

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Mahasiswa**

Mahasiswa adalah seseorang yang sedang dalam proses menimba ilmu ataupun belajar dan terdaftar sedang menjalani pendidikan pada salah satu bentuk perguruan tinggi yang terdiri dari akademik, politeknik, sekolah tinggi, institut dan universitas (Hartaji, 2012: 5). Dalam Kamus Bahasa Indonesia (KBI), mahasiswa didefinisikan sebagai orang yang belajar di Perguruan Tinggi (Kamus Bahasa Indonesia Online, [kbbi.web.id](http://kbbi.web.id))

Menurut Siswoyo (2007: 121) mahasiswa dapat didefinisikan sebagai individu yang sedang menuntut ilmu ditingkat perguruan tinggi, baik negeri maupun swasta atau lembaga lain yang setingkat dengan perguruan tinggi. Mahasiswa dinilai memiliki tingkat intelektualitas yang tinggi, kecerdasan dalam berpikir dan perencanaan dalam bertindak. Berpikir kritis dan bertindak dengan cepat dan tepat merupakan sifat yang cenderung melekat pada diri setiap mahasiswa, yang merupakan prinsip yang saling melengkapi.

#### **2.2 Prestasi**

Prestasi siswa dapat diartikan hasil yang diperoleh karena adanya aktivitas belajar yang telah dilakukan. Prestasi belajar merupakan hal yang tidak dapat dipisahkan dari kegiatan belajar karena kegiatan belajar merupakan proses sedangkan prestasi merupakan hasil dari proses belajar mengajar. Menurut Hadari Nawawi (1998 :100) Prestasi belajar adalah tingkatan keberhasilan dalam mempelajari materi pelajaran disekolah yang dinyatakan dalam bentuk skor yang diperoleh dari hasil tes.

### 2.3 Data Mining

Menurut (Hermawati, 2013) *Data mining* adalah proses yang mempekerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis. Definisi lain diantaranya adalah pembelajaran berbasis induksi (*induction-based learning*) adalah proses pembentukan definisi-definisi konsep umum yang dilakukan dengan mengobservasi contoh-contoh spesifik dari konsep-konsep yang akan dipelajari. *Knowledge Discovery in Databases* (KDD) adalah penerapan metode *saintifik* (untuk mengidentifikasi atau menemukan masalah) pada *data mining*. Dalam konteks ini *data mining* merupakan satu langkah dari proses KDD.

Secara garis besar tehnik-tehnik data mining adalah sebagai berikut :

a. Klasifikasi (*classification*)

Klasifikasi adalah menentukan sebuah *record* data baru ke salah satu dari beberapa kategori (atau klas) yang telah didefinisikan sebelumnya.

b. Regresi

Memprediksi nilai dari suatu variabel kontinyu yang diberikan berdasarkan nilai dari variabel yang lain, dengan mengasumsikan sebuah model ketergantungan linier atau nonlinier.

c. Klasterisasi (*clustering*)

Mempartisi data-set menjadi beberapa sub-set atau kelompok sedemikian rupa sehingga elemen-elemen dari suatu kelompok tertentu memiliki *set property* yang *dishare* bersama, dengan tingkat similaritas yang tinggi dalam satu kelompok dan tingkat similaritas antar kelompok yang rendah.

d. Kaidah asosiasi (*association rules*)

Mendeteksi kumpulan atribut-atribut yang muncul bersama (*co-occur*) dalam frekuensi yang sering, dan membentuk sejumlah kaidah dari kumpulan-kumpulan tersebut.

e. Pencarian pola sekuensial (*sequence mining*)

