

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Konsep Dasar Sistem

Sistem adalah suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antarbagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerjasama antar bagian yang interdependen satu sama lain. Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. (Marimin, Hendri Tanjung, Haryo Prabowo; 2012 : 1).

II.1.2 Konsep Dasar Informasi

Information is Power (Ralph Stair, George Reynold ; 2012:3). Faktanya, kekuatan dari sebuah informasi bergantung pada bagaimana informasi tersebut menyediakan kebutuhan yang spesifik akan informasi tersebut pada waktu tertentu.

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 akuntansi terdiri dari komponen- komponen yang disebut blok bangunan atau *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

Sistem Informasi Akuntansi Adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi Keuangan. Misalnya salah satu input dari system informasi akuntansi pada sebuah toko baju, seperti contoh sebelumnya, adalah transaksi penjualan. Kita memperoleh transaksi penjualan dengan mencatat penjualan tersebut kedalam jurnal penjualan, mengklafikasikan transaksi dengan menggunakan kode rekening dan memposting transaksi kedalam jurnal. Kemudian, secara prodik system informasi akan menghasilkan output berupa laporan keuangan yang terdiri dari neraca dan laporan laba rugi (Anastasia Diana ; 2011 : 4).

II.3. Visual Basic 2008

Visual basic 2008 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 2008. Perkembangan teknologi dan penambahan banyak sekali fitur Pada Visual Basic 2010 tidak mengakibatkan adanya perubahan sintaks – sintaks dasar yang terdapat didalamnya. Sehingga dapat dikatakan, untuk dapat menjadi programmer Visual Basic 2008 yang sebenarnya anda diharuskan menguasai dan mengerti bagaimana menggunakan dan mengimplementasikan sintaks dasar dalam bahasa pemrograman Basic (Wahana Komputer ; 2010 : 36).

II.4. Basis data

Basis Data, Menurut *Stephens* dan *Plew* (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. dengan basis data, pengguna dapat

menyimpan data secara terorganisir. Setelah data disimpan informasi harus mudah diambil.

Kemudian, Silberschatz, dkk., (2002) mendefinisikan basis data sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Sistem manajemen basis data (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data.

Ramakrishnan dan *Gehkre* (2003) menyatakan basis data sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.

Defenisi Basis data, menurut *Mcleod*, dkk., (2001) adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi. Sistem manajemen basis data adalah aplikasi perangkat lunak yang menyimpan struktur basis data, hubungan antar data dalam basis data, serta berbagai formulir dan laporan yang berkaitan dengan basis data. Basis data yang dikendalikan oleh sistem manajemen basis data adalah suatu set catatan data yang berhubungan dan saling menjelaskan (Janner Simarmata ; 2010 : 2).

