

I. PENDAHULUAN

1.1 Latar Belakang

Salah satu Unit Pelaksana Teknis (UPT) yang berada di bawah pengawasan Direktorat Jenderal Sumber Daya Air adalah Balai Besar Wilayah Sungai Mesuji Sekampung yang merupakan bagian dari Kementerian Pekerjaan Umum dan Perumahan Rakyat. Balai Besar Wilayah Sungai Mesuji Sekampung disebut juga "BBWS Mesuji Sekampung" terletak di Jalan Gatot Subroto No. 57, Garuntang, Kecamatan Bumi Waras, Kota Bandar Lampung, Lampung 35401.

BBWS Mesuji Sekampung masuk dalam kategori BBWS Tipe A sesuai Peraturan Menteri PUPR Nomor 16 Tahun 2020 tentang Organisasi dan Tata Kerja Unit Pelaksana Teknis di Lingkungan Kementerian Pekerjaan Umum dan Perumahan Rakyat. Oleh karena itu, struktur organisasi BBWS Tipe A terbagi menjadi berbagai bagian dan bidang. Salah satunya Bagian Umum dan Tata Usaha dibantu oleh tiga Subkoordinator Pelaksana Tugas yang merupakan Pejabat Fungsional Ahli Muda, yaitu Subkoordinator Kepegawaian, Pengelolaan Arsip dan Layanan Umum; Subkoordinator Keuangan, Fasilitas Pengelolaan Barang Persediaan Bencana dan Pengelolaan BMN; serta Subkoordinator Hukum dan Komunikasi Publik.

Perencanaan dan pengelolaan urusan kas dan perbendaharaan, pengembangan pegawai, pengelolaan organisasi, dan tugas-tugas lainnya berada di bawah lingkup Bagian Umum dan Tata Usaha. Sedangkan Subkoordinator Kepegawaian, Pengelolaan Arsip dan Layanan Umum memiliki tugas yaitu melakukan urusan administrasi kepegawaian seperti surat-menyurat, pengadaan dan pengelolaan inventaris, dan lain-lain.

Menurut (Hendrayani & Lestari, 2020), surat adalah suatu bentuk komunikasi tertulis yang dikirimkan dari satu pihak ke pihak lain. Bagian Umum dan Tata Usaha menggunakan surat sebagai komunikasi tertulis mengenai hal-hal yang terkait dengan pekerjaan pegawai, antar pegawai, atau dengan pihak *eksternal* yang berkaitan dengan pekerjaan pegawai. Sistem surat menyurat yang baik dan pengelolaan arsip yang efektif, membuat informasi kepegawaian dapat diakses

dengan mudah dan administrasi menjadi teratur. Dalam pembuatan surat terdapat nomor surat yang merupakan kode unik berupa angka atau gabungan angka dan huruf. Tujuan nomor surat adalah sebagai identifikasi untuk membedakan setiap surat yang dibuat. Surat-surat yang memiliki nomor surat dapat dengan mudah diidentifikasi dan dibedakan satu sama lain selama proses pencatatan, pengarsipan, dan penelusuran kembali (Sari, 2018).

Saat ini sistem pembuatan nomor surat di BBWS Mesuji Sekampung yaitu, setiap pegawai yang membutuhkan nomor surat akan mengisi *google form* yang disediakan dengan memberikan informasi seperti jenis surat, tanggal surat, dan perihal. Setelah itu pegawai tersebut menghubungi sekretaris secara langsung. Sekretaris akan melihat responden tersebut pada *spreadsheet* dan menentukan sendiri nomor surat yang sesuai dengan informasi yang diberikan lalu nomor surat tersebut akan diberikan secara langsung oleh sekretaris.

Pemberian nomor surat di BBWS Mesuji Sekampung tersebut menggunakan *google form* dan informasi masih disimpan dalam bentuk *spreadsheet* yang memanfaatkan satu akun *google* untuk semua penyimpanan dalam hal kepegawaian. Akun *google* tersebut tidak hanya digunakan oleh satu pegawai melainkan pegawai lain dapat menggunakannya. Menurut (Pertiwi, 2022) penggunaan *spreadsheet* memiliki risiko keamanan data, penyimpanan data terbatas, data mudah terhapus secara tidak sengaja. Selain itu pembuatan nomor surat dengan sistem seperti ini risiko kemungkinan kesalahan manusia lebih tinggi, seperti penulisan nomor surat yang salah berupa duplikasi nomor surat atau kelalaian dalam pencatatan, tidak konsisten dalam format penomoran, dan juga penyimpanan arsip yang kurang baik karena berkas surat disimpan secara terpisah.

