

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Beti Novianti, 2016 mengenai Implementasi data mining dengan algoritma c45 untuk penjurusan siswa, hasil penelitian Beti Novianti adalah dengan adanya aplikasi implementasi data mining dengan algoritma c45 maka dapat dengan mudah mengetahui minat siswa dalam memilih jurusan.

Berdasarkan penelitian yang dilakukan oleh Nurul Azwanti, 2018 mengenai Analisa algoritma c45 untuk memprediksi penjualan motor pada PT capela dinamik nusatra, hasil penelitian Nurul Azwanti adalah dengan adanya aplikasi algoritma c45 untuk memprediksi penjualan motor maka PT capela dinamik nusantara dapat dengan mudah mengetahui prediksi penjualan motor.

Berdasarkan penelitian yang dilakukan oleh Fitriana Harahap, 2015 mengenai Penerapan Data Mining dalam Memprediksi Pembelian cat, hasil penelitian Fitriana Harahap adalah dengan adanya penerapan data mining maka dapat dengan mudah memprediksi pembelian cat.

II.2. Landasan Teori

Berikut ini adalah beberapa landasan teori yang dapat digunakan sebagai referensi untuk penelitian ini :

II.2.1. *Data Mining*

Data mining adalah penambangan atau penemuan informasi baru dengan mencari pola atau aturan dengan mencari pola atau aturan tertentu dari sejumlah data yang besar. *Data mining* didefinisikan sebagai proses menemukan pola-pola dalam data. Proses ini otomatis atau seringnya semiotomatis. Pola yang ditemukan harus penuh arti dan pola tersebut memberikan keuntungan, biasanya keuntungan secara ekonomi. Data yang dibutuhkan dalam jumlah besar. *Data mining*, sering juga disebut sebagai *knowledge discovery in database* (KDD) adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar (Heroe Santoso, dkk, 2016).

Data mining adalah proses ekstraksi informasi dari kumpulan data melalui penggunaan algoritma dan teknik yang melibatkan bidang ilmu statistik, mesin pembelajaran, dan sistem manajemen *database*. Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menentukan aturan asosiasi antara suatu kombinasi *item* (Khairul, 2015).

II.2.2. Karakteristik *Data Mining*

1. *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
2. *Data mining* biasa menggunakan data yang sangat besar.
3. *Data mining* berguna untuk membuat keputusan yang kritis, terutama dalam strategi. (Khairul, 2015).

II.2.3. Tahap-Tahap *Data Mining*

1. Pembersihan data (*data cleaning*)

Merupakan proses menghilangkan *noise* dan data yang tidak konsisten.

2. Integrasi data (*data integration*)

Merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru.

3. Seleksi data (*data selection*)

Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*.

4. *Data transformation*

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*

5. Proses *mining*.

Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

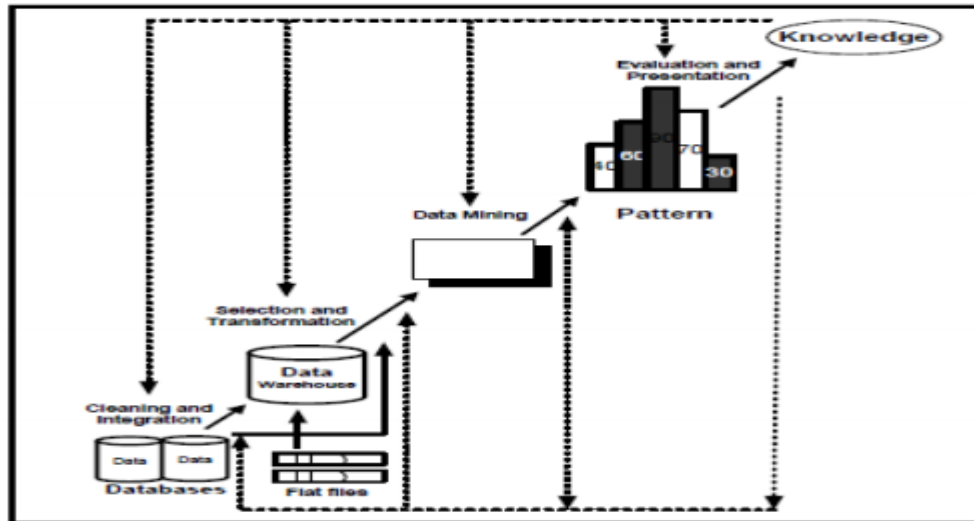
6. Evaluasi pola (*pattern evaluation*).

Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan bermanfaat.

7. Presentasi pengetahuan.

Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan

hasil *data mining*. (Khairul, 2015).



Gambar II.1. Tahapan *Data Mining*

(Sumber : Heroe Santoso, dkk ; 2016)

II.2.4. Teknik Data Mining

1. *Association Rules*

Association rules (aturan asosiasi) atau *affinity analysis* (analisis afinita) berkenaan dengan studi tentang “apa bersama apa”. Aturan asosiasi juga sering dinamakan *market basket analysis* (analisis keranjang belanja), aturan asosiasi ingin memberikan informasi dalam bentuk hubungan “*if-then*” atau “jika-maka”. Aturan ini dihitung dari data yang sifatnya probabilitas.

2. *Clustering*

Clustering termasuk metode yang sudah cukup dikenal dan banyak dipakai dalam data mining. Sampai sekarang para ilmuwan dalam bidang *datamining* masih melakukan berbagai usaha untuk melakukan perbaikan model clustering karena metode yang dikembangkan sekarang

masih bersifat heuristik. Usaha-usaha untuk menghitung jumlah *cluster* yang optimal dan pengklasteran yang paling baik masih terus dilakukan. Dengan demikian menggunakan metode yang sekarang, tidak bisa menjamin hasil pengklasteran sudah merupakan hasil yang optimal. Namun, hasil yang dicapai biasanya sudah cukup bagus dari segi praktis.

3. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari data untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menentukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungansering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

4. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variable target estimasi lebih kearah numerik dari pada kearah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variable

prediksi dalam proses pembelajaran akan menghasilkan model estimasi.

Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya

5. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan datang dimasa mendatang. Sebagai contoh prediksi harga beras dalam tiga bulan yang akan datang, serta contoh lain seperti contoh prediksi presentasi kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

6. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. (Khairul, 2015).

II.2.5. *Algoritma C4.5*

Algoritma C4.5 merupakan *algoritma* yang digunakan untuk membentuk pohon keputusan. Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Dan mereka juga dapat diekspresikan dalam bentuk bahasa basis data seperti *Structured Query Language* untuk mencari *record* pada kategori tertentu.

Cara *algoritma C4.5* untuk membangun pohon keputusan yaitu:

1. Pilih atribut yang akan digunakan sebagai akar
2. Buatlah sebuah cabang untuk setiap nilai
3. Bagilah kasus dalam sebuah cabang
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Ada beberapa tahap dalam membuat sebuah pohon keputusan dengan *algoritma C4.5*.

1. Menyiapkan data training. Data ini diambil dari data yang sudah pernah ada sebelumnya dan sudah dikelompokkan kedalam kelas tertentu.
2. Setelah itu tentukan akar dari pohon. Pilih akar dari atribut, cara adalah dengan menghitung nilai gain dari semua atribut, yang menjadi akar pertama adalah nilai gain yang paling. Sebelum menentukan nilai gain, terlebih dahulu hitung nilai entropy. Untuk menentukan nilai entropy gunakan rumus :

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i$$

Di mana :

S = himpunan kasus

n = jumlah partisi S

p_i = proporsi S_i terhadap S

3. Setelah itu tentukan nilai gain menggunakan rumus:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i)$$

Keterangan :

