BAB II
TINJAUAN PUSTAKA

2.1. Studi Literatur

2.2.1. Penelitian Terdahulu

Penelitian berjudul "Rancangan Sistem Informasi Pengelolaan Audit Mutu Internal (AMI) Perguruan Tinggi" yang dilakukan Eva Faja Ripanti dan H. A. Oramahi. Penelitian tersebut merupakan studi literatur yang mengkaji tentang perancangan sistem informasi AMI. Pada penelitian ini diterapkan di Universitas Tanjungpura sebagai peningkatkan efisiensi kegiatan AMI. Alasan lain adalah banyaknya program studi yang dimiliki Untan dan auditor-auditor yang belum cukup sehingga diperlukan perancangan sistem. Dalam pengembangannya dilakukan menggunakan pendekatan SDLC. Sedangkan setiap tahapannya menggunakan instrumen analisis dan perancangan yaitu Analisis Berorientasi Objek dan Desain (OOAD) dan hasilnya divisualisasikan menggunakan UML. Kesimpulan dari penelitian ini dinyatakan bahwa sistem ini dapat berfungsi dengan baik. Namun diperlukan improvisasi dari segi fitur sehingga aplikasi menjadi semakin memudahkan kegiatan AMI [6].

Penelitian berjudul "Rancang Bangun Sistem Monitoring Pelaksanaan Dan Hasil Audit Mutu Internal" yang dilakukan I Putu Adi Pratama dan I Gede Totok Suryawan. Penelitian ini merupakan studi literatur yang bermaksud untuk merancang sebuah sistem informasi sebagai pemanfataan dalam kegiatan AMI di STMIK STIKOM Indonesia. Pada penelitian ini diterapkan dengan maksud untuk meningkatkan kemudahan dan efektivitas pelaksanaan kegiatan AMI. Hal ini dikarenakan proses pelaksanaannya masih dilakukan dengan aplikasi spreadsheet yang memiliki beberapa keterbatasan seperti tidak bisa melakukan monitoring secara rutin. Selain itu, riwayat dan capaian laporan juga tidak terpantau dengan baik. Dengan pemanfaatan teknologi berbasis *web* diharapkan dapat mengatasi

permasalahan tersebut. Kesimpulan yang diperoleh adalah sistem informasi ini sudah dapat memenuhi kebutuhan kegiatan AMI [7].

Penelitian berjudul "Analisis dan Perancangan Sistem Informasi Audit Mutu Internal Pada STMIK GI MDP" yang dilakukan Desi Pibriana dan Desy Iba Ricoida. Penelitian ini melakukan analisis dan perancangan sistem informasi AMI di STMIK GI MDP. Penelitian ini dilaksanakan untuk mengatasi permasalahan-permasalahan yang dihadapi oleh kepala Badan Penjaminan Mutu (BPM) dalam kegiatan AMI seperti dokumen-dokumen dari unit tertentu sulit untuk didistribusikan kepada auditor sebagai bahan untuk mengaudit, waktu untuk rekap dan analisis hasil kegiatan audit cukup lama dan proses monitoring hasil audit yang tidak efektif. Penelitian ini menggunakan metode FAST (*Framework for the Application of System Thinking*) dengan pendekatan *Model Driven Development Strategy*. Perolehan kesimpulan yang didapat adalah sistem ini dapat mengatur hak akses dan membantu dalam proses pengelolaan dokumentasi dalam kegiatan AMI sehingga Badan Penjaminan Mutu (BPM) dapat melaksanakan kegiatan secara efisien dan efektif [8].

Penelitian berjudul "Sistem Informasi Audit Mutu Internal (SIAMI)" yang dilakukan Andie, Muhammad Hasbi dan Hasanuddin. Penelitian ini bertujuan dalam meningkatkan keefektifan dan kemudahan Universitas Islam Kalimantan Muhammad Arsyad Al Banjarmasin dalam melaksanakan kegiatan AMI. Pada tahun 2019, pelaksanaan AMI sudah memasuki siklus keempat. Namun proses AMI di UNISKA ini dilaksanakan secara manual dan *paperless*. Dalam perjalanannya, UNISKA memerlukan waktu yang lama dan tidak efektif bagi auditor. Selain itu, meskipun telah disediakan *template* laporan audit, konsistensi dalam menuliskan format laporan audit tidak terkontrol. Oleh karena demikian, Andie dkk. (peneliti) melakukan perancangan sistem secara otomatis untuk memudahkan proses AMI. Metode *waterfall* merupakan model SDLC yang digunakan oleh peneliti. Metode ini memungkinkan departementalisasi dan kontrol. Hasil yang dicapai dalam penelitian ini adalah implementasi sistem terhadap pengguna memiliki respon

yang sangat baik dan aplikasi berjalan dengan lancer dan sesuai prosedur pelaksanaan AMI [9].

Penelitian berjudul “Implementasi Model *Scrum* pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang” yang dilakukan Usman Ependi. Penelitian ini bertujuan sebagai pemenuhan target porsi jumlah mahasiswa baru yang diterima. Sebelumnya itu proses Seleksi Masuk Mahasiswa saat itu masih sangat berketergantungan pada sistem Seleksi Bersama Masuk Sekolah Tinggi, Akademi dan Politeknik Pariwisata (SBMSTAPP) yang dilakukan secara serentak oleh beberapa perguruan tinggi pariwisata yang dibawahi kepada Kementerian Pariwisata. Permasalahannya yang timbul adalah ketika target jumlah mahasiswa baru pada perguruan tinggi tersebut tidak tercapai, maka muncul dampak yang dialami perguruan tinggi tersebut. Oleh karena itu, penelitian ini dilaksanakan untuk merancang sistem informasi SSM khusus untuk Politeknik Pariwisata Palembang. Dalam pengembangannya, model *scrum*-lah yang digunakan oleh peneliti. Sistem informasi Seleksi Masuk Mahasiswa (SMM) yang dihasilkan dapat mengatasi permasalahan yang dialami oleh Politeknik Pariwisata Palembang. Selain itu, hasil pengujian sistem dapat berjalan sesuai fungsinya [10].

