

BAB II

LANDASAN TEORI

Dalam penyusunan Tugas Akhir ini penulis mengumpulkan beberapa sumber untuk dapat lebih memahami teori dari apa yang telah, dan akan dilaksanakan dalam penyusunan Tugas Akhir ini.

2.1 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien.[1]

2.2 Pengertian Website

Website atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).[13]

Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti Friendster, Multiply, dll. Dalam sisi pengembangannya, *website* statis hanya bisa di-*update* oleh pemiliknya saja, sedangkan *website* dinamis bisa di-*update* oleh pengguna maupun pemilik.

2.3 Object Oriented (OO)

Teknologi *object oriented* merupakan paradigma baru dalam rekayasa *software* yang didasarkan pada objek dan kelas.[14]

Diakui para ahli bahwa *object oriented* merupakan metodologi terbaik yang ada saat ini dalam rekayasa *software*. *Object oriented* memandang *software* bagian per bagian, dan menggambarkan satu bagian tersebut dalam satu objek.

Satu objek dalam sebuah model merupakan suatu fokus selama dalam proses analisis, desain, dan implementasi dengan menekankan pada *state*, perilaku (*behavior*), dan interaksi objek-objek dalam model tersebut. [14]

Konsep *object oriented* memiliki karakteristik utama yaitu :

1. Enkapsulasi (*Encapsulation*), sebuah prinsip yang digunakan ketika membangun struktur program secara keseluruhan yang mana setiap komponen dari program dibungkus (pembungkusan atribut dan metode dalam satu even).
2. Pewarisan (*Inheritance*), merupakan sarana untuk menghilangkan penulisan ulang terhadap kode yang dapat digunakan berulang kali.
3. Kebanyakrapaan (*Polimorphism*), suatu kondisi dimana dua *object* atau lebih mempunyai antarmuka yang identik namun mempunyai perilaku berbeda.

2.4 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah “Keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).”[11]

Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corp. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

UML dideskripsikan oleh beberapa diagram, diantaranya:

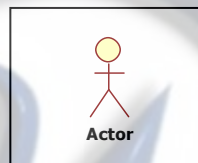
1. Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case diagram* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Komponen-komponen yang terlibat dalam *use case* diagram :

a. Aktor

Pada dasarnya aktor bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan *stick man*.

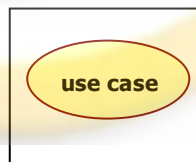


Sumber <http://id.wikipedia.org/wiki/UML>[15]

Gambar 2.1 Aktor

b. Use Case

Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun.



Sumber <http://id.wikipedia.org/wiki/UML>[15]

Gambar 2.2 Use Case

Ada beberapa relasi yang terdapat pada *use case* diagram:

- 1) *Association*, menghubungkan link antar element.
- 2) *Generalization*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
- 3) *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.

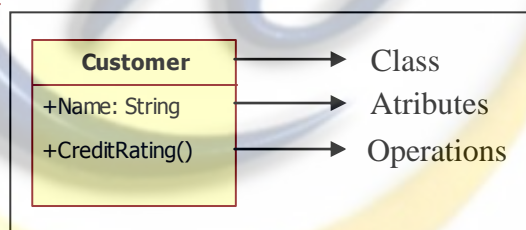
- 4) *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi yang mungkin terjadi pada *use case* diagram:

- 1) *<<include>>*, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
- 2) *<<extends>>*, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
- 3) *<<communicates>>*, merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara aktor dan *use case*.

c. *Class Diagram*

Class adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain.[15]



Sumber [http://id.wikipedia.org/wiki/Class diagram](http://id.wikipedia.org/wiki/Class_diagram)[15]

Gambar 2.3 *Class Diagram*

Class memiliki tiga area pokok :

- 1) Nama (*Class Name*)
- 2) Atribut
- 3) Metode (*Operations*)

Pada UML, *class* digambarkan dengan segi empat yang dibagi beberapa bagian. Bagian atas merupakan nama dari *class*. Bagian yang tengah merupakan struktur dari *class* (atribut) dan bagian bawah merupakan sifat dari *class* (metode/operasi).

Atribut dan metode dapat memiliki salah satu sifat berikut :

- 1) *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
- 2) *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan *class* lain yang mewarisinya.
- 3) *Public*, dapat dipanggil oleh *class* lain.

