

## **BAB II**

### **KAJIAN PUSTAKA**

#### **A. Kajian Teori**

##### **1. Rekayasa Perangkat Lunak**

Menurut Janner Simarmata (2010: 10) rekayasa atau teknik adalah penerapan ilmu dan teknologi untuk menyelesaikan permasalahan manusia. Hal ini diselsaikan lewat pengetahuan, matematika, dan pengalaman praktis yang diterapkan untuk mendesain objek atau proses yang berguna. Sedangkan menurut *Institute of Electrical and Electronics Engineers* (IEEE) mendefinisikan rekayasa perangkat lunak sebagai penerapan sebuah pendekatan yang sistematik, tertib, dan terukur terhadap pengembangan, pengoperasian, dan perawatan perangkat lunak.

##### **2. Aplikasi Web**

Menurut Shelly (2009: 14), “Sebuah aplikasi *web* adalah situs *web* yang memungkinkan pengguna untuk mengakses dan berinteraksi dengan perangkat lunak dari komputer atau perangkat yang terhubung ke Internet.” *User* sering berinteraksi dengan aplikasi *web* langsung melalui situs *web* (host) melalui *web browser*. Mercer (2004: 392) menambahkan aplikasi *web* atau e-application dapat diakses oleh pemakai dengan menggunakan *browser* sebagai *client*. *Browser* berperan sebagai medium antara sistem yang dibangun dengan fungsi yang akan dijalankan oleh pengguna aplikasi tersebut.

### **3. Metodologi Pengembangan**

Menurut Pressman (2010) pengembangan perangkat lunak meliputi suatu proses, kumpulan metode dan sederetan perkakas yang memungkinkan para profesional mengembangkan *software* komputer berkualitas tinggi. Lebih lanjut, menurut Kittlaus dan Fricker (2016: 223), untuk sebuah proyek metodologi pengembangan yang sesuai harus dipilih dan diikuti. Pemilihan metodologi berpengaruh terhadap menejemen proyek dan pengembangan produk. Metodologi yang dipilih dalam pengembangan produk berdampak pada bagaimana kebutuhan produk ditangani pada tahap implementasi dan bagaimana memanajemen penerimaan pada produk yang dikembangkan. Ada empat tahap fundamental dalam kebanyakan metodologi rekayasa perangkat lunak. Tahapan tersebut adalah analisis, desain, implementasi, dan pengujian. Tahapan-tahapan tersebut menunjuk pada ‘produk apa yang akan dibuat’, ‘bagaimana membuat produk tersebut’, ‘membuat produk tersebut’, dan ‘membuatnya memiliki kualitas yang tinggi’ (Burbank, 1998:3).

#### a. Metode ADDIE

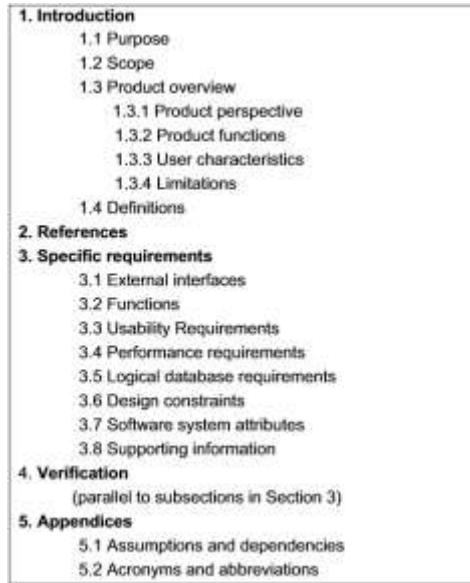
ADDIE merupakan singkatan dari *Analysis, Design, Development, Implementation, dan Evaluation*. Model ADDIE dikembangkan oleh Dick dan Carry pada tahun 1996 untuk merancang sistem pembelajaran (Mulyatiningsih, 2012:184). Semula methodologi ini dikembangkan untuk U.S Army oleh *Center for Educational Technology* di *Florida State University*. Berdasarkan struktur hirarki langkah-

langkahnya, setiap proses harus diselsaikan terlebih dahulu untuk dapat memulai proses selanjutnya (Kurt, 2017: 1).

### 1) *Analysis*

Fase *analyis* dapat dinyatakan sebagai tahap penentuan tujuan. Pada tahap ini kegiatan utamanya adalah menganalisa kebutuhan produk yang dikembangkan. Pada tahapan analysis didefinisikan *requirement* dari produk yang akan dikembangkan. Tahapan ini mendefinisikan permasalahan yang akan diselsaikan (Burback, 1998:6). Sedangkan menurut Simarmata, *requirement* (analisis kebutuhan) adalah sebuah proses untuk mendapatkan informasi, model, spesifikasi tentang perangkat lunak yang diinginkan klien/pengguna. Kedua belah pihak yaitu klien dan pembuat perangkat lunak terlibat aktif dalam tahap ini. Informasi yang diperoleh dari klien/pengguna inilah yang akan menjadi acuan untuk melakukan desain perangkat lunak.