S = Himpunan Kasus

A = Atribut

n = Jumlah Partisi Atribut A

| S_i | = Jumlah Kasus pada partisi ke-i

| S | = Jumlah Kasus dalam S

4. Setelah itu ulangilah langkah ke-2 sampai semua record terpartisi secara sempurna.
5. Proses partisi pohon keputusan akan berhenti saat :
 - a. Semua record dalam simpul N mendapat kelas yang sama.
 - b. Tidak ada atribut di dalam record yang dipartisi lagi.
- c. Tidak ada record di dalam cabang yang kosong. (Fitriana Harahap, 2015)

II.2.6. Basis Data (*Database*)

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi. *Database* mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah :

1. Redundansi dan Inkonsistensi data.
2. Kesulitan dalam pengaksesan data.
3. Isolasi data untuk standarisasi.
4. *Multi user*.
5. Keamanan data.

6. Integritas data.

7. Kebebasan data. (Urva dan Siregar, 2015 : 93).

II.2.7. Normalisasi

Normalisasi merupakan parameter digunakan untuk menghindari duplikasi terhadap tabel dalam basis data dan juga merupakan proses dekomposisi sebuah tabel yang masih memiliki beberapa anomali atau ketidak wajaran sehingga menghasilkan tabel yang lebih sederhana dan struktur yang bagus, yaitu sebuah tabel yang tidak memiliki *data redundancy* dan memungkinkan *user* untuk melakukan *insert*, *delete*, dan *update* pada baris (*record*) tanpa menyebabkan inkonsistensi data (Triyono, 2015 : 19).

1. First Normal Form (1 NF)

Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa atribut atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal. Atribut *multivalued*, *composite*, *derive* tidak tunggal. Setiap nilai dari atribut hanya mempunyai nilai tunggal.

Tabel II.1. Bentuk Normal Pertama (1NF)

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga	Prediksi
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah	Laris

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah

2. Second Normal Form (2 NF)

Untuk menjadikan tabel normal tingkat ke 2 maka sudah 1NF dan setiap atribut yang bukan *primary key* sepenuhnya secara *funksional* tergantung pada semua atribut pembentuk *primary key*.

Tabel II.2. Bentuk Normal Kedua (2NF)

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga	Prediksi
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah	Laris

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah

Kode	Nama Produk	Keterangan
P1	Ainiqua 330 MI	Minuman

3. Third Normal Form (3 NF)

Tabel sudah 2NF dan tidak memiliki *transitive dependencies*, *Transitive dependency* adalah ketika ada atribut yang secara tidak langsung tergantung pada *primary key* dan atribut tersebut juga tergantung pada atribut lain yang bukan *primary key*.

Tabel II.3. Bentuk Normal Ketiga (3NF)

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga	Prediksi
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah	Laris

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah

Kode	Nama Produk	Keterangan
P1	Ainiqua 330 MI	Minuman

Kode	Banyak Terjual	Range
K1	Banyak	500-800

4. Boyce-codd Normal Form (BCNF)

Tabel dalam BCNF jika sudah 3NF dan semua *determinants* adalah *candidate keys*. Perbedaan 3NF dan BCNF adalah untuk *functional dependency* $A \rightarrow B$, 3NF memperbolehkan ketergantungan ada dalam relasi jika B adalah *Primary Key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menuntut untuk ketergantungan tetap ada dalam relasi, A harus menjadi *candidate key*.

Tabel II.4. Bentuk BCNF

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga	Prediksi
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah	Laris

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah

Kode	Nama Produk	Keterangan
P1	Ainiqua 330 MI	Minuman

Kode	Banyak Terjual	Range
K1	Banyak	500-800

Kode	Berkualitas	Keterangan
K2	Tinggi	Banyak Terjual Dan Peminat Konsumen

5. Fourth Normal Form (4 NF)

Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalued dependency*.