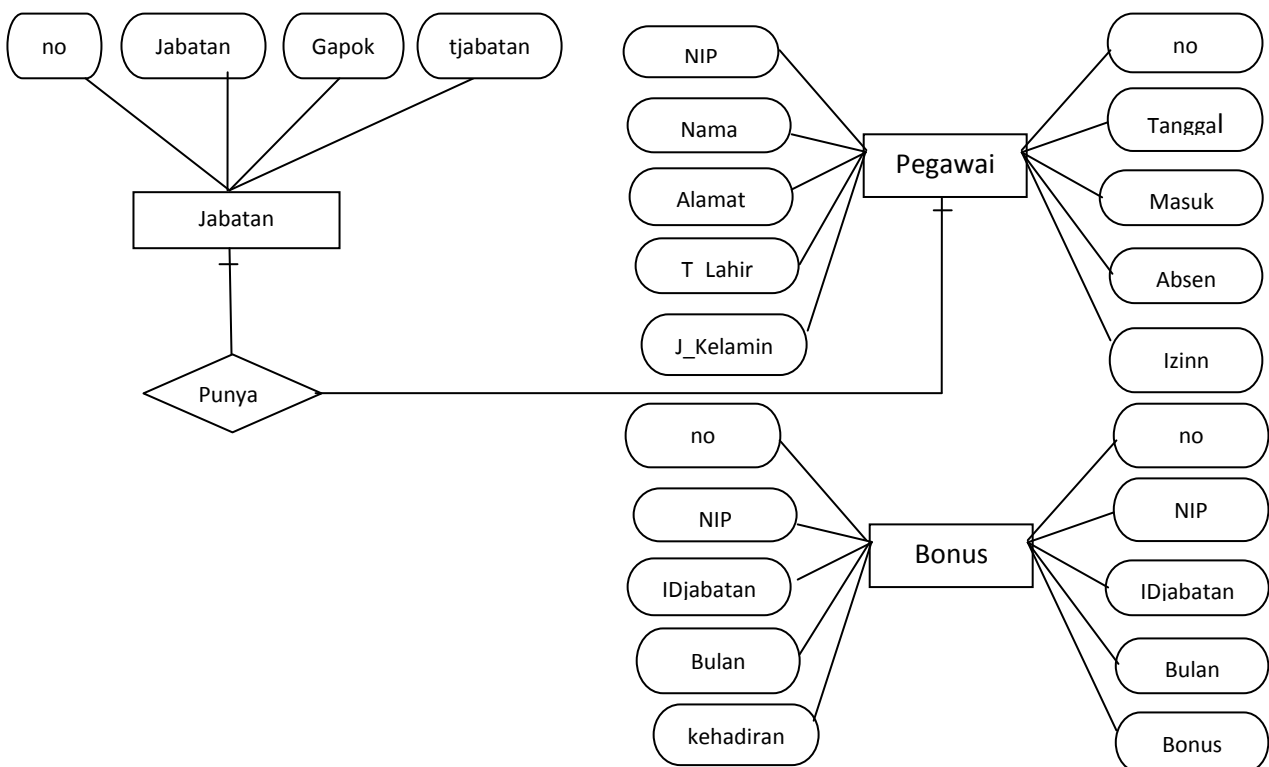
II.5. SQL Server 2008

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. SQLServer

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 data base yang baik.



Gambar II.1. Entity Relationship Diagram

Sumber : Wahana Komputer ; 2010 : 31

II.7. Data Defenition Language (DDL)

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 : 38).

II.8. Tabel

Tabel adalah unit penyimpanan fisik utama untuk data dalam basis data. Ketika mengakses basis data, pengguna mengacu tabel untuk data yang diinginkan. Basis data terdiri atas banyak *table* sehingga harus dapat relasi antar tabel. Karena menyimpan data , tabel membutuhkan penyimpanan fisik pada host komputer untuk basis data (Janner Simarmata ; 2010 : 38).

II.9. Normalisasi

II.9.1 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 sesuai keadaannya.

II.9.2 Bentuk Normal Tahap Pertama

Bentuk normal tahap pertama (1NF) terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*multivalued attribute*) atau lebih dari satu *atribut* dengan nilai *domain* yang sama. Masing-masing sel bernilai tunggal. Permintaan yang menyatakan tidak ada yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci (Janner Simarmata ; 2010 : 79).

II.9.3. Bentuk Normal Tahap Kedua

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).

II.8.4. Bentuk Normal Tahap Ketiga

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).

II.9.5. Bentuk Normal Tahap Keempat

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).

