

BAB II

KAJIAN PUSTAKA

A. Deskripsi Teori

1. Sistem Informasi Berbasis *Web*

a. Pengertian sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan untuk menyelesaikan suatu sasaran tertentu. (Jogiyanto, 2005:1). Menurut James A. Hall (2001:5), Sistem adalah sekelompok dua atau lebih komponen-komponen yang berkaitan atau subsistem-subsistem yang bersatu untuk mencapai tujuan yang sama.

Menurut Syahrina Ramadhina (2015) sistem adalah kumpulan elemen-elemen yang saling bekerja sama dan saling terkait untuk memproses suatu *input* yang diterima sistem tersebut dan memproses *input* tersebut sampai menghasilkan *output* yang diinginkan sehingga fungsi sistem yang utama adalah menerima masukan (*input*), memproses masukan (*input*), dan menghasilkan keluaran (*output*).

Dari beberapa pendapat diatas dapat ditarik kesimpulan bahwa sistem adalah kumpulan-kumpulan elemen yang saling berkaitan satu sama lainnya untuk mencapai sebuah tujuan yang sama dengan karakteristik tertentu.

b. Pengertian informasi

Informasi adalah data yang telah diolah menjadi bentuk yang lebih bermanfaat dan menjadi berarti bagi yang menerimanya. Jogiyanto (2005:11) mengemukakan

bahwa data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Informasi dapat dikatakan bernilai apabila manfaatnya lebih efektif jika dibandingkan dengan biaya untuk memperoleh informasi tersebut. Kualitas informasi sangat dipengaruhi atau ditentukan oleh beberapa hal yaitu : Relevan, Akurat, Tepat Waktu, Ekonomis, Efisien, Ketersediaan, Dapat Dipercaya, Konsisten. (Edhy Sutanta, 2009:8).

c. Pengertian sistem informasi

Sistem informasi menurut Alter (1992) adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Sedangkan menurut Bodnar dan Hopwood (1993), sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.

Pengertian sistem informasi menurut Hall (2001) adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai. Sedangkan menurut Wilkinson (1992), sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan.

Jadi, secara umum sistem informasi dapat didefinisikan sebagai suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk

mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

d. *Website*

Website adalah kumpulan dari berbagai halaman web yang tersimpan dalam sebuah domain dimana pada halama-halaman tersebut terdapat informasi di dalamnya. Sebuah website normalnya terbentuk dari banyak halaman web yang saling berkaitan. Hubungan dari suatu halaman web dengan halaman web yang lainnya dikenal dengan sebutan hyperlink, sedangkan tulisan yang dijadikan sarana penghubung dikenal dengan sebutan hypertext.

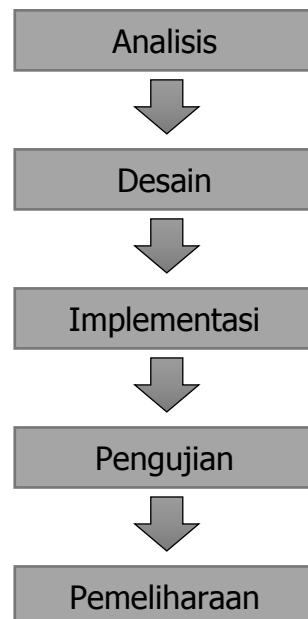
Domain adalah sebuah nama yang bersifat unik yang dimiliki oleh suatu instansi atau organisasi tertentu sehingga bisa diakses secara online melalui jaringan internet, misalnya google.com, detik.com, dan yang lainnya. Untuk memiliki sebuah domain kita diharuskan untuk melakukan registrasi terlebih dahulu pada fasilitator layanan domain tertentu.

Instilah lain yang berkaitan dengan website yang kerap ditemui yaitu Homepage. Homepage merupakan halaman yang pertama kali dimuat ketika mengakses sebuah website. Misalnya, ketika kita membuka website tertentu, halaman pertama yang muncul disebut dengan homepage, lalu ketika kita menuju halaman-halaman yang tersedia pada menu, halaman tersebut dikenal dengan web page.

2. Model Pengembangan

Model pengembangan dari sistem informasi yang akan dikembangkan berpedoman pada model pengembangan perangkat lunak dari Martin Fowler yaitu Waterfall Model. Martin Fowler (2004: 18) mengemukakan bahwa model *waterfall* merupakan salah satu model pengembangan perangkat lunak yang dilakukan secara sekuensial. Dalam mengembangkan sebuah perangkat lunak menggunakan model *waterfall*, pengembangan dilakukan secara sistematis dan berurutan mengikuti alur mulai dari analisis, desain, implementasi, dan pengujian (Martin Fowler, 2004: 18-22).

Alur yang diterapkan pada model air terjun secara berurutan dimulai dari analisis, desain, implementasi, dan pengujian. Berikut adalah gambar model air terjun:



Gambar 1. Ilustrasi Alur Pengembangan Model *Waterfall*

Deskripsi dari alur pengembangan perangkat lunak menggunakan model *Waterfall* adalah sebagai berikut:

a. Analisis

J. W. Satzinger, R. B. Jackson, & S. D. Burd (2010:4) menuturkan bahwa tahap analisis adalah suatu proses untuk mengetahui dan mengumpulkan informasi secara detil kebutuhan apa saja yang harus ada dalam sebuah sistem informasi . Jika mengarah pada model SDLC, analisis berfungsi untuk mengetahui dan membuat dokumentasi secara detail kebutuhan dan kriteria penggerjaan sistem (J. W. Satzinger, R. B. Jackson, & S. D. Burd, 2010:40).

b. Desain

Menurut J. W. Satzinger, R. B. Jackson, & S. D. Burd (2010:40), tahap desain adalah proses penentuan secara detail komponen apa saja yang akan diimplementasikan pada sistem informasi. Tahap desain berdasarkan model pengembangan SDLC berfungsi untuk membuat rancangan solusi pada suatu sistem dengan kriteria yang ditetapkan dan keputusan yang sudah ditentukan pada tahap analisis (J. W. Satzinger, R. B. Jackson, & S. D. Burd, 2010:40).

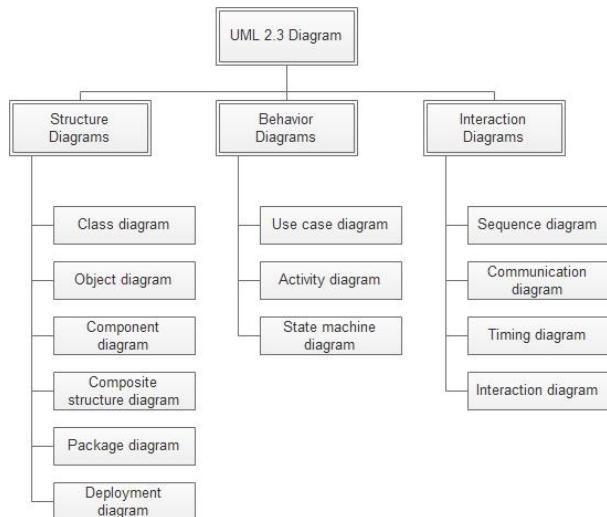
1) Desain sistem menggunakan UML pada *Waterfall Model*

a) Pengertian UML

UML adalah visualisasi dari pemodelan dan komunikasi dari sebuah sistem dalam bentuk diagram dan kode-kode pendukung. Fungsi dari UML hanya terbatas pada pemodelan, sehingga penggunaan UML tidak fokus pada metode tertentu meskipun pada umumnya UML digunakan hanya pada metode berorientasi obyek.

b) Diagram UML

Pada UML 2.3 terdapat 13 jenis diagram yang digolongkan dalam 3 kategori. Pembagian kategori dan jenis-jenis diagram tersebut dapat dilihat pada Gambar 2.



Gambar 2. Kategori Diagram UML

Dari semua jenis diagram tersebut, diagram dapat dibuat seluruhnya maupun hanya sebagian saja sesuai dengan kebutuhan pada perancangan model sistem yang akan dikembangkan. Semakin lengkap fitur sistem yang dikembangkan maka semakin banyak diagram UML yang dibuat. Rizki dkk (2012: 81).

- *Use case diagram*

Shalahuddin (2014: 155) mengemukakan bahwa diagram *use case* adalah pemodelan untuk tingkah laku (behavior) sistem informasi ingin dikembangkan. Interaksi antara aktor dengan sistem informasi yang akan dikembangkan dideskripsikan oleh diagram *use case*. Sedangkan menurut Hamilton (2006: 20), diagram *use case* merupakan diagram yang memodelkan interaksi antara sistem dengan pengguna.

Kriteria penamaan pada diagram *use case* yaitu nama dituliskan sesederhana mungkin serta mudah untuk dipahami.

- *Activity diagram*

Activity diagram adalah gambaran dari *workflow* (aliran kerja) atau aktivitas dari sebuah sistem. Diagram aktivitas menggambarkan aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan oleh aktor. Shalahuddin (2014: 161). Sedangkan menurut Fowler (2004: 163), *Activity diagram* merupakan suatu diagram yang berisi gambaran mengenai model alur kerja, prosedur dan skenario dari sebuah sistem.

- *Class diagram*

Class diagram merupakan diagram yang berisi gambaran dari struktur kelas-kelas yang akan diimplementasikan pada sistem yang akan dibuat. (Shalahuddin, 2014: 141). Di dalam kelas terdapat dua komponen yaitu atribut dan operasi atau metode. Atribut adalah variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi yang dimiliki oleh kelas tersebut.

- *Sequence diagram*

Sequence diagram merupakan gambaran dari tingkah laku obyek yang ada pada *use case* yang berisi deskripsi dari waktu hidup (*life time*) dari obyek dan pesan atau data yang dikirim serta diterima dari setiap obyek tersebut (Shalahuddin, 2014: 165). Menurut Fowler (2004: 81), diagram sekuen atau *sequence diagram* adalah proses menggambarkan perlakuan obyek yang terdapat pada *use case* dengan memberikan deskripsi setiap perlakuan tunggal antara aktor dan sistem dengan lebih mendetail.

2) Desain basis data

Desain data mentransformasikan model domain informasi yang dibuat selama analisis ke dalam struktur data yang akan diperlukan untuk mengimplementasi perangkat lunak. Desain data diimplementasikan dalam bentuk entity relationship diagram (ERD), database logik, dan database fisik.

3) Desain antarmuka (*interface*)

Desain interface menggambarkan bagaimana sistem berkomunikasi dengan pengguna.

c. Implementasi

Pada tahap ini, desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain. Tahap ini juga terdapat proses uji coba pada masing-masing unit sehingga dapat diketahui bagian atau unit mana yang harus diperbaiki sebelum melangkah ke tahap selanjutnya.

1) *Object oriented programming* (OOP)

Object Oriented Programming atau biasa disebut dengan pemrograman berorientasi obyek memberikan gambaran mengenai sebuah perangkat lunak atau *software* yang dijadikan sebagai kumpulan obyek-obyek yang saling berkomunikasi dalam suatu sistem.

Untuk membuat sebuah produk dengan teknik pemrograman berorientasi obyek, *programmer* membuat fungsi-fungsi berdasarkan tanggung jawabnya dimana pembagian tanggung jawab ditetapkan pada kelas yang dibangun. Setiap kelas

menyediakan pelayanan untuk mengerjakan sebuah operasi tertentu. Operasi tersebut dilakukan oleh obyek-obyek yang dibuat dari kelas tersebut.

Keuntungan dalam menggunakan pendekatan *Object Oriented Programming* mudah untuk dikelola kompleksitas produknya dan mudah untuk dilakukan perubahan serta penambahan fitur dari produk tersebut.

Beberapa istilah penting dalam *Object Oriented Programming* yang harus diketahui yaitu:

a) *Abstraction*

Abstraction atau lebih dikenal dengan abstraksi adalah sebuah cara dalam menentukan karakteristik, sifat atau informasi penting dari suatu obyek yang akan ditampilkan dan mana yang tidak ditampilkan.

b) *Object*

Elemen dasar dari konsep *Object Oriented Programming* adalah obyek itu sendiri. *Object* atau obyek merupakan abstraksi dari sesuatu dalam dunia nyata. Kecenderungan pada obyek selalu terkandung *attribute* dan *method* di dalamnya.

Attribute adalah data yang terdapat pada obyek, sedangkan *method* adalah operasi-operasi yang disediakan oleh obyek untuk mengakses atau melakukan modifikasi atau perubahan terhadap *attribute* yang dimiliki.

c) *Class*

Class merupakan sekumpulan obyek yang memiliki kesamaan keadaan dan perilaku. *Class* berperan sebagai sarana pengapsulan kumpulan data dan kumpulan

method. Kumpulan *method* berfungsi untuk melakukan operasi data pada *class* tersebut.

d) *Inheritance*

Inheritance atau lebih dikenal dengan pewarisan merupakan *class* yang dibuat berdasarkan *class* yang sudah ada sebelumnya. Dalam hal ini dapat dikatakan bahwa *inheritance* merupakan *class* baru yang dibuat dengan mewarisi sifat-sifat dari *class* sebelumnya yang disebut sebagai *subclass*. *Class* yang mewariskan sifat-sifatnya disebut dengan *superclass*.

e) *Polymorphism*

Polymorphism adalah suatu obyek yang memungkinkan untuk memiliki berbagai bentuk yaitu sebagai obyek dari *class*-nya sendiri atau sebagai obyek dari *superclass*-nya.

2) Bahasa pemrograman PHP

Welling dan Thompson (2003: 2) menuturkan bahwa bahasa pemrograman PHP merupakan kumpulan dari kode-kode khusus untuk sisi *server* yang dibangun secara spesifik untuk *web*. Kode-kode tersebut dapat disisipkan ke dalam halaman HTML yang kemudian diproses setiap kali halaman tersebut diakses. Kode-kode php kemudian diterjemahkan oleh *web server* dan memberikan *output* yang dapat dilihat oleh pengguna yang mengakses *web* tersebut.

Pada mulanya PHP adalah singkatan dari *Personal Home Page* (Situs pribadi). PHP pertama kali dirancang pada tahun 1995 oleh Rasmus Lerdorf. Pada masa itu PHP

dipergunakan dalam pengolahan data formulir dari sebuah *web* dan masih dikenal dengan sebutan *Form Interpreted* (FI).

Dibandingkan dengan bahasa pemrograman lain, PHP memiliki beberapa kelebihan di antaranya:

- a) Bahasa pemrograman yang dalam penggunaannya tidak memerlukan kompilasi.
- b) Mudah dalam mengkonfigurasinya dan didukung oleh berbagai *platform* penyedia layanan.
- c) Mudah untuk dikembangkan.
- d) Mudah dipahami.
- e) Tidak memerlukan lisensi dalam penggunaannya.

3) *Framework*

Framework secara sederhana dapat didefinisikan sebagai gabungan dari fungsi atau langkah dan kelas untuk mencapai suatu tujuan yang sudah siap untuk digunakan sehingga memberikan kemudahan dan efisiensi waktu kepada *programmer* dalam membuat program tanpa harus membangun fungsi atau *class* dari awal.

Framework berbeda dengan *Content Management System* (CMS), karena CMS cukup diinstall dan dijalankan saja, sedangkan *framework* tidak demikian. Menggunakan *framework* untuk membuat program, pengguna atau *programmer* diharuskan melakukan *coding* atau penulisan kode-kode program pada lingkungan *framework* tersebut. Hal penting yang perlu disadari adalah *programmer* harus mengerti dan memahami terlebih dahulu tentang alur *framework* yang akan digunakan.

Sebagian besar framework yang telah ada mengimplementasikan pola desain atau Model-View-Controller (MVC), yang memisahkan bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (tampilan). Menurut Hofmeister (1999), pola MVC terbukti efektif untuk generasi modul. Adapun komponen MVC adalah sebagai berikut:

- a) *Model*, merupakan bagian yang menangani hal yang berhubungan dengan pengolahan dan manipulasi data, seperti menambah, merubah, mengambil, dan menghapus data yang ada pada basis data.
 - b) *View*, merupakan bagian yang mengatur tampilan sistem informasi yang digunakan untuk berinteraksi dengan pengguna.
 - c) *Controller*, merupakan bagian yang menghubungkan Model dan View secara langsung.
- 4) *Framework Codeigniter*

Codeigniter adalah sebuah *framework* PHP yang dikatakan mempunyai waktu eksekusi lebih cepat bila dibandingkan dengan *framework* PHP yang lain. Dalam *Codeigniter* tidak terdapat lisensi dan menggunakan konsep MVC (*Model View Controller*), dimana konsep tersebut merupakan pola konsep *modern framework* yang banyak diaplikasikan saat ini. *Codeigniter* pertama kali dikembangkan pada tahun 2006 oleh Rick Ellis pendiri *EllisLab.com*.

Dalam mengembangkan *framework codeigniter* Rick Ellis memiliki tujuan yaitu untuk membentuk sebuah struktur yang dapat dipergunakan dalam mengembangkan sebuah *website* agar dapat terselesaikan dengan lebih cepat. Langkah yang dilakukan

yaitu dengan membangun berbagai *library* yang diperlukan dalam pengembangan *website* sesuai dengan kebutuhan.

Kelebihan menggunakan *framework codeigniter* dalam mengembangkan sebuah *website*, di antaranya yaitu:

- a) Bebas digunakan tanpa lisensi
 - b) Dapat dijalankan di berbagai *platform*
 - c) Penggunaan waktu yang lebih efisien
 - d) Menggunakan konsep MVC
 - e) Mudah digunakan.
 - f) *Framework* yang lengkap
 - g) Mudah dipelajari karena terdapat dokumentasi mengenai cara penggunaan *codeigniter*
 - h) Dapat dijalankan pada PHP versi 4 ke atas
- 5) MySQL

SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server database*. Semenjak tahun 70-an, bahasa ini telah dikembangkan oleh IBM, yang kemudian diikuti dengan adanya Oracle, Informix, dan Sybase. Dengan SQL, proses akses *database* menjadi lebih *user friendly* dibandingkan dengan misalnya dBBase ataupun Clipper yang masih menggunakan perintah-perintah pemrograman murni.

MySQL adalah sebuah *server database* SQL *multiuser* dan *multi-threaded*. SQL sendiri adalah salah satu bahasa pemrograman *database* yang paling populer di dunia.

Implementasi program *server database* ini adalah program daemon ‘mysqld’ dan beberapa program lain serta beberapa pustaka.

MySQL adalah *database server* yang sangat ideal untuk data segala ukuran. Dengan kemampuannya yang dapat bekerja di lingkungan Unix maupun Win32 dan sifat yang *Open Source Freeware* (di bawah kungkungan GNU, *General Public License*), MySQL menjadi pilihan yang tepat bagi pengembang aplikasi kelas menengah ke bawah dan kelas korporat. Kemampuan paling menonjol MySQL *server* adalah dalam hal kecepatannya yang sangat tinggi dalam melakukan proses data, *multi-threaded*, *multi-user*, dan sangat mudah dalam melakukan *query* dibandingkan SQL *server* yang lain.

Beberapa keuntungan dalam menggunakan *database MySQL server* antara lain:

- a) MySQL merupakan program yang *multi-threaded*, sehingga dapat dipasang pada *server* yang memiliki *multi-CPU*.
- b) Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, TCL APIs, dan lain-lain.
- c) Bekerja pada berbagai *platform*. (Tersedia berbagai versi untuk berbagai sistem operasi).
- d) Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem basis data.
- e) Memiliki sistem keamanan yang cukup baik dengan verifikasi *host*.
- f) Mendukung ODBC untuk sistem operasi *Microsoft Windows*.

- g) Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi, dan masih banyak keunggulan lainnya.
- h) MySQL merupakan *software* yang gratis untuk digunakan.
- i) MySQL dan PHP saling. Maksudnya adalah pembuatan *database* menggunakan sintak PHP dapat dibuat. Sedangkan *input* yang dimasukkan melalui aplikasi *web* yang menggunakan *script server-side* seperti PHP dapat langsung dimasukkan ke *database* MySQL yang ada di *server* dan tentunya *web* tersebut berada di sebuah *web server*.

Sebagai *server database* dengan konsep *database* modern, MySQL memiliki banyak sekali kelebihan. Beberapa kelebihan yang dimiliki MySQL sebagai berikut:

a) *Portability*

Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP-Unix, dan lain-lain.

b) *Open source*

MySQL merupakan *database open source* (gratis), dibawah lisensi GPL sehingga pengguna dapat memperoleh dan menggunakannya secara cuma-cuma tanpa harus membayar.

c) *Multi user*

MySQL merupakan *database* yang dapat digunakan untuk menangani beberapa *user* dalam waktu bersamaan tanpa mengalami masalah atau konflik. MySQL memungkinkan sebuah *database server* dapat diakses *client* secara bersamaan pula.

d) *Performance tuning*

MySQL mempunyai kecepatan yang cukup baik dalam menangani *query-query* sederhana, serta mampu memproses lebih banyak SQL per satuan waktu.

e) *Column type*

Database MySQL didukung dengan tipe data yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, data, time, datetime, timestamp, year, set serta enum.

f) *Command dan function*

MySQL *server* memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

g) *Security*

Sistem *security* pada MySQL mempunyai beberapa lapisan sekuritas seperti *level subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

h) *Scalability dan limits*

MySQL mempunyai kemampuan menangani *database* dalam skala cukup besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu dapat menampung indeks sampai 32 indeks pada tiap tabelnya.

i) *Connectivity*

Adanya kemampuan MySQL melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix Socket (Unix), atau Named Pipes (NT).

d. Pengujian

Sebuah perangkat lunak perlu dijaga kualitasnya dimana kualitas bergantung pada kepuasan pelanggan (*customer*). Perangkat lunak banyak mengandung kesalahan (*error*) pada proses-proses tertentu pada saat digunakan oleh *user*. Kesalahan-kesalahan (*error*) pada perangkat lunak tersebut disebut dengan “*bug*”. Untuk menghindari banyaknya *bug* maka diperlukan adanya pengujian perangkat lunak sebelum perangkat lunak diberikan ke pelanggan atau selama perangkat lunak masih terus dikembangkan (Rosa dan Shalahuddin, 2011:209).

Pengujian adalah satu *set* aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan dan menempatkan desain kasus uji yang spesifik dan metode pengujian (Rosa dan Shalahuddin, 2011:210). Pengujian perangkat lunak adalah sebuah elemen pada sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) (V&V). Verifikasi mengacu pada aktifitas yang menjamin perangkat lunak dapat mengimplementasikan sebuah fungsi benar. Validasi mengacu pada aktifitas yang menjamin perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan (*customer*) (Rosa dan Shalahuddin, 2011:211). Tahapan pengujian secara keseluruhan ditampilkan pada Gambar 4.



Gambar 3. Tahapan Pengujian Perangkat Lunak

1) Pengujian Unit (*Unit Testing*)

Pengujian unit fokus pada usaha verifikasi pada unit yang terkecil pada desain perangkat lunak. Setiap unit perangkat lunak diuji agar dapat diperiksa apakah aliran masukan (*input*) dan keluaran (*output*) dari unit sudah sesuai dengan yang diinginkan. Pengujian unit dilakukan saat kode program dibuat. Karena dalam sebuah perangkat lunak banyak memiliki unit-unit kecil maka untuk menguji unit-unit kecil ini biasanya dibuat program kecil (*main program*) untuk menguji unit-unit perangkat lunak (Rosa dan Shalahuddin, 2011:215).

Setiap unit diuji menggunakan sebuah program pengujian yang khusus dibuat untuk menguji sebuah unit menggunakan kumpulan kasus uji yang didefinisikan. Menurut Pressman (2002:581), kompleksitas relatif dari pengujian dan kesalahan dibatasi oleh ruang lingkup batasan yang dibangun untuk pengujian unit. Pengujian unit dilakukan dengan teknik pengujian *whitebox* oleh peneliti.