Tabel II.5. Bentuk Normal Keempat (4NF)

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga	Prediksi
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah	Laris

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah

Kode	Nama Produk	Keterangan
P1	Ainiqua 330 MI	Minuman

Kode	Banyak Terjual	Range
K1	Banyak	500-800

Kode	Berkualitas	Keterangan
K2	Tinggi	Banyak Terjual Dan Peminat Konsumen

6. Fifth Normal Form (5 NF)

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika tabel tersebut dapat dipecah atau diproyeksikan menjadi beberapa tabel dan dari proyeksi-proyeksi itu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula. Jika penyusunan ini tidak mungkin dilakukan dikatakan pada relasi itu terdapat *join dependencies* dan dikatakan bersifat *lossy join* (Triyono, 2015 : 20).

Tabel II.6. Bentuk Normal Kelima (5NF)

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga	Prediksi
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah	Laris

Kode	Kode Produk	Nama Produk	Bulan	Tahun	Banyak Terjual	Berkualitas	Harga
1	P1	Ainiqua 330 MI	Januari	2019	253	Ya	Murah

Kode	Nama Produk	Keterangan
P1	Ainiqua 330 MI	Minuman

Kode	Banyak Terjual	Range
K1	Banyak	500-800

Kode	Berkualitas	Keterangan
K2	Tinggi	Banyak Terjual Dan Peminat Konsumen

Kode	Harga	Range
K3	Murah	< 5.000

II.2.8. *Unified Modeling Language (UML)*

Menurut Windu Gata (2015) 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

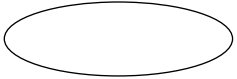
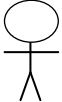
bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Urva dan Siregar, 2015 : 93).


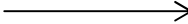
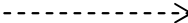
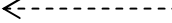
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

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 dapat dilihat pada tabel II.1 dibawah ini :

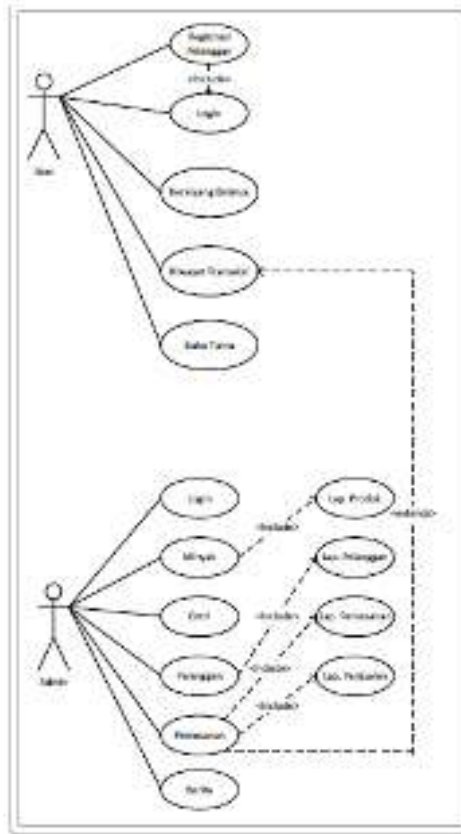
Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>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.</p>

	Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki <i>control</i> 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 mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan 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:Urva dan Siregar, 2015 : 94)

Contoh *Use Case Diagram* :





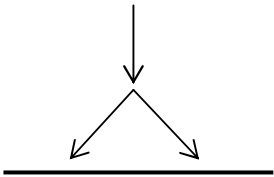
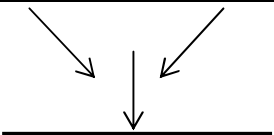
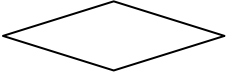
Gambar II.2. Contoh Penggunaan *Use Case Diagram*

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* dapat dilihat pada tabel II.2 dibawah ini :

