BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Tata Sutabri, 2012).

Norman L. Enger menyatakan bahwa suatu sistem dapat terdiri atas kegiatan-kegiatan yang berhubungan berguna mencapai tujuan-tujuan perusahaan seperti pengendalian inventaris atau penjadwalan produksi (Tata Sutabri, 2012).

II.1.2. Karakteristik Sistem

Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, adapun karakteristik yang dimaksud adalah sebagai berikut : (Tata Sutabri, 2012)

1. Komponen Sistem *(Components)*

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan Supra sistem.

2. Batasan Sistem *(Boundary)*

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem *(Environments)*

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan kalau tidak akan mengganggu kelangsungan hidup sistem tersebut .

4. Penghubung Sistem *(Interface)*

   Media yang menghubungkan sistem dengan subsitem yang lain disebut dengan penghubung sistem atau *Interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem *(Input)*

   Energi yang dimasukan kedalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan *(maintenance input)* dan sinyal *(signal input)*. Sebagai contoh, didalam suatu unit komputer "program" adalah *maintenance input* yang digunakan untuk mengoperasikan *computer*. Sementara "data" adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran Sistem *(Output)*

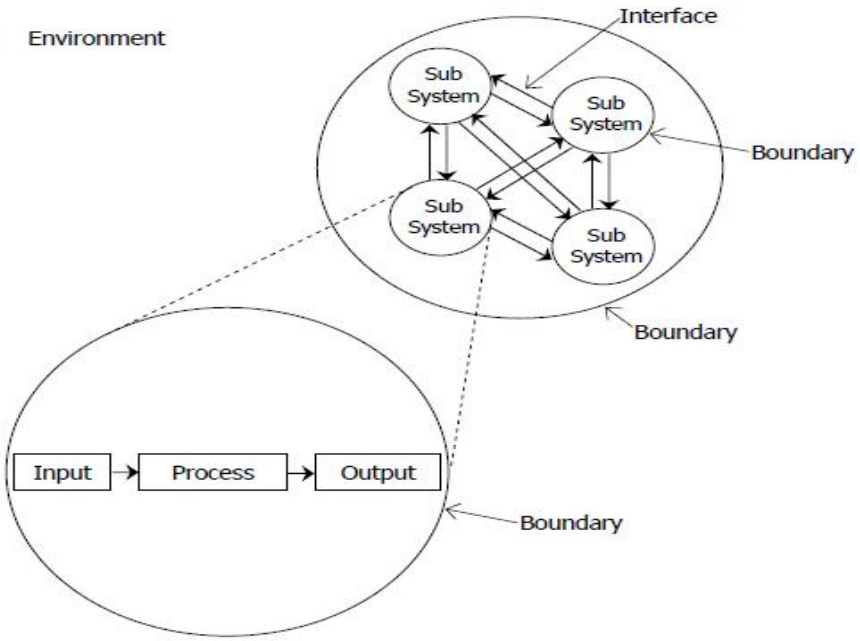
   Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang akan dihasilkan adalah informasi, dimana informasi ini dapat digunakan sebagai masukan untuk mengambil keputusan atau hal-hal lain yang merupakan *input* bagi subsistem lainnya.

7. Pengolahan Sistem *(Process)*

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh sistem akuntansi, sistem ini mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem *(Objctive)*

   Suatu sistem pasti mempunyai tujuan atau sasaran yang pasti bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan yang telah direncanakan.



Gambar II.1. Karakteristik Sistem

(Sumber : Tata Sutabri, 2012)

II.2. Pengertian Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterprestasikan untuk digunakan dalam proses pengambilan keputusan (Tata Sutabri, 2012).

Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian - kejadian dari kesatuan nyata. Informasi dihasilkan lebih berharga maka informasi harus memenuhi kriteria sebagai berikut :

1. Informasi harus tepat waktu, sehingga tidak ada keterlambatan ketika dibutuhkan.
2. Informasi harus relevan, benar - benar terasaa manfaatnya bagi yang membutuhkan.

3. Informasi harus bernilai, informasi yang berharga untuk suatu pengambilan keputusan.
4. Informasi harus dipercaya, sehingga mendukung pihak manajemen dalam mengambil keputusan.