Berdasarkan latar belakang tersebut penulis ingin membantu pegawai BBWS Mesuji Sekampung dalam pembuatan nomor surat dengan merancang “Aplikasi Pembuatan Nomor Surat Berbasis *Web* Menggunakan *Framework Codeigniter* Pada Balai Besar Wilayah Sungai Mesuji Sekampung”. Aplikasi ini dapat secara otomatis membuat nomor surat dengan format yang konsisten sesuai dengan informasi yang diberikan pegawai tanpa harus mencari nomor surat yang dibuat sebelumnya.

Selain itu, aplikasi ini memiliki fitur *login* untuk meningkatkan keamanan data dan mengurangi risiko kesalahan manusia seperti ketidaksengajaan menghapus data ataupun mengubah data. Arsip surat juga dapat disimpan bersamaan dengan informasi surat sehingga memudahkan dalam pencarian surat.

1.2 Tujuan

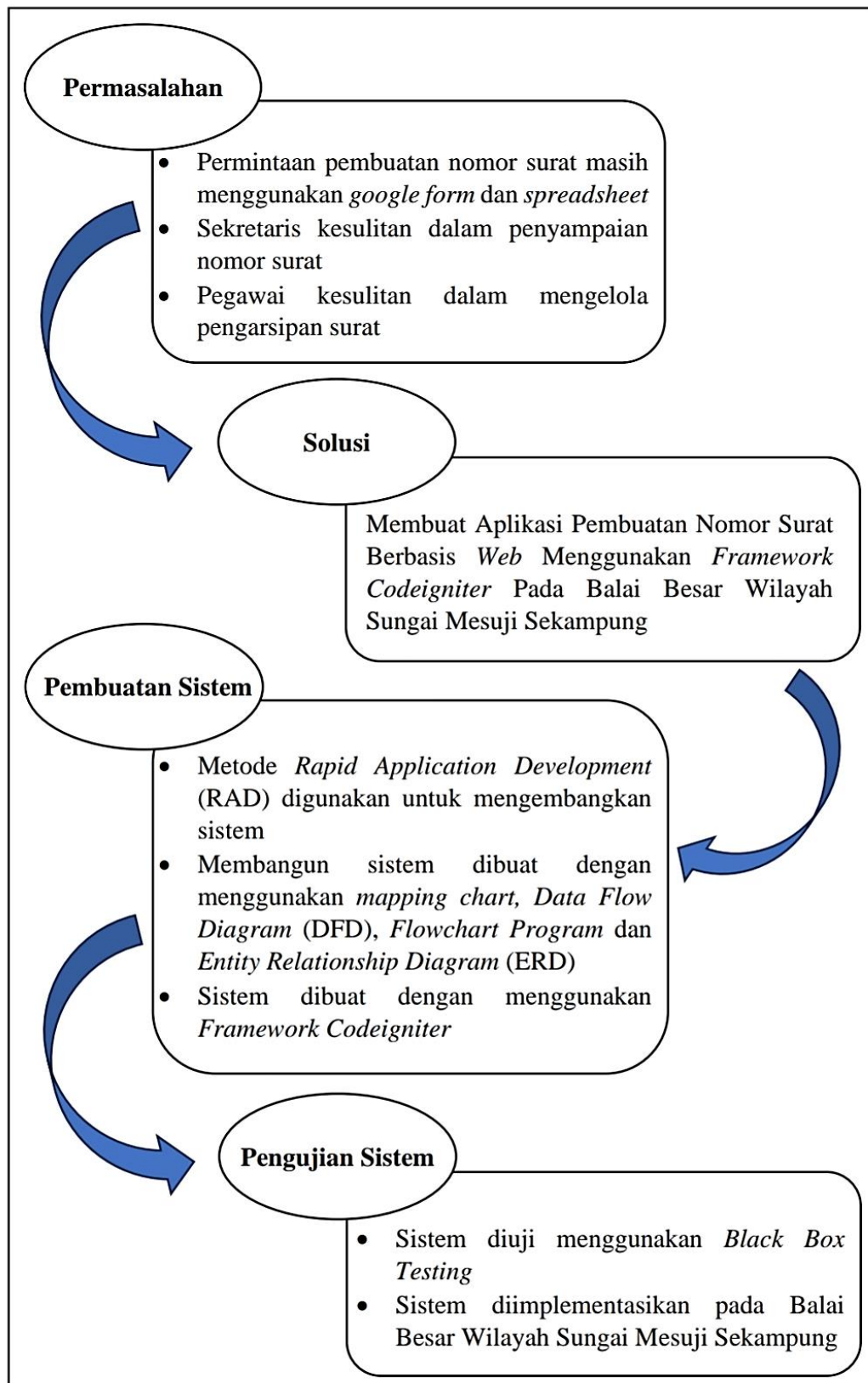
Nomor surat digunakan untuk membedakan setiap surat yang diterbitkan. Tujuan dibentuknya tugas akhir ini yaitu menghasilkan *web* yang mendukung proses pembuatan dan pengelolaan nomor surat di Balai Besar Wilayah Sungai Mesuji Sekampung, mengurangi kemungkinan kesalahan manusia dalam pembuatan nomor surat. Untuk mencegah kesalahan penulisan atau penomoran, *web* ini akan menghasilkan nomor surat secara otomatis sesuai dengan format dan aturan yang telah ditentukan. *Web* ini juga bertujuan untuk memudahkan pegawai mengelola dan menyimpan surat untuk membedakan dalam pengarsipan, pencatatan, dan pencarian surat di kemudian hari.

1.3 Kerangka Pemikiran

Salah satu cara untuk membedakan surat yang diterbitkan yaitu dengan memberi nomor surat yang unik. Nomor surat ini membedakan setiap surat yang dikirim oleh suatu perusahaan sehingga dapat dibedakan satu sama lain dan diidentifikasi dengan mudah. Aplikasi pembuatan nomor surat berbasis *web* dimungkinkan untuk dibuat berdasarkan masalah yang ada. Metode pengembangan sistem *Rapid Application Development* (RAD) dengan 4 tahapan yaitu *requirements planning*, *user design*, *construction* dan *cutover* digunakan untuk menghasilkan aplikasi ini (Afifah dkk., 2019).

Requirements planning adalah salah satu tahap penting dalam pengembangan aplikasi. Berdasarkan wawancara dan pengamatan disimpulkan bahwa BBWS Mesuji Sekampung membutuhkan sebuah aplikasi untuk pembuatan nomor surat. Langkah selanjutnya yaitu *user design*, yang melibatkan pembuatan alur kerja dan *interface* aplikasi menggunakan *mapping chart* atau bagan alir dokumen, *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), dan *flowchart* program.

Membangun aplikasi berdasarkan desain yang telah dihasilkan sebelumnya dengan menggunakan bahasa pemrograman PHP, HTML, dan MySQL merupakan langkah selanjutnya dalam tahap *construction*. Tahap *cutover* ialah tahap terakhir yang sangat penting dalam pengembangan perangkat lunak. Tahapan ini merupakan proses pengujian oleh pengguna untuk memastikan aplikasi berfungsi dengan baik menggunakan *Black Box Testing*. Gambar 1 menunjukkan kerangka pemikiran berdasarkan uraian yang telah diberikan.



Gambar 1. Kerangka pemikiran

1.4 Kontribusi

Aplikasi Pembuatan Nomor Surat Berbasis *Web* Menggunakan *Framework Codeigniter* pada Balai Besar Wilayah Sungai Mesuji Sekampung ini diharapkan dapat memberikan kontribusi yang bermanfaat bagi pihak-pihak yang berkepentingan, antara lain:

1. Pegawai

Adapun kontribusi yang didapatkan pegawai terhadap aplikasi pembuatan nomor surat ini yaitu:

- a. Memudahkan pegawai dalam melakukan permintaan nomor surat kepada sekretaris.
- b. Memudahkan pegawai dalam mengelola pengarsipan surat yang lebih terstruktur dan terorganisir.
- c. Memudahkan pegawai dalam pencarian atau penelusuran surat yang telah dibuat.

2. Sekretaris

- a. Memudahkan sekretaris dalam memberikan nomor surat kepada pegawai. Agar tidak terjadi nomor berulang dan kesalahan format nomor surat.
- b. Memudahkan sekretaris dalam melakukan pencocokan data yang diberikan pegawai dengan *file* surat yang telah di *input* pegawai.
- c. Memudahkan sekretaris dalam mengelola pengarsipan surat yang lebih terstruktur dan terorganisir.
- d. Memudahkan sekretaris dalam pencarian atau penelusuran surat yang sudah dibuat.

II. TINJAUAN PUSTAKA

2.1 Aplikasi

Sebuah program atau perangkat lunak yang disebut aplikasi dibuat untuk menawarkan fitur, layanan, atau solusi tertentu kepada pengguna. Aplikasi adalah program siap pakai yang memungkinkan pengguna menjalankan instruksi aplikasi untuk memperoleh hasil yang lebih tepat sesuai dengan tujuan penggunaan aplikasi. Dengan memanfaatkan salah satu teknik pemrosesan data aplikasi yang dimaksudkan, aplikasi juga berfungsi sebagai pemecah masalah (Wahyuni & Irawan, 2020).

Aplikasi mampu melakukan berbagai fitur dan fungsi, termasuk pemrosesan data, komunikasi, interaksi pengguna, visualisasi, keamanan, dan lain-lain. *User interface* sering dikenal sebagai UI, biasanya digunakan untuk mendukung program. Ini memungkinkan pengguna untuk berinteraksi dengan aplikasi, memasukkan data, memeriksa hasil, dan mengakses fitur-fiturnya. Aplikasi ini membantu pengguna menyelesaikan tugas dengan lebih efisien, meningkatkan produktivitas dan membuat pekerjaan lebih mudah.

Aplikasi dapat dibagi menjadi beberapa jenis berdasarkan perkembangan saat ini yaitu: aplikasi desktop, aplikasi berbasis *web*, dan aplikasi seluler. Aplikasi berbasis *web* memungkinkan pengguna untuk mengakses dan menggunakannya dari mana saja dan kapan saja, mereka dapat terhubung ke internet dengan menggunakan *web browser* dan terhubung ke *server* melalui jaringan.

2.2 Surat

Surat adalah sebuah dokumen tertulis yang berisi pesan atau komunikasi yang ditujukan kepada orang lain. Orang yang satu dengan orang yang lain dapat mengkomunikasikan informasi secara tertulis melalui surat. Informasi dapat berupa pernyataan, pengetahuan, laporan, instruksi, ide, kritik, dan lain-lain (Hendrayani & Lestari, 2020). Agar komunikasi berjalan dengan baik dan efektif, penting untuk memperhatikan tata cara penulisan yang benar, termasuk format, etika, serta kejelasan dan kebijaksanaan dalam menyampaikan pesan.

Pengelolaan surat yang baik sangat diperlukan karena dokumentasi yang dibuat memuat informasi yang lengkap, yang akan sangat membantu untuk memperoleh arsip dengan cepat dan tepat. Kearsipan merupakan dasar dari pemeliharaan surat, itu mencakup menyusun dan menyimpan dokumen sehingga dapat ditemukan kembali saat diperlukan. Data dan informasi harus disediakan dengan cepat dan tepat ketika diperlukan. (Ikasari dkk., 2020).

2.3 Nomor Surat

Menurut (Rumaf, 2022) nomor surat adalah lambang yang terdiri dari angka-angka atau gabungan angka dan huruf yang tertera pada surat keluar suatu perusahaan, organisasi, atau instansi dan mempunyai arti tertentu. Tujuan nomor surat adalah untuk membedakan satu surat dengan surat lainnya. Selain itu, digunakan untuk memudahkan dalam pelacakan, pengarsipan, dan pengelolaan surat-surat tersebut. Dengan nomor surat yang unik, instansi atau organisasi dapat merujuk kembali ke surat-surat yang telah diterbitkan.

Nomor surat sering kali mencerminkan informasi seperti tanggal, jenis surat, atau kode unit kerja yang membantu dalam mengklasifikasikan dan mengatur surat-surat. Tanpa nomor surat, mencari dan mengidentifikasi surat tertentu dalam arsip bisa menjadi lebih rumit. Proses pencarian akan menjadi lebih sulit karena tidak ada tanda pengenal unik yang dapat digunakan sebagai referensi. Proses administrasi menjadi tidak teratur atau membutuhkan usaha dan waktu lebih untuk memverifikasi atau membandingkan surat-surat yang terkait.

2.4 Web

Tim Berners-Lee, yang dianggap sebagai pelopor dalam pengembangan *web* dan internet dan sering disebut sebagai bapak *web* dan internet, menciptakan *web* yang juga dikenal sebagai *World Wide Web* (WWW) untuk pertama kalinya pada tahun 1999. Tim bergabung dengan laboratorium penelitian ilmiah CERN di Genewa, Swiss. Teknologi *web server* berfungsi sebagai layanan yang dapat dilakukan pada teknologi internet sebagai penyedia halaman *online*. (Solichin, 2016).

Menurut (Rerung, 2018), halaman *web* adalah *file* teks murni (*plain text*) yang dapat dibuka, dijelajahi, atau diterjemahkan oleh *browser* internet berisi *sintaks* HTML. Internet sekarang identik dengan *web* karena semakin populernya internet sebagai *interface* standar untuk berbagai layanan. Awalnya hanya digunakan untuk mendapatkan informasi, tetapi sekarang juga digunakan untuk berkomunikasi melalui *email*, *chat*, hingga melakukan transaksi bisnis.

2.5 XAMPP

Program manajemen *server open source* bernama XAMPP dibuat oleh *Apache Friends* dan ditawarkan tanpa biaya. X di XAMPP adalah singkatan dari *class platform*, sesuai dengan namanya artinya mendukung sejumlah sistem operasi, termasuk Windows, *macOS*, dan Linux. Solusi yang mudah digunakan dan ringan, XAMPP terdiri dari *Apache*, *MariaDB* (dibuat oleh *MySQL*), PHP, dan *Perl* yang memungkinkan pembuatan *web server* lokal untuk menguji situs *web*. Sistem tambahan yang didukung oleh XAMPP termasuk Windows, *macOS*, dan Linux (Fitri, 2020).

2.6 Rapid Application Development

Rapid Application Development (RAD) adalah metode pendekatan berulang dalam pengembangan aplikasi yang mengutamakan siklus pengembangan cepat (Khoerudin dkk., 2023). RAD dalam pengembangan perangkat lunak dianggap sebagai metodologi tambahan yang menekankan pada siklus pengembangan yang singkat, jelas, dan cepat. Berikut 4 tahapan pengembangan perangkat lunak RAD:

1. Requirement Planning

Tahap *requirement planning* yaitu tahapan melakukan penelitian untuk menemukan masalah, mengumpulkan data yang relevan dengan penelitian, dan melakukan analisis kebutuhan pengguna dan pengelola untuk menetapkan tujuan sistem dan kebutuhan informasi yang diinginkan. Untuk memutuskan apa yang diperlukan untuk membangun suatu sistem pada saat ini, keterlibatan kedua belah pihak sangat penting.

2. *User Design*

Pada tahap *user design*, pengguna dan analis sistem berkolaborasi untuk menghasilkan model pengembangan yang merinci semua operasi, masukan, dan keluaran sistem. Pada titik ini, proses dimulai dengan mengembangkan aplikasi untuk memenuhi kebutuhan, membangun *database*, dan merancang alur proses saat ini dan yang diusulkan.

3. *Construction*

Tahap *construction* yaitu mengubah tahapan *user design* menjadi bahasa pemrograman. Tahap ini berfokus pada pengembangan perangkat lunak dan aplikasi yang memungkinkan pengguna terus berkontribusi pada penelitian dengan memberikan saran untuk menyempurnakan sistem yang akan dikembangkan.

4. *Cutover*

Tahap *cutover* adalah tahap terakhir di mana sistem di mana sistem diimplementasikan dengan melakukan konversi data, dan menguji program untuk menemukan kesalahan pada sistem yang telah dikembangkan.

2.7 *Database*

Database atau basis data merupakan kumpulan data yang saling berhubungan dalam bentuk berkas atau *file* yang tersimpan dalam sistem komputer dan digunakan untuk menyimpan, mengelola, dan mengakses data dengan mudah (Subagia, 2016). Informasi yang berkaitan satu sama lain dan disimpan dalam tabel dianggap sebagai data dalam *database*. Perangkat lunak yang dikenal sebagai *Database Management System* (DBMS) berfungsi sebagai *interface* untuk mengendalikan dan menggunakan *database*.

2.7.1 *MySQL*

Menurut (Widia & Asriningtias, 2021) *MySQL* merupakan salah satu perangkat lunak *open source* menggunakan *Structured Query Language* (SQL) sebagai bahasa utama untuk mengakses *database*. Biasanya digunakan untuk membuat, mengelola, dan mengakses basis data yang mengikuti model relasional.

2.7.2 *PHPmyAdmin*

PHPMyAdmin adalah perangkat lunak gratis dan *open source* yang mengelola *database MySQL* baik secara lokal maupun internet menggunakan bahasa pemrograman PHP (Wijayanto & Putra, 2020). Dengan perangkat lunak ini, pengguna dapat membuat, mengubah, menghapus, dan mengelola tabel, kolom, dan baris dalam basis data *MySQL*.

2.8 *Visual Studio Code*

Visual Studi Code (VS Code) merupakan perangkat lunak untuk mengembangkan aplikasi. Dengan *VS Code* kode-kode program dapat ditulis, dijalankan, ataupun diuji. *VS Code* mendukung berbagai bahasa pemrograman dan *platform*, termasuk HTML, CSS, *JavaScript*, *Python*, *C++*, *Java*, *Ruby*, dan banyak lagi (Enterprise, 2019).

2.9 Bahasa Pemrograman

Bahasa pemrograman adalah bahasa yang digunakan untuk membuat program yang dapat dibaca komputer (Hidayat & Gumilang, 2017). Setiap bahasa pemrograman memiliki *sintaks*, struktur, dan aturan tata bahasa yang unik, dan dapat digunakan untuk membuat berbagai jenis program, mulai dari program *desktop*, aplikasi *web*, perangkat seluler, hingga perangkat keras khusus.

