

## **BAB II**

### **KAJIAN PUSTAKA**

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

##### **1. Pengembangan Sistem**

##### **a. Belajar dan Pembelajaran**

Belajar memiliki peranan penting dalam pembentukan perilaku dan pribadi seseorang. Perkembangan individu sebagian besar terjadi melalui aktivitas belajar. Aktivitas belajar dapat dilakukan secara psikologis (proses mental) dan fisiologis (proses penerapan atau praktik) (Rusman, 2018: 87). Belajar adalah proses perubahan tingkah laku individu yang relatif tetap sebagai hasil dari pengalaman (Skinner, 2013: 98). Belajar mencakup adanya perkembangan pengetahuan, sikap dan tingkah laku, keterampilan peserta sebagai dampak dari aktivitas mendengar, mengobservasi, mencontoh dan mempraktikkan kegiatan (Rusman, Riyana, & Kurniawan 2013: 50). Belajar adalah proses perubahan dalam kepribadian seseorang, terlihat dari peningkatan kualitas dan kuantitas tingkah laku sebagai hasil dari interaksi antara individu dengan lingkungan atau pengalaman yang telah dilalui (Priansa, 2017: 55). Belajar merupakan usaha sadar seorang individu dalam perubahan perilakunya melalui serangkaian pengalaman dan latihan terkait aspek kognitif, afektif dan psikomotor untuk mencapai suatu tujuan (Khuluqo, 2017: 1).

Menurut Karwono dan Mularsih (2017: 13), beberapa hal yang perlu dicermati dari definisi belajar, diantaranya:

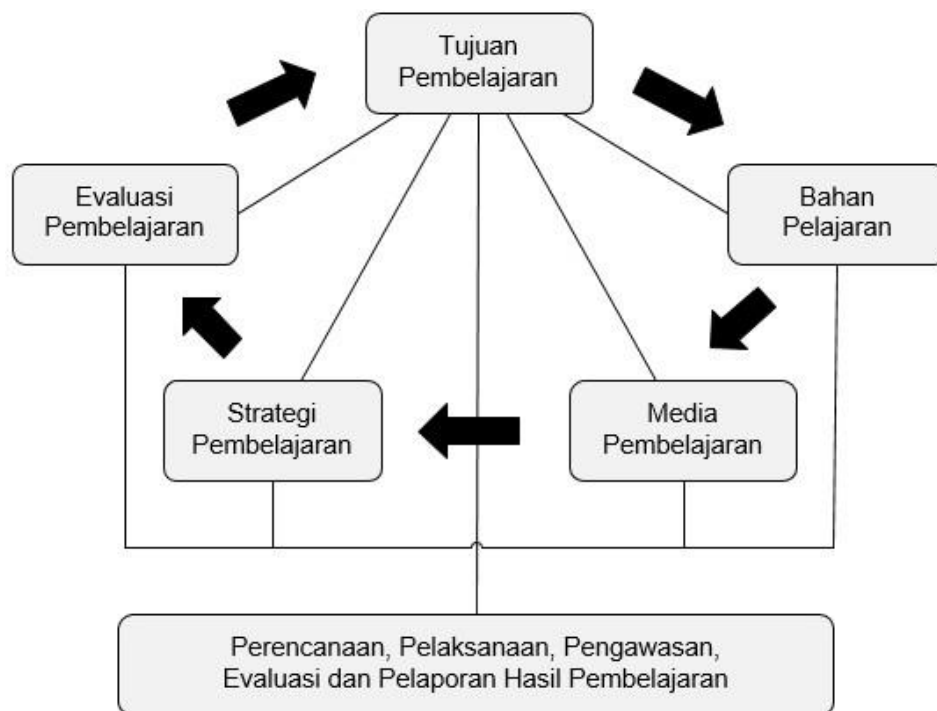
- a. Produk belajar berupa perubahan perilaku, proses mental dimana hasil belajar harus selalu diterjemahkan kedalam tindakan yang dapat diamati.

- b. Perubahan perilaku relatif permanen, artinya menetap dalam jangka yang lama, tetapi dapat hilang atau berubah.
- c. Perubahan perilaku tidak selalu terjadi langsung setelah proses belajar selesai.
- d. Perubahan (potensi behavioral) berasal dari pengalaman atau latihan, bukan karena faktor kematangan (*maturation*) dan insting.
- e. Pengalaman atau latihan harus diperkuat artinya hasil belajar itu bisa hilang/tidak dikuasai lagi jika tidak pernah dialami atau dilatih secara berulang-ulang (terjadi pelemahan karena lupa).

Dari berbagai definisi belajar dapat disimpulkan bahwa belajar adalah perubahan tingkah laku manusia menyangkut peningkatan kualitas dan kuantitas aspek kognitif, afektif, psikomotorik melalui aktivitas (psikologis dan fisiologis), latihan dan pengalaman untuk mencapai tujuan tertentu.

Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional Pasal 1 ayat 20 menerangkan bahwa pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pembelajaran merupakan usaha menata lingkungan untuk menunjang program belajar tumbuh dan berkembang dengan optimal. Proses pembelajaran bersifat eksternal yang sengaja dirancang dan bersifat rekayasa perilaku, sedangkan proses belajar bersifat internal dan unik dalam pribadi peserta didik (Triwiyanto, 2015: 33). Pembelajaran adalah daya upaya agar berlangsung proses belajar yang dikerjakan oleh pendidik untuk peserta didik. Secara implisit, di dalam pembelajaran terdapat aktivitas pengelolaan strategi dan metode untuk mencapai hasil pembelajaran yang diharapkan (Khuluqo, 2017: 52). Pembelajaran

merupakan sistem, terdiri dari berbagai komponen (tujuan, bahan/materi, strategi, media dan evaluasi pembelajaran) saling berinteraksi atau berhubungan satu sama lain. Proses interaksi terjadi baik secara langsung melalui tatap muka maupun secara tidak langsung dengan menggunakan media antara sumber belajar, guru dan siswa (Rusman, 2018: 95). Dari berbagai pengertian pembelajaran dapat ditarik kesimpulan bahwa pembelajaran adalah proses belajar yang tercipta karena adanya interaksi antara pendidik dan peserta didik dengan berbagai komponen meliputi tujuan, materi, metode dan evaluasi baik secara langsung maupun tidak langsung pada suatu lingkungan belajar.



Gambar 1. Hubungan Antar Komponen dalam Manajemen Pembelajaran

Manajemen pembelajaran adalah pemanfaatan sumber daya pembelajaran yang ada, baik faktor internal individu yang sedang belajar maupun faktor eksternal individu guna mencapai tujuan pendidikan secara efektif dan efisien. Aktivitas

manajemen pembelajaran terdiri dari perencanaan, pelaksanaan, pengawasan, evaluasi dan pelaporan hasil belajar (Triwiyanto, 2015: 37). Gambar 1 memperlihatkan hubungan antar komponen manajemen pembelajaran. Manajemen pembelajaran sebagai sebuah sistem, didalamnya terdapat berbagai komponen (tujuan pembelajaran, bahan pembelajaran, media pembelajaran, strategi pembelajaran, dan evaluasi pembelajaran) yang saling berkaitan dan mempengaruhi mutu proses dan hasil pembelajaran. Dalam setiap komponen terdapat aktivitas manajemen mulai dari tahapan perencanaan, pelaksanaan, pengawasan, evaluasi, dan pelaporan hasil pembelajaran. Keterpaduan diantara komponen tersebut akan menghasilkan tujuan pembelajaran yang efektif dan efisien.

**b. Pembelajaran Online (*E-learning*)**

Teknologi informasi dan komunikasi (TIK) menawarkan lingkungan belajar yang menarik dan cepat berkembang, mengaburkan batas-batas antara pendidikan formal dan informal, mengajak guru untuk berinovasi dalam pembelajaran dan memudahkan siswa untuk belajar (Fitriyadi, 2013). Salah satu pemanfaatan TIK dalam dunia pendidikan adalah penggunaan *e-learning*. *E-learning* merupakan sistem pembelajaran yang mendukung dan memfasilitasi penggunaanya untuk belajar melalui komputer dan teknologi berjaringan. Sehingga pengguna bisa melakukan kegiatan belajar baik melalui bantuan komputer saja atau menggunakan komputer yang terkoneksi dengan internet (Wahyuningsih & Rakhmat, 2017: 7). Filosofi *e-learning* bisa dijelaskan sebagai perkembangan penyampaian informasi dan komunikasi dalam pendidikan secara *online*, menyajikan seperangkat teknologi yang dapat menambah nilai-nilai pembelajaran untuk mengatasi tantangan era

globalisasi, dan memperkuat model pembelajaran konvensional melalui pengembangan TIK (Hidayat, Hadi, Basith, & Suwandi, 2018).

*E-learning* berbeda dengan pembelajaran konvensional, fokus utamanya meningkatkan kemampuan personal dan mewujudkan kemandirian belajar. Menurut Littlejohn & Pegle (2007: 12-14) beberapa prinsip penerapan *e-learning* dalam pembelajaran yaitu:

- 1) Personalisasi, penggunaan *e-learning* memungkinkan peserta didik dapat belajar sesuai kecepatannya sendiri berdasarkan minat dan kebutuhan belajarnya. Disamping itu peserta didik mendapat respon secara personal dari pendidik atau salah satu teman tanpa diketahui teman yang lain.
- 2) Keamanan, *e-learning* menawarkan fasilitas untuk menyimpan data atau dokumen berupa catatan, tugas dan ujian dengan aman pada *server*. Data yang disimpan dalam bentuk digital akan tetap aman dan terjaga kontennya selama tidak ada kerusakan pada *server* tersebut.
- 3) Belajar mandiri, memungkinkan peserta didik untuk meninjau ulang materi sesering yang mereka inginkan. Dengan cara ini peserta dapat belajar dengan kecepatan yang berpusat pada kemampuan individual, bukan pada kecepatan yang ditetapkan oleh orang lain.
- 4) Pelacakan, pendidik bisa melakukan pelacakan aktivitas yang dikerjakan peserta didik, seperti penggunaan waktu, bantuan, dan tugas yang berhasil dikumpulkan. Informasi yang diperoleh melalui pelacakan ini dapat digunakan sebagai peringatan ketika peserta didik mengalami kesulitan

belajar. Pelacakan juga dapat berfungsi sebagai masukan bagi pendidik dalam menyusun materi pelajaran.

- 5) Aplikasi pihak ketiga, dukungan fasilitas internet akan mengoptimalkan pengembangan materi yang menarik. Tersedianya berbagai aplikasi pihak ketiga bisa digunakan *user* untuk menyelesaikan pekerjaannya, salah satunya penggunaan LMS untuk mengelola pembelajaran *online*.

Ada dua pendekatan umum untuk *e-learning* yaitu *self-paced* (serba mandiri) dan *instructor-led* (difasilitasi/dipimpin oleh instruktur) (Ghirardini, 2011: 10).

- 1) *Self-paced e-learning*

Peserta didik ditawarkan *e-learning courseware* juga disebut *Web Based Training* (WBT), yang dapat dilengkapi dengan sumber daya tambahan dan penilaian. *Courseware* biasanya ditempatkan di *server web*, dan pelajar dapat mengaksesnya dari *platform* pembelajaran *online* atau pada CD-ROM. Peserta didik bebas untuk belajar dengan kecepatan mereka sendiri dan untuk menentukan jalur pembelajaran pribadi berdasarkan kebutuhan dan minat masing-masing. Penyedia *e-learning* tidak harus menjadwalkan, mengelola, atau melacak peserta didik melalui suatu proses. Konten *e-learning* dikembangkan sesuai dengan serangkaian tujuan pembelajaran. Pendekatan ini harus memberikan dukungan pembelajaran sebanyak mungkin (melalui penjelasan, contoh, interaktivitas, umpan balik, glosarium), untuk membuat pembelajar mandiri. Namun, beberapa jenis dukungan, seperti dukungan teknis berbasis *e-mail* atau bimbingan belajar, biasanya ditawarkan kepada pelajar. Ketika *e-learning* serba otomatis ditawarkan melalui koneksi Internet, ada potensi untuk melacak tindakan peserta didik.

