

BAB II. TINJAUAN PUSTAKA

2.1 Jaringan Dokumentasi dan Informasi Hukum (JDIH)

Pada Peraturan Presiden Republik Indonesia Nomor 33 tahun 2012 tentang Jaringan Dokumentasi dan Informasi Hukum Nasional (JDIHN) pasal 1, dokumen hukum adalah produk hukum yang berupa peraturan perundang-undangan atau produk hukum selain peraturan perundang-undangan yang meliputi namun tidak terbatas pada putusan pengadilan, yurisprudensi, monografi hukum, artikel majalah hukum, buku hukum, penelitian hukum, pengkajian hukum, naskah akademis, dan rancangan peraturan perundang-undangan (Peraturan Presiden Republik Indonesia, 2012).

Jaringan Dokumentasi dan Informasi Hukum (JDIH) merupakan salah satu jenis jaringan informasi yang berfokus pada pendayagunaan dokumen hukum. Untuk membentuk JDIH, setiap unit yang ingin bergabung harus melakukan kegiatan dokumentasi terlebih dahulu yang terkait dengan kerjasama jaringan di masa depan. Oleh karena itu, pengumpulan dokumen hukum menjadi kunci utama dalam kegiatan JDIH.

Secara umum, dokumentasi melibatkan serangkaian teknik untuk mengelola dokumen dan memastikan kelestariannya. Kegiatan dokumentasi mencakup aktivitas seperti memilih dokumen, mengumpulkan dokumen, dan mengolahnya menggunakan teknik tertentu agar dapat dengan mudah didistribusikan dan dimanfaatkan oleh orang lain sesuai kebutuhan mereka. Tujuannya adalah untuk memastikan bahwa dokumen tersebut tetap dapat digunakan dan didistribusikan untuk kepentingan individu yang membutuhkan informasi yang terkandung di dalamnya. Fokus utama dalam kegiatan dokumentasi adalah menyediakan dokumen agar dapat disebarakan kepada pengguna yang membutuhkan informasi tersebut sebagai referensi dalam pembuatan dokumen baru (Salami & Kurniawan, 2016).

Pada Peraturan Presiden Republik Indonesia Nomor 33 tahun 2012 tentang Jaringan Dokumentasi dan Informasi Hukum Nasional (JDIHN) pasal 3 menyatakan bahwa JDIHN bertujuan untuk:

- a. Menjamin terciptanya Pengelolaan Dokumentasi dan Informasi Hukum yang terpadu dan terintegrasi di berbagai instansi pemerintah dan institusi lainnya;
- b. Menjamin ketersediaan dokumentasi dan informasi hukum yang lengkap dan akurat, serta dapat diakses secara cepat dan mudah;
- c. Mengembangkan kerja sama yang efektif antara Pusat jaringan dan Anggota jaringan serta antar sesama Anggota jaringan dalam rangka penyediaan dokumentasi dan informasi hukum; dan
- d. Meningkatkan kualitas pembangunan hukum nasional dan pelayanan kepada publik sebagai salah satu wujud ketatapemerintahan yang baik, transparan, efektif, efisien, dan bertanggung jawab.

Saat ini, beberapa lembaga sudah memiliki sistem JDIH, salah satunya adalah Badan Pemeriksaan Keuangan Republik Indonesia (BPK RI).

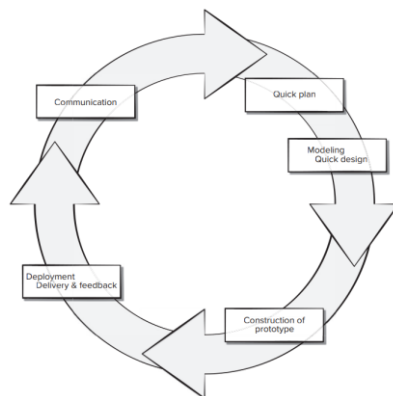


Gambar 2.1. Tampilan *Home* Sistem JDIH BPK RI

BPK RI membuat sistem JDIH yang bertujuan untuk menyebarkan informasi peraturan perundang-undangan dan dokumentasi hukum secara mudah, cepat dan akurat kepada para pengguna baik kalangan internal BPK RI maupun masyarakat. Dalam penelitian ini, sistem JDIH milik BPK RI dijadikan referensi dalam perancangan sistem JDIH Universitas Pertamina.

2.2 Metode *Prototyping*

Sering kali, pengguna menentukan serangkaian tujuan perangkat lunak masih secara umum dan tidak mengidentifikasi *requirement* secara rinci untuk setiap fiturnya. Dalam kasus lainnya, pengembang mungkin tidak yakin dengan efisiensi algoritma, kemampuan beradaptasi sistem operasi, atau bentuk interaksi manusia dengan mesin yang harus dilakukan. Dalam situasi ini, penggunaan paradigma *prototyping* mungkin akan menjadi pendekatan yang terbaik. Menurut (Pressman & Maxim, 2020), paradigma *prototyping* dapat membantu kita dan *stakeholders* untuk lebih memahami apa yang akan dibangun saat *requirement* perangkat lunak masih tidak jelas.



Gambar 2.2. *The Prototyping Paradigm* (Pressman & Maxim, 2020)

Terdapat beberapa tahapan dalam mengimplementasikan pengembangan perangkat lunak dengan menggunakan metode *prototyping* seperti pada Gambar 2.2 yang akan dijelaskan sebagai berikut.

2.2.1 Communication

Menurut (Pressman & Maxim, 2020), paradigma *prototyping* dimulai dari proses *communication*. Pada proses tersebut, pengembang dan pengguna akan bertemu dan berkomunikasi untuk menentukan *requirement* yang dibutuhkan perangkat lunak atau biasa disebut sebagai proses *requirement gathering*. Proses komunikasi ini diperlukan untuk menentukan tujuan keseluruhan perangkat lunak serta mengidentifikasi *requirement* secara detail sehingga perangkat lunak yang dibuat sesuai dengan keinginan atau kebutuhan pengguna perangkat lunak.

Requirement perangkat lunak biasanya terbagi menjadi 2 bagian, yaitu *functional requirements* dan *non-functional requirements*.

2.2.1.1 Functional Requirements

Functional requirements adalah persyaratan yang menentukan fungsi atau tugas yang harus dilakukan oleh perangkat lunak. Persyaratan ini menentukan apa yang harus dilakukan oleh perangkat lunak, seperti tugas-tugas yang harus dilakukan, input yang diperlukan, dan output yang diharapkan. Contohnya adalah sistem pendaftaran online yang harus memiliki fitur-fitur seperti formulir pendaftaran, verifikasi email, dan pemberitahuan status pendaftaran (Pressman & Maxim, 2020).

2.2.1.2 Non-functional Requirements

Berbeda dengan *functional requirements*, *non-functional requirements* adalah persyaratan yang menentukan atribut atau kualitas yang harus dipenuhi oleh perangkat lunak. Persyaratan ini menentukan bagaimana perangkat lunak harus bekerja, seperti kecepatan, keamanan, dan ketersediaan. Contohnya adalah sistem perbankan online yang harus memiliki keamanan yang tinggi untuk melindungi data nasabah (Pressman & Maxim, 2020).

2.2.2 Quick Plan

Setelah proses pengumpulan *requirement* dilakukan dan sudah terdefinisi dengan baik, selanjutnya terjadi proses *quick plan*, yaitu perencanaan secara cepat. Pada proses ini, pengembang akan mengidentifikasi sumber daya yang tersedia untuk memenuhi kebutuhan perangkat keras dan perangkat lunak yang diperlukan untuk menghasilkan produk dengan lebih rinci. Kemudian, melakukan estimasi terhadap risiko-risiko utama yang mungkin akan terjadi (Andrianita, 2015).

2.2.3 Modeling Quick Design

Selanjutnya, terjadi proses perancangan pemodelan desain secara cepat (*Modeling Quick Design*). Proses ini berfokus pada merancang desain elemen-elemen yang terlihat oleh pengguna akhir. Pengembang akan membuat format tampilan antarmuka pengguna atau desain *User Interface (UI)*. Model perangkat lunak yang akan dibangun direpresentasikan dalam bentuk *Unified Modelling Language (UML)* seperti *Use Case Diagram*, *Activity Diagram*, *Entity Relationship Diagram*, *Physical Data Modelling* serta *UI Design*. Kemudian, desain cepat tersebut digunakan sebagai acuan untuk memulai pembuatan prototipe (Andrianita, 2015).

Commented [DF4]: Cari buku Pressman terbaru di Perpustakaan ada buku Pressman yang terbaru.

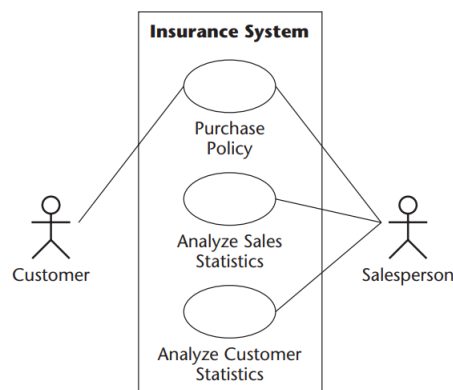
2.2.3.1 Unified Modeling Language (UML)

UML atau *Unified Modeling Language* adalah sebuah bahasa pemodelan visual yang digunakan dalam rekayasa perangkat lunak untuk mendokumentasikan, merancang, dan memvisualisasikan sistem perangkat lunak. UML juga dapat digunakan untuk menganalisis kebutuhan sistem, merencanakan arsitektur perangkat lunak, dan menyediakan dokumen spesifikasi teknis. UML menggambarkan konsep dan elemen dalam sistem perangkat lunak dengan menggunakan notasi grafis, seperti *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram* (Object Management Group, 2021).

Pada perancangan sistem JDIH, UML yang akan dibuat untuk mendokumentasikan, merancang dan memvisualisasikan sistem JDIH adalah *Use Case Diagram* dan *Activity Diagram*.

2.2.3.1.1 Use Case Diagram

Use case diagram adalah jenis diagram UML yang digunakan untuk menggambarkan interaksi antara pengguna atau aktor dengan sistem perangkat lunak. *Use case diagram* menggambarkan berbagai aksi atau aktivitas yang dapat dilakukan oleh pengguna pada sistem perangkat lunak dan bagaimana sistem akan merespon aksi tersebut. *Use case diagram* membantu untuk memahami kebutuhan pengguna dan kebutuhan sistem perangkat lunak yang sedang dikembangkan (Fowler, 2004).

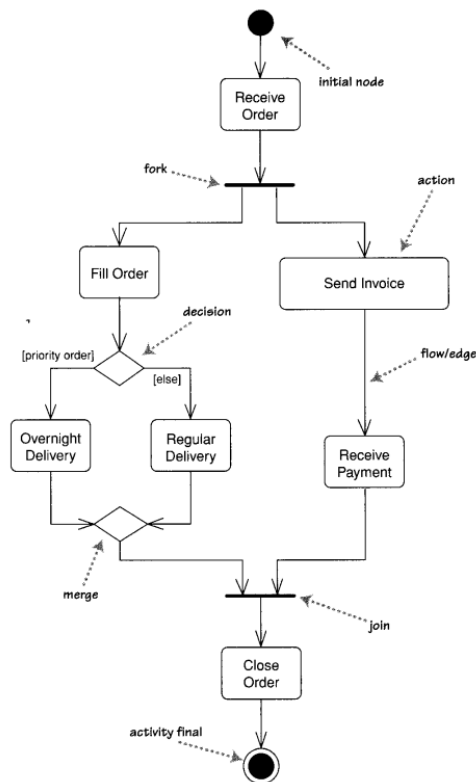


Gambar 2.3. *Use Case Diagram* untuk Sistem Asuransi (Eriksson, Penker, Lyons, & Fado, 2004)

Use case diagram terdiri dari tiga elemen utama: aktor, *use case*, dan hubungan antara aktor dan *use case*. Aktor adalah entitas yang terlibat dalam interaksi dengan sistem, seperti pengguna atau sistem eksternal. *Use case* adalah aktivitas atau aksi yang dapat dilakukan oleh aktor pada sistem, seperti melakukan login, membuat pesanan atau seperti contoh pada Gambar 2.3. Hubungan antara aktor dan *use case* menunjukkan cara aktor terlibat dalam setiap *use case* (Eriksson, Penker, Lyons, & Fado, 2004).

2.2.3.1.2 Activity Diagram

Activity diagram adalah jenis diagram UML yang digunakan untuk menggambarkan alur aktivitas atau proses bisnis dari sebuah sistem perangkat lunak. Activity diagram menunjukkan urutan aktivitas dalam suatu proses, tindakan yang dilakukan di setiap aktivitas, serta alur kontrol dan kondisi untuk setiap aktivitas (Eriksson, Penker, Lyons, & Fado, 2004).



