

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Konsep Dasar Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh *Kenneth Boulding*, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem.

Selain itu, suatu sistem tidak bisa lepas dari lingkungan sekitarnya maka umpan balik atau *feedback* dapat berasal dari lingkungan sistem yang dimaksudkan. Organisasi dipandang sebagai suatu sistem yang tentunya akan memiliki semua unsur ini. (*Tata Sutabri, 2012 : 10*)

II.1.2 Konsep Dasar Informasi

Informasi adalah sebuah istilah yang tepat dalam pemakaian umum. Informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan lain sebagainya.

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. (*Tata Sutabri, 2012 : 29*)

II.1.3 Konsep Dasar Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam 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 kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sistem informasi terdiri dari komponen- komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran. (*Tata Sutabri, 2012 :46*)

II.2 .Sistem Informasi Akuntansi

Analog dengan defenisi sebelum nya, sistem informasi akuntansi (SIA) adalah sistem berbasis komputer yang dirancang untuk mentransformasi data akuntansi menjadi informasi. Namun, dalam buku ini , istilah *sistem informasi akuntansi* memiliki cakupan yang sanga luas, yaitu mencakup juga siklus pemrosesan transaksi , penggunaan sistem teknologi informasi dan pengembangan sistem informasi. (*George H.Bodnar & William S.Hopwood, 2006 : 8*)

Defenisi Sistem informasi Akuntansi menurut *Krismiaji* dalam bukunya yang berjudul Sistem Informasi Akuntansi, menjelaskan bahwa sistem informasi akuntansi adalah sebuah sistem yang memproses data dan transaksi guna

menghasilkan informasi yang bermanfaat untuk merencanakan, mengendalikan dan memproses bisnis. (*Rizal Qosidi, Supriyati dan Rio Yunanto, S.Kom, 2010 : 4*)

II.2.1 Penggajian

Gaji adalah sebuah komponen yang mutlak dikeluarkan oleh perusahaan sebagai kompensasi bagi karyawan, yang mana hal ini untuk menjamin keberlangsungan perusahaan itu sendiri. (*Miftakhul Huda; 2013 : 1*).

II.2.2. Contoh Teknik Penggajian

1. *Pay For Position*, yaitu penetapan gaji karyawan disebabkan pertimbangan nilai suatu pekerjaan yang menjadi tanggung jawabnya.
2. *Pay For Person*, yaitu penetapan gaji karyawan disebabkan pertimbangan orangnya yang dapat dibedakan dari aspek kompetensinya (*Pay For competency*), atau *skill based pay*, atau juga karena faktor kelangkaan ketersediaan orang dengan kualifikasi yang sesuai di pasar tenaga kerja.
3. *Pay For Performance*, yaitu penetapan gaji karyawan disebabkan pertimbangan pencapaian kinerja seseorang.
4. *Pay For year of services*, yaitu penetapan gaji karyawan disebabkan pertimbangan masa kerja seseorang. (*Miftakhul Huda; 2013 : 3*)