## 2) *Instructor-led and facilitated e-learning*

Dalam model ini, kurikulum linier dikembangkan dengan mengintegrasikan beberapa elemen konten dan kegiatan ke dalam silabus. Kursus dijadwalkan dan dipimpin oleh instruktur atau fasilitator melalui *platform* pembelajaran *online*. Konten *e-learning* untuk belajar individu dapat diintegrasikan dengan kuliah instruktur, tugas individu dan kegiatan kolaboratif di antara peserta didik. Peserta didik, fasilitator, dan instruktur dapat menggunakan alat komunikasi seperti *email*, forum diskusi, obrolan, *polling*, papan tulis, berbagi aplikasi dan konferensi audio video untuk berkomunikasi dan bekerja bersama. Pada akhirnya, langkah terakhir biasanya mencakup latihan atau penilaian untuk mengukur pembelajaran.

Implementasi *e-learning* mempunyai fungsi yang berbeda-beda. Perbedaan tersebut dipengaruhi oleh karakteristik materi, peserta didik, tujuan yang dipakai, sarana dan prasarana yang dimiliki, hingga kebijakan yang diterapkan dalam pelaksanaan pembelajaran. Terdapat tiga fungsi *e-learning* dalam kegiatan pembelajaran yaitu sebagai *supplement* (tambahan), *complement* (pelengkap), dan *replacement* (pengganti). Ketiga fungsi tersebut mencakup dimensi bentuk kegiatan belajar serta cakupan dan jenis materinya, yang dijelaskan sebagai berikut:

### 1) *Supplement*

*E-learning* memiliki fungsi sebagai *supplement* pada dimensi bentuk kegiatan belajar jika dipakai sebagai tambahan untuk memperkaya pengalaman belajar dalam pembelajaran tatap muka. Biasa disebut sebagai pembelajaran difasilitasi *web*. *E-learning* berfungsi sebagai suplemen pada dimensi cakupan dan jenis materi, jika dimanfaatkan sebagai pengayaan terhadap materi yang harus dikuasai

oleh peserta didik. Peserta didik memiliki kebebasan dalam mengakses materi yang ingin dikuasai untuk mendukung materi utama, jumlah dan keberagaman materi yang diakses setiap peserta didik melalui *online* dapat bervariasi. Penggunaan *e-learning* sebagai *supplement* pembelajaran bisa dipakai untuk mengukur tingkat kemandirian belajar peserta didik.

## 2) *Complement*

*E-learning* memiliki fungsi sebagai *complement* pada dimensi bentuk kegiatan belajar jika dimanfaatkan sebagai pelengkap pembelajaran tatap muka. Proporsi pembelajaran tatap muka dengan penggunaan *e-learning* dapat seimbang, sering disebut *hybrid/blended learning*. Fungsi *e-learning* sebagai komplemen pada dimensi cakupan dan jenis materi jika dipakai sebagai pelengkap materi pembelajaran yang diterima peserta didik. Tujuannya supaya peserta didik semakin memantapkan tingkat penguasaan materi yang telah diterima melalui proses pembelajaran. Biasanya penyampaian materi *online* berfungsi sebagai penunjang dalam pembelajaran tatap muka dan sudah diprogramkan oleh pendidik untuk diikuti oleh peserta didik. Materi pelengkap pembelajaran sebaiknya dirancang untuk lebih memperjelas materi utama agar mempermudah peserta didik menguasai kompetensi pembelajaran.

## 3) *Replacement*

*E-learning* memiliki fungsi sebagai *replacement* pada dimensi bentuk kegiatan belajar jika digunakan untuk mengganti pembelajaran tatap muka. Sering disebut sebagai *full e-learning*, sudah mulai dipakai di berbagai instansi atau lembaga di Indonesia. Tujuannya mempermudah peserta didik mengelola kegiatan



pembelajaran sehingga dapat menyesuaikan dengan waktu dan aktivitas lainnya yang memiliki prioritas sama pentingnya. Pembelajaran tatap muka bisa digantikan *e-learning* mulai dari proses pembelajaran sampai kegiatan evaluasinya. Pada dimensi cakupan dan jenis materinya, bentuk pembelajaran ini memberikan berbagai pilihan jenis materi yang dapat diakses oleh peserta didik.

Dari ketiga *fungsi e-learning* yang telah dijelaskan diatas, berikut disajikan proporsi penggunaan *e-learning* yang diadaptasi dari Allen (2013):

Tabel 1. Proporsi Penggunaan *E-learning*

Proporsi	Tipe Pembelajaran	Deskripsi	Fungsi
0 %	Pembelajaran Tradisional	Pembelajaran tatap muka, tidak memanfaatkan fasilitas <i>online</i> , bisa menggunakan bantuan komputer dalam pembelajaran.	<i>Supplement</i>
1 – 29 %	Pembelajaran Difasilitasi	Sebagian kecil aktivitas pembelajaran difasilitasi web walaupun secara keseluruhan pembelajaran masih dilakukan secara tatap muka.	<i>Supplement</i>
30 – 79 %	Pembelajaran Gabungan	Pelaksanaan aktivitas belajar kelas dan materi pembelajaran dilaksanakan dengan campuran antara <i>online</i> dan tatap muka	<i>Complement</i>
80 – 100%	Pembelajaran online penuh	Hampir atau bahkan tidak terdapat pembelajaran tatap muka yang dilakukan dan materi disampaikan secara <i>online</i>	<i>Replacement</i>

### c. **Sistem Informasi Manajemen**

Suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan (Rochaety, 2017: 4). Sistem merupakan seperangkat komponen terkait yang menghasilkan hasil spesifik (Tilley & Rosenblatt, 2017: 5). Menurut Yakub dan Hisbanarto (2014: 1-2) sistem adalah himpunan atau kumpulan dari unsur, variabel-variabel, atau komponen yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu untuk tujuan tertentu. Sedangkan menurut O'Brien dan Marakas (2014: 39) sistem merupakan kelompok komponen yang saling terkait dengan batasan yang jelas, bertujuan mencapai tujuan umum dengan menerima masukan dan menghasilkan keluaran melalui proses transformasi yang terorganisasi. Dari paparan tersebut dapat disimpulkan bahwa sistem adalah sekumpulan komponen yang terintegrasi dan berinteraksi satu sama lain untuk mencapai tujuan tertentu.

Informasi adalah data yang sudah diolah sehingga berguna untuk pembuatan keputusan, sedangkan data adalah representasi suatu objek (Winarno, 2017: 5). Informasi merupakan data yang telah diubah menjadi *output* berharga bagi pengguna, dimana data terdiri dari fakta dasar yang merupakan bahan baku sistem (Tilley & Rosenblatt, 2017: 5). Menurut Rochaety (2017: 7) informasi merupakan kumpulan data yang telah diolah baik bersifat kualitatif maupun kuantitatif menjadi bentuk yang lebih berguna, memiliki nilai nyata, dapat mengurangi derajat ketidakpastian. Data merupakan satu kesatuan yang menggambarkan suatu kejadian (*event*) atau kesatuan nyata (*fact*) (Yakub & Hisbanarto, 2014: 13-18).

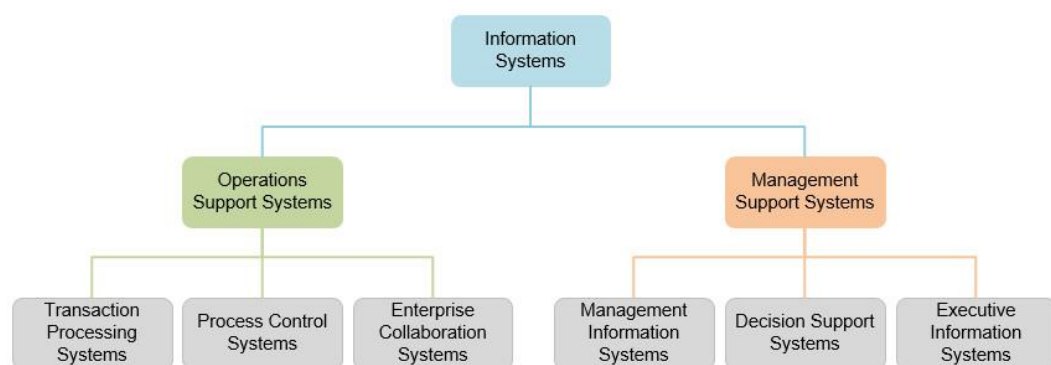
Secara spesifik data merupakan ukuran objektif dari atribut dari entitas. Informasi sebagai data yang sudah diubah menjadi konteks yang bermakna dan berguna bagi pengguna akhir yang spesifik (O'Brien & Marakas, 2014: 35). Dari beberapa definisi tersebut, maka informasi merupakan data yang telah diolah sehingga memiliki nilai dan berguna bagi penggunanya. Sedangkan data adalah representasi atau fakta dasar yang menerangkan suatu entitas.

Manajemen merupakan aktivitas dalam organisasi yang dikerjakan oleh seseorang, manajer, atau pemimpin untuk mewujudkan tujuan bersama. Pemaknaan secara operasional sebagai serangkaian proses mengintegrasikan, mengkoordinasi, menyederhanakan dan mensinkronisasikan sumber daya manusia, material, dan metode (*Man, Material, Methods/3M*) dengan mengimplementasikan fungsi-fungsi manajemen seperti perencanaan, pengorganisasian, pelaksanaan, pengontrolan dan lain-lain supaya tujuan bisa tercapai secara efektif dan efisien. Dukungan data, informasi serta sistem pengelolaannya dibutuhkan untuk menjalankan fungsi-fungsi manajemen (Ramadhina, 2015).

Menurut Laudon & Laudon (2018), sistem informasi adalah sekumpulan komponen yang berfungsi mengumpulkan, menyimpan, mengelola data dan bertujuan untuk memberi informasi, pengetahuan dan produk digital, saling bekerjasama untuk mencapai suatu tujuan. Sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Kadir, 2014: 8). Sistem informasi terdiri dari kombinasi teknologi informasi, orang, dan data untuk mendukung kebutuhan bisnis. Lima

komponen utama adalah perangkat keras, perangkat lunak, data, proses, dan orang-orang (Tilley & Rosenblatt, 2017: 35). Sistem informasi menggunakan sumber daya manusia, perangkat keras, perangkat lunak, data dan jaringan untuk melakukan aktivitas *input, process, output, store* dan *control* yang mengubah sumber daya data menjadi produk informasi (O'Brien & Marakas, 2014: 35). Sehingga dapat dikatakan bahwa sistem informasi adalah sekumpulan komponen yang saling terintegrasi melakukan serangkaian aktivitas pengumpulan data (masukan), manipulasi dan mengubah data menjadi informasi (pemrosesan), disimpan untuk digunakan di masa depan (penyimpanan) atau dikomunikasikan kepada pengguna akhir (keluaran) menurut prosedur pemrosesan yang benar (kendali).

Menurut O'Brien dan Marakas (2010: 13) dalam dunia bisnis saat ini penerapan aplikasi sistem informasi diklasifikasikan dalam beberapa cara berbeda. Beberapa jenis sistem informasi dapat dikategorikan sebagai *support of business operations* (dukungan operasi bisnis) atau *support of managerial decision making* (dukungan pengambilan keputusan manajerial). Menurut Patterson (2005) ada beberapa kategori sistem informasi seperti *Data Processing Systems*, *Management Information Systems*, *Decision Support Systems* dan *Executive Information System*.



Gambar 2. Klasifikasi Sistem Informasi

Sistem informasi manajemen merupakan perpaduan antara sumber daya manusia dan aplikasi teknologi informasi untuk memilih, menyimpan, mengolah dan mengambil kembali data dalam rangka mendukung proses pengambilan keputusan sebuah perusahaan (Rochaety, 2017: 11). Sistem informasi manajemen (SIM) atau *Management Information System* (MIS) adalah sistem informasi yang digunakan untuk menyajikan informasi, mendukung operasi, manajemen, dan pengambilan keputusan dalam suatu organisasi. Sistem informasi manajemen menghasilkan informasi untuk memantau kinerja, memelihara koordinasi, dan menyediakan informasi untuk operasi organisasi (Kadir, 2014: 106). Sistem informasi manajemen secara formal adalah suatu subsistem informasi yang menyeluruh dan terkoordinasi secara rasional, serta mampu mentransformasikan data menjadi informasi berkualitas kepada pihak-pihak yang membutuhkan sehingga mampu meningkatkan efisiensi, efektivitas dan produktivitas (Yakub & Hisbanarto, 2014: 61-62). Beberapa definisi terkait sistem informasi manajemen dapat disimpulkan menjadi suatu sistem informasi yang mampu memanipulasi data menjadi informasi kepada pihak yang membutuhkan dalam rangka mendukung operasi manajemen dan pengambilan keputusan dalam suatu organisasi.

