

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. Sistem Informasi

Sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan (Jogiyanto, 2005). Sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur (O'Brien, 2005). Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama (Sutarman, 2009). Sehingga bisa dikatakan, sistem adalah elemen-elemen yang berpadu satu sama lainnya yang pada akhirnya menghasilkan pekerjaan dengan meraih tujuan yang sama.

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan (Irmawati & Indrihapsari, 2014). Informasi adalah sekumpulan fakta (data) yang diorganisasikan dan diproses untuk memperoleh pemahaman, pengalaman, pembelajaran yang terakumulasi sehingga dapat diaplikasikan dalam pemecahan masalah atau proses bisnis tertentu (Sutarman, 2009). Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang

menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan (Jogiyanto, 2005). Informasi sebagai data yang telah diubah menjadi konteks yang berarti dan berguna bagi para pemakai akhir tertentu (O'Brien, 2005).

Sistem informasi mendefinisikan sebagai kombinasi teratur apa pun dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi (O'Brien, 2005). Sistem Informasi mempunyai arti adalah suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian 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 (Hariyanto, 2008). Konklusi dari berbagai pendapat di atas, sistem informasi merupakan himpunan dari berbagai elemen yang terdiri dari kombinasi orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian dalam mengolah data menjadi informasi pada akhirnya mendistribusikan informasi tersebut kepada penggunanya.

1. Rekayasa Perangkat Lunak

Rosa & Shalahuddin (2012:14) mengemukakan bahwa rekayasa perangkat lunak merupakan pembangunan dengan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

Rekayasa perangkat lunak pada dasarnya merupakan aplikasi dari suatu pendekatan yang sistematik, disiplin, dan dapat diukur pada pengembangan, operasi, dan perawatan perangkat lunak yaitu demi penerapan rekayasa pada perangkat lunak (Pressman, 2012:15). Rekayasa perangkat lunak merupakan teknologi yang berlapis seperti ditunjukkan pada Gambar 1.



Gambar 1. Lapisan-lapisan rekayasa perangkat lunak (Pressman, 2012:16)

Menurut apa yang terdapat pada Gambar 1, rekayasa perangkat lunak mencakup di dalamnya berupa peralatan yang diperlukan, metode-metode untuk mengelola, proses didalam merekayasa perangkat lunak serta peralatan yang diperlukan.

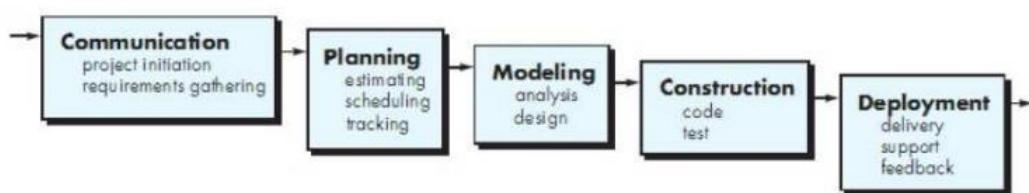
Proses rekayasa perangkat lunak merupakan perekat kuat lapisan-lapisan teknologi dan memungkinkan pengembangan perangkat lunak yang rasional dan tepat waktu. Metode-metode rekayasa perangkat lunak menyediakan berbagai prosedur teknis untuk pengembangan atau membangun perangkat lunak. Peralatan didalam merekayasa perangkat lunak menyediakan dukungan semiotomatis maupun otomatis bagi proses dan metode-metode. Fokus pada kualitas rekayasa perangkat lunak dilakukan untuk memperjelas dan melakukan kegiatan-kegiatan yang diperlukan untuk memastikan kualitas perangkat lunak.

Dengan segala pendekatan rekayasa termasuk rekayasa perangkat lunak harus bersandar pada komitmen organisasional pada peningkatan kualitas (Pressman, 2012:15).

2. Model Pengembangan

SDLC atau *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya yang berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rosa A.S. & Shalahuddin, 2011:24).

Model yang digunakan untuk pengembangan sistem pada penelitian ini menggunakan model *waterfall*. Model *waterfall* menurut Pressman (2012:46), adalah model klasik yang sifatnya sistematis serta berurutan yang digunakan dalam pengembangan perangkat lunak, dimulai dari spesifikasi kebutuhan pengguna (*communication*) dengan alur selanjutnya yaitu perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), dan penyerahan sistem kepada pengguna perangkat lunak (*deployment*). Tahapan pengembangan sistem menggunakan model *waterfall* dapat dilihat pada Gambar 2.



Gambar 2. Model pengembangan *waterfall* (Pressman, 2012:46)

Berikut penjelasan dari masing-masing fase pengembangan model *waterfall* (Pressman, 2012:17):

1. Komunikasi (*communication*)

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan pelanggan agar dalam memahami dan tujuan pengembangan mampu terwujud nyata. Hasil dari tahapan komunikasi berupa inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan pengumpulan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi perangkat lunak yang dikembangkan. Pengumpulan beberapa data tambahan yang bisa diperoleh dari jurnal, artikel, dan internet.

2. Perencanaan (*planning*)

Tahap selanjutnya yaitu tahapan perencanaan penjelasan mengenai estimasi atau perkiraan tugas-tugas teknis seperti apa yang akan dilakukan, risiko-risiko yang barangkali terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja, serta *tracking* proses untuk penggerjaan sistem.

