overhead Pabrik Kepada Produk.
3. Menghitung Tarif Biaya Overhead Pabrik (Ida Rosita : 2014 : 166).

II.4. Metode Alokasi Bertahap

Metode ini digunakan apabila jasa yang dihasilkan departemen jasa tidak hanya dinikmati oleh departemen produksi saja, melainkan digunakan pula oleh departemen jasa yang lain. Sebagai contoh bagian jasa terdiri dari bagian pembangkit tenaga listrik dan bagian reparasi. Bagian pembangkit tenaga listrik menggunakan sebagian jasa yang disediakan oleh bagian reparasi untuk memperbaiki mesin-mesin diesel. Sebaliknya bagian reparasi menggunakan pula sebagian jasa yang disediakan oleh bagian pembangkit listrik untuk menggerakkan peralatan reparasi. (Rinna Moniaga : 2014 : 735)

Satu metode untuk mengalokasikan biaya overhead pabrik bagian jasa apabila terjadi hal tersebut dapat dilakukan dengan menggunakan metode aljabar (Algebraic method). Dalam metode ini biaya overhead pabrik yang timbul di masing-masing bagian jasa dinyatakan dalam formula berikut:

$$X = a_1 + b_1 Y$$

$$Y = a_2 + b_2 X$$

$$Z = a_3 + b_3 Z, \text{ dst}$$

Keterangan:

X = jumlah BOP bagian jasa X setelah menerima alokasi BOP dari bagian jasa Y

Y = jumlah BOP bagian jasa Y setelah menerima alokasi BOP dari bagian jasa X

a1 = BOP bagian jasa X sebelum alokasi

a2 = BOP bagian jasa Y sebelum alokasi

b1 = Persentase penggunaan jasa bagian jasa Y oleh bagian X

b2 = Persentase penggunaan jasa bagian jasa X oleh bagian Y (Rinna Moniaga :
2014 : 735)

II.5. Model Jaringan

1. Peer to Peer

Model jaringan peer to peer (P2P) memungkinkan seorang user untuk membagi sumber daya yang ada pada komputernya, baik berupa data/ informasi, hardware, dan lain-lain serta mengakses sumber daya yang terdapat pada computer lain. Namun, model ini tidak dapat memiliki sebuah file server atau sumber daya yang terpusat, sehingga seluruh computer mempunyai kemampuan yang sama untuk memakai sumber daya yang tersedia di jaringan. Model ini disesain untuk jaringan berskala kecil dan menengah. (Macdons Madiun : 2015 : 4)

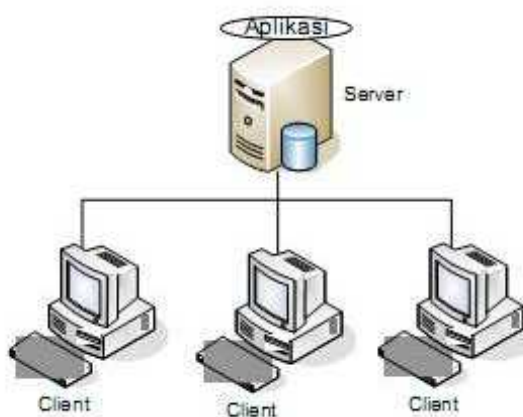
2. Client Server

Client-server merupakan tema utama dalam mengoptimalkan topologi yang telah ada didukung dengan *database* akademik dengan tujuan mempermudah birokrasi data serta *sharing resource* baik penggunaan perangkat

keras (*hardware*) maupun perangkat lunak (*software*) dengan tujuan terciptanya sebuah sistem yang efektif dan efisien. Tipe dalam mengoptimalkan jaringan ini adalah model jaringan *client server* yang terdiri dari sejumlah komputer dengan memakai satu atau beberapa komputer untuk dijadikan *server* dan dihubungkan dengan sejumlah *client*. Jadi pada jaringan ini terdapat satu atau lebih komputer dan satu *server* untuk mengendalikan beberapa komputer *client*. (Macdons Madiun : 2015 : 4)

a. Arsitektur Client Server

Arsitektur *client server* adalah suatu cara untuk meningkatkan kinerja konfigurasi file *server* yang menurun karena faktor skalabilitas (penambahan *workstation* dalam jumlah yang signifikan). Pada arsitektur ini dua aplikasi yang terpisah beroperasi secara mandiri dan bekerja sama untuk menyelesaikan suatu pekerjaan. Suatu aplikasi yang cocok dengan arsitektur ini adalah DBMS (*Database Management System*) berbasis SQL (*Structured Query Language*).