#### **d. *Software***

*Software* atau perangkat lunak merupakan program yang dijalankan komputer untuk fungsi tertentu. Perangkat lunak merujuk pada program yang melakukan kontrol terhadap perangkat keras untuk menghasilkan hasil yang diharapkan. Perangkat lunak meliputi perangkat lunak aplikasi dan perangkat lunak sistem (Tilley & Rosenblatt, 2017: 7). Perangkat lunak komputer terdiri dari detail,

instruksi terprogram yang mengontrol dan mengkoordinasikan komponen perangkat keras komputer dalam pemrosesan informasi (Laudon & Laudon, 2018: 51). *Software* adalah program komputer dan dokumentasi terkait yang memberikan fungsionalitas dan kinerja yang diperlukan oleh pengguna (Sommerville, 2011: 6). *Software* merupakan sekumpulan perintah program komputer, ketika dijalankan memberikan fungsi, fitur dan kinerja yang diharapkan, deskripsi informasi mengenai operasi dan kegunaan program-programnya dalam bentuk hard copy dan virtual (Pressman, 2015: 4).

Saat ini terdapat tujuh kategori dengan cakupan luas terkait perangkat lunak komputer, diantaranya sebagai berikut:

- 1) *System software*, perangkat lunak yang dirancang untuk berkomunikasi atau sebagai *interface* antara program komputer dengan *hardware*.
- 2) *Application software*, program *stand alone* untuk menyelesaikan kebutuhan bisnis tertentu sehingga memudahkan kerja bisnis dan pembuatan keputusan.
- 3) *Scientific/Engineering software*, ditandai dengan algoritma “angka-angka”. Berbeda jauh dengan algoritma numerik konvensional, telah mulai mengambil secara *real-time* dan bahkan karakteristik perangkat lunak sistem.
- 4) *Embedded software*, sistem tertanam dalam suatu produk, biasa dipakai untuk menerapkan dan mengontrol fitur serta fungsi untuk kebutuhan pengguna akhir.
- 5) *Product-line-software*, perangkat lunak dirancang dengan kemampuan khusus agar bisa dipakai oleh banyak kalangan pengguna. Dapat fokus pada pasar terbatas atau ditujukan untuk pasar massal.

- 6) *Web application*, perangkat lunak yang diakses melalui jaringan ini meliputi beragam aplikasi diantaranya berbasis *browser* dan perangkat *mobile*.
- 7) *Artificial intelligence software*, memanfaatkan algoritma non-numerik dalam menyelesaikan masalah rumit yang tidak sesuai dengan perhitungan atau analisis langsung.

**e. *Software Engineering***

*Software engineering* adalah proses pengembangan perangkat lunak yang menekankan desain yang solid, struktur yang efektif, dokumentasi yang akurat, dan pengujian yang cermat. Penerapan disiplin dari prinsip-prinsip teknik dalam pembuatan aplikasi yang kompleks dan berumur panjang. *Software engineering* lebih luas daripada pengembangan yang mencakup lima bidang kegiatan teknis : *requirements, design, construction, maintenance* dan *evolution*. Didukung juga oleh kegiatan non-teknis seperti estimasi biaya dan usaha, manajemen proyek dan peningkatan proses (Tilley & Rosenblatt, 2017: 405). *Software engineering* merupakan disiplin teknik terkait keseluruhan aspek produksi *software* dengan aktivitas fundamental : *software specification, software development, software validation, and software evolution* (Sommerville, 2011: 6). Menurut Pressman (2015: 15-16) *software engineering* merupakan teknologi berlapis yang meliputi:



Gambar 3. *Software Engineering Layers*

1) *A quality focus* (Fokus kualitas)

Pendekatan teknik apapun termasuk rekayasa perangkat lunak harus didasarkan pada komitmen organisasi terhadap kualitas. *Total Quality Management* (TQM), Six Sigma dan filosofi serupa menumbuhkan budaya peningkatan proses yang berkelanjutan. Pada akhirnya budaya tersebut mengarahkan pada pendekatan pengembangan yang semakin efektif dalam Rekayasa Perangkat Lunak (RPL). Fokus pada kualitas adalah lapisan dasar yang mendukung RPL.

2) *Process* (Proses)

Lapisan proses adalah fondasi untuk rekayasa perangkat. Proses RPL adalah perekat yang menyatukan lapisan teknologi dan memungkinkan pengembangan program komputer secara rasional dan tepat waktu. Mendefinisikan *framework* agar teknologi yang digunakan dalam pembuatan program dapat dimanfaatkan dengan efektif. Membentuk dasar pengendalian manajemen proyek dan menetapkan konteks dimana metode teknis diterapkan, tahapan ditetapkan, kualitas terjamin dan perubahan dikelola dengan benar.

3) *Methods* (Metode)

Metode RPL menyediakan cara teknis untuk mengembangkan perangkat lunak. Metode meliputi beragam tugas diantaranya komunikasi, analisis kebutuhan, perancangan model, konstruksi program, pengujian, dan dukungan. Metode RPL mengandalkan seperangkat prinsip dasar yang mengatur setiap bidang teknologi, mencakup aktivitas pemodelan dan teknik deskriptif lainnya.



#### 4) *Tools* (Alat)

Menyediakan dukungan otomatis atau semi-otomatis untuk proses dan metode. Saat *tools* terintegrasi maka informasi yang dibuat oleh satu alat dapat dimanfaatkan oleh yang lain, sebuah sistem untuk mendukung pengembangan perangkat lunak, yang disebut rekayasa perangkat lunak berbantuan komputer.

#### **f. Metode Pengembangan Perangkat Lunak**

*Software process* adalah serangkaian kegiatan terkait yang mengarah pada produksi produk perangkat lunak. Setiap model proses mewakili suatu proses dari perspektif tertentu, dan dengan demikian hanya memberikan informasi parsial tentang proses itu. Ada banyak proses perangkat lunak yang berbeda tetapi semua harus mencakup empat kegiatan yang mendasar untuk rekayasa perangkat lunak (Sommerville, 2011:28).

- 1) *Software specification*, penentuan fungsionalitas perangkat lunak beserta batasan-batasan operasinya.
- 2) *Software design and implementation*, produksi perangkat lunak sesuai kebutuhan pengguna (spesifikasi).
- 3) *Software validation*, validasi perangkat lunak melalui serangkaian pengujian untuk memastikan memberikan kinerja sesuai harapan pelanggan.
- 4) *Software evolution*, pembaharuan perangkat lunak untuk mengakomodasi kebutuhan pelanggan yang berubah.

Proses perangkat lunak sebagai kerangka kerja atau model berisikan sekumpulan *process element* (kegiatan, tindakan, dan tugas) dengan *process flow* (cara di mana elemen-elemen proses saling terkait) untuk membangun perangkat

lunak berkualitas tinggi. Proses perangkat lunak memiliki kerangka kerja umum meliputi *communication*, *planning*, *modeling*, *construction*, dan *deployment* yang berlaku untuk semua proyek perangkat lunak (Pressman, 2015: 17).

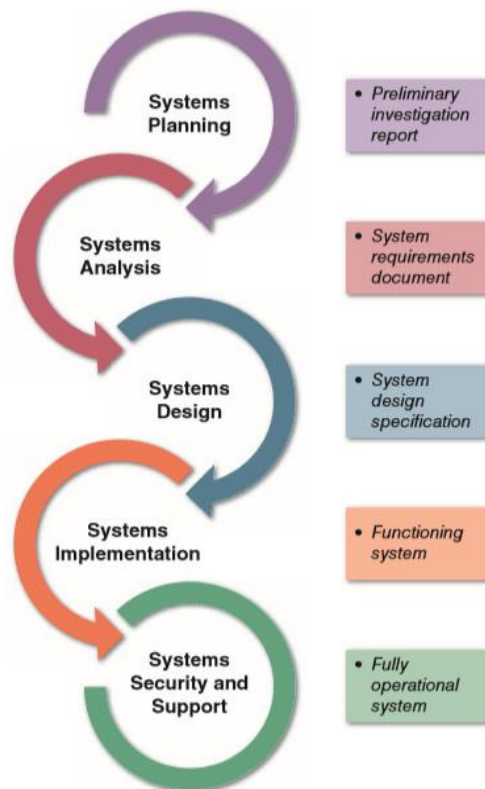
- 1) *Communication*, melakukan pembahasan bersama pelanggan untuk memahami tujuan proyek dan mengumpulkan persyaratan yang digunakan dalam menentukan fitur dan fungsi perangkat lunak.
- 2) *Planning*, mendefinisikan pekerjaan dengan menjelaskan teknis kerja, risiko, kebutuhan sumber daya, produk kerja, dan penjadwalan.
- 3) *Modeling*, melakukan analisis dan desain arsitektur sistem untuk lebih mengerti persyaratan perangkat lunak yang akan digunakan dalam proyek.
- 4) *Construction*, aktivitas ini meliputi pembuatan dan pengujian kode program agar tidak terjadi kesalahan dan fungsi berjalan sebagaimana mestinya.
- 5) *Deployment*, pemindahan perangkat lunak ke lingkungan pengguna untuk dievaluasi dan memberikan umpan balik berdasarkan hasil evaluasi.

Ada banyak pilihan untuk mengembangkan sistem informasi, tetapi alternatif yang paling populer adalah *structured analysis*, yang merupakan metode tradisional yang masih banyak digunakan, *object-oriented (O-O) analysis*, yang merupakan pendekatan yang lebih baru yang disukai banyak analis, dan *agile method* juga disebut *adaptive methods* yang mencakup tren terbaru dalam pengembangan perangkat lunak. (Tilley & Rosenblatt, 2017: 17) Ada banyak alternatif, dan para pakar teknologi informasi sepakat bahwa tidak ada metode pengembangan tunggal yang terbaik dalam semua kasus. Suatu pendekatan yang bekerja dengan baik untuk satu proyek mungkin memiliki kekurangan atau risiko dalam situasi lain. Yang

penting adalah memahami berbagai metode dan kekuatan serta kelemahan masing-masing pendekatan.

1) *Structured analysis*

Analisis terstruktur merupakan pendekatan pengembangan sistem tradisional yang mudah dimengerti dan sudah teruji oleh waktu. Menerapkan serangkaian fase *Software Development Life Cycle* (SDLC), untuk merencanakan, menganalisis, merancang, mengimplementasikan, dan mendukung sistem informasi. Analisis terstruktur didasarkan pada rencana keseluruhan, mirip dengan cetak biru untuk membangun gedung, sehingga disebut pendekatan prediktif. Analisis terstruktur memakai serangkaian model proses untuk menggambarkan sistem secara grafis. Teknik ini berpusat pada proses karena memiliki fokus pada proses merubah data menjadi informasi yang bermanfaat. Selain memodelkan proses, analisis terstruktur juga membahas organisasi dan struktur data, desain basis data relasional, dan masalah antarmuka pengguna. Model proses menunjukkan data yang mengalir masuk dan keluar dari proses sistem. Di dalam setiap proses, input data ditransformasikan oleh aturan bisnis yang menghasilkan output. Analisis terstruktur menggunakan SDLC untuk merencanakan dan mengelola proses pengembangan sistem. SDLC menggambarkan aktivitas dan fungsi yang dilakukan semua pengembang sistem, terlepas dari pendekatan apa yang mereka gunakan. Hasil dari setiap fase pada *waterfall* disebut *deliverable*, yang mengalir ke fase berikutnya.