Gambar 2.4. A simple Activity Diagram (Fowler, 2004)

Activity diagram terdiri dari dua jenis elemen: node dan edge. Node mewakili aktivitas atau tindakan dalam proses, sedangkan edge menghubungkan node dan menunjukkan urutan aktivitas. Ada beberapa jenis node dalam activity diagram, seperti initial node, final node, decision node, fork node, join node, dan action node (Fowler, 2004).

2.2.3.2 Database Design

Suatu *database* atau basis data adalah sebuah kumpulan informasi dan data yang diatur secara terstruktur agar mudah diakses oleh pengguna sistem informasi (Muslihudin & Oktafianto, 2016). Biasanya, sebelum sebuah *database* pada sistem akan dibuat, pengembang akan melakukan perancangan *database* terlebih dahulu. Menurut (Connolly & Begg, 2014), untuk melakukan perancangan sistem *database*, ada beberapa langkah yang perlu dilakukan, antara lain:

a. Analisis Kebutuhan Data

Langkah awal yang harus dilakukan adalah menganalisis kebutuhan sistem dan menentukan persyaratan data yang diperlukan. Langkah ini meliputi pengumpulan data, identifikasi data yang relevan, serta pemahaman terhadap hubungan antar data.

- b. Desain Konseptual Basis Data (*Conceptual Database Design*)
Setelah kebutuhan sistem diketahui, langkah berikutnya adalah merancang konsep atau model *database*. Model database dapat dirancang menggunakan beberapa metode, seperti pembuatan *Entity Relationship Diagram* (ERD).
- c. Desain Logis Basis Data (*Logical Database Design*)
Setelah model konseptual selesai dibuat, langkah selanjutnya adalah membuat model database secara logis. Desain logis meliputi penentuan tipe data untuk setiap entitas dan atribut, serta hubungan antar tabel.
- d. Desain Fisik Basis Data (*Physical Database Design*)
Setelah model logis selesai dibuat, langkah selanjutnya adalah merancang struktur fisik database. Desain fisik meliputi penentuan format data, indeks, dan aturan integritas data. Pada langkah ini, kita dapat membuat sebuah *Physical Data Modelling* (PDM).
- e. Implementasi Basis Data
Setelah desain database selesai, langkah selanjutnya adalah mengimplementasikan database dengan membuat tabel, relasi, dan aturan integritas data.

2.2.3.2.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah sebuah model visual yang digunakan untuk merepresentasikan hubungan antara entitas (*entity*) dalam suatu sistem informasi. ERD sering digunakan dalam pengembangan sistem informasi untuk membantu para pengembang dan analis dalam merancang struktur *database* (Elmasri & Navathe, 2016).

ERD terdiri dari tiga jenis entitas utama yaitu *Entity* (entitas), *Attribute* (atribut), dan *Relationship* (hubungan antara entitas). Entitas merepresentasikan objek atau subjek dalam dunia nyata yang akan disimpan dalam *database*, sedangkan atribut adalah sifat atau karakteristik dari entitas tersebut. *Relationship* merepresentasikan hubungan antara dua atau lebih entitas (Elmasri & Navathe, 2016).

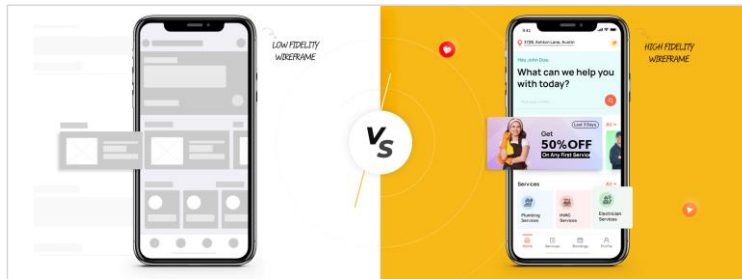
2.2.3.2.2 Physical Data Modelling (PDM)

Physical Data Modeling (PDM) adalah proses merancang struktur fisik atau implementasi *database*, seperti pemilihan tipe data, *constraint*, *index*, dan relasi antar tabel. PDM mengubah skema logikal yang telah dibuat dalam konsep ERD menjadi struktur fisik dalam bentuk diagram tabel, kolom, dan konstrain-konstrain untuk mendukung implementasi *database* pada sistem yang digunakan (Sumadya, Ginardi, & Akbar, 2016).

Pada PDM, sudah terdapat *primary key* serta *foreign key* dari setiap tabel yang berelasi. *Foreign key* ditentukan berdasarkan dari jenis relasi pada tabel database yang berupa *one to one*, *one to many* atau *many to many*. Selanjutnya, pengembang dapat memilih jenis *database* yang tepat untuk sistem yang sedang dibangun dan menentukan skema fisik yang cocok untuk menjalankan sistem tersebut.

2.2.3.3 User Interface Design (UI Design)

UI Design adalah proses merancang tampilan antarmuka suatu aplikasi atau website yang akan digunakan oleh pengguna. Salah satu aspek penting dalam perancangan UI adalah menentukan level kehalusan atau tingkat detail dari desain antarmuka yang akan dibuat. Tingkat kehalusan ini biasanya dibedakan menjadi *low-fidelity* dan *high-fidelity*.



Gambar 2.5. Perbedaan *Low Fidelity* dan *High Fidelity* (Codiant, 2023)

Low-fidelity dan *high-fidelity design* adalah dua jenis desain antarmuka yang memiliki tingkat kehalusan atau tingkat detail yang berbeda. *Low-fidelity design* memiliki tingkat kehalusan yang rendah, sedangkan *high-fidelity design* memiliki tingkat kehalusan yang tinggi. Kedua jenis desain ini digunakan dalam proses desain antarmuka untuk mencapai tujuan yang berbeda.

1.2.4 Construction of Prototype

Dalam tahap pembuatan *prototype*, desain antarmuka yang telah dirancang akan diimplementasikan menjadi sebuah *prototype*. Pengembang memastikan bahwa fitur dan fungsi perangkat lunak telah sempurna diterapkan pada *prototype*. Setelah *prototype* berhasil dibuat, maka tahap selanjutnya dapat dilanjutkan.

