

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian terdahulu yang dilakukan oleh Mannang menunjukkan bahwa sistem informasi berbasis *web* dapat mendukung proses pemantauan dan pengelolaan data transaksi secara terstruktur, sehingga memfasilitasi pencatatan, pengolahan, dan penyajian informasi yang diperlukan oleh pihak terkait. Sistem informasi berbasis *online* memungkinkan integrasi data dan akses yang mudah bagi berbagai peran pengguna dalam suatu organisasi [7].

Pengembangan sistem informasi berbasis *web* pada bidang pengelolaan administrasi dan sumber daya manusia telah banyak dilakukan untuk meningkatkan kualitas layanan publik sebagaimana dikaji oleh Alfidyah [8], serta Nazhira dan Candradewini [9]. Sistem tersebut berperan penting dalam membantu pengelolaan data operasional, pendokumentasian aktivitas pendaftaran, serta penyusunan laporan statistik yang dibutuhkan oleh instansi pemerintah. Lebih lanjut, hasil penelitian Satria, Ramadhani, dan Sari menunjukkan bahwa penerapan sistem informasi dalam proses rekrutmen atau penerimaan peserta magang dapat meningkatkan efektivitas kerja [10].

Penelitian lain dari Wibowo dan Setiawan mengungkapkan bahwa penggunaan *framework* dalam pengembangan *website* dapat mempercepat proses pembangunan sistem, meningkatkan keamanan, serta mempermudah pengelolaan modul dan hak akses pengguna. Sistem berbasis *website* yang

dibangun dengan *framework* modern mampu mendukung pengelolaan data yang kompleks dan terstruktur [11].

Terkait pengelolaan peserta, Puspita, Alkhalifi, dan Basri menjelaskan bahwa penerapan sistem informasi manajemen pelayanan magang memfasilitasi proses registrasi, seleksi, pelaporan, penilaian, dan pemberian sertifikat magang, serta meningkatkan pengelolaan administratif. Sistem ini efektif membantu mahasiswa dalam melakukan registrasi secara daring dan memperoleh informasi mengenai ketersediaan tempat magang [12].

Selain itu, Chandra menyatakan bahwa sistem informasi absensi dapat dirancang dengan memanfaatkan teknologi RFID untuk meningkatkan efisiensi proses administrasi absensi. Penelitian ini bertujuan merancang sistem informasi absensi guna meningkatkan kecepatan dan akurasi pencatatan administrasi absensi [13].

2.2 Landasan Teori

2.2.1 Sistem Informasi Berbasis *Web*

Sistem informasi merupakan suatu sistem yang digunakan untuk mengumpulkan, mengolah, menyimpan, dan menyajikan data menjadi informasi yang bermanfaat untuk mendukung kegiatan operasional dan pengambilan keputusan [14]. Sistem informasi berbasis *web* memanfaatkan teknologi internet sebagai media utama sehingga sistem dapat diakses melalui *browser* tanpa bergantung pada perangkat tertentu [15].

Penerapan sistem informasi berbasis *web* memungkinkan perusahaan mengelola data secara terpusat, meningkatkan efisiensi pengolahan data, mempermudah *monitoring* aktivitas, serta mendukung penyusunan laporan secara cepat dan akurat. Sistem ini sangat sesuai diterapkan pada instansi pemerintahan.

2.2.2 *Website*

Website atau situs *web* adalah kumpulan halaman *web* yang saling terkait dan dapat diakses secara publik melalui jaringan *internet* menggunakan sebuah alamat *domain* atau URL (*Uniform Resource Locator*) [16]. Halaman-halaman pada *website* umumnya dibangun menggunakan bahasa markah seperti HTML dan dapat berisi berbagai macam informasi dalam bentuk teks, gambar, maupun media interaktif lainnya [17].

Dalam konteks penelitian ini, *website* berfungsi sebagai wadah atau antarmuka utama tempat berjalannya sistem manajemen magang, sehingga seluruh proses administrasi dan *monitoring* dapat diakses secara fleksibel dari berbagai perangkat.