3. Pemodelan (*modeling*)

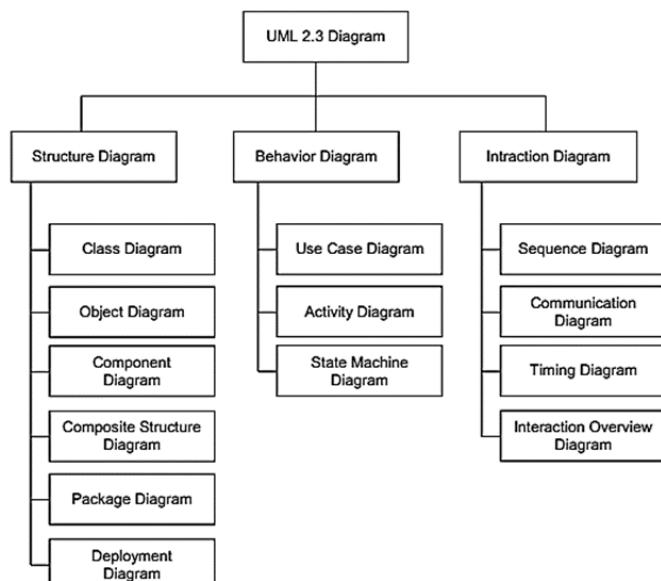
Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang fokusnya ada pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya agar lebih mudah dalam memahami gambaran besar kebutuhan perangkat lunak ataupun rancangan-rancangan dari apa yang akan dikerjakan terutama untuk memenuhi kebutuhan tersebut.

Penggunaan UML merupakan salah satu bentuk pemodelan perangkat lunak yang sesuai dengan konsep pengembangan pemrograman berorientasi objek (Nugroho, 2011:124).

UML yang merupakan akronim dari kata *Unified Modeling Language* ialah bahasa yang digunakan untuk menspesifikasikan, memvisualisasikan, serta mengkonstruksikan bangunan dasar dari sebuah sistem perangkat lunak, juga melibatkan aturan-aturan bisnis (Nugroho, 2011:119).

Menurut Rosa A.S. & Shalahuddin (2011, 113) desain UML adalah salah satu standar bahasa pemodelan yang kerap kali digunakan di dunia industri dalam pendefinisian kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Menurut Sugiarti (2013:36) UML didefinisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem sebelum mulai mengimplementasikannya menjadi perangkat lunak. UML terdiri atas 13 jenis diagram yang digolongkan menjadi tiga kategori. Penggolongan diagram-diagram UML dapat dilihat pada Gambar 3.



Gambar 3. Penggolongan Diagram UML (Rosa A. S. & Shalahuddin, 2011)

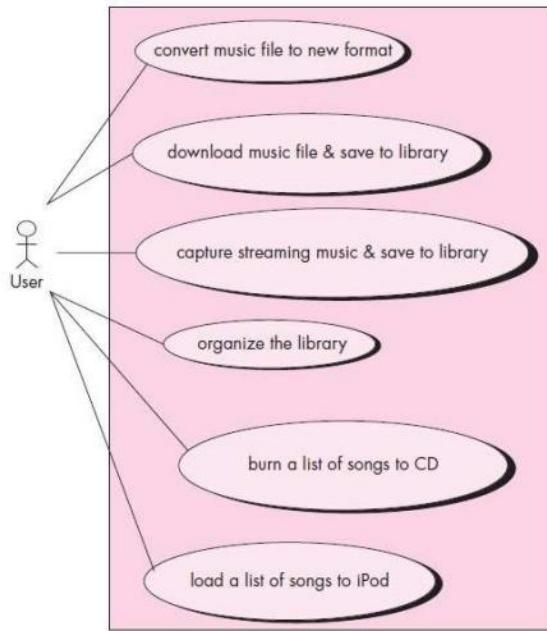
Jenis diagram didalam UML bermacam-macam, namun dalam penelitian ini dibatasi dengan menggunakan empat jenis diagram yang diantaranya *use case diagram*, *class diagram*, *activity diagram* dan *sequence diagram*. Berikut deskripsi masing-masing mengenai empat jenis diagram tersebut :

a) *Use Case Diagram*

Menurut Rosa A.S & Shalahuddin (2011:130), *use case* atau *diagram use case* merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang nantinya dikembangkan.

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang nantinya dikembangkan. Secara garis besarnya, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah aplikasi informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Sugiarti, 2013:41).

Dalam *use case diagram*, *use case* ditampilkan dalam model oval. Para aktor terhubung dengan setiap *use case* menggunakan garis. Karena *use case diagram* menampilkan semua *use case*, hal ini sangat menunjang untuk menentukan bahwa semua fungsi dari sistem sudah terjangkau (Pressman, 2012 : 993). Contoh *use case diagram* dapat dilihat pada Gambar 4.



Gambar 4. Contoh *use case diagram* (Pressman, 2012:993)

b) *Class Diagram*

Class Diagram merupakan diagram yang mengambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Sugiarti, 2013:57).