Kegunaan informasi adalah untuk mengurangi ketidak pastian didalam proses pengambilan keputusan tentang suatu keadaan. Nilai dari suatu informasi ditentukan dari dua hal yaitu manfaat dan biaya untuk mendapatkannya (Tata Sutabri, 2012).

II.3. Pengertian Sistem Informasi

Sistem informasi menjelaskan bahwa suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan - laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri, 2012).

Menurut definisi tersebut penulis dapat menyimpulkan bahwa sistem informasi itu adalah kumpulan dari komponen-komponen yang saling bekerjasama secara harmonis untuk bertujuan menyajikan informasi yang bermanfaat. Sistem informasi dapat didefenisikan sebagai berikut :

1. Suatu sitem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan menyajikan informasi.

2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambilan keputusan dan untuk mengendalikan organisasi.
3. Suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi dan menyediakan pihak luar tertentu dengan laporan-laporan yang di perlukan.

II.4. Komponen Sebuah Sistem

Sistem informasi terdiri dari beberapa komponen, yaitu :

1. Masukan *(input)*

   Komponen yang mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media untuk menagkap data yang akan di masukkan yang dapat berupa dokumen - dokumen dasar.

2. Model

   Komponen ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang akan tersimpan di baris data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang di inginkan.

3. Keluaran *(output)*

   Yaitu produk dari suatu sistem informasi yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem.

4. Teknologi

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Basis Data

Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lain, tersimpan dalam perangkat keras komputer dan digunakan untuk memanipulasinya. Data di dalam basis data perlu di organisasikan sedemikian rupa agar dihasilkan informasi yang berkualitas. Organisasi basis data yang baik juga digunakan untuk efisiensi kapasitas penyimpanan.

6. Kendali

Banyak hal yang dapat merusak informasi, seperti misalnya bencana alam, kecurangan-kecurangan, kegagalan sistem itu sendiri dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat dicegah, ataupun bila terlanjur terjadi langsung cepat diatasi.

II.5. *Microsoft Visual Studio* 2010 *(Visual Basic)*

Microsoft Visual basic merupakan salah satu aplikasi pemograman visual yang memiliki bahasa pemograman yang cukup populer dan mudah untuk dipelajari (Suprianto, Dodit, 2010). Dengan *microsoft visual basic* kita bisa membuat program dengan aplikasi GUI *(Graphical User Interface)* atau program

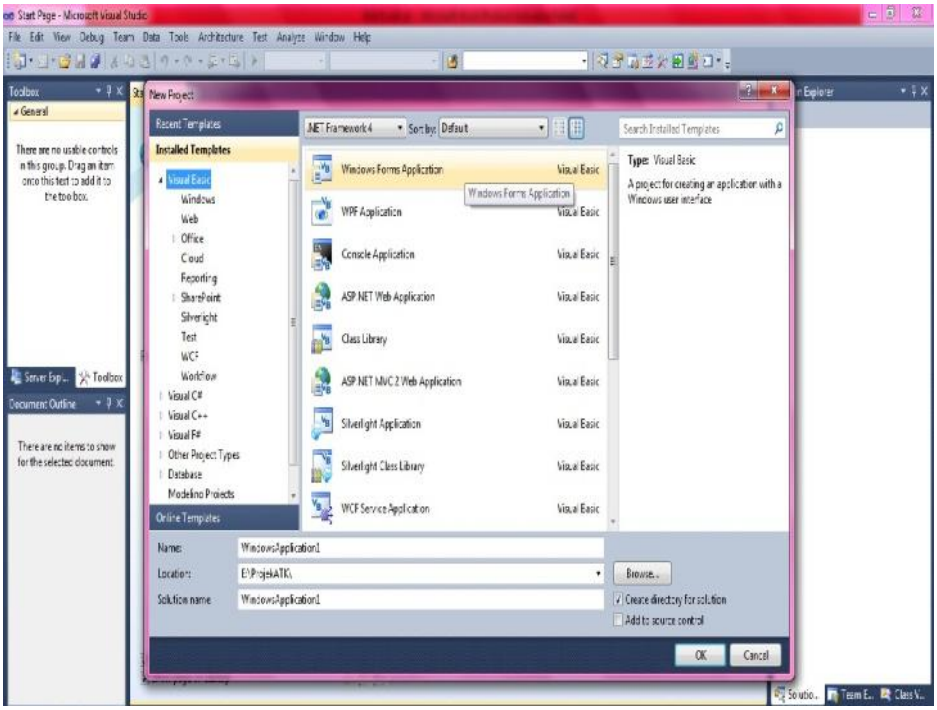
yang memungkinkan pengguna komputer berkomunikasi dengan komputer tersebut menggunakan grafik atau gambar.