Gambar II.1 Arsitektur Client Server
Sumber : (Macdons Madiun : 2015 : 4)

Konfigurasi *client server* mencakup dua entitas yaitu : *client* dan *server*. *Client* meminta sesuatu pada *server* kemudian *server* melakukan sesuatu pekerjaan yang diminta oleh *client*. Permintaan dapat berupa query SQL yang dikirim pada mesin basis data SQL. Kemudian mesin basis data SQL memproses *query* dan hasilnya (berupa *resultset*) dikembalikan pada *client* yang meminta. Sistem *client server* memungkinkan satu aplikasi tunggal dibagi-bagi antar *workstation* maupun *server*. Aplikasi yang berbasis *client server* didefinisikan sebagai berikut:

- a. Komputasi *client server* melibatkan pembagian suatu aplikasi ke dalam beberapa *task* dan meletakkan setiap *task* pada *platform* yang dapat menangani secara efisien.
- b. Pemrosesan data bisa berada pada *client* atau di-*split* diantara *client* dan *server*.
- c. *Server* dihubungkan ke *client* melalui suatu jaringan. (Macdons Madiun : 2015 : 5)

3. Topologi jaringan

Topologi jaringan merupakan gambaran pola hubungan antara komponen-komponen jaringan, yang meliputi computer server, computer Client/ Workstation, Hub/ Switch, pengkabelan, dan komponen jaringan yang lain. Terdapat beberapa topologi jaringan yang dapat digunakan dengan kondisi di lapangan. Topologi jaringan dapat juga diartikan sebagai skema fisik jaringan yang saling terhubung satu sama lain. (Macdons Madiun : 2015 : 5)

1. Topologi Bus

Pada topologi Bus ini terdapat satu jalur umum yang berbentuk garis lurus. Ciri utama dari topologi ini adalah bahwa setiap sambungan saling bergantung, artinya apabila salah satu sambungan mengalami gangguan atau terputus, maka seluruh sambungan akan terputus dan susah untuk terhubung kembali. Meskipun topologi ini membentuk jaringan yang saling terhubung namun tidak membentuk jalur tertutup. Pada topologi bus menyediakan 1 jalur yang digunakan untuk komunikasi antar perangkat sehingga setiap perangkat harus bergantian dalam menggunakan jalur yang ada. Dalam berkomunikasi antar perangkat hanya ada 2 perangkat yang dapat saling berkomunikasi. (Macdons Madiun : 2015 : 5)

2. Topologi Token Ring

Topologi token ring memiliki ciri khusus yaitu berbentuk lingkaran seperti cincin. Semua workstation dan server dihubungkan menjadi satu membentuk pola lingkaran, setiap workstation maupun server akan menerima data atau informasi yang disharing, jika alamat yang dimaksud sesuai dengan maka informasi atau data akan diterima, dan jika alamat yang tidak sesuai maka informasi tersebut akan dilewatkan ke computer yang lain. Kelebihan dari topologi ini adalah desain cincin dalam jaringannya yang sangat mudah dibuat dan sederhana, serta tidak membutuhkan banyak kabel konektor jaringan untuk menghubungkan. (Macdons Madiun : 2015 : 5)

3. Topologi Ring

Pada topologi ini, semua PC yang terhubung ke jaringan akan saling dikaitkan sehingga membentuk satu koneksi yang tidak terputus. Namun, sistem topologi ring ini memiliki kelemahan, yaitu apabila terdapat kabel konektor yang terputus atau mengalami gangguan pada dua arah sekaligus, maka seluruh koneksi yang mengarah ke server juga ikut berpengaruh.

(Macdons Madiun : 2015 : 6)

4. Topologi Star

Topologi ini mempunyai bentuk fisik layaknya bintang, dimana setiap node dihubungkan ke pusat. Jadi, setiap transfer data selalui melalui pusat. Media transmisinya bersifat tertutup dan setiap client, sehingga apabila salah satu client mengalami kegagalan, maka client yang lain tetap bisa berkomunikasi/ terhubung dengan computer server. (Macdons Madiun : 2015 : 6)

5. Topologi Tree

Topologi tree (Pohon) didesain berdasarkan kombinasi antara topologi bus dan topologi ring. Hal ini dapat dicapai dengan mengintegrasikan HUB Multi jaringan star bersama-sama pada satu bus. Bayangkan sebuah diagram topologi tree dengan banyak jaringan star dengan HUB pusatnya. Kemudian gambarlah garis yang hanya dihubungkan HUB bersama ke titik pada satu line. HUB ini umumnya terhubung dengan satu HUB utama yang merupakan akar untuk seluruh jaringan data diantara semua computer. (Macdons Madiun : 2015 : 6)

6. Topologi Mesh

Topologi mesh adalah topologi yang menyerupai bentuk jaringan atau jala yang digunakan pada desain jaringan LAN. Topologi mesh menggunakan salah satu dari dua pengaturan koneksi adalah menggunakan mesh penuh maupun mesh parsial. Meskipun topologi mesh ini dapat diandalkan karena kemampuan interkoneksinya yang namun juga memiliki masalah redundansi (perulangan). (Macdons Madiun : 2015 : 6)

II.6. Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal / data mentah biasa disebut juga *unnormalized form*. Masing – masing level normalisasi mempunyai aturan tersendiri.

1. *First Normal Form*

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan array.
- c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. *Second Normal Form*

Suatu tabel dikatakan dalam keadaan *Second Normal Form* (2NF) jika tabel tersebut sudah dalam keadaan *First Normal Form* (1NF) dan jika semua

atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form*

Suatu tabel dikatakan dalam keadaan *third normal form* (3NF) jika tabel tersebut sudah dalam keadaan *second normal form* (2NF) dan jika tidak terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

4. *Boyce-Codd Normal Form* (BCNF)

Tabel dikatakan dalam keadaan *boyce-codd normal form* (BCNF) jika tabel tersebut dalam keadaan *third normal form* (3NF) dan setiap determinan adalah kunci kandidat.

5. *Fourth Normal Form* (4NF)

Suatu tabel dikatakan dalam keadaan *fourth normal form* (4NF) jika tabel tersebut dalam keadaan *boyce-codd normal form* (BCNF) dan jika tidak terdapat ketergantungan nilai ganda.

6. *Fiveth Normal Form* (5NF)

Tabel dikatakan dalam keadaan *Fiveth Normal Form* (5NF) jika tabel tersebut dalam keadaan *fourth normal form* (4NF) dan jika setiap ketergantungan dalam join ada pada tabel sudah konsekuen dengan kunci kandidat pada tabel tersebut (Ema Utami ; 2012 : 73-76)

II.7. Visual Studio 2010

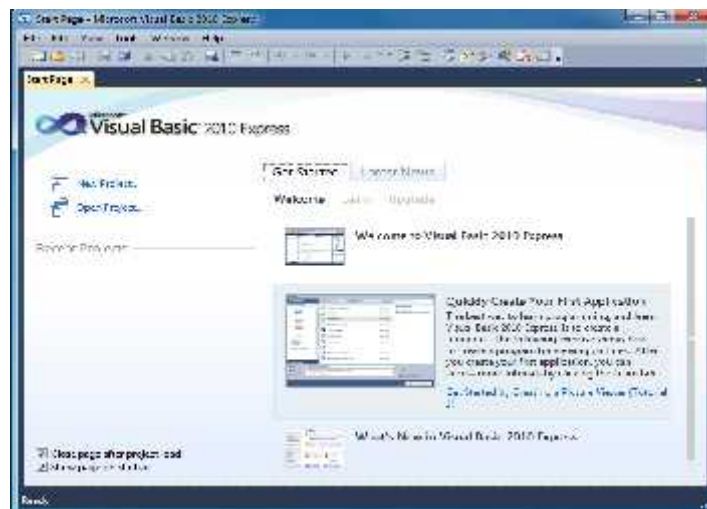
Visual Studio 2010 adalah inkarnasi dari bahasa visual basic yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat lainnya seperti C++. Anda dapat menggunakan visual basic 2010 untuk membuat aplikasi windows, mobile, web, dan office atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke program lainnya. Visual basic menyediakan berbagai tools dan fitur canggih yang memungkinkan dapat menulis kode, menguji dan menjalankan program tunggal atau terkadang serangkaian program yang terkait dengan satu aplikasi. (Christopher Lee:2014:1)