II.2.3 Sistem Informasi Akuntansi Penggajian

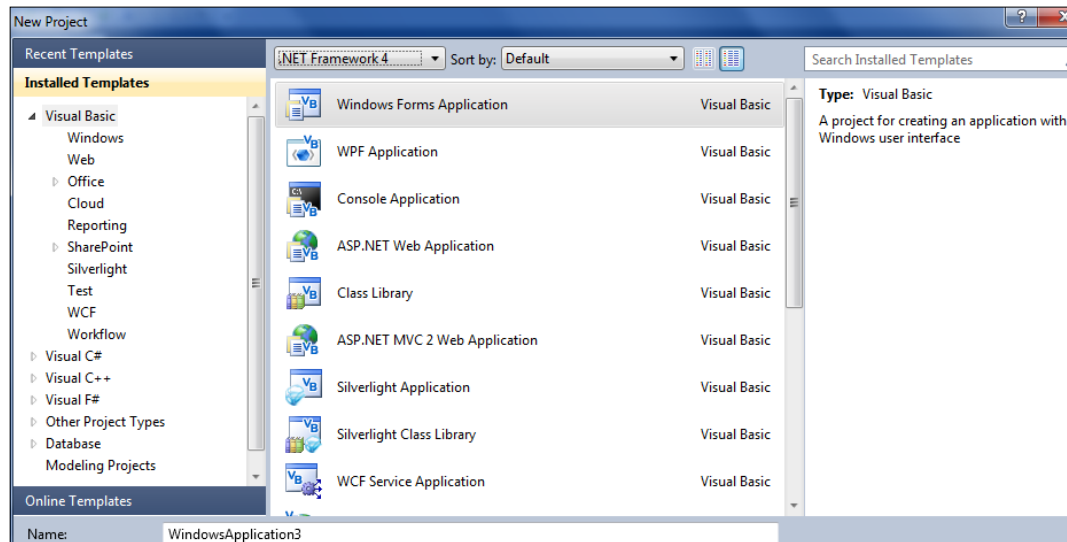
Sistem Informasi Akuntansi Penggajian adalah gabungan antara sekumpulan manusia dan sumber – sumber modal dalam suatu organisasi yang bertanggung jawab atas tersedianya informasi keuangan dalam bidang pembayaran gaji, dimana informasi yang diperlukan tersebut berasal dari pengumpulan dan pengolahan data – data transaksi yang terjadi. (*Dwi Prasetyo Wibowo*)

II.3. Visual Basic 2010

Visual basic 2010 merupakan lingkungan pengembangan terintegrasi atau bisa disebut *IDE* yang dikembangkan bahasa pemrograman *BASIC*. Bahasa *BASIC* sendiri sebenarnya sudah lama dibuat dan dikembangkan oleh *microsoft corporation* dengan nama *Microsoft Quick Basic*. Kesederhanaan sintaks dan *fleksibilitas* bahasa *Basic* dengan menyebabkan bahasa pemrograman ini begitu fenomenal sehingga banyak disukai dan dipakai oleh programmer di seluruh dunia.

Berbekal kepopuleran tersebut, *Microsoft* mengembangkan bahasa *Basic* ini menjadi produk yang sangat terkenal di kalangan programmer, yaitu mulai *Microsoft Basic 6.0* sampai sekarang, yaitu *Microsoft Visual Basic 2010*. Perkembangan teknologi dan penambahan banyak sekali fitur Pada *Visual Basic 2010* tidak mengakibatkan adanya perubahan sintaks –sintaks dasar yang terdapat di dalamnya. Sehingga dapat dikatakan, untuk dapat menjadi programmer *Visual Basic 2010* yang sebenarnya anda diharuskan menguasai dan mengerti

bagaimana menggunakan dan mengimplementasikan sintaks dasar dalam bahasa pemrograman Basic. (Wahana Komputer, 2010 : 36)



Gambar II.1 **Jendela pembuatan project**
(Sumber : Wahana Komputer ; 2010 : 39)

II.4. Database

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. Database sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. (Wahana Komputer, 2010:24)

Dalam database ada sebutan – sebutan untuk satuan data yaitu :

- a. Karakter, ini adalah satuan data terkecil.

