

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi sistem pendukung keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi sistem pendukung keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.

Sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan) (Dicky Nofriansyah ; 2014 : 1)

II.2.1. Karakteristik Sistem Pendukung Keputusan

Berikut ini adalah uraian beberapa karakteristik SPK, yaitu:

1. Kapabilitas interaktif

Yaitu SPK memberi pengambil keputusan akses cepat ke data dan informasi yang dibutuhkan.

2. Fleksibilitas

Yaitu SPK dapat menunjang para manajer pembuat keputusan di berbagai bidang fungsional (keuangan, pemasaran, operasi produksi, dan lain-lain).

3. Kemampuan menginteraksi model

Yaitu SPK memungkinkan para pembuat keputusan berinteraksi dengan model-model, termasuk memanipulasi model- model tersebut sesuai dengan kebutuhan.

4. Fleksibilitas output

Yaitu SPK mendukung para pembuat keputusan dengan menyediakan berbagai macam output, termasuk kemampuan grafik menyeluruh atas pertanyaan- pertanyaan pengandaian .

II.2.2. Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan dibangun oleh lima komponen yaitu subsistem manajemen data, subsistem antarmuka pengguna, dan subsistem manajemen berbasis pengetahuan. Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama yaitu subsistem manajemen model, subsistem manajemen moden dan antarmuka pengguna. Sistem Pendukung Keputusan disusun dari beberapa subsistem yaitu :

1. Subsistem manajemen data

Basis data yang relevan untuk suatu situasi dan dikelola oleh *database management system* (DBMS). Subsistem ini dapat diinterkoneksi dengan data *warehouse* perusahaan yang relevan untuk pengambilan keputusan.

2. Subsistem manajemen model

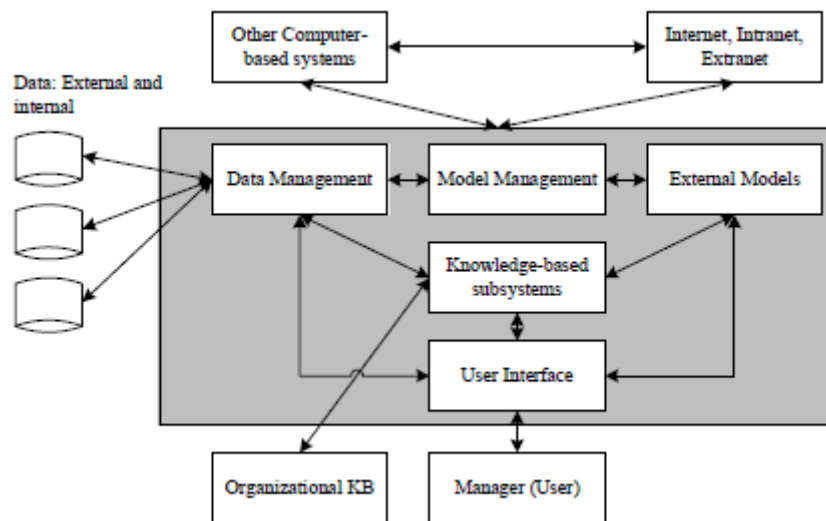
Merupakan paket perangkat lunak yang menyimpan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kemampuan analitik yang tepat. Perangkat lunak ini sering disebut *modelbase management system* (MBMS) dan dapat diimplementasikan pada sistem pengembangan web untuk berjalan pada *server* aplikasi.

3. Subsistem antarmuka pengguna

Merupakan dukungan komunikasi antara dengan pengguna. *Web browser* menjadi salah satu antarmuka yang menampilkan dalam bentuk grafis dan interaktif dengan pengguna.

4. Subsistem manajemen

Berbasis pengetahuan bertindak sebagai komponen independen yang memberikan kemampuan intelegensi untuk memperbesar pengetahuan pengambil keputusan. Perusahaan memiliki sistem manajemen pengetahuan. Keterhubungan subsistem ini dengan sistem pendukung keputusan dapat melalui interkoneksi dengan *web server* (Karismariyanti ; 2011 : 55).



