

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Pada penelitian Handini, dkk (2020) dengan judul Rancang Bangun Sistem Informasi Penjualan Baju Berbasis Web. Penelitian ini menghasilkan sistem informasi berbasis web yang dapat diakses oleh siapapun secara *online*, mempermudah manajemen pengelolaan produk ataupun pesanan, memperluas daerah jangkauan bisnis, dan menjadi media promosi yang dapat meningkatkan penjualan baju. Metode pengembangan sistem yang digunakan adalah metode *waterfall*. Metode pengujian sistem menggunakan *blackbox testing* yang fokus terhadap proses input dan output program.

Pada penelitian yang dilakukan Paryanti, dkk (2020) dengan judul Peranan Sistem Informasi E-Cuti Untuk Pegawai di Lingkungan Ditjen Badilmiltun, bertujuan untuk mengimplementasikan *e-government* pada sektor sumber daya manusia (SDM) khususnya sistem pengajuan cuti pegawai yang belum memaksimalkan penggunaan teknologi informasi. Pada pelaksanaan pengajuan cuti ataupun proses penyampaian informasi data cuti, pegawai harus melapor secara mandiri dengan cara mendatangi langsung ke petugas administrasi pada bagian kepegawaian. Petugas administrasi pun dalam pembuatan laporan rekapitulasi cuti pegawai setiap bulannya masih menggunakan *Microsoft Word* ataupun *Microsoft Excel* yang penggunaannya masih manual.

Pada penelitian yang dilakukan Achmad dan Amaliyah (2023) dengan judul Aplikasi Sistem Informasi Pengajuan Cuti Berbasis *Website* "SIPECUT", bertujuan untuk memperbaiki sistem pengajuan cuti pegawai yang masih manual dengan membuat aplikasi sistem informasi pengajuan cuti berbasis *website* sehingga proses pengajuan cuti menjadi lebih mudah dan cepat. Disamping mempermudah proses pengajuan cuti dari sisi pegawai, aplikasi cuti berbasis *website* tersebut dapat mempermudah perusahaan mengolah data cuti seluruh pegawai.

Pada penelitian Yusmanisari, dkk (2023) dengan judul Pengembangan Sistem Pengajuan Cuti *Online* Pegawai di Rumah Sakit Umum Anwar Medika, bertujuan untuk meningkatkan kedisiplinan serta memberikan kenyamanan pada karyawan dalam rangka meningkatkan kualitas dari rumah sakit, yaitu dengan caranya pemberian cuti pada setiap pegawai berdasarkan kebijakan internal rumah sakit. RSUD Anwar Medika masih menerapkan sistem cuti secara manual, yaitu dengan menggunakan dokumen atau formulir arsip berbentuk *hardcopy* serta belum memiliki *database*. Sehingga dengan kondisi sistem manual akan mempengaruhi efektivitas dan efisiensi pekerjaan, dikarenakan petugas setiap harinya harus melakukan input pengajuan cuti dari banyak pegawai dan membutuhkan waktu yang lama untuk mengelola dan serta menambah beban kerja bagian kepegawaian. Disamping itu kondisi tersebut dapat menyulitkan proses pengajuan cuti yang diperlukan secara mendadak.

Pada penelitian Karina, dkk (2022) dengan judul Hubungan Sistem Informasi Manajemen Kepegawaian Dengan Kualitas Pelayanan Administrasi Kepegawaian, menjelaskan bahwa pelayanan administrasi kepegawaian yang tidak didukung dengan menggunakan sistem informasi manajemen kepegawaian mengakibatkan hambatan dalam pelayanan kebutuhan PNS. Hasil dari penelitian ini menunjukkan bahwa ada hubungan yang positif dan signifikan antara sistem informasi manajemen kepegawaian dengan kualitas pelayanan administrasi kepegawaian. Sehingga sistem informasi manajemen kepegawaian yang semakin baik maka akan semakin baik pula kualitas pelayanan administrasi kepegawaian.

## **2.2 State Of The Art**

Untuk membantu dalam proses penelitian ini, telah dipelajari penelitian yang relevan terhadap kajian penelitian yang telah dilakukan oleh beberapa pihak sebelumnya. Agar tidak terjadi duplikasi, maka dilakukan penelusuran untuk membandingkan antara penelitian yang sedang dilakukan dengan penelitian yang telah dilakukan oleh pihak lain.

