

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional (dengan tugas/fungsi khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses/pekerjaan tertentu. Sebagai contoh, sistem kendaraan terdiri dari : komponen stater, komponen pengapian, komponen penggerak, komponen pengerem, komponen kelistrikan – *speedometer*, lampu dan lain-lain. Komponen-komponen tersebut diatas memiliki tujuan yang sama, yaitu untuk membuat kendaraan tersebut bisa dikendarai dengan nyaman dan aman (Kusrini ; 2007 : 11).

II.2. Pengertian Informasi

Informasi menjadi penting, karena berdasarkan informasi itu para pengelola dapat mengetahui kondisi obyektif perusahaannya. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan cara tertentu. Informasi disajikan dalam bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan untuk menambah wawasan bagi pemakainya guna mencapai suatu tujuan.

Sebagai contoh, data dapat berupa nama karyawan, jumlah jam kerja, dan lain sebagainya. Jika banyaknya jam kerja dikalikan dengan besarnya upah per jam, maka akan diperoleh gaji kotor. Jika gaji kotor tersebut dijumlahkan, maka

akan diperoleh total gaji kotor. Setelah pemrosesan dilakukan terhadap data, maka akan diperoleh informasi yang dapat mengungkapkan tentang gaji kotor per karyawan dan total biaya gaji yang harus disediakan oleh perusahaan.

Informasi tersebut diperlukan sebagai dasar pertimbangan para pengelola organisasi dalam mengambil keputusan manajerial dan strategis.

Pengolahan data menjadi informasi itu merupakan suatu siklus, yang terdiri dari tahap-tahap sebagai berikut :

1. Pengumpulan data. Pada tahap ini dilakukan suatu proses pengumpulan data yang asli dengan cara tertentu, seperti sampling, data transaksi, data *warehouse*, dan lain sebagainya yang biasanya merupakan proses pencatatan data ke dalam suatu *file*.
2. *Input*. Tahap ini merupakan proses pemasukan data dan prosedur pengolahan data ke dalam komputer melalui alat *input* seperti *keyboard*. Prosedur pengolahan data itu merupakan urutan langkah untuk mengolah data yang ditulis dalam suatu bahasa pemrograman yang disebut program.
3. *Pengolahan data*. Tahap ini merupakan tahap di mana data diolah sesuai dengan prosedur yang telah dimasukkan. Kegiatan pengolahan data ini meliputi pengumpulan data, klasifikasi (pengelompokkan), kalkulasi, pengurutan, penggabungan, peringkasan baik dalam bentuk table maupun grafik, penyimpanan dan pembacaan data dari tempat penyimpanan data (Budi Sutedjo Dharma Oetomo ; 2006 : 12).

II.3. Pengertian Sistem Informasi

Sistem Informasi dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi. Dengan kata lain, SI merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya perusahaan.

Sistem Informasi juga mampu mendukung para pengelola dan staf perusahaan untuk menganalisa permasalahan, memvisualisasikan ikhtisar analisa melalui grafik-grafik dan tabel-tabel, serta memungkinkan terciptanya produk serta layanan yang baru. Sistem Informasi yang baik tentu memiliki sistematika yang jelas, ringkas, dan sederhana. Mulai dari tahap pemasukan data, pengolahan dengan prosedur yang ditentukan, penyajian informasi yang akurat, interpretasi yang tepat dan distribusinya (Budi Sutedjo Dharma Oetomo ; 2006 : 11).

II.4. Akuntansi

Menurut Abubakar (2007:1) dalam bukunya *Akuntansi Untuk Bisnis Usaha Kecil dan Menengah*, Akuntansi adalah merupakan proses identifikasi, pencatatan dan komunikasi terhadap transaksi ekonomi dari suatu entitas (perusahaan). Jadi secara umum terdapat tiga aktivitas dalam akuntansi yaitu :

1. Aktivitas identifikasi (*Identifying*), dalam aktivitas ini akan dilakukan identifikasi terhadap transaksi yang terjadi dalam suatu entitas (perusahaan),

2. Aktivitas pencatatan (*recording*), dalam aktivitas ini semua transaksi ekonomi atau transaksi keuangan yang telah diidentifikasi pada tahap pertama akan dicatat secara kronologis dan sistematis dengan ukuran nilai moneter tertentu, dan
3. Aktivitas komunikasi (*external user*), misalnya kreditur, investor dan fiskus.

II.5. Sistem Informasi Akuntansi

Menurut Hall (2007:10) dalam bukunya *Sistem Informasi Akuntansi*, Sistem Informasi Akuntansi merupakan sistem yang memroses berbagai transaksi keuangan dan transaksi nonkeuangan yang secara langsung memengaruhi pemrosesan transaksi keuangan.