Standar pendokumentasian *software requirements* yang sering digunakan adalah ISO 29148:2011 *System and software engineering—Life cycle processes—Requirement engineering*. Standar tersebut mendefinisikan *software requirement specification* (SRS) sebagai spesifikasi untuk produk *software*, program, atau set program tertentu yang mengerjakan fungsi tertentu pada lingkungan yang spesifik. Garis besar SRS dapat dilihat pada Gambar 1.



Gambar 1. Garis Besar *Software Requirement Spesification*

a) *Purpose*

Berisi definisi spesifik mengenai tujuan dikembangkannya *software*.

b) *Scope*

Berisi deskripsi cakupan *software* yang dikembangkan.

c) *Product perspective*

Berisi definisi relasi sistem yang dikembangkan dengan produk serupa.

d) *Product functions*

Berisi ringkasan mengenai fungsi utama dari kerja *software*.

e) *User characteristics*

Berisi deskripsi mengenai karakteristik umum dari kelompok pengguna yang akan menggunakan *software*.

f) *Limitation*

Berisi diskripsi batasan-batasan penggunaan *software*.

g) *Definition*

Berisi definisi dari istilah-istilah asing yang berhubungan dengan proses pengembangan *software*.

h) *Specific requirement*

Berisi spesifikasi dari setiap kebutuhan sofware yang cukup detil sehingga dapat membuat desainer dapat merancang *software* untuk dapat memenuhi *requirement* tersebut.

i) *External interfaces*

Mendefinisikan semua input maupun output atas sistem *software*.

j) *Functions*

Berisi definisi aksi dasar dalam *software* dalam menerima dan memproses input serta dalam memporses dan menghasilkan output.

k) *Usability requirements*

Berisi definisi *usability (quality in use) requirements*. Termasuk didalamnya kriteria yang terukur atas efektivitas, efisiensi, kepuasan dalam konteks penggunaan *software*.

l) *Performance requirements*

Berisi spesifikasi *static* maupun *dynamic numerical requirement* dalam *software* atau dalam interaksi manusia dengan sofware sebagai satu kesatuan.

m) *Logical database requirements*

Berisi spesifikasi *logical requirements* dari setiap informasi yang ada di dalam *database*.

n) *Design constrain*

Berisi spesifikasi keharusan-keharusan atas desain sistem atas paksaan standar eksternal, peraturan-peraturan atau batasan proyek.

o) *Software system attribute*

Berisi diefinisi attribut yang dibutuhkan dari produk *software*.

p) *Supporting information*

Berisi informasi yang mendukung yang dapat membantu pembaca memahami SRS.

q) *Verification*

Berisi pendekatan verifikasi dan perencanaan metode dalam memenuhi kebutuhan *software*.

r) *Assumption and dependencies*

Berisi daftar asumsi dan ketergantungan yang berlaku pada sistem *requirement* yang harus dijelaskan dalam turunanya sebagai *lower-level system requirement s*.

s) *Acronym and abbreviations*

Berisi penjelasan dari akronim dan singkatan yang digunakan dalam dokumen.

**2) *Design***

Pada tahapan *design* ditetapkan arsitektur dari sebuah *software*. Arsitektur *software* mendefinisikan komponen tampilan dan perilaku (Burbank, 1998:13).

a) Antarmuka

Secara sempit, antarmuka dapat didefinisikan sebagai input dan output dan perangkat lunak yang memberikan pelayanan terhadap sebuah fungsi, sedangkan secara luas antar muka mencakup segala bentuk yang membentuk pengalaman pengguna dalam menggunakan komputer, termasuk didalamnya dokumentasi, pelatihan, dan dukungan manusia (Beacker, 1995). Antrmuka komputer termasuk didalamnya *hardware* dimana sebuah sistem dijalankan seperti *keyboard*, *mouse*, *joystick*, *trackball*, dan layar display itu sendiri. Sedangkan komponen antarmuka *software* adalah semua yang user lihat, dengar, tujuk, atau sentuh di dalam layar untuk berinteraksi dengan komputer itu sendiri, serta informasi pengguna itu sendiri (Mandel, 1997).

b) *Flow chart*

Menurut Suprapto (2008: 56), *Flow chart* (diagram alir) adalah penggambaran secara grafik dari langkah-langkah pemecahan masalah uang harus diikuti oleh pemroses. Sedangkan menurut Barakbah (2013: 19), *Flow chart* adalah cara penulisan algoritma dengan menggunakan notasi grafis. *Flow chart* merupakan gambaran atau bagan yang memperlihatkan urutan atau langkah dari suatu program dan hubungan antar proses beserta pernyataanya. *Flow chart* terdiri atas sekumpulan simbol dimana masing-masing simbol menggambarkan suatu kegiatan tertentu. Simbol-simbol *Flow chart* yang biasanya dipakai adalah simbol-simbol *Flow chart* standar yang

dikeluarkan oleh ANSI dan ISO. Tabel 1 merupakan beberapa simbol *flow chart* yang digunakan dalam suatu *flow chart*.