2.2.3 *Presensi dan RFID (Radio Frequency Identification)*

Presensi merupakan data kehadiran serta pelaporan aktivitas yang disusun untuk mempermudah pemantauan kedisiplinan [18]. Dalam sistem ini, presensi tidak lagi dilakukan secara manual, melainkan terintegrasi dengan teknologi RFID. RFID adalah teknologi identifikasi menggunakan gelombang radio yang terdiri dari

Tag (kartu) dan Reader (pembaca) [19]. Integrasi RFID ke dalam *website* memungkinkan pencatatan jam masuk dan keluar peserta magang dilakukan secara otomatis, sehingga menghasilkan laporan rekapitulasi kehadiran yang akurat, minim manipulasi, dan sistematis [20].

Penerapan teknologi ini sangat efektif untuk menggantikan sistem absensi tanda tangan konvensional yang memiliki risiko manipulasi tinggi. Dengan adanya integrasi antara perangkat keras RFID dan sistem informasi berbasis *web*, setiap data kartu yang terbaca akan dikirimkan secara langsung ke basis data *server*, sehingga staf pembimbing di DINKOMINFOTIK dapat melakukan pemantauan kehadiran peserta magang secara *real-time* dan transparan.

2.2.4 Structured Query Language (MySQL)

Database adalah kumpulan data yang tersimpan secara terstruktur dan saling berhubungan, yang digunakan untuk mendukung proses penyimpanan, pengolahan, dan pengambilan data dalam suatu sistem informasi. *Database* berfungsi menjaga konsistensi, keakuratan, keamanan, serta ketersediaan data[21].

Laragon merupakan lingkungan pengembangan *web* lokal yang menyediakan paket server seperti Apache, PHP, dan MySQL/MariaDB dalam satu *platform*. Laragon digunakan untuk membantu proses pengembangan dan pengujian sistem informasi berbasis *web* sebelum diimplementasikan pada *server* [22]. Dengan

Laragon, pengembang dapat mengelola *database*, menjalankan aplikasi, dan melakukan pengujian sistem secara lokal.



Gambar 2. 1 MySQL

2.2.5 Framework Laravel

Laravel merupakan *framework* PHP yang digunakan untuk membangun sistem informasi berbasis *web* dengan konsep *Model-View-Controller* (MVC). Laravel menyediakan berbagai fitur pendukung seperti manajemen *database*, autentikasi pengguna, pengelolaan hak akses, serta mekanisme keamanan aplikasi[23].

Penggunaan Laravel membantu proses pengembangan sistem menjadi lebih terstruktur, mudah dikembangkan, dan lebih aman [24]. *Framework* ini juga mendukung pembuatan modul sistem, pengelolaan *multi-user*, serta integrasi berbagai fitur yang dibutuhkan dalam sistem informasi instansi.



Gambar 2. 2 Framework Laravel

2.2.6 *Application Programming Interface (API)*

Application Programming Interface (API) merupakan mekanisme yang memungkinkan komunikasi antar sistem perangkat lunak yang berbeda [25]. API digunakan sebagai penghubung dalam proses pengiriman dan penerimaan data sehingga berbagai sistem dapat saling berinteraksi [26]. Penggunaan API sangat penting dalam arsitektur sistem berbasis *Internet of Things (IoT)* [27] karena memungkinkan transmisi data dari perangkat keras ke sistem *back-end* secara otomatis. Dengan adanya API, proses integrasi antar sistem menjadi lebih terstruktur dan efisien. Dalam penelitian ini, API berfungsi sebagai jalur komunikasi utama untuk menerima data masukan dari perangkat keras IoT (seperti pemindaian kartu RFID), lalu meneruskannya ke dalam sistem informasi berbasis *framework* Laravel untuk diproses dan disimpan ke dalam basis data.

2.2.7 *Hypertext Preprocessor (PHP)*

PHP adalah bahasa pemrograman yang dieksekusi di *server*. Fungsi utama PHP dalam pengembangan situs adalah untuk mengelola informasi dalam kumpulan data [28]. PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dipahami oleh komputer, bersifat *server-side*, dan dapat disisipkan ke dalam HTML[29].



Gambar 2. 3 PHP

2.2.8 *Hypertext Markup Language (HTML)*

HTML (*HyperText Markup Language*) adalah bahasa standar untuk mendesain struktur halaman *web* [30]. HTML merupakan dasar fundamental dari sebuah *situs online*, karena semua elemen seperti teks, gambar, tautan, tabel, dan formulir dibangun menggunakan *tag* HTML [31].

Dalam pengembangan sistem informasi manajemen magang ini, HTML digunakan sebagai kerangka dasar antarmuka pengguna yang nantinya akan diintegrasikan dengan sistem.