Dalam memodelkan kelas-kelas, termasuk diantaranya atribut kelas, operasi-operasi, dan hubungan antara satu kelas dengan kelas lainnya, UML menyediakan model *class diagram*. *Class diagram* menyediakan perspektif *view* statis atau struktural atas suatu sistem. *Class diagram* tidak menunjukkan sifat dinamis dari komunikasi antara objek-objek kelas dalam diagram (Pressman, 2010:988).

Elemen-elemen utama suatu *class diagram* adalah kotak-kotak, yang merupakan ikon-ikon yang gunanya untuk merepresentasikan kelas-kelas. Pada bagian setiap kotak dibagi kedalam bagian-bagian horizontal. Bagian atas sendiri berisi label dengan nama *class*. Pada bagian tengah berisi atribut setiap kelas. Setiap atribut

menunjukan pada sesuatu yang merupakan objek pada sebuah kelas (Pressman, 2010:988).

Setiap atribut dapat memiliki nama, jenis dan tingkat visibilitas. Visibilitas ditunjukkan dengan awalan -, #, ~ atau +, yang masing-masingnya menunjukan visibilitas *private*, *protected*, *package* atau *public* (Pressman, 2010:988). Contoh *class diagram* dapat dilihat pada Gambar 5.



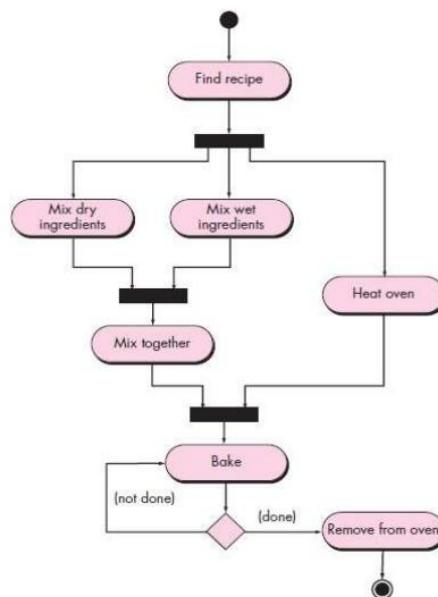
Gambar 5. Contoh *class diagram* (Pressman, 2012:988)

c) *Activity Diagram*

Menurut Sugiarti (2013:75) mendefinisikan *activity diagram* sebagai pemodelan yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. *Activity diagram* menggambarkan aktifitas sistem bukan apa yang dilakukan aktor, akan tetapi aktifitas yang dapat dilakukan sistem.

Komponen utama dari *activity diagram* adalah *action node* (node aksi), yang direpresentasikan dengan persegi panjang dengan sudut bulat, yang berhubungan dengan tugas yang dilakukan oleh sistem *software*. Anak panah dari satu node aksi ke node aksi lainnya menjelaskan adanya aliran kontrol. Dengan artian bahwa setelah aksi pertama selesai dilakukan, aksi kedua baru dijalankan setelahnya.

Lingkaran hitam pekat menguraikan titik awal proses aktifitas dimulai. Titik hitam dengan lingkaran hitam dibagian sisinya menerangkan akhir dari proses aktifitas tersebut. Garis horizontal berwarna hitam yang direpresentasikan sebagai percabangan merupakan pemisah dua aksi atau lebih secara bersamaan atau konkuren (Pressman, 2012:1000). Contoh *activity diagram* dapat dilihat pada Gambar 6.



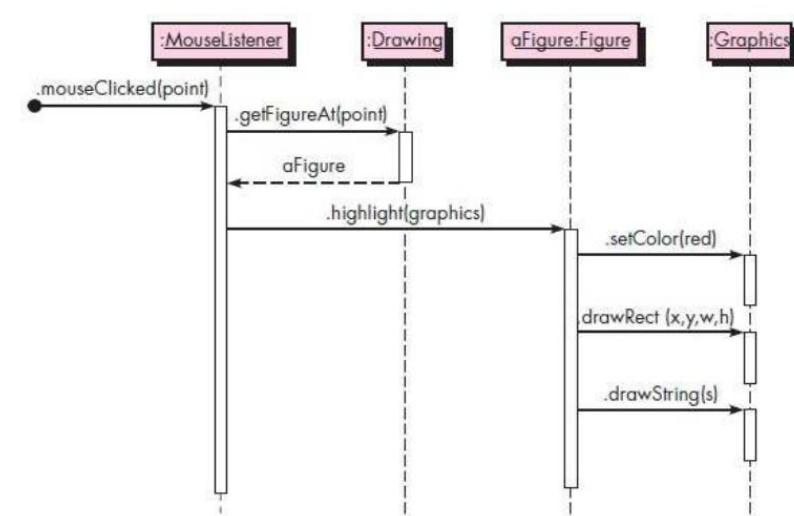
Gambar 6. Contoh *activity diagram* (Pressman, 2012:999)

d) *Sequence Diagram*

Sequence Diagram adalah pemodelan yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Sugiarti, 2011:69). Harus diketahui terlebih dahulu objek-objek yang terlibat dalam sebuah *use case* agar mampu menggambarkan pemodelan *sequence diagram*.