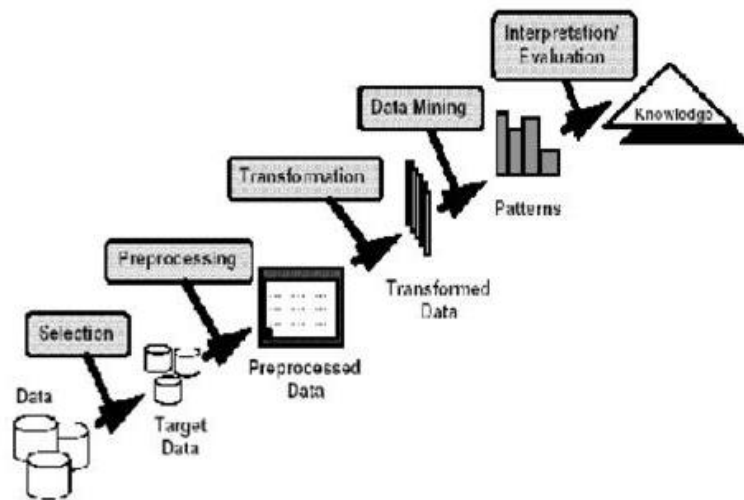
Mencari sejumlah *event* yang secara umum terjadi bersama-sama.

### 2.3.1 Operasi Data Mining

Operasi data mining menurut sifatnya dibedakan menjadi dua, yaitu:

- Prediksi (*prediction driven*) untuk menjawab pertanyaan apa dan sesuatu yang bersifat transparan. Operasi prediksi digunakan untuk validasi hipotesis, querying dan pelaporan.
- Penemuan (*discovery driven*) bersifat transparan dan untuk menjawab pertanyaan “mengapa?”. Operasi penemuan digunakan untuk analisis data eksplorasi, pemodelan prediktif, segmentasi database, analisis keterkaitan (*link analysis*) dan deteksi deviasi.

Adapun tahapan proses dalam penggunaan data mining dapat dilihat pada gambar 2.1 proses KDD.



Gambar 2.1 Proses KDD

Tahapan proses dalam penggunaan *data mining* yang merupakan proses *Knowledge Discovery in Database* (KDD) sseperti yang terlihat pada gambar 2.1 dapat diuraikan sebagai berikut :

- Memahami domain aplikasi untuk mengetahui dan menggali pengetahuan awal serta apa sasaran pengguna.
- Membuat target data-set yang meliputi pemilihan data dan fokus pada sub-set data.

- c. Pembersihan dan transformasi data meliputi eliminasi drau, *outliers*, *missing value* serta pemilihan fitur dan reduksi dimensi.
- d. Penggunaan algoritma *data mining* yang terdiri dari asosiasi, sekuensial, klasifikasi, dan lain-lain.
- e. Interpretasi, evaluasi dan visualisasi pola untuk melihat apakah ada sesuatu baru dan menarik dan dilakukan iterasi jika diperlukan.

#### **2.4 Algoritma *K-Means***

Menurut (Santoso, 2007) dari beberapa tehnik klastering yang paling sederhana dan umum dikenal adalah klastering k-means. Dalam tehnik ini kita ingin mengelompokkan objek kedalam cluster. Algoritma K-Means yaitu :

1. Tentukan jumlah cluster
2. Tentukan centroid awal dengan cara mengambil hasil nilai rata-rata terbesar sedang dan terkecil dari data.
3. Alokasikan data dengan jumlah cluster yang ditentukan berdasarkan jarak terdekat dengan centroid menggunakan rumus Ecludien distance space. Jarak terdekat akan masuk dalam hasil iterasi dimana diperoleh dari nilai terkecil diantara cluster.
4. Setelah cluster dan anggotanya terbentuk, hitung mean tiap cluster dan jadikan sebagai centroid baru.
5. Tugaskan lagi setiap objek dengan memakai pusat cluster yang baru. Jika pusat cluster sudah tidak berubah lagi, maka proses pengklasteran selesai. Atau kembali lagi ke langkah yang ketiga sampai pusat klaster tidak berubah lagi.