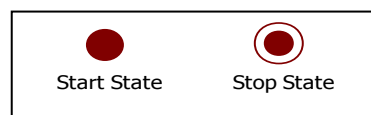
Hubungan antar *Class* :

- 1) Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.
- 2) Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
- 3) Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya serta bisa menambahkan fungsionalitas baru. Sehingga *class* tersebut disebut anak dari *class* yang diwarisinya.
- 4) Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

d. *Statechart Diagram*

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. *Statechart* diagram tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

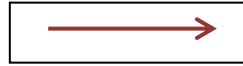
State adalah sebuah kondisi selama kehidupan sebuah objek atau ketika objek memenuhi beberapa kondisi, melakukan beberapa aksi atau menunggu sebuah *event*. *State* dari sebuah objek dapat dikarakteristikan oleh nilai dari satu atau lebih atribut-atribut dari *class*. *State* dari sebuah objek ditemukan dengan pengujian/pemeriksaan pada atribut dan hubungan dari objek. Notasi UML untuk *state* adalah persegi panjang/bujur sangkar dengan ujung yang dibulatkan.



Sumber : <http://www.agilemodeling.com/style/StatechartDiagram.htm>[16]

Gambar 2.4 Start State dan Stop State

Masing-masing diagram harus mempunyai satu dan hanya satu *start state* ketika objek mulai dibuat. Sebuah objek boleh mempunyai banyak *stop state*.



Sumber : <http://www.agilemodeling.com/style/StatechartStateTransitionDiagram.htm>[17]

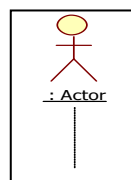
Gambar 2.5 State Transition

Sebuah *state transition* dapat mempunyai sebuah aksi dan/atau sebuah kondisi penjaga (*guard condition*) yang terasosiasi dengannya, dan mungkin juga memunculkan sebuah *event*. Sebuah aksi adalah kelakuan yang terjadi ketika *state transition* terjadi. Sebuah *event* adalah pesan yang dikirim ke objek lain di sistem. Kondisi penjaga adalah ekspresi *boolean* (pilihan Ya atau Tidak) dari nilai atribut-atribut yang mengizinkan sebuah *state transition* hanya jika kondisinya benar. Kedua aksi dan penjaga adalah kelakuan dari objek dan secara tipikal menjadi operasi.

e. **Sequence Diagram**

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

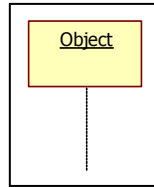
Dibawah merupakan simbol yang digunakan pada *sequence diagram* :



Sumber : <http://www.agilemodeling.com/style/sequenceDiagram.htm>[18]

Gambar 2.6 Aktor

Actor adalah pesan dari seseorang atau sistem lain yang bertukar informasi dengan sistem yang lainnya, kemudian lifeline berhenti atau mulai pada titik yang tepat.



Sumber : <http://www.agilemodeling.com/style/sequenceDiagram.htm>[18]

Gambar 2.7 Object Lifeline

Object lifeline menunjukkan keberadaan dari sebuah objek terhadap waktu. Yaitu objek dibuat atau dihilangkan selama suatu periode waktu diagram ditampilkan, kemudian lifeline berhenti atau mulai pada titik yang tepat.



Sumber : <http://www.agilemodeling.com/style/sequenceDiagram.htm>[18]

Gambar 2.8 Activation

Activation menampilkan periode waktu selama sebuah objek atau aktor melakukan aksi. Dalam *object lifeline*, *activation* berada diatas *lifeline* dalam bentuk kotak persegi panjang, bagian atas dari kotak merupakan inisialisasi waktu dimulainya suatu kegiatan dan yang dibawah merupakan akhir dari waktu.



Sumber : <http://www.agilemodeling.com/style/sequenceDiagram.htm>[18]

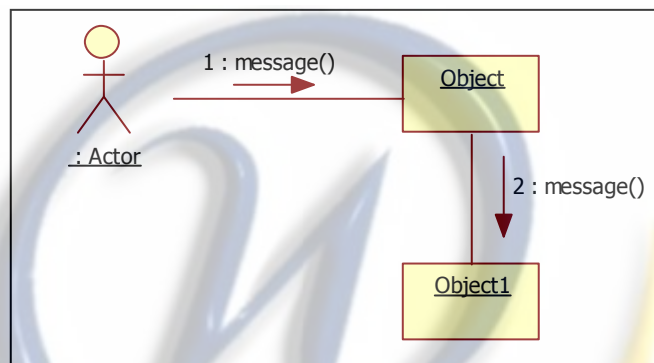
Gambar 2.9 Message

Message adalah komunikasi antar objek yang membawa informasi dan hasil pada sebuah aksi. *Message* menyampaikan dari *lifeline* sebuah objek kepada *lifeline* yang lain, kecuali pada kasus sebuah *message* dari objek kepada objek itu

sendiri, atau dengan kata lain *message* dimulai dan berakhir pada *lifeline* yang sama.