Sistem Informasi Akuntansi mempunyai tiga subsistem, yaitu:

- a. Sistem pemrosesan transaksi yang mendukung operasi bisnis harian melalui berbagai dokumen serta pesan untuk para pengguna di seluruh perusahaan.
- b. Sistem buku besar/pelaporan keuangan yang menghasilkan laporan keuangan, seperti laporan laba rugi, neraca, arus kas, dan sebagainya.

Sistem pelaporan manajemen yang menyediakan pihak manajemen internal berbagai laporan keuangan bertujuan khusus serta informasi yang dibutuhkan untuk pengambilan keputusan, seperti anggaran, laporan kinerja serta laporan pertanggungjawaban.

II.6. Pengertian Pajak

Pajak merupakan pemindahan sumber daya dari sector *privat* (perusahaan) ke sector *public*. Pemindahan sumber daya tersebut akan memengaruhi daya beli

(*purchasing power*) atau kemampuan belanja (*spending power*) dari sektor privat. Agar tidak terjadi gangguan yang serius terhadap jalannya perusahaan, maka pemenuhan kewajiban perpajakan harus dikelola dengan baik.

Bagi Negara, pajak adalah salah satu sumber penerimaan penting yang akan digunakan untuk membiayai pengeluaran Negara, baik pengeluaran rutin maupun pengeluaran pembangunan. Sebaliknya bagi perusahaan, pajak merupakan beban yang akan mengurangi laba bersih. Keputusan bisnis sebagian besar dipengaruhi oleh pajak, baik secara langsung maupun keputusan bisnis yang kurang baik, begitu juga sebaliknya (Erly Suandi ; 2008 : 1).

II.7. Pengertian PPH

Pajak Penghasilan (PPH) pasal 21 merupakan pajak yang dikenakan atas penghasilan berupa gaji, upah, honorarium, tunjangan, dan pembayaran lain dengan nama apapun sehubungan dengan pekerjaan, jasa, atau kegiatan yang dilakukan oleh wajib pajak orang pribadi dalam negeri.

Dalam penjelasan UU No 28 pasal 1 Tahun 2007, pajak adalah kontribusi wajib kepada Negara yang terhutang oleh orang pribadi atau badan yang bersifat memaksa berdasarkan Undang-Undang, dengan tidak mendapatkan imbalan secara langsung dan digunakan untuk keperluan Negara bagi sebesar-besarnya kemakmuran rakyat (Novita Tri Astuti ; 2011 : 3).

II.8. Pengertian Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java

dikembangkan oleh *Sun Microsystem* dan dirilis tahun 1995. Java merupakan suatu teknologi perangkat lunak yang digolongkan *multi platform*. Selain itu, Java juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program.

Bahasa pemrograman java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan *versi* yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, serta Sembilan pemrograman lainnya dari *Sun Microsystem*. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang (Wahana Komputer ; 2010 : 1).

II.9. Pengertian NetBeans

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeanss IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans Platform adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi *desktop* yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi *desktop*), pemrograman *enterprise*, dan pemrograman perangkat *mobile*. Saat ini NetBeans telah mencapai versi 6.8 (Wahana Komputer ; 2010 : 15).

II.10. Pengertian Database

Database merupakan kumpulan data yang saling berhubungan, hubungan antar data dapat ditunjukkan dengan adanya *field* kunci dari setiap tabel yang beda.

Dalam satu *file* atau tabel terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field* yang saling berhubungan menunjukkan bahwa *fiel* tersebut satu pengertian yang lengkap dan disimpan dalam satu *record*. Basis data mempunyai beberapa kriteria penting yaitu :

1. Bersifat data oriented dan bukan program oriented.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat dikembangkan dengan mudah, baik *volume* maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara-cara yang berbeda.

Prinsip utama *database* adalah pengaturan data dengan tujuan utama fleksibel dan kecepatan pada saat pengambilan data kembali. Adapun ciri-ciri basis data di antaranya adalah sebagai berikut :

1. Efisiensi meliputi kecepatan, ukuran dan ketepatan.
2. Data dalam jumlah besar.
3. Berbagi pakai (dipakai bersama-sama atau *sharebility*).

Mengurangi bahkan menghilangkan terjadinya duplikasi dan data yang tidak konsisten (Windu Gata ; 2013 : 19).

II.11. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang programmer database bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna.

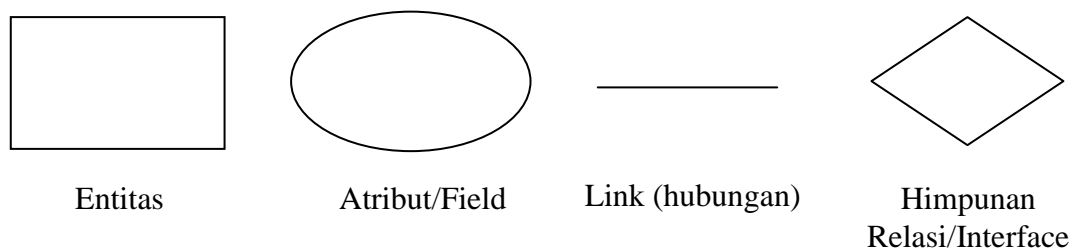
Mysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer di antara *database* yang lain.