### Rumus Rumus Jarak Euklidean

$$d(x,y) = |x - y| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

$d(x,y)$  = Jarak objek antara  $x_i$  dan  $y_i$

$n$  = Dimensi data

$x_i$  = Koordinat dari objek  $x_i$  pada dimensi  $i$  (pada data mahasiswa)

$y_i$  = koordinat dari objek  $y_i$  pada dimensi  $i$  (pada data centroid)

## 2.5 Bahasa Pemrograman Yang Digunakan

Komputer membutuhkan *software* untuk beroperasi dan membutuhkan sistem operasi atau program-program untuk membuat komponen-komponen komputer bekerja secara baik. Merupakan perangkat yang dapat dilihat oleh mata, tetapi tidak dapat diraba. *Software* juga sering digunakan untuk menunjukkan semua program yang dapat dipakai dalam sistem komputer. Dalam pengertian yang sempit, istilah ini menunjuk pada sebuah program yang dapat mempermudah pemakai dari berbagai jenis komputer untuk mendayagunakan *hardware* dengan baik. Untuk merancang dan membangun sistem ini pembuat membutuhkan *software-software* penunjang untuk memaksimalkannya antara lain :

### 2.5.1 HTML 5 (*Hypertext Markup Language 5*)

Menurut (Sianipar, 2015) HTML5 adalah sebuah markup untuk menstrukturkan dan menampilkan isi dari halaman web. HTML (yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997) dan hingga bulan juni 2011 tetap dalam proses pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan salah satu karya *World Wide Web Consortium*, W3C untuk mendefinisikan sebuah bahasa markup tunggal yang dapat ditulis dengan HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan

terpisah. Dan di implementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat web.

### **2.5.2 PHP (*Hypertext Preprocessor*)**

Menurut (Ardhana, 2012) PHP *Hypertext Preprocessor* atau sering disebut php merupakan bahasa pemrograman berbasis server-side yang dapat melakukan parsing *script* php menjadi *script* web sehingga dari sisi *client* menghasilkan suatu tampilan yang menarik. Php merupakan pengembangan FI atau *Form Interface* yang dibuat oleh rasmus lerdoff pada tahun 1995.

Berbeda dengan HTML, kode PHP tidak diberikan secara langsung oleh server ketika ada permintaan atau request dari sisi client namun dengan cara pemrosesan dari server. Kode PHP disisipkan pada HTML. Perbedaan dari kode (*script*) HTML dan PHP yaitu setiap kode PHP ditulis selalu diberi tag pembuka yaitu `<?php` dan pada akhir kode PHP diberi tag penutup yaitu `?>`. Kelebihan dan kekurangan PHP sebagai berikut :

- a. PHP menjadi populer karena kesederhanaannya dan kemampuannya dalam menghasilkan berbagai aplikasi web.
- b. Tidak detail untuk pengembangan skala besar.
- c. PHP adalah salah satu bahasa *server-side* yang di desain khusus untuk aplikasi web.
- d. Tidak memiliki sistem pemrograman berorientasi objek sesungguhnya.
- e. PHP termasuk dalam *open source product*.

### **2.5.3 Adobe Dreamweaver CS6**

Menurut (Madcoms, 2013) menyatakan *Adobe Dreamweaver CS6* adalah perangkat lunak terkemuka untuk desain web yang menyediaka visual yang intuitif termasuk pada tingkat kode yang dapat digunakan untuk membuat

dan mengedit website HTML, serta aplikasi mobile seperti smartphone, tablet dan perangkat lainnya.

#### **2.5.4 MySQL**

Menurut (Rudianto, 2011) MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB yang pada saat itu bernama TcX data consult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan mengembangkan aplikasi web untuk klien. Tcx merupakan perusahaan software dan konsultan *database*. Saat ini MySQL sudah diakui oleh Oracle Crop. MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang mana database sebagai sumber dan pengolahan datanya.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* sehingga mudah untuk digunakan. MySQL juga bersifat *open source* dan *free* pada berbagai *platform* kecuali pada *windows* bersifat *shareware*. MySQL didistribusikan dengan *lisensi open source GPL (General Public License)* mulai versi 3.23, pada bulan juni 2000.


