

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Rancang Bangun

Rancang Bangun adalah rangkaian proses yang mengartikan konsep sistem menjadi bahasa pemrograman, dengan tujuan untuk menjelaskan secara mendetail setiap komponen yang akan diimplementasikan. Sementara itu, pembangunan sistem yang merujuk pada aktivitas untuk mengembangkan sistem baru atau melakukan perbaikan pada sistem yang sudah ada, baik itu pada bagian tertentu maupun secara keseluruhan. Rancang bangun terdiri dari perancangan dan pembangunan sistem sebagai satu kesatuan (Surahman, Prastowo, & Aziz, 2022). Rancang bangun yaitu menggambarkan, merencanakan, dan membuat desain atau mengatur berbagai elemen yang terpisah menjadi satu kesatuan yang berfungsi (S. Wulandari, Jupriyadi, & Fadly, 2021). Merancang sistem bertujuan untuk memenuhi kebutuhan pengguna serta menyediakan panduan yang detail kepada programmer. Proses ini berfokus pada pembuatan desain sistem yang terstruktur, yang berfungsi sebagai peta untuk programmer dalam mengembangkan perangkat lunak. Desain sendiri mencakup berbagai komponen, mulai dari struktur data, alur kerja, hingga antarmuka pengguna, sehingga memastikan semua elemen sistem selaras dengan kebutuhan yang diidentifikasi.

Tujuan perancangan ini adalah untuk sebuah desain yang dapat dengan mudah diubah menjadi kode program. Dengan demikian, programmer memiliki tahapan proses yang jelas dan detail, yang bukan

hanya menggambarkan alur kerja sistem tetapi juga komponen-komponen teknis yang diperlukan. Sehingga dapat disimpulkan jika rancang bangun sistem adalah proses yang menerjemahkan suatu hasil analisis kebutuhan pengguna menjadi sebuah sistem perangkat lunak. Proses ini dapat digunakan untuk menciptakan sistem baru yang sepenuhnya baru atau memperbaiki sistem yang sudah ada, baik dalam skala kecil maupun besar.

Sehingga dapat disimpulkan jika Rancang Bangun merupakan serangkaian alur yang mengubah sebuah konsep sistem menjadi bahasa pemrograman, dengan tujuan untuk menjelaskan secara rinci bagaimana setiap komponen diimplementasikan. Tujuan utama perancangan ini yaitu untuk menghasilkan rancang bangun yang dapat dengan mudah diimplementasikan menjadi kode program.

2. Surat Perjalanan Daerah

Surat Perjalanan Dinas (SPD) merupakan berkas resmi yang diterbitkan oleh instansi untuk mengesahkan perjalanan dinas pegawai ke suatu lokasi tertentu (Mallisza, Hadi, & Aulia, 2022). Tugas kerja yang dilakukan memerlukan Surat Perjalanan Dinas (SPD) umumnya terdiri dari berbagai aktivitas penting yang harus dilakukan di luar lingkungan kantor utama (Vernanda, Purnawan, & Gusnanda, 2023). Kegiatan perjalanan dinas dapat berupa studi banding, di mana pegawai melakukan observasi dan pembelajaran di instansi. Selain itu, pegawai dapat ditugaskan untuk melakukan audit ke perusahaan cabang atau kantor cabang lain untuk memastikan kesesuaian operasional, keuangan, atau kinerja dengan standar yang telah ditetapkan oleh kantor pusat. Di samping itu. Tugas dinas lainnya yang juga memerlukan SPD termasuk pelatihan, seminar, atau sosialisasi

kebijakan baru yang diadakan di luar kota. SPD memiliki peran penting dalam administrasi instansi karena berfungsi sebagai bukti tertulis perjalanan, catatan pengingat, serta arsip yang mencatat aktivitas dan perkembangan instansi secara historis. Selain itu, SPD juga berperan sebagai pedoman kerja, di mana dokumen ini berfungsi sebagai bentuk surat keputusan yang memberikan kejelasan mengenai tujuan, waktu, dan detail perjalanan dinas, sehingga mendukung transparansi dan akuntabilitas dalam pelaksanaan tugas pegawai.

Dari penjelasan diatas mengenai Surat Perjalanan Dinas dapat ditarik kesimpulan bahwa Surat Perjalanan Dinas digunakan sebagai dokumen resmi yang diterbitkan untuk menugaskan karyawan melakukan perjalanan dinas.