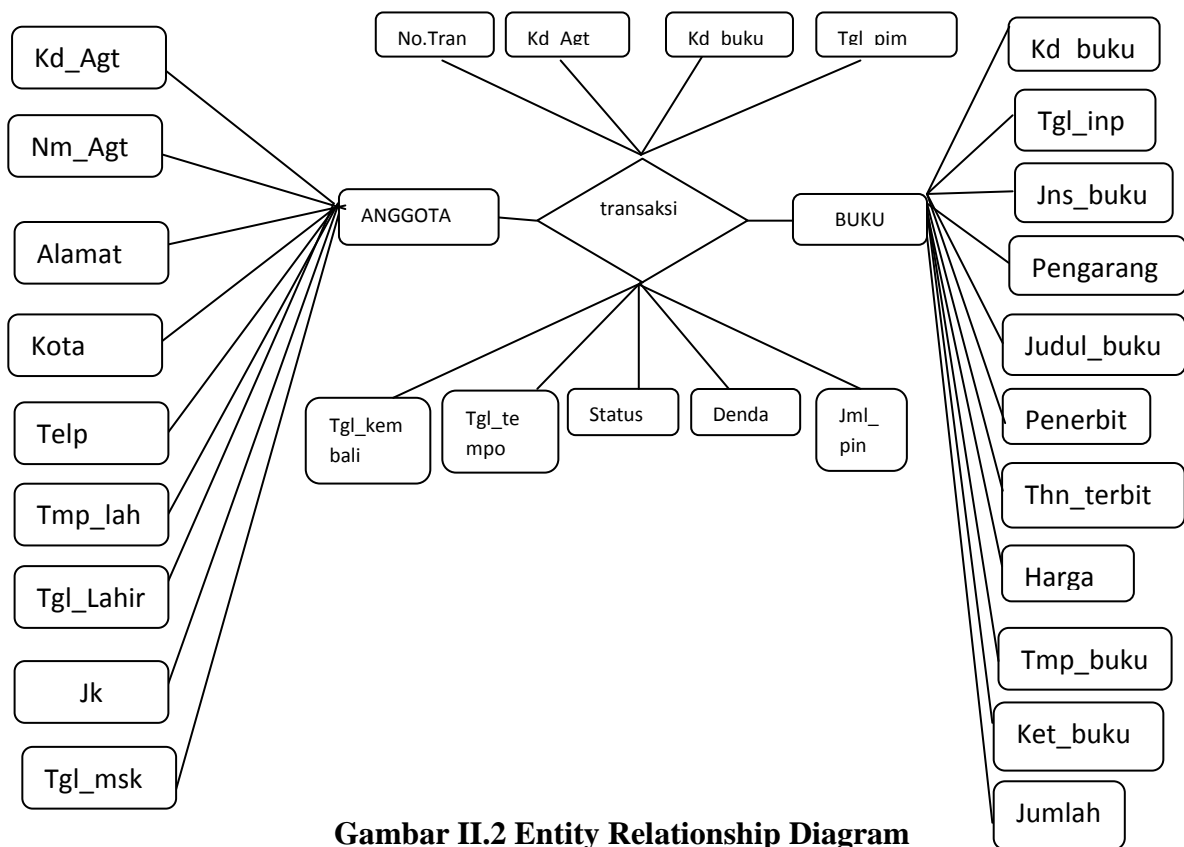
- b. *Field*, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir dan lain – lain.
- c. *Record*, adalah kumpulan dari field. Pada *record* anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan *field – field* yang ada.
- d. Tabel, adalah sekumpulan dari *record – record* yang memiliki kesamaan entity dalam dunia nyata.
- e. *File*, adalah bentuk fisik dari penyimpanan data. *File database* berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi.

II.5. SQL Server 2008

SQL Server adalah Sebuah terobosan baru dari microsoft dalam bidang *database*. *SQL Sever* adalah sebuah *DBMS (Database Management System)* yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolah data menyusul pendahulunya seperti *IBM* dan *Oracle*. *SQL Server 2008*, dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa terobosan dalam bidang pengolahan dan penyimpanan data. (Wahana computer, 2010 : 2)

II.6. ENTITY RELATIONSHIP DIAGRAM (ERD)

Pada dasarnya *ERD* (*Entity Relationship Diagram*) sebuah diagram yang secara konseptual, memetakan hubungan antara penyimpanan, *ERD* ini digunakan untuk melakukan pemodelan terhadap struktur data dan hubungannya. Penggunaan *ERD*, ini dilakukan untuk mengurangi tingkat kesulitan dalam penyusunan sebuah *database* yang baik.



Gambar II.2 Entity Relationship Diagram

(Sumber : Wahana Komputer ; 2010 : 30-31)

II.6.1. DDL (*Data Defenition Language*)

Skema basis data ditentukan sekumpulan defenisi yang dinyatakan dengan bahasa tertentu yang disebut *Data Defenition Language (DDL)*. Struktur

penyimpanan dan metode akses yang digunakan oleh sistem basis data dengan menggunakan sekumpulan pernyataan dalam tipe *DDL* tertentu disebut bahasa penyimpanan dan defenisi data. pernyataan ini menentukan penerapan detail skema basis data, yang seiring disembunyikan dari pengguna. Contohnya : *Create table rekening, (no_rekening char(10), Saldo integer).* (Janner Simarmata, 2010:16)

II.6.2 DML (*Data Manipulation Language*)

