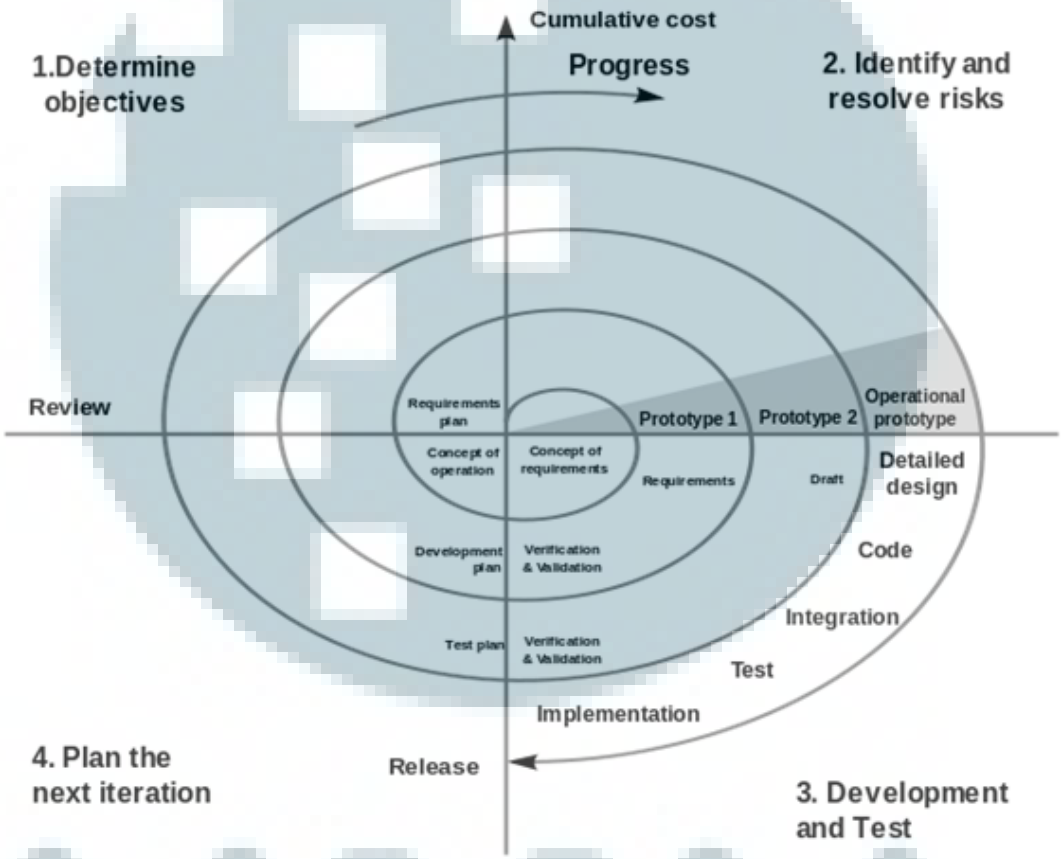
BAB II

LANDASAN TEORI

2.1 Sistem Desain



Gambar 2.1 stuktur SDLC *spiral model*

Dalam melakukan pengembangan aplikasi pada penelitian kali ini penulis menggunakan *SDLC (software development life cycle)* dengan model *spiral*, menurut (Boehm, 1988) tahapan dalam sistem desain berikut yaitu :

1. *Determine objectives*

Tahap pertema pada *SDLC* ini adalah menentukan hal apa yang ingin diselesaikan lewat aplikasi yang akan dikembangkan, pada penelitian ini penulis akan membuat aplikasi yang dapat membantu para pejalan kaki yang ingin menggunakan angkutan umum khususnya Metromini. Dan penulis harus tahu kira – kira aplikasi seperti apa yang dapat membantu pejalan kaki menggunakan angkutan umum, maka dari itu penulis harus mengumpulkan pendapat dari penjalan kaki yang digunakan sebagai acuan *user requirement* dalam melakukan pengembangan program.