Tabel 1. Simbol-Simbol *Flow Chart*

Simbol	Nama	Fungsi
	Terminator	Simbol awal (start)/ simbol akhir (end)
	Flow line	Simbol aliran / penghubung
	Proses	Simbol proses / perhitungan / pengolahan
	Input / output data	Merepresentasikan pemabcaan / penulisan data
	Decision	Simbol pernyataan pilihan, berisi suatu kondisi yang selalu menhasilkan 2 nilai kebenaran yaitu benar atau salah
	Predifined process (subprogram)	Proses menjalankan fungsi/ prosedur
	On page connector	Penghubung flowchart pada satu halaman

### 3) *Development*

Kegiatan development dalam model ADDIE berisi kegiatan realisasi rangangan produk (Mulyatiningsih, 2012: 184). Setelah desain terbaik atau paling tepat dipilih,

implementasi akan segera dimulai. Pemrogram harus mengembangkan perangkat lunak sesuai dengan *requirement* dan pada saat yang sama juga mengikuti standar pengkodean yang ditentukan (Tiky, 2016: 8). Penting dalam konteks ini untuk memahami bahwa *requirement* dan proses *development* saling berhubungan karena metode *development* yang berbeda akan menghasilkan pemenuhan *requirement* yang berbeda pula (Kittlaus 2017: 135).

#### **4) *Implementation***

Selama proses implementasi, rancangan model/metod yang telah dikembangkan diterapkan pada kondisi yang sebenarnya (Mulyatiningsih, 2012: 185). Pada tahap ini produk yang dikembangkan diserahkan kepada pengguna dengan mengunggahnya pada layanan *hosting* situs web.

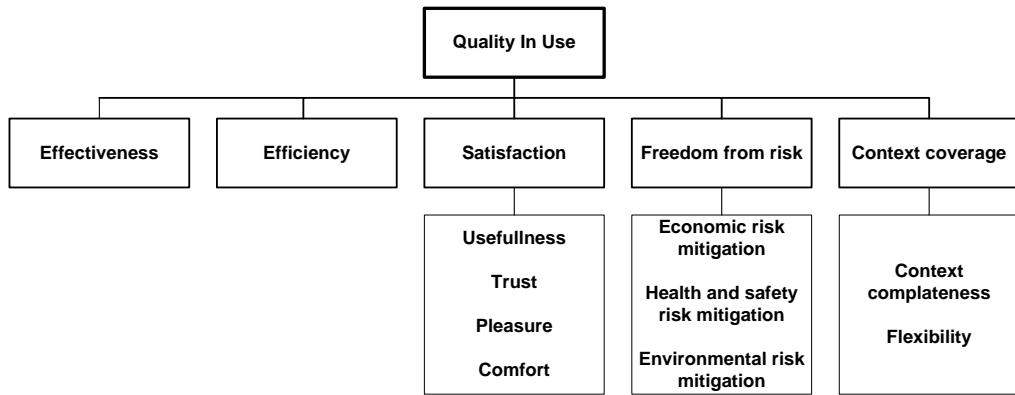
#### **5) *Evaluation***

Pengujian perangkat lunak harus dilakukan pada semua tahap. Meskipun demikian dua yang utama yang harus dilakukan adalah pengujian oleh programer, pengguna dan ahli untuk menjamin kualitas (Tiky, 2016: 8). Menurut Rosa, A. S dan Shalahuddin (2013) “Sebuah perangkat lunak atau *software* perlu dijaga kualitasnya. Kualitas ini mempengaruhi kepuasan penggunanya. Kualitas *software* agar bisa bertahan hidup perlu dijaga, dan juga bersaing dengan *software* yang lain, efektif dalam biaya pengembangan, dapat bersaing dalam hal pemasaran, serta mendapatkan pelanggan yang sama”.

*Quality assurance* terdiri atas audit dan pelaporan atas fungsi produk perangkat lunak. Tujuan adanya *Quality assurance* adalah untuk menyediakan data mengenai kualitas produk sehingga dapat meyakinkan bahwa kualitas produk dapat memenuhi tujuannya. Jika data yang dihasilkan menunjukkan adanya masalah, maka menjadi tanggung jawab pengembang untuk mengatasi masalah tersebut dan menerapkan sumber daya yang dibutuhkan untuk mengatasi masalah kualitas produk tersebut (Pressman 2011).

Pada penelitian ini, *Quality assurance* mengacu pada ISO 25010 “*Software product Quality Requirment and Evaluation* (SQuaRE)”. ISO 25010 adalah standar internasional yang digunakan sebagai pengganti ISO/IEC 9126. Ada suatu buah model yang didefinisikan dalam standar ini, yaitu:

1. Model kualitas penggunaan (*quality in use*) yang didalamnya terkandung lima karakteristik yang berhubungan dengan hasil interaksi pengguna dengan sistem. Karakteristik tersebut meliputi: efektivitas, efisiensi, kepuasan, kebebasan dari resiko, dan kecakupan konteks. Karakteristik-karakteristik tersebut nantinya akan terbagi dalam beberapa sub karakteristik yang dapat diukur/diuji untuk menentukan kualitas penggunaan sistem ketika digunakan.



Gambar 2. Karakteristik ISO 25010

- a) Efektivitas (*effectiveness*). Keakuratan serta kelengkapan yang disediakan agar pengguna dapat mencapai tujuan yang spesifik..
- b) Efisiensi (*efficiency*). Tingkat penggunaan sumber daya yang diperlukan agar pengguna dapat mencapai tujuan secara akurat dan lengkap. Sumber daya yang dimaksud di sini dapat berupa biaya keuangan, bahan, maupun waktu.
- c) Kepuasan (*satisfaction*). Tingkat di mana pengguna merasa puas ketika sistem digunakan dalam konteks yang spesifik.
  - 1) Daya guna (*Usefulness*).  
Tingkat di mana pengguna secara pragmatis merasa puas dengan sistem yang digunakan, termasuk terhadap konsekuensi dan hasil dari penggunaan sistem tersebut.
  - 2) Kepercayaan (*Trust*)  
Tingkat dimana pengguna maupun pemangku kepentingan merasa yakin bahwa sistem atau produk yang digunakan akan memiliki unjuk kerja sesuai dengan yang dimaksudkan.

3) Kesenangan/kenikmatan (*Pleasure*)

Tingkat di mana pengguna merasa senang atau dapat menikmati penggunaan sistem atau produk untuk memenuhi kebutuhan mereka masing-masing..

4) Kenyamanan (*Comfort*)

Tingkat di mana pengguna puas terhadap kenyamanan secara fisik yang diberikan oleh sistem atau produk.

d) Kebebasan dari resiko (*Freedom from risk*). tingkat dimana sistem atau produk dapat mencegah potensi kerugian yang dialami oleh pengguna secara kesehatan, ekonomi, maupun lingkungan.

1) Pencegahan resiko ekonomi (*Economic risk mitigation*)

Tingkat di mana sistem atau produk dapat mencegah potensi kerugian pada status finansial, efisiensi pengoperasian, aset yang memiliki nilai komersil, reputasi, atau sumberdaya lainnya dalam konteks penggunaan sistem atau produk tersebut.

2) Pencegahan resiko kesehatan dan keamanan (*Health and safety risk mitigation*)

Tingkat di mana sistem atau produk dapat mencegah potensi resiko kesehatan dan keamanan dalam konteks penggunaan sistem atau produk itu sendiri.

3) Pengurangan resiko lingkungan (*Environmental risk mitigation*)

Tingkat di mana sistem atau produk dapat mencegah resiko kerugian terhadap properti maupun lingkungan dalam konteks penggunaan sistem atau aplikasi tersebut.

e) Kecakupan konteks (*Context coverage*). Tingkat di mana sistem atau produk memiliki daya guna yang efektif, efisien, bebas dari resiko, serta memuaskan dalam

konteks penggunaan yang spesifik maupun dalam konteks di luar penggunaan yang sebelumnya sudah diidentifikasi secara eksplisit.