3. Website

Website merupakan suatu sumber informasi dalam bentuk modern yang dapat diakses melalui media internet menggunakan browser. *Website* adalah kumpulan halaman yang terhubung satu sama lain, biasanya disimpan di *server* yang sama dan berisi sumber informasi (Nor Ramadha, 2022). Menurut Gregorius, *website* adalah sekumpulan halaman *website* memiliki koneksi dan memiliki file yang saling berkaitan. Didalam *website* memuat halaman atau *pages*, dengan sekumpulan halaman utama yang dikenal sebagai *homepage*. *Homepage* terletak di posisi paling atas, halaman terkait di bawahnya disebut sebagai *child pages*. Setiap halaman yang di bawah *homepage* biasanya berisi *hyperlink* yang mengarah dari halaman satu ke halaman yang lain (Anamisa & Mufarroha, 2020).

Website digunakan secara luas sebagai media komunikasi untuk memberikan suatu informasi atau layanan (Aziz, Sulaiman, Hassan, Zakaria, & Yaacob, 2021). *Website* memiliki sifat unik dikarenakan setiap halamannya dapat saling terhubung melalui *hyperlink* (Dermawan, Mashuri, Permadi, Gunawan, & Widiasih, 2021). *Website* biasanya tersimpan di server dan bisa dijalankan melalui jaringan, dengan menggunakan alamat unik yang dikenal dengan URL. *Website* umumnya terbagi menjadi tiga jenis utama yaitu *website* statis yang merupakan jenis *website* yang konten yang tidak diperbarui secara berkala, sehingga informasi di dalamnya tetap sama dari waktu ke waktu. *Website* statis ini sering kali digunakan untuk menampilkan profil perusahaan atau organisasi, di mana isi *website* hanya menampilkan informasi dasar tanpa perlu perubahan yang sering. *Website* Dinamis memiliki konten yang dapat diperbarui secara rutin oleh pemilik *website*. Jenis *website* dinamis populer di kalangan perusahaan atau individu yang membutuhkan pembaruan konten secara aktif, terutama untuk bisnis yang terhubung dengan internet. *Website* dinamis memungkinkan pemilik atau tim pengelola untuk mengedit dan menambah informasi baru sesuai kebutuhan.

Website Interaktif termasuk dalam kategori *website* dinamis namun memberikan peran lebih besar bagi pengguna untuk turut serta dalam memperbarui atau menambahkan informasi. Pada *website* interaktif, pengunjung dapat berinteraksi langsung, misalnya dengan memberikan komentar, mengunggah konten, atau berpartisipasi dalam forum. Contoh *website* ini adalah media sosial dan situs komunitas, di mana konten utamanya dihasilkan oleh kontribusi dari pengguna. Ketiga jenis *website* ini

memiliki karakteristik dan fungsi tersendiri yang disesuaikan dengan tujuan serta kebutuhan pengguna, dari yang sekadar menampilkan informasi dasar hingga menyediakan interaksi aktif bagi pengguna.

Dari penjelasan diatas mengenai website dapat ditarik kesimpulan bahwa *Website* adalah sumber informasi dalam bentuk *modern* yang dapat diakses melalui media internet menggunakan peramban (*browser*). *Website* memiliki sifat unik karena setiap halamannya dapat saling terhubung melalui *hyperlink*.

4. PHP

PHP atau yang disebut dengan *Hypertext Preprocessor* adalah bahasa *script* tingkat tinggi yang digunakan dalam dokumen HTML, dengan struktur kode mirip C, Java, dan Perl, namun PHP memiliki sejumlah fungsi khusus yang lebih spesifik (Sandria, Nurhayoto, Ramadhani, Harefa, & Syahputra, 2023). Bahasa PHP dirancang untuk mempermudah pengembang *web* yang dinamis dan interaktif dengan cara yang efisien. Sebagai bahasa pemrograman sisi *server* (*server-side scripting*), PHP memerlukan *web server* untuk dapat dijalankan. PHP akan memproses perintah di *server* dan akan mengirim hasilnya ke *browser* pengguna, sehingga php cocok untuk aplikasi *web* yang interaktif (Nor Ramadha, 2022). Dalam bahasa yang menggunakan *compiler*, kode program (atau *source code*) diubah menjadi *object code*, yaitu bentuk yang lebih sederhana dan ringan dibandingkan kode aslinya. Kemudian, *object code* ini dikompilasi menjadi sebuah program mandiri yang dapat dijalankan tanpa perlu bantuan program lain, seperti file EXE yang bisa langsung diimplementasikan oleh komputer. Sebaliknya, karena PHP merupakan bahasa interpreter, PHP dijalankan

langsung oleh *server* pada saat itu juga, tanpa perlu dikonversi menjadi bentuk lain atau menjadi file EXE.

