

BAB II

TINJAUAN PUSTAKA

2.1 Perencanaan Sistem

Pada bab ini Penulis akan membahas teori yang menjadi dasar penelitian ini, meliputi sistem informasi, aplikasi berbasis website, dan sistem *task manager* sebagai salah satu acuan pembuatan aplikasi serta komponen-komponen pembangun aplikasi ini.

2.1.1 Definisi Sistem

Pengertian sistem menurut (Tukino, 2018) sistem dapat dikatakan sebagai sebuah rangkaian jaringan kerja dari berbagai elemen - elemen yang saling berhubungan guna untuk mencapai tujuan tertentu.

Menurut (Erawati, 2019) sistem adalah jaringan proses kerja yang saling terkait dan berkumpul guna untuk mencapai sebuah tujuan serta melakukan suatu.

2.1.2 Aplikasi

Syani & Werstantia (2019: 88) menyatakan bahwa aplikasi adalah sebuah perangkat lunak yang berisi sebuah coding atau perintah yang dimana bisa diubah sesuai dengan keinginan.

Sari (2017: 83) menyatakan bahwa aplikasi adalah sebuah perangkat lunak yang dimana tujuannya adalah agar bisa melayani setiap aktivitas komputerisasi yang dilakukan oleh pengguna.

2.2 Aplikasi Berbasis Website

Menurut (Elgamar, 2020:3), website adalah suatu media yang terdiri dari beberapa halaman yang saling berkaitan satu sama lain, dan berfungsi sebagai media untuk menampilkan suatu informasi, baik berbentuk gambar, video, teks, suara, ataupun gabungan dari semuanya.

a. HTML

HyperText Markup Language atau dikenal dengan HTML, adalah bahasa markah standar yang digunakan untuk membangun dan merancang halaman web. HTML menggunakan tag dan elemen yang mengandung instruksi-instruksi untuk mengatur tampilan dan struktur konten pada halaman website.

b. PHP

Hypertext Preprocessor/ PHP merupakan bahasa pemrograman yang digunakan secara luas untuk pengembangan aplikasi web. PHP adalah bahasa pemrograman server-side, artinya kode PHP dijalankan di server web dan menghasilkan output dalam bentuk HTML yang kemudian dikirimkan ke peramban website pengguna.

c. CSS

Cascading Style Sheets/ CSS merupakan bahasa yang digunakan untuk mengatur tampilan dan tata letak halaman web yang ditulis dalam HTML. CSS berfungsi untuk memisahkan presentasi visual dari struktur konten pada halaman web, memungkinkan pengembang web untuk lebih fleksibel dan efisien dalam merancang tampilan situs web.

d. MySQL

MySQL adalah sebuah sistem manajemen basis data relasional (Relational Database Management System/RDBMS) yang berbasis open-source.

2.2.1 Kelebihan dan Kekurangan Aplikasi Berbasis Website

Aplikasi berbasis website memiliki kelebihan dan kekurangan yang perlu dipertimbangkan sebelum memutuskan untuk mengembangkannya. Berikut adalah beberapa kelebihan dan kekurangan dari aplikasi berbasis website:

Kelebihan Aplikasi Berbasis Website:

- Aplikasi web dapat diakses dari berbagai perangkat dengan koneksi internet, termasuk komputer, tablet, dan *smartphone*.
- Aplikasi web tidak memerlukan instalasi pada perangkat pengguna. Hal ini mengurangi beban pengguna karena tidak perlu melakukan pembaruan atau mengelola memori penyimpanan untuk aplikasi.
- Aplikasi web cenderung lebih kompatibel lintas platform, karena hanya memerlukan browser web untuk diakses. Ini mempermudah pengembangan dan distribusi aplikasi kepada pengguna dengan berbagai sistem operasi.
- Aplikasi berbasis web dapat lebih hemat biaya daripada aplikasi berbasis desktop.