Microsoft visual basic menyediakan berbagai perangkat control yang dapat digunakan untuk membuat program aplikasi dalam sebuah *form* baik aplikasi kecil, sederhana hingga ke aplikasi pengolahan *database*.

II.5.1. Memulai *Microsoft Visual Studio* 2010

Untuk memulai pemrograman dengan *micorosoft visual basic* kita harus terlebih dahulu menjalankan program *microsoft visual studio* 2010. Selanjutnya pada tampilan awal akan ditampilkan kotak dialog *new project* seperti gambar berikut ini. Pada kotak dialog tersebut terdapat tab yang terdiri dari :

1. *New* (menampilkan daftar pilihan untuk membuat project baru).



Gambar II.2. Kotak Dialog *New Project*
(Sumber : Handphan Rianto : 2012)

Untuk pembuatan program pertama kali pilih *new project*, pilih *Windows Forms Application* lalu ketik nama, klik OK.

II.5.2. Komponen-komponen Visual Studio 2010

1. Menu *Bar*

Menampilkan daftar menu yang berisi daftar perintah-perintah yang dapat digunakan saat bekerja pada visual basic. Terdiri dari menu *file, edit, view, project, build, debug, team, data, tools,architecture, test, analyze, window dan help.*

Gambar II.3. Menu *Bar*
(Sumber : Handphan Rianto: 2012)

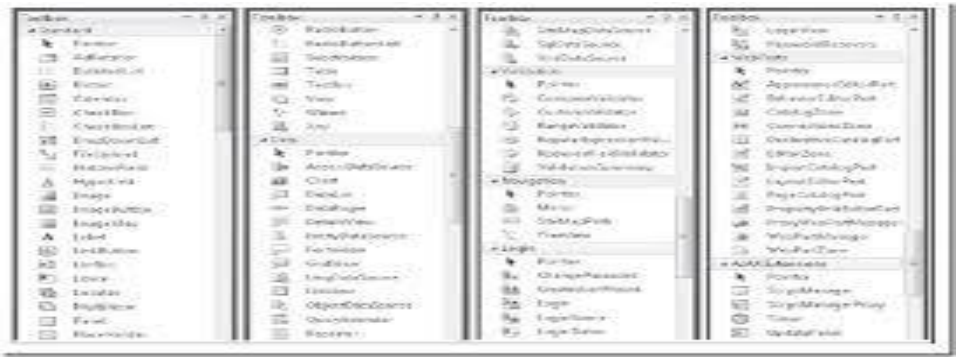
2. *Toolbar*

Digunakan untuk mengakses perintah-perintah dalam menu yang sering dipakai secara cepat.

Gambar II.4. Menu *Toolbar*
(Sumber : Handphan Rianto: 2012)

3. *Toolbox*

Merupakan daftar komponen - komponen yang dapat digunakan untuk mendesain tampilan program aplikasi yang akan dibuat.



Gambar II.5. *Toolbox*
(Sumber : Hadphan Rianto: 2012)

4. *Solution Explorer*

Menampilkan daftar *form* dan modul yang ada dalam project yang sedang aktif.