2.9.1 *Hypertext Markup Language*

Menurut (Pamungkas, 2017) bahasa pemrograman *web* yang disebut *Hypertext Markup Language (HTML)* menggunakan format data untuk menghasilkan *hypertext* yang dapat dibaca di beberapa platform tanpa memerlukan konversi format. Struktur dan komponen dasar yang diperlukan untuk mengatur tampilan dan konten pada halaman *web* disediakan oleh HTML.

2.9.2 *Hypertext Preprocessor*

Hypertext Preprocessor (PHP) merupakan bahasa pemrograman *open source* menggunakan *script server-side*, yang berarti kode program dibuat di *server* dan kemudian ditampilkan di *browser*. PHP dapat berfungsi dalam dokumen

HTML dan dapat menghasilkan konten untuk halaman *web* sesuai permintaan (Nurhidayah dkk., 2020).

2.9.3 Cascading Style Sheet

Cascading Style Sheets (CSS) merupakan bahasa pemrograman yang digunakan untuk mengendalikan tampilan dan tata letak elemen-elemen pada halaman *web*. Dengan menggunakan CSS, penulisan elemen *tag* HTML seperti *font*, warna, teks, ukuran, tabel, dan masih banyak lagi menjadi lebih ringkas dan mencegah penulisan kembali. Dengan CSS dapat mengatur tampilan dan menampilkan halaman dengan format yang berbeda (Lewenusa, 2020).

2.10 Framework Codeigniter

Framework merupakan struktur kerja yang memiliki komponen, aturan dan pedoman telah ditentukan dalam membantu mempercepat proses pengembangan perangkat lunak dengan pengelolaan kode yang lebih mudah.

Codeigniter adalah *framework* PHP yang mengimplementasikan pola desain *Model, View, Controller* (MVC) untuk membuat *website* dinamis. *Codeigniter* juga menyediakan dokumentasi menyeluruh, contoh implementasi kode dan desain yang lebih sederhana dan mudah beradaptasi (Nurhidayah dkk., 2020).

2.10.1 Model, View, Controller (MVC)

Pola desain arsitektur perangkat lunak yang disebut *Model, View, Controller* (MVC) membagi fungsionalitas aplikasi menjadi tiga bagian utama: *Model, View*, dan *Controller*. Pola desain ini digunakan untuk mengelola kode dalam aplikasi secara sistematis dan mengurangi penulisan kode yang berulang-ulang (Sulistiono, 2018). *Model, view* dan *controller* dijelaskan sebagai berikut:

1) Model

Model adalah sekelompok aturan yang mengontrol bagaimana *database* diatur. Biasanya memiliki hubungan langsung dengan *database* dan dapat menyisipkan, memperbarui, menghapus, dan mencari data (*insert, update,*

delete, search) saat menangani validasi *controller*, namun tidak dapat memiliki hubungan langsung dengan bagian *view*.

2) *View*

View adalah komponen struktur yang mengatur tampilan program dan menampilkan data kepada pengguna. *File template* HTML di bagian aplikasi *web* ini biasanya dikelola oleh *controller* dan tidak memiliki akses langsung ke *model*. Berisi satu atau lebih *file* ekstensi *.php* di folder *views* yang sering kali berisi formulir, tabel, paragraf, dan konten lainnya.

3) *Controller*

Controller berfungsi menghubungkan *model* dan *view*, memproses permintaan *client* yang dikirim ke *server*, menafsirkannya, dan merespons dengan menginstruksikan tampilan yang menghadap pengguna untuk menampilkan informasi yang relevan.

2.10.2 Bootstrap

Bootstrap merupakan *toolkit* untuk keperluan *front-end* yang banyak digunakan dalam pengembangan aplikasi *web* bersifat *open source*. *Bootstrap* banyak digunakan dalam merancang kebutuhan aplikasi *web* karena dapat mempercepat proses pengembangan, menghemat waktu dan membuat halaman *web* menjadi responsif dengan cepat serta tampilan yang menarik (Kholil & Akhsani, 2021).


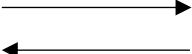
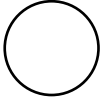
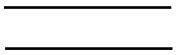
2.11 Rancangan Alur dan Data Sistem

Pada pembuatan sebuah aplikasi diperlukan sebuah desain rancangan yang berisi alur setiap data yang dikelola. Hal ini dilakukan untuk mendapatkan sebuah data yang berisikan hasil analisa. Rancangan ini membantu dalam pemahaman dan dokumentasi tentang bagaimana sistem beroperasi, bagaimana data diproses, dan bagaimana komponen sistem berinteraksi satu sama lain. Setelah data dari hasil analisa tersebut sudah di dapatkan maka *developer* dapat membuat desain dan rancangan untuk membangun aplikasi yang akan dibuat.