Manipulasi data adalah :

- a. Pengambilan informasi yang disimpan dalam basis data
- b. penempatan informasi baru dalam basis data
- c. penghapusan informasi dari basis data
- d. Modifikasi informasi yang disimpan dalam basis data

Bahasa manipulasi data (*DML*) adalah bahasa yang memungkinkan pengguna mengakses atau memanipulasi data seperti yang diatur oleh model data. (Janner Simarmata, 2010 : 16)

II.7. Normalisasi

II.7.1 Bentuk Normal Tahap Pertama (1NF)

Sebuah tabel relasional secara defenisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom – kolomnya adalah atomik. Ini berarti kolom –

kolom tidak mempunyai nilai berulang. Gambar dibawah menunjukkan tabel pemasok dalam 1NF. (*Janner Simarmata, 2010 :80*)

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 | Yogyakarta | b2 | 200 |
| p4 | 20 | Yogyakarta | b2 | 200 |
| p4 | 20 | Yogyakarta | b4 | 300 |
| p4 | 20 | Yogyakarta | b4 | 400 |

Gambar II.3 Tabel bentuk normal pertama (1NF)

(Sumber : *Janner Simarmata, 2010 : 80*)

II.7.2. Bentuk Normal Tahap Kedua (2NF)

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama (*Janner Simarmata, 2010 :81*).

PEMASOK2

| p# | Status | Kota |
|----|--------|------------|
| p1 | 20 | Yogyakarta |
| p2 | 10 | Medan |
| p3 | 10 | Medan |
| p4 | 20 | Yogyakarta |
| p5 | 30 | Bandung |

BARANG

| p# | b# | Qty |
|----|----|-----|
| p1 | b1 | 300 |
| p1 | b2 | 200 |
| p1 | b5 | 100 |
| p1 | b6 | 100 |
| p2 | b1 | 300 |
| p3 | b2 | 200 |
| p4 | b2 | 200 |
| p4 | b4 | 300 |
| p4 | b4 | 400 |

Gambar II.3 Tabel bentuk normal kedua**(Sumber : Janner Simarmata, 2010 : 82)****II.7.3. Bentuk Normal Tahap Ketiga (3NF)**

Bentuk Normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga atau 3NF jika tabel sudah berada pada 2NF 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 (*Janner Simarmata, 2010 :82*).

PEMASOK_KOTA

| p# | Status | Kota |
|----|--------|------------|
| p1 | 20 | Yogyakarta |
| p2 | 10 | Medan |
| p3 | 10 | Medan |
| p4 | 20 | Yogyakarta |
| p5 | 30 | Bandung |

KOTA_STATUS

| p# | Status | kota |
|----|--------|------------|
| p1 | 20 | Yogyakarta |
| p2 | 10 | Medan |
| p3 | 10 | Medan |
| p4 | 20 | Yogyakarta |
| p5 | 30 | Bandung |

Gambar II.4 Tabel bentuk normal ketiga (3NF)**(Sumber : Janner Simarmata, 2010 : 83)**

II.7.4. Bentuk Normal Tahap Keempat (4NF)

Sebuah tabel relasional berada pada bentuk normal keempat atau 4NF jika dia dalam *BCNF* dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat atau 4NF didasarkan pada konsep ketergantungan multivalued atau *MVD*. Sebuah ketergantungan multivalued terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak garis bernilai sama, tetapi kolom lain bernilai berbeda (*Janner Simarmata, 2010 :85*).

| Pegawai_Proyek | Pegawai_Proyek | | | | | | | | | | | | |
|--|----------------|--------|------|----|------|----|--|------|--------|------|----|------|----|
| <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="padding: 2px 5px;">peg#</th> <th style="padding: 2px 5px;">Status</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px 5px;">1211</td> <td style="padding: 2px 5px;">p1</td> </tr> <tr> <td style="padding: 2px 5px;">1211</td> <td style="padding: 2px 5px;">p3</td> </tr> </tbody> </table> | peg# | Status | 1211 | p1 | 1211 | p3 | <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="padding: 2px 5px;">peg#</th> <th style="padding: 2px 5px;">Status</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px 5px;">1211</td> <td style="padding: 2px 5px;">p1</td> </tr> <tr> <td style="padding: 2px 5px;">1211</td> <td style="padding: 2px 5px;">p3</td> </tr> </tbody> </table> | peg# | Status | 1211 | p1 | 1211 | p3 |
| peg# | Status | | | | | | | | | | | | |
| 1211 | p1 | | | | | | | | | | | | |
| 1211 | p3 | | | | | | | | | | | | |
| peg# | Status | | | | | | | | | | | | |
| 1211 | p1 | | | | | | | | | | | | |
| 1211 | p3 | | | | | | | | | | | | |