Gambar II.6. Tampilan *Solution Explorer*
(Sumber : Handphan Rianto: 2012)

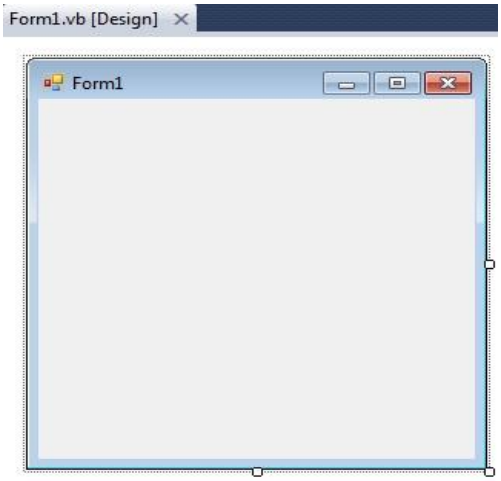
5. *Property Window*

Digunakan untuk mengatur property dari komponen-komponen yang sedang diaktifkan. *Property* merupakan karakteristik dari sebuah objek. Berikut ini adalah tampilan dari *property form.*

Gambar II.7. Tampilan *Property Window*
(Sumber : Handphan Rianto: 2012)

6. *Form Designer*

Merupakan jendela yang digunakan untuk melakukan perancangan tampilan dari aplikasi yang akan dibuat.



Gambar II.8. Tampilan Awal *Form*
(Sumber : handphan Rianto: 2012)

7. *Code Window*

Merupakan jendela yang digunakan untuk menuliskan kode program.

Gambar II.9. Jendela *Code Window*
(Sumber : Handphan Rianto: 2012)

Didalam jendela kode *project* terdapat :

- *Object Listbox* : untuk menampilkan nama-nama *control* yangterdapat pada *form* ditambah secara *form* dan *general.*

- *Procedure Listbox* : untuk menampilkan *event procedure* dari setiap *control* yang dipilih. Misalnya *Click, Double Click, Mouse Move*, dan lain - lain.

Selain dari komponen layar utama diatas, terdapat juga komponen program *visual basic* yang biasa digunakan yaitu:

1. *Event*

Event adalah segala sesuatu yang dialami oleh sebuah objek yang diakibatkan baik oleh tindakan user atau dari tindakan program itu sendiri. Misalnya klik, tunjuk, dan lain sebagainya. Seluruh program aplikasi berbasis *Windows* saat ini desebut dengan istilah *event-drive* program. Lingkungan *windows* yang *multitasking* memberikan berbagai pilihan yang dapat dilakukan oleh pemakai terhadap suatu objek yang terdapat pada program aplikasi. Dengan demikian suatu program harus memberikan beberapa pilihan yang bisa dilakukan oleh para *user*nya (pemakainya).

Didalam *visual basic*, kode program tidak dituliskan sekaligus dalam satu tempat, melainkan disebar menjadi prosedur atau rutin-rutin yang lebih kecil yang terdapat didalam objek-objek. Prosedur inilah yang akan dijalankan apabila *user* melakukan suatu event tertentu pada sebuah objek. Setiap objek dapat memiliki satu atau lebih event. Contoh :

```
Private Sub Form1_Load()
Form1.text=”Persediaan”
End Sub
```

Keterangan:

Program akan mengubah text dari *form* menjadi Persediaan ketika *form* pertama kali ditampilkan (*Form_Load*).

2. *Variable*

Variable adalah tempat untuk menyimpan nilai-nilai atau data-data secara sementara pada aplikasi *microsoft visual basic* 2010.

Dim<Nama Variabel> As <TipeData>

Dim Nama *As String* * 20

Keterangan :

Dim : *Statement* dari *visual basic* untuk mendeklarasikan *visual basic*.

Nama Variabel : Nama Variabel yang digunakan untuk menyimpan nilai.

Tipe Data : Tipe data yang akan disimpan didalam variabel tersebut.