Penelitian berjudul “Implementasi *Scrum* dalam Pengembangan Sistem Informasi Jasa Desain Grafis” yang dilakukan Lalu Mutawali, Buyung Kurnia Fathoni dan Hasyim Asyari. Penelitian ini bertujuan dalam memberikan kemudahan bagi pemilik UMKM untuk mendapatkan informasi jasa layanan desain. Permasalahan yang ada yaitu pelaku UMKM merasa kesulitan dalam memeroleh informasi terkait layanan desain produk. Oleh karena itu, peneliti melakukan perancangan sistem informasi layanan jasa desain grafis. Dalam pengembangan sistem ini menggunakan *scrum*. *Scrum* dapat mengatasi pengembangan perangkat lunak yang rumit. Pencapaian penelitian ini adalah pengembangan sistem ini telah berhasil melakukan pemesanan jasa desain dengan baik [11].

Penelitian berjudul “Implementasi *Scrum Model* dalam Pengembangan Aplikasi Pelaporan Sampah sebagai Wujud *Smart Cleaning*” yang dilakukan Hutrianto dan Ade Putra. Penelitian ini bertujuan untuk memudahkan masyarakat dalam melakukan pelaporan terhadap penumpukan sampah yang belum diambil oleh petugas kebersihan. Permasalahan yang dihadapi masih terdapat sampah yang terlewati oleh petugas, baik karena pengangkutan yang terlewati maupun karena keterbatasan penyimpanan truk. Oleh karena itu, sang peneliti perlu mengembangkan sistem pelaporan sampah oleh warga kepada pihak yang bertanggung jawab. Model pengembangan sistem yang digunakan menggunakan model *scrum*. Hasil yang dicapai dalam penelitian ini adalah implementasi *scrum* memiliki hasil yang baik dalam pengembangan sistem. Aplikasi memiliki tiga jenis hak akses pengguna sehingga sistem pelaporan berjalan dengan baik. Selain itu, aplikasi ini juga dikembangkan untuk dua buah *platform* yaitu *website* untuk admin dan *mobile* untuk masyarakat [12].

Penelitian berjudul “*Several Ideas on Integration of SCRUM Practices within Microsoft Solutions Framework*” yang dilakukan Jai Vigneshwar Alavandhar dan Oksana Nikiforova. Penelitian ini bertujuan untuk melakukan perbandingan dua metode dalam pengembangan perangkat lunak. Dengan mengikuti metode pengembangan sistem yang telah terkenal dan diakui dapat membuahkan hasil yang memuaskan. Namun permasalahannya, ketika terdapat metode pengembangan yang baru, membuat organisasi bertanya-tanya terhadap tingkat keberhasilan yang didapatkan. Oleh sebab itu, sang peneliti perlu mempelajari dan menguji kedua metode pengembangan sistem ini menggunakan analisis komparatif. Kedua metode ini adalah *Microsoft Solutions Framework* dan *Scrum*. Perolehan hasil yang dicapai dalam penelitian ini menunjukan bahwa metode *scrum* lebih mengedepankan tahapan-tahapan dalam pengembangan sistem sedangkan *Microsoft Solutions Framework* berfokus pada solusi yang dihasilkan terhadap pengembangan sistem. Namun keduanya memiliki keunggalan masing-masing dan kesamaan prinsip [13].

Tabel 2.1. Perbandingan penelitian-penelitian yang terkait

No.	Judul Penelitian	Penulis	Metode Penelitian	Hasil
1.	Rancangan Sistem Informasi Pengelolaan Audit Mutu Internal (AMI) Perguruan Tinggi [6]	Eva Faja Ripanti dan H. A. Oramahi	*System Development Life Cycle* (SDLC)	Hasil pengembangan sistem AMI berjalan dengan baik dan sesuai dengan prosedur. Namun diperlukan improvisasi dari segi fitur sehingga aplikasi menjadi semakin memudahkan kegiatan AMI.
2.	Rancang Bangun Sistem Monitoring Pelaksanaan Dan Hasil Audit Mutu Internal [7]	I Putu Adi Pratama dan I Gede Totok Suryawan	-	Sistem informasi yang dirancang sudah dapat memenuhi kebutuhan kegiatan AMI.
3.	Analisis dan Perancangan Sistem Informasi Audit Mutu Internal Pada STMIK GI MDP [8]	Desi Pibriana dan Desy Iba Ricoida	FAST (*Framework for the Aplication of System Thinking*) dengan pendekatan *Model Driven Development Strategy*	Sistem informasi ini dapat mengatur hak akses dan membantu dalam proses pengelolaan dokumentasi dalam kegiatan AMI (AMI) sehingga Badan Penjaminan Mutu (BPM) dapat melaksanakan kegiatan secara efisien dan efektif.
4.	Sistem Informasi Audit Mutu Internal (SIAMI) [9]	Andie, Muhammad Hasbi dan Hasanuddin	*System Development Life Cycle* (SDLC) dengan pendekatan metode *Waterfall*	Implementasi sistem terhadap pengguna memiliki respon yang sangat baik dan aplikasi berjalan dengan lancer dan sesuai prosedur pelaksanaan AMI
5.	Implementasi Model *Scrum* pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang [10]	Usman Ependi	*Agile Development* dengan menggunakan pendekatan *scrum model*	Pengembangan sistem yang memerlukan kecepatan dan ketepatan sangat cocok jika menerapkan s*crum* sebagai model pengembangan sistem. Sistem informasi Seleksi Masuk Mahasiswa dapat memenuhi kebutuhan sesuai perancangan. Selain itu, hasil pengujian sistem dapat berjalan sesuai fungsinya.