Gambar 2. 4 HTML

2.2.9 *Cascading Style Sheets (CSS)*

CSS (*Cascading Style Sheets*) merupakan bahasa desain yang digunakan untuk mengatur tampilan visual dan tata letak sebuah halaman *web* [32]. CSS berfungsi untuk mengontrol elemen antarmuka seperti komposisi warna, jenis huruf (tipografi), latar

belakang, serta penempatan tata letak elemen agar tampilan *website* menjadi lebih menarik, terstruktur, dan mudah dibaca [33]. Penggunaan CSS memungkinkan sistem informasi memiliki bahasa desain yang konsisten di seluruh halamannya, serta berperan penting dalam meningkatkan kenyamanan pengguna saat mengakses informasi administrasi magang.



Gambar 2. 5 CSS

2.2.10 Tailwind CSS

Tailwind CSS adalah kerangka kerja (*framework*) CSS berbasis *utility-first* yang memungkinkan pengembang untuk membangun antarmuka pengguna secara cepat dan kustom langsung di dalam berkas HTML [34], [35]. Berbeda dengan *framework* CSS tradisional yang menyediakan komponen desain siap pakai, Tailwind menyediakan kelas-kelas utilitas tingkat rendah (*low-level utility classes*) yang dapat dikombinasikan untuk membentuk desain yang spesifik. Pada penelitian ini, Tailwind CSS digunakan untuk mempercepat proses desain antarmuka *web* agar tampil modern, bersih, dan responsif [36].



Gambar 2. 6 Tailwind CSS

2.2.11 JavaScript

JavaScript adalah bahasa pemrograman yang berjalan di sisi klien (*client-side*) pada *browser* untuk menciptakan halaman *web* yang interaktif dan dinamis [37]. Pada sistem absensi terintegrasi ini, JavaScript sangat diperlukan untuk menangani pemrosesan data secara *real-time* di sisi antarmuka, seperti melakukan validasi formulir dan memperbarui tampilan data kehadiran secara otomatis saat RFID dipindai tanpa perlu memuat ulang keseluruhan halaman *web*.



Gambar 2. 7 JavaScript

2.2.12 Visual Studio Code

Aplikasi pengedit kode sumber terbuka yang dikembangkan oleh Microsoft untuk *platform* Windows, Linux, dan MacOS [38]. Visual Studio Code memfasilitasi penulisan kode untuk berbagai bahasa pemrograman, seperti C++, C#, Java, Python, PHP, dan HTML. Aplikasi ini dapat mengidentifikasi jenis bahasa pemrograman yang digunakan dan menawarkan fitur pelengkapan

otomatis (*auto-complete*) sesuai dengan kemampuan eksekusi kode [39]. Selain itu, *code editor* ini juga telah terintegrasi dengan GitHub dan memiliki fitur untuk menyertakan ekstensi tambahan yang sangat berguna bagi pengembang.



Gambar 2. 8 Visual Studio Code

2.2.13 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk membuat diagram atau model perangkat lunak [40]. UML memungkinkan pengembang atau *developer* untuk *memvisualisasikan*, *mendefinisikan*, *membangun*, dan *mendokumentasikan* sistem perangkat lunak secara jelas dan terstruktur [41]. Analogi yang sering digunakan adalah seperti seorang arsitek bangunan yang membuat denah sebagai acuan bagi perusahaan konstruksi; demikian pula seorang *arsitek* perangkat lunak membuat diagram UML agar pengembang dapat membangun sistem sesuai dengan rancangan yang telah ditetapkan. Dengan memahami notasi dan kosakata UML, seseorang dapat lebih mudah memahami alur dan struktur sistem, serta menjelaskan desain perangkat lunak kepada pihak lain yang terlibat dalam pengembangan.

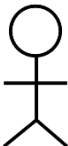
Unified Modeling Language sendiri termasuk salah satu metode pemodelan visual yang banyak digunakan dalam perancangan perangkat lunak berbasis objek, yang berfungsi sebagai norma penulisan atau semacam cetak biru, yang mencakup proses bisnis, penulisan kelas, dan berbagai elemen sistem lainnya dalam bahasa yang khusus. Beragam diagram sering digunakan dalam pengembangan sistem untuk menggambarkan struktur, perilaku, dan interaksi antar elemen sistem.

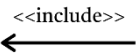
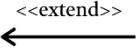
1. Use Case Diagram

Menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dan merepresentasikan sebuah interaksi antara aktor dan sistem.