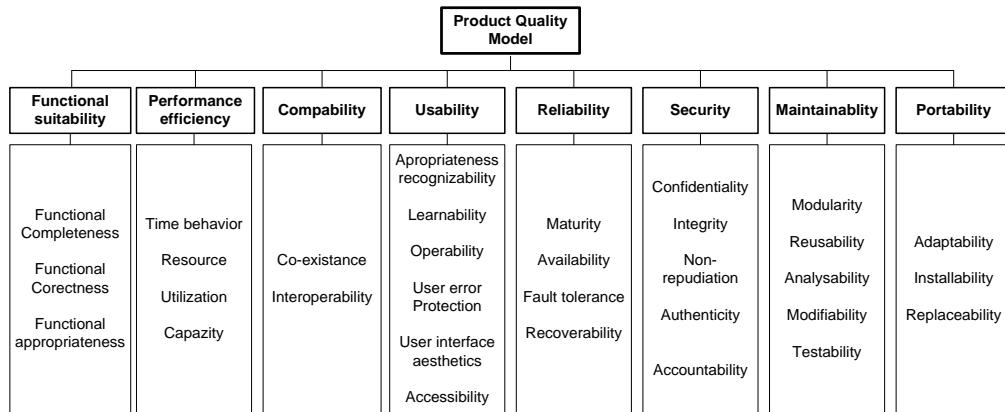
1) Kelengkapan konteks (*Context Completeness*)

Kelengkapan konteks dapat ditentukan atau diukur sebagai tingkat dimana produk dapat digunakan oleh pengguna yang spesifik untuk mencapai tujuan yang spesifik pula secara efektif, efisien, bebas dari resiko, dalam konteks penggunaan sistem atau produk yang dimaksudkan, atau dengan adanya fitur produk yang mendukung dapat tercapainya hal tersebut.

2) Fleksibilitas (*Flexibility*)

Tingkat di mana sistem atau produk dapat digunakan secara efektif, efisien, bebas dari resiko, dan memuaskan dalam konteks yang melebihi spesifikasi yang sebelumnya sudah ditentukan. Untuk mencapai fleksibilitas dapat dilakukan dengan mengadaptasikan produk untuk kategori kelompok pengguna, budaya, maupun tugas tertentu.

f) Model kualitas produk (*product quality model*) mengategorikan produk perangkat lunak dalam 8 karakteristik meliputi: keseuaian fungsional, efisiensi kerja, kompatibilitas, kemampuan penggunaan, keandalan, keamanan, kemampuan pemeliharaan, dan kemampuan pemindahan. Model kualitas produk berfokus pada target sistem computer yang ada dalam target produk perangkat lunak.



*Gambar 3. Model Kualitas Produk*

1) Kesesuaian fungsional (*Functional suitability*)

Tingkat di mana produk atau sistem menyediakan fungsi yang memenuhi kebutuhan ketika digunakan dalam kondisi yang spesifik. Kesesuaian fungsional hanya mencakup pada apakah fungsi dapat mencapai kebutuhan yang dinyatakan maupun yang tersirat, bukan pada spesifikasi fungsional.

(a) Kekomplitan fungsional (*Functional completeness*). Tingkat di mana fungsi dapat mencakup semua tugas tertentu dan tujuan pengguna.

(b) Ketepatan fungsional (*Functional correctness*). Tingkat di mana produk atau sistem menyediakan hasil yang tepat dengan tingkat ketepatan kebutuhan.

(c) Kepatuhan fungsional (*Functional appropriateness*). Tingkat di mana fungsi-fungsi memfasilitasi pemenuhan tugas yang telah ditentukan.

2) Efisiensi kinerja (*Performance efficiency*)

Kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang ditentukan.

- (a) Perilaku terhadap waktu (*Time behavior*). Tingkat di mana respon, waktu proses, dan tingkat keseluruhan produk atau sistem memenuhi persyaratan ketika melakukan fungsinya.
- (b) Pemanfaatan sumber daya (*Resource utilization*). Tingkat di mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem memenuhi persyaratan ketika melakukan fungsinya.
- (c) Kapasitas (*Capacity*). Tingkat di mana batas maksimum parameter produk atau sistem memenuhi persyaratan.

### 3) Kompabilitas (*Compatibility*)

Tingkat di mana produk, sistem, atau komponen dapat bertukar informasi dengan produk yang lain, sistem atau komponen, dan/atau menjalankan fungsi-fungsi yang diperlukan sementara berbagi lingkukan perangkat keras atau perangkat lunak yang sama.

- (a) Koeksistensi (*Co-existence*). Tingkat di mana suatu produk dapat melakukan fungsi yang diperlukan secara efisien sementara berbagi lingkungan dan sumber daya umum dengan produk lainnya, tanpa adanya dampak merugikan pada produk lain.
- (b) Kemampuan saling beroprasи (*Interoperability*). Tingkat di mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan yang telah ditentukan.

### 4) Kemampuan penggunaan (*Usability*)

Tingkat di mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan tertentu. Kemampuan penggunaan dapat ditentukan atau diukur sebagai karakteristik kualitas produk dalam hal subkarakteristik, atau ditentukan atau diukur langsung oleh ukuran yang merupakan bagian dari kualitas penggunaan.