II.9.6. Bentuk Normal Tahap Kelima

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.10. UML (*Unified Modelling Language*)

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standar. (Chonoles,2003: bab 1)

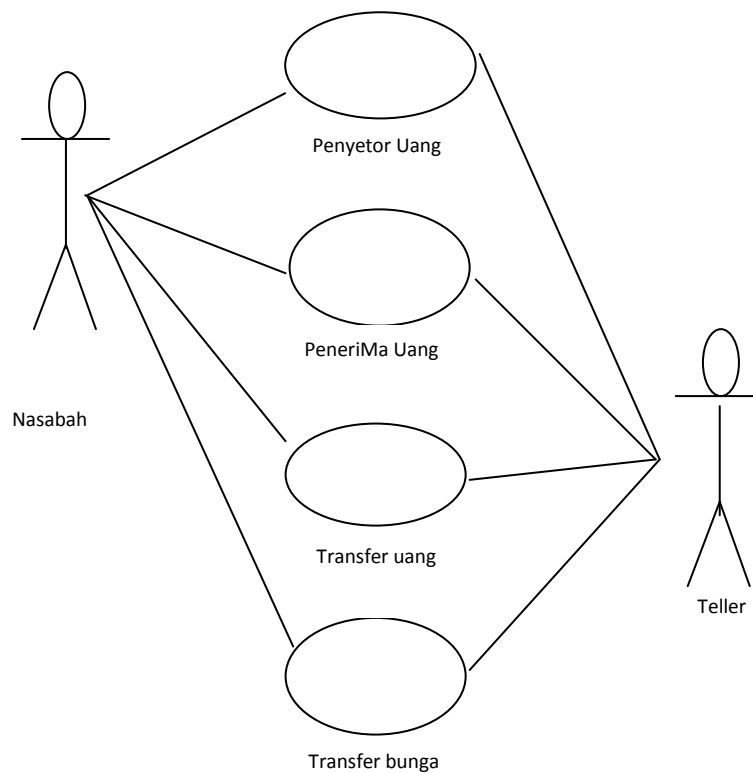
Mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan semantik. Ketika kita berbuat model menggunakan konsep *UML* ada aturan- aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya ? Bagaimana sistem mengatasi eror yang terjadi ? Bagaimana keamanan terhadap sistem yang kita buat ? Dan sebagainya dapat dijawab dengan *UML*.

Unified Modelling Language (*UML*) adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisis dan desain berorientasi objek. *UML* menyediakan standar dan diagram yang bisa digunakan untuk memodelkan suatu sistem. *UML* memungkinkan developer melakukan pemodelan secara visual, yaitu penekanan pada penggambaran. (Prabowo P.Widodo, dkk, menggunakan *UML* 2011)Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari objek, mempermudah penggambaran interaksi antar elemen dalam sistem dan mempertahankan konsistensi antara desain dan implementasi dalam pemrograman. Tiga artifak penting *UML* untuk proses analisis dan desain adalah Use Case Diagram, Sequence Diagram dan Class (Prabowo Pudjo Widodo ; 2011 : 7).

II.10.1. Use Case Diagram

Menurut (Pilone,2005: bab 9.1) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten,

2004: 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal . *Use case* digambarkan dalam bentuk elips/ oval. *Use case* sangat menentukan karakteristik sistem yang kita buat, oleh karena itu (Chonoles, 2003: bab 8) menawarkan cara untuk menghasilkan *use case* yang baik.