Sequence diagram menampilkan metode yang dipanggil menggunakan anak panah horizontal dari pelaku ke target pelaku, dilabeli dengan nama metode dan

pilihan masukkan berupa parameter, jenis atau jenis timbal balik. Berbeda kasus pada *looping*, *conditional*, dan struktur kontrol lainnya dalam *sequence diagram*, bisa menggunakan *frame-frame* interaksi berbentuk persegi panjang yang mengelilingi bagian-bagian diagram dan dilabeli untuk jenis struktur kontrol yang diwakili (Pressman, 2012:994-996). Contoh *sequence diagram* dapat dilihat pada Gambar 7.



Gambar 7. Contoh *sequence diagram* (Pressman, 2012:996)

4. Konstruksi (*construction*)

Tahapan konstruksi adalah proses aktifitas mengintegrasikan pembentukan kode (*code generation*) dan pengujian yang sangat dibutuhkan untuk menemukan kesalahan-kesalahan atau hal yang cacat di dalam kode program komputer yang dihasilkan sebelumnya. Tahapan inilah yang merupakan tahapan konkret di dalam mengembangkan *software* karena terealisasikan secara nyata.

Tahapan konstruksi yang dilakukan pada penelitian adalah dengan menggunakan bahasa pemrograman PHP, beserta *database* yang menggunakan MySQL. Berikut adalah penjelasan mengenai PHP dan MySQL.

a) PHP

PHP atau yang lebih dikenal sebagai PHP *Hypertext Preprocessor* adalah bahasa yang hanya dapat dijalankan pada sisi server yang hasilnya dapat ditampilkan pada pengguna. PHP merupakan bahasa standar yang digunakan dalam dunia website. Bahasa program berbentuk script yang diletakkan dalam server web. Hal ini berbeda apabila dibandingkan dengan javascript dimana proses berjalannya ada pada sisi *client* atau pada web *browser* (Nugroho, 2004: 140).

Menurut Tom Butler & Kevin Yank (2017) PHP merupakan bahasa skrip yang berada pada sisi *server* sehingga bisa memasukkan instruksi ke halaman web yang akan dieksekusi oleh perangkat lunak web *server* sebelum mengirimkan halaman-halaman itu ke *browser* yang memintanya (Butler & Yank, 2017).

b) MySQL *Database*

Jogiyanto (2005 : 711) mendefinisikan basis data (*database*) merupakan himpunan data yang saling berhubungan satu sama lain, himpunan tersebut tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data merupakan salah satu komponen yang sangat esensial dalam sistem informasi, dikarenakan basis yang menyediakan informasi kepada para pemakainya.

Basis data digunakan sebagai tempat untuk menyimpan data juga sebagai tempat untuk menyediakan data. Basis data juga bisa berarti sekumpulan item data

yang saling berhubungan satu sama lain yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di perangkat keras komputer dan juga menggunakan perangkat lunak untuk melakukan manipulasi yang mempunyai kegunaan tertentu (Hariyanto, 2008).

MySQL adalah sebuah program yang dibuat membuat *database* yang bersifat *open source*, artinya siapa saja berhak untuk menggunakannya dan tidak dicekal. MySQL merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *multi user* (Banyak Pengguna). Dengan menggunakan bahasa query standar yang dimiliki SQL (*Structure Query Language*), maka termasuk bahasa permintaan yang terstruktur yang telah distandarisasi untuk semua program pengakses database seperti Oracle, Progress SQL, SQL Server, dan lain-lain (Nugroho, 2004:29).

Basis data *server* atau MySQL adalah program yang dapat menyimpan sejumlah besar informasi dalam format terorganisir yang mudah diakses melalui bahasa pemrograman seperti PHP (Butler & Yank, 2017).

5. Pendistribusian (*deployment*)

Tahapan *deployment* merupakan tahapan implementasi perangkat lunak ke pengguna, pemeliharaan perangkat lunak secara berkala, perbaikan perangkat lunak, evaluasi perangkat lunak, dan pengembangan perangkat lunak berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya (Pressman, 2012:17).

3. Analisis Kualitas Perangkat Lunak

Perangkat lunak yang dikembangkan membutuhkan pengujian karena kualitas perangkat lunak perlu disesuaikan dengan kebutuhan dan kepuasan pengguna. Meminimalisir kesalahan secara teknik serta kesalahan secara non teknis merupakan bagian tak terpisahkan dari pengujian kualitas dari perangkat lunak (Rosa A.S & Shalahuddin, 2011:210). Pengujian perangkat lunak adalah sebuah elemen suatu topik yang mempunyai lingkupan luas dan kerap kali dihubungkan dengan verifikasi dan validasi (Rosa A.S & Shalahuddin, 2011:211).

Menurut Pressman (2012:485), kualitas perangkat lunak dapat didefinisikan sebagai suatu proses perangkat lunak yang efektif diterapkan dalam arti kata proses perangkat lunak yang menyediakan nilai yang dapat diukur untuk mereka yang menghasilkan.

Pengujian untuk mengetahui kualitas perangkat lunak dapat dilakukan dengan berbagai model, diantaranya ISO 25010, ISO 25010, McCall, Boehm, Dromey, dan FURPS (Al-Qutaish, 2010). Dalam proses pengujian kualitas perangkat lunak diperlukan sebuah parameter yang nantinya digunakan untuk melakukan uji kelayakan perangkat lunak yang dikembangkan.