Gambar II.1 Komponen Sistem Pendukung Keputusan
(Sumber : Karismariyanti ; 2011 : 55)

II.2.3. Tujuan Sistem Pendukung Keputusan

Tujuan yang ingin dicapai dalam SPK adalah :

1. Membantu dalam pengambilan keputusan atas masalah yang terstruktur.
2. Memberikan dukungan atas pertimbangan manager dan bukan dimaksudkan untuk mengganti fungsi manager.
3. Meningkatkan efektifitas keputusan yang diambil lebih dari pada perbaikan efesiensinya.
4. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat. (Dicky Nofriansyah ; 2014 : 1)

II.2.4. Manfaat Sistem Pendukung Keputusan

Manfaat yang dihasilkan dari SPK antara lain :

1. Memperbesar kemampuan pengambil keputusan untuk memproses informasi dan pengetahuan.
2. Memperbesar kemampuan pengambil keputusan dalam menangani permasalahan yang kompleks, berskala besar, dan menggunakan banyak waktu.
3. Memperpendek waktu pengambilan keputusan.
4. Mendorong pelaksanaan eksplorasi bagi pengambil keputusan.
5. Memberikan pendekatan baru dalam proses berpikir mengenai lingkup permasalahan dan konteks keputusan.
6. Membangkitkan bukti baru dalam mendukung sebuah keputusan atau konfirmasi dari asumsi yang sudah ada.
7. Menghasilkan keunggulan strategis dan kompetitif di dalam persaingan antar organisasi.
8. Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.

II.2.5. Proses Pengambilan Keputusan

Pengambilan keputusan meliputi beberapa tahap dan melalui beberapa proses. Pengambilan keputusan meliputi empat tahap yang saling berhubungan dan berurutan. Empat proses tersebut adalah :

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi dan menguji kelayakan solusi.

3. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan.

4. *Implementation*

Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana proses pengambilan keputusan.

II.3.1. Metode *Fuzzy*

Fuzzy secara bahasa diartikan sebagai kabur atau samar-samar. Suatu nilai dapat bernilai besar atau salah secara bersamaan. Dalam *fuzzy* dikenal derajat keanggotaan yang memiliki rentang nilai 0 (*no*) hingga 1 (*satu*). Berbeda dengan himpunan tegas yang memiliki nilai 1 atau 0 (*ya* atau *tidak*). Logika *Fuzzy* merupakan suatu logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antar benar atau salah. Dalam teori logika *fuzzy* suatu nilai bias

bernilai benar atau salah secara bersama. Namun berapa besar keberadaan dan kesalahan suatu tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* memiliki derajat keanggotaan dalam rentang 0 hingga 1. Berbeda dengan logika digital yang hanya memiliki nilai 1 atau 0. Logika *fuzzy* digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Dan logika *fuzzy* menunjukan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah. Tidak seperti logika klasik (*scrisp*)/ tegas, suatu nilai hanya mempunyai 2 kemungkinan yaitu merupakan suatu anggota himpunan atau tidak. Derajat keanggotaan 0 (*no!*) artinya nilai bukan merupakan anggota himpunana dan 1(satu) berarti nilai tersebut adalah anggota himpunan.

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output, mempunyai nilai kontinu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu yang dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama. Logika *fuzzy* memungkinkan nilai keanggotaan antar 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti “sedikit”, “lumayan” dan “sangat” Kelebihan dari teori logika *fuzzy* adalah kemampuan dalam proses penalaran secara bahasa (*linguistic reasoning*). Sehingga dalam perancangannya tidak memerlukan persamaan matematik dari objek yang akan dikendalikan.

II.3.2. Fuzzy Model Tahani

Model Tahani merupakan metode fuzzy yang masih menggunakan relasi database yang bersifat standar, dengan lebih menekankan penggunaan fuzzy pada beberapa field dalam tabel-tabel yang ada pada database dan pada perhitungan matematisnya [3].

1. Fungsi Keanggotaan

Apabila μ_S adalah fungsi keanggotaan suatu elemen pada himpunan S maka untuk suatu elemen X dapat dinyatakan $\mu_S(X)$ yang bernilai antara “0” dan “1” sehingga ada tiga kemungkinan :

$$\mu_S(X) = 1 \rightarrow X \text{ mutlak anggota } S.$$

$\mu_S(X) = 0 \rightarrow X$ mutlak bukan anggota S . $\mu_S(X) < 1 \rightarrow X$ anggota S dengan derajat keanggotaan antara 0 dan 1.

