

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini menguraikan teori-teori yang mendasari dalam pengembangan aplikasi. Berikut akan dijelaskan tentang teori-teori yang berhubungan dengan skripsi yang dibuat.

#### **2.1 Pengembangan**

Pengembangan adalah proses yang dilakukan untuk meningkatkan suatu sistem menjadi lebih baik, lebih efektif, dan lebih maju. Dalam konteks pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, pengembangan mengacu pada langkah-langkah yang diambil untuk merancang, membangun, mengimplementasikan, dan memelihara sistem informasi surat berbasis *web* yang dapat memenuhi kebutuhan dan tujuan universitas (Rafidah, 2019).

#### **2.2 Sistem Informasi**

Sistem informasi adalah kumpulan elemen-elemen yang berinteraksi dan terorganisasi secara bersama-sama untuk mengumpulkan, menyimpan, memproses, mengirimkan, dan mengelola data serta informasi dengan tujuan mendukung pengambilan keputusan, pengendalian, koordinasi, dan operasi suatu organisasi dan entitas (O'Brien, 2018).

Dalam konteks pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, sistem informasi merujuk pada keseluruhan sistem yang dirancang dan diimplementasikan untuk mengelola informasi terkait surat, baik surat masuk maupun surat keluar. Sistem informasi ini melibatkan pengguna, perangkat keras, perangkat lunak, basis data, prosedur, dan kebijakan yang mendukung proses pengelolaan surat secara efisien dan efektif.

### 2.3 Surat

Surat adalah dokumen tertulis yang berfungsi sebagai media komunikasi untuk menyampaikan informasi, pesan, pernyataan antara pihak-pihak tertentu, baik individu maupun organisasi (Bakri, 2022). Isi surat biasanya tertulis dalam bentuk tulisan pada kertas dan digunakan untuk berbagai tujuan seperti bisnis, akademik, pribadi, dan resmi. Dalam pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi. Surat merujuk pada jenis surat yang diterima dan dikirim oleh universitas dari pihak eksternal, seperti institusi eksternal, lembaga pemerintah, dan individu. Surat mencakup berbagai jenis dokumen tertulis, seperti surat resmi, pemberitahuan, undangan, dan permohonan, yang ditujukan kepada universitas.

### 2.4 Web

*World Wide Web* adalah sistem yang terdiri dari halaman-halaman elektronik yang saling terhubung melalui tautan *hypertext* yang merujuk pada salah satu layanan internet (Laily, 2022). Pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, *web* berfungsi sebagai platform utama yang menyajikan, mengakses, dan mengelola informasi surat. Sistem ini memanfaatkan teknologi *web* untuk menyajikan antarmuka pengguna melalui *browser web*, memungkinkan pengguna untuk mengirim, menerima, mencari, dan mengelola surat secara *online* dari berbagai lokasi dan kapan saja (Krishna Putra et al., 2020).

### 2.5 Framework Codeigniter

*Framework CodeIgniter* (CI) merupakan kerangka kerja dalam pengembangan aplikasi *web* yang berdasarkan konsep *Model View Controller* (MVC) (Adani, 2020). MVC memisahkan logika bisnis, tampilan, dan kontrol dalam sebuah aplikasi. Penggunaan *CodeIgniter* dalam pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi memiliki alasan yang kuat karena *framework codeigniter* menyediakan struktur yang terorganisir

dan konsisten, serta fitur-fitur yang mempercepat pengembangan aplikasi *web*. Dengan menggunakan CI, pengembang dapat dengan mudah memisahkan komponen-komponen dalam sistem informasi surat, yang memungkinkan pemeliharaan dan pengembangan yang lebih mudah di masa depan .