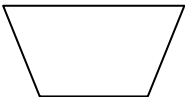
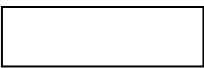
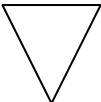

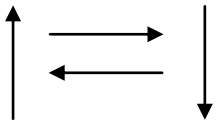
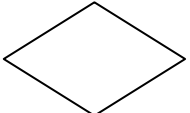
### **2.6 Diagram Flowchart**

Menurut (Jogiyanto, 2005) menjelaskan bahwa bagan alir merupakan alat pengembangan sistem yang akan membantu menggambarkan sistem yang sedang berjalan pada perusahaan. Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.




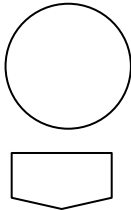
### **2.7 Bagan Alir Dokumen (*Document Flowchart*)**

Bagan alir dokumen (*document flowchart*) atau disebut dengan bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Adapun simbol bagan alir dapat dilihat pada tabel 2.1 sebagai berikut :

Tabel 2.1 Diagram Alir Dokumen (*document flowchart*)

SIMBOL	KETERANGAN
Dokumen 	Menunjukkan dokumen yang digunakan untuk input dan output baik secara manual maupun komputerisasi
Proses Manual 	Menunjukkan pekerjaan yang dilakukan secara manual.
Proses komputerisasi 	Menunjukkan proses dari operasi program komputer.
Simpanan 	Menunjukkan arsip atau Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
Terminator 	Menyatakan permulaan atau akhir suatu program
Garis alir 	Menyatakan jalannya arus suatu proses
Decision 	Digunakan untuk suatu penyeleksian kondisi di dalam program.




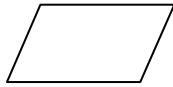

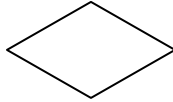

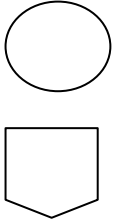


<p>Keyboard</p> 	<p>Memasukkan data secara manual dengan menggunakan keyboard</p>
<p>Hard disk</p> 	<p>Media penyimpanan, menggunakan perangkat hard disk.</p>
<p>Keterangan</p> 	<p>Digunakan untuk memberikan keterangan yang lainnya.</p>
<p>Penghubung</p> 	<p>Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda</p>

## 2.8 Bagan Alir Program (*Program Flowchart*)

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program computer terperinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah didalam program komputer secara logika, sedangkan bagan alir program komputer terperinci digunakan untuk menggambarkan instruksi-instruksi program komputer secara terperinci. Bagan alir logika program dipersiapkan oleh analis sistem. Sedangkan bagan alir program komputer terperinci dipersiapkan pemrogram. Berdasarkan penjelasan diatas penulis dapat menarik kesimpulan bahwa bagan alir (*flowchart*) adalah suatu gambaran umum tentang sistem yang berjalan yang berfungsi sebagai alat bantu komunikasi dan untuk dokumentasi serta menyajikan kegiatan baik secara manual maupun komputerisasi. Adapun symbol bagan alir program dapat dilihat pada tabel 2.2 sebagai berikut

Tabel 2.2 Bagan Alir Program (*Program Flowchart*)

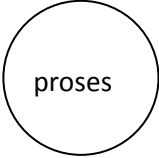



Simbol	Keterangan
Terminator 	Digunakan untuk memberikan awal dan akhir suatu proses.
Proses 	Menunjukkan proses dari operasi program komputer.
<i>Preparation</i> 	Proses inisialisasi/pemberian harga awal.
Input/Output Data 	Proses input/output data, parameter, informasi.
Garis alir 	Digunakan untuk menunjukkan arus dari proses.
<i>Decision</i> 	Digunakan untuk suatu penyeleksian kondisi di dalam program.
Proses terdefinisi 	Simbol yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
Penghubung 	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda

## 2.9 Data Flow Diagram

Menurut (Salahuddin, 2015) menjelaskan bahwa data flow diagram adalah representasi grafik yang menggambarkan aliran informasi dan transformasi

informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level; yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Notasi-notasi pada DFD adalah seperti tabel 2.3 dibawah ini :