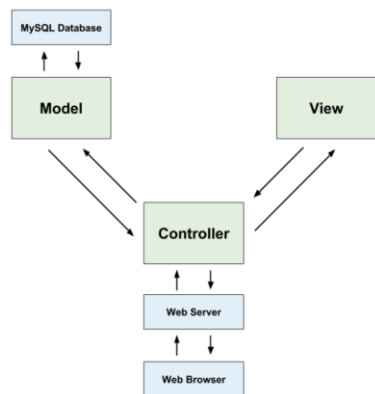
2.2.5 Deployment Delivery & Feedback

Setelah selesai dibuat, *prototype* akan diserahkan oleh pengembang kepada pengguna untuk dievaluasi. Pengguna akan memberikan umpan balik mengenai *prototype* yang telah dibuat, sehingga pengembang dapat memperhalus spesifikasi kebutuhan. Dengan mempelajari kebutuhan tersebut, pengembang dapat melakukan perbaikan *prototype* pada iterasi selanjutnya (Andrianita, 2015).

Setelah proses iterasi tersebut selesai dan pengguna sudah puas dengan *prototype*, pengembang akan melakukan proses *development* atau mengkodekan sistem. Dalam proses *development*, *prototype* yang sudah disepakati diterjemahkan kedalam bahasa pemrograman yang sesuai. Setelah *development* selesai, proses selanjutnya adalah pengujian dan pemeliharaan.

2.3 Konsep Model View Controller (MVC)

Konsep *Model View Controller* (MVC) adalah salah satu konsep yang digunakan dalam pembuatan aplikasi dengan cara memisahkan bagian-bagian *code* menjadi 3 bagian, yaitu *model*, *view* dan *controller* (Restaldo & Beeh, 2022).



Gambar 2.6. Proses MVC (Forte, 2016)

Controller adalah perantara antara model dan *view* (tampilan). Tugas dari *controller* adalah mengontrol aliran informasi dengan menerima input pengguna dari tampilan, dan menginstruksikan model dan *view* untuk melakukan tindakan berdasarkan informasi tersebut. Model bertanggung jawab atas pengelolaan manajemen data dalam aplikasi, biasanya *create*, *read*, *update*, dan *delete* (CRUD). *View* bertanggung jawab untuk menampilkan data dari model (Forte, 2016).

Model adalah bagian dari kode program yang berfungsi untuk mengelola data, termasuk memasukkan data, memperbarui data, dan menghapus data, dan berhubungan langsung dengan database. Fungsi-fungsi ini tidak dapat berkomunikasi langsung dengan bagian *view*. Bagian kode program yang mengatur tampilan website disebut *View*. Biasanya, *View* ditulis dalam bentuk file skrip HTML. Selain itu, *view* berfungsi untuk menampilkan data dan inputan dari pengguna, sehingga *View* dapat diartikan sebagai halaman web. *Controller* merupakan bagian dari program yang berperan sebagai penghubung antara Model dan *View*. Fungsi utama dari *Controller* adalah mengatur perintah-perintah yang terkait dengan pemrosesan data dan pengiriman data ke halaman web. Selain itu, *Controller* juga berperan dalam menerima permintaan dan data dari pengguna dan menentukan tindakan apa yang harus diambil oleh aplikasi (Wijaya & Christian, 2019).

2.4 Pengembangan Situs Web

Pengembangan sebuah situs web mengacu pada pembuatan, pembangunan, dan pemeliharaan situs web tersebut. Pengembangan situs web biasanya mencakup aspek-aspek seperti merancang desain web, pemrograman web, manajemen basis data hingga penerbitan sebuah web atau *deployment*. Ketika melakukan pengembangan situs web, biasanya proses pengembangan ini dibagi menjadi 2 bagian yaitu pengembangan *front-end* dan *back-end*.

2.4.1 Pengembangan *Front-end*

Front-end adalah bagian dari situs web dimana tempat pengguna berinteraksi secara langsung. Sebelum membuat bagian tampilan atau *front-end* dari aplikasi, biasanya seorang pengembang akan diberikan sebuah desain *User Interface* (UI) oleh seorang *UI Designer*. Untuk pengembangan *front-end* itu sendiri biasanya menggunakan HTML dan CSS, serta terdapat

juga salah satu *framework* untuk mempermudah dalam pengembangan *front-end* yaitu Bootstrap (Arhandi, 2016).

2.4.2 Pengembangan *Back-end*

Back-end adalah *server side* dari sebuah situs web, yaitu bagian dari situs web yang tidak dapat dilihat dan berinteraksi dengan pengguna. *Back-end* dapat disebut juga merupakan bagian dari perangkat lunak yang tidak bersentuhan langsung dengan pengguna. *Back-end* digunakan untuk menyimpan dan mengatur data. Untuk pengembangan *back-end*, bahasa pemrograman yang digunakan adalah bahasa pemrograman yang dapat digunakan untuk mengelola database, mengolah file dan I/O seperti Python, PHP, Java atau NodeJs (Arhandi, 2016).

2.5 Framework

Perancang *website* telah mengembangkan cara untuk memudahkan pembuatan *website*, yaitu dengan menggunakan *framework*. *Framework* adalah sebuah kerangka kerja yang dibuat untuk menyederhanakan pembuatan *website* dengan menyediakan komponen dan variabel yang dibutuhkan oleh perancang *website* untuk mempermudah pembacaan kode, perencanaan, pengujian dan pemeliharaan. *Framework* yang tersedia saat ini sudah lebih dari 30 jenis dengan berbagai fungsi spesifik. Awalnya, publikasi *framework* dimulai pada tahun 2004 dengan Prado 1 sebagai *framework* pertama, diikuti oleh tiga *framework* mendunia pada tahun 2005 yaitu CakePHP 1, Symfony 1 dan Prado 2 (Prasena & Sama, 2020). *Framework* terbagi menjadi 2 jenis, terdapat *framework* untuk memudahkan dalam pengembangan *Front-end* dan juga *framework* untuk pengembangan *Back-end*.

2.5.1 Laravel

Laravel adalah salah satu *framework* yang digunakan untuk membangun aplikasi berbasis web dengan menggunakan bahasa pemrograman PHP. Dalam pengembangan aplikasi berbasis web, laravel menjadi salah satu *framework* PHP yang paling populer. Laravel dilengkapi dengan banyak fitur yang sangat membantu dalam mempercepat pembuatan aplikasi atau sistem informasi berbasis web. Selain itu, laravel dapat digunakan pada pengembangan *backend* dan *front-end*, atau hanya pada pengembangan *back-end* saja (Arhandi, 2016).