Tabel 2.1. *State Of The Art*

No	Judul	Penulis	Masalah	Ketidaksesuaian
1	Peranan Sistem Informasi E-Cuti Untuk Pegawai di Lingkungan Ditjen Badilmiltun	Paryanti, dkk (2020)	Pegawai harus melaporkan usulan cutinya secara langsung dengan mendatangi petugas administrasi di bagian kepegawaian. Dan proses rekapitulasi cuti pegawai setiap bulannya masih dilakukan menggunakan <i>Microsoft Word</i> atau <i>Microsoft Exel</i>	Sistem ini hanya digunakan dalam lingkup kecil satu unit kerja saja, yaitu Direktorat Jenderal Badan Peradilan Militer dan Tata Usaha Negara
2	Aplikasi Sistem Informasi Pengajuan Cuti Berbasis <i>Website</i> "SIPECUT"	Achmad dan Amaliyah (2023)	Proses pengajuan cuti masih dilakukan secara manual menggunakan sistem berbasis kertas, dan mengharuskan pegawai mencetak formulir dan menyerahkannya ke bagian kepegawaian, yang menyebabkan proses pengajuan cuti menjadi lambat	Sistem ini tidak terdapat integrasi <i>database</i> kepegawaian, sehingga data profile pegawai harus dilakukan input satu per satu. Dan belum ada pengaturan <i>quota</i> cuti pegawai
3	Pengembangan Sistem Pengajuan Cuti <i>Online</i> Pegawai di Rumah Sakit Umum Anwar Medika	Yusmanisari, dkk (2023)	Proses pengajuan cuti masih dilakukan secara manual menggunakan dokumen atau formulir fisik, dan tanpa adanya <i>database</i> khusus. Yang mengakibatkan petugas harus menginput data cuti dari banyak pegawai secara manual, yang memakan waktu lama dan meningkatkan beban kerja bagian kepegawaian	Sistem hanya mengakomodir satu jenis cuti saja, dan belum ada sistem verifikasi bertingkat. Dan pelaporan usulan cuti tanpa adanya <i>upload</i> dokumen pendukung

## 2.3 Teori Pendukung

### 2.3.1 Cuti

Cuti merupakan hak yang diberikan kepada setiap pegawai dalam suatu instansi atau perusahaan. Dengan cuti memungkinkan pegawai untuk tidak hadir dari pekerjaan dengan berbagai alasan, seperti untuk istirahat karena sakit, waktu melahirkan, pelaksanaan kewajiban keagamaan, dan keperluan lain

yang sesuai dengan ketentuan cuti yang telah ditetapkan oleh perusahaan atau instansi masing-masing (Rizki dan Pasaribu, 2021).

### **2.3.2 Sistem Informasi**

Sistem informasi dapat diartikan sebagai suatu rangkaian sistem yang dirancang dan dikembangkan oleh sumber daya manusia dengan tujuan utama menyajikan informasi. Di dalam struktur sistem ini terdapat beragam prosedur organisasi yang secara terinci dirancang untuk memberikan informasi yang relevan kepada para pengambil keputusan, memfasilitasi proses pengambilan keputusan yang lebih baik, dan secara keseluruhan mendukung operasional dan tujuan strategis dari suatu organisasi (Ladjamudin, 2005).

### **2.3.3 Aplikasi**

Aplikasi merupakan suatu perangkat lunak (*software*) yang dirancang dan dikembangkan untuk menjalankan fungsi atau tugas tertentu pada suatu sistem komputer atau perangkat lainnya. Definisi aplikasi secara umum adalah program yang berbentuk *software* yang dirancang untuk beroperasi di dalam suatu sistem tertentu, dengan tujuan memberikan bantuan atau kemudahan dalam melaksanakan berbagai macam aktivitas atau tugas yang dilakukan oleh manusia (Yanuar dan Senubekti, 2022).

### **2.3.4 Website**

*Website* adalah suatu *platform* perangkat lunak (*software*) yang menampilkan berbagai informasi dalam berbagai format seperti teks, tabel, gambar, suara, dan bentuk lainnya. Informasi-informasi ini disimpan di *web server* internet dan disajikan kepada pengguna dalam format *Hypertext* (Yanuar dan Senubekti, 2022).

### **2.3.5 HTML**

HTML (*HyperText Markup Language*) merupakan dasar atau pondasi dari bahasa pemrograman untuk sebuah halaman web. HTML muncul sebagai standar baru dari kemajuan dan berkembangnya internet. Pada awalnya internet hanya menampilkan teks secara monoton dan tanpa format visual, seperti teks

tanpa paragraf, menggunakan satu warna, tanpa gambar, serta tanpa tata letak visual seperti yang ditemukan dalam aplikasi seperti Microsoft Word. Sehingga situasi tersebut membuat pengalaman membaca menjadi sangat membosankan. Dengan munculnya pemrograman HTML, situasi ini mengalami perubahan signifikan seiring dengan evolusi teknologi dan informasi. HTML menjadi kunci dalam memberikan struktur, tata letak, dan elemen visual yang diperlukan untuk menciptakan halaman web yang lebih menarik, *user-friendly*, dan berfungsi dengan baik (Sepriano dan Ardiyansa, 2022).