Konsep *Model View Controller* (MVC) dalam *framework CodeIgniter* (CI) adalah pola desain yang memisahkan komponen-komponen utama dalam pengembangan aplikasi *web* (Rasuliano Laberto Kelen et al., 2018). Berikut adalah penjelasan tentang setiap komponen dalam konsep MVC *CodeIgniter*:

**a. Model**

*Model* bertanggung jawab atas manajemen data dan logika bisnis dalam aplikasi. *Model* berinteraksi dengan *database* untuk melakukan operasi seperti pengambilan, pembaruan, dan penghapusan data yang dibutuhkan (Rasuliano Laberto Kelen et al., 2018) Selain itu, *Model* juga bertanggung jawab atas validasi data dan menyediakan fungsionalitas bisnis khusus.

**b. View**

*View* adalah tampilan antarmuka pengguna yang menampilkan informasi kepada pengguna akhir (Baguswap, 2022). Dalam *CodeIgniter*, tampilan umumnya terdiri dari *file HTML* yang mengandung *sintaks PHP* untuk memproses data yang dikirim oleh *Controller* (Hidayat et al., 2019). Tugas *View* adalah mengatur tampilan yang menarik dan mudah dipahami bagi pengguna (Komputindo, 2018).

**c. Controller**

*Controller* berperan sebagai penghubung antara *Model* dan *View*. Ketika pengguna berinteraksi dengan aplikasi, *Controller* menerima permintaan dan memprosesnya (Simaremare & Andi, 2017). Ini mencakup memanggil fungsi-fungsi pada *Model* untuk mengambil data, dan memilih *View* yang sesuai untuk menampilkan hasil kepada pengguna. *Controller*

juga mengatur logika bisnis tambahan yang diperlukan sebelum dan setelah pemrosesan data (I. Muhammad & Yosef, 2018).

## **2.6 Universitas Nasional Karangturi**

Universitas Nasional Karangturi adalah salah satu universitas swasta di Semarang, Jawa Tengah yang mengusung nilai nasionalisme Universitas Nasional Karangturi hadir untuk memberikan solusi melalui program-program pendidikan terkini untuk mencetak generasi muda Indonesia yang berkarakter (Karangturi, 2021).

Di bawah naungan Yayasan Pendidikan Nasional Karangturi, UNKARTUR berdiri sejak 19 juni 2017. Dengan bekal pengalaman selama lebih dari 90 tahun Yayasan Pendidikan Nasional Karangturi dan dukungan utama alumni, UNKARTUR menawarkan akses luas untuk kesempatan akademik (Karangturi, 2021).

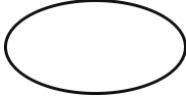



## **2.7 Unified Modeling Language**


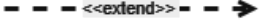
*Unified Modeling Language* (UML) adalah sebuah bahasa dan notasi yang digunakan untuk memodelkan dan mendokumentasikan sistem perangkat lunak (Mulyani, 2018). Dalam konteks pengembangan sistem informasi surat berbasis *web*, UML memberikan alat yang kuat untuk memvisualisasikan struktur, interaksi, dan perilaku komponen-komponen dalam sistem. Penggunaan UML penting karena dapat memberikan pemahaman yang jelas tentang desain sistem informasi surat, memfasilitasi komunikasi antara pengembang dan pemangku kepentingan, serta meminimalkan kesalahan dalam perancangan sistem (Nasution & Mulyono, 2019). Dengan UML, pengembang dapat merencanakan dan merancang sistem informasi surat yang sesuai dengan kebutuhan Universitas Nasional Karangturi dengan efektif. Berikut beberapa jenis *diagram* UML yang digunakan dalam sistem informasi surat:

**a. Use Case Diagram**

*Use Case Diagram* digunakan untuk menggambarkan interaksi antara aktor dan sistem (Andy, 2018). *Use Case Diagram* mengidentifikasi fitur-fitur utama yang akan diimplementasikan dalam sistem dan menggambarkan hubungan antara aktor dan *use case* (Rifandianto et al., 2023). *Use Case Diagram* membantu memahami kebutuhan fungsional sistem dan memberikan gambaran tingkat tinggi tentang interaksi antara pengguna dan sistem (Muhammad & Yosef, 2018). Kesimpulan simbol dan fungsi *use case diagram* dapat dilihat dari tabel 2.1