Dari berbagai macam standar pengujian tersebut, ISO 9126 dan ISO 25010 merupakan standar internasional dalam pengujian perangkat lunak. ISO 25010 adalah standar kualitas perangkat lunak mutakhir, menjadi pembaharu dari standar kualitas perangkat lunak sebelumnya yaitu ISO 9126. Standar kualitas perangkat lunak ini dirancang oleh *International Organization for Standardization (ISO) and International Electrotechnical Commission* (Wagner, 2013). Maka karena itu,

dalam penelitian ini menggunakan standar ISO 25010 sebagai standar dalam pengujian perangkat lunak. ISO 25010 mempunyai 8 karakteristik antara lain *functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability, dan portability*.

Standar kualitas aplikasi web dinilai dari lima aspek diantaranya yaitu fungsionalitas, kemudahan penggunaan, keandalan, efisiensi dan kemudahan pemeliharaan (Olsina, Godoy, Lafuente, & Rosi, 1999). Rincian uraian dari lima aspek tersebut dapat dilihat pada Tabel 1.

Tabel 1. Standar kualitas aplikasi *web* menurut Olsina

No	Faktor	Sub Faktor
1	Fungsionalitas	Kemampuan pencarian dan penerimaan
		Fitur-fitur navigasi dan perambahan
		Fitur-fitur aplikasi yang berhubungan dengan ranah
2	Kemudahan penggunaan	Kemudahan pemahaman situs global
		Umpang balik dari pengguna dan fitur-fitur bantuan
		Antarmuka pengguna dan fitur-fitur estetika
3	Keandalan	Fitur-fitur khusus
		Pembetulan pemrosesan tautan
		Pemulihan dari kesalahan
4	Effisiensi	Validasi dan pemulihan asupan pengguna
		Kinerja waktu tanggap aplikasi web
		Kecepatan pembentukan halaman-halaman
		Kecepatan penggambaran grafik-grafik

5	Kemudahan pemeliharaan	Kemudahan untuk melakukan koreksi
		Kemampuan aplikasi <i>web</i> untuk beradaptasi
		Kemampuan aplikasi <i>web</i> untuk dikembangkan

Berdasarkan standar kualitas web menurut Olsina dkk dalam Pressman (2012:456) maka peneliti hanya akan mengambil 5 karakteristik ISO 25010 yaitu *functional suitability*, *usability*, *reliability*, *performance efficiency*, dan *maintainability*.

Tabel 2. Perbandingan ISO 25010 dengan standar kualitas Olsina (2012)

Olsina	ISO 25010
Fungsionalitas	<i>Functional suitability</i>
Kemudahan penggunaan	<i>Usability</i>
Keandalan	<i>Reliability</i>
Effisiensi	<i>Performance Efficiency</i>
Kemudahan pemeliharaan	<i>Maintainability</i>

Berikut merupakan penjelasan terkait 5 karakteristik yang akan digunakan untuk menguji kualitas sistem informasi akademik SD Negeri Kemiren Srumbung Magelang:

1) *Functional Suitability*

Karakteristik yang menguji sejauh mana produk atau aplikasi dapat menyediakan fungsi untuk memenuhi kebutuhan yang dapat digunakan dalam kondisi tertentu. *Functional suitability* terbagi menjadi 3 sub karakteristik:

- a. *Functional completeness*, mengukur sejauh mana fungsi mencakup semua tugas dan tujuan pengguna.

- b. *Functional correctness*, mengukur sejauh mana produk atau aplikasi memberikan hasil yang tepat sesuai kebutuhan.
- c. *Functional appropriateness*, mengukur sejauh mana fungsi dapat memfasilitasi untuk menyelesaikan tugas dan tujuan tertentu.

Pengujian *functional suitability* dengan cara menggunakan metode *black-box testing*. *Black-box testing* atau biasa disebut dengan pengujian perilaku, persyaratan fungsionalitas perangkat lunak merupakan hal yang dikaji dalam *black-box testing* ini (Pressman, 2012). *Black-box testing* berguna sebab dapat menjumpai kesalahan pada berikut: (1) beberapa fungsi cacat atau hilang, (2) kesalahan antarmuka, (3) kesalahan pada struktur data, (4) kesalahan perilaku, dan (5) kesalahan instalasi.

2) *Usability*

Sejauh mana produk atau aplikasi dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektif, efisien dan kepuasan dalam konteks penggunaan tertentu. Karakteristik ini terbagi menjadi 6 sub karakteristik:

- a. *Appropriateness recognizability*, sejauh mana pengguna dapat mengenali apakah suatu produk atau aplikasi yang sesuai dengan kebutuhan mereka.
- b. *Learnability*, sejauh mana produk atau aplikasi dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dari pembelajaran untuk menggunakan produk atau aplikasi dengan efektif, efisien, bebas dari risiko dan kepuasan dalam konteks tertentu dalam penggunaan.
- c. *Operability*, sejauh mana produk atau aplikasi memiliki atribut yang membuatnya mudah dioperasikan dan dikontrol.