### 2.3.6 CSS

CSS (*Cascading Style Sheets*) merupakan bahasa yang digunakan untuk mengatur tampilan halaman web, termasuk warna, tata letak, dan jenis huruf. Dengan menggunakan CSS pengembang web memiliki kemampuan untuk menciptakan halaman web yang responsif atau dapat beradaptasi dengan berbagai ukuran layar, memiliki desain yang konsisten dan efisien. Sehingga dengan penggunaan CSS akan mempermudah pengelolaan dan pengaturan halaman HTML karena dapat diterapkan pada seluruh halaman web (Ramadhani, dkk, 2023).

### 2.3.7 PHP

PHP merupakan bahasa pemrograman yang dirancang untuk dijalankan melalui halaman web, dan khususnya digunakan untuk memproses informasi di internet. PHP juga didefinisikan sebagai bahasa pemrograman *server-side* yang bersifat *open source* atau dapat digunakan secara gratis. PHP bukan hanya sekadar sebuah *script*, melainkan merupakan sebuah kerangka kerja yang terintegrasi dengan HTML, ditempatkan di sisi server untuk mengelola dan menyajikan data dengan efisien melalui halaman web (Sari, dkk, 2022).

### 2.3.8 JavaScript

JavaScript adalah sebuah bahasa skrip dinamis yang digunakan untuk meningkatkan interaktifitas pada halaman HTML yang bersifat statis. JavaScript memungkinkan pembuatan dokumen HTML yang tidak hanya menarik secara visual, tetapi juga menjadi lebih interaktif dan meningkatkan

pengalaman pengguna. Dengan JavaScript, pengembang dapat menciptakan program yang disajikan melalui antarmuka web, memungkinkan pengguna berinteraksi dengan berbagai elemen pada halaman tersebut. Penggunaan JavaScript ditandai dengan penggunaan *tag* yang diawali dengan `<script>` dan diakhiri dengan `</script>`, yang memungkinkan penyisipan kode JavaScript ke dalam halaman HTML untuk memberikan efek dinamis dan fungsionalitas yang lebih kaya (Risaldi, dkk, 2020).

### 2.3.9 MySQL

MySQL adalah sebuah sistem manajemen basis data yang menggunakan bahasa non-prosedural dan terstruktur yang memiliki kemampuan untuk mengakses dan memanipulasi data. Sifat terstruktur dari MySQL termanifestasi melalui penggunaan SQL (*Structured Query Language*), sebuah bahasa pemrograman yang mengikuti standar yang telah ditetapkan oleh organisasi internasional seperti ISO (*International Standards Organization*) dan ANSI (*The American National Standards Institute*) (Rohendi, 2015).

### 2.3.10 CodeIgniter





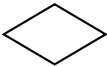


CodeIgniter adalah sebuah *framework* PHP yang mempercepat pengembangan aplikasi web berbasis PHP dengan menyediakan berbagai fitur dan *library*. Dengan CodeIgniter pengembang dapat menghemat waktu yang seharusnya digunakan untuk menulis kode program dari awal. Selain efisiensi waktu, struktur dan logika CodeIgniter memberikan kerapian dan keteraturan pada aplikasi yang dikembangkan. Hal ini memungkinkan pengembang fokus pada pengembangan fitur-fitur yang dibutuhkan tanpa harus menghabiskan banyak waktu untuk mengatur struktur dasar aplikasi. CodeIgniter, sebagai PHP *framework open source* yang mendukung pola MVC (*Model-View-Controller*), dirancang untuk menyediakan pendekatan sederhana dan efisien dalam menyelesaikan proyek pengembangan web. Kelebihan lain dari CodeIgniter adalah ukurannya yang relatif kecil, memberikan keunggulan dalam hal efisiensi sumber daya. Dibandingkan dengan *framework* PHP lain yang mungkin lebih besar dan memerlukan

sumber daya yang signifikan, CodeIgniter memungkinkan pengaturan untuk memuat hanya *library* yang diperlukan, menghasilkan aplikasi yang berjalan dengan ringan dan cepat (Hamonangan, 2021).

