

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi (SI)

Sistem informasi (SI) 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 (Budi Sutedjo Dharama Oetomo: 2006 ;11)

II.2. Siklus 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 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 seluruh 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 strategi (Budi Sutedjo Dharama Oetomo ; 2006 : 12-13).

II.3. Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar meliputi enam hal berikut ini

II.3.1. Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik dari pada seseorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang :

1. Fakta-fakta tentang bidang permasalahan tertentu
2. Teori-teori tentang bidang permasalahan tertentu
3. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya
4. Aturan *herustic* yang harus dikerjakan dalam suatu situasi tertentu
5. Strategi global untuk memecahkan permasalahan
6. Pengetahuan tentang pengetahuan (meta knowledge) (Sutejo,T,dkk; 2011:163).

II.3.2. Pakar (*Expert*)

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberikan nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topic permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya. Jadi seseorang pakar harus mampu melakukan kegiatan-kegiatan berikut :

1. Mengenal dan memformulasikan permasalahan
2. Memecahkan permasalahan secara cepat dan tepat
3. Menerangkan pemecahannya
4. Belajar dari pengalaman
5. Merestrukturisasi pengetahuan
6. Memecahkan aturan-aturan
7. Menentukan relevansi (Sutejo, T, dkk; 2011:163-164).

II.3.3. Pemindahan Kepakaran (*Transferring Expertise*)

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam computer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu :

1. Akuisisi pengetahuan (dari pakar atau sumber lain)
2. Representasi pengetahuan (*pada computer*)
3. Inferensi pengetahuan

4. Pemindahan pengetahuan ke pengguna (Sutejo,T, dkk ; 2011: 164).

II.4. Backward Chaining

Backward Chaining adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari Goal (yang berada dibagian THEN dari rule IF-THEN, kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian IF. Jika cocok, rule dieksekusi, kemudian hipotesis dibagian THEN ditempatkan dibasis data sebagai fakta baru. Jika tidak cocok, simpan premis dibagian IF kedalam stack sebagai subGoal. Proses berakhir jika Goal ditemukan atau tidak ada rule yang bisa membuktikan kebenaran dari subGoal atau Goal (Sutejo,T, dkk ; 2011: 178-179).

Untuk memahami cara kerja *backward chaining*.

Contoh :

Misalkan diketahui sistem pakar yang menggunakan beberapa rule berikut :

R1 : **IF (Y AND D) THEN Z**

R2 : **IF (X AND B AND E) THEN Y**

R3 : **IF A THEN X**

R4 : **IF C THEN L**

R5 : **IF (L AND M) THEN N**

Fakta- fakta : A, B, C, D, Dan E bernilai benar.

Goal : menentukan apakah Z bernilai benar atau salah

II.5. Pengenalan MySQL

MySQL adalah sebuah program database *server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi User* serta menggunakan perintah standar SQL (Structured Query Language). MySQL memiliki dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa kita gunakan adalah MySQL FreeSoftware yang berada dibawah Lisensi GNU/GPL (General Public Lisensi). Selain itu anda juga dapat memiliki produk MySQL yang sifatnya komersial, biasa disebut dengan MySQL AB. MySQL menggunakan logo standar berbentuk ikan lumba-lumba, seperti berikut ini :



**Gambar II.1. Logo MySQL dengan gambar lumba – lumba
(Bunafit Nugroho ; 2004 :1)**

MySQL merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius. Selain sebagai database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai server. Pada saat itu berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah database yang dapat digunakan baik sebagai *Client* maupun *Server*.

Database mungkin biasa kita baca **mai-es-ki-el** atau bisa juga **mai-se-kuel**, merupakan suatu perangkat lunak database yang berbentuk database relasional atau dalam bahasa basis data sering kita sebut dengan Relation Database Management System (RDMS) yang menggunakan suatu bahasa permintaan bernama SQL.