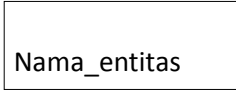
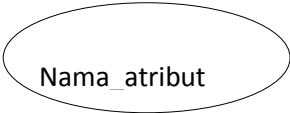
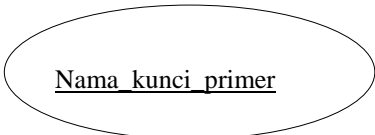
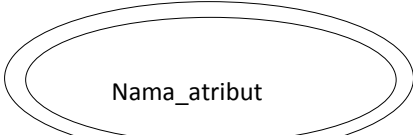
Tabel 2.3 simbol *Data Flow Diagram*

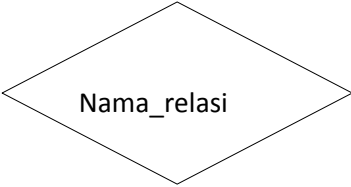
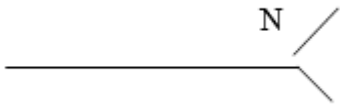
Notasi	Keterangan
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harus menjadi fungsi atau prosedur didalam kode program</p>
	<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>) pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat mejadi table-tabel basis data yang dibutuhkan, table-tabel ini juga harus sesuai dengan perancangan table-tabel pada basis data (ERD)</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan atau keluaran atau orang-orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan</p>
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p>

## 2.10 ERD (*Entity Relational Diagram*)

Menurut (Salahuddin, 2015) pemodelan awal basis data yang paling banyak digunakan adalah Entity Relatioanal Diagram (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Notasi-notasi pada ERD adalah seperti pada tabel 2.4 di bawah ini :

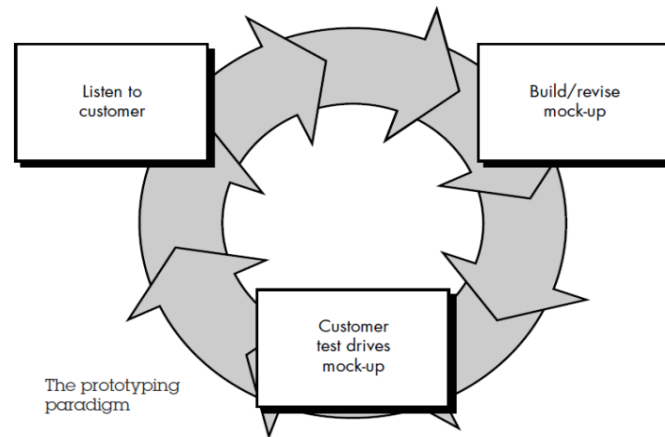
Tabel 2.4 Notasi pada model ERD

Simbol	Keterangan
Entitas /entity 	Entitas/entity merupakan data inti yang akan disimpan; bakal table pada basis data
Atribut 	Atribut/field;atau kolom data yang butuh disimpan dalam suatu entitas
Atribut Kunci Primer 	<i>Field</i> atau kolom yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses recod yang diinginkan; biasanya berupa id
Atribut Multi nilai 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatau entitas yang dapat memiliki nilai lebih dari satu.

<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p>
<p>Asosiasi / Association</p> 	<p>Penghubung adalah penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian, Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.</p>

### 2.11 Model *Prototype*

(A.S, Rosa & Salahudin. M, 2016) Model *prototype* (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *prototype* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *prototype* biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program *prototype* ini dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*. (S, Rosa & M, 2016) Berikut adalah gambar dari model *prototype* :



Gambar 2.2 Model *Prototype*

( A.S, Rosa & Salahudin. M, 2016) *Mock-up* adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai *prototype* perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan *prototype* sampai sesuai dengan keinginan pelanggan (*customer*) atau *user*. Proses pada model *prototyping* dapat dijelaskan sebagai berikut :

- a. Pengumpulan kebutuhan : *Developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
- b. Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
- c. Evaluasi *prototype* : Klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak. Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype* dibuat untuk memuaskan kebutuhan klien dan untuk membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan.