

BAB 1

TINJAUAN PUSTAKA

1.1 Sistem Pendukung Keputusan (SPK)

2.1.1 Pengertian

Sistem adalah suatu kumpulan elemen yang saling berhubungan dalam bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*)

Keputusan merupakan suatu tindakan pilihan diantara beberapa alternative yang bertujuan untuk mencapai suatu tujuan. Teori keputusan yaitu sebuah area kajian matematika diskrit yang bermodelkan penentuan keputusan oleh manusia dalam pengetahuan, rekaya, dan seluruh kegiatan social manusia. Ini terkait oleh seberapa benarnya ideal seorang pengambil keputusan dalam membuat atau harus membuat keputusan, dan seberapa optimal keputusan dapat dicapai.

Beberapa penjelasan keputusan yang dirumuskan para ahli dijelaskan sebagai berikut[4] (Hasan, 2004):

1. Menurut Ralph C. Davis

Keputusan merupakan suatu dampak pemecahan masalah yang dihadapi dengan jelas. Suatu keputusan adalah jawaban yang tetap atas suatu pertanyaan. Keputusan wajib bisa menjawab pertanyaan mengenai apa yang sedang dibahas dalam keterkaitannya dengan suatu perencanaan. Keputusan bisa juga berupa perlakuan tentang pelaksanaan yang begitu membelok dari rencana awal

2. Menurut Mary Follet

Keputusan merupakan suatu hukum kondisi. Jika seluruh hal tentang situasi tersebut bisa didapatkan dan semua yang berperan, baik pengarah dan pelaku akan mengikuti hukumnya atau kewajibannya, maka tidak sama dengan

menuruti perintah. Kewajibantetap dijalankan, namun itu adalah wewenang dari hukum situasi.

3. Menurut James A.F.Stoner

Keputusan merupakan penyaringan di antara alternative-alternatif. Definisi tersebut mempunyai tiga arti, yaitu :

- a. Ada pilihan atas dasar logika atau pertimbangan
- b. Ada beberapa alternative yang harus dan dipilih salah satu yang terbaik
- c. Ada tujuan yang ingin dicapai, dan keputusan itu makin mendekati pada tujuan tertentu.

4. Menurut Prof.Dr.Prajudi Atmosudirjo,SH

Keputusan merupakan suatu penyelesaian daripada proses gagasan terhadap suatu masalah guna menyelesaikan permasalahan apa yang harus dikerjakan guna mengatasi masalah itu, dengan menjatuhkan pilihan pada suatu alternative.

Tabel 2.1 Penelitian Terkait

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	Ariyanto (2012)	Pemilihan karyawan secara periode namun belum optimal pelaksanaannya	Aplikasi SPK dengan metode <i>Simple Additive Weighting</i> (SAW)	Berhasil dibuatnya sistem informasi pendukung keputusan pemilihan karyawan terbaik dengan menggunakan metode <i>Simple Additive Weighting</i>
2.	Aji Dewantoro(2013)	Penentuan guru teladan yg bertujuan mengevaluasi kinerja guru yang kurang	Aplikasi SPK dengan metode <i>Simple Additive Weighting</i>	Sisten yang telah dibuat membuktikan bahwa metode <i>Simple Additive Weighting</i> dapat diterapkan dal telah

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
			(SAW)	dibuktikan dalam tahap pengujian sistem

2.1.2 Karakteristik dan Kemampuan SPK

Menurut [5]Turban (2005), ada beberapa karakteristik dari system pendukung keputusan, diantaranya adalah sebagai berikut:

1. Mendukung seluruh kegiatan organisasi
2. Mendukung beberapa keputusan yang saling berinteraksi
3. Dapat digunakan berulang kali dan bersifat konstan
4. Terdapat dua komponen utama, yaitu data dan model
5. Menggunakan baik data eksternal maupun internal
6. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*

Selain itu, Turban juga memiliki kemampuan yang harus dimiliki oleh sebuah system pendukung keputusan, diantaranya adalah sebagai berikut:

1. Menunjang pembuatan keputusan manajemen dalam menangani masalah semi terstruktur dan tidak terstruktur
2. Membantu manajer pada berbagai tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah
3. Menunjang pembuatan keputusan secara kelompok dan perorangan
4. Menunjang pembuatan keputusan yang saling bergantung dan berurutan
5. Menunjang tahap-tahap pembuatan keputusan antara lain *intelligence, design, choice* dan *implementation*
6. Menunjang berbagai bentuk proses pembuatan keputusan dan jenis keputusan
7. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat fleksibel
8. Kemudahan melakukan interaksi system,

9. Meningkatkan efektivitas dalam pembuatan keputusan daripada efisiensi
10. Mudah dikembangkan oleh pemakai akhir
11. Kemampuan pemodelan dan analisis dalam pembuatan keputusan
12. Kemudahan melakukan pengaksesan berbagai sumber dan format data

2.1.3 Komponen Sistem Pendukung Keputusan

1. Data Management Sistem

Segala aktivitas yang berhubungan dengan pengambilan, penyimpanan dan pengaturan data-data yang relevan dengan konteks keputusan yang akan diambil. Selain itu, komponen ini juga menyediakan berbagai fungsi keamanan, prosedur integritas data, dan administrasi data secara umum yang berkaitan dengan SPK. Berbagai tugas ini dilakukan dalam data management system beserta beberapa sub sistemnya yang diantaranya meliputi database, database management system, repository data, dan fasilitas query data.

2. Model Management Sistem

Sistem ini menampilkan aktivitas pengambilan, penyimpanan dan pengaturan data dengan berbagai model kuantitatif, yang menyediakan kemampuan analitis untuk SPK.

3. Knowledge Base

Aktivitas yang berkaitan dengan pengenalan masalah, dan menghasilkan solusi final maupun sementara, hal-hal yang berkaitan dengan manajemen proses pemecahan masalah merupakan inti dari komponen ini. Knowledge base merupakan “otak” dari kelima komponen SPK. Data dan model diolah untuk kemudian hasilnya menjadi bahan pertimbangan bagi user dalam mengambil keputusan.

4. User interface

Adalah jalur penghubung antara system dengan user, sehingga komponen-komponen system SPK dapat diakses dan dimanipulasi dengan mudah oleh user untuk memberikan dukungan pada pengambilan keputusan. Kemudahan penggunaan dan komunikasi antar user dan SPK pada dasarnya merupakan ukuran keberhasilan penggunaan SPK itu sendiri.

5. User(s)

Desain, implementasi dan pemanfaatan SPK tidak akan efektif jika tidak disertai peran pengguna, kemampuan, ketrampilan, motivasi, dan pengetahuan pengguna sebagai pengatur SPK, akan menentukan efektivitas dari penggunaan SPK.

1.2 Penjelasan Mengenai PHP

PHP adalah bahasa pemrograman script yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Sebagai sebuah *server-side, script* PHP juga memiliki keunggulan sebagai berikut:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server mendukung PHP dapat ditemukan dimana-mana dari muai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relative muda.
3. Dalam sisi pengembangan lebih muda, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

1.3 Unified Modelling Language (UML)

1.3.1 Pengertian

UML merupakan singkatan dari “*United Modeeling Language*” yaitu suatu metode permodel secara visual untuk sarana perancangan sistem berioentasi objek, atau definis UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software. Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software

1.3.2 Tujuan atau fungsi dari penggunaan UML

Inilah beberapa tujuan atau fungsi dari penggunaan UML, yang di antaranya :

1. Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep beriorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
6. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun mesin.

1.3.3 Jenis-jenis diagram UML




1.3.3.1 Use case diagram

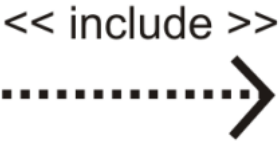
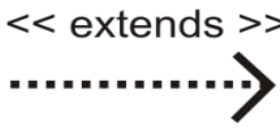
Yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga men-dekripsikan tipe interaksi antara si pemakai sistem dengan sistemnya. Syarat penamaan pada *use case* adalah nama

didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Pada dasarnya actor bukanlah bagian dari diagram, namun untuk dapat terciptanya suatu use case diagram diberikan beberapa actor dimana actor tersebut menjelaskan seseorang atau sesuatu (seperti perangkat, system lain) yang berinteraksi dengan system. Sebuah actor mungkin hanya memberikan informasi inputan pada system, hanya menerima informasi dari system atau keduanya menerima dan member informasi pada system, actor hanya berinteraksi dengan *use case* tetapi tidak memiliki control atau use case. Actor digambarkan secara umum atau spesifik, dimana untuk membedakannya anda dapat menggunakan relationship.
2. Use case merupakan gambaran fungsional dari suatu system, sehingga customer atau pengguna system paham dan mengerti kegunaan system yang akan dibangun.

Tabel 2.2 Simbol Use Case

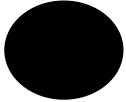
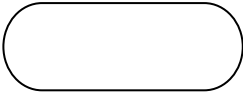
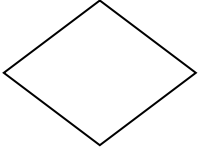
Symbol	Pengertian
 actor	Sebuah entitas manusia / mesin yang berinteraksi dengan sistem
 use case	Merepresentasikan fungsionalitas dari suatu system / pekerjaan tertentu pada sistem
 interaksi	Menjelaskan hubungan antar use case yang berwujud pertukaran informasi


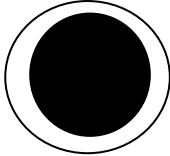
	Menunjukkan suatu use case harus dipenuhi
	Menunjukkan suatu use case akan dilaksanakan bersifat optional

1.3.3.2 Activity Diagram

Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada system

Tabel 2.3 Simbol Activity Diagram


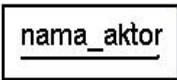


Simbol	Deskripsi
	Status awal merupakan symbol aktivitas sistem
	Aktivitas merupakan symbol yang dilakukan sistem
	Decision merupakan symbol percabangan dimana jika ada pilihan aktivitas lebih dari satu




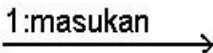
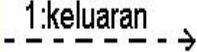
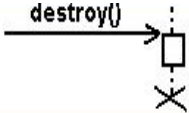
	Penggabungan merupakan simbbol lebih dari satu aktivitas digabungkan menjadi satu
	Status akhir merupakan symbol aktivitas memiliki sebuah akhir

1.3.3.3 Sequence diagram

Diagram sekuan menggambarkan kelakuan objek pada *use case* dengan mendiskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuan juga dibutuhkan untuk melihat scenario yang ada pada use case.



Tabel 2.4 Sequence Diagram


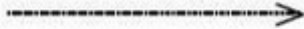

Simbol	Pengertian
aktor  atau 	Orang, proses, atau system lain yang berinteraksi dengan sismte informasi dan mendapat manfaat dari sisstem Ditempatkan di bagian atas diagram
objek 	Berartisipasi secara berurutan dengan mengirimkan dan/ atau menerima pesan
Garis hidup objek 	Menandakan kehidupakn objek selama urutan

Objek sedang aktif berinteraksi 	Fokus control : Adalah persegi panjang yang sempit panjang ditempatkan di atas sebuah garis hidup
pesan 	Objek mengirim satu pesan ke objek lainnya
	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
	Menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya
	Objek menghasilkan suatu kembalian ke objek tertentu
	Menyatakan suatu objek mengakhiri hidup objek yang lain

1.3.3.4 Class diagram

Class diagram adalah jenis diagram pada UML yang dipergunakan untuk menampilkan kelas-kelas yang terdapat pada sebuah system yang akan digunakan nantinya. Oleh karena itu diagram ini bisa memberikan suatu gambaran mengenai system ataupun relasi-relasi yang terdapat pada system tersebut.

Simbol	Deskripsi
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai <i>multiplicity</i>
Asosiasi Berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan

	<i>multiply</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-generalisasi (umum khusus)
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi 	Relasi antar kelas dengan makna semua-bagian (Whole-Part)

1.4 Database

2.4.1 Pengertian

Database adalah: “Kumpulan dari data yang berhubungan satu dengan yang lainnya yang tersimpan di perangkat keras computer dan digunakan perangkat lunak untuk memanipulasinya.” (Jogianto H,1999:20)

1.4.2 Komponen utama database

1. Perangkat keras (*hardware*)

Merupakan perangkat keras yang dibutuhkan dalam pengelolaan database, berupa computer beserta seluruh kelengkapan yang dibutuhkan, seperti prosesor, memori, harrdisk sebagai media penyimpanan datanya.

2. Data

Merupakan komponen yang akan diolah sehingga bernilai informasi dan dapat dijadikan sebagai bahan pengambilan keputusan.

3. Perangkat Lunak (Software)

Perangkat lunak, dalam hal ini DBMS, berfungsi sebagai aplikasi untuk mengelola database.

4. Pengguna (User)

User merupakan pengguna, biasanya ditujukan kepada pengguna suatu system yang umumnya adalah manusia.

1.5 Metode *Simple Additive Weighting* (SAW)

1.5.2 Pengertian

Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternative pada semua atribut.

Konsep dasar metode *Simple Additive Weighting* (SAW) adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternative pada semua atribut. Metode *Simple Additive Weighting* (SAW) membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternative yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making (MADM)*. *MADM* itu sendiri merupakan suatu metode yang digunakan untuk mencari alternative optimal dan sejumlah alternative dengan criteria tertentu.

Metode *Simple Additive Weighting* (SAW) ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk alternative diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya

1.5.3 Langkah-langkah Metode *Simple Additive Weighting*

1. Menentukan criteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i .

2. Menentukan rating kecocokan setiap alternative pada setiap criteria.
3. Membuat matriks keputusan berdasarkan criteria (C_i), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R .
4. Hasil akhir diperoleh dari proses perangkangan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternative terbaik (A_i) sebagai solusi.