Sebagai pengguna awal database mungkin anda merasa bingung dengan kedua istilah antara MySQL dengan SQL. Nah, yang perlu anda pahami adalah bahwa kedua istilah tersebut sangat berbeda artinya. MySQL adalah sebuah program database, sedangkan SQL adalah bahasa perintah *query* dalam program MySQL. Selain MySQL ada beberapa program database server lain menggunakan standar query berupa SQL antara lain adalah :

1. Oracle
2. PostgreSQL
3. MySQL front
4. MsQL
5. SQL Server 97/ ...
6. Dll

Kelebihan MySQL adalah sebagai berikut :

1. MySQL Sebagai *Data Management System*(DBMS)
2. MySQL sebagai *Relation Database Mangement System* (RDBMS)
3. MySQL adalah sebuah Software database yang *OpenSource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.

4. MySQL merupakan sebuah database *server*, jadi dengan menggunakan database ini dapat anda dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
5. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Treading* (Bunafit Nugroho ; 2005 : 1-3).

II.6. Bahasa Pemrograman PHP

PHP adalah program aplikasi yang bersifat *Server Side*, artinya hanya dapat berjalan pada sisi server saja dan tidak dapat berfungsi tanpa adanya sebuah server di dalamnya. PHP juga bukan sebuah bahasa pemrograman yang lengkap. Maksudnya, program ini tidak menyertakan sebuah *compiler* tersendiri yang membuat program hasilnya menjadi menjadi *program.exe* yang dapat dijalankan tersendiri. Program ini akan selalu membutuhkan sebuah server pendukung yang disebut *Web Server* dan program PHP itu sendiri untuk menjalankan sebuah *script* program.

PHP (*Hypertext Preprocessor*) merupakan sebuah bahasa pemrograman scripting berlisensi *Open Source*. *Script* ini dapat bercampur dengan *Script Tag* HTML sehingga karena kemampuannya tersebut, ia disebut sebagai bahasa yang *embeded* pada Tag HTML anda dapat memperoleh program mirror PHP versi terbaru pada alamat situs <http://www.php.net>, atau apabila menggunakan system operasi Linux, Anda dapat langsung mengekstrak bentuk kompresan dari CD distro Linux anda. (Bunafit Nugroho ; 2005 : 369 - 370).

II.7. Kamus Data

Kamus data ikut berperan dalam perancangan dan pembangunan SI karena peralatan ini berfungsi untuk :

1. Menjelaskan arti aliran data dan penyimpanan dalam penggambaran dalam *data flow diagram*.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran, misalnya data alamat diurai menjadi nama jalan , nomor, kota, negara dan kode pos
3. Menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut (Budi Sutedjo Dharma Oetomo ; 2006 : 118-119).

Sejumlah simbol yang digunakan dalam penggambaran kamus data ditunjukkan dalam tabel.

Tabel II.1. Simbol – simbol Dalam Kamus Data

Simbol	Uraian
=	Terdiri atas, mendefinisikan, diuraikan menjadi, artinya contoh : nama=sebutan+nama1+nama2+gelar1+gelar2
+	Dan
()	Optional (pilihan boleh ada atau boleh tidak) Contoh : alamat=alamat rumah +(alamat surat)
{ }	Pengulangan Contoh : nama1={karakter_valid}
[]	Memilih salah satu dari sejumlah alternatif, seleksi Contoh : sebutan = [Bapak/Ibu/Yang Mulia]
**	Komentar Contoh : *seminar yang akan diikuti*

I	Pemisah sejumlah alternatif pilihan antara simbol []
---	---

(Budi Sutedjo Dharma Oetomo ; 2006 :118-119)

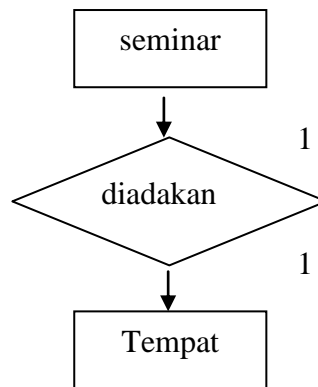
Berikut ini contoh penggunaan kamus data untuk perancangan database pembicaraan seminar :

1. Data_pembicara=
kode_pembicara+nama+alamat+telepon+tanggal_lahir+honor+spesialisasi
2. Kode_pembicara=level_pembicara+nomor_urut
3. Nama=sebutan+nama1+nama2+gelar1+gelar2+gelar3
4. Alamat=nama_jalan+nomor+kota+negara+kodepos
5. Telepon =telepon_rumah+ponsel
6. Tanggal_lahir=*dd/mm/yyyy*
7. Honor=*99.999.999*
8. Spesialisasi=0{karakter} 15