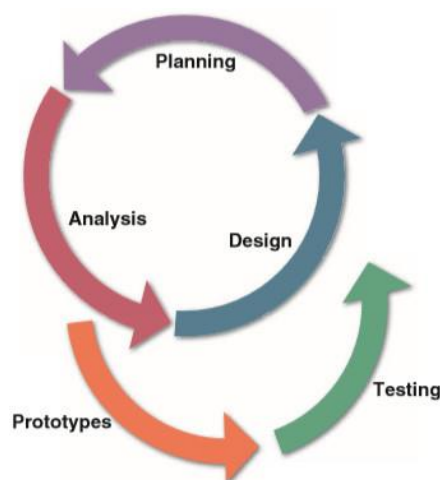
Gambar 4. Fase Pengembangan dan *deliverable Model Waterfall*

Beberapa analis melihat kelemahan dalam struktur SDLC yang diterapkan, karena model *waterfall* tidak menekankan interaktivitas di antara fase. Kritik ini bisa valid jika fase SDLC diikuti terlalu kaku. Namun, fase yang berdekatan dapat dan memang berinteraksi, seperti yang ditunjukkan oleh panah melingkar pada Gambar 4, dan interaksi di antara beberapa fase tidak jarang. Digunakan dengan cara ini, model tradisional tidak berbeda dari metode tangkas. Model SDLC biasanya mencakup lima langkah, meliputi: perencanaan sistem, analisis sistem, desain sistem, implementasi sistem, dan dukungan dan keamanan sistem.

## 2) *Object-oriented (O-O) analysis*

Sedangkan analisis terstruktur memperlakukan proses dan data sebagai komponen terpisah, analisis berorientasi objek menggabungkan data dan proses

yang bertindak pada data menjadi sesuatu yang disebut objek. Analisis sistem menggunakan O-O untuk memodelkan proses dan operasi bisnis dunia nyata. Hasilnya adalah seperangkat objek perangkat lunak yang mewakili orang, benda, transaksi, dan peristiwa aktual. Menggunakan bahasa pemrograman O-O, seorang *programmer* kemudian menulis kode yang menciptakan objek. Objek adalah anggota dari suatu kelas, yang merupakan kumpulan dari benda-benda serupa. Objek memiliki karakteristik yang disebut properti, yang diwarisi objek dari kelasnya atau dimiliki sendiri. Dalam desain O-O, proses bawaan yang disebut metode dapat mengubah properti objek. Satu objek dapat mengirim informasi ke objek lain dengan menggunakan pesan. Pesan meminta perilaku atau informasi spesifik dari objek lain. Analisis O-O menggunakan model objek untuk mewakili data dan perilaku, dan untuk menunjukkan bagaimana objek mempengaruhi objek lain. Dengan menjelaskan objek dan metode yang diperlukan untuk mendukung operasi bisnis, pengembang sistem dapat merancang komponen yang dapat digunakan kembali sehingga mempercepat implementasi sistem dan mengurangi biaya pengembangan.



Gambar 5. Model Pengembangan *Object-Oriented*

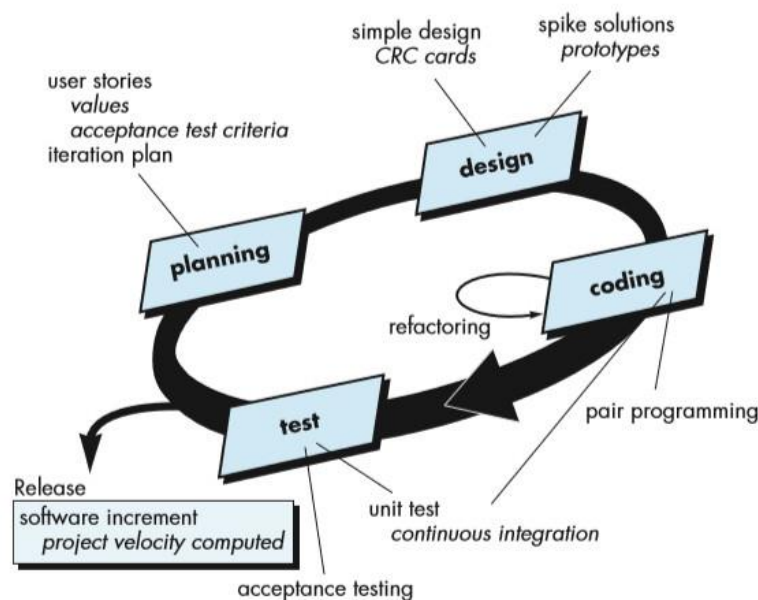
Metode berorientasi objek biasanya mengikuti serangkaian fase analisis dan desain yang mirip dengan SDLC, meskipun ada sedikit kesepakatan tentang jumlah fase dan namanya. Dalam model O-O, fase cenderung lebih interaktif. Gambar 5 menunjukkan model pengembangan O-O dimana kegiatan perencanaan, analisis, dan desain berinteraksi untuk menghasilkan prototipe yang dapat diuji dan diimplementasikan. Hasilnya adalah model interaktif yang dapat secara akurat menggambarkan proses bisnis dunia nyata. Metodologi O-O populer karena menyediakan transisi yang mudah ke bahasa pemrograman O-O seperti C ++, Java, dan Swift.

### 3) *Agile Methods*

Teknik pengembangan berubah seiring waktu. Sebagai contoh, analisis terstruktur adalah pendekatan tradisional, dan metode *agile* adalah pengembangan terbaru. Analisis terstruktur membangun rencana keseluruhan untuk sistem informasi, seperti halnya cetak biru untuk membangun gedung. Metode *agile*, sebaliknya, berupaya mengembangkan sistem secara bertahap, dengan membangun serangkaian prototipe dan terus-menerus menyesuaikannya dengan kebutuhan pengguna. Seiring proses *agile* berlanjut, pengembang merevisi, memperluas, dan menggabungkan versi sebelumnya menjadi produk akhir. Pendekatan *agile* menekankan umpan balik yang berkelanjutan, dan setiap langkah tambahan dipengaruhi oleh apa yang dipelajari dalam langkah sebelumnya.

Metode *agile* biasanya menggunakan model spiral, yang mewakili serangkaian iterasi, atau revisi, berdasarkan umpan balik pengguna. Seiring proses berlanjut, produk akhir secara bertahap berkembang. Pendekatan *agile*

membutuhkan interaktivitas intens antara pengembang dan pengguna individu, dan tidak dimulai dengan tujuan keseluruhan. Sebaliknya, proses *agile* menentukan hasil akhir. Para pendukung model spiral percaya bahwa pendekatan ini mengurangi risiko dan mempercepat pengembangan perangkat lunak. Barry Boehm, seorang profesor rekayasa perangkat lunak terkemuka, awalnya menyarankan model spiral pada 1990-an. Dia menyatakan bahwa setiap iterasi, atau fase, dari model harus memiliki tujuan spesifik yang diterima, ditolak, atau diubah oleh pengguna atau klien. Dengan demikian, setiap iterasi menghasilkan umpan balik dan peningkatan, yang memungkinkan tim untuk mencapai tujuan proyek secara keseluruhan. Biasanya, setiap iterasi dalam model spiral termasuk perencanaan, analisis risiko, teknik, dan evaluasi. Iterasi berulang menghasilkan serangkaian prototipe, yang berkembang menjadi sistem jadi. Perhatikan bahwa fase-fase ini menyerupai tugas-tugas SDLC, yang juga bisa berulang.



Gambar 6. *Extreme Programming Process*

Ada banyak variasi adaptif dan metode terkait lainnya, dan sebagian besar pengembang teknologi informasi mengharapkan tren ini berlanjut di masa mendatang. Dua contoh adalah Scrum dan Extreme Programming (XP). Meskipun metode *agile* menjadi populer, analis harus menyadari bahwa pendekatan ini memiliki kelebihan dan kekurangan. Secara alami, metode *agile* dapat memungkinkan pengembang menjadi lebih fleksibel dan responsif, tetapi bisa lebih berisiko daripada metode yang lebih tradisional. Misalnya, tanpa serangkaian persyaratan sistem yang terperinci, fitur tertentu yang diminta oleh beberapa pengguna mungkin tidak konsisten dengan rencana permainan perusahaan yang lebih besar. Kerugian potensial lain dari metode *agile* dapat mencakup dokumentasi yang lemah, garis akuntabilitas yang kabur, dan terlalu sedikit penekanan pada gambaran bisnis yang lebih besar. Selain itu, ketika tidak diterapkan dengan benar, serangkaian iterasi panjang akan menambah biaya dan waktu pengembangan. Intinya adalah bahwa analis sistem harus memahami pro dan kontra dari setiap pendekatan sebelum memilih metode pengembangan untuk proyek tertentu.



Tabel 2. Perbandingan Ketiga Metode Pengembangan Sistem

	<i>Structured</i>	<i>Object-Oriented</i>	<i>Agile</i>
<b>Deskripsi</b>	<ul style="list-style-type: none"> <li>• Merepresentasikan sistem dalam hal data dan proses sebagai komponen terpisah.</li> <li>• Pengembangan sistem diatur dalam beberapa fase, dengan hasil dan tonggak untuk mengukur kemajuan.</li> <li>• Model air terjun biasanya terdiri dari lima fase: persyaratan, desain, konstruksi, pengujian, dan pemeliharaan &amp; evolusi.</li> </ul>	<ul style="list-style-type: none"> <li>• Tampilan sistem dalam hal objek yang menggabungkan data dan proses. Objek mewakili orang, benda, transaksi, dan peristiwa aktual.</li> <li>• Objek mewakili orang, benda, transaksi, dan peristiwa aktual.</li> <li>• Dibandingkan dengan analisis struktural, fase O-O cenderung lebih interaktif.</li> </ul>	<ul style="list-style-type: none"> <li>• Menekankan upaya berbasis tim yang intens. Memecah perkembangan menjadi siklus, atau iterasi, yang menambah fungsionalitas.</li> <li>• Setiap siklus dirancang, dibangun, dan diuji dalam proses yang sedang berlangsung.</li> <li>• Upaya untuk mengurangi risiko utama dengan langkah bertahap dalam interval waktu singkat.</li> </ul>

	<i>Structured</i>	<i>Object-Oriented</i>	<i>Agile</i>
<b>Modeling Tools</b>	<ul style="list-style-type: none"> <li>• <i>Data flow diagram</i> (DFD) dan deskripsi proses, juga pemodelan proses bisnis</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Unified Modeling Language</i> (UML) menggambarkan aktor sistem, metode, dan pesan. juga pemodelan proses bisnis.</li> </ul>	<ul style="list-style-type: none"> <li>• Alat yang meningkatkan komunikasi, seperti perangkat lunak kolaboratif, brainstorming, dan <i>whiteboards</i>.</li> <li>• Pemodelan proses bisnis bekerja dengan baik dengan metode <i>agile</i>.</li> </ul>
<b>Pro</b>	<ul style="list-style-type: none"> <li>• Metode tradisional yang telah sangat populer dari waktu ke waktu. Sangat bergantung pada dokumentasi tertulis.</li> <li>• Iterasi fase yang sering dapat memberikan fleksibilitas yang sebanding dengan metode lain.</li> </ul>	<ul style="list-style-type: none"> <li>• Terintegrasi dengan mudah dengan bahasa pemrograman berorientasi objek.</li> <li>• Kode bersifat modular dan dapat digunakan kembali, yang dapat mengurangi biaya dan waktu pengembangan.</li> </ul>	<ul style="list-style-type: none"> <li>• Sangat fleksibel dan efisien dalam menghadapi perubahan.</li> <li>• Menekankan interaksi tim dan mencerminkan serangkaian nilai berbasis komunitas.</li> <li>• Hasil yang sering secara konstan memvalidasi proyek dan mengurangi risiko.</li> </ul>

	<i>Structured</i>	<i>Object-Oriented</i>	<i>Agile</i>
	<ul style="list-style-type: none"> <li>• Cocok untuk alat dan teknik manajemen proyek tradisional</li> </ul>	<ul style="list-style-type: none"> <li>• Mudah dirawat dan dikembangkan karena memiliki sifat pewarisan.</li> </ul>	
<b>Kontra</b>	<ul style="list-style-type: none"> <li>• Perubahan bisa mahal, terutama di fase selanjutnya.</li> <li>• Persyaratan didefinisikan lebih awal, dan dapat berubah selama pengembangan.</li> <li>• Pengguna mungkin tidak dapat menggambarkan kebutuhan mereka sampai mereka dapat melihat contoh fitur dan fungsi.</li> </ul>	<ul style="list-style-type: none"> <li>• Metode yang agak baru mungkin kurang akrab bagi anggota tim pengembangan.</li> <li>• Interaksi objek dan kelas bisa rumit dalam sistem yang lebih besar.</li> </ul>	<ul style="list-style-type: none"> <li>• Anggota tim membutuhkan keterampilan teknis dan komunikasi tingkat tinggi.</li> <li>• Kurangnya struktur dan dokumentasi dapat menyebabkan faktor risiko.</li> <li>• Keseluruhan proyek mungkin mengalami perubahan ruang lingkup saat kebutuhan pengguna berubah.</li> </ul>

**g. *Rational Unified Process (RUP)***

*Unified Process* adalah sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan inkremental (bertahap dengan progres menaik). Iteratif bisa dilakukan di setiap tahap, atau iteratif tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental (bertambah menaik) dimana setiap iterasi akan memperbaiki iterasi berikutnya. Salah satu *Unified Process* yang terkenal adalah *Rational Unified Process* (Rosa & Shalahudin 2016: 124).