Gambar II.2. Diagram Use Case

Sumber : Prabowo Pudjo Widodo ; 2011 : 17

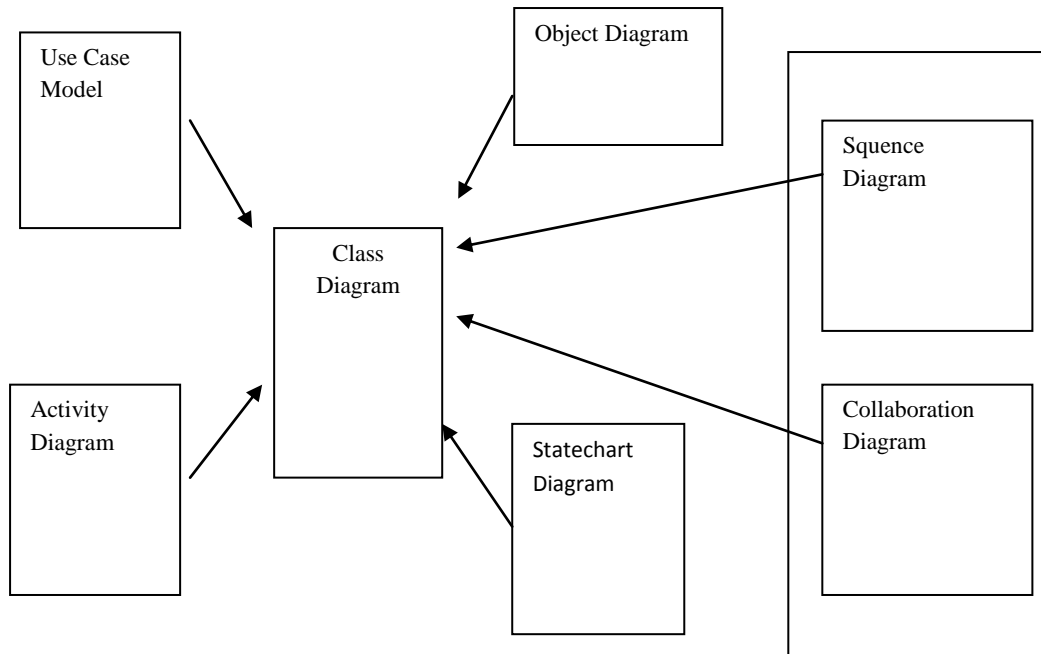
II.10.2. Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan yang dianalisa dalam *use case*, hal ini meliputi penjabaran interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Pembuatan *sequence* diagram merupakan aktivitas yang paling kritikal dari proses disain karena artifak inilah yang menjadi pedoman dalam proses pemrograman nantinya dan juga berisi aliran kontrol program.

Sequence diagram biasanya terdiri dari elemen objek, interaction dan *message*. *Interaction* menghubungkan 2 objek dengan pesannya. Diagram ini menjelaskan aspek dinamis sistem yang sedang dibangun.

II.10.3. Class Diagram

Class diagram 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.



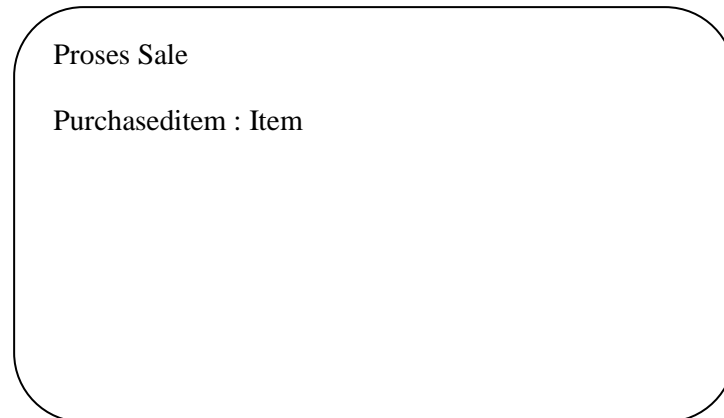
Gambar II.3. Class Diagram

Sumber : Prabowo Pudjo Widodo ; 2011 :38

II.10.4. Activity Diagram

Diagram aktifitas lebih mefokuskan diri pada eksekusi dan alur sistem, daripada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan softwear, melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk, kumpulan aksi-aksi. Ketika digunakan dalam pemodelan softwear, diagram aktivitas mempresentasikan, pemanggilan sesuatu fungsi tertentu, misalnya *call*, Sedangkan dipergunakan dalam pemodelan bisnis, diagram ini digambarkan atau dipicu, dengan kejadian-kejadian yang di luar seperti pemesanan, atau kejadian internal, misalnya proses pendataan (prabowo pujo widodo ; 2011 : 143).