- d. *User error protection*, sejauh mana aplikasi melindungi dari pengguna jika membuat kesalahan.
- e. *User interface aesthetics*, sejauh mana *user interface* memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
- f. *Accessibility*, sejauh mana produk atau aplikasi dapat digunakan oleh orang dalam jangkauan luas dari karakteristik dan kemampuan untuk mencapai suatu tujuan tertentu dalam konteks tertentu dalam penggunaan.

Usability dapat didefinisikan sebagai kemudahan dalam penggunaan. *Usability* merupakan usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan *input*, dan menginterpretasikan *output* suatu program. Tingkat usabilitas yang semakin baik akan semakin memudahkan dan kenyamanan pada saat program tersebut digunakan.

Pada penelitian ini pengujian *usability* menggunakan USE Questionnaire yang dikembangkan oleh Arnold M. Lund. USE merupakan kuesioner yang terdiri dari tiga dimensi yaitu *usefulness*, *satisfaction* dan *ease of use* yang terdiri dari 30 pertanyaan (Lund, 2001). Skala yang akan digunakan yaitu skala likert. Rentang skala likert dimulai dari sangat tidak setuju sampai sangat setuju. Menurut Sudaryono (2015:62), skala likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang kejadian atau gejala sosial yang telah ditetapkan secara spesifik oleh peneliti.

3) *Performance Efficiency*

Menurut ISO 25010 (2011) mendefinisikan tingkat kinerja relatif terhadap sumber daya yang digunakan dalam kondisi yang ditetapkan. Sedangkan menurut

Pressman (2012:673) pengujian kinerja dilakukan untuk memastikan sistem dapat menangani beban ekstrem tanpa penurunan kemampuan operasi yang tidak dapat diterima. Pengujian kinerja digunakan untuk mengungkap masalah-masalah kinerja yang muncul akibat kurangnya sumber daya pada *server*, *bandwidth* jaringan yang kurang, kemampuan basis data yang tidak memadai, kemampuan sistem operasi yang rusak atau lemah, fungsionalitas aplikasi yang dirancang buruk, dan masalah perangkat keras lainnya atau perangkat lunak yang mengakibatkan rusaknya kinerja *client-server*.

Karakteristik ini terbagi menjadi beberapa sub-karakteristik yaitu :

- a. *Time behaviour*, sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- b. *Resource utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- c. *Capacity*, sejauh mana batas maksimum parameter produk atau sistem dapat memenuhi persyaratan.

Pengujian *performance efficiency* dapat dilakukan dengan *load testing* (Dhiman & Sharma, 2015). Tujuan dari load testing adalah untuk menentukan bagaimana aplikasi web dan lingkungan sisi server akan menanggapi berbagai kondisi pemuatan (Pressman, 2012:657).

Salah satu *software* untuk melakukan *load testing* yaitu GTMetrix (Sukhpuneet, Kulwant, & Parminder, 2016). GTMetrix merupakan perangkat lunak yang dapat digunakan secara bebas untuk menganalisis *speed performance* sebuah halaman

website. Hasilnya berupa waktu memuat halaman, ukuran halaman serta jumlah *request* dan kecepatan halaman serta skor Yslow. YSlow adalah alat yang dikembangkan oleh Yahoo! yang memeriksa semua komponen kinerja halaman dan memberikan saran untuk perbaikan (Priyadarsini & Mamatha, 2013:319).

4) *Reliability*

Reliability merupakan karakter sebuah sistem yang menunjukkan fungsi tertentu sesuai kondisi pada periode waktu tertentu. Karakteristik ini terbagi menjadi beberapa sub-karakteristik, yaitu:

- a. *Maturity*, sejauh mana produk atau sistem mampu memenuhi kebutuhan secara handal di bawah keadaan normal.
- b. *Availability*, sejauh mana produk atau sistem siap beroperasi dan dapat diakses saat perlu digunakan.
- c. *Fault tolerance*, sejauh mana produk atau sistem tetap berjalan sebagaimana yang dimaksud meskipun terjadi kesalahan pada perangkat keras atau perangkat lunak.
- d. *Recoverability*, sejauh mana produk atau sistem mampu dapat memulihkan data yang terkena dampak secara langsung dan menata ulang kondisi sistem seperti yang dinginkan ketika terjadi gangguan.

Dari beberapa sub-karakteristik *reliability* tersebut, sub-karakteristik *maturity* yang paling berpengaruh terhadap *software* berbasis *web* (Olsina, Covella, & Rossi, 2006). Aspek *maturity* merupakan tingkat kematangan dari perangkat lunak. Maksud dari kematangan adalah tingkat frekuensi kegagalan perangkat lunak jika digunakan dalam periode waktu tertentu.

Menurut (ISO/IEC, 2002), dalam pengujian sub-karakteristik maturity disarankan menggunakan *stress testing*. *Stress testing* bekerja dengan menghitung *test case* yang berhasil dilakukan oleh *software* dan membandingkannya dengan total *test case* yang dilakukan. Setelah perhitungan dilakukan kemudian dilakukan interpretasi berdasarkan standar telecordia *reliabilitas* perangkat lunak. Menurut GR-282 “*Software Reliability and Quality Acceptance Criteria*” yang dikutip oleh (Asthana & Olivieri, 2009), 100% dari perencanaan pengujian *fungsionalitas* sistem harus dilakukan dan keberhasilan reliabilitas perangkat lunak harus lebih dari 95% atau 0,95 dalam bilangan desimal.

