

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Sistem

Sistem adalah rangkaian dua atau lebih komponen yang saling berhubungan dan berfungsi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang berfungsi untuk mendukung sistem yang lebih besar. (Jurnal et al., 2019).

2.2 Pengertian Sistem Informasi

Empat komponen utama membentuk sistem informasi: perangkat lunak (*software*), perangkat keras (*hardware*), *infrastruktur* dan sumber daya manusia terlatih (Kurniawan et al., 2019).

Sistem dalam suatu organisasi dikenal sebagai sistem informasi. Sistem ini menggabungkan kebutuhan untuk pemrosesan transaksi sehari-hari dengan kegiatan *operasional* dan *strategis* serta memberikan laporan yang diperlukan kepada pihak *eksternal* (Purwati & Rahardi, 2018).

2.3 Website

Website atau situs dapat diartikan sebagai Kumpulan halaman-halaman yang digunakan untuk menampilkan informasi *teks*, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Berdasarkan pengertian tersebut, dapat dikatakan bahwa *website* adalah sekumpulan halaman-halaman situs yang memuat informasi berupa data *teks*, gambar, suara, animasi, video atau gabungan dari data-data tersebut baik dalam bentuk statis maupun dinamis (Nuryansyah & Ratnawati, 2020).

2.4 *Repository*

Repository adalah karya yang dihasilkan oleh *civitas* akademika berupa laporan teknis, skripsi, tesis, dan bahan ajar. Dari dua sudut pandang ini juga dapat disimpulkan bahwa elemen lain dari layanan *repositori* adalah mudah diakses karena diinstal secara *online* (Rauf & Prastowo, 2021).

2.5 HTML 5

HTML adalah bahasa yang digunakan untuk menciptakan struktur halaman web dan digunakan untuk mempublikasikan dokumen secara online. Tag adalah instruksi dasar HTML. Tag dibuat di antara tanda kurung siku (<>), dan mereka harus dipasang untuk setiap dokumen atau bagian dari dokumen. Ini mencakup tag pembuka dan penutup. Untuk membuat tag penutup, garis miring (/) dimulai dengan nama tag. (Sari et al., 2022).

2.6 PHP 8.1

PHP adalah bahasa skrip sisi *server* yang digunakan bersama HTML untuk membuat halaman web yang terus berubah. Pembuatan *website* adalah gabungan dari PHP itu sendiri sebagai bahasa pemrograman dan HTML sebagai pembuat *website*. Tujuannya adalah agar sintaks dan perintah yang diberikan dijalankan seluruhnya di server, tetapi terdapat di dokumen HTML. Saat pengguna menggunakan skrip sisi *server* PHP untuk membuka halaman web, *server* yang terkait terlebih dahulu memproses format HTML di *browser web* pengguna. (Nur Salim, 2021).

2.7 Laravel 9

Laravel adalah salah satu dari banyak *framework* PHP yang dibangun di atas konsep *Model View Controller* atau MVC. Framework ini digunakan untuk membuat *website* berbasis template MVP atau presenter tampilan yang ditulis dalam bahasa pemrograman PHP.

Tujuan dari Laravel adalah untuk meningkatkan kualitas perangkat lunak, mengurangi biaya pengembangan dan pemeliharaan, dan meningkatkan pengalaman bekerja dengan aplikasi yang memberi (Harahap et al., 2022).

2.8 Javascript

JavaScript adalah bahasa pemrograman sisi klien yang digunakan untuk pemrograman web. Ini mengacu pada *browser web* seperti *Google Chrome*, *Mozilla Firefox*, dan *Opera Mini*, antara lain. JavaScript awalnya dibuat pada pertengahan tahun 1990-an. JavaScript berbeda dari Java, meskipun namanya mirip. JavaScript dapat dimasukkan ke dalam dokumen HTML atau diubah menjadi dokumen terpisah yang dapat ditautkan ke dokumen lain. JavaScript mengaktifkan fitur yang dimaksudkan untuk mengatur cara situs *web* berinteraksi dengan pengguna (Sari et al., 2022).