Laravel juga sudah menerapkan konsep MVC seperti yang sudah dijelaskan sebelumnya. Versi laravel yang akan digunakan didalam perancangan sistem JDIH adalah laravel versi 10. Beberapa fitur laravel versi 10 yang akan digunakan pada perancangan sistem JDIH ini adalah sebagai berikut.

1. Blade Templates

Blade adalah sebuah template engine yang disediakan oleh Laravel. Blade memungkinkan pengembang untuk menulis template HTML yang elegan dan efisien dengan fitur-fitur seperti sintaksis template yang sederhana, ekstensi template, dan penggunaan kembali template.

Salah satu kelebihan Blade adalah kemampuannya untuk mengkompilasi template menjadi kode PHP yang cepat dan efisien, yang membuat aplikasi menjadi lebih cepat. Selain itu, Blade juga mendukung fitur-fitur seperti inheritance, sections, dan partials

yang memungkinkan pengembang untuk membagi template menjadi bagian-bagian yang lebih kecil dan mudah dikelola (Laravel, 2023).

2. Eloquent

Eloquent adalah ORM (Object-Relational Mapping) yang disediakan oleh Laravel. ORM memungkinkan pengembang untuk memetakan objek PHP ke tabel database. Dengan Eloquent, pengembang dapat melakukan operasi CRUD (Create, Read, Update, Delete) pada tabel database menggunakan model yang merepresentasikan tabel tersebut (Laravel, 2023).

Eloquent pada laravel memiliki beberapa fitur, salah satunya adalah Eloquent Relationship, fitur ini digunakan mendefinisikan relasi antara model, seperti one-to-one, one-to-many, dan many-to-many relationships.

3. Artisan

Artisan adalah command-line interface (CLI) yang disediakan oleh Laravel. Artisan ini biasanya digunakan oleh pengembang untuk memudahkan proses pengembangan aplikasinya, seperti membuat model, migrasi database, menjalankan test, dan lain sebagainya.

4. Authorization

Authorization atau autentikasi adalah proses verifikasi identitas pengguna dan hak aksesnya ke bagian-bagian tertentu dari aplikasi web. Dalam Laravel, Authorization dilakukan melalui gate dan policy. Gate adalah mekanisme untuk menentukan apakah pengguna memiliki akses ke sebuah tindakan atau tidak, sementara policy adalah mekanisme untuk menentukan kebijakan yang lebih kompleks untuk sebuah tindakan atau sumber daya (Laravel, 2023).

2.5.2 Bootstrap

Bootstrap adalah *framework front-end* yang berisi HTML, CSS, dan JavaScript. Saat ini, bootstrap sangat populer digunakan untuk mengembangkan *website* yang responsif dan ramah karena dapat dipelajari dengan mudah. Penggunaan bootstrap akan mempermudah pengembang dalam membuat tampilan *website*. Bootstrap dapat diunduh dan digunakan secara gratis. Bootstrap menyertakan desain template berbasis HTML dan CSS untuk tipografi, formulir, tombol, tabel, navigasi, modals, carousel gambar, dan masih banyak lagi. Bootstrap juga dapat menggunakan *plug-in* JavaScript (Gaikwad & Adkar, 2019).

Adapun versi bootstrap yang akan digunakan didalam perancangan sistem JDIH adalah Bootstrap versi 5.3. Beberapa fitur bootstrap yang akan digunakan pada perancangan sistem JDIH ini adalah:

1. *Tables*, yang akan digunakan pada halaman dashboard untuk Administrator dan pada beberapa halaman menu pada navigation bar;
2. *Form control*, yang akan digunakan pada halaman untuk menambah peraturan dan SK yang baru;

3. *Input group*, yang akan digunakan pada tempat input pencarian dokumen SK dan peraturan;
4. *Alerts*, yang akan digunakan untuk memberikan pesan kepada pengguna apabila berhasil menambah, menghapus atau mengedit SK dan peraturan;
5. *Button*, yang akan digunakan untuk beberapa button pada sistem;
6. *Navbar*, yang akan digunakan untuk pembuatan navigation bar pada sistem, dan masih banyak lagi.

2.6 Metode Black Box Testing

Black box testing adalah jenis pengujian perangkat lunak yang dilakukan tanpa mengetahui bagaimana kode atau logika program di dalamnya bekerja. Dalam *black box testing*, pengujian dilakukan berdasarkan spesifikasi dan fungsi yang diharapkan dari perangkat lunak tersebut. Tujuan dari *black box testing* adalah untuk menemukan kesalahan dalam perilaku fungsional sistem yang dapat mempengaruhi pengguna atau lingkungan di mana sistem beroperasi (Spillner, Linz, & Schaefer, 2014).

2.6.1 Teknik Black Box Testing

Menurut (Setiawan, 2021), pada metode *black box testing* terdapat beberapa teknik yang dapat dilakukan dalam pengujiannya, diantaranya adalah sebagai berikut.

1. *All pair testing*
Teknik ini disebut juga sebagai *pairwise testing*. Teknik pengujian ini digunakan untuk menguji semua kemungkinan kombinasi dari setiap pasangan input parameter.
2. *Boundary value analysis*
Teknik ini berfokus pada identifikasi kesalahan atau pencarian error dari perspektif pengguna atau sisi eksternal perangkat lunak.
3. *Cause-effect graph*
Teknik pengujian ini menggunakan diagram grafik sebagai acuan, yang menggambarkan hubungan antara penyebab dan efek dari kesalahan yang terjadi dalam perangkat lunak.
4. *Equivalence partitioning*
Teknik ini beroperasi dengan membagi data masukan perangkat lunak menjadi beberapa bagian atau partisi.
5. *Fuzzing*
Fuzzing adalah teknik pencarian bug dalam perangkat lunak yang melibatkan pengujian dengan memasukkan data yang tidak valid atau tidak terduga.
6. *Orthogonal array testing*
Teknik ini berguna untuk menguji input dengan ukuran kecil, namun sulit untuk dijalankan pada skala yang besar.
7. *State transition*
Teknik ini dapat digunakan untuk menguji mesin dan navigasi tampilan antarmuka dengan representasi grafis.

2.7 Studi Literatur Penelitian yang Relevan

Terdapat beberapa peneliti terdahulu yang sudah pernah merancang sistem serupa yaitu sistem informasi pengarsipan dokumen. Pada bagian ini, akan dilakukan sebuah studi literatur dengan menggunakan teknik 3C+2S (*compare, contrast, criticize, synthesize* dan *summarize*).