2. *Identify and resolve riks*

Dari *user requierment* yang sudah dikumpulkan pada tahap kedua ini penulis akan melakukan identifikasi dari setiap pendapat bagaimana cara menyelesaikan masalah tersebut dan memilih masalah mana saja yang dapat diselesaikan dengan pembuatan program yang nantinya pada tahap ini penulis akan merancang UML *(unified modeling language).*

3. *Development and test*

Setelah menggambarkan setiap alur dari fungsi – fungsi yang dapat program jalankan, pada tahap ini dimulai dengan proses implementasi dari UML yang sudah dibuat menjadi sebuah proses nyata dalam aplikasi yang nantinya bekerja seperti yang sudah didesain pada UML.

4. *Plannext iteration*

Pada tahap ini penulis akan melakukan tes pada pengguna, dimana tanggapan dari pada pengguna terhadap aplikasi dapat dijadikan evaluasi terhadap aplikasi yang sekarang untuk dikembangkan kembali menjadi aplikasi yang lebih baik.

2.2 Pengambilan dan Pengolahan Data

2.2.1 Rumus Perhitungan Estimasi Waktu

Prediksi yang yang akan ditampilkan pada pengguna diperoleh dari :

waktu tempuh = jarak tempuh / Kecepatan

Estimasi2 = Estimasi Halte1 + watku tempuh

Jadi pada halte pertama akan ditentukan starting point sebanyak bus yang akan di prediksi sesuai dengan asumsi. Setelah itu pada halte kedua estimasi dihitung dari startime pada halte ditambah dengan waktu tempuh, dan estimasi halte kedua akan menjadi startime pada halte kedua yang akan ditambahkan dengan waktu tempuh antara halte 2 ke halte 3 dan menghasilkan estimasi halte 3 dan seterusnya hingga mencapai halte terakhir, dan estimasi halte terakhir akan ditambahkan dengan waktu tempuh dari halte 1 ke halte terakhir sehingga menjadi estimasi pada halte satu dan variable tersebut akan diputar hingga batas waktu estimasi yaitu jam 18.00 WIB.

2.2.2 Teknik Pengambilan Sample

Pada dasarnya pengambilan sample dapat disesuaikan dengan kebutuhan penelitian menurut (Sugiyono, 2010) *sampling* pada dasarnya dibagi menjadi dua jenis yaitu :

1. *probability sampling,* merupakan teknik pengambilan data yang memberikan peluang yang sama bagi setiap unsur populasi untuk dipilih menjadi anggota sample, beberapa contoh dari *probability sampling* adalah :
    a. *Simple random samlping, sampling* secara acak tanpa memperhatikan strata;
    b. *Proportionate stratified random sampling, sampling* secara acak, namun jumlahnya disesuaikan dengan banyak strata;
    c. *Disproportionate stratified random sampling, sampling* secara acak, dan jumlah yang diambil perstrata juga berbeda sesuai dengan populasi dalam setiap stratanya;
    d. *Area sampling, sampling* acak berdasarkan dearah populasi yang telah ditetapkan.

2. *Non probability sampling,* teknik pengambilan sample yang tidak memberi peluang yang sama bagi setiap unsur populasi untuk dipilih menjadi sample, beberapa contoh dari *Non probability sampling* adalah :
    a. *Sampling sistematis,* pengambilan sampel yang didasari oleh nomor urut anggota populasi yang sudah ditentukan sebelumnya;

