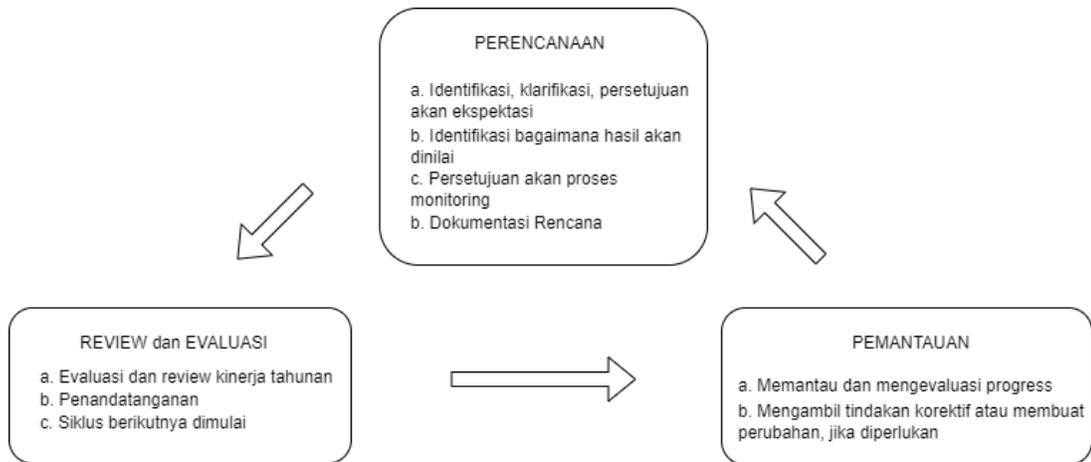


BAB II TINJAUAN PUSTAKA

II.1 Manajemen Kinerja

Manajemen kinerja adalah suatu proses yang berlangsung di dalam perusahaan, di mana manajer dan karyawan berkolaborasi dalam merencanakan, memantau, dan meninjau target kerja serta kontribusi karyawan terhadap organisasi. Proses ini mencakup komunikasi yang terbuka dan berkelanjutan, pembentukan visi bersama, serta pendekatan strategis yang terpadu yang berfungsi sebagai faktor utama dalam mencapai tujuan organisasi (Budiarto, 2017). Gambar II.1 menunjukkan siklus manajemen kinerja.



Gambar II. 1 Siklus Manajemen Kinerja

Berikut adalah penjelasan dari siklus sistem manajemen kinerja yang tergambar pada Gambar II.1:

1. **Tahap Perencanaan (*Planning*):** Tahap ini merupakan usaha kolaboratif antara manajer dan karyawan, meliputi:
 - a. Peninjauan deskripsi pekerjaan karyawan untuk memastikan kesesuaian dengan pekerjaan yang dilakukan.
 - b. Penilaian keterkaitan antara deskripsi pekerjaan karyawan dengan rencana kerja, tujuan, dan strategi perusahaan.

- c. Pembuatan rencana kerja yang *mendetail* tentang tugas-tugas yang harus dilakukan, hasil yang diharapkan, dan standar atau langkah untuk evaluasi kinerja.
 - d. Penentuan 3-5 area kinerja utama yang menjadi target selama satu tahun.
 - e. Identifikasi tujuan pelatihan untuk mengembangkan keterampilan dan kompetensi karyawan sesuai dengan kebutuhan pekerjaan mereka.
 - f. Pengidentifikasian tujuan pengembangan karir yang akan diintegrasikan dalam perencanaan karir jangka panjang karyawan.
2. **Tahap Pemantauan:** Proses ini tidak melibatkan pengawasan secara *detail* terhadap cara karyawan menjalankan tugas-tugasnya, melainkan fokus pada hasil yang dicapai serta perilaku individu dan dinamika tim yang berdampak pada lingkungan kerja.
 3. **Tahap Penilaian (*Review*):** Tahap ini merupakan kesempatan untuk meninjau dan meringkas kinerja karyawan selama periode tersebut, yang diwujudkan melalui pertemuan penilaian kinerja atau *appraisal meeting*.