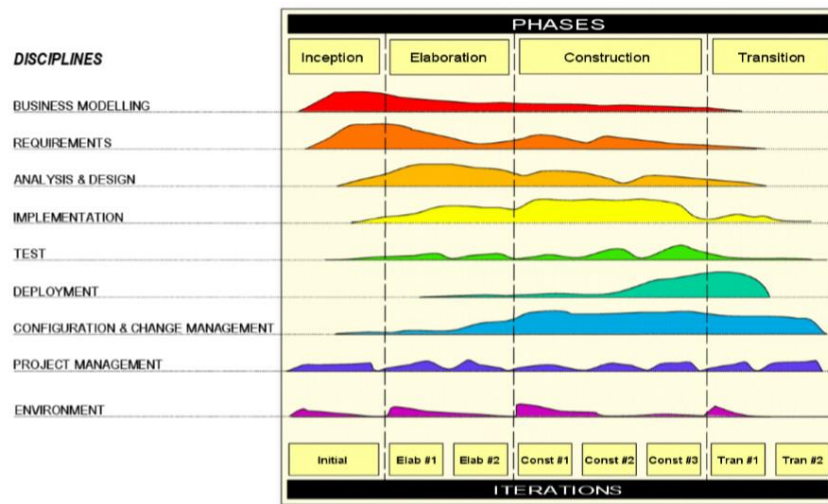
*Rational Unified Process (RUP)* adalah model pengembangan perangkat lunak yang menggunakan teknik disiplin dalam penetapan tugas dan tanggung jawab untuk menghasilkan perangkat lunak berkualitas tinggi, memenuhi kebutuhan pengguna, dalam estimasi waktu dan biaya yang dapat diprediksi (Rational Software, 1998). *Rational Unified Process* didasarkan pada prinsip-prinsip *use case driven, architecture centered, iterative and incremental, risk oriented* (Wazlawick, 2014: 3). Rosa & Shalahudin (2016: 125) menambahkan bahwa RUP adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*) khusus pemrograman berorientasi objek yang dikembangkan oleh Rational Software sebuah divisi dari IBM. RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well-defined*) dan penstrukturan yang baik (*well structured*).

Pendekatan iteratif/perulangan dari *Rational Unified Process* dapat mengakomodir beberapa kekurangan pengembangan tanpa konsep perulangan layaknya pada model *waterfall*, diantaranya sebagai berikut:

- 1) Mengakomodasi perubahan kebutuhan perangkat lunak.
- 2) Pengintegrasian bukanlah proses besar dan cepat di akhir proyek.
- 3) Risiko biasa ditemukan atau dialamatkan selama pengintegrasian awal.
- 4) Manajemen artinya membuat perubahan taktik pada produk.
- 5) *Reusable* artinya mendukung fasilitas penggunaan kembali.
- 6) Kesalahan bisa ditemukan dan diperbaiki pada beberapa iterasi sehingga menghasilkan arsitektur yang baik dan aplikasi berkualitas tinggi.
- 7) Lebih baik menggunakan anggota proyek dibandingkan susunan secara seri pada tim proyek.
- 8) Anggota tim belajar selama proyek berlangsung.
- 9) Perbaikan perangkat lunak dapat dilakukan seiring proses pengembangannya.

*Rational Unified Process* menyajikan pengembangan perangkat lunak sebagai aktivitas (*workflow*) berulang disusun pada empat tahapan fase yaitu *inception*, *elaboration*, *construction*, dan *transition* (Sommerville, 2011: 121). RUP biasanya digambarkan dari tiga perspektif:

- 1) Perspektif dinamis, menunjukkan fase model dari waktu ke waktu.
- 2) Perspektif statis, menunjukkan aktivitas proses yang diberlakukan.
- 3) Perspektif praktik, menyarankan praktik yang baik digunakan selama proses.



Gambar 7. Model Pengembangan Perangkat Lunak RUP

Model bertahap RUP mengidentifikasi empat fase diskrit dalam proses perangkat lunak. Tidak seperti model *waterfall* dimana fase disamakan dengan kegiatan proses, fase dalam RUP lebih terkait dengan bisnis daripada masalah teknis. Berikut penjelasan empat fase pengembangan RUP:

#### 1) *Inception*

Tahap *inception* memiliki fokus utama terhadap memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan sistem yang akan dibuat (*requirements*). Tujuan dari fase *inception* adalah menetapkan *business case* sistem. Mengidentifikasi semua entitas eksternal (orang dan sistem) yang akan berinteraksi dengan sistem dan menentukan interaksinya. Selanjutnya menggunakan informasi tersebut untuk menilai kontribusi yang diberikan sistem kepada bisnis. Apabila kontribusi tidak signifikan, proyek bisa dibatalkan.

#### 1. *Elaboration*

Tahap ini lebih pada proses analisis dan desain perancangan arsitektur sistem. Mempunyai tujuan untuk mengembangkan pemahaman terkait domain masalah,

membangun kerangka kerja arsitektur sistem, mengembangkan rencana proyek, mengidentifikasi risiko proyek utama. Setelah menyelesaikan fase ini, diharuskan memiliki *requirements model* untuk sistem, yang mungkin berupa serangkaian UML *use-cases*, deskripsi arsitektur, dan rencana pengembangan untuk perangkat lunak.

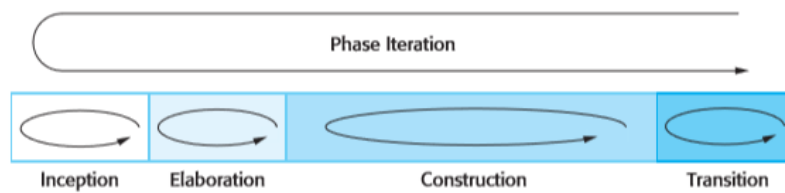
## 2) *Construction*

Tahap ini lebih pada pengembangan komponen dan fitur-fitur sistem, khususnya implementasi dan pengujian sistem yang fokus pada kode program. Tahap *construction* melibatkan desain sistem, pemrograman, dan pengujian. Bagian dari sistem dikembangkan secara paralel dan terintegrasi selama fase ini. Setelah menyelesaikan fase ini, diharuskan memiliki sistem perangkat lunak yang berfungsi dan dokumentasi terkait yang siap dikirim ke pengguna.

## 3) *Transition*

Tahapan *transition* lebih fokus terhadap *deployment* sistem supaya diketahui oleh *user* atau pengguna. Berkaitan dengan pemindahan sistem dari komunitas pengembangan ke komunitas pengguna dan membuatnya bekerja di lingkungan nyata. Setelah menyelesaikan fase ini, diharuskan memiliki sistem perangkat lunak terdokumentasi yang berfungsi dengan benar di lingkungan operasionalnya.

Iterasi dalam RUP didukung dengan dua cara. Setiap fase dapat diberlakukan secara iteratif dengan hasil yang dikembangkan secara bertahap. Selain itu, seluruh rangkaian fase juga dapat diberlakukan secara bertahap seperti yang ditunjukkan oleh panah perulangan dari pada Gambar 8.



Gambar 8. Iterasi dalam RUP

Perspektif statis dari *Rational Unified Process* berfokus pada kegiatan yang terjadi selama proses pengembangan. Ini disebut alur kerja (*workflow*) dalam deskripsi RUP. Ada enam alur kerja proses inti (*core process workflows*) dan tiga alur kerja pendukung inti (*core supporting workflows*) yang diidentifikasi dalam proses. RUP telah dirancang bersama dengan UML, sehingga uraian alur kerja berorientasi di sekitar model UML terkait seperti *sequence models*, *object model*, dan *component model*.

Tabel 3. *Static Workflow* dalam *Rational Unified Process* (RUP)

No	Workflow	Deskripsi
1	<i>Business modelling</i>	Memodelkan proses bisnis menggunakan <i>business use cases</i> .
2	<i>Requirements</i>	Mengidentifikasi aktor yang berinteraksi dengan sistem dan <i>use cases</i> dikembangkan untuk memodelkan persyaratan sistem.
3	<i>Analysis and design</i>	<i>Design model</i> dibuat dan didokumentasikan menggunakan <i>architectural models</i> , <i>component models</i> , <i>object models</i> , dan <i>sequence models</i> .
4	<i>Implementation</i>	Komponen dalam sistem diimplementasikan dan disusun menjadi sub-sistem implementasi.
5	<i>Testing</i>	Proses pengujian dilakukan berulang bersamaan dengan implementasi. Pengujian sistem mengikuti selesainya implementasi.



No	Workflow	Deskripsi
6	<i>Deployment</i>	Rilis produk dibuat, didistribusikan ke pengguna, dan diinstal di tempat kerja operasionalnya.
7	<i>Configuration and change management</i>	Berkaitan dengan kebijakan, proses dan alat untuk mengelola perubahan sistem perangkat lunak. Melibatkan kegiatan: manajemen perubahan, manajemen versi, membangun sistem dan manajemen rilis.
8	<i>Project management</i>	Alur kerja pendukung ini mengelola perubahan dan prosedur serta alat manajemen konfigurasi.
9	<i>Environment</i>	Berkaitan dengan pembuatan perangkat lunak yang sesuai dengan ketersediaan alat untuk tim pengembangan perangkat lunak.

Keuntungan dalam menyajikan perspektif dinamis dan statis adalah bahwa fase proses pengembangan tidak terkait dengan alur kerja tertentu. Pada prinsipnya, setidaknya semua alur kerja RUP mungkin aktif di semua tahap proses. Pada fase awal proses, sebagian besar upaya mungkin akan dihabiskan untuk alur kerja seperti pemodelan dan persyaratan bisnis, kemudian pada fase selanjutnya dalam pengujian dan penyebaran.

Perspektif praktik pada RUP menggambarkan praktik rekayasa perangkat lunak yang baik direkomendasikan untuk digunakan dalam pengembangan sistem. Enam praktik terbaik mendasar yang direkomendasikan:

- 1) Mengembangkan perangkat lunak secara iteratif. Merencanakan penambahan sistem berdasarkan prioritas pelanggan dan mengembangkan fitur sistem prioritas tertinggi di awal proses pengembangan.

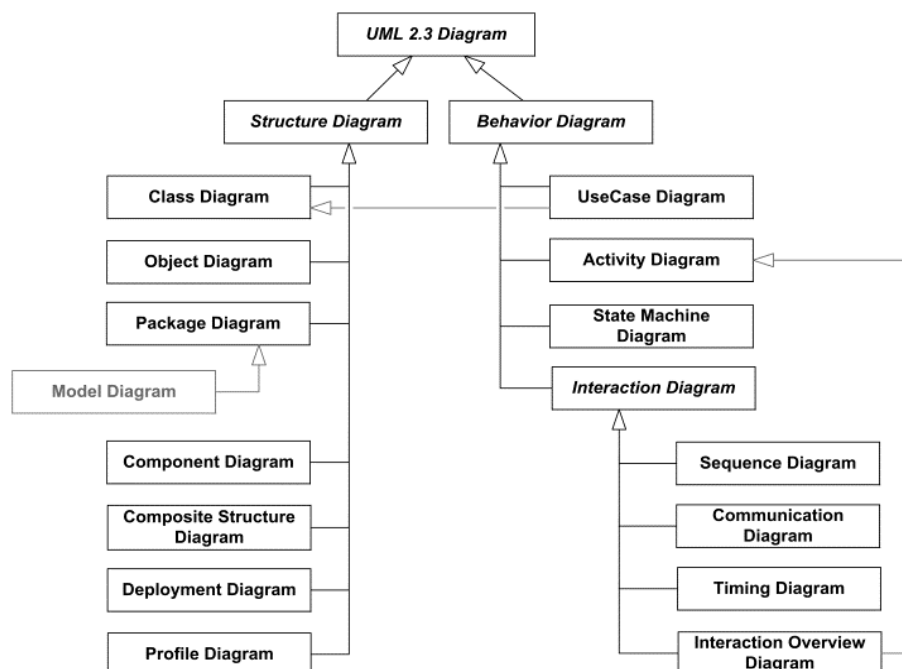
- 2) Kelola persyaratan. Secara eksplisit dokumentasikan persyaratan pelanggan dan catat perubahan pada persyaratan ini. Analisis dampak perubahan pada sistem sebelum menerimanya.
- 3) Gunakan arsitektur berbasis komponen. Struktur arsitektur sistem dikembangkan dengan mengintegrasikan komponen (koleksi objek) yang sesuai dengan standar model komponen.
- 4) Perangkat lunak model visual. Gunakan grafis model UML untuk menyajikan tampilan statis dan dinamis perangkat lunak.
- 5) Verifikasi kualitas perangkat lunak. Pastikan perangkat lunak tersebut memenuhi standar kualitas organisasi.
- 6) Mengontrol perubahan pada perangkat lunak. Mengelola perubahan pada perangkat lunak menggunakan sistem manajemen perubahan, prosedur, dan alat manajemen konfigurasi.