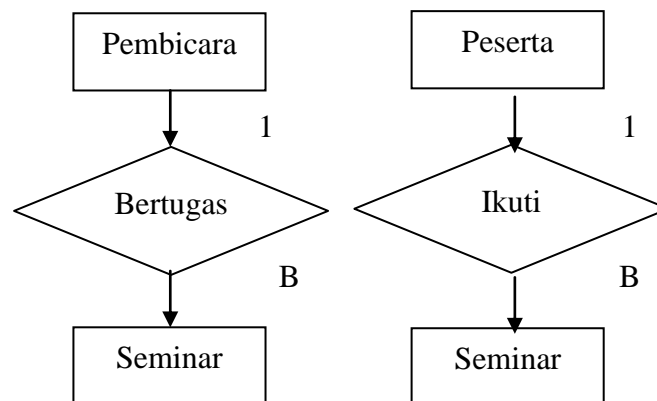
II.8. E-R Diagram

E-R Diagram berfungsi untuk menggambarkan relasi dari dua file atau dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi, yaitu 1-1, satu-banyak, dan banyak-banyak

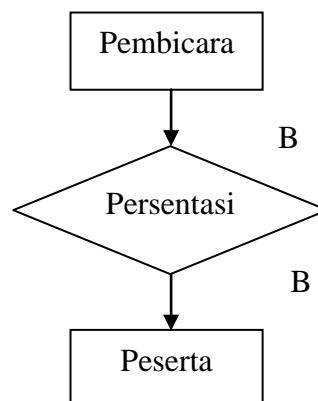
Penggambaran ini akan membantu analisis sistem dalam melakukan perancangan proses yang kelak akan dituangkan dalam bentuk baris-baris program ((Budi Sutedjo Dharma Oetomo ; 2006 : 130-131).



Gambar II.2. Ilustrasi E-R Diagram satu –satu
(Budi Sutedjo Dharma Oetomo ; 2006 :130)

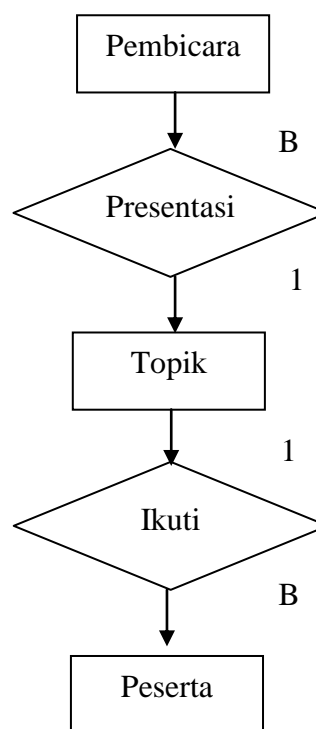


Gambar II.3. Ilustrasi E-R Diagram satu-banyak
(Budi Sutedjo Dharma Oetomo ; 2006 : 130)



Gambar II.4. Ilustrasi E-R Diagram banyak-banyak
(Budi Sutedjo Dharma Oetomo ; 2006 : 130)

Khusus untuk relasi banyak-banyak, harus dipecah dengan penambahan file perantara agar hubungan yang terbentuk antara file yang satu dengan file yang lain tidak secara langsung tetapi melalui file perantara tersebut. Untuk kasus diatas, maka dapat digambarkan sebagai berikut :



**Gambar II.5. Ilustrasi E-R Diagram banyak-banyak
(Budi Sutedjo Dharma Oetomo ; 2006 : 131)**

II.8. Unified Modeling Language (UML)