3. Konstanta

Konstanta adalah variabel yang nilai didalamnya bersifat tetap. *Konstanta* diperlukan apabila memerlukan sebuah nilai tetap yang sering dipakai dibanyak bagian dalam program. Berikut adalah contoh pendeklarasian *konstanta* :

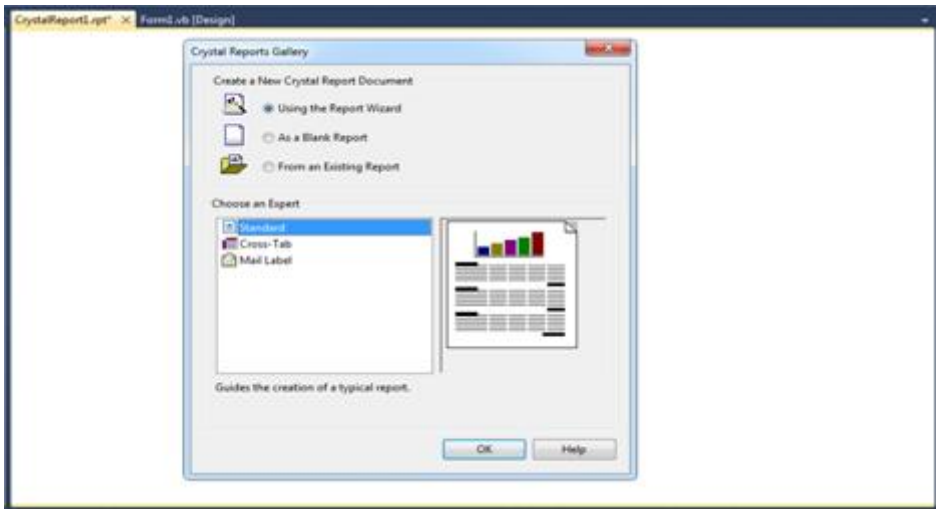
```
Const ConNama =”Budi”
Private Const Jumlah=100
Public Const Angka=5
```

4. Metode (*Methods*)

Metode adalah suatu set perintah seperti halnya fungsi dan prosedur, tetapi sudah tersedia didalam suatu objek. Metode biasanya akan mengerjakan suatu tugas khusus kepada suatu objek.

II.5.3. *Crystal Report*

Crystal Report adalah suatu aplikasi *windows* yang dikembangkan oleh *Seaget Software* yang berguna untuk membuat format laporan yang terpisah dari program *Microsoft Visual Studio* 2010, tetapi keduanya dapat dihubungkan. Mencetak laporan dengan *Crystal Report* hasilnya akan lebih baik dan lebih mudah untuk dikerjakan, karena pada *Crystal Report* banyak tersedia objek yang terdapat pada *Crystal Report* (Ardian M., 2013).



Gambar II.10. Tampilan *Crystal Report*
(Sumber : Nursal: 2011)

Ada beberapa komponen yang terdapat pada *Crystal Report*, yaitu :

1. *Report Header*, merupakan tempat dimana biasanya judul laporan diletakkan atau informasi lain yang ingin kita munculkan diawal halaman.
2. *Page Header*, merupakan tempat dimana apabila kita ingin memunculkan informasi diatas setiap halaman, contohnya nama bab, nama dokumen atau informasi lainnya yang serupa.

3. *Details*, merupakan inti dari laporan, yang akan memunculkan saetiap *record* dari *database*.
4. *Report Footer*, merupakan tempat apabila kita ingin memunculkan informasi hanya sekali disetiap akhir laporan, contoh *Grand* Total.
5. *Page Footer*, merupakan tempat dimana biasanya nomor halaman diletakkan atau informasi lain yang muncul dihalaman ini.

II.6. Perancangan Basis Data (*Database)*

II.6.1. *Database*

Database didefinisikan sebagai kumpulan informasi yang integritasi, diorganisasikan dan disimpan dalam suatu cara yang memudahkan pengambilan kembali (Budi Sutedjo Dharma Oetomo, 2011).

Tujuan dari desain *database* adalah untuk menentukan data-data yang dibutuhkan dalam sistem sehingga informasi yang dihasilkan dapat terpenuhi dengan baik. *Desain database* perlu dilakukan untuk menghindari pengulangan data.

II.6.2. *MYSQL*

MySQL merupakan salah satu database server yang berkembang di lingkungan open source dan didistribusikan secara *free* (gratis) dibawah lisensi GPL(http://dir.unikom.ac.id/s1-final-project/fakultas-teknik-dan ilmukomputer/manajemen-informatika/2011/jbptunikompp-gdl-indrawardh-24507/3 unikom-i-i.pdf/ori/3-unikom-i-i.pdf).