Inovasi yang paling penting dalam RUP adalah pemisahan fase pengembangan dan alur kerja, pengakuan bahwa menggunakan perangkat lunak dalam lingkungan pengguna adalah bagian dari proses. Fase bersifat dinamis dan memiliki tujuan. Alur kerja bersifat statis dan merupakan kegiatan teknis yang tidak terkait dengan satu fase tetapi dapat digunakan sepanjang pengembangan untuk mencapai tujuan dari setiap fase.

#### **h. *Unified Modeling Language (UML)***

Model membantu pengguna, manajer, dan profesional TI memahami desain sistem. Pemodelan melibatkan metode grafis dan bahasa nonteknis yang mewakili sistem pada berbagai tahap pengembangan. Selama *requirement modeling*, analisis

dapat menggunakan berbagai alat untuk menggambarkan proses bisnis, kebutuhan, dan interaksi pengguna dengan sistem. *Unified Modeling Language* (UML) adalah metode yang digunakan secara luas untuk memvisualisasikan dan mendokumentasikan desain sistem perangkat lunak. UML menggunakan konsep desain berorientasi objek, tetapi tidak tergantung pada bahasa pemrograman tertentu dan dapat digunakan untuk menjelaskan proses dan persyaratan bisnis secara umum (Tilley & Rosenblatt, 2017: 113). Bahasa grafis yang digunakan dalam pengembangan berorientasi objek, mencakup beberapa jenis model sistem yang memberikan pandangan yang berbeda dari suatu sistem. UML telah menjadi standar *de facto* untuk pemodelan berorientasi objek (Sommerville, 2011:121). UML adalah bahasa visual yang memakai diagram dan teks pendukung untuk pemodelan dan komunikasi suatu sistem (Rosa & Shalahudin, 2016: 137).



Gambar 9. Klasifikasi Diagram UML

Pada UML 2.3 versi terbaru terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori :

- 1) *Structure diagrams*, menunjukkan struktur statis sistem dan bagian-bagiannya pada tingkat abstraksi dan implementasi yang berbeda termasuk bagaimana mereka saling terkait. Meliputi *object diagram*, *class diagram*, *package diagram*, *component diagram*, *composite structure diagram*, *deployment diagram*.
- 2) *Behavior diagrams*, menunjukkan perilaku dinamis dari objek dalam suatu sistem yang dapat digambarkan sebagai serangkaian perubahan pada sistem dari waktu ke waktu. Meliputi *state machine diagram*, *activity diagram*, *use case diagram*.
- 3) *Interaction diagrams*, menunjukan susunan struktural objek, interaksi antar objek, dan aliran pesan dalam sistem. Meliputi *communication diagram*, *sequence diagram*, *interaction overview diagram*, *timing diagram*.

Pemodelan sistem informasi manajemen pembelajaran siswa menggunakan beberapa model diagram UML, diantaranya:

- 1) *Use Case Diagram*

Secara visual mewakili interaksi antara pengguna dan sistem informasi. Dalam diagram *use case*, pengguna menjadi aktor dengan peran spesifik yang menjelaskan bagaimana dia berinteraksi dengan sebuah sistem (Tilley & Rosenblatt, 2017: 113). *Use case* menunjukkan interaksi antara sistem dan lingkungannya yaitu eksternal aktor berupa user atau sistem lain (Sommerville, 2011: 124). Sebuah *use case* memberikan gambaran interaksi pengguna dengan

sistem, beserta langkah yang dibutuhkan untuk mencapai tujuan. *Use case diagram* menggambarkan keterkaitan dari keseluruhan *use case* sehingga nampak fungsionalitas sistem yang akan dikembangkan (Pressman, 2015: 875).

## 2) *Class Diagram*

Menurut Sugiarti (2013: 57) *class diagram* merupakan gambaran susunan sistem dari segi pendefinisian *class* (memiliki atribut dan operasi) yang akan dibuat untuk mengembangkan sistem. *Class Diagram* adalah bentuk rinci dari *use case* tunggal, menunjukan *class* yang berpartisipasi di dalam *use case* dan mendokumentasikan relasi antar *class* (Tilley & Rosenblatt, 2017: 189).

## 3) *Activity Diagram*

Diagram yang menyerupai diagram alur horizontal yang menunjukkan tindakan dan peristiwa saat terjadi. *Activity diagram* menunjukkan urutan tindakan yang dilakukan dan mengidentifikasi hasilnya (Tilley & Rosenblatt, 2017: 193). Menunjukkan aktivitas yang terlibat dalam suatu proses atau dalam pemrosesan data (Sommerville, 2011: 120). Menggambarkan perilaku dinamis suatu sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang dilakukan sistem. Pelengkap *use case* dengan menunjukan representasi visual dari aliran interaksi dalam proses tertentu (Pressman, 2015: 881).

## 4) *Sequence Diagram*

Menunjukkan waktu transaksi antar objek saat terjadi selama eksekusi sistem. Seorang analis sistem mungkin menggunakan *sequence diagram* untuk menunjukkan semua hasil yang mungkin, atau fokus pada satu skenario. Interaksi berlangsung dari atas ke bawah di sepanjang garis waktu vertikal, sementara panah

horizontal mewakili pesan dari satu objek ke objek lainnya (Tilley & Rosenblatt, 2017: 115). Diagram yang menunjukkan urutan interaksi yang diperlukan untuk menyelesaikan beberapa operasi. Dalam UML, *sequence diagram* mungkin dikaitkan dengan kasus penggunaan (Sommerville, 2011: 124). *Sequence diagram* menggambarkan komunikasi antara objek selama pengerjaan tugas secara dinamis. Menunjukkan urutan pesan yang dikirim antara objek secara temporal untuk menyelesaikan pekerjaan tertentu. Biasa dipakai untuk memvisualkan suatu skenario sistem atau satu kasus pengguna (Pressman, 2012: 876-877).

## **2. Uji Sistem**

### **a. Analisis Kualitas Perangkat Lunak**

*Software quality* adalah *software process* yang efektif digunakan melalui pembuatan produk bermanfaat, memberikan nilai terukur bagi pengembang dan penggunanya (Pressman, 2015: 414-415). Terdapat penekanan dalam 3 hal terkait *software quality*:

- 1) *Software process* yang efektif, menetapkan infrastruktur yang menunjang segala usaha pembangunan produk berkualitas tinggi.
- 2) Produk yang bermanfaat memberikan konten, fungsi, dan fitur yang diinginkan pengguna akhir, tetapi yang penting produk memberikan aset tersebut dengan cara yang handal serta bebas dari kesalahan.
- 3) Dengan menambahkan nilai bagi produsen dan pengguna produk perangkat lunak, perangkat lunak berkualitas tinggi memberikan manfaat bagi organisasi perangkat lunak dan komunitas pengguna akhir.

*Software quality* merupakan kesesuaian dengan persyaratan perangkat lunak yang ditetapkan, kemampuan produk perangkat lunak untuk memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan dalam kondisi tertentu [ISO 25010]. *Software quality* menunjukkan sejauh mana produk perangkat lunak memenuhi persyaratan yang ditetapkan, namun kualitas tergantung pada sejauh mana persyaratan yang ditetapkan secara akurat mewakili kebutuhan, keinginan, dan harapan pemangku kepentingan [IEEE 730] (Laporte & April, 2018: 19).

*Quality assurance* (QA) berperan penting dalam menjamin kualitas perangkat lunak. QA merupakan suatu proses atau prosedur untuk meminimalkan kesalahan dan memastikan kualitas dalam produk. Kualitas yang buruk dapat dihasilkan dari persyaratan yang tidak akurat, masalah desain, kesalahan pengkodean, dokumentasi yang salah, dan pengujian yang tidak efektif. Tim QA meninjau dan menguji semua perubahan aplikasi sistem untuk memverifikasi spesifikasi dan standar kualitas perangkat lunak (Tilley & Rosenblatt, 2017: 361). Keseluruhan proses mendefinisikan bagaimana kualitas perangkat lunak dapat dicapai dan bagaimana organisasi mengembangkan perangkat lunak tahu bahwa perangkat lunak telah memenuhi tingkat kualitas yang diperlukan (Sommerville, 2011: 6).

#### **b. Model Kualitas Perangkat Lunak**

Kualitas perangkat lunak perlu diuji menggunakan *software quality model* agar memenuhi standar. *Software quality model* sendiri adalah seperangkat karakteristik yang ditentukan, dan hubungan di antara mereka, yang menyediakan kerangka kerja untuk menentukan persyaratan kualitas dan mengevaluasi kualitas [ISO 25000] (Laporte & April, 2018: 67). Terdapat beberapa model kualitas

diantaranya Bertoa, Boehm, Dromey, FURPS, ISO 9126, ISO 25010, dan McCall (Miguel, Mauricio, & Rodriguez, 2014). Dari berbagai macam standar pengujian tersebut, ISO 9126 dan ISO 25010 merupakan standar internasional dalam pengujian perangkat lunak. ISO/IEC 25010 adalah bagian dari *seri Standar Internasional SquaRE* (persyaratan dan evaluasi kualitas sistem dan perangkat lunak) menggantikan ISO/IEC 9126-1: 2001, yang telah direvisi secara teknis. Maka dari itu, peneliti menggunakan standar pengujian model ISO/IEC 25010.

Kualitas model dalam standar internasional ini dapat digunakan untuk mengidentifikasi karakteristik kualitas yang relevan, lebih jauh digunakan untuk menetapkan persyaratan, kriteria untuk memenuhi kepuasan dan tindakan yang sesuai. Standar internasional ini mendefinisikan:

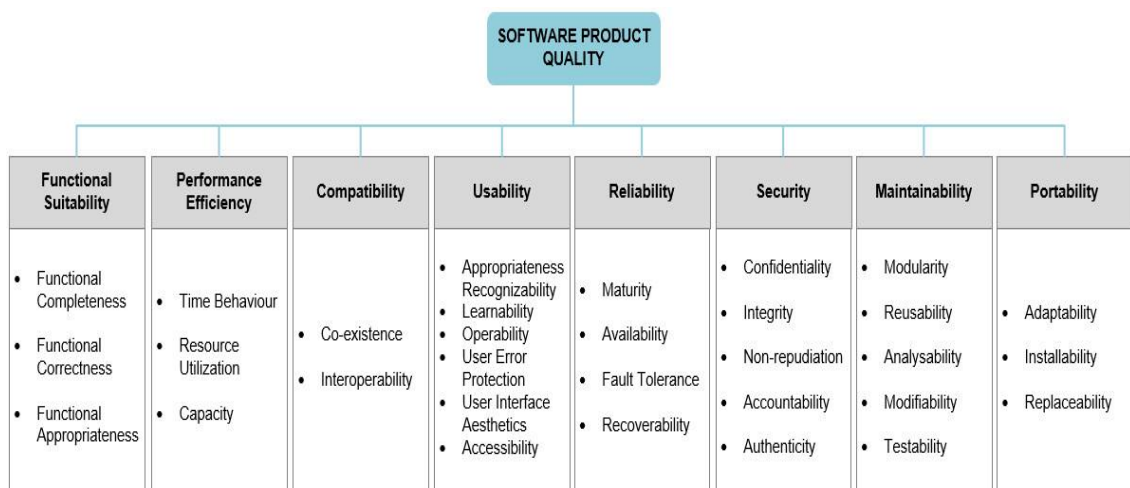
- 1) Model kualitas yang digunakan terdiri dari lima karakteristik (beberapa diantaranya dibagi lagi menjadi subkarakteristik), berkaitan dengan hasil interaksi ketika suatu produk digunakan dalam konteks penggunaan tertentu. Model sistem ini berlaku untuk sistem komputer manusia yang lengkap, termasuk sistem komputer dan produk perangkat lunak yang digunakan.
- 2) Model kualitas produk yang terdiri dari delapan karakteristik (yang kemudian dibagi lagi menjadi subkarakteristik) terkait dengan sifat dinamis sistem komputer dan sifat statis perangkat lunak sehingga berlaku untuk sistem komputer dan produk perangkat lunak.