2) Pengujian Integrasi (*Integration Testing*)

Pengujian integrasi adalah sebuah teknis yang sistematik untuk mengonstruksi struktur program seiring dengan menggabungkan fungsi program dengan antarmukanya. Pengujian ini bertujuan untuk mempergunakan komponen unit program yang sudah diuji dan membangun struktur seperti yang telah didesain sebelumnya (Rosa dan Shalahuddin, 2011:216).

Menurut Wafda Adita (2015:23), setelah melakukan pengujian unit, langkah berikutnya adalah memeriksa bagaimana unit-unit tersebut bekerja sebagai suatu kombinasi, bukan lagi sebagai suatu unit yang individual. Pada tahap pengujian integrasi, peneliti memeriksa hasil dari kombinasi unit-unit tersebut. Peneliti juga memastikan bahwa seluruh kondisi yang mungkin terjadi dari hasil interaksi antarunit tersebut menghasilkan *output* yang diharapkan. Pengujian integrasi dilakukan dengan teknik pengujian *blackbox*.

3) Pengujian Sistem (*System Testing*)

Pengujian sistem adalah sederetan pengujian yang berbeda yang tujuan utamanya adalah sepenuhnya menggunakan sistem berbasis komputer. Meskipun masing-masing pengujian memiliki tujuan yang berbeda, perlu dilakukan pemeriksaan untuk mengetahui apakah semua elemen sistem telah diintegrasikan dengan tepat dan melakukan fungsi yang dialokasikan (Pressman, 2002:596).

Menurut Ian Sommerville (2003:59), pengujian ini berkenaan dengan penemuan kesalahan yang diakibatkan dari interaksi yang tidak diharapkan antara subsistem dan

interface subsistem. Proses ini berhubungan dengan validasi bahwa sistem telah memenuhi persyaratan fungsional dan non-fungsionalnya.

a) Pengujian *Stress* (*Stress Testing*)

Menurut Adi Nugroho (2010:306), pengujian stress mengidentifikasi permasalahan-permasalahan yang mungkin muncul saat sumber daya komputasi yang dimilikinya (misalnya penggunaan waktu kerja prosesor dan alokasi memori) terlampaui.

4) Pengujian Penerimaan (*Acceptance Testing*)

Menurut Ian Sommerville (2003:59), pada *acceptance testing*, menunjukkan kesalahan dan penghapusan definisi persyaratan sistem karena data *real* menjalankan sistem dengan cara yang berbeda dari data uji. Pengujian penerimaan juga dapat mengungkapkan masalah persyaratan dimana fasilitas sistem tidak memenuhi keperluan *user* atau kinerja sistem tidak dapat diterima. Pengujian validasi memiliki beberapa pendekatan sebagai berikut:

a) *Blackbox Testing*

Blackbox testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Rosa dan Shalahuddin, 2011:213). Menurut Didik Hariyanto (2008:154), *blackbox testing* menguji beberapa aspek sistem dengan memerhatikan struktur logika internal perangkat lunak. Pengujian ini memerlukan

fungsi perangkat lunak beroperasi, yaitu saat *input* diterima maka *output* benar. Kedua jenis rancangan pengujian ini bersifat komplementer.

b) *Whitebox Testing*

Whitebox testing adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case* (Pressman, 2002:533). *Whitebox testing* menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan (Rosa dan Shalahuddin, 2011:214). Tahapan pengujian pada penelitian ini kemudian diterapkan pada aspek yang terdapat pada model ISO 9126 untuk menguji kualitas dari sistem informasi yang dikembangkan.

Berdasarkan kajian di atas, pengujian perangkat lunak adalah aktivitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Pengujian dilakukan sebelum produk diserahkan kepada pengguna. Tahapan pengujian pada penelitian ini adalah pengujian verifikasi dan validasi. Pengujian tersebut diterapkan pada aspek yang terdapat pada model ISO 9126.

e. Pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tetapi tidak untuk membuat perangkat lunak baru.

3. Kualitas Perangkat Lunak

Pengujian perangkat lunak adalah bagian yang sangat penting dari dalam menjamin kualitas perangkat lunak dan mempresentasikan kajian utama dari spesifikasi, desain dan pengkodean.