II.3 Management Information System (MIS)

Manajemen Sistem Informasi (MIS) adalah sistem yang menggabungkan teknologi informasi dengan kegiatan bisnis untuk membantu perusahaan mencapai tujuan strategisnya. Menekankan bahwa integrasi antara teknologi informasi dan proses bisnis sangat penting untuk meningkatkan efisiensi dan efektivitas organisasi. MIS melibatkan komponen seperti perangkat keras, perangkat lunak, data, prosedur, dan sumber daya manusia yang bekerja sama untuk mengumpulkan, menyimpan, memproses, dan mendistribusikan informasi. Fungsi utama MIS meliputi peningkatan efisiensi melalui otomatisasi proses bisnis, penyediaan data akurat untuk pengambilan keputusan yang lebih baik, dan membantu perusahaan merespons perubahan pasar serta kebutuhan pelanggan guna memperoleh keunggulan kompetitif (Laudon, 2020)

II.4 Rapid Application Development (RAD)

Rapid Application Development adalah metodologi pengembangan perangkat lunak yang berfokus pada pengembangan cepat dan iteratif, dengan keterlibatan aktif pengguna. Ini memungkinkan tim pengembang untuk menghasilkan perangkat lunak yang memenuhi kebutuhan pengguna dengan lebih cepat dan efisien. Meskipun memiliki beberapa kekurangan, seperti kebutuhan akan keterlibatan pengguna yang tinggi dan mungkin tidak cocok untuk proyek skala besar, RAD tetap menjadi pilihan yang efektif untuk banyak proyek pengembangan perangkat lunak yang membutuhkan fleksibilitas dan kecepatan. Pendekatan ini memastikan bahwa solusi yang dihasilkan tidak hanya cepat dan tepat sasaran, tetapi juga sesuai dengan kebutuhan dan tujuan organisasi (McConell, 2010). Setiap fase dalam *Rapid Application Development* (RAD) menekankan aspek tertentu dari pengembangan sistem. Berikut adalah penjelasan mengenai fase-fase utama dalam RAD (Sasmito, 2020):

1. Perencanaan Kebutuhan (*Requirement Planning*): Pada tahap ini, pengguna dan analis bertemu untuk mengidentifikasi tujuan sistem dan kebutuhan informasi yang diperlukan untuk mencapai tujuan tersebut. Keterlibatan dari kedua belah pihak dianggap sebagai aspek krusial pada tahap ini.
2. Proses Desain Sistem (*Design System*): Pada fase ini, keterlibatan aktif dari pengguna menjadi penentu dalam mencapai tujuan, karena proses ini melibatkan perancangan dan perbaikan berdasarkan umpan balik antara pengguna dan analis. Pengguna memiliki kemampuan untuk memberikan komentar langsung terkait ketidaksesuaian desain, dan desain sistem dibangun dengan merujuk pada dokumentasi kebutuhan pengguna yang telah dibuat pada tahap sebelumnya. Hasil dari tahap ini mencakup spesifikasi perangkat lunak, termasuk struktur sistem secara keseluruhan, struktur data, dan elemen-elemen lainnya.
3. Implementasi (*Implementation*): Para pengembang program bertanggung jawab untuk merancang program berdasarkan desain yang telah disetujui oleh pengguna dan analis. Sebelum program diimplementasikan di organisasi, dilakukan uji coba untuk memastikan tidak ada kesalahan. Pengguna juga

memberikan umpan balik terhadap sistem yang telah dibuat dan memberikan persetujuan untuk implementasinya. Namun, tahap ini tidak diterapkan dalam tugas akhir ini karena rancangan yang dibuat hanya berupa usulan dan tidak akan langsung diimplementasikan.

II.5 Perancangan Sistem

II.5.1 *Laravel Framework*

Laravel merupakan *framework* PHP yang sangat berguna baik bagi pemula maupun *programmer* berpengalaman. *Framework* ini membantu mempercepat pengembangan dan peluncuran aplikasi *web* dengan metode PHP berorientasi objek yang canggih. Sintaksisnya yang mudah dipahami dan fungsionalitas modernnya menarik bagi pengembang yang ingin membuat aplikasi yang andal. Penggunaan *framework* mempermudah pengembangan karena menyediakan berbagai modul dan koneksi yang diperlukan. *Web-framework* ini menyediakan fondasi awal untuk pembuatan aplikasi dan memungkinkan pengembang untuk lebih fokus pada aspek-aspek yang lebih menarik. *Laravel* juga menawarkan fitur-fitur kuat seperti lapisan abstraksi *database* yang ekspresif, *dependency injection*, dan skalabilitas yang tinggi (Subecz, 2021).