Kedua model tersebut mempunyai karakteristik yang relevan untuk seluruh sistem komputer dan produk perangkat lunak. Mereka juga menyediakan serangkaian karakteristik kualitas yang dapat dibandingkan dengan persyaratan kualitas yang



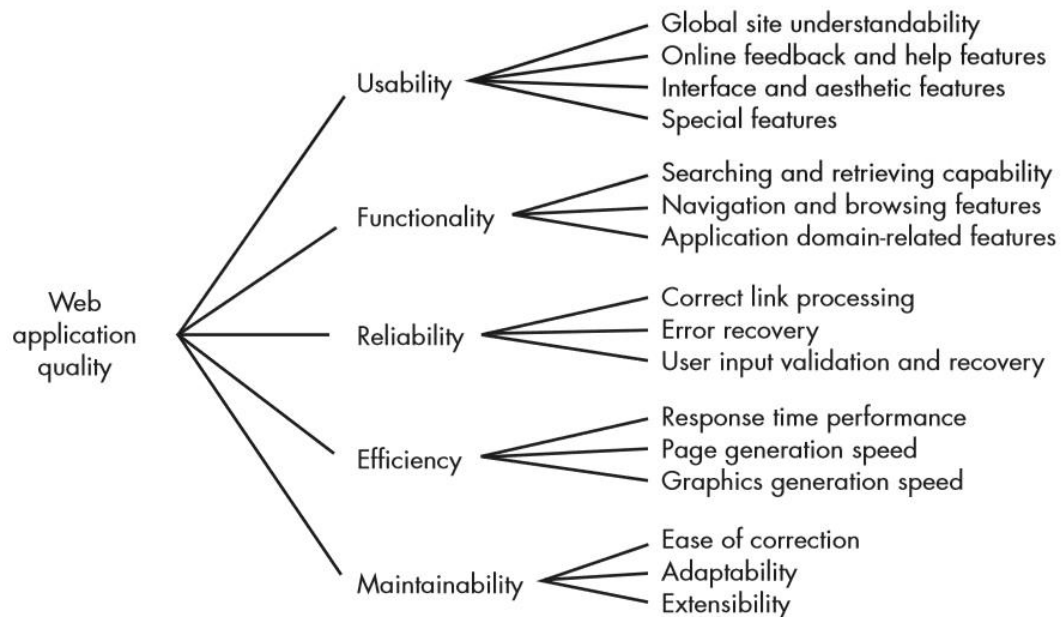
dinyatakan untuk kelengkapan. Karakteristik dan subkarakteristik menyediakan terminologi yang konsisten untuk menentukan, mengukur, dan mengevaluasi kualitas produk sistem dan perangkat lunak.

*Model quality product ISO 25010* dikategorikan menjadi 8 karakteristik, diantaranya *functional suitability*, *performance efficiency*, *compatibility*, *usability*, *reliability*, *portability*, *maintainability* dan *security*. Setiap karakteristik terdiri dari seperangkat subkarakteristik terkait.



Gambar 10. *Software Product Quality ISO/IEC 25010*

Penilaian standar kualitas aplikasi *web* meliputi lima aspek yaitu kemudahan pengguna, fungsionalitas, keandalan, efisiensi dan pemeliharaan (Olsina, Godoy, Lafuente, & Rosi, 1999). Gambar 11 menunjukkan rincian dari kelima aspek tersebut. Mengacu pada standar kualitas web menurut Olsina, dari 8 karakteristik ISO/IEC 25010 peneliti hanya akan mengambil 5. Kelima karakteristik yang dipakai untuk melakukan pengujian sistem informasi manajemen pembelajaran tersebut adalah *usability*, *functional suitability*, *reliability*, *performance efficiency*, dan *maintainability*.



Gambar 11. Standar Kualitas *Web* Menurut Olsina

1) *Functional suitability*

Karakteristik sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan ketika digunakan dalam kondisi tertentu. Karakteristik ini terbagi menjadi tiga subkarakteristik:

- a) *Functional completeness*, sejauh mana serangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.
- b) *Functional correctness*, sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan.
- c) *Functional appropriateness*, sejauh mana fungsi memfasilitasi penyelesaian tugas dan tujuan yang ditentukan.

Metode *black-box testing* (pengujian perilaku) digunakan untuk menguji *functional suitability*. Persyaratan fungsionalitas perangkat lunak menjadi fokus dalam pengujian. (Pressman, 2015: 509). Teknik pengujian melalui *black-box*

memungkinkan untuk memperoleh serangkaian kondisi input yang sepenuhnya akan menjalankan semua persyaratan fungsional untuk suatu program. Alasan penggunaan metode ini karena beberapa kategori kesalahan bisa ditemukan, diantaranya: (1) kesalahan/hilangnya fungsi, (2) kesalahan antarmuka, (3) kesalahan struktur data atau akses *database* eksternal, (4) kesalahan perilaku, (5) kesalahan instalasi dan terminasi.

## 2) *Usability*

Karakteristik sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan spesifik dengan efektivitas, efisiensi dan kepuasan dalam konteks penggunaan tertentu. Karakteristik ini mempunyai tiga subkarakteristik :

- a) *Appropriateness Recognizability*, sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.
- b) *Learnability*, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran yang ditentukan, menggunakan produk atau sistem dengan efektivitas, efisiensi, kebebasan dari risiko dan kepuasan dalam konteks penggunaan tertentu.
- c) *Operability*, sejauh mana suatu produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan.
- d) *User Error Protection*, sejauh mana suatu sistem melindungi pengguna dari membuat kesalahan.
- e) *User Interface Aesthetics*, sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.

- f) *Accessibility*, sejauh mana suatu produk atau sistem dapat digunakan oleh orang-orang dengan jangkauan karakteristik dan kemampuan terluas untuk mencapai tujuan tertentu dalam konteks penggunaan tertentu.

Menurut Sauro dan Lewis (2012) *post study questionnaires* terstandarisasi yang paling banyak digunakan untuk melakukan penilaian persepsi *usability* pada akhir studi (setelah menyelesaikan serangkaian skenario uji) dan yang dikutip dalam standar nasional dan internasional diantaranya: PSSUQ (*Post-study System Usability Questionnaire*), QUIS (*Questionnaire for User Interaction Satisfaction*), SUMI (*Software Usability Measurement Inventory*), SUS (*Software Usability Scale*), UMUX (*Usability Metric for User Experience*), USE (*Usefulness, Satisfaction, and Ease of Use Questionnaire*).

Penelitian ini akan menggunakan USE *Questionnaire* Arnold M. Lund untuk menguji karakteristik *usability* sistem. Kuesioner terdiri dari 30 skala penilaian terbagi kedalam empat kategori: *usefulness*, *satisfaction*, *ease of use*, dan *ease of learning*. Masing-masing adalah pernyataan positif dimana pengguna menilai dengan memberikan tingkat persetujuan pada skala likert tujuh poin (Tullis & Albert, 2013: 142). Berdasarkan beberapa hasil penelitian sebelumnya, pengguna melakukan evaluasi produk terutama pada dimensi *usefulness*, *satisfaction*, dan *ease of use*. Korelasi yang dihitung menggunakan skala dari dimensi tersebut menunjukkan bahwa *ease of use* dan *usefulness* saling mempengaruhi. Peningkatan parameter *ease of use* akan diikuti peningkatan pada *usefulness*, begitu sebaliknya. Sementara keduanya memberi pengaruh besar terhadap *satisfaction* (Rahadi, 2014). Jika sistem adalah sistem internal yang harus digunakan, maka faktor *usefulness*

relatif kurang penting. Item yang berperan terhadap parameter *ease of use* dalam sistem internal terbagi menjadi *ease of learning* dan *ease of use* (Lund, 2001).

### 3) *Performance Efficiency*

Karakteristik sejauh mana tingkat kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi lain. Karakteristik ini terbagi menjadi beberapa subkarakteristik :

- a) *Time Behaviour*, sejauh mana respon dan waktu pemrosesan serta tingkat *throughput* suatu produk atau sistem, saat menjalankan fungsinya, memenuhi persyaratan.
- b) *Resource Utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, saat menjalankan fungsinya, memenuhi persyaratan.
- c) *Capacity*, sejauh mana batas maksimum suatu produk atau parameter sistem memenuhi persyaratan.

*Load testing* bisa digunakan untuk menguji *Performance efficiency* (Janani & Krishnamoorthy, 2015). Metode ini digunakan untuk menentukan respon aplikasi *web* dan ruang lingkup server melalui beragam kondisi pemuatan (Pressman, 2015: 562). Pengujian *load testing* bisa dilakukan menggunakan salah satu *software* yaitu GTMetrix (AlBalushi, Ali, Ashrafi, & AlBalushi, 2016). GTMetrix adalah perangkat lunak *freeware* untuk melakukan analisis performa *website*. Halaman laporan GTmetrix dengan rapi merangkum kinerja halaman *web* berdasarkan indikator kunci dari *page load speed*.

- a) Analisis halaman dengan aturan Google PageSpeed dan Yahoo! YSlow.

- b) Tinjauan terhadap *page load time*, *total page size* dan *total of request*.
- c) Melihat kinerja halaman *web* relatif terhadap rata-rata semua situs yang dianalisis pada GTmetrix.
- 4) *Reliability*

Karakteristik sejauh mana suatu sistem, produk atau komponen melakukan fungsi tertentu dalam kondisi tertentu untuk jangka waktu tertentu. Karakteristik ini memiliki beberapa subkarakteristik :

- a) *Maturity*, sejauh mana suatu sistem, produk atau komponen memenuhi kebutuhan untuk keandalan dalam operasi normal.
- b) *Availability*, sejauh mana sistem, produk atau komponen beroperasi dan dapat diakses bila diperlukan untuk digunakan.
- c) *Fault Tolerance*, sejauh mana sistem, produk atau komponen beroperasi sebagaimana dimaksud meskipun ada kesalahan perangkat keras atau perangkat lunak.
- d) *Recoverability*, sejauh mana, dalam hal terjadi gangguan atau kegagalan, suatu produk atau sistem dapat memulihkan data yang terkena dampak langsung dan membangun kembali keadaan sistem yang diinginkan.

Aspek yang mempunyai pengaruh dan sebaiknya dilakukan dalam pengujian *reliability* adalah aspek *maturity* (Losavio, Chirinos, Levy, & Ramdane-Cherif, 2003). Metode *stress testing* bisa digunakan dalam uji *reliability* dengan skenario pengujian *user* melakukan akses halaman *web* secara bersamaan dalam jangka waktu tertentu. Perhitungan *reliability* dilakukan dengan membandingkan jumlah fungsi yang berjalan serta jumlah kegagalan yang dieksekusi. Menurut Pradhan

(2013) *stress testing* bisa dilakukan dengan *Software Web Application Load, Stress, and Performance Testing* (WAPT). WAPT adalah *software* analisis kinerja *web* yang menyediakan *load, stress*, dan *performance testing* sehingga memudahkan pengembang untuk melakukan optimasi (Softlogica, 2019). WAPT mensimulasikan pengguna virtual yang akan mengulangi skenario pengujian URL yang telah direkam serta menentukan berapa kali iterasi pengguna virtual harus mengulangi skenario tersebut. Dengan demikian, alat ini berguna untuk memeriksa *bottleneck* (kemacetan) dan *performance leakage* (kebocoran kinerja) di situs *web* atau aplikasi *web* yang sedang diuji (Kundu, 2012).