Tabel II.1. Jendela Aplikasi Visual Studio 2010

No	Bagian	Keterangan
1	Title Bar	Menampilkan nama aplikasi yang sedang terbuka.
2	Mneu Bar	Menampilkan daftar perintah yang memungkinkan anda dapat menulis, mengedit, menyimpan, mencetak, menguji dan menjalankan program visual basic.
3	Standard Toollbar	Berisi Tombol yang menjalankan perintah yang sering digunakan seperti open project, new project, save, cut, copy, paste, dan undo
4	Toolbox	Berisi komponen NET yang dapat anda gunakan untuk mengembangkan antarmuka pengguna grafis untuk program visual studio 2010.
5	Area Kerja Utama	Menampilkan item yang sedang di kerjakan
6	Solution Explorer	Menampilkan elemen dari visual basic solution, yaitu nama yang ddiberikan kepada program visual basic dan item lainnya yang dihasilkan oleh visual basic 2010 sehingga program akan mengeksekusi dengan benar.
7	Properties Window	Setiap Objek dalm program visual basic memiliki seperangkat karakteristik yang disebut sifat-sifat objek.

(Sumber : Christopher Lee ; 2014 : 2)

Untuk melihat tampilan visual basic 2010 dapat dilihat pada gambar II.2. sebagai berikut :



Gambar II.2. Tampilan Utama Visual Basic 2010
(Sumber : Christopher Lee ; 2014 ; 2)

II.8. MYSQL

MySQL adalah sebuah *system* manajemen *database* relasi (*relational database management system*) yang bersifat “terbuka” (*open source*). *MySQL* termasuk *RDMS* (*Relational Database Management System*). Pada *MySQL* sebuah *database* terdiri atas tabel-tabel. Sebuah tabel terdiri atas baris dan kolom.

MySQL dapat didefinisikan sebagai :

- a) *MySQL* merupakan *system* manajemen *database*. *Database* merupakan struktur penyimpanan data. Untuk menambah, mengakses, dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan *system* manajemen *database* seperti *MySQL Server*.
- b) *MySQL* merupakan *system* manajemen *database* atau basis data terhubung (*relational database management system*). *Database* terhubung menyimpan data pada tabel-tabel terpisah. Hal tersebut akan menambah kecepatan dan fleksibilitasnya. Kata *SQL* pada *MySQL* merupakan singkatan dari

“Structured Query Language”. SQL merupakan bahasa standar yang digunakan untuk mengakses *database* dan ditetapkan oleh ANSI/ISO SQL Standard.

- c) *MySQL* merupakan *Software Open Source*. *Open Source* berarti semua orang diizinkan menggunakan menggunakan dan memodifikasi software. Semua orang dapat mendownload software *MySQL* dari internet dan menggunakannya tanpa membayar. Anda dapat mempelajari *Source Code* dan akan menggunakannya sesuai kebutuhan .
- d) *Server database MySQL* mempunyai kecepatan akses tinggi, mudah digunakan, dan handal. *MySQL* dikembangkan untuk menangani *database* yang besar secara cepat dan telah sukses digunakan selama bertahun-tahun. Konektifitas, kecepatan, dan keamanannya memuat *server MySQL* cocok untuk mengakses *database* di internet.
- e) *MySQL Server* bekerja di *client / server* atau *system embedded*. *Software database MySQL* merupakan sistem *client / server* yang terdiri atas *multithread SQL server* yang mendukung *software client* dan *library* yang berbeda, *tool administrative*, dan sejumlah *Application Programming Interface (APIs)*.
- f) *MySQL* tersedia dalam beberapa macam bahasa. (Doro Edi : 2012).