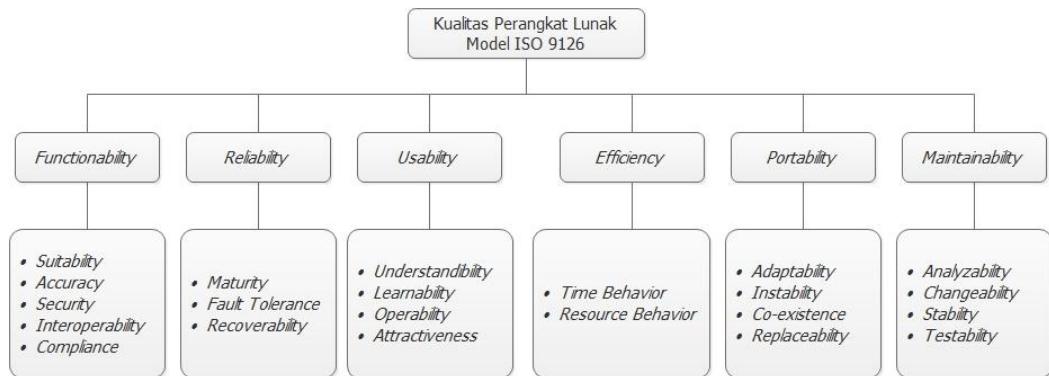
Salah satu barometer kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh International Organization for Standardization (ISO) dan International Electrotechnical Commission (IEC). Kualitas produk, model, karakteristik mutu, dan metrik terkait digunakan untuk menilai dan melakukan penetapan kualitas suatu produk *software* didefinisikan oleh ISO 9126.

Beberapa karakteristik yang ditetapkan oleh ISO 9126 di antaranya yaitu:

- a. *Functionality*, produk perangkat lunak dalam menyediakan fungsi-fungsi yang sesuai dengan kebutuhan sistem yang telah melalui tahap analisis sebelumnya.
- b. *Efficiency*, produk perangkat lunak untuk menyediakan performa sesuai dengan kebutuhan pengguna.
- c. *Reliability*, kemampuan perangkat lunak untuk perawatan dengan level performansi.
- d. *Usability*, kapasitas produk perangkat lunak dalam memberikan kecepatan, kemudahan, kepuasan penggunaan dan menyediakan bantuan mengenai kesalahan yang terjadi sesuai dengan keadaan dan kebutuhan.
- e. *Portability*, kapasitas yang berhubungan dengan kemampuan perangkat lunak yang dikirim ke lingkungan berbeda.

f. *Maintainability*, kapasitas yang diperlukan untuk melakukan perubahan pada perangkat lunak.

Dari keseluruhan karakteristik yang ada pada ISO 9126, digolongkan menjadi beberapa sub-bab seperti yang dipaparkan pada gambar berikut.



Gambar 4. Karakteristik Kualitas Perangkat Lunak ISO 9126

a. *Functionability*

Tabel 1. Sub-karakteristik Aspek *Fucntionality*

Sub-Karakteristik	Deskripsi
<i>Suitability</i>	Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna dalam mengelola nilai rapor.
<i>Accuracy</i>	Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.
<i>Security</i>	Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (hacker) maupun otorisasi dalam modifikasi data.
<i>Interoperability</i>	Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
<i>Compliance</i>	Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.

b. Reliability

Tabel 2. Sub-karakteristik Aspek *Reliability*

Sub-Karakteristik	Deskripsi
<i>Maturity</i>	Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.
<i>Fault tolerance</i>	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak.
<i>Recoverability</i>	Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.
<i>Fault tolerance</i>	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak.

c. Usability

Tabel 3. Sub-karakteristik aspek *Usability*

Sub-Karakteristik	Deskripsi
<i>Understandability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipahami.
<i>Learnability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipelajari.
<i>Operability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.
<i>Attractiveness</i>	Kemampuan perangkat lunak dalam menarik pengguna.

d. Efficiency

Tabel 4. Sub-karakteristik aspek *Efficiency*

Sub-Karakteristik	Deskripsi
<i>Time behavior</i>	Kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya.
<i>Resource behavior</i>	Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.

e. *Portability*

Tabel 5. Sub-karakteristik aspek *Portability*

Sub-Karakteristik	Deskripsi
<i>Adaptability</i>	Kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda.
<i>Instalability</i>	Kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
<i>Coexistence</i>	Kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya.
<i>Replaceability</i>	Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya.

f. *Maintainability*

Tabel 6. Sub-karakteristik aspek *Maintainability*

Sub-Karakteristik	Deskripsi
<i>Analyzability</i>	Kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan.
<i>Changeability</i>	Kemampuan perangkat lunak untuk dimodifikasi tertentu.
<i>Stability</i>	Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
<i>Testability</i>	Kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain.

Dengan memenuhi karakteristik-karakteristik pada ISO 9126 bukan berarti sebuah perangkat lunak mendapatkan sertifikat ISO melainkan perangkat lunak juga harus memenuhi sisi manajemen dari pembuat perangkat lunak tersebut.