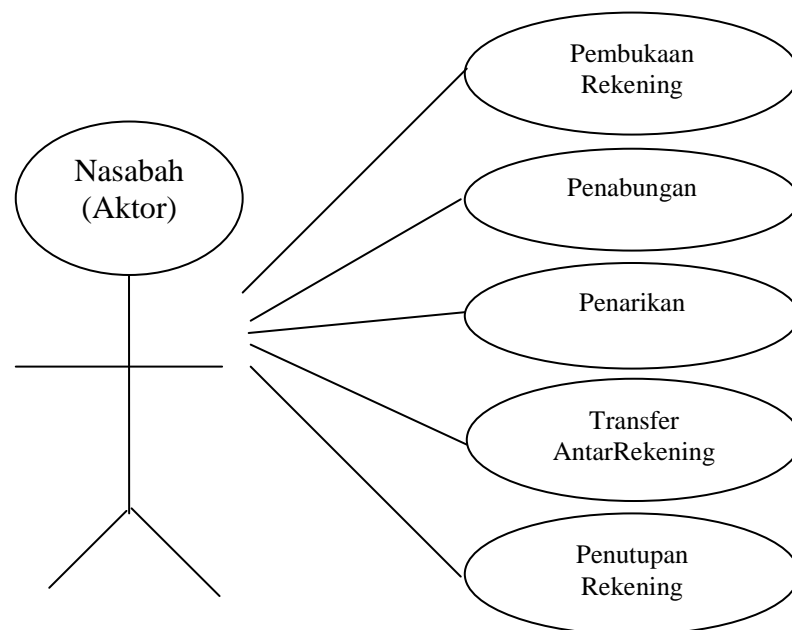
UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal itu disebabkan karena UML menyediakan bahasa pemodelan *visual* yang

memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar ; 2005 : 17).

Beberapa jenis *Diagram Unified Modeling Language* adalah sebagai berikut :

II.8.1. Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah system dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeskripsikan *tipikal interaksi* antara *user* (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai (Munawar ; 2005 : 71).



Gambar II.6. Contoh Use Case Diagram
(Adi Nugroho ; 2009 : 8)

Aktor pada dasarnya ditentukan berdasarkan perannya (*role*) pada program/aplikasi yang sedang kita kembangkan, bukan sebagai objek-objek secara mandiri. Sebagai contoh, jika kita mengambil kasus ATM (Anjungan Tunai Mandiri), seseorang (objek tunggal) mungkin bisa dikelompokkan sebagai aktor karyawan bank serta nasabah jika orang tersebut merupakan karyawan bank yang bersangkutan sekaligus sebagai nasabah karena memiliki tabungan di bank tersebut. Sementara itu, Adi, Ana, Geuis, dan beberapa orang lainnya dapat dikelompokkan menjadi aktor nasabah jika mereka semua masing – masing memiliki tabungan di bank tersebut. dalam hal ini, kita akan coba mengambil contoh aktor nasabah untuk menentukan *use case*-nya.

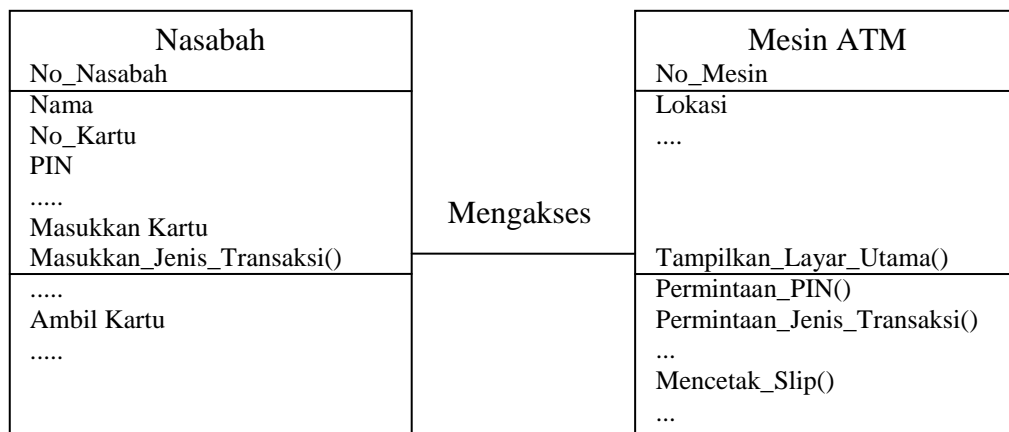
Untuk aktor nasabah ini, kita akan mencari tahu mengenai apa yang sebenarnya nasabah lakukan saat berinteraksi dengan aplikasi perbank. Dalam hal ini, kita mungkin mendapatkan perilaku-perilaku (*behaviour*) aktor nasabah seperti berikut.