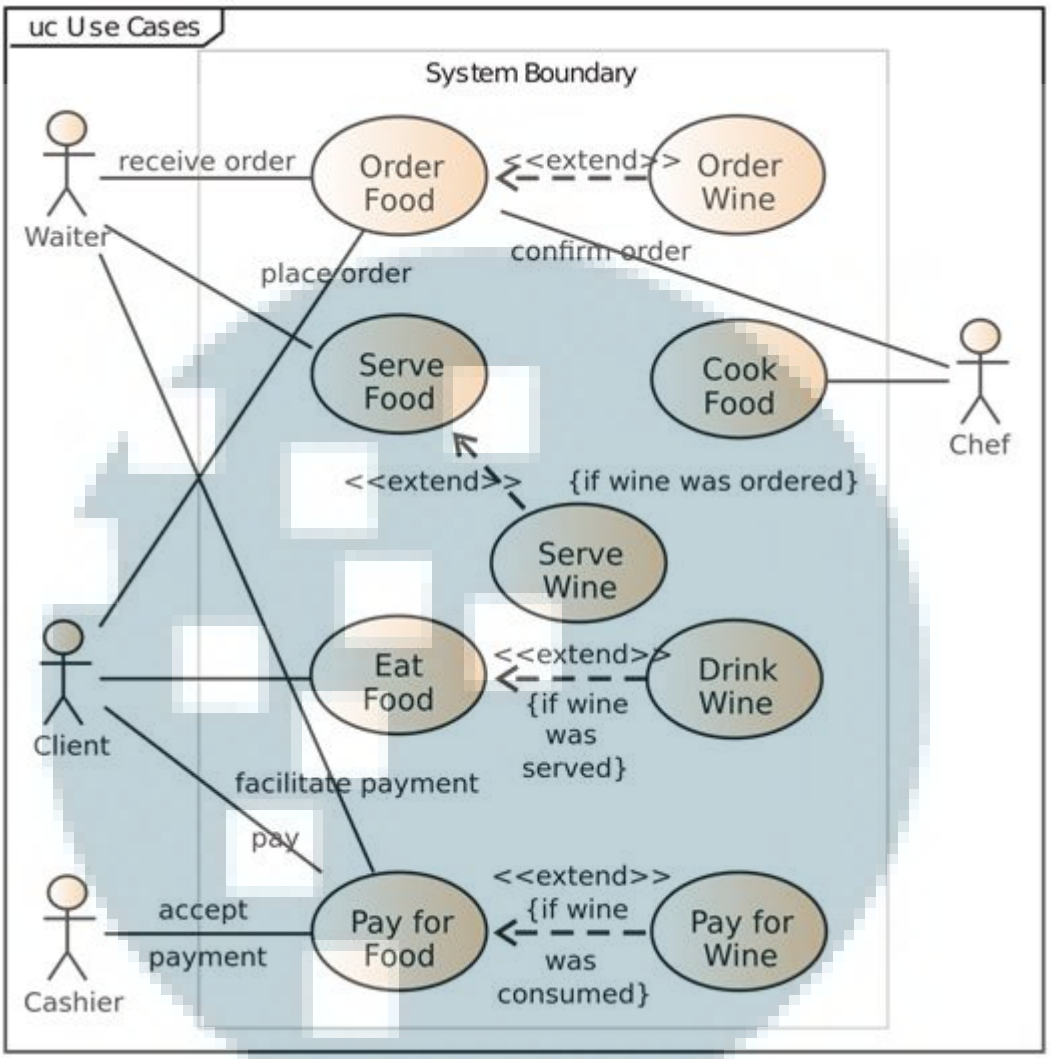
b. *Sampling* kuota, pemgambilan sampel yang disesuaikan dengan kebutuhan;
c. *Sampling aksidental*, pengambilan sampel yang berdasarkan kebetulan;
d. *Sampling purposive*, pengambilan sampel yang didasari oleh suatu pertimbangan atau kriteria tertentu;
e. *Sampling* jenuh, pengambilan sampel yang menggunakan seluruh populasi sebagai sampel;
f. *Sampling snowball*, sampel yang diambil sedikit dan terus ditambahkan jumlahnya;

2.3 UML

UML memiliki kepanjangan *Unified Modeling Language* merupakan bahasa spesifikasi standar untuk melakukan dokumentasi dan membangun sistem perangkat lunak, adapun himpunan struktur untuk pemodelan desain program berorientasikan pada objek, UML merupakan metodologi untuk mengembangkan sistem OOP untuk medukung pengembangan sistem. UML memiliki berbagai jenis model yang dapat digunakan sesuai dengan objek yang di representasikan, jenis – jenis UML yaitu:

1. *Usecase* diagram

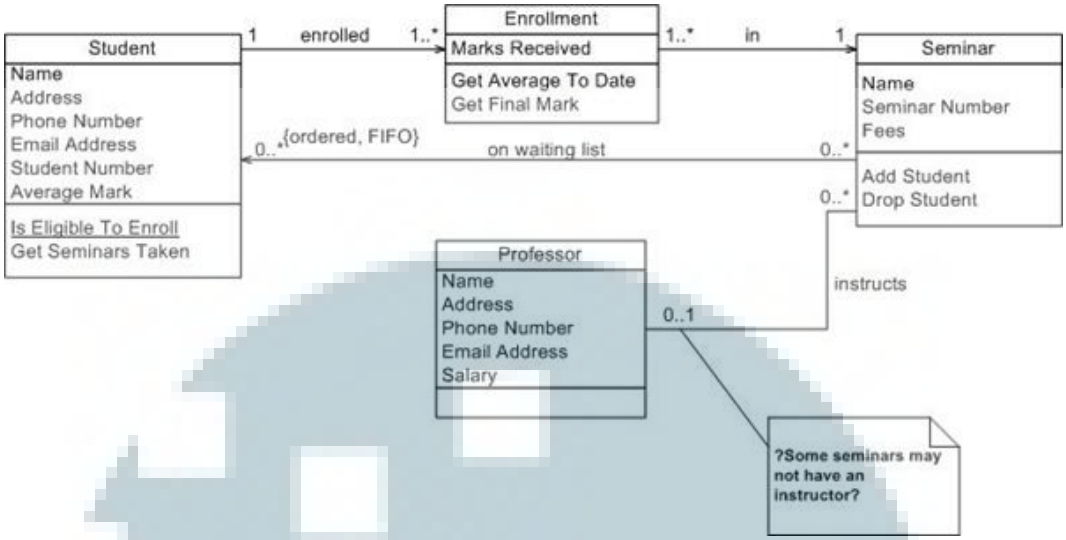
   Usecase diagram merupakan sebuah teknik untuk mengambarkan kebutuhan fungsional dari sebuah sistem. *Use case* diagram dapat menjelaskan jenis – jenis interaksi antar pengguna dalam sistem dan sistem itu sendiri.



Gambar 2.2 Contoh *usecase* diagram

2. *Class* diagram

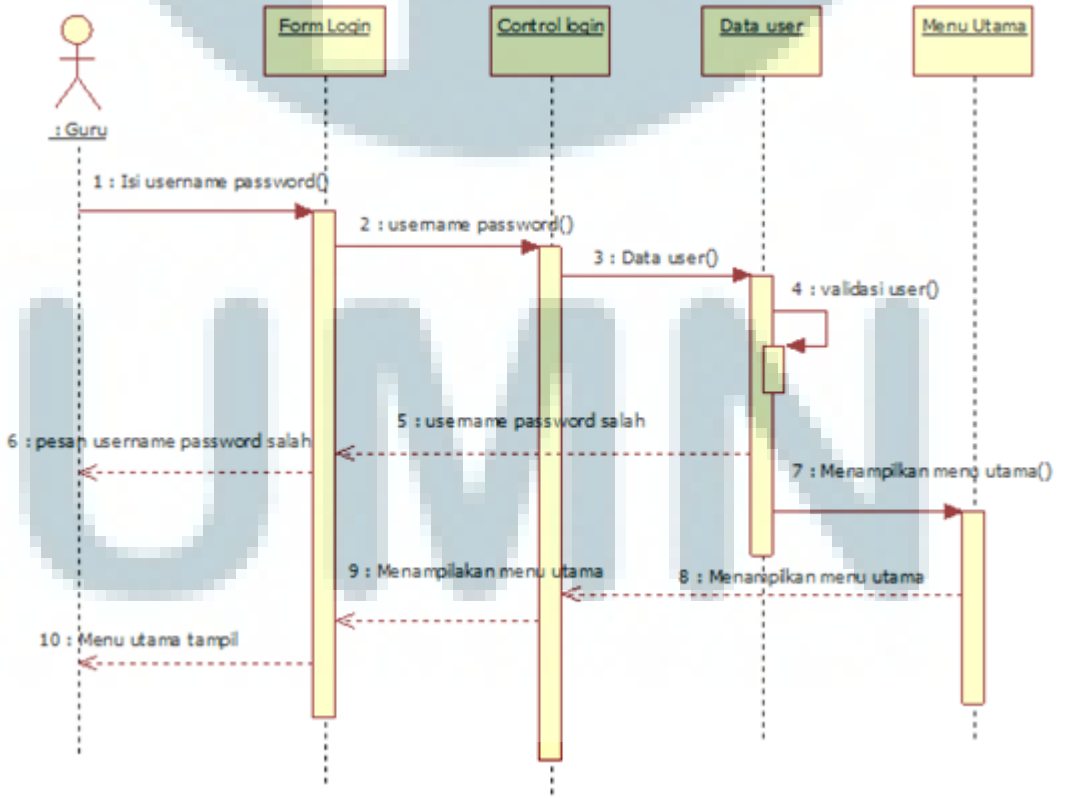
Class diagram mengambarkan tipe–tipe dari objek dalam sebuah sistem dan jenis relasi statik yang ada pada objek tersebut. *Class* diagram juga mengambarkan sifat dan operasi dari suatu *class* dan kondisi yang ada pada hubungan antar objek.