II.5.2 *Mysql*

MySQL adalah sistem manajemen basis data relasional yang menggunakan *Structured Query Language* (SQL) untuk mengelola dan mengakses data secara efisien. Data dalam *MySQL* disimpan dalam bentuk tabel yang terdiri dari baris dan kolom, di mana setiap baris merepresentasikan rekaman dan setiap kolom berisi bidang data tertentu. SQL memungkinkan permintaan sederhana dan pengambilan data yang cepat, membuat *MySQL* ideal untuk berbagai aplikasi dan sistem yang memerlukan pengelolaan data terstruktur dan cepat (Nixon, 2021).

II.5.3 *Unified Modelling Language*

UML (*Unified Modeling Language*) adalah suatu bahasa standar yang umumnya digunakan dalam industri untuk merinci persyaratan, melakukan analisis, dan merancang sistem. Selain itu, UML juga digunakan untuk menjelaskan struktur dan arsitektur dalam pemrograman berorientasi objek (Putra & Andriani, 2019). UML

dapat ditampilkan dalam bentuk grafik dan bagan beserta grafik dan bagan yang menunjukkan kondisi dunia nyata saat ini. Berikut merupakan beberapa model dari UML:

a. *Use Case Diagram*

Use Case Diagram adalah metodologi yang digunakan dalam analisis sistem untuk mengidentifikasi, mengklarifikasi, dan persyaratan sistem.

b. *Sequence Diagram*

Sequence Diagram merupakan salah satu representasi perilaku yang sering digunakan dalam UML. Diagram ini memvisualisasikan interaksi dan pertukaran pesan antar objek sepanjang waktu. Diagram urutan digunakan untuk mengilustrasikan bagaimana pesan-pesan dikirim di antara objek atau instance untuk mengeksekusi suatu tugas. Secara sederhana, diagram urutan membantu memahami cara objek berkomunikasi dan berbagi informasi dalam suatu proses atau skenario tertentu.

c. *Activity Diagram*

Activity Diagram adalah merupakan salah satu dari lima diagram UML yang sering digunakan dalam industri. Prioritas tinggi pada penggunaan Diagram Kelas, yang menempati peringkat pertama sebagai diagram UML yang paling umum, menjadi alasan mengapa memilih untuk menekankan kedua model ini.

d. *Class Diagram*

Diagram kelas menggambarkan struktur sistem dengan mendefinisikan berbagai kelas yang akan digunakan untuk membangun sistem tersebut.

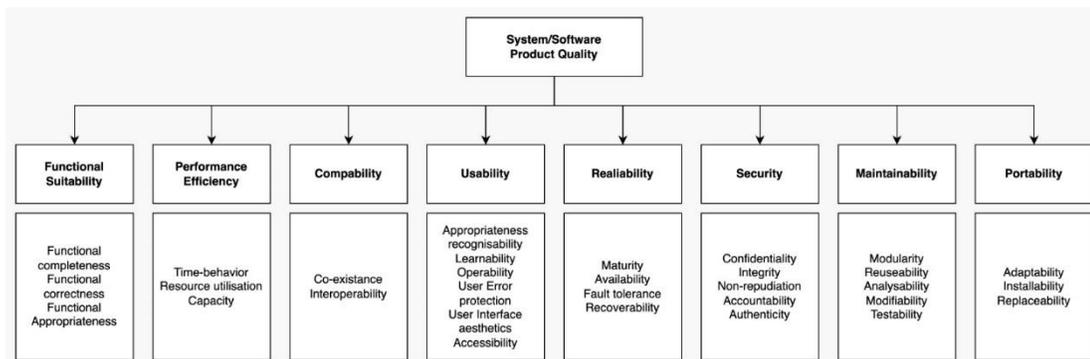
II.5.4 Black Box Testing

Black box testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian aspek fungsional dari perangkat lunak tersebut. Tujuannya adalah untuk menemukan fungsi-fungsi yang tidak berjalan dengan benar, kesalahan pada antarmuka pengguna, kesalahan dalam struktur data, masalah kinerja, serta kesalahan pada proses inisialisasi dan terminasi sistem (Wijaya & Astuti, 2021).