B. Kajian Penelitian Yang Relevan

Beberapa penelitian yang relevan dengan penelitian ini antara lain :

1. Pengembangan Sistem Informasi Pengolah Nilai Siswa Kurikulum 2013 Berbasis *Web* SMK N 2 Wonosari oleh Wisnu Eka Rihandana pada tahun 2016 dari Program

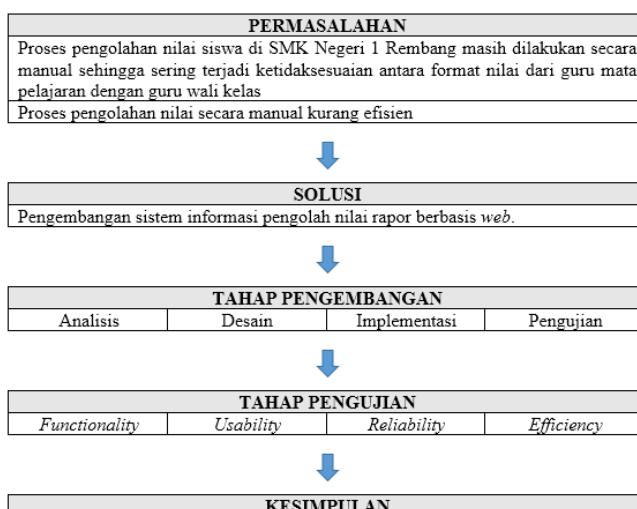
Studi Pendidikan Teknik Informatika, Universitas Negeri Yogyakarta. Penelitian ini bertujuan untuk mengembangkan sistem informasi pengolah nilai siswa berbasis *web* berdasarkan kurikulum 2013 di SMK N 2 Wonosari dan mengetahui kualitas perangkat lunak yang dikembangkan tersebut berdasarkan standar ISO 9126. Penelitian ini menghasilkan sistem informasi pengolah nilai siswa kurikulum 2013 berbasis *web* yang dapat berfungsi untuk membantu proses pengolahan nilai siswa di SMK N 2 Wonosari. Sistem informasi tersebut telah diuji dan memenuhi aspek-aspek standar kualitas ISO 9126 yaitu *functionality, reliability, usability, efficiency, maintainability, dan portability*.

2. Pengembangan dan Analisis Kualitas Sistem Informasi Akademik SMK Negeri 2 Depok Sleman Berbasis *Web* oleh Tika Novita Sari pada tahun 2014 dari Program Studi Pendidikan Teknik Informatika, Universitas Negeri Yogyakarta. Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik SMK Negeri 2 Depok Sleman berbasis *web* dan mengetahui kualitas perangkat lunak yang dikembangkan berdasarkan standar ISO 9126. Hasil dari penelitian ini dapat disimpulkan bahwa penelitian ini menghasilkan perangkat lunak sistem informasi akademik SMK Negeri 2 Depok Sleman untuk kurikulum 2013 dengan menggunakan *framework codeigniter* dan berdasarkan model pengembangan *waterfall*. Hasil uji kualitas perangkat lunak yang dikembangkan memenuhi standar kualitas ISO 9126 yaitu dari aspek *functionality, reliability, usability, efficiency, maintainability, dan portability*.

C. Kerangka Pikir

Sistem informasi nilai rapor berbasis *web* ini dibuat untuk mempermudah proses pengolahan nilai dan pelaporan nilai siswa di SMK Negeri 1 Rembang, Purbalingga. Pengembangan sistem informasi nilai rapor ini menggunakan *framework* PHP *codeigniter* agar mudah dikembangkan kembali di masa yang akan datang. Pembuatan sistem informasi nilai rapor ini menggunakan model *waterfall* sebagai model pengembangannya yang menerapkan pendekatan alur hidup perangkat lunak secara sekuensial atau berurutan mulai dari analisis, desain, pengkodean, dan pengujian.

Tahap pengujian pada perangkat lunak sangat diperlukan agar memperoleh hasil pengembangan perangkat lunak yang berkualitas. Salah satunya yaitu dengan pengujian kelayakan perangkat lunak. Pengujian dilakukan dengan standar ISO 9126 sebagai barometer. Karakteristik yang diujikan pada penelitian ini yaitu *functionality*, *efficiency*, *reliability* dan *usability*. Kerangka pikir dalam penelitian ini dijelaskan melalui Gambar 5.



Gambar 5. Kerangka Pikir

D. Pertanyaan Penelitian

Bersumber pada latar belakang dan tujuan maka peneliti mengajukan pertanyaan sebagai berikut:

1. Apakah penelitian ini dapat menghasilkan produk berupa perangkat lunak Sistem Informasi Nilai Rapor Kurikulum 2013 Berbasis *Web* menggunakan *framework Codeigniter* di SMK Negeri 1 Rembang Purbalingga?
2. Bagaimana kualitas perangkat lunak yang dibuat dari segi *functionality*?
3. Bagaimana kualitas perangkat lunak yang dibuat dari segi *usability*?
4. Bagaimana kualitas perangkat lunak yang dibuat dari segi *reliability*?
5. Bagaimana kualitas perangkat lunak yang dibuat dari segi *efficiency*?