Kekurangan Aplikasi Berbasis Website:

- Salah satu kelemahan utama adalah ketergantungan pada koneksi internet yang stabil. Jika pengguna tidak memiliki akses internet atau sambungan lemah, kinerja aplikasi dapat terpengaruh atau bahkan tidak dapat diakses sama sekali.
- Aplikasi web mungkin memiliki kinerja yang lebih rendah daripada aplikasi berbasis desktop karena ketergantungan pada kecepatan internet dan kemampuan server. Aplikasi yang membutuhkan pemrosesan data yang intensif mungkin terasa lambat.

- Aplikasi berbasis website rentan terhadap ancaman keamanan, seperti serangan peretasan dan penyalahgunaan data. Pengembang harus memperhatikan praktik keamanan terbaik untuk melindungi informasi pengguna dan data aplikasi.
- Beberapa fitur yang lebih canggih mungkin sulit diimplementasikan dalam aplikasi web dibandingkan dengan aplikasi desktop yang memiliki akses lebih mendalam ke sumber daya perangkat keras.

Untuk menghasilkan produk dan sistem perangkat lunak yang baik, diperlukan sebuah perancangan dan rencana kerja yang maksimal. Dalam hal ini Penulis menggunakan metode pengembangan software dengan *Rapid Application Development (RAD)*. Dengan RAD durasi proses produksi dapat dipersingkat, mengurangi biaya efektif dalam meningkatkan serta menjaga kualitas produk.

2.3 Model Pengembangan Software dengan Rapid Application Development (RAD)

Menurut (Habibi, dkk., 2019), Rapid Application Development (RAD) yaitu suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak.

Menurut (Riska, dkk., 2021), metode RAD terdiri dari tiga tahapan yang terstruktur dan saling bergantung disetiap tahapannya:

a. Requirement Planning

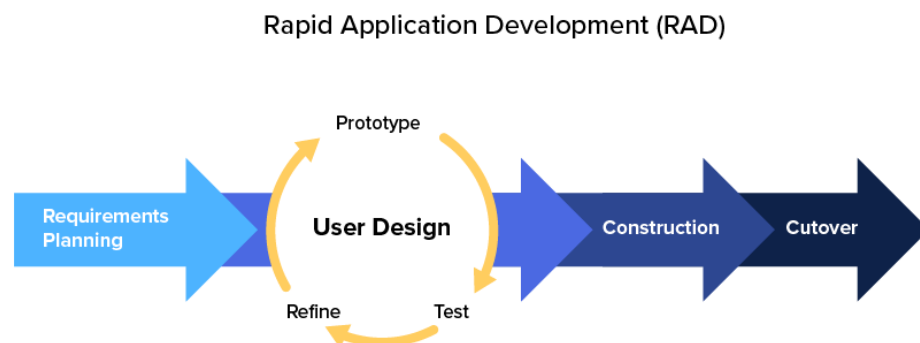
Pada tahap ini, tim pengembang bekerja sama dengan para pemangku kepentingan untuk mengidentifikasi dan mendefinisikan kebutuhan fungsional dan non-fungsional dari perangkat lunak yang akan dikembangkan. Kebutuhan ini harus jelas, terperinci, dan dapat diukur.

b. Design Workshop

Pada tahap ini, tim pengembang bekerja sama dengan para pemangku kepentingan untuk merancang solusi perangkat lunak secara kolaboratif. Workshop ini dapat berlangsung selama beberapa hari, di mana ide-ide dan konsep-konsep dijajaki, dan prototipe awal dibangun.

c. Implementation

Pada tahap ini, tim pengembang mulai mengimplementasikan solusi perangkat lunak sesuai dengan desain yang telah disepakati. Metode RAD menekankan penggunaan alat dan teknik pengembangan yang cepat dan efisien, seperti penggunaan komponen perangkat lunak yang telah ada atau penggunaan generasi kode otomatis.