### 2.3.11 Flowchart

*Flowchart* atau yang sering disebut diagram alir program adalah komponen yang menguraikan dengan rinci langkah-langkah dari proses sebuah program. *Flowchart* dibuat dengan menggunakan simbol-simbol geometris dengan anak panah yang menghubungkan satu simbol dengan simbol lainnya. Sistem *flowchart* menggambarkan jalur data yang mengalir dari satu entitas organisasi atau satu perangkat pemroses ke entitas atau perangkat lainnya. Simbol-simbol *flowchart* dapat dilihat pada Tabel 2.1 (Rhosyid, 2022).

Tabel 2.2. Simbol *Flowchart*

No	Simbol	Keterangan
1		<i>Terminator</i> , simbol yang digunakan untuk awal ( <i>start</i> ) atau akhir ( <i>end</i> ) dari suatu alur program
2		<i>Flow/Arrow</i> , simbol yang digunakan untuk menghubungkan simbol 1 dengan simbol lainnya
3		<i>Input/Output</i> , simbol yang menyatakan proses <i>input</i> atau <i>output</i>
4		<i>Action/Process</i> , simbol yang melambangkan tindakan atau operasi yang dilakukan pada data
5		<i>Decision</i> , simbol yang menunjukkan cabang dalam alur program berdasarkan kondisi tertentu
6		<i>Document</i> , simbol yang digunakan untuk mewakili dokumen atau <i>file</i> yang digunakan dalam proses
7		<i>Display</i> , simbol yang digunakan untuk menampilkan informasi atau output kepada pengguna

### 2.3.12 Unified Modeling Language

*Unified Modeling Language* (UML) merupakan salah satu alat bantu di dunia pengembangan sistem yang berorientasi obyek. UML menawarkan berbagai keunggulan, dengan tiga karakteristik penting yang melekat pada UML,




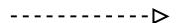
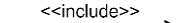



yaitu cetak biru, sketsa, dan bahasa pemrograman. Sketsa dalam UML berfungsi sebagai alat komunikasi untuk menyampaikan berbagai aspek dari sistem. Sebagai cetak biru, UML digunakan untuk membaca program dan menginterpretasikannya kembali dalam bentuk diagram. UML juga berperan sebagai bahasa pemrograman yang dapat mengubah diagram menjadi kode program yang dapat dieksekusi (Safira dan Purtiningrum, 2022).

### 2.3.13 Use Case Diagram

*Use case diagram* menggambarkan interaksi antara satu atau lebih aktor dengan sistem informasi yang sedang dikembangkan. Secara umum *use case* digunakan untuk mengidentifikasi fungsi-fungsi yang ada dalam sistem informasi dan menentukan siapa saja yang berhak untuk menggunakan fungsi-fungsi tersebut. Prinsip dasar dalam penamaan *use case* adalah untuk menggunakan nama yang sederhana dan mudah dipahami. Ada dua elemen utama dalam *use case*, yaitu definisi aktor dan deskripsi *use case*. Simbol-simbol *use case diagram* dapat dilihat pada Tabel 2.2 (Khairunnisa, dkk, 2019).

Tabel 2.3. Simbol *Use Case Diagram*





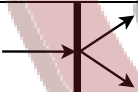
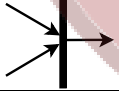

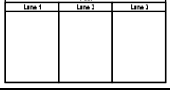
No	Simbol	Keterangan
1		<i>Actor</i> , simbol yang mewakili entitas diluar sistem yang berinteraksi dengan sistem
2		<i>Use Case</i> , simbol yang menunjukkan tindakan atau aktivitas spesifik yang dilakukan oleh <i>actor</i>
3		<i>Association</i> , simbol yang menunjukkan hubungan antara <i>actor</i> dan <i>use case</i> lainnya
4		<i>Generalisasi</i> , simbol yang menunjukkan hubungan antara dua <i>use case</i> dimana satu <i>use case</i> adalah lebih umum (induk) dan yang lainnya lebih spesifik (anak)
5		<i>Include</i> , simbol yang menunjukkan bahwa <i>use case</i> memasukkan fungsionalitas dari <i>use case</i> lainnya
6		<i>Extend</i> , simbol yang menunjukkan bahwa <i>use case</i> dapat menambahkan fungsionalitas optional ke <i>use case</i> lain



### 2.3.14 Activity Diagram

*Activity Diagram* disusun berdasarkan aliran alternatif dan aliran dasar dalam skenario yang tergambar dalam *Use Case Diagram*. Diagram ini menjelaskan interaksi antara aktor dan sistem dalam *Use Case Diagram*, sehingga dapat disimpulkan bahwa *Activity Diagram* dibangun berdasarkan struktur *Use Case Diagram* yang telah disusun. Simbol-simbol *activity diagram* dapat dilihat pada Table 2.3 (Yanuar dan Senubekti, 2022).