Berikut ini adalah beberapa kelebihan dalam menggunakan:

- PHP yaitu bahasa *script* yang tidak memerlukan proses kompilasi, sehingga lebih praktis dan mudah untuk pengembangan *web*.
- PHP didukung oleh berbagai *web server* dengan konfigurasi yang cukup mudah.
- Penyempurnaan menggunakan PHP cukup mudah dilakukan karena adanya komunitas pengembang dan forum diskusi yang aktif, yang siap membantu jika terjadi kesusahan dalam pengembangannya.
- PHP dikenal mudah untuk dipahami, dengan banyaknya referensi yang tersedia sehingga memudahkan proses belajar dan implementasinya.
- Sebagai bahasa *open-source*, PHP bisa dijalankan di beberapa platform, serta mendukung eksekusi *runtime* melalui *console*, termasuk perintah sistem, yang memudahkan penggunaannya (Risawandi, 2019).

Dari penjelasan diatas mengenai PHP dapat disimpulkan bahwa PHP merupakan bahasa *scripting* tingkat tinggi yang diimplementasikan dalam dokumen HTML. Bahasa PHP dirancang guna memudahkan pengembang web untuk mengembangkan halaman *web* yang responsif dan interaktif secara efisien.

5. HTML (*HyperText Markup Language*)

HTML (*HyperText Markup Language*) merupakan sintaks pemrograman dasar yang menyusun struktur dan isi dari hampir semua halaman web. Pada halaman website, HTML berfungsi sebagai bahasa skrip dasar yang bekerja bersama berbagai bahasa pemrograman *scripting* lainnya (Nor Ramadha, 2022). Fungsi dari HTML untuk menciptakan dan menyusun struktur halaman *web*. Meskipun HTML adalah sebuah bahasa markah, bukan bahasa pemrograman, sehingga tidak mampu membangun fungsi dinamis. Seperti *Microsoft Word*, HTML memungkinkan pengguna untuk mengatur dan memformat teks (Kirsan, Arisa, & Putra, 2022).

Dalam pembuatan halaman web memiliki komponen penting yaitu HTML (*Hypertext Markup Language*) dan CSS (*Cascading Style Sheets*). HTML berfungsi untuk menyusun struktur dasar halaman, sementara CSS berfungsi untuk mengatur desain visualnya dengan bahasa markup yang memberi label pada konten seperti “paragraf”, “daftar”, “tabel” dan lain – lain. Disisi lain, CSS digunakan untuk mengatur tata letak visual dari halaman *web* pada berbagai perangkat (Rumetor, Sentinuwo, & Dirko G.S. Ruindungan, 2021).

Sehingga dapat ditarik kesimpulan jika HTML adalah bahasa standar dari semua halaman *web* bersama dengan CSS dan *JavaScript*, yang memungkinkan pengembang untuk membuat situs *website* yang interaktif dan menarik.

6. Laravel

Framework Laravel yang dikembangkan oleh Taylor Otwell pada tahun 2011, berbasis pada *Symfony 2*, dan dengan cepat menjadi salah satu

framework PHP paling populer bagi pengembang web dikarenakan fleksibel dan mudah untuk digunakan (Laaziri, Benmoussa, Khouli, Larbi, & Yamami, 2019). Laravel merupakan salah satu *framework* yang populer saat ini (Alfarisi, Priandika, & Puspaningrum, 2023). Didesain untuk menyederhanakan proses pengembangan. Laravel menawarkan berbagai fitur yang memungkinkan pengembang untuk mengembangkan aplikasi *web* yang efektif. *Framework* ini memiliki sintaksis yang jelas, dokumentasi yang lengkap, serta dukungan komunitas yang besar, menjadikannya pilihan utama bagi pengembang yang ingin mempercepat proses pengembangan aplikasi mereka.

Laravel menyediakan fitur utama seperti sistem *routing* yang fleksibel, sistem pengelolaan otentikasi yang terintegrasi, dan kapabilitas migrasi *database* yang mempermudah pengelolaan skema *database*. Salah satu fitur unggulan Laravel adalah *Eloquent*, yaitu ORM bawaan yang menyediakan pengembang untuk berinteraksi dengan database menggunakan sintaksis PHP sederhana dan mudah dipahami. Selain itu, Blade, mempermudah pengembangan antarmuka dengan sintaksis yang mudah dimengerti. *Framework* ini juga menyediakan alat untuk pengelolaan antrian, sistem notifikasi, dan *caching*, semuanya dirancang untuk meningkatkan kecepatan dan kinerja aplikasi. Serta dukungan ekosistem paket tambahan yang solid dan komunitas pengembang yang aktif, Laravel menjadi pilihan ideal untuk mengembangkan aplikasi web dengan cepat dan efisien.

Dari penjelasan Laravel seperti diatas dapat ditarik kesimpulan jika Laravel merupakan salah satu jenis *framework* pengembangan aplikasi *web*.