Gambar 2.2. Metode Pengembangan Software dengan RAD

(Sumber: Bitlabs Blog, 2020)

Dalam melakukan pengembangan sistem informasi, model biasanya digambarkan dalam bentuk fisik dan abstrak. Dalam membentuk sistem aplikasi lebih baik dibutuhkan tahapan-tahapan tertentu yang dapat mendukung model sistem perangkat lunak yang

lebih sistematis dan terstruktur. Model biasanya didasarkan pada notasi grafis, yang hampir selalu didasarkan pada notasi dalam *Unified Modeling Language* (UML). UML adalah bahasa pemodelan visual untuk menjelaskan sebuah sistem perangkat lunak. Yang termasuk UML antara lain Diagram Kelas, *Diagram Activity*, *Diagram Use Case*.

2.3.1 Tools yang Digunakan

Tools dalam pengembangan perangkat lunak berbasis website memiliki berbagai kegunaan yang membantu mempermudah dan meningkatkan efisiensi dalam proses pengembangan. *Tools* ini secara keseluruhan membantu dalam meningkatkan efisiensi, kecepatan, dan kualitas pengembangan perangkat lunak berbasis website, sehingga memudahkan pengembang untuk mencapai tujuan dan kebutuhan proyek dengan lebih baik.

2.3.2 Unified Modelling Language (UML)




Menurut (Rosa & Shalahuddin, 2018), Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML adalah alat yang sangat berguna dalam pengembangan perangkat lunak karena membantu dalam merancang, berkomunikasi, dan menggambarkan sistem secara visual dan terstruktur. Dengan menggunakan UML, pengembang dapat memahami kompleksitas sistem, berkomunikasi dengan anggota tim, dan membangun solusi yang tepat dengan pemahaman yang lebih baik tentang masalah dan solusi yang akan diimplementasikan.

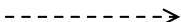

2.3.3 Use Case Diagram

Menurut (Rosa & Shalahuddin, 2018), Use Case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use Case*

Diagram merupakan salah satu jenis diagram dalam Unified Modeling Language (UML) yang berfungsi untuk menyajikan gambaran interaksi antara aktor (entitas yang berinteraksi dengan sistem) dan use case. Use Case Diagram membantu dalam memodelkan fungsionalitas sistem dari perspektif pengguna, menyoroti interaksi antara aktor dan use case dalam skenario berbeda. Adapun simbol-simbol pada use case diagram dapat dilihat pada Tabel 2.1.

Tabel 2.1. Notasi *Use Case Diagram*




No	Notasi	Nama	Deskripsi
1		<i>Actor</i>	Actor/role adalah orang atau sistem lain yang berinteraksi dengan sistem saat ini.
2		<i>Use Case</i>	Use case adalah bagian utama dari fungsionalitas sistem. Bisa extend (memperluas) use case lainnya. Ditempatkan di dalam system boundary (batasan sistem). Dilabeli dengan kata kerja – frase kata benda.
3		<i>Association</i>	Menghubungkan actor dengan use case. Menunjukkan komunikasi dua arah (Menunjukkan komunikasi satu arah jika menggunakan tanda panah). Tanda * untuk keragaman dari asosiasi (multiplicity of the association). Namun umumnya hanya digambarkan garis saja.



5		<i>Include</i>	Memasukkan satu use case dalam use case lainnya. Perilaku (behavior) yang harus terpenuhi agar sebuah event dapat terjadi, di mana kondisi ini sebuah use case adalah bagian dari use case lainnya. Tanda panah mengarah dari base use case (pusat) menuju ke use case yang di-include.
6		<i>Generalization</i>	Mewakili use case khusus untuk use case yang lebih umum. Tanda panah mengarah dari use case khusus (specialized) ke use case yang lebih umum.

2.3.4 Activity Diagram

Menurut (Rosa & Shalahuddin, 2018), Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam activity diagram. Adapun simbol-simbol pada *activity* diagram dapat dilihat pada Tabel 2.2.