Tabel 2. 1 Simbol keterangan *use case*

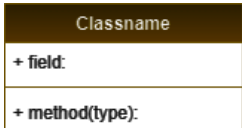
Simbol	Keterangan
<i>Use Case</i> 	Aktivitas yang dijalankan oleh sistem, mewakili interaksi antara aktor dan sistem.
Aktor 	Representasi eksternal yang berinteraksi dengan sistem. Aktor dapat berupa pengguna,
 <i>Association</i>	Garis lurus yang menghubungkan aktor dengan <i>use case</i> , menunjukkan hubungan antara aktor dan <i>use case</i> yang terlibat dalam interaksi.
 <i>Generalization</i>	Garis panah yang menghubungkan aktor satu dengan yang lain, menunjukkan hubungan <i>hirarkis</i> antara aktor. Aktor yang lebih spesifik mewarisi atribut dan perilaku dari aktor yang lebih umum.





 <i>Include (Inclusion)</i>	Menghubungkan satu <i>use case</i> dengan <i>use case</i> lainnya, menunjukkan bahwa <i>use case</i> yang satu mencakup <i>use case</i> lainnya.
 <i>Extend (Ekstensi)</i>	Garis dengan tanda panah yang menghubungkan satu <i>use case</i> dengan <i>use case</i> lainnya, menunjukkan bahwa <i>use case</i> yang satu dapat memperluas <i>use case</i> lainnya dengan fitur tambahan.

## b. *Class Diagram*

*Class diagram* menggambarkan struktur *class* dalam sistem perangkat lunak. *Class diagram* menunjukkan atribut-atribut *class*, metode-metode yang dapat diakses, hubungan antara kelas, dan hubungan antara objek-objek (Muhammad & Yosef, 2018). *Class diagram* membantu dalam merancang struktur kelas yang solid dan memahami relasi antar *class* dalam sistem. Kesimpulan simbol dan fungsi *class diagram* dapat dilihat dari tabel 2.2.

Tabel 2. 2 Simbol keterangan *class diagram*

Simbol	Keterangan
<p><i>Class</i></p> 	<i>Template</i> yang mendefinisikan atribut dan perilaku objek dalam sistem.





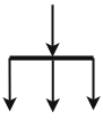
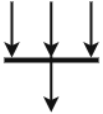

<i>Association</i> 	Garis lurus yang menghubungkan dua kelas, menunjukkan hubungan yang terjadi antara objek kelas dalam sistem.
<i>Generalization</i> 	Garis panah yang menghubungkan kelas satu dengan yang lain, menunjukkan hubungan hierarkis antara kelas.
<i>Aggregation</i> 	Menghubungkan kelas satu dengan yang lain, menunjukkan hubungan "berisi" antara kelas. Hubungan ini menunjukkan bahwa satu kelas merupakan bagian dari kelas lainnya.
Dependensi 	Garis putus-putus yang menghubungkan satu kelas dengan kelas lainnya, menunjukkan ketergantungan antara kelas-kelas tersebut.

### c. *Activity Diagram*

*Activity diagram* digunakan untuk menggambarkan aliran kontrol dalam suatu aktivitas. *Activity diagram* menggunakan notasi seperti tindakan, keputusan dan garis-garis aliran untuk menggambarkan urutan langkah-langkah dalam proses (Muhammad & Yosef, 2018). *Activity diagram* membantu memvisualisasikan alur kerja sistem dan memberikan gambaran yang jelas tentang aktivitas yang terjadi (Rifandianto et al., 2023). Kesimpulan simbol dan fungsi *activity diagram* dapat dilihat dari tabel 2.3.