Gambar II.5 Tabel bentuk normal keempat (4NF)

(Sumber : *Janner Simarmata, 2010 : 86*)

II.7.5. Bentuk Normal Tahap Kelima (5NF)

Sebuah tabel berada pada bentuk normal kelima 5NF jika dia tidak dapat mempunyai dekomposisi *lossles* menjadi sebuah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan konsep ketergantungan gabungan atau *join dependence*. Ketergantungan gabungan berarti bahwa sebuah tabel, setelah didekomposisi menjadi tiga atau lebih tabel yang lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain, 5NF

menunjukkan ketika sebuah tabel tidak dapat didekomposisi lagi (*Janner Simarmata, 2010 :86*).

II.8. UML (*Unified Modeling Language*)

UML singkatan dari *Unified Modeling Language* yang berarti pemodelan standart. (*Chonoles, 2003 : bab 1*) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan semantik. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya.

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

UML merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem, mau tidak mau pasti akan menjumpai *UML*, baik kita sendiri yang membuat atau sekedar membaca diagram *UML* buatan orang lain. (*Prabowo Pudjo Widodo, 2011 : 7*)

II.8.1. Use Case Diagram

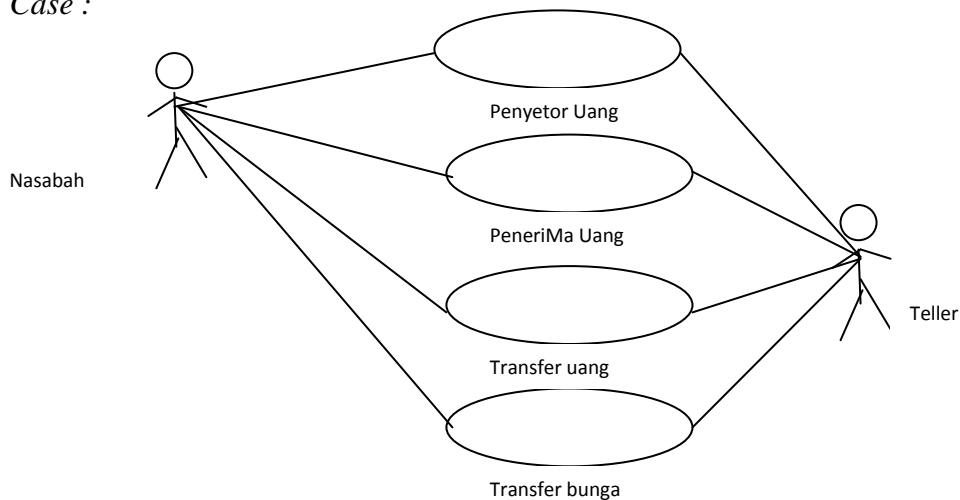
Salah satu kontributor terhadap diagram *Use Case* dalam *UML* adalah *Ivar Jacobsen*. *Use Case* menggambarkan *external view* dari sistem yang akan kita buat modelnya. (*Pooley,2003:15*) mengatakan bahwa model *Use Case* dapat dijabarkan dalam diagram *Use Case*, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram. (*Prabowo Pudja Widodo, 2011 : 16*)

Komponen pembentuk diagram *Use Case* adalah :

- a. Aktor (*Actor*) , menggambarkan pihak – pihak yang berperan dalam sistem
- b. Use Case, aktifitas/sarana yang disiapkan oleh bisnis/sistem
- c. Hubungan (*Link*) , aktor mana saja yang terlibat dalam *Use Case* ini

Gambar dibawah ini merupakan salah satu contoh bentuk diagram *Use*

Case :