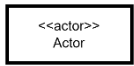
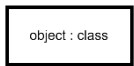



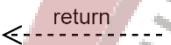
Tabel 2.4. Simbol Activity Diagram

No	Simbol	Keterangan
1		<i>Start Point</i> , merupakan awal dari aktivitas
2		<i>End Point</i> , merupakan akhir dari aktivitas
3		<i>Control Flow</i> , menunjukkan urutan aktivitas
4		<i>Activities</i> , menggambarkan suatu bisnis proses atau aktivitas
5		<i>Fork</i> / percabangan, digunakan untuk memecah aktivitas yang dilakukan secara paralel
6		<i>Join</i> / penggabungan, digunakan untuk menggabungkan kembali aktivitas yang dilakukan paralel
7		<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i>
8		<i>Swimlane</i> , pemisahan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

### 2.3.15 Sequence Diagram

*Sequence diagram* atau juga disebut diagram urutan, digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah interaksi yang terjadi secara berurutan dalam rentang waktu tertentu antara sejumlah objek sebagai respons terhadap suatu kejadian untuk menghasilkan output tertentu. Tabel simbol *Sequence Diagram* dapat dilihat pada Tabel 2.4 (Afrianto, dkk, 2020).

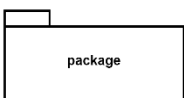

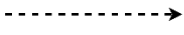


Tabel 2.5. Simbol *Sequence Diagram*

No	Simbol	Keterangan
1		<i>Actor</i> , merupakan eksternal sistem berupa orang atau sistem lain yang mendapatkan manfaat dari <i>use-case</i>
2		Object, merupakan sesuatu yang ikut dalam rangkaian kegiatan dengan mengirim atau menerima pesan
3		<i>Lifeline</i> , merupakan waktu hidup suatu objek dalam rangkaian kegiatan
4		<i>Execution Occurrence</i> , menandakan ketika suatu objek mengirim atau menerima pesan
5		<i>Request Message</i> , merupakan permintaan informasi dari satu objek ke objek lain
6		<i>Return Message</i> , merupakan permintaan informasi yang dikirim oleh suatu objek

### 2.3.16 Component Diagram

*Component diagram* menunjukkan organisasi dan hubungan ketergantungan antara kumpulan komponen dalam suatu sistem. *Component diagram* berfokus pada komponen sistem yang dibutuhkan dan ada di dalam sebuah sistem. Simbol-simbol *component diagram* dapat dilihat pada Tabel 2.5 (Afrianto, dkk, 2020).





Tabel 2.6. Simbol Component Diagram

No	Simbol	Keterangan
1		<i>Package</i> , merupakan sebuah bungkusan dari satu atau lebih komponen
2		<i>Component</i> , merupakan komponen sistem
3		<i>Depedency</i> , menunjukkan hubungan ketergantungan antara komponen, dengan panah mengarah ke komponen yang digunakan
4		<i>Interface</i> , sebagai antarmuka komponen agar tidak mengakses langsung komponen
5		<i>Link</i> , menunjukkan relasi antar komponen

### 2.3.17 Deployment Diagram

*Diagram deployment* atau juga disebut sebagai diagram distribusi, digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem. Diagram ini menunjukkan proses-proses yang berjalan dalam sistem dan hubungan relasinya. Simbol-simbol *deployment diagram* dapat dilihat pada Tabel 2.6 (Afrianto, dkk, 2020).

Tabel 2.7. Simbol *Deployment Diagram*

No	Simbol	Keterangan
1		<i>Component</i> , didalam <i>deployment diagram</i> komponen-komponen yang ada diletakkan didalam <i>node</i> untuk memastikan keberadaan posisi mereka
2		<i>Node</i> , menggambarkan bagian-bagian hardware dalam sebuah sistem
3		<i>Depedency</i> , menunjukkan hubungan ketergantungan antara <i>node</i> , dengan panah mengarah ke <i>node</i> yang digunakan
4		<i>Link</i> , menunjukkan relasi antar <i>node</i>

### 2.3.18 Blackbox Testing

Pengujian *Blackbox* merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Dengan pengujian *blackbox* memungkinkan pengembang perangkat lunak untuk mengidentifikasi serangkaian kondisi input yang mencakup semua persyaratan fungsional dari suatu program (Khairunnisa, dkk, 2019).