f. *Collaboration Diagram*

Diagram ini menggambarkan interaksi objek yang diatur objek sekelilingnya dan hubungan antara setiap objek dengan objek yang lainnya. Dalam menunjukkan pertukaran pesan, *collaboration diagram* menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagram*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

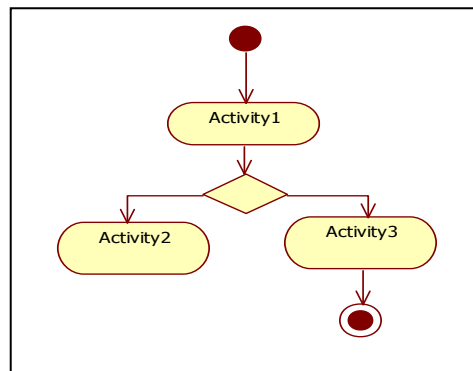


Sumber : <http://www.agilemodeling.com/style/collaborationDiagram.htm>[19]

Gambar 2.10 Collaboration Diagram

g. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*.



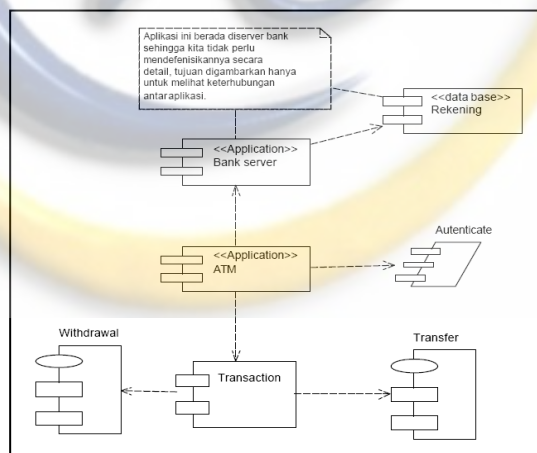
Sumber : <http://www.agilemodeling.com/style/activityDiagram.htm>[20]

Gambar 2.11 Activity Diagram

h. Component Diagram

Menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.



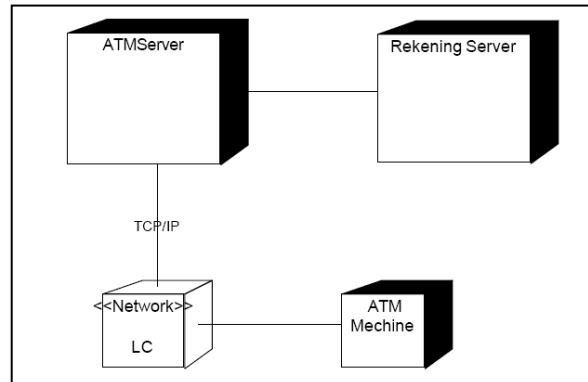
Sumber : <http://www.agilemodeling.com/style/componentDiagram.htm>[21]

Gambar 2.12 Component Diagram

i. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executable component* dan objek

yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.



Sumber : <http://www.agilemodeling.com/style/deploymentDiagram.htm>(22)

Gambar 2.13 *Deployment Diagram*

2.5 *Object Oriented Programming (OOP)*

Object Oriented Programming (OOP), artinya “pengelolaan program sepenuhnya diarahkan pada pembentukan objek (termasuk tipe data, kecuali tipe dasar : *int, float, double, char*), program akan lebih mudah dikembangkan karena sifatnya yang lebih modular.” [6]

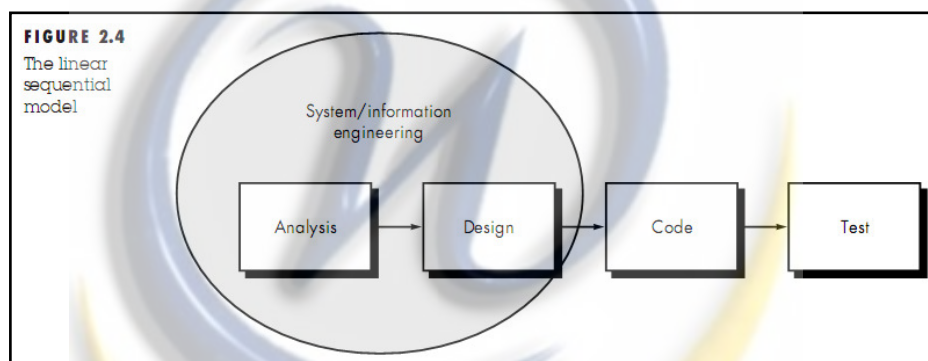
Pemrograman prosedural murni yang tidak menerapkan konsep *object oriented* (karena ada bahasa pemrograman prosedural yang juga sudah berorientasi objek, meskipun belum sepenuhnya) banyak menitikberatkan ke arah pembentukan fungsi-fungsi, sehingga di dalam program akan terdapat banyak sekali fungsi dan variabel yang menyulitkan pemrogram untuk mengelola dan mengembangkannya. Oleh karena itu, dengan memperhatikan kekurangan-kekurangan tersebut, maka dibentuklah bahasa yang menerapkan pendekatan *object oriented* untuk menyederhanakan fungsi-fungsi dan variabel-variabel ke dalam bentuk objek.

Dalam OOP dibutuhkan *memory* lebih besar dibandingkan dengan program prosedural (tradisional). Dua objek yang identik akan memerlukan dua area *memory* berbeda walaupun dari sisi data dan proses keduanya memiliki jumlah dan jenis yang sama. Hal ini disebabkan karena data dan proses pada kedua objek tersebut dipisahkan oleh komputer.

Secara garis besar yang menjadi ciri dari OOP adalah adanya proses abstraksi (*abstraction*), pengkapsulan (*encapsulation*), penurunan sifat (*inheritance*), dan polimorfisme (*polymorphism*) pada objek-objek yang dibentuk.

2.6 Linear Sequential Model

Terkadang disebut *Classic Life Cycle* atau *Waterfall Model*, yaitu yang mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh *analisis, desain, kode, pengujian, dan pemeliharaan*. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut :[4]



(Sumber Roger S. Pressman [4])

Gambar 2.14 Pemodelan Linear Sequential atau Waterfall

1. Rekayasa dan Pemodelan Sistem/ Informasi (*System/ Information Engineering and Modeling*).

Karena perangkat lunak selalu merupakan bagian dari sebuah sistem (bisnis) yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain seperti perangkat lunak, manusia, dan database. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan jumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisa Kebutuhan Perangkat Lunak (*Software Requirements Analysis*)

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perencana perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja, dan antar muka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain (*Design*)

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda :

Struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat/ kebutuhan dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasi dan menjadi bagian dari konfigurasi perangkat lunak.

4. Pengkodean (*Coding*)

Desain harus diterjemahkan dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian (*Testing*)

Sesudah kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional -- yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil actual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan (*Support*)

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan/ user (pengecualian yang mungkin adalah kepada perangkat lunak yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan didalam lingkungan eksternalnya (contohnya

perubahan yang dibutuhkan sebagai akibat perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

Mengapa model ini sangat populer ? Selain karena pengaplikasian menggunakan model ini mudah, kelebihan dari model ini adalah ketika semua kebutuhan sistem dapat didefinisikan secara utuh, eksplisit, dan benar di awal *project*, maka SE dapat berjalan dengan baik dan tanpa masalah. Meskipun seringkali kebutuhan sistem tidak dapat didefinisikan secara eksplisit yang diinginkan, tetapi paling tidak, problem pada kebutuhan sistem di awal *project* lebih ekonomis dalam hal uang (lebih murah), usaha, dan waktu yang terbuang lebih sedikit jika dibandingkan problem yang muncul pada tahap-tahap selanjutnya.

Meskipun demikian, karena model ini melakukan pendekatan secara urut / sequential, maka ketika suatu tahap terhambat, tahap selanjutnya tidak dapat dikerjakan dengan baik dan itu menjadi salah satu kekurangan dari model ini. Selain itu, ada beberapa kekurangan pengaplikasian model ini, antara lain adalah sebagai berikut:

1. Ketika problem muncul, maka proses berhenti, karena tidak dapat menuju ke tahapan selanjutnya. Bahkan jika kemungkinan problem tersebut muncul akibat kesalahan dari tahapan sebelumnya, maka proses harus membenahi tahapan sebelumnya agar problem ini tidak muncul. Hal-hal seperti ini yang dapat membuang waktu pengerjaan SE.
2. Karena pendekatannya secara sequential, maka setiap tahap harus menunggu hasil dari tahap sebelumnya. Hal itu tentu membuang waktu yang cukup lama, artinya bagian lain tidak dapat mengerjakan hal lain selain hanya menunggu hasil dari tahap sebelumnya. Oleh karena itu, seringkali model ini berlangsung lama pengerjaannya.
3. Pada setiap tahap proses tentunya dikerjakan sesuai spesialisasinya masing-masing. Oleh karena itu, ketika tahap tersebut sudah tidak dikerjakan, maka sumber dayanya juga tidak terpakai lagi. Oleh karena itu, seringkali pada

model proses ini dibutuhkan seseorang yang “multi-skilled”, sehingga minimal dapat membantu pengerjaan untuk tahapan berikutnya.

Masing-masing dari masalah tersebut bersifat riil. Tetapi paradig siklus kehidupan klasik memiliki tempat yang terbatas namun penting di dalam kerja rekayasa perangkat lunak. Paradima itu memberikan *template* di mana metode analisis, desain, pengkodean, pengujian, dan pemeliharaan bisa dilakukan. Siklus kehidupan klasik tetap menjadi model bagi rekaya perangkat lunak yang paling luas dupakai. Skalaupun memiliki kelemahan, secara signifikan dia lebih baik dari pada pendekatan yang sifatnya asal kepada pengembang perangkat lunak.

2.7 Bahasa Pemograman yang Digunakan

Adapun bahasa pemograman yang digunakan untuk mendukung adalah :

2.7.1 PHP

PHP merupakan bahasa *script* yang digunakan untuk membuat halaman *web* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan. Oleh karena itu, spesifikasi *server* lebih berpengaruh pada eksekusi dari *script php* daripada spesifikasi *client*. Namun tetap diperhatikan bahwa halaman *web* yang dihasilkan tentunya harus dapat dibuka oleh *browser* pada *client*. Dalam hal ini versi dari html yang digunakan harus didukung oleh *browser client*.

PHP termasuk dalam *Open Source Product*. Jadi dapat merubah *source code* dan mendistribusikannya secara bebas. PHP juga diedarkan secara gratis.

PHP terkenal dengan 4 kelebihanannya :

1. *Practical/Praktis* : PHP adalah bahasa pemrograman yang sangat longgar dalam penulisan, dan ini meningkatkan kepraktisan buat para penggunanya. Programmer tidak diharuskan untuk menuliskan atau menghapus variabel. Walaupun terkadang juga tidak bisa mengatakan dengan mudah yang kemudian dipanggil dalam sebuah skrip, menebak formula terbaik dalam penetapan variabel secara otomatis kemudian menghapus variabel dan mengembalikan *resource* ke sistem setelah skrip berhasil di eksekusi. Pada akhirnya, PHP mampu membuat programmer lebih berpikir pada tujuan akhir dari *project* yang akan dibuat.
2. *Power* : PHP mampu membuat halaman dinamis, memanipulasi form, dan dapat dihubungkan dengan database. Selain yang disebutkan tadi, ternyata PHP juga mampu melakukan hal-hal dibawah ini:

- a. Membuat dan memanipulasi *file* Macromedia Flash, gambar, dan *Portable Document Format* (PDF).
 - b. Mengevaluasi sebuah *password*. Misal: Apakah sebuah *password* yang dibuat, gampang ditebak atau tidak.
 - c. Berkomunikasi dengan LDAP Menguraikan *string*, bahkan yang kompleks sekalipun, dengan menggunakan POSIX dan *Perl Based Regular Expression Library*.
 - d. Berkomunikasi dengan banyak protokol, termasuk IMAP, POP3 dan NNTP.
 - e. Berkomunikasi dengan *credit-card processing solution*.
3. *Possibility* : Jarang ada developer PHP yang terikat pada suatu implementasi pemecahan masalah. Dilain sisi, ada banyak pilihan yang ditawarkan oleh PHP. Contohnya ada pada database yang didukung oleh PHP. Kurang lebih semuanya ada 25 database, termasuk Adabas D, dBase, Empress, *FilePro*, FrontBase, Hyperwave, IBM DB2, Informix, Ingres, Interbase, mSQL, direct MS-SQL, MySQL, Oracle, Ovrimos, PostgreSQL, Solid, Sybase, Unix dbm, dan Velocis. Kemampuan *string*-parsing juga bisa dianggap sebagai banyaknya kemungkinan yang ditawarkan oleh PHP. Dalam hal ini, PHP memiliki lebih dari 85 *function* untuk memanipulasi *string*. Kelebihan ini tak hanya akan menawarkan keleluasaan untuk melakukan operasi *string* yang kompleks, namun juga menjembatani program yang memiliki fungsionalitas yang sama (seperti Python dan Perl) lewat PHP.
4. *Price* : PHP merupakan salah satu *open source software*, yang dapat diartikan sebagai berikut:
- a. PHP dapat dimodifikasi, didistribusikan, dan diintegrasikan dengan produk lain oleh penggunanya.
 - b. Pengembangan dan auditing yang dilakukan secara terbuka.
 - c. Semua orang bebas berpartisipasi

2.7.2 Database

Database atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu

program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system/DBMS*).

Ada beberapa definisi yang umum digunakan dalam basis data, yaitu :

1. *Entitas* : *Entitas* adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang Administrasi Siswa misalnya, siswa, buku, pembayaran.
2. *Atribut* : *Atribut* biasa disebut juga data elemen, data *field*, atau data *item* yang digunakan untuk menerangkan suatu *entitas* dan mempunyai harga tertentu, misalnya *atribut* dari *entitas* pegawai diterangkan oleh, nama, umur, alamat, pekerjaan.
3. *Data Value (Nilai Data)* : *Data Value* adalah data aktual atau informasi yang disimpan pada tiap data, elemen, atau atribut. Atribut nama pegawai menunjukkan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo, dan lain-lain yang merupakan isi data nama pegawai tersebut.
4. *File/Tabel* : Kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.
5. *Record/Tuple* : Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entitas* secara lengkap.

Satu kelebihan utama dari sistem basis data adalah adanya kontrol penuh dan terpusat terhadap data yang terkandung dalam basis data. Dengan adanya kontrol yang terpusat ini, maka permasalahan-permasalahan yang ditemukan dalam sistem *file* dapat ditangani. Misalnya, DBMS dapat mengontrol apakah terdapat duplikasi dalam basis data, meminimalisir duplikasi data, mengatur agar setiap perubahan terhadap sebuah *entry* akan diterapkan secara otomatis terhadap duplikatnya (jika ada), mengontrol hak akses terhadap data, dsb. Satu lagi kelebihan utama sistem basis data adalah *independensi data*, yaitu kondisi dimana jika terjadi perubahan dalam struktur penyimpanan dan/atau teknik pengaksesan data, maka aplikasi yang menggunakan basis data tersebut tidak akan terpengaruh.

Adapun Perangkat lunak yang digunakan dalam pengolahan database sistem informasi antara lain adalah PHP dan MySQL.

2.8 Tools yang Digunakan

Adapun aplikasi yang digunakan untuk mendukung adalah :

2.8.1 Macromedia Dreamweaver

Adobe Dreamweaver merupakan program penyunting halaman *web* keluaran Adobe Systems yang dulu dikenal sebagai Macromedia Dreamweaver keluaran Macromedia. Program ini banyak digunakan oleh pengembang *web* karena fitur-fiturnya yang menarik dan kemudahan penggunaannya. Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh Adobe Systems yaitu versi 8. Versi terakhir Dreamweaver keluaran Adobe Systems adalah versi 10 yang ada dalam Adobe Creative Suite 4 (sering disingkat Adobe CS4).[9]

2.8.2 Xampp

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan *web* server yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat *men-download* langsung dari *web* resminya.[12]

XAMPP adalah singkatan yang masing-masing hurufnya adalah:

X : Program ini dapat dijalankan di banyak sistem operasi, seperti Windows, Linux, Mac OS, dan Solaris.

A : Apache, merupakan aplikasi *web server*. Tugas utama Apache adalah menghasilkan halaman *web* yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman *web*. jika diperlukan juga berdasarkan kode PHP yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman *web* yang dihasilkan.

M : MySQL, merupakan aplikasi database *server*. Perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query Language. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam database.

P : PHP, bahasa pemrograman *web*. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat *web* yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman *web* yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQL. namun PHP juga mendukung sistem manajemen database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya.

P : Perl, bahasa pemrograman.

Mengenal bagian XAMPP yang biasa digunakan pada umumnya:

1. **htdocs** adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
2. **phpMyAdmin** merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka *browser* lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman phpMyAdmin.
3. **Kontrol Panel** yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

2.9 Pengertian Berita

Berita adalah informasi baru atau informasi mengenai sesuatu yang sedang terjadi, disajikan lewat bentuk cetak, siaran, internet, atau dari mulut ke mulut kepada orang ketiga atau orang banyak.[7]

Laporan berita merupakan tugas profesi wartawan, saat berita dilaporkan oleh wartawan laporan tersebut menjadi fakta / ide terkini yang dipilih secara sengaja oleh redaksi pemberitaan / media untuk disiarkan dengan anggapan bahwa berita yang terpilih dapat menarik khalayak banyak karena mengandung unsur-unsur berita.

Stasiun televisi biasanya memiliki acara berita atau menayangkan berita sepanjang waktu. Kebutuhan akan berita ada dalam masyarakat, baik yang melek huruf maupun yang buta huruf.

2.10 Pengertian Pariwisata

Pariwisata merupakan salah satu sektor pembangunan yang saat ini sedang digalakkan oleh pemerintah. Hal ini disebabkan pariwisata mempunyai peran yang sangat penting dalam pembangunan Indonesia khususnya sebagai penghasil devisa Negara disamping sector migas.tujuan pengembangan pariwisata di Indonesia terlihat dengan jelas dalam Intruksi Presiden Repoblik Indonesia No 9 Tahun 1969, khususnya bab II pasal 3,yang menyebutkan”usaha usaha pengembangan pariwisata di Indonesia bersifat suatu pengembangan”industry pariwisata”dan merupakan bagian dari usaha pengembangan dan pembangunan serta kesejahteraan masyarakat dan Negara.[23]

2.10.1 Pariwisata Alternatif

Berdasarkan latar belakang dan isu di atas maka perlu dicarikan jalan keluar agar Pulau Bali tetap mendapat sebutan *The Island of God, The Island of Thousand Temples, The Last Paradise* dan tetap sebagai pulau wisata terbaik dunia (<http://www.balitourismboard.org/>), (cited 20 February 2009). Sesuatu yang perlu dilakukan adalah dengan pengembangan pariwisata alternatif. Pariwisata alternatif adalah suatu bentuk pariwisata yang mengutamakan nilai-nilai alam, sosial dan nilai-nilai masyarakat serta memungkinkan masyarakat lokal dan wisatawan menikmati interaksi yang positif dan bermanfaat serta menikmati pengalaman secara bersama-sama. (Eadington & Smith, 1992:3)

Pariwisata alternatif dapat memberikan sesuatu yang berbeda dengan pariwisata konvensional yang identik dengan pariwisata massal yang telah menyebabkan kebisingan, polusi, dan hal-hal negatif lainnya. Kegiatan-kegiatan pariwisata alternatif dapat berupa: mempelajari sosial budaya orang lokal seperti belajar menari, bahasa, memasak makanan lokal, jalan-jalan menikmati keindahan suasana kehidupan alam pedesaan, dan kegiatan-kegiatan lain yang jauh dari suasana bising dan polusi. (Eadington &Smith, 1992:135)

Beberapa bentuk pariwisata alternatif mencakup kegiatan ekowisata, agrowisata, desa wisata dan wisata spiritual.

1. Ekowisata

Ekowisata merupakan salah satu bentuk wisata alternatif (khusus) yang menaruh perhatian besar terhadap kelestarian sumberdaya wisata atau perjalanan wisata alam yang bertanggung jawab dengan cara mengkonservasi lingkungan dan meningkatkan kesejahteraan masyarakat lokal (TIES, 2000 dalam Damanik, Janianton & Weber, Helmut F. 2006:37). Dari definisi tersebut dapat dikatakan bahwa ekowisata merupakan kegiatan wisata yang bersifat *sustainable* atau berkelanjutan yang dapat membawa keuntungan pada semua aspek baik lingkungan, sosial budaya maupun ekonomi masyarakat lokal.

2. Pariwisata Agro

Pariwisata Agro adalah suatu kegiatan wisata yang menaruh perhatian besar terhadap sektor pertanian dan perkebunan seperti: berwisata ke kebun strawberry, apel, jeruk, kopi, anggur, dan sebagainya. Dalam kegiatan pariwisata agro menekankan pada pengalaman dan belajar tentang tanaman dan dikelola secara baik sehingga dapat membawa dampak positif bagi masyarakat lokal.

3. Pariwisata Pedesaan (Desa Wisata)

Pariwisata Pedesaan merupakan kegiatan wisata yang ditujukan bagi wisatawan yang ingin menikmati suasana pedesaan sebagai tempat untuk istirahat, sebagai tempat belajar budaya suatu daerah (seperti belajar menari, melukis, memahat), dan tempat untuk mendapatkan pengalaman hidup yang berbeda dari daerah asalnya. Menurut Picard (1992) keunikan dan keanekaragaman budaya Bali yang tersebar di desa-desa dapat dimanfaatkan sebagai daya tarik wisata (Picard, 1992).

Bali merupakan sebuah pulau yang memiliki beraneka ragam budaya yang unik dan tidak ada duanya di dunia, mulai dari tari-tarian, kerajinan, adat-istiadat, kebiasaan, agama, yang tidak ada tandingannya di dunia. Sudah terbukti dari jaman dahulu hingga sekarang budaya Bali sangat kesohor keunikannya ke manca negara. Sekarang tinggal bagaimana masyarakat Bali bisa mempertahankan budayanya agar tetap terpelihara dengan baik sehingga pariwisata dapat berkelanjutan.

4. Pariwisata Spiritual

Rogers (2002:3) menyatakan spiritual merupakan jalan kembali ke dasar pluralitas bentuk agama yang menjadi dasar rasional bagi keberagaman tanpa batas pada jalan seseorang di dunia. Spiritualitas adalah hal alami dan

universal dan oleh karenanya tidak dapat hanya dikaitkan dengan budaya agama tertentu. Selanjutnya Dana (2008:21) mendefinisikan wisata spiritual sebagai perjalanan wisata menuju tempat-tempat suci untuk melaksanakan kegiatan spiritual berupa sembahyang, yoga, semadi, meditasi, konsentrasi, dekonsentrasi, dan istilah lainnya sesuai dengan kepercayaan masing-masing. Keberadaan ribuan pura yang tersebar di desa-desa di Bali menjadi aset utama dalam kegiatan pariwisata spiritual. Pura-pura tersebut dapat dijadikan tempat untuk sembahyang, bersemadi, dan meditasi. Kegiatan ini sebaiknya dikoordinasikan dengan para *pemangku* (pendeta) yang bertanggung pada masing-masing pura sehingga bisa berjalan dengan lancar.

2.11 Pengertian Kebudayaan

Kebudayaan sangat erat hubungannya dengan masyarakat. Melville J. Herskovits dan Bronislaw Malinowski mengemukakan bahwa segala sesuatu yang terdapat dalam masyarakat ditentukan oleh kebudayaan yang dimiliki oleh masyarakat itu sendiri. Istilah untuk pendapat itu adalah Cultural-Determinism.[24]

Herskovits memandang kebudayaan sebagai sesuatu yang turun temurun dari satu generasi ke generasi yang lain, yang kemudian disebut sebagai *superorganic*.

Menurut Andreas Eppink, kebudayaan mengandung keseluruhan pengertian nilai sosial, norma sosial, ilmu pengetahuan serta keseluruhan struktur-struktur sosial, religius, dan lain-lain, tambahan lagi segala pernyataan intelektual dan artistik yang menjadi ciri khas suatu masyarakat.

Menurut Edward Burnett Tylor, kebudayaan merupakan keseluruhan yang kompleks, yang di dalamnya terkandung pengetahuan, kepercayaan, kesenian, moral, hukum, adat istiadat, dan kemampuan-kemampuan lain yang didapat seseorang sebagai anggota masyarakat.

Menurut Selo Soemardjan dan Soelaiman Soemardi, kebudayaan adalah sarana hasil karya, rasa, dan cipta masyarakat.

Dari berbagai definisi tersebut, dapat diperoleh pengertian mengenai kebudayaan adalah sesuatu yang akan memengaruhi tingkat pengetahuan dan meliputi sistem ide atau gagasan yang terdapat dalam pikiran manusia, sehingga dalam kehidupan sehari-hari, kebudayaan itu bersifat abstrak.

Sedangkan perwujudan kebudayaan adalah benda-benda yang diciptakan oleh manusia sebagai makhluk yang berbudaya, berupa perilaku dan benda-benda yang bersifat nyata, misalnya pola-pola perilaku, bahasa, peralatan hidup, organisasi sosial, religi, seni, dan lain-lain, yang kesemuanya ditujukan untuk membantu manusia dalam melangsungkan kehidupan bermasyarakat.