Gambar 2.3 contoh *class* diagram

3. *Sequence* diagram

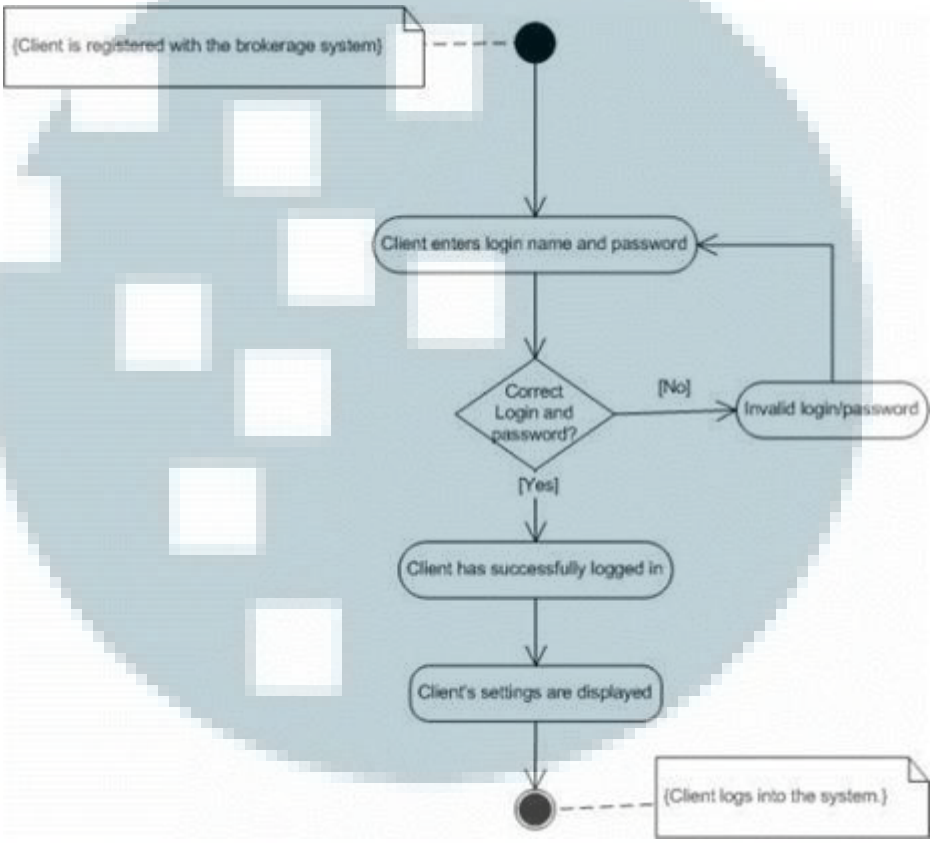
Sequence diagram menjelaskan bagaimana sebuah group yang terdiri dari beberapa objek berkolaborasi dengan beberapa kondisi.



Gambar 2.4 contoh *sequence* diagram

4. *Acitivity* diagram

Acitivity diagram merupakan sebuah teknik yang mengambarkan logika prosedur, proses bisnis dan siklus pekerjaan



Gambar 2.5 contoh *activity* diagram

2.4 Penyajian dan penampilan Data

2.4.1 Teori simulasi

Simulasi adalah meniru atau menggambarkan operasi–operasi yang terjadi pada berbagai macam proses yang terjadi dengan mengunakan bantuan komputer (M.Law & Kelton, 2014). Proses tersebut sering dikenal sebagai sistem, sistem merupakan kesatuan yang saling berinteraksi untuk menghasilkan sebuah *output*.

Dalam melakukan simulasi ada beberapa jenis simulasi yang dapat disesuaikan dengan data yang diperoleh yaitu :

1. Simulasi statis dan dinamis

Simulasi statis adalah simulasi yang menggambarkan suatu sistem yang tidak dipengaruhi oleh waktu atau hanya dipengaruhi oleh waktu–waktu tertentu yang cenderung tetap, sementara simulasi dinamis merupakan simulasi yang hasil tergantung waktu.