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*), yang dapat anda download pada alamat resminya <http://www.mysql.com>. MySQL sudah cukup lama dikembangkan, beberapa *fase* penting dalam pengembangan MySQL adalah sebagai berikut :

- a. MySQL dirilis pertama kali secara internal pada 23 Mei 1995
- b. Versi *windows* dirilis pada 8 Januari 1998 untuk *windows 95* dan *windows NT*.
- c. Versi 3.23 : beta dari Juni 2000, dan dirilis pada Januari 2001.
- d. Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (*unions*)
(Wahana ; 2010 : 5).

II.12. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram atau ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas. Proses memungkinkan analisis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata ; 2010 : 67).



Gambar. II.2 Bentuk Simbol ERD
(Sumber : Janner Simarmata ; 2010 : 67)

II.13. Kamus Data

Kamus data (*data dictionary*) mencakup definisi-definisi dari data yang disimpan di dalam basis data dan dikendalikan oleh sistem manajemen basis data. Figur 6.5 menunjukkan hanya satu tabel dalam basis data jadwal. Struktur basis data yang dimuat dalam kamus data adalah kumpulan dari seluruh definisi *field*, definisi tabel, relasi tabel, dan hal-hal lainnya. Nama *field* data, jenis data (seperti teks atau angka atau tanggal), nilai-nilai yang valid untuk data, dan karakteristik-karakteristik lainnya akan disimpan dalam kamus data. Perubahan-perubahan pada struktur data hanya dilakukan satu kali di dalam kamus data, program-program aplikasi yang mempergunakan data tidak akan ikut terpengaruh (Raymond McLeod ; 2008 : 171).

II.14. Teknik Normalisasi

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan update.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

II.14.1. Bentuk-bentuk Normalisasi

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

2. Bentuk normal tahap pertama (1” Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

4. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- a. X haruslah superkey pada tabel tersebut.
- b. Atau A merupakan bagian dari primary key pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan

pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

6. Boyce Code Normal Form (BCNF)

- a. Memenuhi 1st NF
- b. Relasi harus bergantung fungsi pada atribut superkey (Kusrini ; 2007 : 39-43).

II.13. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

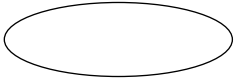
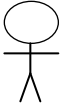


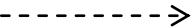
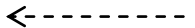
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. Use case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat

dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.2. Simbol Use Case




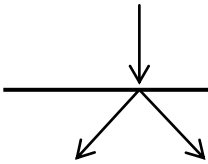
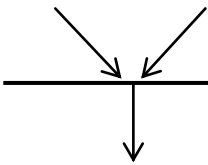
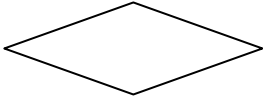

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.3. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

3. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

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*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. *Multiplicity Class Diagram*

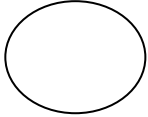
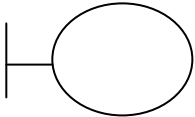
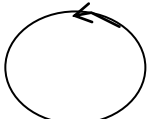

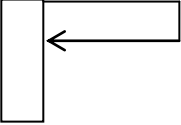
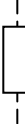

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 9)

4. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.5. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)

II.14. Daftar Pustaka.

- Arif, Abubakar, 2007. *Akuntansi Untuk Bisnis Usaha Kecil dan Menengah*. Grasindo, Jakarta.
- Gata, Windu, 2013. *Sukses Membangun Aplikasi Penjualan Dengan Java*. Elex Media, Jakarta.
- Hall, James A, 2007. *Accounting Information System*. Salemba Empat, Jakarta.
- Kusrini, 2007. *Strategi Perancangan dan Pengolaan Basis Data*. Andi, Yogyakarta.
- Komputer, Wahana, 2010. *Panduan Belajar MySQL Database Server*. TransMedia, Jakarta.
- McLeod, Raymond, 2008. *Sistem Informasi Manajemen*. Salemba Empat, Jakarta.
- Simarmata, Janner, 2010. *Basis Data*. Andi, Yogyakarta.
- Suandi, Erly, 2008. *Perencanaan Pajak*. Salemba Empat, Jakarta.
- Supardi, Yuniar, 2010. *Semua Bisa Menjadi Programmer Java Basic Programming*. Elex Media, Jakarta.
- Sutedjo, Budi, 2006. *Perencanaan dan Pembangunan Sistem Informasi*. Andi, Yogyakarta.
- Tri, Novita, 2011. *Jurnal : Aplikasi Penggajian dan Penghitungan Pajak Penghasilan Pasal 21*. Teknik Telkom Bandung.