No.	Judul Penelitian	Penulis	Metode Penelitian	Hasil
6.	Implementasi *Scrum* dalam Pengembangan Sistem Informasi Jasa Desain Grafis [11]	Lalu Mutawali, Buyung Kurnia Fathoni dan Hasyim Asyari	*Agile Methodology* dengan menggunakan pendekatan *scrum model*	Sistem informasi jasa layanan desain grafis yang dikembangkan telah sukses sebagaimana fungsinya seperti melakukan pemesanan jasa desain.
7.	Implementasi *Scrum Model* dalam Pengembangan Aplikasi Pelaporan Sampah sebagai Wujud *Smart Cleaning* [12]	Hutrianto dan Ade Putra	*Agile Methodology* dengan menggunakan pendekatan *scrum model*	Implementasi *scrum* membuahkan hasil yang baik dalam pengembangan sistem pelaporan sampah di Kota Palembang. Aplikasi memiliki tiga jenis hak akses pengguna sehingga sistem pelaporan berjalan dengan baik. Selain itu, aplikasi ini juga dikembangkan untuk dua buah *platform* yaitu *website* untuk admin dan *mobile* untuk masyarakat.
8.	*Several Ideas on Integration of SCRUM Practices within Microsoft Solutions Framework* [13]	Jai Vigneshwar Alavandhar dan Oksana Nikiforova	Analisis Komparatif	Metode *scrum* lebih mengedepankan tahapan-tahapan dalam pengembangan sistem sedangkan *Microsoft Solutions Framework* berfokus pada solusi yang dihasilkan terhadap pengembangan sistem. Namun keduanya memiliki keunggalan masing-masing dan kesamaan prinsip.

2.2.2. Audit Mutu Internal

Audit Mutu Internal (AMI) adalah kegiatan yang bersifat independen dan objektif guna memberi keyakinan dan konsultasi untuk memberi nilai tambah dan peningkatan operasional sebuah organisasi [1]. Menurut [4] AMI merupakan serangkaian kegiatan terstruktur dan mandiri untuk melakukan evaluasi perolehan hasil temuan audit sehingga dapat terukur sejauh mana kriteria audit telah terpenuhi berdasarkan standar mutu Pendidikan Tinggi dan Perguruan Tinggi.

Dalam kegiatan AMI memiliki beberapa tahapan yang berkelanjutan. Tahapan-tahapan tersebut dapat dilihat pada **Tabel 2.2** berikut ini.

Tabel 2.2. Siklus tahapan dari kegiatan AMI di ITTP [1]

Tahap	Aktivitas	Pelaksana
1	Menetapkan tujuan Audit	Pimpinan Institusi
2	Merencanakan kegiatan audit	Pimpinan Institusi
3	Menetapkan target dan lingkup audit	Pimpinan Institusi
4	Membentuk kelompok audit	Tim SAI
5	Menelaah kembali dokumen audit dan menyiapkan daftar instrument audit	Tim Audit
6	Menetapkan jadwal audit	Tim Audit
7	Memberikan informasi kepada tim audit tentang kegiatan audit	Tim Audit
8	Melaksanakan kegiatan audit	Tim Audit
9	Menyusun laporan audit	Tim Audit
10	Diskusi tinjauan manajemen	Manajemen
11	Penutupan kegiatan audit	Manajemen

2.2.3. Pengembangan Sistem Informasi

Dalam penelitian ini, pengembangan sistem informasi menggunakan pendekatan *System Development Life Cycle* (SDLC). Menurut [14], SDLC adalah metodologi pengembangan perangkat lunak atau sistem informasi dengan alur yang sistematis untuk membuat sistem yang berkualitas tinggi. SDLC merupakan pendekatan pengembangan *software* yang sudah biasa diterapkan. Terdapat banyak model dalam SDLC yang digunakan, seperti *waterfall*, *spiral*, *agile*, dan lain-lain. Namun dari model-model tersebut, model pengembangan yang modern dan terkenal adalah *agile methodology*. Metodologi pengembangan ini terbagi lagi menjadi beberapa macam model

pengembangan. Namun dalam penelitian ini, model pengembangan yang mengedepankan kecepatan proses pengembangan sistem dan sekaligus akan peneliti terapkan adalah *scrum*. *Scrum* merupakan metode yang mampu mengatasi permasalahan dalam proses *development* pada sistem yang rumit [5]. Metode pengembangan ini banyak digunakan dalam bidang pengembangan sistem dan cukup fleksibel jika terdapat perubahan atau penambahan fitur yang diperlukan [11].