2. Simulasi deterministik dan stokastik

Simulasi deterministik merupakan simulasi yang menggabarkan suatu proses yang pasti terjadi, sementara simulasi stokastik merupakan simulasi yang prosesnya cenderung tidak pasti

3. Simulasi kontinu dan diskrit

simulasi kontinu merupakan simulasi dari suatu proses yang komponen–komponen sistemnya bersifat kontinu, sedangkan simulasi diskrit merupakan simulasi suatu proses yang komponen komponennya berubah sesuai dengan perubahan waktu.

2.5 Interaksi Manusia dan Komputer

Menurut Shneiderman (Shneiderman, 2005), interaksi manusia dan komputer adalah disiplin ilmu yang berfokus pada perancangan desain halaman antarmuka atau sering disebut sebagai *user interface* (UI), yang dimana dalam bukunya Shneiderman meyebutkan bahwa UI harus dibuat sedemeikian rupa

sehingga seorang pengguna dapat mengerti bagaimana cara berinteraksi dengan sistem.

UI adalah bagian dari sebuah sistem komputer, dimana lewat UI manusia dapat melakukan perintah–perintah atau fungsi yang dapat diproses oleh komputer. Tujuan adanya sebuah UI dalam sebuah sistem tentunya untuk mempermudah manusia dalam melakukan pengoprasian komputer. Salah satu contohnya sebuah minimart sistem kasirnya tidak mengunakan UI dan kasir harus memasukkan *query* untuk meyimpan data penjualan ke *database* kantor pusat, belum tentu kasir mengerti bagaimana cara melakukannya, apabila kasir mengunakan UI maka kasir akan lebih mudah mengerti bagaimana cara menggunakan sistem tersebut.

2.5.1 Program Interaktif

Menurut Scheiderman (Shneiderman, 2005), sistem yang interaktif adalah sistem yang *user friendly*, dan Scheiderman memiliki beberapa kriteria agar suatu sistem dapat dikatakan *user friendly* ,yaitu:

1. Waktu belajar yang sikat;
2. Kecepatan pengolahan infromasi yang tepat;
3. Tingkat kesalahan pemakaian yang rendah;
4. Penghafalan dalam jangka waktu tertentu;
5. Kepuasaan pribadi.

2.5.2 Pedoman Merancang *User Interface*

Pedoman yang dipakai dalam melakukan penelitian ini demi mendapatkan sistem yang *user friendly* adalah *eight golden rules*. Beberapa kriteria yang harus diperhatikan dalam pembuatan UI dalam *eight golden rules* adalah :

1. *Strive for consistency*

   Dalam hal ini hal yang konsisten harus dilakukan oleh parah desainer dalam melakukan perancangan sistem baik dari pemilihan warna, bentuk ukuran huruf serta peletakan tombol fungsi atapun penampil hasil, terkecuali untuk kasus khusus contohnya seperti tulisan merah untuk peringatan.

2. *Enable frequent user to user shortcuts*

   Buatlah suatu sistem yang dapat digunakan oleh semua kalangan dengan memfailitasi pengguna untuk melakukan transformasi pada konten sistem. Contohnya kostumisasi shortcut atau memberikan fungsi pengaturan pada sistem.

3. *Offer informative feedback*

   Memberikan tanggapan apabila pengguna melakukan hal tertentu. Ini berguna agar pengguna mendapatkan sebuah konfirmasi tetang apa yang mereka lakukan dalam sebuah sistem, contohnya pengguna memasukkan data ke dalam sistem, maka sistem akan memberi tanggapan bahwa data telah masuk dengan sukses ataupun gagal karena kesalahan.

4. *Design dialogs to yield closure*

Tidakan yang berulang harusnya dikelompokan menjadi beberapa grup seperti tahap awal, tahap menengah dan tahap akhir. Memberikan tanggapan yang informatif kepada pengguna setelah menyelesaikan salah satu grup tidakan untuk memberikan rasa puas ataupun untuk membuat pengguna siap untuk maju ke grup selanjutnya.

5. *Prevent erros*

Sebisa mungkin, kita mendesain sistem buatlah desain yang membuat pengguna tidak melakukan kesalahan fatal. Contohnya dengan membuat fungsi yang menampilkan kesalahan pengguna serta bagaimana cara memperbaikanya.