1. Nasabah membuka rekeningnya.
2. Nasabah memeriksa saldo rekeningnya.
3. Nasabah menyimpan uang pada rekeningnya.
4. Nasabah menarik uang dari rekeningnya.
5. Nasabah melakukan transfer uang antarrekening.
6. Nasabah menutup rekeningnya.

Untuk menentukan *use case* dari kasus bank tersebut, kita harus melihat mekanisme yang dilakukan oleh masing – masing aktor pada masing – masing perilaku yang telah disebutkan.

II.8.2. Class Diagram

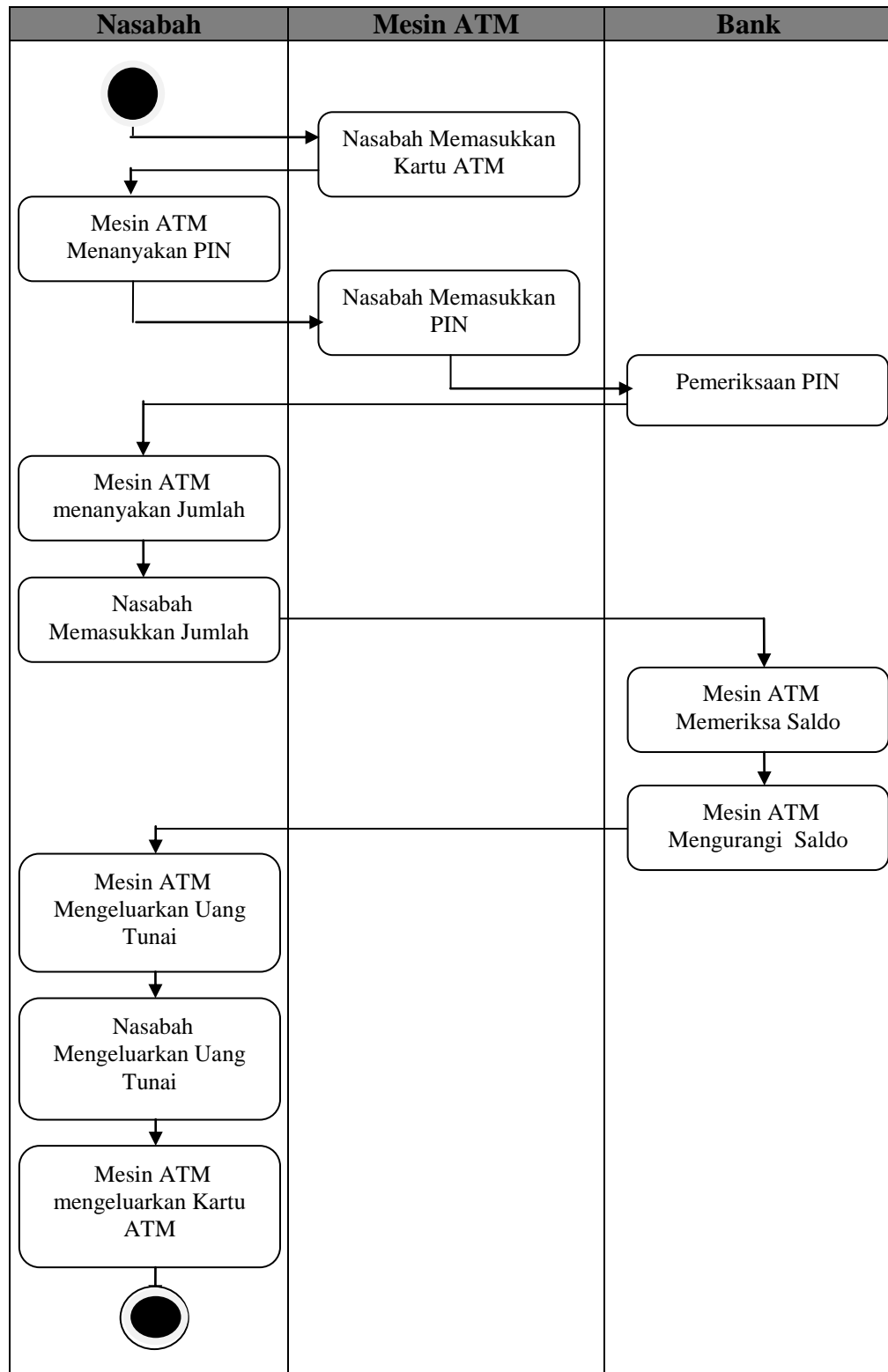
Attribute adalah *property* dari sebuah *class*. *Attribute* ini melukiskan batas nilai yang mungkin ada pada objek *class*. Sebuah *class* mungkin mempunyai nol atau lebih *attribute*. Secara *konvensi*, jika nama *attribute* terdiri dari satu suku kata, maka ditulis dengan huruf kecil. Akan tetapi jika nama *attribute* mengandung lebih dari satu suku kata maka semua suku kata digabungkan dengan suku kata pertama menggunakan huruf kecil dan awal suku kata berikutnya menggunakan huruf besar (Munawar ; 2005 : 35).