2.2. Landasan Teori

2.2.1. Sistem Informasi

Sistem Informasi atau *Information System* merupakan serangkaian komponen yang memiliki visi dan misi yang sama untuk mencapai tujuan tertentu. Pada prinsipnya, sistem informasi berperan sebagai penyedia informasi yang digunakan untuk keperluan manajemen. Beberapa kegiatan yang dilakukan untuk mencapai tujuan tersebut seperti pengumpulan, pemrosesan, penyimpanan dan pendistribusian informasi. Buah dari kegiatan ini dapat digunakan sebagai pengambilan keputusan, koordinasi dan control [6], [15], [16]. Sistem Informasi terdiri dari dua kata yang memiliki makna tersendiri, yaitu sistem dan informasi.

2.2.1.1. Informasi

Data dan informasi adalah hal yang berbeda namun merupakan istilah saling berkaitan erat. Data merupakan sebuah fakta yang belum diproses, baik berupa angka ataupun pernyataan. Sedangkan, informasi merupakan fakta yang sudah diproses dan digabungkan sehingga memiliki makna dan maksud tertentu [15], [17]. Informasi yang diperoleh melalui proses yang melibatkan pengumpulan data dan mengarahkannya ke proses transformasi.

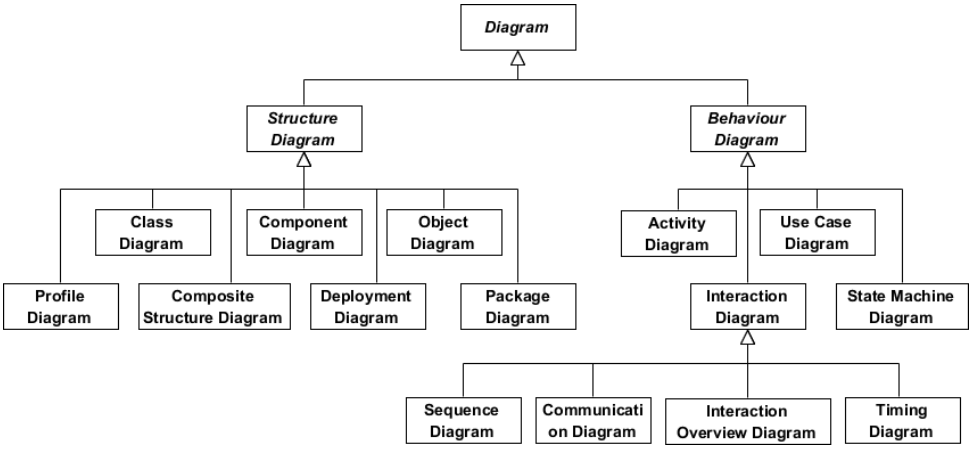
2.2.1.2. Sistem

Sistem adalah serangkaian komponen yang saling berkelompok dan memiliki kesamaan tujuan. Pada prinsipnya,

sistem menghasilkan *output* sebagai data yang diharapkan dari alur proses bisnis yang melibatkan satu atau lebih masukan (*input*) [17].

2.2.2. *Unified Modelling Language* (UML)

Dalam mengembangkan sistem, diperlukan pemodelan sebagai implementasi sistem yang memvisualisasikan bagaimana meletakkan suatu rancang bangun dalam bentuk diagram. Seorang pengembang akan lebih mudah memahami karakteristik dan proses bisnis sistem melalui UML sehingga mempermudah pengembangannya [18]. Terdapat 14 diagram UML yang memiliki sifat dan tujuan tertentu untuk menjelaskan bagaimana karakteristik sistem. Setiap diagram UML terbagi menjadi dua kategori seperti pada **Gambar 2.1** dibawah ini [19].



Gambar 2.1. Kategori dari UML Diagram

Structure Diagram menunjukan struktur statis dari sistem di tingkat abstraksi dan implementasi yang berbeda serta bagaimana mereka terkait satu sama lain. *Behaviour Diagram* menunjukkan sifat dinamis pada objek-objek dari sistem dan dapat memvisualisasikan perubahan sistem pada waktu tertentu. Dari kedua kategori tersebut, terdapat tipe-tipe diagram yang sering digunakan yaitu sebagai berikut [18].

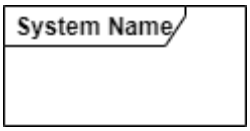
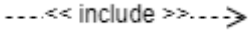
2.2.2.1. *Use Case Diagram*

Diagram ini merepresentasikan bagaimana fungsionalitas atau proses bisnis dari sistem. Biasanya melibatkan antara sistem dengan actor untuk menunjukkan perlaku pada sistem. Pemodelan ini dapat

diterapkan pada sistem perangkat lunak, prediksi prosedur dalam bisnis, dan sebagainya.

Notasi-notasi dalam use case dapat dilihat pada **Tabel 2.3** berikut ini.

Tabel 2.3. Notasi-notasi dari *Use Case Diagram*

Istilah	Keterangan	Simbol
Actor	Seseorang atau sistem yang memperoleh keuntungan dari dan diluar sistem	
Use Case	Merepresentasikan bagian utama dari fungsionalitas sistem	
System Boundary	Merepresentasikan cakupan dalam sistem	
Association relationship	Menghubungkan aktor dengan satu atau lebih use case yang berinteraksi dengannya	
Include relationship	*Use case* memerlukan use case lain yang dihubungkan dengan relasi ini	
Extend relationship	*Use case* dapat berdiri sendiri tanpa use case lain yang dihubungkan dengan relasi ini	

2.2.2.2. *Activity Diagram*

Diagram ini merepresentasikan berbagai aktifitas atau perilaku pada sistem yang dihubungkan oleh aliran data dan kontrol. Secara spesifik, diagram ini memerlihatkan bagaimana jalannya proses bisnis yang didefiniskan pada *use case diagram*.