6. *Permit easy reversal of actions*

Sebisa mungkin, memberikan keempatan pada pengguna untuk mereverse suatu tindakan. Karena dengan begitu pengguna mengetahui bahwa *error* dapat dicegah dan membuat pengguna ingin mengeksplorisasi pilihan – pilihan yang tidak familiar.

7. *Support internal locus of control*

Sebisa mungkin dalam membuat sistem hasil dari tindakan pengguna ditampilkan sesederhana mungkin dalam artian langsung pada hasilnya. Karena pengguna yang berpengalaman tidak mau kesulitan dalam mendapatkan informasi.

8. *Reduces shorth-term memory load*

Dalam membuat sistem hal yang harus dihindari adalah dimana pengguna harus mengingat informasi tertentu dari halaman ke halaman yang lainnya dengan tujuan dapat melanjutkan akses ke halaman berikutnya.

2.6 PHP

Menurut agus saputra PHP atau yang memiliki kepanjangan *hypertext preprocessor* merupakan suatu bahasa pemrograman yang dapat digunakan untuk Membangun *dynamic website*, PHP merupakan bahasa premoresessan dalam HTML (*hyper text markup language*) yang biasanya digunakan untuk tujuan seperti :

a. *Server-side scripting*
b. *Command-line scripting*
c. *Client-side GUI application*

Menurut (welling & Thomson, 2003) PHP meimiliki beberapa keunggulan yaitu :

a. Scalability

   Dapat secara efektif mengimplementasikan *horizontal scaling* dengan beberapa server.

b. Database integration

   Di dukung oleh beberapa database, seperti oracle, dbm, file pro , hyperwave dan lainnya. PHP juga menggunakan ODBC yang membuat anda dapat melakukan koneksi ke database yang mendukung ODBC

driver.Memiliki tambahan *native libraries* yaitu berupa *database abstration layer* yang disebut PHP *database object* (PDO), yang memungkinkan pengguna untuk melakukan koneksi secara konsisten dan aman.

c. Built-in libraries

PHP memiliki banyak *built-in* function yang dapat membantu pengguna dalam membuat website sesuai kriteria pekerjaan yang diinginkan dengan tetap mempertahankan baris code yang sederhana.

d. Cost

PHP tidak memerlukan biaya dalam instalasi ataupun license.

e. Ease of learning

Syntax PHP dibuat dari beberapa programming language dasar seperti C dan Perl. Apabila pengguna sudah familar dengan bahasa C maka tidak akan mengalami kesulitan pada saat menggunakan PHP.

f. Object-oriented support

PHP didukung OOP yang membuat pengguna dapat menggunakan fitur seperti private ataupun protected function.

g. Portability

Dapat digunakan pada banyak sistem operasi seperti linux, freeBSD ataupun unix berbayar seperti Solaris, OS X dan Windows.

h. Flexibility of development approach

PHP memungkinkan anda untuk mengimplementasikan fungsi sederhana yang dapat di gabungkan dengan aplikasi yang lebih besar walaupun aplikasi tersebut berbasis *design patterns* seperti MVC.

i. Source code

Pengguna memiliki hak akses ke *source code* yang memungkinkan modifikasi / menambahkan *source code*.

j. Avaibility of support of documentation

PHP didukung dengan dokumentsi dan forum – forum diskusi yang kaya akan informasi dan mudah untuk dijangkau.

2.7 MySQL

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread, multi-user*. Pada awalnya MySQL diciptakan pada tahun 1979, oleh Michael Widenius, seorang programmer komputer asal Swedia. Dia mengembangkan sebuah sistem *database* sederhana yang dinakan UNIREG yang menggunakan koneksi *low-level* ISAM *database engine* dengan *indexing*.

Beberapa keuntungan dari MySql adalah :

a. Low cost

MySQL gratis dan murah apabila ingin membayar lisensi untuk mempublikasikan aplikasi anda, apabila anda hanya ingin menggunakannya saja maka anda tidak perlu membayar lisensi.

b. Ease of use

MySQL dapat beradapatasi dengan RDBMS lainnya karena kebanyakan *database* menggunakan bahasa SQL sebagai *source code*.

c. Portability

Dapat digunakan pada banyak macam unix.

d. Source code

PHP dapat memodifikasi dan menentukan source code untuk MySQL.

e. Avability support

Mendapatkan dukungan seperti cosulting dan probelm resolving.