2.11.1 Data Flow Diagram

Menurut (Tanjung, 2017) suatu sistem dapat direpresentasikan secara logis menggunakan *Data Flow Diagram* (DFD), yang tidak bergantung pada perangkat keras, perangkat lunak, format data, atau sistem *file* sistem. Manfaat DFD adalah memudahkan individu yang kurang memiliki pengalaman komputer dalam memahami sistem yang akan dibangun. DFD menggunakan teknik visualisasi untuk mendemonstrasikan bagaimana data diproses dalam suatu sistem informasi, termasuk sumber data, langkah-langkah pemrosesan data, dan hasil yang diharapkan. Simbol-simbol dalam DFD disajikan pada Tabel 1.

Tabel 1. Simbol-simbol *Data Flow Diagram* (DFD)

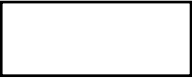

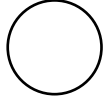

Simbol	Nama	Keterangan
	Entitas	Menunjukkan dari mana data masuk ke sistem atau ke mana data keluar dari sistem
	Aliran Data	Menggambarkan bagaimana data bergerak dari satu entitas atau proses ke entitas atau proses lain dalam sistem
	Proses	Menggambarkan aktivitas atau tindakan yang dilakukan dalam konteks bisnis atau sistem yang dianalisis
	<i>Data Store</i>	<i>Data store</i> atau penyimpanan data proses.

Sumber: (Soufitri, 2019)

2.11.2 Entity Relationship Diagram (ERD)

Entity Relation Diagram (ERD) adalah jenis diagram yang digunakan untuk menggambarkan hubungan antara entitas (obyek data) yang disimpan dalam suatu sistem *database* atau sistem informasi (Hasan & Muhammad, 2020). ERD digunakan dalam perancangan basis data untuk menunjukkan entitas, atribut, dan hubungan antara entitas. Tabel 2 menyajikan simbol-simbol dalam *Entity Relation Diagram*.

Tabel 2. Simbol-simbol *Entity Relationship Diagram* (ERD)

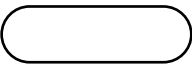

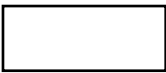
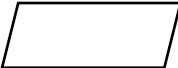
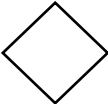
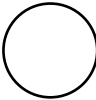
Simbol	Nama	Keterangan
	Entitas	Menggambarkan tabel dalam basis data
	Relasi	Menggambarkan hubungan (relasi) antar entitas
	Atribut	Menggambarkan karakteristik atau properti khusus yang dimiliki oleh entitas dalam sistem atau basis data
	Alur	Mencakup bagaimana data terkait dan mengalir antara entitas

Sumber : (Rusmawan, 2019)


2.11.3 *Flowchart*

Flowchart merupakan teknik analitis bergambar yang digunakan untuk menggambarkan urutan langkah-langkah atau alur data dalam suatu sistem, menjelaskan secara jelas, ringkas, dan logis komponen sistem informasi. *Flowchart* mencakup pengendalian *intern* perusahaan serta sistem dan prosedurnya (Tuasamu dkk., 2023). Tabel 3 menyajikan simbol-simbol *flowchart*.

Tabel 3. Simbol-simbol *Flowchart*

Simbol	Nama	Keterangan
	<i>Start/End Point</i>	Menunjukkan awal atau akhir dari suatu aliran proses
	Aliran	Menggambarkan alur dalam suatu proses
	Proses	Menggambarkan langkah-langkah yang dilakukan dalam proses
	<i>Input/ output</i>	Menggambarkan interaksi antara proses dengan data yang masuk dan data keluar
	Keputusan	Menggambarkan titik di dalam proses di mana pengambilan keputusan atau percabangan logika
	Penghubung halaman	Menggambarkan hubungan antara langkah dalam <i>flowchart</i> yang berada di halaman berbeda atau dalam lembaran yang berbeda

Tabel 3. (Lanjutan)



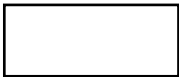
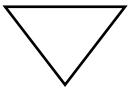

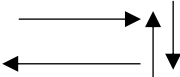

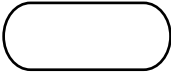
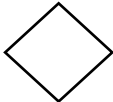
Simbol	Nama	Keterangan
	Proses terdefinisi	Menunjukkan proses yang sudah ditentukan di tempat lain

Sumber: (Rohaeti & Zaliluddin, 2018)

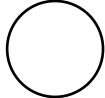
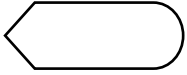
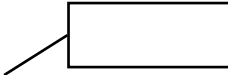
2.11.4 Mapping Chart

Mapping chart adalah diagram alir dokumen yang menunjukkan pergerakan laporan, formulir, dan salinan. *Mapping chart* menggambarkan pergerakan dokumen dan informasi mulai dari saat dokumen digunakan hingga tidak digunakan lagi dalam sistem (Verawati & Liksha, 2018). Tabel 4 menunjukkan simbol dalam *Mapping Chart*.