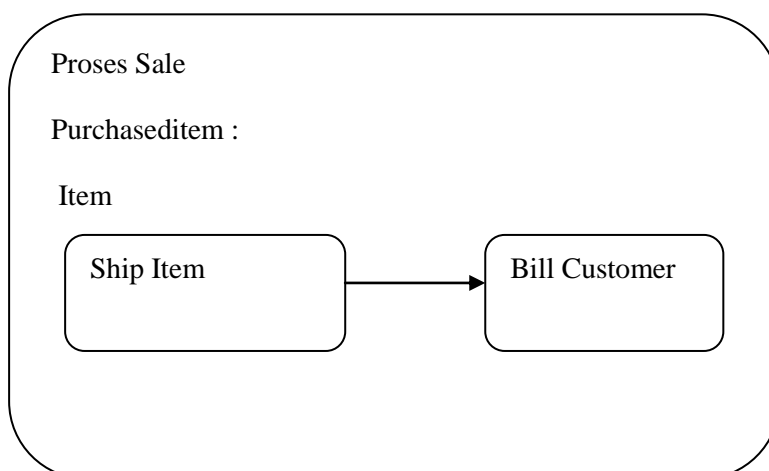
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis dikiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.4. Aktivitas sederhana Tanpa rincian

Sumber : Prabowo Pudjo Widodo Dan Herlawati(2011:145)

Detail aktivitas dapat dimasukkan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan nama nya diletak didalam persegi panjang.

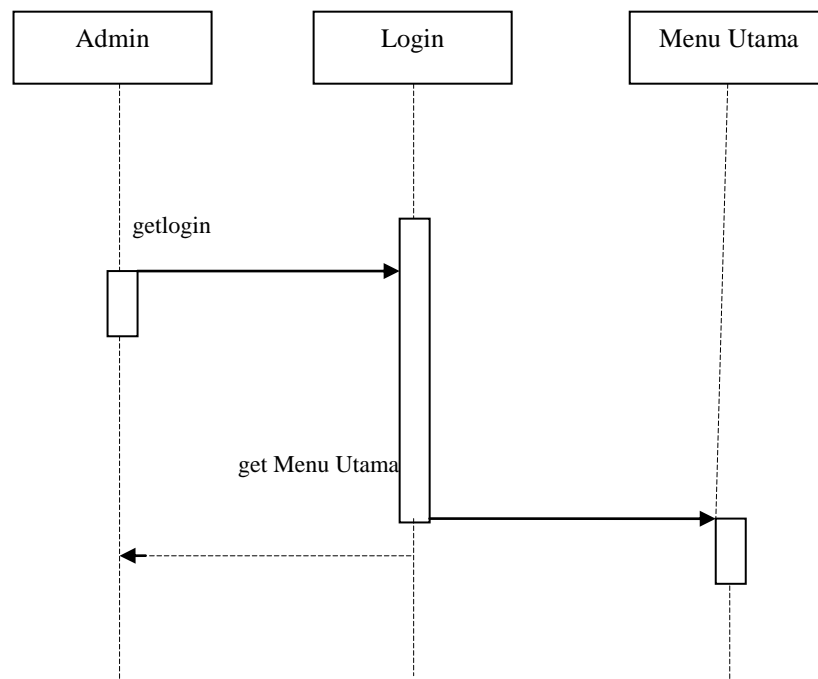


Gambar II.5. Aktivitas sederhana Tanpa rincian

Sumber : Prabowo Pudjo Widodo Dan Herlawati(2011:145)

Menurut Douglas (2004 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Menurut Pilone (2005 : 174) menyatakan bahwa diagram paling banyak dipakai adalah diagram urutan. Gambar II.10. memperlihatkan contoh diagram urutan dengan notasi-notasi nya yang akan dijelaskan nanti nya (Prabowo Pudjo Widodo Dan Herlawati, ; 2011:174-175).



Gambar II.6 Sequence Diagram Urutan

Sumber : Prabowo Pudjo Widodo Dan Herlawati(2011:145)