II.5.5 User Acceptance Testing

User Acceptance Testing (UAT) adalah tahap pengujian dalam metode *V-Model* yang melibatkan pengguna akhir atau pemangku kepentingan yang relevan untuk menguji sistem perangkat lunak. Tujuan UAT adalah memastikan bahwa sistem memenuhi kebutuhan bisnis dan persyaratan pengguna. Pengujian ini dilakukan dalam lingkungan dan skenario penggunaan nyata sebelum sistem diterima dan diimplementasikan. UAT biasanya dilakukan sebelum peluncuran fitur baru dalam aplikasi untuk memastikan bahwa semua aspek sistem telah diuji dan siap digunakan dalam operasi sehari-hari (Anjasmara & Arif, 2023). Dalam tugas akhir ini, UAT diterapkan dengan standar ISO 25010. Standar ini mencakup delapan karakteristik yang mengacu pada sifat dinamis dan statis dari sistem komputer. Karakteristik dan sub-karakteristik tersebut memberikan istilah yang konsisten untuk mendefinisikan, mengukur, dan mengevaluasi kualitas sistem dan perangkat lunak (Pratama & Mutiara, 2021)

Hal ini dapat dicapai dengan mendefinisikan karakteristik kualitas yang diperlukan dan diinginkan yang terkait dengan tujuan dan sasaran pemangku kepentingan terhadap sistem tersebut, berikut merupakan gambaran delapan indikator dalam ISO 25010 (ISO, 2011).



Gambar II. 2 *Product Quality Model*

Dalam pengembangan aplikasi, prioritas utama adalah memenuhi kebutuhan dan harapan pengguna. Oleh karena itu, dengan fokus pada tiga parameter utama yaitu *Functional Suitability*, *Performance Efficiency*, dan *Usability*, dapat secara tepat menargetkan kebutuhan utama pengguna. Berikut penjelasan lebih lanjut mengenai ketiga parameter ini:

1. *Functional Suitability*: Memastikan aplikasi menawarkan semua fungsi yang dibutuhkan pengguna (*Functional Completeness*). Hal ini penting untuk memberikan solusi lengkap bagi kebutuhan pengguna, sehingga meningkatkan kepuasan dan kepercayaan pengguna terhadap aplikasi.
2. *Performance Efficiency* : Memastikan aplikasi memiliki waktu respons yang cepat (*Time-behaviour*). Pengguna menginginkan aplikasi yang responsif dan cepat untuk meningkatkan efisiensi dalam menyelesaikan tugas, mengurangi frustrasi, dan memberikan pengalaman yang lebih baik.
3. *Usability* : Memastikan aplikasi mudah digunakan dan dipahami (*Learnability*). Dengan membuat aplikasi yang mudah dipelajari, pengguna baru dapat dengan cepat memahami cara menggunakan aplikasi, mengurangi waktu pembelajaran, dan meningkatkan adopsi aplikasi.

II.5.6 Software Development Life Cycle (SDLC)

Siklus Hidup Pengembangan Sistem (SDLC) atau Siklus Hidup Sistem adalah proses dalam rekayasa sistem dan perangkat lunak yang melibatkan pembuatan dan perubahan sistem, serta model dan metodologi yang digunakan untuk mengembangkan sistem tersebut. Konsep SDLC ini biasanya mengacu pada pengembangan sistem komputer atau sistem informasi. Dalam rekayasa perangkat lunak, SDLC menjadi dasar bagi berbagai metodologi pengembangan perangkat lunak. Metodologi ini membentuk kerangka kerja untuk merencanakan dan mengendalikan proses pengembangan sistem informasi, yang pada dasarnya adalah proses pengembangan perangkat lunak. Terdapat tiga jenis metode siklus hidup sistem yang banyak digunakan: siklus hidup sistem tradisional, siklus hidup menggunakan *prototyping*, dan siklus hidup sistem berorientasi objek. Konsep SDLC juga mendasari berbagai model pengembangan perangkat lunak, seperti model *waterfall*, *prototyping*, *iteratif*, *spiral*, *rapid application development* (RAD), dan lainnya (Kurniyanti & Murdiani, 2022). Tahapan-tahapan dalam SDLC secara keseluruhan, sebagaimana dijelaskan dalam buku Rosa (2022), adalah sebagai berikut:

1. **Inisiasi (*initiation*):** Tahap ini biasanya dimulai dengan penyusunan proposal proyek perangkat lunak.
2. **Pengembangan konsep sistem (*system concept development*):** Menentukan lingkup konsep yang mencakup dokumen lingkup sistem, analisis manfaat dan biaya, rencana manajemen, serta pembelajaran kemudahan sistem.
3. **Perencanaan (*planning*):** Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya, serta menyediakan dasar untuk memperoleh sumber daya yang diperlukan guna mencapai solusi.
4. **Analisis kebutuhan (*requirement analysis*):** Menganalisis kebutuhan pengguna sistem perangkat lunak, mengidentifikasi kebutuhan tersebut, dan menyusun dokumen kebutuhan fungsional.
5. **Desain (*design*):** Mengubah kebutuhan *detail* menjadi kebutuhan yang lengkap, dengan dokumen desain sistem yang berfokus pada cara memenuhi fungsi-fungsi yang diperlukan.
6. **Pengembangan (*development*):** Mengonversi desain menjadi sistem informasi yang lengkap, mencakup bagaimana memperoleh dan menginstalasi lingkungan sistem yang diperlukan.
7. **Implementasi (*implementation*):** Termasuk persiapan untuk implementasi, penerapan perangkat lunak dalam lingkungan produksi (pengguna), serta penanganan dan penyelesaian masalah yang teridentifikasi.

II.6 Pemilihan Teori/Model/Kerangka Standar Perancangan

Tabel II.1 merupakan perbandingan dalam pemilihan metode standar perancangan.

Tabel II. 1 Metode Standar Perancangan

Aspek	Metode RAD	Metode <i>Waterfall</i>
Evaluasi	Melibatkan pengguna secara langsung dalam evaluasi pengembangan sistem. Berfokus pada iterasi dan umpan balik kontinu untuk memastikan sistem memenuhi kebutuhan pengguna dengan baik (Laudon & Laudon, 2014).	Evaluasi pengembangan sistem terjadi pada akhir setiap fase. Melibatkan pengguna lebih terbatas dan umpan balik mungkin baru diterima setelah seluruh sistem selesai dikembangkan (Pricillia & Zulfachmi, 2021).
Dokumentasi	Dokumentasi kebutuhan sistem dilakukan baik secara formal maupun informal. Lebih fleksibel terhadap perubahan kebutuhan, sehingga dokumentasi dapat disesuaikan secara iteratif selama siklus pengembangan	Dokumentasi dilakukan secara mengikuti alur yang sudah ditetapkan. Setiap fase memiliki dokumen yang lengkap sebelum melanjut ke fase berikutnya.
Pengembangan Sistem	Pengembangan sistem dilakukan secara berkelanjutan mengikuti kebutuhan dari pengguna sistem (Dennis, Wixom , dan Tegarden , 2015).	Pengembangan sistem harus dijalankan sesuai alur yang sudah ditetapkan, apabila ada perubahan perlu dilakukan iterasi ulang, dan dimulai dari iterasi awal (Pricillia & Zulfachmi, 2021).

Berdasarkan Perbandingan pada Tabel II.1, maka diputuskan bahwa tugas akhir menggunakan metode *Rapid Application Development* sebagai metode untuk pengembangan sistem pada tugas akhir ini dengan mempertimbangkan beberapa aspek sebagai berikut:

1. Fleksibilitas perubahan yang dapat mengadaptasi perubahan atau penyesuaian dari *stakeholder* yang memiliki kepentingan dengan cepat selama siklus pengembangan tanpa mengganggu proses lain yang bersangkutan.
2. Memungkinkan untuk mendorong partisipasi langsung dari pengguna yang memungkinkan untuk mendapat umpan balik secara langsung. Hal tersebut dapat meminimalisir kesalahan dari pengembangan sistem dan juga dapat memenuhi ekspektasi dari pengguna akhir.
3. Pengembangan yang cepat, dengan RAD dapat memungkinkan menghasilkan *prototipe* dengan cepat. Yang memungkinkan untuk *Stakeholder* yang memiliki kepentingan dapat melihat dan menguji solusi lebih awal ketika dalam siklus pengembangan.