Gambar II.7. Contoh Class Diagram
(Adi Nugroho ; 2009 : 39)

II.8.3. Activity Diagram

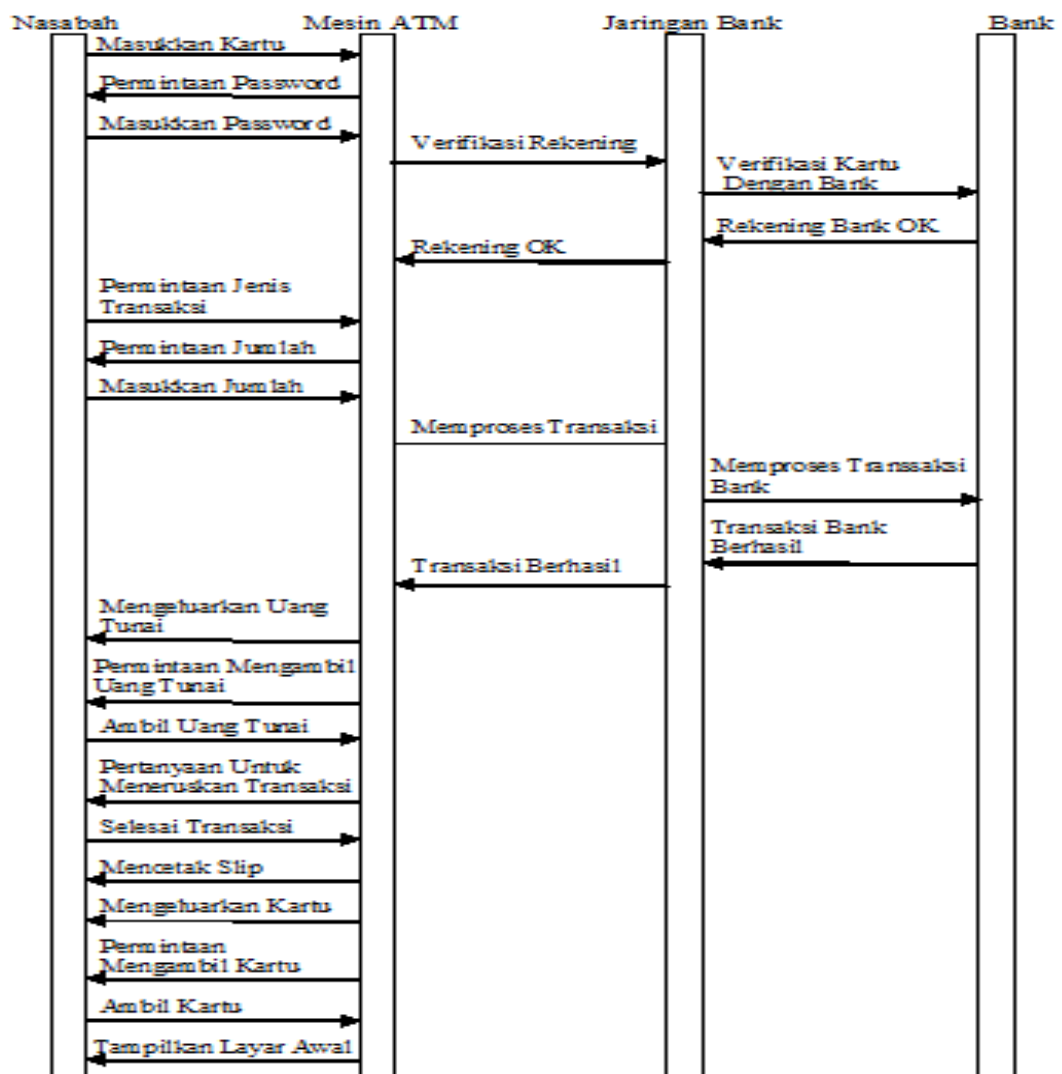
Activity diagrams adalah tehnik untuk mendeskripsikan *logika prosedural*, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagrams* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *Activity diagrams* bisa mendukung perilaku *pararel* sedangkan *flowchart* tidak bisa (Munawar ; 2005 : 103).