- (a) Kemampuan mengenal (*Appropriateness recognizability*). Tingkat di mana pengguna dapat mengenali apakah suatu produk atau sistem tepat untuk kebutuhan mereka.
- (b) Kemampuan dipelajari (*Learnability*). Tingkat di mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran tertentu untuk menggunakan produk atau sistem dengan efektivitas, efisiensi, bebas dari resiko dan kepuasan dalam konteks penggunaan tertentu.
- (c) Kemampuan penggunaan (*Operability*). Tingkat di mana produk atau sistem memiliki atribut yang membuatnya mudah untuk dioperasikan dan dikendalikan.
- (d) Perlindungan kesalahan pengguna (*User error protection*). Tingkat di mana sistem dapat melindungi pengguna dari membuat kesalahan.
- (e) Estetika antarmuka pengguna (*User interface aesthetics*). Tingkat di mana antarmuka mampu memberikan interaksi yang menyenangkan dan memuaskan dengan pengguna.
- (f) Aksesibilitas (*Accessibility*). Tingkat di mana produk atau sistem dapat digunakan oleh orang-orang pada cakupan karakteristik dan kapabilitas yang luas untuk meraih tujuan tertentu dalam kontek penggunaan yang tertentu.

5) Keandalan (*Reability*)

Tingkat di mana produk atau komponen melakukan fungsi tertentu dibawah kondisi dalam periode waktu tertentu.

- (a) Kematangan (*Maturity*). Tingkat di mana sistem, produk atau komponen dapat memenuhi kebutuhan keandalan dalam penggunaan normal.
- (b) Ketersediaan (*Availability*). Tingkat di mana sistem, produk atau komponen tersedia dan dapat dioperasikan ketika digunakan.
- (c) Toleransi kesalahan (*Fault tolerance*). Tingkat di mana sistem, produk atau komponen beroperasi sebagaimana dimaksudkan meskipun ada kesalahan perangkat keras maupun lunak.
- (d) Kemampuan pemulihan (*Recoverability*). Tingkat di mana ketika terjadi gangguan atau kegagalan, produk atau sistem dapat memulihkan data yang secara langsung terkena dampak dan kembali ke keadaan yang diinginkan.

6) Keamanan (*Security*)

Tingkat di mana produk atau sistem dapat melindungi informasi dan data sehingga orang-orang, produk lain, atau sistem memiliki tingkat akses data sesuai dengan jenis dan tingkat otoritas mereka.

- (a) Kerahasiaan (*Confidentiality*). Tingkat di mana produk atau sistem menjamin bahwa data hanya dapat diakses oleh mereka yang memiliki kewenangan untuk mengakses.
- (b) Integritas (*Integrity*). Tingkat di mana produk atau komponen dapat mencegah akses tidak sah terhadap, atau memodifikasi, program komputer atau data.

(c) Pengakuan/Bukan penyangkalan (*Non-repudiation*). Tingkat di mana tindakan atau kejadian dapat dibuktikan telah terjadi sehingga peristiwa atau tindakan tersebut tidak dapat ditolak/disangkal kemudian.

(d) Akuntabilitas (*Accountability*). Tingkat di mana tindakan dari suatu entitas dapat ditelusuri secara unik untuk entitas tersebut.

(e) Keaslian (*Aunthenticity*). Tingkat di mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim.

7) Kemampuan pemeliharaan (*Maintainability*)

Tingkat efektivitas dan efisiensi suatu produk atau sistem dapat dimodifikasi oleh pengembang yang dimaksudkan.

(a) Modularitas (*Modularity*). Tingkat di mana sistem atau program komputer terdiri dari komponen berlainan sehingga perubahan pada salah satu komponen memiliki dampak minimal pada komponen lainnya.

(b) Kemampuan penggunaan kembali (*Reuseability*). Tingkat di mana asset-asset dapat digunakan pada lebih dari satu sistem atau dalam membangun asset lain.

(c) Kemampuan analisa (*Analysability*). Tingkat efektivitas dan efisiensi di mana dimungkinkan untuk menilai dampak perubahan yang dimaksudkan untuk satu atau lebih bagian-bagian produk atau sistem, untuk mendiagnosis kekurangan/cacat atau penyebab kegagalan suatu produk, atau untuk mengidentifikasi bagian yang akan dimodifikasi.

(d) Kemampuan modifikasi (*Modifiability*). Tingkat dimana produk atau sistem dapat secara efektif dan efisien dimodifikasi tanpa menimbulkan kerusakan atau menurunkan kualitas produk yang sudah ada.

(e) Kemampuan pengujian (*Testability*). Tingkat efektivitas dan efisiensi di mana kriteria pengujian dapat dibentuk untuk sistem, produk, atau komponen dan pengujian dapat dilakukan untuk menentukan kriteria tersebut sudah dapat dipenuhi.