Tabel 2. 1 Spesifikasi Activity Diagram

No	Notasi	Nama	Deskripsi
1		<i>Action or Activity</i>	<i>Activity</i> atau aktivitas Merupakan tindakan atau serangkaian tindakan yang dilakukan sistem.
2		Control Flow	<i>Control flow</i> Menunjukkan urutan eksekusi dari proses aktivitas yang terjadi.
3		<i>Initial Node</i>	<i>Initial Node</i> merupakan awal dari serangkaian tindakan yang berlangsung.


4		Final Node	Final Node merupakan akhir atau menghentikan semua aliran dalam suatu aktivitas.
7		<i>Decision Node</i>	<i>Decision node</i> merupakan aktivitas dalam mewakili kondisi pengujian atau percabangan.

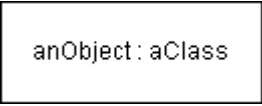


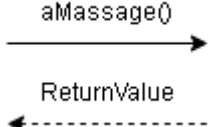
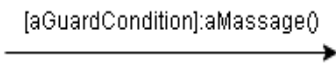

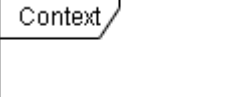
2.3.5 *Sequence Diagram*

Menurut (Rony, 2021), sequence diagram adalah diagram yang merinci dan menampilkan interaksi antar objek dalam suatu sistem. Sequence Diagram digunakan untuk memodelkan dan menggambarkan interaksi antara objek-objek dalam sistem perangkat lunak. Diagram ini memberikan pandangan visual tentang bagaimana pesan atau panggilan metode dikirimkan dari satu objek ke objek lain dalam urutan waktu.

Sequence Diagram membantu dalam menjelaskan alur logika dan interaksi yang terjadi dalam berbagai skenario. Hal ini membantu pengembang untuk memahami dan merencanakan bagaimana objek-objek berinteraksi dan bekerja bersama untuk mencapai tujuan tertentu. Simbol atau spesifikasi yang ada pada *Sequence diagram* dilihat pada Tabel 2.3.

Tabel 2. 2 Spesifikasi *Sequence Diagram*

No	Notasi	Nama	Deskripsi
1	 Actor	<i>Actor</i> (Aktor)	Orang atau sistem yang menerima manfaat atau menggunakan sistem yang berinteraksi dengan sistem informasi yang akan dibuat.

2		Object	Objek yang terlibat dalam sistem, terlibat secara berurutan dengan mengirim atau menerima pesan.
3		<i>Lifeline</i>	<i>Lifeline</i> merupakan garis yang menggambarkan masa hidup dari suatu objek, menunjukkan urutan suatu objek.
4		<i>Execution Occurrence</i>	<i>Execution Occurrence</i> menggambarkan waktu terjadinya pengiriman atau penerimaan pesan
5		<i>Message</i>	Menyampaikan informasi dari satu objek ke objek lainnya. Panggilan operasi diberi simbol dengan pesan yang dikirim dan panah padat. Sedangkan pengembalian diberi simbol sebagai panah putus-putus.
6		<i>Guard Condition</i>	<i>Guard condition</i> merupakan tes yang harus dipenuhi agar suatu pesan dapat dikirimkan.
7		<i>Object Destruction</i>	<i>Object Destruction</i> merupakan gambaran akhir dari <i>lifeline object</i>
8		<i>Frame</i>	<i>Frame</i> menunjukkan konteks dari suatu <i>Sequence diagram</i> .

2.3.6 XAMPP

Menurut (Iqbal, 2019), XAMPP merupakan sebuah *software* web server apache yang didalamnya sudah tersedia database server MySQL dan support PHP programming.

XAMPP memungkinkan untuk mengembangkan dan menguji situs web secara lokal di komputer sebelum mengunggahnya ke server web yang sebenarnya. Ini dapat membantu dalam menghindari potensi masalah dan kerusakan pada situs yang sedang dikembangkan.

2.3.7 MySQL

MySQL merupakan *software database open source* yang sering digunakan untuk mengolah basis data yang menggunakan bahasa SQL (Subagia, 2018: 67).