2. Fuzzyfikasi

Proses fuzzyfikasi dimulai dengan memasukkan inputan crisp lalu diikuti dengan memasukkan fungsi keanggotaan. Selanjutnya proses fuzzyfikasi yang menghasilkan inputan fuzzy.

3. Operasi Himpunan Fuzzy

Seperti pada himpunan konvensional, ada beberapa operasi yang didefinisikan untuk mengkombinasikan dan memodifikasi himpunan fuzzy. Sangat mungkin

digunakan operator dasar dalam proses query berupa operator AND dan OR (Janusz Kacprzyk, 1995). Dalam irisan (intersection) himpunan fuzzy digunakan operator AND, dinotasikan : $\mu A \cap B = \min(\mu A[x], \mu B[x])$.

4. Fuzzy Database

Fuzzyfikasi Query diasumsikan sebuah query konvensional (nonfuzzy), DBMS yang akan mencoba membuat dan menerapkan sebuah sistem dasar logika fuzzy query (fuzzy logic based querying system). Kelebihan query fuzzyfikasi yaitu dapat mencapai kelenturan (flexibility) dari DBMS, penanganan error otomatis, pencarian yang fleksibel, dan kesanggupan merespon kosong.

II.4. Toko Eiger

Bagi pecinta alam, tentu tak asing dengan nama *Eiger*. Sebuah produk peralatan *outdoor* dan tas yang banyak digemari pecinta alam maupun anak muda karena kualitas dan ketahanannya. Meski namanya *Eiger*, merek ini merupakan merek asli Indonesia. *Eiger* pertama kali diproduksi pada tahun 1993. Nama *Eiger* sendiri diambil dari nama Gunung *Eiger* di Swiss dan dicetuskan oleh pemilik *Eiger*, Ronny Lukito.

Eiger ditujukan untuk peralatan kegiatan *outdoor*, seperti mendaki gunung, kemah, panjat tebing dan aktifitas lainnya yang masih menyangkut masalah kegiatan luar. Saat pertama kali dibentuk, *Eiger* memulai dengan tas.

Membidik berbagai segmen pasar, Ronny-pun mengembangkan sayapnya dengan memasarkan merek *Eiger*, *Exsport*, *Neosack*, *Bodypack*, *Nordwand*, *Morphosa*, *World Series*, *Extrem*, *Vertic*, *Domus Danica* serta *Brooklyn*.

Tak berhenti di situ, sekarang perusahaan *Ronny* juga sudah memproduksi jenis lain seperti dompet, sarung *handphone* dan berbagai jenis produk lain.[1]

II.5. PHP

PHP merupakan singkatan dari "*Hypertext Preprocessor*" dan PHP adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaksnya mirip dengan bahasa pemrograman C, Java, ASP dan Perl, di tambah beberapa fungsi PHP yang spesifik dan mudah dimengerti.

PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP anda bias menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di *include* atau *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa database walaupun dengan kelengkapan yang berbeda, yaitu seperti: DBM, FilePro (Personic, Inc), Informix, Ingres, InterBase, Microsoft Access, MSQl, Oracle, PostgrSQL, dan Sybase.

Dari uraian di atas maka dapat diambil 4 point utama tentang PHP:

1. PHP adalah singkatan dari : *Hypertext Preprocessor*
2. PHP adalah bahasa scripting server-side, artinya di jalankan di server, kemudian outputnya di kirim ke client (browser).
3. PHP di gunakan untuk membuat aplikasi web
4. PHP mendukung banyak database (MSQl, Informix, Oracle, Sybase, Solid, PostgresSQL, Generic ODBC, dll) (Madcoms;2013:310).

II.6. *My Struktur Query Language (MySQL)*

MySQL adalah salah satu program yang dapat digunakan sebagai database dan merupakan salah satu software untuk database server yang banyak digunakan. MySQL bersifat Open Source dan menggunakan SQL. MySQL bisa dijalankan diberbagai platform misalnya Windows, Linux, dan lain sebagainya.

Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL memiliki beberapa kelebihan , antara lain :

1. MySQL dapat digunakan oleh berbagai user dalam waktu yang bersamaan tanpa mengalami masalah.
2. MySQL memiliki kecepatanyang bagus dalam menangani query sederhana.
3. MySQL memiliki operator dan fungsi secara penuh dan mendukung perintal Select dan Where dalam perintah query.
4. MySQL memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perijinan yang mendetail serta sandi terenkripsi.
5. MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
6. MySQL dapat berjalan setabil pada berbagai sistem informasi seperti Windows, Linux, FreeBSD, Mac Os Server, Solaris, dan masih bnayak lagi.
7. MySQL didistribusikan secara gratis (Madcoms;2013:310).

II.7. XAMPP

XAMPP merupakan paket PHP dan MYSQL berbasis open source yang mampu membuat web dinamis. XAMPP aplikasi webserver yang mendukung instalasi Linux dan Windows selain itu XAMPP juga termasuk aplikasi open source. XAMPP aplikasi yang terdiri dari beberapa paket aplikasi server seperti Apache Http Server, MYSQL Database Server, Filezilla FTP Server XAMPP.[2]

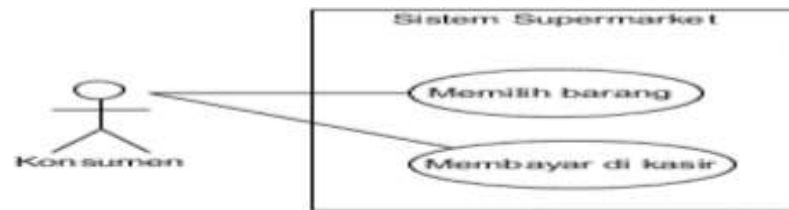
II.8. UML (*Unified Modeling Language*)

Menurut Arif Rachman (2012), Pemecahan masalah utama dari *Object Oriented* biasanya dengan penggambaran dalam bentuk model. Model abstrak (semu) merupakan gambaran detail dari inti masalah yang ada, umumnya sama seperti refleksi dari problem yang ada pada kenyataan. Beberapa modeling tool yang dipakai adalah bagian dari dasar UML, kependekan dari *United Modeling Language*. UML terdiri atas beberapa diagram, yaitu :

1.Diagram *Use Case*

Diagram *Use Case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu *apa yang dilakukan* bukan *bagaimana melakukannya*. Diagram *Use Case* dekat kaitannya dengan kejadian-kejadian. Kejadian (*scenario*) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. untuk lebih memperjelas lihat gambaran suatu peristiwa untuk sebuah klinik kesehatan di bawah ini :

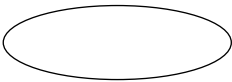
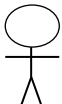
“Konsumen memasuki toko dan memastikan barang yang akan di beli, kemudian membuka sistem untuk proses pemilihan barang/tas yang direkomendasikan sesuai dengan kriteria yang dicari oleh konsumen. Setelah memutuskan pilihan maka konsumen akan membayar barang/tas yang sudah di pilih.”

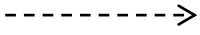
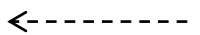


Gambar II.2. Usecase Diagram
(Sumber : Arif Rachman ; 2012 : 2)

Berikut adalah simbol-simbol yang ada pada diagram *use case* dapat dilihat pada tabel II.1 sebagai berikut:

Tabel II.1. 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 : Arif Rachman ; 2012 : 3)

2. Diagram Class

Diagram *Class* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram *Class* bersifat statis : menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan.

Multiplicity dari suatu titik *association* adalah angka kemungkinan bagian dari hubungan kelas dengan single *instance* (bagian) pada titik yang lain. *Multiplicity* berupa single number (angka tunggal) atau range number (angka batasan). Pada contoh, hanya bisa satu

'*Customer*' untuk setiap '*Order*', tapi satu '*Customer*' hanya bisa memiliki beberapa '*Order*'. Tabel II.2 di bawah mengenai *multiplicity* yang sering digunakan :

Tabel II.2 *Multiplicity*

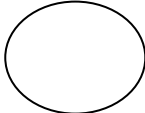
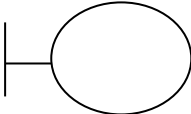
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

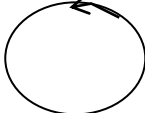
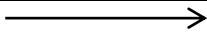
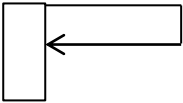
(Sumber : Arif Rachman ; 2012 : 3)