2.9 JQuery

JQuery adalah library JavaScript yang populer. JQuery dibuat oleh John Resig untuk memudahkan pengembang mengimplementasi JavaScript di dalam aplikasinya. JQuery berfungsi untuk memanipulasi DOM (*Document Object Model*) halaman dengan ringkas dan intuitif sehingga pengembang dapat melakukan proses-proses manipulasi elemen-elemen di dalam halaman dengan mudah. JQuery juga memiliki fungsi lain yaitu AJAX. AJAX adalah singkatan dari *Asynchronous Javascript and XML* dan mengacu pada sekumpulan teknik pengembangan *web* yang memungkinkan aplikasi *web* bekerja secara asinkron (tidak langsung) - dengan menangani setiap permintaan (*request*) ke *server* di latar belakang (Bevidianka & Safitri, 2019).

2.10 Bootstrap

Bootstrap adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. Bisa dikatakan, *bootstrap* adalah template desain *web* dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain *web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar HTML dan CSS (sari et al., 2021).

2.11 MySQL

MySQL mulai digunakan sebagai standard yang resmi pada tahun 1986 oleh ANSI (American Standards Institute) dan pada tahun 1987 oleh ISO (Internasional Organization For Standardization) dan disebut sebagai ASQ-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi (Indera et al., 2022).

2.12 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache *HTTP Server*, *MySQL* database dan penerjemah Bahasa yang ditulis dengan Bahasa pemrograman *PHP* dan *Perl* (Raharjo et al., 2019).

2.13 UML

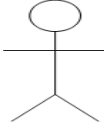
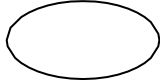

Unified Modelling Language (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem berorientasi object. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lainnya (Marshella Febriani et al., 2022).

Dalam proses merancang sistem menggunakan UML, terdapat beberapa diagram, seperti:


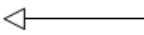

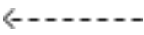

1. *Use Case Diagram*

Diagram use case membuat dan menjelaskan fungsionalitas yang diinginkan dari sistem. *Use case* mewakili deskripsi lengkap dari interaksi yang terjadi antara agen dan sistem atau perangkat lunak yang sedang dikembangkan

Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang dibangun di luar sistem informasi itu sendiri
2		<i>UseCase</i>	Menjelaskan interaksi antar sistem dan <i>actor</i> untuk menampilkan sistem.
3		<i>System</i>	Menjelaskan yang ditampilkan sistem secara tertentu dan terbatas.


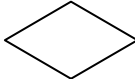



Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)

No	Simbol	Nama	Keterangan
4		<i>Dependency</i>	hubungan di mana elemen yang bergantung pada elemen non-independen.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-spesifik) antara dua kasus penggunaan di mana satu fungsi lebih umum daripada yang lain.
6		<i>Include</i>	Hubungan use case komplementer dengan use case dimana use case yang ditambahkan membutuhkan use case tersebut untuk menjalankan fungsinya.
7		<i>Extend</i>	hubungan antara use case dan use case tambahan, di mana use case tambahan dapat berdiri sendiri
8		<i>Assosiation</i>	Use case atau use case yang berinteraksi dengan aktor melibatkan komunikasi antar aktor.

2. *Activity Diagram*

Diagram aktivitas dibuat dalam bahasa pemodelan terpadu untuk menggambarkan aktivitas komputasi dan aliran aktivitas dalam suatu organisasi. Ini adalah visualisasi alur kerja yang berisi aktivitas dan tindakan, serta pilihan, loop, dan kompetisi.

Tabel 2.2 Simbol *Activity Diagram*

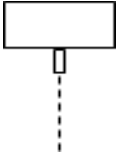
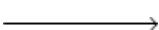
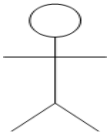
No	Simbol	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana setiap kelas antarmuka berinteraksi satu sama lain.
2		<i>Decision</i>	Untuk pilihan pengambilan keputusan.
3		<i>Initial Node</i>	Awal.
4		<i>Acifity</i> <i>Final Node</i>	Akhir.
5		<i>Fork</i>	Menunjukkan kegiatan yang dilakukan secara paralel atau kombinasi dari dua kegiatan paralel disatu.