Tabel 2. 3 Simbol keterangan *activity diagram*

Simbol	Keterangan
--------	------------


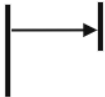
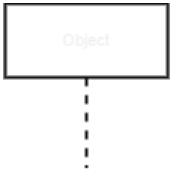

<p><i>Initial Node</i></p> 	<p>Simbol Start (<i>Initial Node</i>) Menunjukkan titik awal dari alur aktivitas. Mewakili titik mulai suatu proses.</p>
<p><i>Line Connector</i></p> 	<p>Garis Penghubung : Menyambungkan <i>node-node</i> aktivitas dalam <i>diagram</i> untuk menggambarkan urutan aliran tindakan.</p>
<p><i>Activity</i></p> 	<p>Melambangkan tindakan dalam <i>diagram</i>. <i>Node</i> ini dapat berupa aktivitas (<i>activity</i>), keputusan (<i>decision</i>), garis penghubung (<i>connector</i>), dan awal/akhir (<i>start/end</i>).</p>
<p><i>Decision</i></p> 	<p>Garis Penghubung Keputusan (<i>Decision</i>): menggambarkan kondisi dalam alur proses. Berguna untuk mengambil keputusan dan menentukan jalur aliran yang berbeda sesuai dengan kondisi tertentu.</p>
 <p><i>Fork</i></p>	<p><i>Fork</i> Digunakan untuk menggambarkan pencabangan dari satu jalur aliran menjadi beberapa alur aliran.</p>
 <p><i>Join</i></p>	<p>Menggambarkan alur aktivitas bercabang akan bergabung kembali menjadi satu alur tunggal untuk melanjutkan alur proses yang sama.</p>
<p><i>Final Node</i></p> 	<p>Simbol <i>End</i> Menunjukkan titik akhir dari alur aktivitas. Mewakili terminasi akhir dari suatu proses.</p>



#### d. *Sequence Diagram*

*Sequence Diagram* adalah jenis *diagram* interaksi yang digunakan untuk memvisualisasikan interaksi antara objek-objek dalam sistem secara berurutan (Kurniawan, 2018). *Sequence diagram* menunjukkan pesan metode yang dikirim antara objek-objek dalam proses tertentu, sehingga membantu dalam pemahaman bagaimana komponen-komponen sistem berinteraksi satu sama lain (Destriana, Husain, Handayani, & Prahara Siswanto, 2021). Kesimpulan simbol dan fungsi *sequence diagram* dapat dilihat dari tabel 2.4.

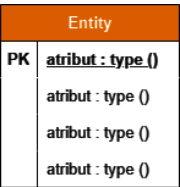

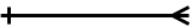

Tabel 2. 4 Simbol keterangan *sequence diagram*

Simbol	Keterangan
<i>Object</i> 	Representasi dari entitas yang berinteraksi dalam sistem. digambarkan sebagai persegi panjang dengan nama objek di atasnya.
<i>Message</i> 	Mewakili pesan yang dikirim dari satu objek ke objek lainnya. Pesan digambarkan sebagai panah yang menghubungkan dua objek
<i>Lifeline</i> 	Menggambarkan durasi waktu dalam skenario, menunjukkan kapan objek aktif dalam interaksi dengan objek lain melalui pertukaran pesan.
Aktor 	pengguna yang berinteraksi dengan objek dalam sistem yang sedang dianalisis. Aktor digambarkan sebagai gambar manusia.

## 2.8 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram (ERD)* merupakan representasi *visual* mengenai entitas, proses, dan relasi antara tabel-tabel dalam sebuah basis data (Putri, Voutama, & Heryana, 2023). Dengan memanfaatkan ERD, kita dapat menguji model yang berfokus pada struktur data dan mengabaikan elemen proses yang terlibat. Di bawah ini tertera simbol-simbol yang digunakan dalam diagram urutan (*sequence diagram*) beserta contoh-contoh diagramnya:

Tabel 2. 5 Keterangan ERD

Simbol	Keterangan
<p><i>Entity</i></p> 	Entitas adalah benda atau objek yang ada dalam dunia nyata dan akan diwujudkan dalam struktur data basis data. Atribut Setiap entitas memiliki informasi yang terkait yang dikenal sebagai atribut. <i>type</i> Merupakan jenis data yang sesuai dengan atribut-atribut yang ada.
<p><i>One to one</i></p> 	<i>One to one</i> merupakan tipe hubungan di mana satu entitas hanya dapat memiliki hubungan dengan satu entitas lain, dan sebaliknya.
<p><i>One to many</i></p> 	<i>One to many</i> merupakan bentuk hubungan di mana satu entitas dapat terhubung dengan sejumlah besar entitas lainnya.
<p><i>Many to many</i></p> 	<i>Many to many</i> merupakan tipe hubungan di mana satu entitas dapat memiliki koneksi dengan banyak entitas lainnya dan sebaliknya.