Notasi-notisi dalam *activity diagram* dapat dilihat pada **Tabel 2.4** berikut ini.

Tabel 2.4. Notasi-notasi dari *Activity Diagram*

Istilah	Keterangan	Simbol
A State	Memiliki nama yang merepresentasikan kondisi dari objek	A State
Initial State	Merepresentasikan *starting point* sebuah objek	
Final State	Merepresentasikan selesainya sebuah *activity*	
Transition	Merepresentasikan transisi antar kondisi	

2.2.2.3. *Class Diagram*

Class diagram merupakan model statis yang mendukung sudut pandang statis pada sistem yang berkembang [19]. Diagram ini merepresentasikan, mendifinisikan, dan merelasikan antar class-class. Sebuah *class* memiliki atribut dan operasi untuk mendefinisikan *class* tersebut. Adapun notasi-notasi dalam diagram ini dapat dilihat pada **Tabel 2.5** berikut ini.

Tabel 2.5. Notasi-notasi dalam *Class Diagram*

Relasi	Keterangan	Simbol
Class	Merepresentasikan pewarisan atribut dan operasi yang dimiliki oleh *parent class* kepada *child class*	Classname + field: type + method(type): type
Attribute	Merepresentasikan properti yang mendeskripsikan sebuah class	+ field: type
Operation	Merepresentasikan operasi atau fungsi yang dapat dikerjakan sebuah class	+ method(type): type

Class memiliki relasi tertentu dengan *class* lain untuk menunjukkan sifat pada *class* tersebut. Relasi-relasi yang digunakan dalam diagram ini dapat dilihat pada **Tabel 2.6** berikut ini [20].

Tabel 2.6. Tipe-tipe hubungan dalam *Class Diagram*

Relasi	Keterangan	Simbol
Inheritance atau Generalization	Merepresentasikan pewarisan atribut dan operasi yang dimiliki oleh parent class kepada child class	
Association	Merepresentasikan hubungan antara jenis class	
Aggregation	Merepresentasikan bentuk khusus dari relasi asosiasi, yang memiliki hubungan yang erat	
Composition	Suatu class tidak dapat berdiri sendiri tanpa class lain	
Dependency	Perubahan yang terjadi pada perilaku suatu class, dapat menyebabkan perubahan juga pada yang lain, namun tak sebaliknya	

Relasi asosiasi juga memiliki *multiplicity*, yang menunjukkan bagaimana class yang instansiasi dapat diasosiasikan dengan objek lain. Angka ditempatkan pada relasi asosiasi yang menunjukkan angka maksimal dan minimal. Hal ini mirip dengan kardinalitas pada *Entity Relationship Diagram* (ERD). Hal tersebut dapat dilihat pada **Tabel 2.7** berikut ini [19].

Tabel 2.7. *Multiplicity* pada dalam *Class Diagram*

Instansiasi	Representasi dari Instansiasi	Keterangan
Tepat satu	1	Hanya memiliki atau dimiliki oleh satu
Nol atau lebih banyak	0..*	Tidak memiliki atau dimiliki oleh nol atau lebih
Satu atau lebih banyak	1..*	Hanya memiliki atau dimiliki oleh satu atau lebih
Nol atau satu	0..1	Tidak memiliki atau dimiliki oleh nol atau satu
Jangkauan angka secara spesifik	2..4	Memiliki atau dimiliki oleh sebanyak dua hingga empat
Beberapa, rentang terputus-putus	1..3, 5	Memiliki atau dimiliki oleh satu hingga tiga dan lima

2.2.2.4. *Sequence Diagram*

Diagram ini menunjukkan interaksi-interaksi yang terjadi antar objek beserta urutannya. Diagram ini bertujuan untuk menyajikan secara rinci sebuah perilaku pada sistem.

2.2.2.5. *Deployment Diagram*

Diagram ini merepresentasikan model arsitektur dari node *hardware* dan *processor* dari sistem dan menggambarkan node dimana komponen perangkat lunak berada. *Node* merupakan *hardware* yang memiliki memori atau menyediakan lingkungan untuk menjalankan proses.

2.2.2.6. *State Machine Diagram*

Diagram ini merepresentasikan siklus hidup *runtime* pada sebuah objek dalam memori atau menggambarkan transisi serta perubahan pada objek pada sistem.

2.2.3. *Website*

Halaman *web* adalah sebuah halaman yang menampilkan dokumen, seperti teks, gambar, suara, *video*, dan lain-lain. *Website* atau *web* adalah sekumpulan halaman web yang dikelompokan dan berkaitan satu sama lain [21]. Menurut [22], *web* adalah kumpulan halaman situs yang menampilkan sejumlah informasi, baik berupa teks, gambar, *audio*, *video* dan/atau gabungan dari media-media tersebut, yang tersedia melalui jaringan nirkabel/internet.

Dalam pengembangan *website*, beberapa perlengkapan yang dibutuhkan seperti *code editor*, alat pengunggah *website*, layanan *web hosting*, dan browser.

2.2.3.1. *Code Editor*

Terdapat berbagai macam *code editor* yang memiliki berbagai tingkat fungsionalitas juga. Contohnya Notepad yang merupakan aplikasi buatan Microsoft yang secara gratis dan *default* telah ada jika kita adalah pengguna sistem operasi windows. Untuk menggunakannya, *web developer* cukup memasukkan teks yang

menjadi baris kode sehingga pada halaman web akan muncul sesuai apa yang kita masukkan [23].