8) Kemampuan pemindahan (*Portability*)

Tingkat efektivitas dan efisiensi di mana sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau lingkungan operasional atau penggunaan lainnya.

(a) Kemampuan adaptasi (*Adaptability*). Tingkat di mana produk atau sistem dapat secara efektif dan efisien menyesuaikan pada perangkat keras atau lunak yang berbeda/berkembang, atau lingkungan operasional atau penggunaan yang lain.

(b) Kemampuan instalasi (*Instalability*). Tingkat efektivitas dan efisiensi di mana produk atau sistem dapat secara sukses dipasang (install) dan/atau dilepas (uninstal) pada lingkungan tertentu.

(c) Kemampuan penggantian (*Replaceability*). Tingkat di mana produk menggantikan produk perangkat lunak tertentu yang lain untuk tujuan yang sama dalam lingkungan yang sama pula.

## **B. Kajian Penelitian yang Relevan**

1. Penelitian oleh Farizzka Annafi berjudul Pengembangan Aplikasi *virtual tour* Berbantuan Video Sebagai Media Informasi Wilayah Fakultas Teknik Universitas Negeri Yogyakarta. Penelitian ini menggunakan metode R&D dengan medel pengembangan *waterfall* yang terdiri dari 5 tahap yaitu analisis kebutuhan, desain, implementasi, pengujian dan produk. Pengujian dilakukan dengan menggunakan standar ISO 9126 pada aspek *functional suitability, reliability, usability, dan portability*. Berdasarkan hasil pengujian tersebut disimpulkan bahwa apliksai *virtual tour* berbantuan video sangat layak sebagai media informasi wilayah Fakultas Teknik Universitas Negeri Yogyakarta.
2. Penelitian oleh Dimas Hari Trenggono berjudul Sistem Peminjaman Berbasis *Web* Sebagai Media Layanan Di Studio Multimedia SMK 1 Sewon. Dalam penelitian ini metode penelitian yang digunakan adalah R&D. Pengujian sistem dilakukan berpedoman pada ISO 25010:2011 dengan karakteristik pengujian *functional suaitability, preformance efficiency, security, dan portability*. Hasil dari penelitian ini menunjukkan bahwa dengan menggunakan *framework* Codeigniter, menghasilkan perangkat lunak yang berkualitas baik dari segi *functional suitability, performance efficiency, security, maupun portability*.

### C. Kerangka Berpikir

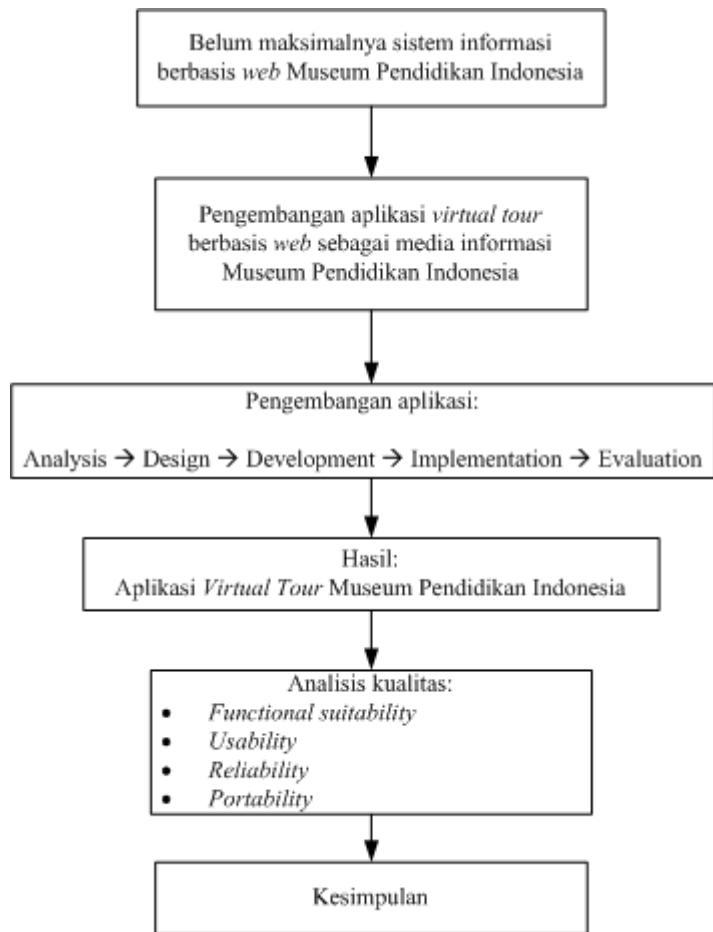
Museum Pendidikan Indonesia merupakan museum pendidikan pertama di Indonesia. Museum ini menyimpan berbagai koleksi berupa benda maupun arsip tentang perkembangan pendidikan di Indonesia. Sebagai lembaga yang mengabadikan perkembangan pendidikan, Museum Pendidikan Indonesia memiliki 5 tujuan penting yaitu: 1) memberikan pemahaman kepada anggota masyarakat dan sivitas akademika tentang eksistensi dan peran Museum Pendidikan. 2) memberikan informasi tentang perkembangan pendidikan nasional baik secara horizontal atau vertikal, baik jenis maupun jenjang pendidikan melalui berbagai koleksi, simbol, dan dokumen yang terkait dengan penyelenggaraan pendidikan di Indonesia dari tahun ke tahun. 3) memberdayakan sivitas akademika UNY dan masyarakat pemerhati pendidikan untuk berkreasi dalam upaya peningkatan kualitas pendidikan. 4) memberikan penghargaan kepada perintis, tokoh dan pejuang pendidikan nasional. 5) menambah dan meningkatkan kualitas sarana dan prasarana wisata kampus yang bersifat edukatif-dan rekreatif. Namun demikian, Museum Pendidikan Indonesia tidak didukung teknologi informasi yang mutakhir dalam mencapai tujuannya. Sehingga fungsi Museum Pendidikan sebagai sumber informasi tentang perkembangan pendidikan nasional tidak maksimal tercapai.