## 2.9 Xampp

*Xampp* adalah aplikasi yang dapat berjalan di berbagai sistem operasi dan berfungsi sebagai paket perangkat lunak lengkap untuk menciptakan lingkungan

pengembangan lokal (Muhammad Yusril Helmi Setyawan & Dinda Ayu Pratiwi, 2020). Dalam pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, *XAMPP* digunakan untuk menginstal server *web Apache*, sistem manajemen basis data *MySQL*, bahasa pemrograman *server-side PHP*, serta beberapa alat bantu seperti *phpMyAdmin*.

Penggunaan *XAMPP* memungkinkan penulis untuk mengembangkan dan menguji aplikasi *web* secara lokal sebelum diterapkan di server produksi. Dengan adanya *XAMPP*, dapat dengan mudah membuat dan menguji sistem informasi surat berbasis *web* dalam lingkungan pengembangan lokal sebelum di-*deploy* secara *live* di lingkungan produksi (Zikri et al., 2023) Ini memberikan kemudahan dalam pengembangan dan pemeliharaan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi

## 2.10 Php My Admin

*PhpMyAdmin* adalah aplikasi berbasis *web* yang digunakan untuk mengelola basis data *MySQL* (Ramadhan & Mukhaiyar, 2020). Dalam pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, *PhpMyAdmin* merupakan salah satu komponen dalam paket perangkat lunak *XAMPP*. *XAMPP* menggabungkan *Apache* sebagai server *web*, *MySQL* sebagai sistem manajemen basis data, dan *PHP* sebagai bahasa pemrograman *server-side*, serta menyediakan alat bantu seperti *PhpMyAdmin* untuk mengelola basis data *MySQL* (Zikri et al., 2023).

Dengan *PhpMyAdmin*, pengguna, termasuk penulis di Universitas Nasional Karangturi, dapat melakukan berbagai tugas pengelolaan basis data, seperti membuat, mengedit, dan menghapus tabel, mengelola data dalam tabel, menjalankan *query SQL*, dan melakukan *backup* dan *restore* basis data. Dengan antarmuka berbasis *web* yang *user-friendly*, *PhpMyAdmin* menyederhanakan tugas-tugas pengelolaan basis data yang kompleks, sehingga memudahkan tim pengembang dalam mengelola dan memanipulasi data yang diperlukan dalam

pengembangan sistem informasi surat berbasis *web* (Tari, 2019). Dengan bantuan *PhpMyAdmin* dalam paket *XAMPP*, penulis dapat dengan mudah mengelola basis data *MySQL* untuk sistem informasi surat berbasis *web* di lingkungan pengembangan lokal mereka sebelum implementasi di server produksi.

## 2.11 Penelitian Terkait

Tabel 2. 6 Penelitian terkait

No	Judul	Penulis	Hasil/Kesimpulan
1	Sistem Informasi Pengarsipan Surat Berbasis <i>Web</i> Pada Smp Negeri 5 Kota Sorong	Diah, Fitri Ramadhani	Sistem informasi pengelolaan arsip surat secara terkomputerisasi di SMP Negeri 5 Kota Sorong mampu mengatasi masalah efektivitas dan efisiensi dalam pengelolaan surat-menyurat. Sistem ini berhasil mempermudah admin dalam mengelola surat keluar dan surat masuk, serta pencarian surat dan laporan pengarsipan. Metode pengembangan <i>Waterfall</i> digunakan dengan tahapan <i>requirement design, analysis, testing, maintenance</i> , dan <i>coding</i> .