II.9. UML (Unified Modelling Language)

Unified Modelling Language (UML) menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku

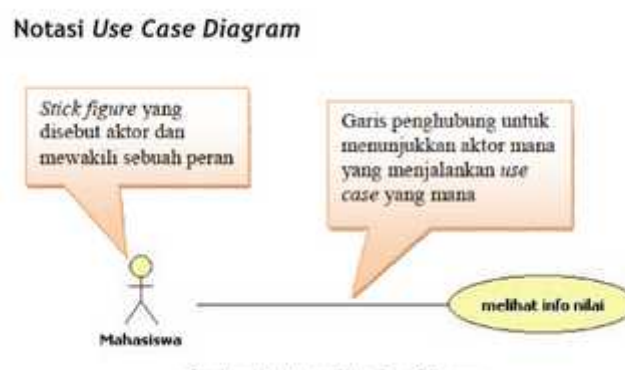
dalam proses analisis dan desain. Artifak dalam UML adalah informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasilkan oleh pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan desain adalah bahwa setiap perubahan terjadi pada suatu artifak yang harus juga dilakukan pada artifak lainnya. (Evi Triandini Dan Igede Suardika, 2012)

II.9.1. Diagram-Diagram UML

1. Diagram Use Case (Use Case Diagram)

Pengertian *use case* menurut Jhon Satzinger (2010) dalam buku “*system analysis and design in a changing world*” menyatakan bahwa “*use case* adalah sebuah kegiatan yang dilakukan oleh sstem, biasanya dalam menanggapi permintaan dari pengguna system. (Evi Triandini Dan Igede Suardika, 2012)

Notasi *Use Case* Diagram dapat dilihat pada gambar berikut ini:



Gambar II.3. Notasi Use Case Diagram

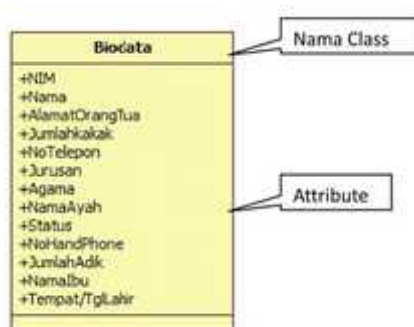
(sumber : Evi Triandini dan I Gede Suardika)

2. Diagram Kelas (*Class Diagram*)

Menurut John Satzinger, dalam buku “*system analysis and design in a changing world*” menyatakan bahwa “dalam UML ada dua jenis *clas diagram* yaitu *domain class diagram* dan *design class diagram*”

1. *Domain class diagram*

Focus domain class diagram adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada *class* perangkat lunak yang nantinya akan dirancang. Berikut gambar dari domain *class diagram*:

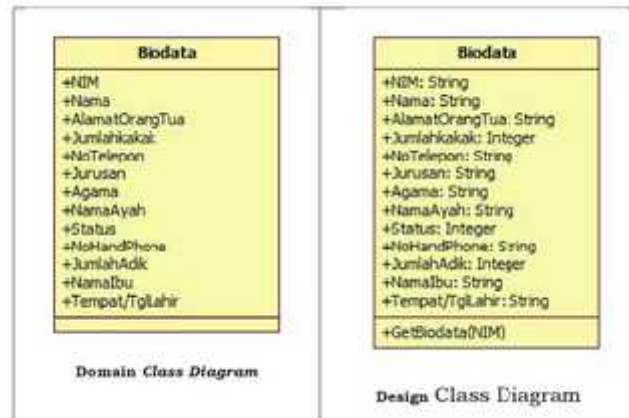


Gambar II.4. Domain Class Diagram

(Sumber : Evi Triandini Dan I Gede Suardika)

2. *Design Class Diagram*

Tujuan utamanya adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun. *Design class diagram* menggambarkan kelas berorientasi objek yang dibutuhkan dalam pemrograman, navigasi diantara kelas, atribut names dan propertinya serta *method names* dan propertinya. Gambar dari *calss diagram* dapat dilihat pada gambar di bawah ini:

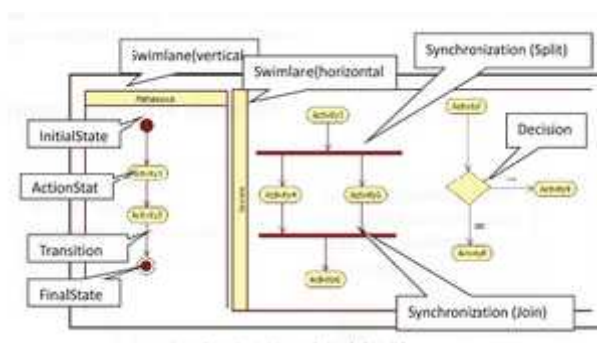


Gambar II.5. Class Diagram

(Sumber : Evi Triandini Dan I Gede Suardika)

3. Diagram Aktivitas (*Activity Diagram*)

Menurut John Satzinger, dalam buku “*system analysis and design in a changing world*” menyatakan bahwa *activity diagram* adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau pun sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut. Notasi *activity diagram* adalah sebagai berikut:



Gambar II.6. Notasi Activity Diagram

(sumber : Evi Triandini dan I Gede Suardika)

Penjelasan dari gambar diatas adalah sebagai berikut:

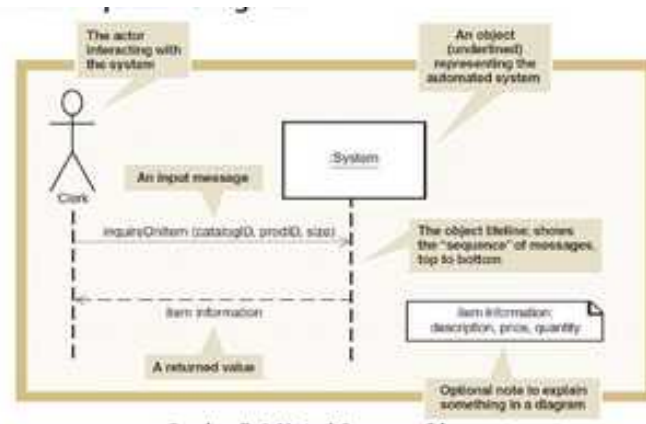
Tabel II.2. Keterangan Notasi *Activity Diagram*

Notasi	Keterangan
<i>Swimlane</i>	Mewakili agen yang melakukan aktivitas karena dalam alur kerja umumnya mempunyai agen yang berbeda dengan yang melakukan langkah yang berbeda dari proses alur kerja.
<i>Initialstate</i>	Awal dari alur kerja
<i>ActionState</i>	Melambangkan aktivitas tersendiri dalam alur kerja
<i>Transition</i>	Melambangkan dari urutan diantara aktivitas
<i>Final State</i>	Akhir dari alur kerja
<i>Synchronization</i>	Membagi alur kerja menjadi beberapa alur yang berbarengan ataupun menggabungkan lagi alur yang berbarengan
<i>Decision</i>	Titik pengambilan keputusan dimana aliran proses tersebut akan mengikuti satu jalur atau jalur lainnya

(Sumber : Evi Triandini dan I Gede Suardika)

4. *Sequence Diagram*

Menurut John Satzinger, dalam “*buku system analysis and design in a changing world*” menyatakan bahwa” *System Sequence Diagram* (SSD) adalah diagram *output* serta urutan interaksi antara pengguna dan *system* untuk sebuah *use case*. Berikut ini adalah gambar dari *sequence diagram* :



Gambar II.7. Sequence Diagram

(Sumber : Evi Triandini dan I Gede Suardika)

Penjelasan dari gambar diatas adalah sebagai berikut:

- Actor*, mewakili seorang aktor (orang atau peran yang berinteraksi dengan sistem)
- Kontak berlabel, sistem adalah objek yang mewakili keseluruhan sistem yang terotomatisasi
- Garis putus-putus vertikal (*lifelines*) adalah perpanjangan objek tersebut, baik aktor maupun objek, sepanjang durasi dari *sequence diagram*.
- Anak panah antara *lifeline* mewakili *message* yang dikirim atau diterima oleh aktor dari sistem.
- Message* diberi label untuk menggambarkan maksud *message* dan *input* apa pun yang sedang dikirim. *Message* dipertimbangkan sebagai sebuah inti yang diminta pada tujuan objek, kebanyakan seperti perintah.