Aplikasi *virtual tour* Museum Pendidikan Indoenesia berbasis *web* ditujukan sebagai pemutakhiran media informasi museum yang ada sekarang. Aplikasi ini menyediakan informasi mengenai Museum Pendidikan Indonesia meliputi: *virtual*

*tour*, profil, alamat, dan informasi koleksi. Dalam fitur *virtual tour* pengguna dapat mengakses Museum Pendidikan Indonesia secara panorama dalam gambar bersudut pandang 360°. Dengan adanya fitur ini diharapkan pengguna dapat merasakan pengalaman mengunjungi Museum Pendidikan Indonesia. Dengan adanya aplikasi ini diharapkan tujuan dan fungsi Museum Pendidikan Indonesia sebagai sumber informasi tentang perkembangan pendidikan nasional dapat tercapai secara makasimal.

Pengembangan aplikasi *virtual tour* Museum Pendidikan Indonesia dilakukan dengan model ADDIE. Tahapan pengembangan aplikasi dimulai dari *analysis, design, development, implantation* dan *evaluation*. Agar perangkat lunak yang dihasilkan dapat diketahui kualitasnya, maka dilakukan serangkaian pengujian sebelum digunakan oleh pengguna. Aplikasi *virtual tour* Museum Pendidikan Indonesia diuji menggunakan standar ISO 25010. ISO 25010 mendefinisikan dua buah model yaitu *quality in use model* dan *product quality model*. *Quality in use model* terbagi dalam 5 karakteristik yaitu: *effectiveness, efficiency, satisfaction, freedom from risk*, dan *context coverage*. Sedangkan *product quality model* terbagi atas 8 karakteristik yaitu: *functional suitability, performane efficiency, compability, usability, reliability, security, maintainability*, serta *portability*. Karena fitur dalam aplikasi *virtual tour* Museum Pendidikan Indonesia tidak mencakup semua aspek karakteristik ISO 25010, maka pengujian yang dilakukan hanya pada aspek *functional suitability, reliability,*

*usability*, dan *portability*. Gambar 4 berikut adalah gambaran kerangka berpikir penelitian.



Gambar 4. Kerangka Pikir Penelitian

#### D. Pertanyaan Penelitian

Pertanyaan penelitian dari pengembangan aplikasi *virtual tour* sebagai media informasi Museum Pendidikan Indonesia adalah sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi *virtual tour* Museum Pendidikan Indonesia berbasis *web* yang sesuai dengan kebutuhan Museum Pendidikan Indonesia?
2. Bagaimana tingkat kualitas aplikasi *virtual tour* Museum Pendidikan Indonesia berbasis *web* yang dikembangkan pada aspek *functional suitability*?
3. Bagaimana tingkat kualitas aplikasi *virtual tour* Museum Pendidikan Indonesia berbasis *web* yang dikembangkan pada aspek *usability*?
4. Bagaimana tingkat kualitas aplikasi *virtual tour* Museum Pendidikan Indonesia berbasis *web* yang dikembangkan pada aspek *reliability*?
5. Bagaimana tingkat kualitas aplikasi *virtual tour* Museum Pendidikan Indonesia berbasis *web* yang dikembangkan pada aspek materi?