Notepad merupakan salah satu pengedit kode dari berbagai macam *code editor*. Meskipun *web developer* dapat membuat *website* menggunakan Notepad, namun *code editor* lain lebih kaya fitur sehingga mempermudah proses pengembangan *web*. Minimal *code editor* dapat memberikan rekomendasi kode yang valid setelah *developer* mengetikkan sebagian dari suatu kode/perintah. Ini dilakukan dengan menampilkan *pop-up* kepada *developer* yang merekomendasikan kode/perintah valid dari perintah yang sedang dimasukkan. Mekanisme ini sering disebut dengan *intellisense*. Contoh *code editor* yang memiliki kemampuan tersebut antara lain Visual Studio Code, Sublime Text, Notepad++, Atom dan sebagainya.

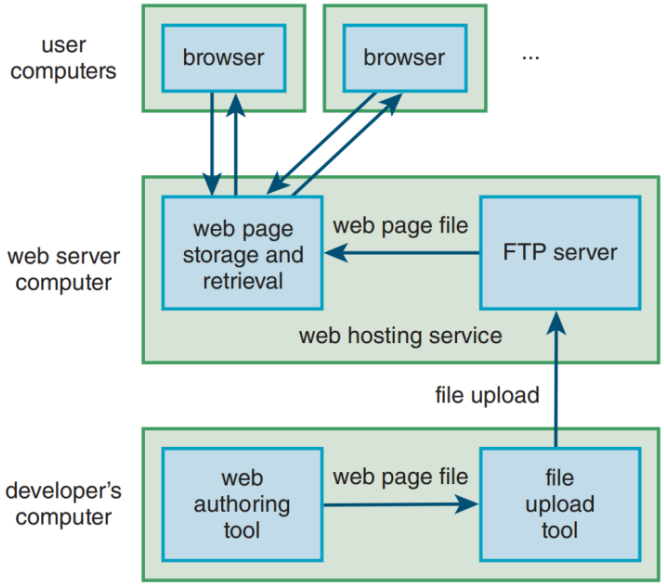
2.2.3.2. Alat Pengunggah *Website*

Aplikasi *website* yang telah dibangun di lokal komputer dengan menggunakan *code editor* kesayangan, tentu perlu dipublikasikan menuju *web server* sehingga dapat diakses/dilihat oleh banyak pengguna *website*. Untuk melakukan pengunggahan diperlukan alat pengunggah berkas *web* seperti WinSCP, Filezilla, dan sebagainya [23].

2.2.3.3. Layanan *Web Hosting*

Web server merupakan tempat penyimpanan seluruh file pengembangan *website* yang telah di bangun sehingga *website* tersebut dapat diakses oleh para pengguna komputer dan *mobile*. Oleh karena itu, seorang developer harus memiliki layanan *web hosting*. Jika pengembang *website* merupakan bagian dari perusahaan sedang hingga besar, biasanya memiliki layanan *web* hosting sendiri. Namun jika tidak memilikinya, terdapat banyak penyedia layanan *web hosting* (misalnya hostinger.com). Terlepas dari siapa yang bertanggung jawab atas layanan *web hosting*, semua

layanan *web hosting* tersebut harus memiliki mekanisme penyimpanan berkas file dari hasil pengembangan *website* [23]. Contohnya adalah FTP *server* yang berjalan pada *web server*. Berikut ini adalah bagan tentang bagaimana proses penyimpanan file melalui FTP *server* menuju *web server*.



Gambar 2.2. Proses unggah berkas *website* [23]

2.2.3.4. *Browser*

Browser merupakan perangkat lunak yang berfungsi sebagai sarana pengguna untuk mengambil dan melihat konten halaman *web*. Di sisi pengembang *web*, *browser* juga digunakan sebagai proses pengembangan *web* untuk menampilkan konten *website* dari hasil penulisan kode-kode pada *code editor*. *Browser* yang biasa digunakan adalah Google Chrome, browser milik Microsoft (Microsoft Edge dan Internet Explorer) dan Mozilla Firefox. Bagi pengguna *mobile device*, terdapat browser popular lainnya seperti Safari (iOS) dan *browser* bawaan Android [23].

Dalam pengembangan *website*, *web developer* tidak hanya dituntut untuk memiliki perangkat lunak pendukung untuk membangun aplikasi *website* yang utuh. Namun seorang pengembang juga harus memahami dan menguasai beberapa bahasa pemrograman sehingga dapat menunjang seluruh

kebutuhan dalam pengembangan *website* dengan baik sebagaimana mestinya. Selain itu, juga diperlukan perancangan *database* yang baik dan benar. Beberapa bahasa pemrograman dan *Database Management System* (DBMS) yang digunakan peneliti adalah sebagai berikut.

2.2.3.1. HTML

Dasar dari dikembangkannya sebuah *website* adalah menggunakan HTML (*Hyper Text Markup Language*), yaitu bahasa *markup* sebagai pembangun kerangka halaman *website*. Halaman *web* akan menampilkan berbagai konten berdasarkan elemen-elemen HTML yang didefinisikan pada berkas berekstensi .html (dibaca: dot html). Beberapa fungsi HTML antara lain sebagai berikut [22].

a. Membentuk tampilan *website*
b. Menampilkan media seperti teks, gambar, video dan audio
c. Membuat daftar
d. Membuat tabel
e. Membuat tautan, baik berupa teks maupun gambar
f. Membuat formulir, sebagai tempat *input* data oleh pengguna.