#### 5) *Maintainability*

Karakteristik sejauh mana tingkat efektivitas dan efisiensi yang dengannya suatu produk atau sistem dapat dimodifikasi oleh pengelola. Karakteristik ini mempunyai beberapa subkarakteristik:

- a) *Modularity*, sejauh mana perubahan pada satu komponen sistem memiliki dampak minimal pada komponen lainnya.
- b) *Reusability*, sejauh mana suatu komponen bisa diimplementasikan kembali pada sistem yang berbeda.
- c) *Analysability*, tingkat keefektifan dan efisiensi sejauh mana menilai dampak dari perubahan yang dilakukan, atau mendiagnosis kekurangan atau penyebab kegagalan, atau mengidentifikasi bagian yang akan dimodifikasi.
- d) *Modifiability*, sejauh mana suatu produk atau sistem dapat secara efektif dan efisien dimodifikasi tanpa menimbulkan kerusakan/menurunkan kualitas.

- e) *Testability*, tingkat efektivitas dan efisiensi yang dengannya kriteria pengujian dapat ditetapkan dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah dipenuhi oleh sistem, produk atau komponen.

Metrik perangkat lunak yang paling banyak digunakan dalam mengkuantifikasi *maintainability* suatu sistem dikenal sebagai *Maintainability Index* (MI). MI mengukur seberapa dapat dikelola (mudah dimodifikasi dan dilakukan perbaikan) *source code* dari suatu sistem (Najm, 2014). MI merupakan kombinasi *software metrics* bernama *McCabe's Cyclomatic Complexity* (CC), *Halstead's Volume* (V), dan *Lines of Code* (LOC) (Ganpati, Kalia, & Singh, 2012). PHPmetrics adalah tools bisa digunakan untuk membantu menghitung dan menganalisis *Maintainability Index* suatu sistem berbasis bahasa pemrograman PHP (Lepine, 2015). Berikut formula pada PHPMetrics untuk menghitung MI:

$$MI = 171 - 5.2 * \log_2(V) - 0.23 * CC - 16.2 * \log_2(LOC) + 50 * \sin(\sqrt{2.4 * CM})$$

Keterangan:

MI = *Maintainability Index*

V = *Halstead's Volume*

CC = *Cyclomatic Complexity*

LOC = *Lines Of Code*

CM = *Lines Of Comment*



## B. Penelitian yang Relevan

Beberapa penelitian yang sesuai dan relevan dengan penelitian ini diantaranya:

1. Penelitian dengan judul “Aplikasi Monitoring Proses Belajar Mengajar Berbasis Web di SMK Telkom Bandung” oleh Akbar Muhammad Arif, Elis Hernawati, dan Ferra Arik Tridalestari dalam naskah publikasi tahun 2017. Tujuan pengembangan aplikasi ini untuk membantu pihak sekolah memantau guru agar dapat mempermudah aktivitas mengajar. Merupakan aplikasi berbasis *web* yang digunakan untuk mempermudah staf kurikulum dalam memantau kinerja guru, memantau perkembangan materi, memantau proses mengajar dan evaluasi perkembangan siswa. Fitur untuk guru meliputi absensi guru, input data kegiatan mengajar, *upload* perangkat pembelajaran, dan kelola penilaian siswa. Fitur untuk staf kurikulum meliputi melihat aktivitas kehadiran guru, mereview perangkat pembelajaran, kelola jadwal mengajar, kelola penilaian guru, dan melihat semua laporan guru. Aplikasi belum dilakukan pengujian menggunakan model kualitas perangkat lunak, baru sebatas pada fungsionalitas sistem.
2. Konsep *Virtual Classroom* Pada Sistem Pembelajaran Jarak Jauh (Studi Kasus : AMIK Lembah Dempo Pagaralam) oleh Mohamad Farozi dalam naskah publikasi tahun 2016. Tujuan penelitian ini adalah mengembangkan sistem pengelolaan pembelajaran kelas berbasis *website*. Fungsi-fungsi yang disediakan sistem ini terbagi untuk tiga user yaitu akademik, guru, siswa. Fungsi untuk siswa: membaca materi kuliah, mengajukan pertanyaan,

menjawab soal dan tugas, mengikuti diskusi terbuka. Fungsi untuk guru: Menyampaikan materi, memberikan pertanyaan, memberikan penugasan, mengajukan diskusi terbuka. Fungsi untuk akademik: Monitoring materi kuliah, jadwal dosen dan mahasiswa, hasil evaluasi tugas, dan soal. Rancang bangun sistem *virtual classroom* ini belum dilakukan pengujian atau evaluasi. Kendala dari desain antarmuka masih sangat sederhana dan pengaturan *layout web* kurang rapi sehingga terkesan kurang *user friendly*.

3. Penelitian dengan judul “Sistem Informasi Monitoring Studi Siswa SMA Berbasis Kurikulum Nasional 2013” oleh Deni Multazam dan Agus Prasetyo utomo dalam naskah publikasi tahun 2014. Tujuan perancangan aplikasi ini adalah sebagai media penyampaian informasi, serta pengolahan data penilaian berdasarkan kurikulum 2013 sehingga membantu dan memudahkan pihak sekolah dalam mengambil keputusan terkait hasil studi siswa di sekolah. Merupakan aplikasi berbasis *web* yang digunakan oleh guru, wali kelas, wakil kepala sekolah bidang akademik, dan kepala sekolah. Sistem ini belum dilakukan uji coba pada lingkungan operasional sekolah. Tidak dapat melakukan penilaian secara harian atau setiap pertemuan guru mengajar siswa. Melainkan penilaian hasil setiap akhir semester dimana hanya data nilai akhir yang dimasukkan kedalam sistem.

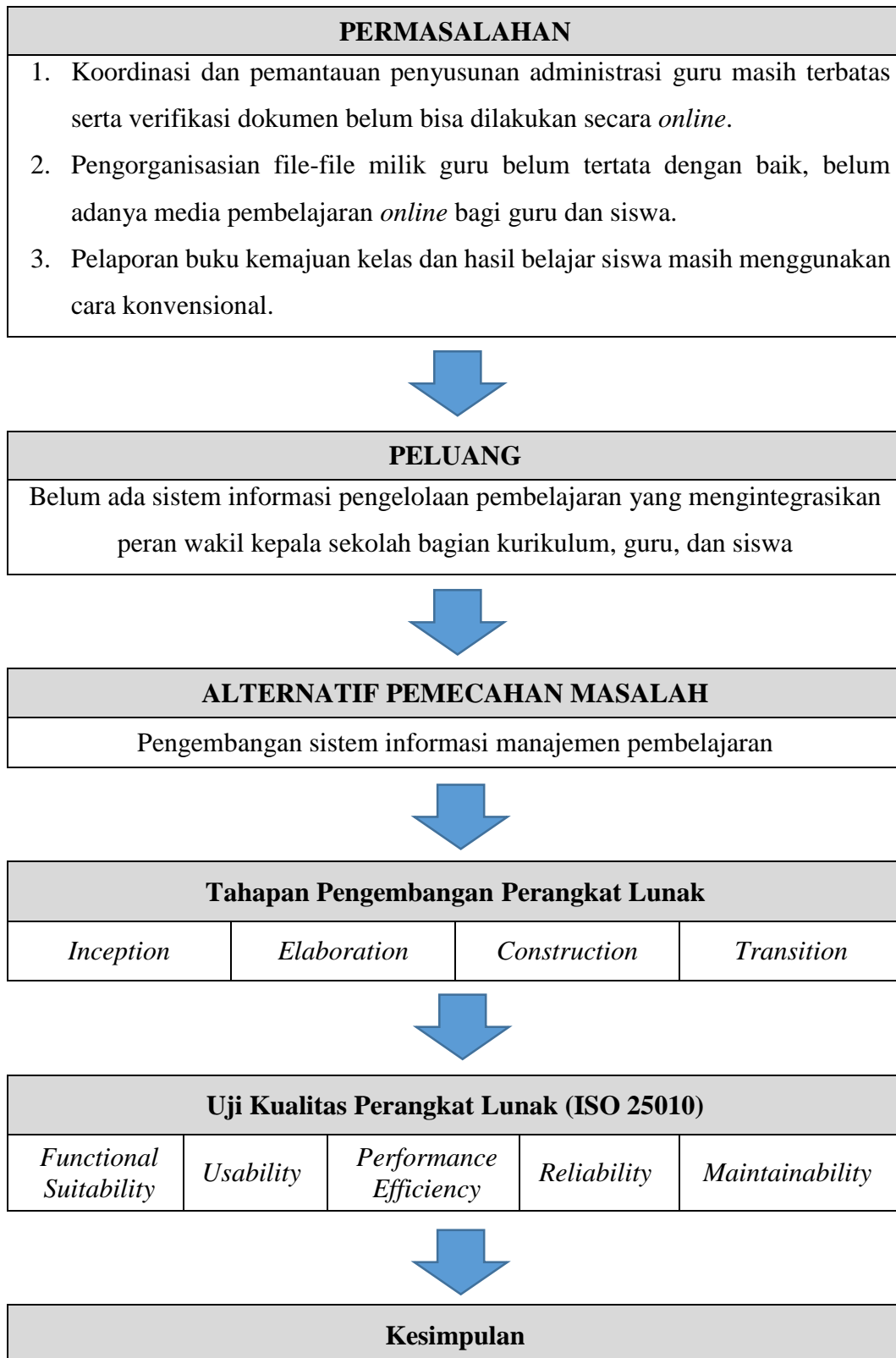
Berdasarkan ketiga penelitian yang relevan tersebut, terdapat dua aspek yang dapat dikaji yaitu aspek pengembangan perangkat lunak dan pengujian perangkat lunak. Dari aspek pengembangan perangkat lunak, ketiga produk yang dihasilkan

hanya fokus pada salah satu aktivitas manajemen yaitu pelaksanaan dan pengawasan pembelajaran.

1. Penelitian pertama, lebih fokus pada menyelesaikan permasalahan administrasi proses belajar mengajar antara staf kurikulum dan guru. Belum ada fitur untuk membantu dan mendukung mengelola aktivitas pembelajaran kelas yang melibatkan peran siswa.
2. Penelitian kedua, lebih fokus pada pengembangan sistem untuk mendukung pengelolaan aktivitas pembelajaran guru dan siswa di kelas. Belum ada fitur untuk membantu menyelesaikan keperluan administrasi pengajaran.
3. Penelitian ketiga, sebatas pada pengelolaan administrasi nilai akhir berdasarkan standar penilaian yang dipakai sekolah terkait.

Dari aspek pengujian perangkat lunak, ketiga produk yang dihasilkan penelitian diatas belum dilakukan uji kualitas. Pengujian sebatas pada fungsionalitas sistem yaitu memastikan agar fitur yang dibuat berjalan dengan benar. Berbeda dari ketiga penelitian tersebut, produk yang dikembangkan dari penelitian ini bisa digunakan untuk membantu menyelesaikan permasalahan administrasi pembelajaran dan mendukung pengelolaan aktivitas belajar kelas yang melibatkan staf kurikulum, guru, dan siswa. Pengintegrasian sistem tidak sebatas pada salah satu aktivitas manajemen, melainkan keseluruhan aktivitas manajemen mulai dari tahap perencanaan, pengorganisasian, pelaksanaan dan pengawasan, serta evaluasi. Tentunya semua itu berdasarkan lingkup manajemen yang berlaku di sekolah dalam kasus ini SMK Negeri 2 Wonosari. Uji sistem untuk menjamin kualitas produk pada penelitian ini menggunakan *software quality model* ISO 25010.

### C. Kerangka Berpikir



#### **D. Pertanyaan Penelitian**

Berikut pertanyaan penelitian dari pengembangan Sistem Informasi Manajemen Pembelajaran Siswa:

1. Bagaimana mengembangkan sistem informasi manajemen untuk mengatasi permasalahan perencanaan, pengorganisasian, pelaksanaan dan pengawasan, serta evaluasi pembelajaran di SMK Negeri 2 Wonosari?
2. Apakah sistem informasi manajemen pembelajaran siswa memenuhi aspek *functional suitability*?
3. Apakah sistem informasi manajemen pembelajaran siswa memenuhi aspek *usability*?
4. Apakah sistem informasi manajemen pembelajaran siswa memenuhi aspek *reliability*?
5. Apakah sistem informasi manajemen pembelajaran siswa memenuhi aspek *performance efficiency*?
6. Apakah sistem informasi manajemen pembelajaran siswa memenuhi aspek *maintainability*?