MySQL adalah sebuah sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*) yang berbasis open-source. MySQL merupakan basis data relasional, yang berarti data disimpan dalam tabel yang terhubung melalui relasi atau kunci-kunci yang diidentifikasi. Pendekatan ini memudahkan untuk menyimpan, mengelola, dan mengakses data dengan cara yang terstruktur dan efisien.

2.4 Sistem Manajemen Kinerja

Perkembangan dan kemajuan suatu organisasi tidak dapat dipungkiri berkaitan erat dengan faktor kualitas manajemen kinerja. Manajemen kinerja yang baik memberi pengaruh dan menjadi kekuatan pendorong yang mampu meningkatkan efisiensi dan transparansi antar staff ataupun manajemen.

Menurut (Rivai & Basri, 2017), kinerja adalah hasil atau tingkat keberhasilan seseorang secara keseluruhan selama periode tertentu dalam melaksanakan tugas dibandingkan dengan berbagai kemungkinan, seperti standar hasil kerja, target atau sasaran maupun kriteria yang telah ditentukan terlebih dahulu telah disepakati bersama.

Sistem Manajemen Kerja memiliki berbagai kegunaan yang membantu organisasi dan tim dalam mengelola pekerjaan, kolaborasi,

dan pencapaian tujuan. Berikut adalah beberapa kegunaan utama dari Sistem Manajemen Kerja:

1. Organisasi dan Pemantauan Tugas

Sistem Manajemen Kerja membantu dalam mengorganisasi tugas-tugas yang perlu diselesaikan oleh individu atau tim. Dengan sistem ini, pengguna dapat membuat tugas, menentukan tenggat waktu, memprioritaskan, dan melacak kemajuan pekerjaan secara terpusat.

2. Kolaborasi dan Komunikasi Tim

Sistem Manajemen Kerja menyediakan fitur-fitur kolaborasi yang memungkinkan anggota tim untuk berinteraksi, berbagi informasi, dan bekerja sama secara efisien. Pengguna dapat memberikan komentar, membagikan file, memberikan tanggapan, atau berkomunikasi melalui fitur pesan.

3. Pengelolaan Proyek

Sistem Manajemen Kerja membantu dalam merencanakan, mengatur, dan melacak proyek secara keseluruhan. Dengan fitur seperti papan cerdas (kanban board), pengguna dapat dengan mudah melihat status tugas, memindahkan tugas dari satu tahap ke tahap berikutnya, dan mengelola aliran kerja proyek secara visual.

4. Penugasan dan Pengalokasian Sumber Daya

Sistem ini memungkinkan manajer atau pemimpin tim untuk menugaskan tugas kepada anggota tim dengan cara yang terstruktur. Mereka juga dapat mengelola pengalokasian sumber daya, mengatur jadwal, dan melihat kapasitas dan ketersediaan anggota tim.

5. Pelaporan dan Analisis

Sistem Manajemen Kerja menyediakan kemampuan untuk menghasilkan laporan dan analisis terkait kinerja individu, tim, atau proyek. Fitur ini membantu manajer dan pemimpin dalam memantau progres, mengidentifikasi hambatan, dan membuat keputusan berdasarkan data yang relevan.

6. Transparansi dan Keterlibatan

Dengan Sistem Manajemen Kerja, semua anggota tim dapat melihat dan memahami tugas-tugas yang perlu diselesaikan, tanggung jawab masing-masing anggota, serta kemajuan pekerjaan. Hal ini meningkatkan transparansi, mempromosikan keterlibatan tim, dan memastikan pemahaman yang jelas tentang tujuan dan harapan yang harus dicapai.

7. Efisiensi dan Produktivitas

Dengan pengelolaan yang terpusat dan fitur-fitur otomatisasi, Sistem Manajemen Kerja membantu meningkatkan efisiensi dan produktivitas. Pengguna dapat menghindari kebingungan, mengurangi kesalahan, dan mengoptimalkan penggunaan waktu dan sumber daya.