Gambar II.8. Contoh Activity Diagram
(Adi Nugroho ; 2009 : 11)

II.8.4. Sequence Diagram

sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical* (Munawar ; 2005 : 87).



Gambar II.9. Contoh *Sequence Diagram*
(Adi Nugroho ; 2009 : 36)

II.10. Tanaman Bunga Angrek

Angrek merupakan tanaman bunga hias berupa benalu benalu yang bunganya indah. Angrek sudah dikenal sejak 200 tahun yang lalu dan sejak 50 tahun terakhir mulai dibudidayakan secara luas di Indonesia.

Jenis angrek yang terdapat di Indonesia termasuk jenis yang indah antara lain: vanda tricolor terdapat di Jawa Barat dan di Kaliurang, *Vanda hookeriana*, berwarna ungu berbintik-bintik berasal dari Sumatera, angrek larat/*Dendrobium phalaenopsis*, angrek bulan/*phalaenopsis amabilis*, angrek Apple Blossom, angrek *Paphiopedilum praestans* yang berasal dari Papua serta angrek *Paphiopedilum glaucophyllum* yang berasal dari Jawa Tengah.

Berdasarkan pola pertumbuhannya, tanaman angrek dibedakan menjadi dua tipe, yaitu simpodial dan monopodial.

1. Angrek Tipe Simpodial

Angrek tipe simpodial adalah angrek yang tidak memiliki batang utama, bunga ke luar dari ujung batang dan berbunga kembali dari anak tanaman yang tumbuh. Kecuali pada angrek jenis *Dendrobium sp.* yang dapat mengeluarkan tangkai bunga baru di sisi-sisi batangnya.

2. Angrek Tipe Monopodial

Angrek tipe monopodial adalah angrek yang dicirikan oleh titik tumbuh yang terdapat di ujung batang, pertumbuhannya lurus ke atas pada satu batang. Bunga keluar dari sisi batang diantara dua ketiak daun.

Berdasarkan habitatnya, tanaman angrek dibedakan menjadi empat kelompok :

1. *Anggrek Epifit*

Anggrek epifit yaitu anggrek yang tumbuh menumpang pada pohon lain tanpa merugikan tanaman inangnya dan membutuhkan naungan dari cahaya matahari.

2. *Anggrek Terrestrial*

Anggrek terrestrial yaitu anggrek yang tumbuh di tanah dan membutuhkan cahaya matahari langsung. Tanaman anggrek terrestrial membutuhkan cahaya matahari 70-100%, dengan suhu siang berkisar antara 19-38⁰C, dan malam hari 18-21⁰C.

3. *Anggrek Litofit*

Anggrek litofit yaitu anggrek yang tumbuh pada batu-batuan, dan tahan terhadap cahaya matahari penuh.

4. *Anggrek Saprofit*

Anggrek saprofit yaitu anggrek yang tumbuh pada media yang mengandung humus atau daun-daun kering, serta membutuhkan sedikit cahaya matahari. (Sumber: Istiati,S.Pd, 2008:1-4)