Tabel 4. Simbol-simbol *Mapping Chart*

Simbol	Nama	Keterangan
	Dokumen	Memberikan deskripsi dokumen terkait proses dapat berbentuk surat, formulir, buku, file, atau hasil cetakan.
	Kegiatan Manual	Menggambarkan aktivitas atau tugas yang dilakukan manusia secara manual
	Proses	Menggambarkan langkah-langkah yang terjadi dalam suatu proses
	Simpanan/ arsip	Menunjukkan dokumen yang diarsipkan (arsip manual) dalam proses
	Database	Menggambarkan tempat atau entitas penyimpanan data
	Garis alir	Menunjukkan arah dari setiap proses
	Pemasukan	Menunjukkan data dimasukkan secara manual melalui <i>keyboard</i>
	Terminasi	Menunjukkan titik awal atau akhir dari proses alur kerja
	Decision	Menunjukkan penyeleksian kondisi di dalam program

Tabel 4. (Lanjutan)

Simbol	Nama	Keterangan
	Penghubung	Penghubung langkah-langkah pada halaman yang berbeda
	<i>Display</i>	Menunjukkan informasi atau data ditampilkan kepada pengguna selama atau setelah proses
	Penjelasan	Menggambarkan penjelasan dari suatu proses

Sumber: (Putra dkk., 2018)

2.12 *Black Box Testing*

Black box testing adalah jenis pengujian yang berkonsentrasi pada persyaratan fungsional perangkat lunak. Persyaratan fungsional suatu perangkat lunak dapat diuji oleh penguji menggunakan serangkaian kondisi masukan yang ditentukan (Solang dkk., 2021). Penguji hanya memeriksa fitur luar perangkat lunak, termasuk bagaimana perangkat lunak memproses masukan dan menghasilkan keluaran yang sesuai dengan standar atau persyaratan yang ditetapkan. Penguji tidak perlu mengetahui bahasa pemrograman atau memiliki akses ke *source code* perangkat lunak.

2.13 Artikel Terkait

Penelitian sebelumnya digunakan sebagai acuan penelitian dan data pendukung. Penelitian berikut digunakan untuk menyelesaikan tugas akhir::

1. Jurnal yang disusun oleh (Ningrum & Puspitasari, 2020) dengan judul “Rancang Bangun Aplikasi Berbasis *Web* Untuk *Generate* Nomor Surat PT. Telekomunikasi Indonesia Witel Surabaya Selatan”. Sistem ini dirancang untuk membantu karyawan PT. Telekomunikasi Indonesia dalam menghasilkan nomor surat secara otomatis, sehingga administrasi dapat melakukan pengecekan surat keluar setiap hari untuk mengetahui apakah surat itu sesuai atau tidak dan untuk menyimpan surat.
2. Jurnal yang disusun oleh (Roza dkk., 2020) dengan judul “Perancangan Sistem Informasi Pengarsipan Berkas Berbasis *Web* Pada PT. Duta Abadi

Primantara”. Landasan sistem informasi ini adalah penyimpanan arsip yang sebagian masih terkomputerisasi dan masih menggunakan buku agenda dan *Microsoft Excel*, sehingga dapat mengakibatkan kesalahan fatal seperti *redundansi* data. Metode yang dikembangkan dalam penelitian ini dapat membantu karyawan dalam prosedur pengarsipan. Fakta bahwa hanya karyawan yang dapat menggunakan sistem ini memungkinkan pencarian *file* lebih cepat dan penyimpanan aman.

3. Jurnal yang disusun oleh (Sonita & Sari, 2018) dengan judul “Implementasi Algoritma *Sequential Searching* Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik”. Dengan menggunakan algoritma *Sequential Searching*, penelitian ini bermaksud untuk mempermudah pencarian nomor surat pada sistem arsip elektronik. Pendekatan *Rapid Application Development (RAD)*, salah satu metodologi pengembangan sistem informasi tercepat, merupakan teknik pengembangan sistem yang digunakan dalam penelitian ini.
4. Jurnal yang disusun oleh (Amalia, 2022) dengan judul “ Pengembangan Aplikasi Penomoran Surat Pada BPJS Ketenagakerjaan Cabang Palembang Berbasis *Website*”. Penelitian ini menciptakan aplikasi penomoran surat untuk BPJS Ketenagakerjaan karena pengelolaan surat menyurat belum efektif karena penomoran surat masih dicatat dengan pena dan buku. Dengan menggunakan PHP, *Framework Codeigniter*, dan *database MySQL* aplikasi ini dibuat.