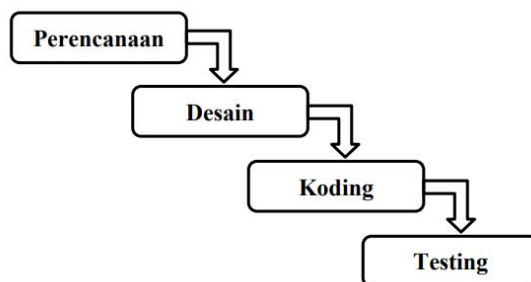
Berikut ini adalah hasil ringkasan atau *summarize* dari beberapa penelitian terkait dengan sistem informasi pengarsipan dokumen.

1. Implementasi Sistem Informasi Pengarsipan Sertifikat Tanah pada Kantor Badan Pertahanan Nasional (BPN) Kota Pekanbaru (Muhammad, Simatupang, & Khair, 2022)

Proses pengarsipan sertifikat tanah pada Kantor BPN Kota Pekanbaru masih dilakukan secara manual sehingga memakan waktu yang cukup lama, kesulitan pencarian data sertifikat tanah serta pegawai kantor BPN juga pernah mengalami kehilangan data sertifikat tanah di ruang arsip.

Tujuan dari penelitian ini adalah membuat suatu sistem informasi berbasis web yang diharapkan dapat membantu pegawai kantor BPN dalam proses pengarsipan data dan penyampaian informasi.

Metode waterfall digunakan pada penelitian ini untuk mengembangkan sistem. Metode *waterfall* pada penelitian ini terdiri dari 4 tahapan secara berurutan seperti berikut.



Gambar 2.7. Tahapan Metode Pengembangan Perangkat Lunak pada Penelitian (Muhammad, Simatupang, & Khair, 2022)

Pada tahap perencanaan, pengumpulan data dengan wawancara serta pengamatan dilakukan oleh peneliti. Selanjutnya peneliti melakukan tahap desain yang terdiri dari 3 tahapan,

- 1) Pemodelan dengan UML
- 2) Merancang *database*
- 3) Membuat tampilan UI

Dilanjutkan tahap selanjutnya adalah pengkodean atau implementasi sistem dan pengujian. Pengujian dilakukan dengan cara menguji satu persatu koneksi dari pengguna ke sistem, menguji semua menu yang ada dan mencoba melakukan penginputan data pada sistem.

Commented [DF5]: Cari yang jurnal sistem arsip aja dan lakukan 3C 2S, penjelasan 3C 2S dengerin di recording. Minimal 3, tapi lebih bagus kalo dapet 5.

Berdasarkan hasil pengujian sistem, sistem pengarsipan ini sudah sesuai dan memenuhi kebutuhan pengguna.

2. Sistem Informasi Pengarsipan Surat Kantor Desa Berbasis Web (Pradini & Sudradjat, 2021)

Penanganan dokumen di kantor desa belum sepenuhnya sesuai dengan prosedur dan belum terorganisir dengan baik. Selain itu, proses penerimaan dan penyimpanan dokumen masih dilakukan secara manual. Dokumen masuk dan keluar masih dicetak dalam bentuk fisik dan ditempatkan di rak arsip, dan pencatatan dokumen dilakukan dalam buku agenda. Selain itu, mencari dokumen di arsip membutuhkan waktu yang lama karena harus membuka satu per satu, sehingga kurang efisien.

Tujuan dari penelitian ini adalah untuk membuat sebuah sistem informasi pengarsipan surat berbasis web yang memudahkan staf Kantor Desa dalam mengelola, menyimpan, dan mencari data terkait surat masuk dan surat keluar yang ada di Kantor Desa. Sistem ini akan memiliki tampilan responsif yang nyaman digunakan oleh pengguna dengan berbagai perangkat.

Metode yang digunakan pada penelitian ini adalah metode *waterfall* yang terdiri dari beberapa tahapan secara berurutan,

- 1) **Analisa Kebutuhan Sistem**, peneliti melakukan eksplorasi tentang kebutuhan pengguna, seperti pegawai arsip, kepala desa, dan administrator, dengan melakukan observasi terhadap sistem pengelolaan arsip surat di Kantor Desa. Tujuannya adalah untuk mengidentifikasi masalah-masalah terkait pengelolaan arsip surat yang perlu dipecahkan.
- 2) **Desain**, peneliti merancang sistem menggunakan UML, yang mencakup *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Sequence Diagram*, *Component Diagram*, dan *Deployment Diagram*. Selain itu, untuk merancang *database*, peneliti menggunakan ERD dan *Logical Record Structure* (LRS) untuk memudahkan implementasi *database* secara fisik.
- 3) **Code Generation**, peneliti mengembangkan sistem pengarsipan surat kantor desa dengan menggunakan bahasa pemrograman PHP, CSS, Javascript dengan *framework* Codeigniter dan Bootstraps.
- 4) **Testing**, penelitian ini menggunakan metode pengujian *black box testing*.
- 5) **Support**, peneliti memilih perangkat keras dan perangkat lunak yang optimal untuk mendukung implementasi sistem informasi pengarsipan surat kantor desa berbasis web, sehingga aplikasi dapat berjalan dengan baik dan efektif dalam mendukung kinerja pegawai bagian arsip surat, serta memberikan manfaat bagi semua *stakeholder* di kantor desa.

Berdasarkan hasil penelitian yang telah dilakukan, sistem informasi pengarsipan surat berbasis web berhasil diimplementasikan dan dapat mengatasi berbagai permasalahan yang sebelumnya dihadapi oleh Kantor Desa serta mampu memenuhi kebutuhan seluruh *stakeholder* terhadap sistem pengarsipan surat tersebut.

3. Sistem Informasi Manajemen Arsip Elektronik (E-Arsip) Berbasis Web pada Marcom BSI Group (Alifudin & Rosyida, 2021)

Saat ini, pengelolaan data barang dan Sumber Daya Manusia (SDM) di Marcom BSI masih menggunakan Microsoft Excel yang menyebabkan data hanya dapat diakses melalui komputer kantor atau komputer yang memiliki duplikat dokumen tersebut. Pencarian data juga terhambat karena dilakukan secara manual. Jika terjadi perubahan data karyawan, administrasi harus mencari dan mengubah data tersebut. Selain itu, karena hanya administrasi yang bertanggung jawab atas pengarsipan data, transparansi data barang menjadi kurang, sehingga ketika tim lain membutuhkan data ketersediaan barang, mereka harus menghubungi administrasi terlebih dahulu.