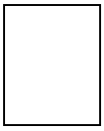
MySQL merupakan *RDBMS* (*Relational Database Management System*) server. *RDBMS* adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model *relational*. Dengan demikian, tabel-tabel yang ada pada *database* memiliki relasi antara satu tabel dengan tabel lainnya.

II.7. Teori Diagram Yang Digunakan

II.7.1. *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* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/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 *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Tabel II.1. *Use Case Diagram*

Gambar	Nama	Keterangan
	*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
	*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber : Adi Nugroho: 2011)

II.7.2. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas 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, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Tabel II.2. ***Activity Diagram***

Gambar	Nama	Keterangan
	*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	*Initial Node*	Bagaimana objek dibentuk atau diawali.
	*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
	*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Adi Nugoroho, 2011)

II.7.3. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Tabel II.3. *Sequence Diagram*

Gambar	Nama	Keterangan
	*Life Line*	Objek *entity*, antarmuka yang saling berinteraksi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

(Sumber : Adi Nugroho: 2011)

II.7.4. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. (http://informatika.web.id/normalisasi.htm).

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 (www.utexas.edu). Normalisasi terbagi atas 4 bagian, yaitu :

1. Bentuk Tidak Normal (*Unnormalized Form*)

   Bentuk ini merupakan bentuk :

a. Data yang akan direkam, tidak akan keharusan mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau duplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.
b. Adanya tabel/data mentah.

2. Bentuk Normal Kesatu (*First Normal Form* / 1NF)

   Bentuk normal kesatuan mempunyai ciri yaitu :

   a. Setiap data dibentuk dalam *flat file* (file datar).
   b. Data dibentuk dalam satu *record* demi satu *record*.

c. Nilai dari *field* berupa "*otomic value*" tidak ada *set attribute* berulang - ulang dan *attribute* bernilai ganda (*multi value*).
d. Tiap *field* hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja bukanlah pecahan kata-kata sehingga artinya lain.

3. Bentuk Normal Kedua (*Second Normal Form* / 2NF)

   Bentuk normal kedua mempunyai syarat sebagai berikut :

   a. Bentuk data telah memenuhi kriteria bentuk normal kesatu.
   b. Atribut bukan kunci, haruslah secara fungsi pada kunci utama/*primy key*, sehingga untuk membentuk normal kedua haruslah ditentukan kunci - kunci *field*.
   c. Kunci *field* haruslah unik dan dapat mewakili atribut lain menjadi anggota.

4. Bentuk Normal Ketiga (*Third Normal Form*/3NF)

   Untuk menjadi bentuk normal ketiga maka :

a. Relasi haruslah dalam bentuk kedua dan semua atribut buka primer tidak punya hubungan transitif.
b. Setiap atribut bukan kunci, haruslah bergantung pada *primary key* secara menyeluruh.

Proses untuk mengorganisasikan *file* untuk menghilangkan arus elemen yang berulang - ulang disebut normalisasi. Normalisasi banyak juga dilakukan dalam merubah struktur hubungan. Dalam hal ini dapat didefenisikan struktur data yang baru yaitu disebut dengan struktur data hubungan (*relational data structure*). Istilah data hubungan menunjukkan suatu struktur data yang mempunyai hubungan dengan elemen - elemen data lainnya, baik satu *file* atau *file* lainnya.

Kamus Data

Menurut Krismiaji kamus data adalah salah satu komponen kunci dalam sistem manajemen *database* adalah file khusus yang disebut kamus data (*data dictionary*)
(http://www.masterdiagram.tk/2014/08/pengertian-dfl-data-flowdiagramkamus.html).

Kamus data berisi tentang struktur *database*. Untuk setiap elemen *database*, seperti nomor pelanggan, diuraikan secara lengkap mulai dari nama, tempat penyimpanan, program computer yang berhubungan dan lain-lain.

Dengan menggunakan kamus data analisis sistem dapat mendefenisikan data yang mengalir disistem dengan lengkap. Kamus data ikut berperan dalam perencangan dan pengembangan sistem karena berfungsi untuk :

1. Menjelaskan arti aliran data dan menyimpan dalam penggambaran data *flow diagram*.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran.
3. Menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.