No	Judul	Penulis	Hasil/Kesimpulan
2	Sistem Informasi Pengarsipan Surat Masuk dan Surat Keluar Pada Dinas Tenaga Kerja dan Transmigrasi Kabupaten Musi Banyuasin	Rahayu Amalia, Nurul Huda	Dinas Tenaga Kerja dan Transmigrasi Kabupaten Banyuasin saat ini masih mengelola surat masuk dan surat keluar secara manual. Ini menyebabkan kurangnya efisiensi dan efektivitas dalam penggunaan tenaga, waktu. Untuk mengatasinya, membuat Sistem Informasi Pengarsipan Surat menyurat menggunakan metodologi RAD, dengan program yang menggunakan PHP sebagai bahasa pemrograman dan <i>MySQL</i> sebagai basis datanya. Diharapkan Sistem Informasi Pengarsipan Surat dapat meningkatkan efisiensi dan efektivitas dalam pengelolaan surat.
3	Pembangunan Aplikasi Sistem Informasi Surat Tugas Mengajar Dan Surat Kerja Praktek Pada Fakultas Teknik Universitas Sangga Buana Dengan	Janjang Sofian, Hanhan Hanafiah Solihin	Pengelolaan informasi surat di Kesekretariatan Fakultas Teknik Universitas Sangga Buana masih mengandalkan sistem konvensional, yang menghambat efektivitas pelayanan. Oleh karena itu, membangun aplikasi sistem informasi pengelolaan surat berbasis <i>web</i> dapat menjadi solusi untuk mengatasi masalah tersebut,

No	Judul	Penulis	Hasil/Kesimpulan
	Memanfaatkan Teknologi <i>Framework Codeigniter</i>		metode <i>Prototype</i> digunakan sebagai pendekatan yang berfokus pada desain, fungsi, dan <i>user-interface</i> . Hasil akhir dari pengembangan sistem ini adalah sebuah aplikasi yang dapat memfasilitasi pengolahan surat dengan cepat, akurat, dan efisien.
4	Perancangan Aplikasi Disposisi Surat Masuk Dan Pembuatan Surat Keluar Di Stmik Musirawas Lubuklinggau Berbasis <i>Web Mobile</i>	Davit Irawan	STMIK Musirawas Lubuklinggau bermasalah dalam pengelolaan surat menyurat secara manual. merancang sebuah aplikasi berbasis <i>web mobile</i> yang dapat membantu mengatasi permasalahan tersebut. Metode <i>Waterfall</i> digunakan dalam pembangunan aplikasi ini, dengan <i>MySQL</i> sebagai <i>database</i> utama. Hasil dari penelitian ini menunjukkan bahwa Aplikasi Disposisi Surat Masuk Dan Pembuatan Surat Keluar di STMIK Musirawas Lubuklinggau telah berhasil membantu dalam pengolahan data surat masuk dan surat keluar dengan lebih efektif.

No	Judul	Penulis	Hasil/Kesimpulan
5	Aplikasi Pengarsipan Surat Masuk dan Surat Keluar Berbasis <i>Web</i> pada SMP Negeri 32 Pekanbaru	Yulisman, Refni Wahyuni, Yuda Irawan	Pengarsipan surat pada SMP Negeri 32 Pekanbaru saat ini dilakukan secara manual, menyebabkan kesulitan dalam mencari arsip surat untuk mengatasi masalah berikut. menciptakan aplikasi pengarsipan surat masuk dan keluar. Metode Model <i>Waterfall</i> digunakan dalam pengembangan sistem dengan UML sebagai bahasa perancangan dan analisis aplikasi. Hasil penelitian menunjukkan bahwa aplikasi ini sangat membantu dan mempermudah SMP Negeri 32 Pekanbaru dalam pengarsipan surat masuk dan keluar, terutama di bagian Tata Usaha karena semua surat tersimpan dalam basis data.