Adapun tujuan dari penelitian ini adalah membuat suatu sistem informasi pengarsipan secara elektronik yang menggunakan platform web yang akan menjadi alat bantu dalam proses pengarsipan dan pengelolaan laporan di Marcom BSI.

Proses pengembangan perangkat lunak pada penelitian ini menggunakan metode *waterfall* yang terdiri dari beberapa tahapan secara berurutan,

- 1) **Analisa Kebutuhan**, melakukan observasi terhadap semua permasalahan yang terjadi pada bagian administrasi Marcom BSI, terutama terkait pencatatan data barang dan SDM Marcom BSI.
- 2) **Desain**, terdapat 2 tahapan pada proses ini yaitu pemodelan sistem menggunakan UML diagram dan pemodelan *database* yang dibuat dengan menggunakan ERD.
- 3) **Code Generation**, setelah melakukan analisis dan perancangan sistem, sistem tersebut akan dibangun dengan menggunakan bahasa pemrograman, yakni PHP, serta menggunakan *framework* Laravel. Langkah ini akan mengikuti alur dari sistem yang telah dirancang dan yang sudah berjalan sebelumnya, dan mengubahnya menjadi sebuah sistem informasi berbasis web.
- 4) **Testing**, pengujian dilakukan untuk memverifikasi program yang telah dibuat dan memastikan bahwa fungsi dan logika program, serta output yang dihasilkan, sesuai dengan kebutuhan pengguna. Pengujian pada penelitian ini menggunakan metode *black box testing*.
- 5) **Support**, pada tahap pemeliharaan, hal yang harus diperhatikan adalah agar sistem dapat berjalan tanpa kendala dan pengguna dapat menggunakannya dengan lancar.

Berdasarkan dari hasil penelitian dan pengujian sistem, sistem informasi manajemen arsip elektronik berbasis web pada Marcom BSI Group berhasil dibuat dengan hasil pengujian yang sesuai harapan.

4. Sistem Pengarsipan Dokumen Akreditasi Berbasis Web (Kusuma & Aryati, 2019)

Permasalahan pada penelitian ini adalah sering terjadinya keterlambatan pengumpulan dokumen yang biasanya dilakukan oleh anggota tim penyusun borang membuat ketua tim tidak memiliki waktu untuk merekap dokumen-dokumen tersebut. Kesulitan ketua tim untuk

mendapatkan dokumen secara tepat waktu serta kesulitan pencarian dokumen saat dibutuhkan karena harus membuka tumpukan *file* satu persatu.

Penelitian ini bertujuan untuk membuat sistem pengelolaan dokumen-dokumen akreditasi yang dapat menyimpan semua dokumen dalam satu wadah dan dapat diakses dari manapun sehingga mempermudah kepala tim untuk memantau kelengkapan dokumen akreditasi.

Metode yang digunakan pada penelitian ini adalah tahapan pengembangan sistem *System Development Life Cycle* (SDLC) yang terdiri dari identifikasi masalah, pengumpulan data dan analisis kebutuhan, desain sistem, implementasi desain, dan pengujian sistem.

No	Fungsi	Skenario Pengujian	Hasil Proses yang diharapkan
1	Login	<i>Username</i> dan <i>password</i> dikosongkan	Sistem akan menolak akses login dan menampilkan notifikasi
		Mengisi <i>username</i> tapi <i>password</i> dikosongkan, atau sebaliknya	Sistem akan menolak akses login dan menampilkan notifikasi
		Mengisi <i>username</i> benar dan <i>password</i> salah atau sebaliknya	Sistem akan menolak akses login dan menampilkan notifikasi
		Mengisi <i>username</i> dan <i>password</i> dengan benar	Sistem menerima akses login dan kemudian langsung menampilkan halaman utama.
2	Menu Unggah file	Memilih halaman unggah	Muncul halaman unggah file
		Klick Tombol <i>Choose File</i>	Dokumen dapat dipilih
		Muncul pilihan <i>user</i> yang akan dibagikan <i>file</i> tersebut dan muncul pilihan standar	Dapat memilih user yang dapat menggunakan file dan masuk ke dalam kelompok standar sesuai dengan pilihan
3	Fungsi unduh File	Memilih file yang diunduh	File yang akan diunduh terpilih dan tombol unduh dan kirim aktif.
		Klick Tombol unduh	Muncul file yang diunduh
		Klick Tombol kirim	Muncul pilihan divisi yang akan dibagikan dan tombol bagi bisa di klick

Gambar 2.8. Skenario Pengujian Sistem pada Penelitian (Kusuma & Aryati, 2019)

Pengujian sistem pada penelitian ini menggunakan metode *black box testing*. Hasil pengujian menggunakan metode tersebut menunjukkan bahwa sistem pengarsipan dokumen akreditasi yang dibangun dapat membantu dalam manajemen *file* borang serta memungkinkan pembagian *file* ke divisi yang membutuhkan. Oleh karena itu, dapat disimpulkan bahwa semua elemen telah berfungsi sesuai dengan rencana yang telah ditetapkan. Semua aspek yang diuji dalam pengujian menunjukkan hasil yang valid.

Tabel 2.1. Perbandingan *Compare*, *Contrast* dan *Criticize* (3C) pada empat penelitian yang relevan dengan Sistem Arsip