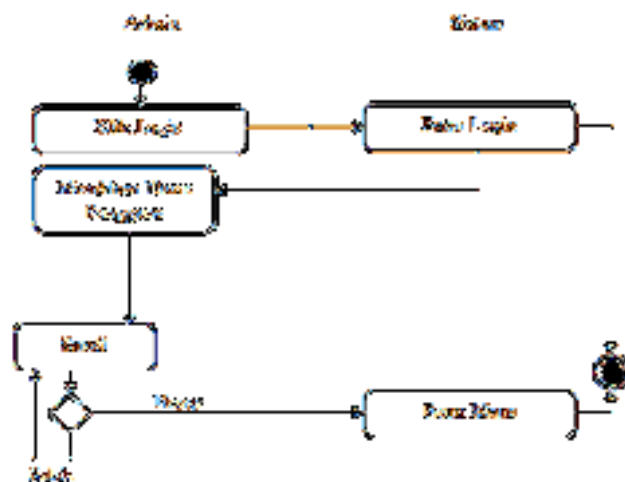
Tabel II.2. Simbol *Activity Diagram*

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 parallel atau untuk menggabungkan dua kegiatan pararel 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> .
New Swimline	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

(Sumber : Urva dan Siregar, 2015 : 94)

Contoh *Activity Diagram* :

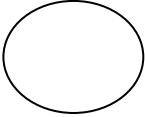
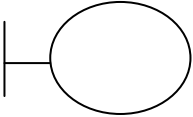
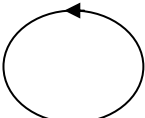

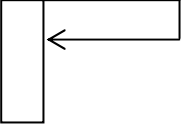




Gambar II.3. Contoh Penggunaan *Activity Diagram*

3. 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* dapat dilihat pada tabel II.3 dibawah ini :

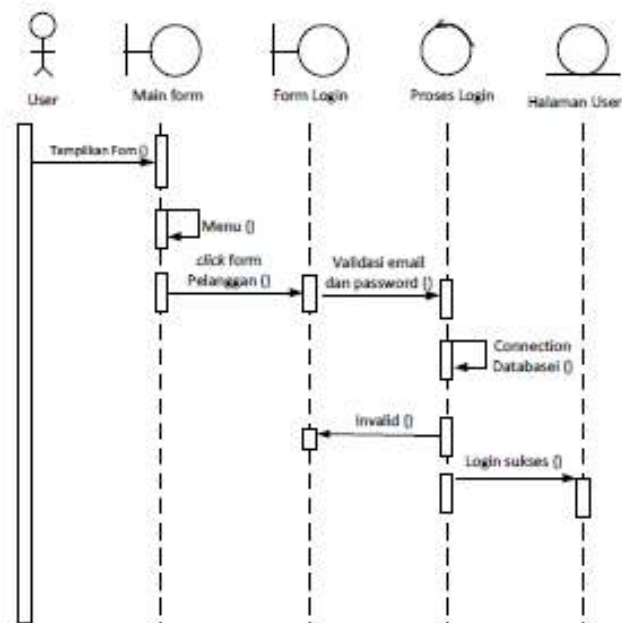
Tabel II.3. Simbol *Sequence Diagram*

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.

	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Urva dan Siregar, 2015 : 95)

Contoh *Sequence Diagram* :



Gambar II.4. Contoh Penggunaan *Sequence Diagram*

4. *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

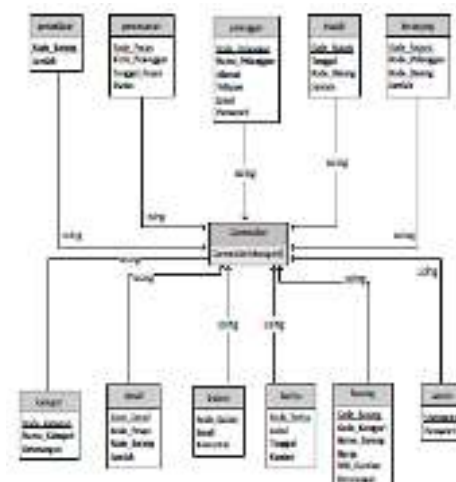
meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini:

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 : Urva dan Siregar, 2015 : 95)

Contoh *Class Diagram* :



Gambar II.5. Contoh Penggunaan Class Diagram