Didalam *Use Case Diagram* terdapat (6 enam aktor) yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Tabel 2. 1 Use Case Diagram

No	Gambar	Simbol	Keterangan
1.		<i>Actor</i>	Mewakili fungsi sistem lain atau alat dalam berkomunikasi dengan <i>Use Case</i> .
2.		<i>Use Case</i>	Mewakiili Abstraksi dan interaksi antara sebuah <i>System</i> dan <i>Actor</i> .

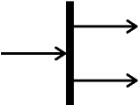
No	Gambar	Simbol	Keterangan
3.		<i>Association</i>	Abstraksi penghubung antar <i>Actor</i> dengan <i>Use Case</i> .
4.		<i>Generalisasi</i>	Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>Use Case</i> .
5.		<i>Include</i>	Menunjukkan bahwa suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya.
6.		<i>Extend</i>	Menunjukkan bahwa suatu <i>Use Case</i> tambahan fungsional dari <i>Use Case</i> lainnya.

2. Activity Diagram

Menggambarkan alur kerja dari suatu proses dan urutan aktivitas, yang dimodelkan sebagai alur kerja dari satu aktivitas ke aktivitas berikutnya atau dari satu aktivitas ke aktivitas lainnya.

Di dalam *Activity Diagram* terdapat beberapa simbol yang digunakan untuk menggambarkan alur aktivitas dan proses kerja dalam sistem.

Tabel 2. 2 Activity Diagram

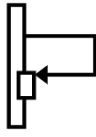
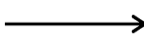
No	Gambar	Simbol	Keterangan
1.		<i>Initial node</i>	Merupakan awal aktivitas yang akan dibuat
2.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi
3.		<i>Decicion</i>	Menjelaskan suatu keputusan atau tindakan yang perlu diambil dalam situasi tertentu.
5.		<i>Fork Join node</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel, menggabungkan dua kegiatan paralel menjadi satu.
6.		<i>Fork node</i>	Sebuah aliran pada tahap tertentu bertransformasi menjadi beberapa aliran.
7.		<i>Activity Final node</i>	Menandakan selesainya aktivitas yang telah dilaksanakan.

3. Sequence Diagram

Menggambarkan interaksi antar objek didalam dan disekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

Di dalam *Sequence Diagram* terdapat beberapa simbol yang digunakan untuk alur komunikasi pesan antar objek dalam sistem.

Tabel 2. 3 Sequence Diagram

No	Gambar	Simbol	Keterangan
1.		<i>Entity Class</i>	Menggambarkan awal dari sebuah sistem, menjadi dasar penyusunan basisdata.
2.		<i>Boundary Class</i>	Menggambarkan komunikasi antara sistem seperti <i>Form</i> .
3.		<i>Control Class</i>	Penghubung antara <i>boundary</i> dengan <i>table</i> .
4.		<i>Recursive</i>	Mengirimkan pesan yang akan dikirim ke dirinya sendiri
5.		<i>Activation</i>	Menghubungkan proses durasi aktivasi sebuah proses
6.		<i>Life Line</i>	Menghubungkan tiap-tiap objek yang mana terdapat <i>activation</i>
7.		<i>Message</i>	Sebagai pengirim pesan ke tiap-tiap <i>Class</i>

4. Class Diagram

Mengambaran struktur dan deskripsi dari *class*, *package* dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Di dalam *Class Diagram* terdapat beberapa simbol yang digunakan untuk hubungan dan struktur data dalam sistem.

Tabel 2. 4 Class Diagram

No	Gambar	Simbol	Keterangan
1.		<i>Class</i>	Menjelaskan sekumpulan objek dengan atribut dan operasi yang serupa
2.		<i>Nary Association</i>	Untuk menghindari asosiasi dengan lebih dari dua objek
3.		<i>Collaboration</i>	Urutan aksi-aksi yang ditampilkan oleh sistem yang menghasilkan suatu hasil yang tertukar bagi suatu <i>actor</i>
4.		<i>Generalization</i>	Menggambarkan dimana hubungan objek <i>descendent</i> berperilaku dan struktur data objek yang di atasnya objek induk <i>ancestor</i>
5.		<i>Realization</i>	Operasi yang secara nyata dilaksanakan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan di mana perubahan pada suatu elemen mandiri akan memengaruhi elemen yang tidak mandiri.