3. *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* dapat dilihat pada tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

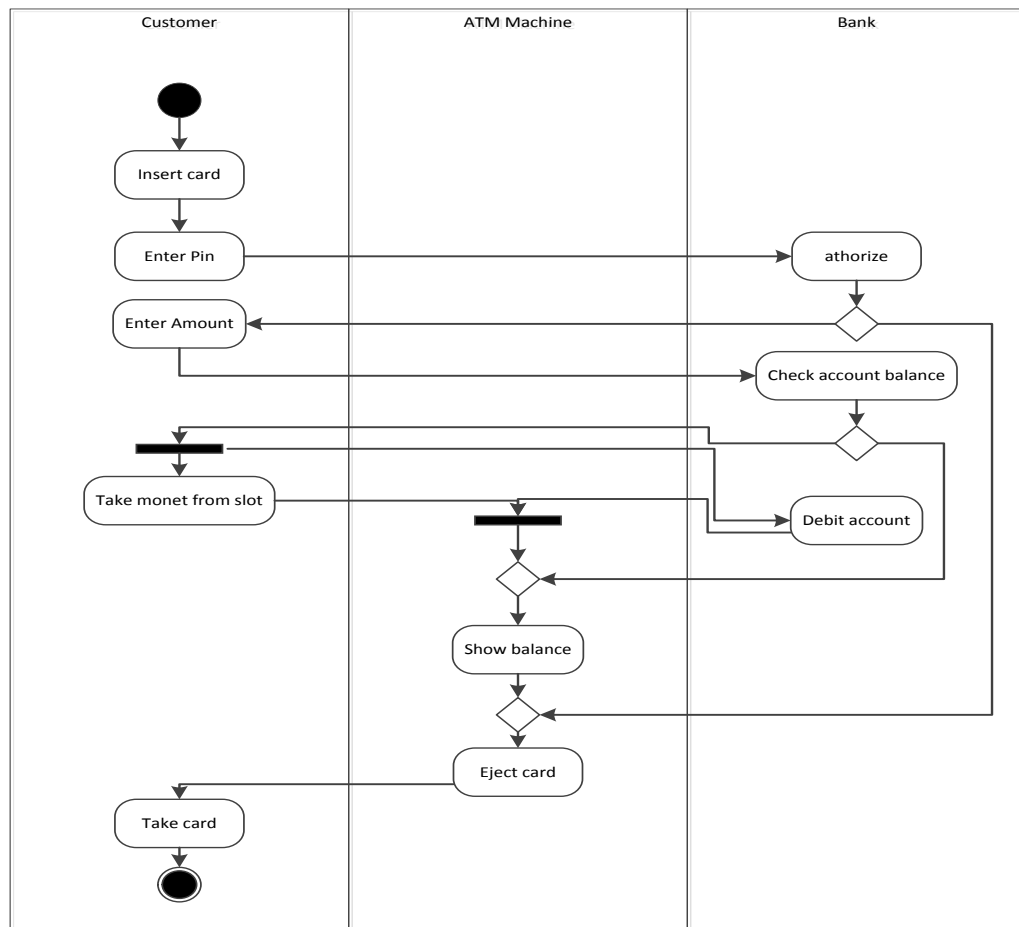
Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara 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 : Arif Rachman ; 2012 : 3)

4. Diagram Activity

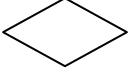
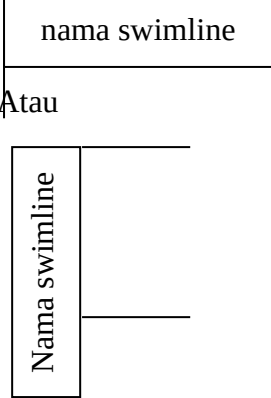
Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *Statechart*. Diagram *Statechart* berfokus pada *obyek yang dalam suatu proses* (atau proses menjadi suatu obyek), diagram *Activity* berfokus pada *aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal*. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.



Gambar II.3. Diagram Activity
(Sumber : Arif Rachman ; 2012 : 9)

Tabel II.4. Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Pengabungan / <i>join</i> 	Asosiasi pengabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang digunakan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimline 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Arif Rachman ; 2012 : 3)

II.9. 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.9.1. 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 (1st 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).