2.2.3.2. CSS

CSS merupakan kependekan dari *Cascading Style Sheets*, yaitu sebuah standar W3C untuk mendefinisikan tampilan dokumen HTML sehingga halaman *website* terlihat lebih menarik [24]. Halaman *website* yang dikembangkan menggunakan HTML dapat dianalogikan sebagai kerangka manusia yang terdiri dari berbagai tulang dan macam indra. Sedangkan CSS merupakan bahasa *markup* untuk menghias tampilan halaman *website*. Berikut ini merupakan manfaat penggunaan CSS [24].

a. Pengaturan tipe dan tata letak yang presisi
b. Memungkinkan mengubah tampilan halaman menggunakan satu berkas CSS
c. Halaman diakses dengan mudah

2.2.3.3. JavaScript

JavaScript merupakan bahasa pemrograman untuk memberikan interaktivitas dan perilaku khusus pada suatu elemen HTML pada halaman *website* [24]. Bahasa ini merupakan bahasa yang berjalan di sisi klien/pengguna, dalam arti lain berjalan pada mesin pengguna, bukan di *server*, seperti bahasa pemrograman lainnya yaitu PHP dan Ruby. Dalam hal ini, javascript bergantung pada kemampuan *browser* di komputer pengguna.

2.2.3.4. PHP

PHP merupakan kependekan dari *Hypertext Preprocessor*. Konten yang ditampilkan secara dinamis dalam halaman *website* atau merender halaman *web* secara lengkap yang dilayani oleh *web server* dibangun menggunakan bahasa *scripting*, yaitu PHP. Dalam [25] menerangkan bahwa PHP adalah bahasa yang digunakan untuk menghasilkan *output-output* dinamis oleh *server* yang berpotensi melakukan *request* yang berbeda ke *server* dari waktu ke waktu.

Saat ini, terdapat banyak *framework* yang diciptakan menggunakan bahasa pemrograman PHP untuk membangun aplikasi modern. Salah satu *framework* yang terkenal dan banyak digunakan oleh para pengembang adalah laravel. Beberapa kelebihan penggunaan *framework* laravel adalah membantu menghindari banyak penulisan kode, implementasi penggunaan standar yang modern, memiliki banyak komunitas, dan sebagainya [26]. Selain itu, dalam [27] menyebutkan bahwa laravel dapat meningkatkan kualitas sistem dengan meminimalisir pengeluaran dana dan peningkatan sistem serta penulisan kode yang menjadi lebih bersih dan sederhana.

2.2.3.5. MySQL

Database adalah kumpulan data atau informasi terstruktur yang disimpan dalam sistem komputer dan dapat dikembalikan dengan cepat dan tepat. Dalam [25] menjelaskan salah satu DBMS

yang paling terkenal untuk *web server* adalah MySQL. SQL dalam MySQL merupakan singkatan dari *Structured Query Language*. *Database* dalam MySQL terdiri dari satu atau lebih tabel, yang masing-masing tabel memiliki *record*/baris yang menyimpan berbagai data yang saling berkaitan. Setiap baris memiliki kolom yang menyimpan data itu sendiri.

Dalam *database* terdapat beberapa istilah yaitu sebagai berikut [25].

a. *Database*, adalah wadah dari sekumpulan tabel dalam MySQL.
b. *Table*, merupakan wadah dalam *database* yang menyimpan berbagai macam data.
c. *Row*, merupakan kumpulan data tunggal dalam *table*.
d. *Column*, merupakan nama kolom/*field* dalam suatu *row*

2.2.4. Pengujian *Black-Box*

Pengujian *black-box* adalah metode pengujian perangkat lunak yang mana seluruh fungsionalitas dari aplikasi di uji tanpa harus memiliki pengetahuan mengenai struktur kode didalamnya, rincian implemetasi kode dan jalur kodenya. Pengujian ini berfokus pada masukan dan keluaran dari aplikasi dan ini sepenuhnya berdasarkan dari kebutuhan dan spesifikasi *software* [28].

Gambar 2.3. Ilustrasi pengujian black-box [28]

Pengujian *black-box* memiliki banyak jenis yang dapat dilakukan oleh seorang penguji. Namun terdapat jenis pengujian yang paling sering digunakan antara lain sebagai berikut [29].

2.2.4.1. *Functional Testing*

Functional Testing merupakan perngujian yang berfokus pada suatu fungsi atau fitur secara rinci pada sebuah perangkat lunak.

2.2.4.2. *Non-Functional Testing*

Non-Functional Testing merupakan pengujian yang berfokus pada pemeriksaan aspek tambahan dari perangkat lunak. Hal tersebut berarti pengujian ini tidak memeriksa dapat melakukan kemampuan tertentu, namun bagaimana perangkat lunak melakukan kemampuan tersebut.

2.2.4.3. *Regression Testing*

Regression Testing adalah pengujian yang memeriksa apakah versi terbaru dari suatu perangkat lunak terjadi penurunan kemampuan saat terjadi peningkatan versi. Pengujian regresi ini dilakukan dari dua aspek, yaitu aspek fungsional dan non-fungsional.

2.2.5. Pengujian *White-Box*

Pengujian *white-box* adalah teknik pengujian perangkat lunak yang mana struktur dalam, desain dan kode program dari perangkat lunak di uji untuk memverifikasi alur input-output serta meningkatkan desain, kegunaan dan keamanan. Dalam pengujian ini, kode program diperlihatkan kepada pengiji sehingga pengujian ini bisa disebuat juga *Clear box testing*, *open box testing*, *transparent box testing*, *code-based testing* dan *glass box testing* [30].