Formula untuk melakukan normalisasi tersebut adalah

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Dimana :

R_{ij} = rating kinerja ternormalisasi

Max_{ij} = nilai maksimum dari setiap baris dan kolom

Min_{ij} = nilai minimum dari setiap baris dan kolom

X_{ij} = baris dan kolom dari matriks

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Dengan r_{ij} adalah rating kinerja ternormalisasi dari alternative A_i , pada atribut C_j ; $i=1,2,\dots, m$ dan $j = 1,2,\dots,n$. Nilai preferensi untuk setiap alternative (V_i) diberikan sebagai :

Dimana L

V_i = nilai akhir dari alternative

W_j = bobot yang telah ditentukan

R_{ij} = Normalisasi matriks

Nilai V_i yang lebih besar mengindikasikan bahwa alternative A_i telah terpilih.

1.6 My SQL

[6]Arief (2011d:152) “MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya”. Kekuatan utama MySQL adalah pada kecepatannya, terutama untuk kecepatan koneksi (overhead koneksi yang rendah) dan kecepatannya untuk *query-query* yang sederhana dan juga cukup mudah untuk digunakan. Hal ini membuatnya cocok dipakai sebagai *backend* untuk aplikasi web

1.7 XAMPP

Fungsi XAMPP sendiri adalah sebagai server yang berdiri (localhost), yang terdiri beberapa program antara lain :**Apache HTTP Server, MySQL database**, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman **PHP dan Perl**. Nama XAMPP sendiri merupakan singkatan dari X(Empat system operasi apapun), Apache, MySQL,PHP dan Perl. Program ini tersedia dalam GNU General Public Licensi dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis.

1.8 Metode Pengembangan Sistem Prototype

1.8.2 Pengertian

Proses pengembangan system seringkali menggunakan metode Prototype. Metode ini sangat bagus dipakai dalam menyelesaikan masalah kesalahpahaman antara *user* dan analis yang berada akibat *user* tidak bisa mengartikan secara jelas kebutuhannya[7](Mulyanto 2009).

Prototype adalah pengembangan yang cepat dan pengujian terhadap model kerja dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli system informasi dan ahli bisnis. Prototype disebut juga desain

aplikasi cepat (*rapid application design/RAD*) karena menyederhanakan dan mempercepat desain system (O'brien,2005).

Sebagian user kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai dengan kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis dengan memahami kebutuhan user dan menerjemahkannya ke dalam bentuk model (prototype). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan user.

1.8.3 Kelebihan dan kekurangan

1. Keunggulan Prototipe adalah

- a. Adanya komunikasi yang baik antara pengembang dan pelanggan
- b. Pelanggan berperan aktif dalam pengembangan system
- c. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
- d. Lebih menghemat waktu dalam pengembangan system
- e. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya

2. Kekurangan Prototiep adalah

- a. Pelanggan tidak melihat bahwa perangkat lunak belum mencerminkan kualitas perangkat lunak secara keseluruhan dan belum memikirkan pemeliharaan dalam jangka waktu yang lama
- b. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma yang sederhana
- c. Hubungan pelanggan dengan computer mungkin tidak menggambarkan teknik perancangan yang baik

1.8.4 Tahapan Pembuatan Prototype

Tahap-tahap pembangunan *Prototype* model menurut *Roger S.Pressman, Ph.D*[8] adalah :

1. Mendengarkan Pelanggan

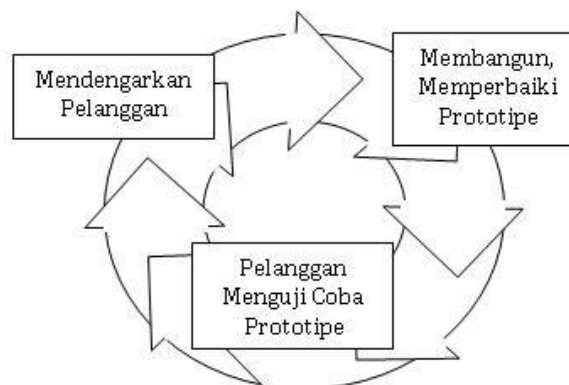
Pada tahap ini dilakukan pengumpulan kebutuhan sistem dengan cara mendengarkan keluhan dari pelanggan. Untuk membuat sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang sedang terjadi.

2. Merancang Dan membuat Prototype

Pada Tahap ini dilakukannya perancangan dan pembuatan prototype sistem. *Prototype* yang dibuat sesuai dengan kebutuhan system yang didefinisikan sebelumnya dari keluhan pelanggan maupun pengguna.

3. Uji Coba

Pada tahap ini, *Prototype* dari sistem ini diuji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan –kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki *Prototype* yang ada.



Gambar 2.1 Metode Prototype