Judul Penelitian 3C	Implementasi Sistem Informasi Pengarsipan Sertifikat Tanah pada Kantor Badan Pertahanan Nasional (BPN) Kota Pekanbaru (Muhammad, Simatupang, & Khair, 2022)	Sistem Informasi Pengarsipan Surat Kantor Desa Berbasis Web (Pradini & Sudradjat, 2021)	Sistem Informasi Manajemen Arsip Elektronik (E-Arsip) Berbasis Web pada Marcom BSI Group (Alifudin & Rosyida, 2021)	Sistem Pengarsipan Dokumen Akreditasi Berbasis Web (Kusuma & Aryati, 2019)
Compare (persamaan)	<p>Permasalahan terdapat pada proses pengarsipan sertifikat tanah yang masih dilakukan secara manual sehingga kurang efektif karena memakan waktu yang cukup lama serta kesulitan pencarian dokumen sertifikat.</p> <p>Metode pengembangan perangkat lunak yang digunakan adalah metode <i>waterfall</i></p>	<p>Permasalahan terdapat pada proses dan penyimpanan dokumen Kantor Desa masih dilakukan secara manual. Selain itu, mencari dokumen di arsip membutuhkan waktu yang lama karena harus membuka satu per satu, sehingga kurang efisien.</p> <p>Metode pengembangan perangkat lunak yang digunakan adalah metode <i>waterfall</i></p>	<p>Permasalahan terdapat pada proses pengelolaan data barang dan SDM di Marcom BSI masih menggunakan Microsoft Excel serta kesulitan pencarian data juga sering terjadi karena proses pencarian masih dilakukan secara manual.</p> <p>Metode pengembangan perangkat lunak yang digunakan adalah metode <i>waterfall</i></p>	<p>Permasalahan terdapat pada proses pengumpulan dokumen borang dan perekapan dokumen masih dilakukan secara manual sehingga kurang efisien. Selain itu, kesulitan pencarian dokumen saat dibutuhkan karena harus membuka tumpukan dokumen satu persatu.</p> <p>Metode pengembangan perangkat lunak yang digunakan adalah metode <i>waterfall</i></p>
Contrast (perbedaan)	Terdapat 2 aktor yang ada pada <i>use case diagram</i> penelitian, yaitu bagian arsip dan bagian pemeliharaan data.	Terdapat hanya 1 aktor yang ada pada <i>use case diagram</i> penelitian, yaitu pegawai arsip.	Terdapat 3 aktor yang ada pada <i>use case diagram</i> penelitian, yaitu administrasi, kepala bagian dan karyawan.	Penelitian ini tidak membuat <i>use case diagram</i> .

Criticize	<p><i>Activity diagram</i> yang dibuat pada penelitian ini tidak lengkap, seharusnya setiap <i>use case</i> dijelaskan alurnya dengan menggunakan <i>activity diagram</i>.</p> <p>Karena terdapat 8 <i>use case</i>, sehingga seharusnya peneliti juga membuat 8 <i>activity diagram</i>.</p>	<p><i>Use case diagram</i> yang dibuat pada peneliti tidak jelas, terdapat <i>use case</i> dengan nama “arsipkan”.</p> <p><i>Activity diagram</i> pada penelitian ini juga tidak lengkap.</p>	<p><i>Use case diagram</i> tidak menggunakan <i>boundaries</i>.</p> <p><i>Activity diagram</i> pada penelitian ini juga tidak lengkap.</p>	<p>Proses perancangan sistem pada penelitian ini hanya membuat <i>Context Diagram</i> (CD) dan <i>Data Flow Diagram</i> (DFD), serta tidak diberikan alasan kenapa tidak membuat <i>use case diagram</i> dan <i>activity diagram</i>.</p> <p>DFD yang dibuat hanya level 0 sehingga tidak detail.</p> <p>Bentuk <i>Physical Data Model</i> (PDM) yang dibuat sangat tidak jelas karena tidak terdapat relasi kardinalitas antara setiap tabel <i>database</i>.</p>
------------------	---	---	--	--

Adapun dari keempat penelitian diatas, terdapat beberapa pendapat terkait manfaat dari pengimplementasian sistem informasi berdasarkan studi kasus atau lembaga yang dijadikan latar belakang pada penelitian, pendapat tersebut adalah sebagai berikut.

1. Menurut (Muhammad, Simatupang, & Khair, 2022) implementasi sistem pengarsipan sertifikat tanah pada Kantor Badan Pertahanan Nasional (BPN) Kota Pekanbaru dapat membantu pihak BPN untuk meningkatkan kinerja pegawai mereka dalam bidang operasional yang memerlukan data sertifikat tanah.
2. Menurut (Pradini & Sudradjat, 2021) implementasi sistem informasi pengarsipan surat berbasis web dapat mengatasi berbagai permasalahan yang sebelumnya dihadapi oleh Kantor Desa, yaitu:
 - 1) Proses pengelolaan arsip surat masuk dan surat keluar menjadi lebih mudah dan cepat;
 - 2) Arsip surat masuk dan surat keluar tersimpan aman di server dalam bentuk digital, sehingga tidak memerlukan banyak tempat fisik, dan menghilangkan kekhawatiran berkas tersebut bisa hilang atau rusak;
 - 3) Proses pencarian arsip surat masuk dan surat keluar dalam bentuk digital dapat dilakukan dengan cepat;
 - 4) Penyimpanan dokumen secara digital mengurangi penggunaan dan pendokumentasian kertas, sehingga tidak menimbulkan penumpukan dokumen fisik secara berlebihan.

3. Menurut (Alifudin & Rosyida, 2021) implementasi sistem informasi manajemen E-Arsip Berbasis Web pada Marcom BSI Group dapat meningkatkan kecepatan karyawan, administrasi dan kepala divisi dalam pengelolaan arsip data barang dan data SDM. Dengan adanya sistem, karyawan dapat mengubah data diri mereka sendiri dan memeriksa ketersediaan barang secara langsung. Sistem ini juga memudahkan bagian Administrasi dalam mencari data barang dan SDM serta menyusun laporan. Kepala divisi dapat menerima laporan terbaru tentang data barang dan SDM secara langsung. Dalam hal ini, pengguna hanya perlu memasukkan data ke dalam aplikasi yang tersedia, sehingga dapat menghemat waktu dan mengurangi kesalahan dalam perhitungan data barang.
4. Menurut (Kusuma & Aryati, 2019) implementasi sistem pengarsipan dokumen akreditasi dapat membantu dalam proses pengelolaan *file* borang dan dapat membagikan *file* ke divisi yang membutuhkan secara langsung.

Dari keempat pendapat diatas, dapat kita gabungkan menjadi sebuah ide yang disebut dengan *synthesize*, sehingga dapat dikatakan bahwa manfaat dari pengimplementasian sistem informasi pengarsipan antara lain:

- a) Meningkatkan efektivitas kinerja karyawan dalam melakukan proses pengarsipan dokumen.
- b) Proses pengelolaan arsip dokumen menjadi lebih mudah dan cepat.
- c) Dokumen yang diarsip tersimpan aman di server dalam bentuk digital sehingga tidak perlu khawatir dokumen akan hilang atau rusak.
- d) Proses pencarian dokumen arsip dapat dilakukan dengan cepat.
- e) Penyimpanan dokumen secara digital mengurangi penggunaan dan pendokumentasian kertas, sehingga tidak menimbulkan penumpukan dokumen fisik secara berlebihan.
- f) Dapat menghemat waktu dan mengurangi kesalahan penginputan atau *human error*.
- g) Dapat mendistribusikan dokumen secara *real-time* kepada divisi lain yang membutuhkan.