3. *Sequence Diagram*

Sequence diagram menunjukkan perilaku objek dalam *use case* dengan menunjukkan siklus hidup mereka dan pesan yang dikirim dan diterima di antara mereka. Interaksi antara objek di dalam dan di sekitar sistem (seperti layar, pengguna, dll.) digambarkan dalam pesan yang dikirim dan diterima secara bertahap. Ini menunjukkan skenario atau rangkaian tindakan yang diambil sebagai tanggapan

atas peristiwa untuk mencapai keluaran tertentu. Diagram kolaborasi, seperti diagram urutan, menunjukkan interaksi antar objek. Mereka tidak berfokus pada menyampaikan pesan, tetapi mereka menunjukkan interaksi antara objek. Nomor menandai pesanan tertinggi.

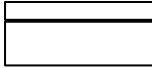
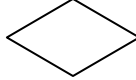



Tabel 2.3 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Object dan lifeline</i>	Semua hal yang ada dan penting untuk aplikasi berinteraksi satu sama lain.
2		<i>Message</i>	memberikan gambaran tentang komunikasi yang terjadi antara objek dan informasi tentang aktivitas yang terjadi.
3		<i>Actor</i>	memberikan penjelasan tentang berbagai tugas yang dilakukan pengguna saat berinteraksi dengan use case.

4. *Class Diagram*

Class diagram atau diagram kelas adalah gambaran struktur sistem dalam hal pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dirancang untuk programmer untuk membuat kelas berdasarkan desain dalam diagram kelas sehingga dokumentasi desain dan perangkat lunak sinkron.

Tabel 2.4 Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Sekumpulan dari objek yang memiliki atribut seroperasi yang sama.
2		<i>Nary Association</i>	Hal untuk menghindari asosiasi atau proses dengan lebih dari 2 objek.
3		<i>Assosiasi</i>	Hubungan antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
4		<i>Generalization</i>	Hubungan antara generalisasi dan spesialisasi (umum-khusus) dalam dua buah use case di mana fungsi yang satu lebih umum daripada yang lainnya
5		<i>Dependency</i>	Hubungan di mana perubahan ke elemen independen memengaruhi ketergantungan elemen non-independen.

2.14 *Black Box Testing*

Uji perangkat lunak sebagai spesifikasi fungsional tanpa memeriksa desain dan kode program untuk memastikan fungsi, input, dan output perangkat lunak memenuhi persyaratan. Pengujian kotak hitam mudah digunakan karena hanya membutuhkan batas atas dan bawah dari data yang diharapkan. Perkiraan jumlah data uji yang dapat dihitung bergantung pada jumlah bidang *input* yang akan diuji, aturan input yang harus diikuti, dan apakah batas atas dan bawah terpenuhi. Dengan cara

ini, kita dapat mengetahui apakah fungsi dapat menerima input yang tidak diinginkan, yang menyebabkan data yang disimpan kurang *efisien* (Nur Cholifah & Melati Sagita, 2018).

2.15 Referensi Penelitian

Tabel 2.5 Referensi Penelitian

No	Judul Penelitian	Nama Penelitian	Hasil
1	Rancang Bangun Aplikasi Berbasis Web Sistem Informasi Repository Laporan PKL Siswa (Studi Kasus: SMKN 1 Terbanggi Besar) Metode <i>Web Engineering</i>	Abdur Rauf JH dan Agung Tri Prastowo (2021)	Perancangan sistem informasi pengarsipan laporan PKL di SMKN 1 Terbanggi Besar dari metode pengumpulan data (wawancara, observasi dan dokumentasi) menggunakan metode pengembangan rekayasa web untuk merancang sistem menggunakan UML dan mengimplementasikan sistem menggunakan PHP (Premium Text) dan MySQL.