Menurut Pradhan (2013) untuk melakukan *stress testing* dapat menggunakan *software Web Application Load, Stress, and Performance Testing* (WAPT). Menurut website resminya, (*Official Website WAPT*, 2016), WAPT adalah alat pengujian *reliability* dengan metode *stress testing* yang sudah digunakan dan hemat biaya untuk menguji situs web, termasuk aplikasi bisnis, situs *mobile*, *web portal*, dll. Menurut (Kundu, 2012), WAPT bekerja dengan memberikan simulasi pengunjung web yang akan merekam dan memilih alamat URL dan mengijinkan pengguna untuk menentukan banyaknya pengunjung dalam waktu tertentu. Dengan melakukan metode tersebut maka sangat berguna untuk mengecek performa dari website yang dikembangkan.

5) *Maintainability*

Pengujian aspek *maintainability* yaitu tingkat efektivitas dan efisiensi pada suatu produk atau sistem untuk dapat dimodifikasi oleh pengembang (ISO 25010, 2011). Modifikasi mencakup koreksi, perbaikan atau adaptasi terhadap perubahan

lingkungan, peryaratan dan spesifikasi (ISO 25010, 2011). Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu:

- a. *Modularity*, tingkat dimana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain.
- b. *Reusability*, tingkat dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak pada pembangunan asset lainnya.
- c. *Analyzability*, merupakan kemampuan dimana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi.
- d. *Changeability*, merupakan tingkat dimana perangkat lunak memungkinkan sebuah modifikasi yang spesifik untuk dapat diimplementasikan. Hal ini bertujuan untuk memudahkan dalam modifikasi.
- e. *Testability*, tingkat dimana modifikasi perangkat lunak untuk dilakukan validasi.
- f. *Compliance*, merupakan tingkatan dimana sebuah perangkat lunak dapat memenuhi standarisasi yang berhubungan dengan maintainability pada perangkat lunak lain.

Dalam penelitian ini yang dimaksud pengujian aspek *maintainability* merupakan pengujian secara operasional untuk mengetahui apabila terjadi kegagalan dalam input data, tampilan sistem, dan apakah sistem mudah untuk dilakukan modifikasi. Pengujian ini dilakukan secara operasional dengan

menggunakan instrumen *maintainability* (Land, 2002). Aspek yang terdapat dalam instrumen tersebut adalah *instrumentation*, *consistency*, dan *simplicity*.

Menurut Najm (2014), untuk mengukur tingkat maintainability suatu sistem menggunakan *Maintainability Index* (MI). MI merupakan kombinasi software metrics yang bernama *McCabe's Cyclomatic Complexity* (CC), *Halstead's Volume* (V), dan *Lines of Code* (LOC) yang berpengaruh terhadap *source code* suatu software mudah untuk dimodifikasi atau dilakukan perbaikan (Ganpati, Kalia, & Singh, 2012).