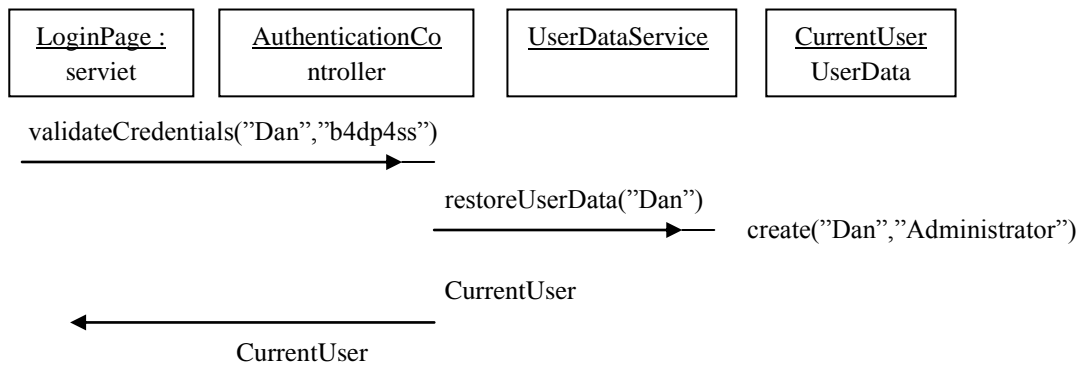


Gambar II.6 Diagram Use Case

(Sumber : Prabowo Pudja Widodo ; 2011 : 17)

II.8.2. Sequence Diagram

(Douglas, 2004: bab 3.5) menyebutkan ada tiga diagram primer UML dalam memodelkan skenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*), dan diagram komunikasi (*communication diagram*.) Namun demikian (Pilone, 2005: bab 10) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. (Prabowo Pudja Widodo, 2011 : 174)



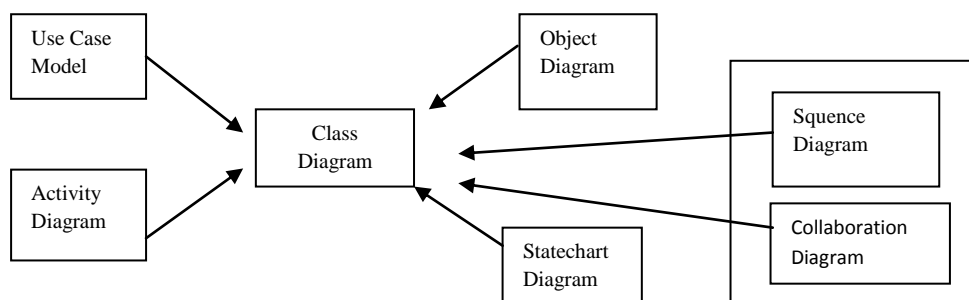
Gambar II.7. Contoh Diagram Urutan (Sequence Diagram)

(Sumber : Prabowo Pudja Widodo ; 2011 : 175)

II.8.3. Class Diagram

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya menggunakan kode program menjadi model.

(Prabowo Pudjo Widodo , 2011 :37)



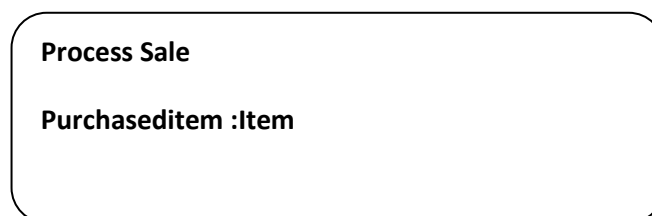
Gambar II.8. Class Diagram

(Sumber : Prabowo Pudjo Widodo ; 2011 :38)

II.8.4. Activity Diagram

Diagram aktifitas lebih mefokuskan diri pada eksekusi dan alur sistem, daripada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software, melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk, kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas mempresentasikan, pemanggilan sesuatu fungsi tertentu, misalnya *call*, Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini digambarkan atau dipicu oleh kejadian-kejadian internal, misalnya proses penggajian tiap jumat sore. (*Prabowo Pudjo Widodo*, 20011:143).

Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.9. Aktivitas sederhana tanpa rincian

(Sumber : Prabowo Pudjo Widodo ; 2011 : 145)