Tabel 2.5 Referensi Penelitian (Lanjutan)

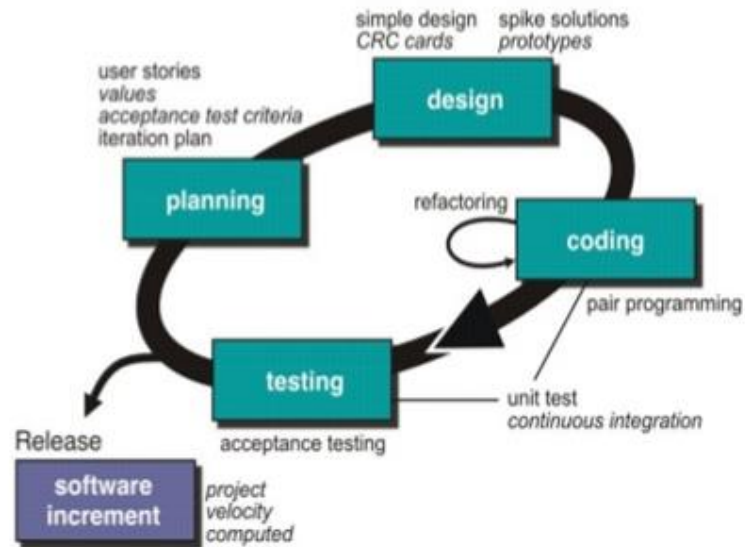
No	Judul Penelitian	Nama Penelitian	Hasil
2	Perancangan Prototype Sistem Informasi Repository Skripsi Berbasis Web Di UNA'IM Yapis Wamena Papua Metode Simulated Prototyping	Syarifah, Chairullah Naury dan Wahyuni Nurindah Sulistiyowati (2022)	Simulasi prototyping dapat digunakan untuk memodelkan sistem informasi repositori disertasi berbasis web. Ini terutama berlaku untuk membuat model layar input dan output. Diagram konteks, aliran data, dan diagram hubungan entitas adalah dokumen hasil perancangan dan analisis sistem informasi yang selalu digunakan dalam perancangan prototipe
3	Perancangan aplikasi Repository Skripsi Universitas Amir Hamzah Berbasis Web Metode Simulated Prototyping	Muhammad Aditya dan Surya Hendra Pura (2022)	Diagram konteks, diagram aliran data, dan diagram hubungan entitas adalah komponen penting dari desain sistem informasi.

Tabel 2.5 Referensi Penelitian (Lanjutan)

No	Judul Penelitian	Nama Penelitian	Hasil
4	Rancang Bangun Sistem Informasi Arsip dan Persuratan Program Studi Teknik Elektro Untirta Berbasis Framework Laravel Metode Framework Laravel	Masjudin, Cakra Adipura Wicaksana Agusutrisno dan Rian Fahrizal (2022)	Sistem informasi telah berhasil dibuat dan digunakan. Berkat sistem informasi ini, program penelitian akan dengan mudah menangani aplikasi dan dokumentasi surat dan dokumen lain yang diperlukan untuk pelaksanaan program penelitian.
5	Implementasi Sistem Informasi <i>repository</i> tugas akhir pada amik mahaputra riau berbasis web Metode SDLC (<i>System Development Life Cycle</i>).	Darmanta, Sukrianto dan Sinta Maria (2022)	Di AMIK Mahaputra Riau, sistem informasi repository berbasis web ini akan membantu menyelesaikan masalah penumpukan perpustakaan dan meningkatkan kualitas kearsipan tugas akhir.

3.12 Metode Extreme Programming

Extreme Programming (XP)



Gambar 2.1 *Extreme Programming*

Paradigma *Extreme Programming* (XP) adalah pendekatan rekayasa perangkat lunak yang digunakan untuk mengembangkan sistem. Pendekatan ini adalah pendekatan berorientasi objek dan menggunakan pendekatan *Extreme Programming* sebagai model pengembangan. Aturan dan praktik ini terjadi dalam empat kerangka kerja operasional: perencanaan, perancangan, pengkodean, dan pengujian. (Alfaridzy & Seprina, n.d, 2021).