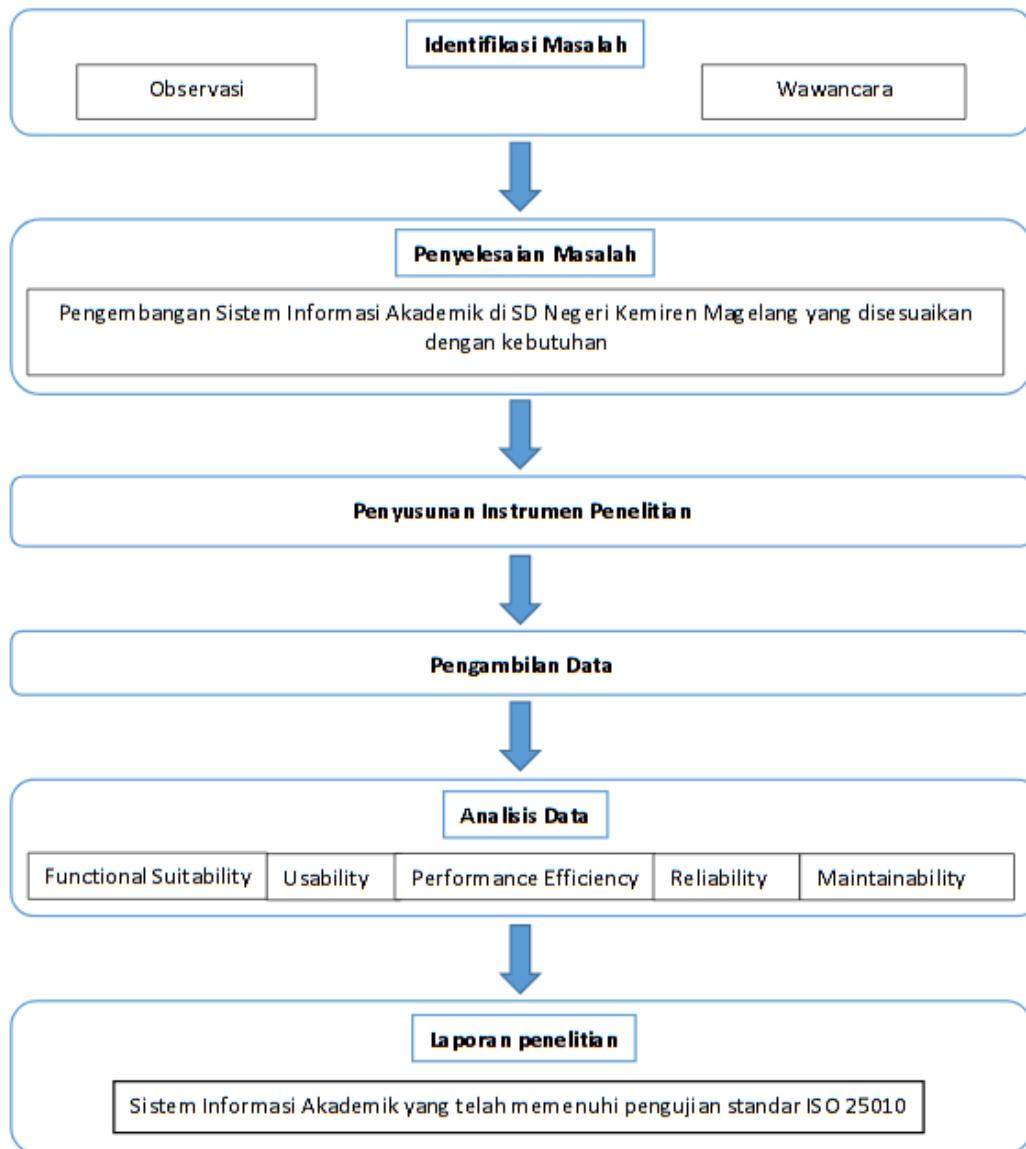
Menurut Lepine (2015), untuk menghitung *Maintainability Index* dapat dibantu menggunakan *software* PHPmetrics. PHPmetrics merupakan perangkat lunak yang digunakan untuk menganalisis perangkat lunak pada karakteristik *maintainability* yang menggunakan bahasa PHP. PHPmetrics akan melakukan perhitungan sesuai dengan rumus pada karakteristik *maintainability* dan akan menampilkan hasil akhir berupa MI.

B. Penelitian yang Relevan

1. Sistem Informasi Akademik berbasis Website di MTs Negeri Model Parakan oleh Pradana Luthfi Suryawan dalam penelitiannya yang dilakukan pada tahun 2011. Hasil dari penelitian ini adalah menghasilkan produk sistem informasi akademik berbasis website sebagai pendukung berlangsungnya proses kegiatan belajar mengajar yang dibangun menggunakan bahasa pemrograman php. Pengujian kelayakan sistem informasi dari penelitian ini adalah dari sisi *Design, Usability, Content* dan *Accessibility*.
2. Analisis dan Pengembangan Sistem Informasi Akademik Siswa berbasis web menggunakan PHP Dan MySQL Di SMA N 1 Tayu yang disusun oleh Lupiyo Hartadi pada tahun 2012. Tujuan dari penelitian ini adalah menghasilkan perangkat lunak Sistem Informasi Akademik Siswa yang terdiri dari pengolahan data akademik secara umum yaitu data administrator, tata usaha, kepala sekolah, wali kelas, guru, dan siswa.
3. Pengembangan Sistem Informasi Akademik di SMK 1 Pundong untuk Meningkatkan Mutu Layanan Sekolah yang diteliti oleh Juri Benedi pada tahun 2014. Tujuan dari penelitian ini ialah mengetahui (1) kebutuhan sistem informasi (2) pengembangan sistem informasi akademik (3) unjuk kerja Sistem Informasi Akademik dan (4) penerapan Sistem Informasi Akademik di SMK 1 Pundong dalam membantu kinerja guru dan karyawan terutama untuk peningkatan mutu layanan sekolah.
4. Pengembangan dan Analisis Kualitas Sistem Informasi Laporan Hasil Belajar Siswa berbasis web di SMK Negeri 1 Banyumas yang disusun oleh Avidah

Amalia Zahro pada tahun 2016. Penelitian ini bertujuan untuk mengembangkan sistem informasi laporan hasil belajar siswa di SMK Negeri 1 Banyumas, mengetahui tingkat kualitas sistem informasi laporan hasil belajar siswa berdasarkan pengujian kualitas sesuai dengan standar ISO/IEC 25010. Kesimpulan dari penelitian ini yaitu sistem mampu untuk menyajikan hasil evaluasi kelas yang diimplementasikan dengan web dengan uji standar internasional ISO 25010.

C. Kerangka Pikir



D. Pertanyaan Penelitian

Pertanyaan penelitian dari pengembangan Sistem Informasi Akademik SD Negeri Kemiren Magelang sebagai berikut:

1. Bagaimana mengembangkan piranti lunak sistem informasi akademik siswa berbasis *web* sesuai dengan keperluan di SD Negeri Kemiren Srumbung Magelang sebagai sumber informasi sekolah ?
2. Bagaimana menguji standar kualitas sebuah perangkat lunak agar dengan kaintannya kesalahan fitur serta fungsionalitas bisa diminilasir menggunakan standar ISO 25010 ?