Pengujian ini memiliki beberapa teknik pengujian diantaranya sebagai berikut [31], [32].

2.2.5.1. *Statement Coverage*

Teknik ini bertujuan untuk menelusuri semua pernyataan setidaknya sekali. Hal ini juga membantu pengiji untuk menemukan kode yang salah.

2.2.5.2. *Branch Coverage*

Teknik ini bertujuan untuk memeriksa setiap cabang kode dilewati setidaknya sekali.

2.2.5.3. *Condition Coverage*

Teknik ini bertujuan untuk menguji seluruh kondisi agar menghasilkan true atau false.

2.2.5.4. *Multiple Condition Coverage*

Teknik ini bertujuan untuk menguji seluruh kombinasi kondisi yang perlu diuji setidaknya sekali.

2.2.5.5. *Basis Path Testing*

Teknik pengujian ini bertugas untuk melakukan pengukuran kompleksitas kode program serta mendefinisikan beberapa kemungkinan alur kode program.

2.2.5.6. *Loop Testing*

Teknik pengujian loop merupakan pengujian yang dilakukan pada kode perulangan, seperti while, for, do-while. Dari pengujian ini, pengiji harus memeriksa kondisi awal dan akhir dari perulangan, apakah telah berjalan sesuai harapan atau sebaliknya.

2.2.6. Pengujian *System Usability Scale*

Pengujian *System Usability Scale* (SUS) merupakan salah satu metode atau teknik dalam pengujian *usability* untuk melakukan pengukuran kemudahan penggunaan perangkat lunak digunakan oleh pengguna. Teknik pengujian ini memiliki sepuluh pertanyaan yang menjadi tolak ukur pengujian. Pengujian ini juga dapat digunakan pada berbagai *platform*, seperti *website*, *mobile application*, dan sebagainya. Instrumen dalam pengujian SUS dapat dilihat pada **Tabel 2.8** berikut ini.

Tabel 2.8. Instrumen dalam pengujian SUS

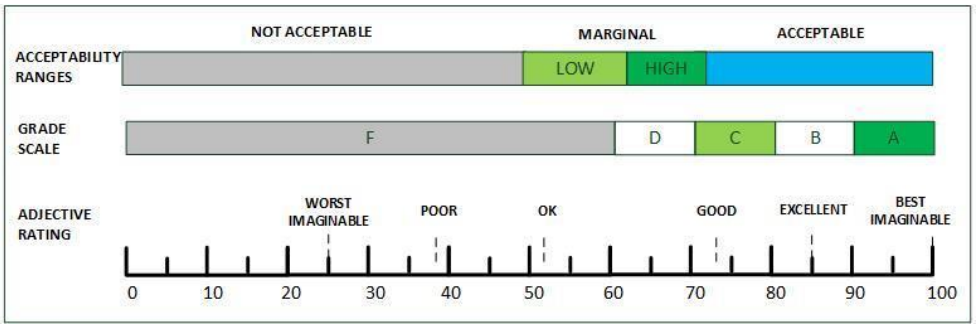
No.	Pertanyaan
1.	Saya akan ingn lebih sering menggunakan aplikasi ini
2.	Saya merasa aplikasi ini tidak harus dibuat serumit ini
3.	Saya pikir aplikasi mudah untuk digunakan
4.	Saya membutuhkan bantuan dari orang teknis untuk menggunakan aplikasi ini
5.	Saya menemukan fitur pada aplikasi terintegrasi dengan baik
6.	Saya pikir ada ketidaksesuaian dalam aplikasi ini
7.	Saya merasa kebanyakan orang mudah untuk mempelajari aplikasi dengan sangat cepat
8.	Saya menemukan, aplikasi sangat rumit untuk digunakan
9.	Saya percaya diri untuk menggunakan aplikasi ini
10.	Saya perlu belajar sebelum saya menggunakan aplikasi

Setiap pertanyaan diatas memiliki rentang skor yang sama, yaitu 1 (satu) – 5 (lima). Pernyataan dengan skor 1 (satu) menunjukkan bahwa penguji sangat tidak setuju dengan pernyataan tersebut. Sedangkan pernyataan dengan skor 5 (lima) menunjukkan bahwa penguji sangat setuju dengan pernyataan tersebut.

Data-data hasil kuesioner dari para responden yang telah terkumpul harus dilakukan proses perhitungan untuk mendapatkan skor rata-rata. Proses perhitungan dilakukan dalam beberapa tahap sebagai berikut.

a. Setiap nilai dari pernyataan ganjil dikurangi satu.
b. Setiap nilai responden dari pernyataan genap digunakan untuk mengurangi dengan angka lima.
c. Hasil dari dua tahap perhitungan sebelumnya dijumlahkan. Kemudian dikalikan dengan angka 2,5 (dua koma lima).
d. Setelah setiap hasil responden diproses, seluruh hasil tersebut dibagi sebanyak jumlah responden yang didapatkan untuk mendapatkan skor rata-rata.

Dalam penentuan hasil perhitungan, pengujian teknik SUS menggunakan cara *SUS score percentile rank* [33]. Berikut ini adalah ketentuan penilaian pada *SUS score percentile rank*.



Gambar 2.4. Penilaian SUS menggunakan SUS score percentile rank [33]