Laravel dirancang sedemikian sederhana untuk memudahkan penggunaanya dalam mengembangkan aplikasi web. Laravel menyediakan berbagai fitur utama seperti sistem *routing*, manajemen otentikasi yang terintegrasi, dan kemampuan migrasi *database* yang mempermudah pengelolaan skema *database*.

7. MySQL

MySQL yaitu suatu sistem *database open-source* yang hingga kini masih populer dan sering diterapkan dalam pengembangan *website* (Risawandi, 2019). Dalam MySQL, data disimpan dalam objek *database* yang disebut dengan tabel. Setiap tabel terdiri dari entri-entri yang saling terhubung, yang disajikan dalam bentuk hubungan antara kolom dan baris. MySQL adalah program server *database* yang dapat menerima dan mengirim data dengan cepat, mendukung banyak pengguna dengan mengaplikasikan perintah SQL (*Structure Query Language*). MySQL mempunyai dua tipe lisensi diantaranya, yaitu *Shareware* dan *FreeSoftware* (Nor Ramadha, 2022). MySQL yang umumnya digunakan adalah MySQL *FreeSoftware* yang berlisensi GNU/GPL (*General Public License*).

MySQL merupakan *server database* gratis, yang memungkinkan pengguna untuk menggunakannya tanpa perlu membayar lisensi. Michael Widenius merupakan *programmer* yang mengembangkan MySQL pertama kali. MySQL memiliki program yang dapat mengeksplorasi *database* sebagai *server* dan *client*. Jadi, MySQL merupakan *database* yang bisa berperan sebagai *client* maupun *server*, tergantung pada kebutuhan.

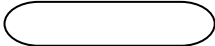
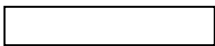
Dari penjelasan yang telah ada dapat disimpulkan jika MySQL adalah sistem *database* yang *open-source* dalam penggunaannya. MySQL dapat disimpan ke dalam tabel. Sebuah tabel terdiri dari berbagai entri yang saling berkaitan, yang disusun dalam bentuk hubungan antara kolom dan baris.

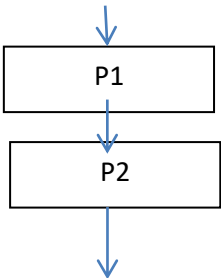
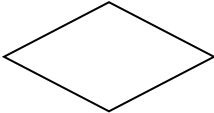
8. FlowChart

Bagan alir (*flowchart*) adalah metode analisis visual yang diterapkan untuk menggambarkan ragam aspek sistem informasi secara jelas. *Flowchart* menggambarkan bagaimana proses bisnis berlangsung dan bagaimana aliran dokumen dalam suatu organisasi. Dengan *flowchart*, alur sistem, prosedur, dan pengendalian internal yang akan diimplementasikan dapat digambarkan secara visual (Tuasamu et al., 2023). *Flowchart* adalah representasi grafik dari prosedur dari suatu program. *Flowchart* memudahkan pemecahan masalah menjadi bagian yang lebih sederhana serta membantu menganalisa berbagai alternatif dalam operasionalnya (Indra Ava Dianta, S.Kom., 2021).

Fungsi dari *flowchart* adalah untuk merancang program dan menggambarkan alur program. Oleh karena itu, *flowchart* harus menggambarkan komponen-komponen dalam bahasa pemrograman seperti pada tabel 2.1. mengenai lambang *flowchart*..

Tabel 2. 1. Lambang Flowchart

Deskripsi	Lambang
Mulai / Selesai (Terminator)	
Proses	

Deskripsi	Lambang
Jalur	
Masukan/Keluaran	
Sequence Proccess	
Percabangan	
Memanggil Prosedut / Fungsi	

Sumber: (Indra Ava Dianta, S.Kom., 2021)

Dari penjelasan mengenai *flowchart* diatas dapat ditarik kesimpulan jika *flowchart* yaitu metode analisis visual yang dimanfaatkan untuk merepresentasikan beberapa aspek sistem informasi secara jelas. Dengan *flowchart*, alur sistem, prosedur, dan pengendalian internal yang diterapkan oleh perusahaan dapat digambarkan secara visual.

9. Use Case Diagram

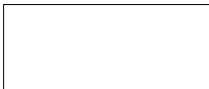
Use case diagram adalah jenis diagram UML yang digunakan untuk menunjukkan interaksi antara sistem dan pengguna dalam lingkungan tertentu. Diagram ini menampilkan fungsionalitas sistem secara visual, sehingga lebih mudah bagi pengembang perangkat lunak untuk

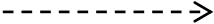
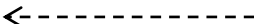
berkomunikasi dengan pengguna (Pranoto, Sutiono, Sarifudin, & Nasution, 2024). Kasus penggunaan menentukan fungsi sistem informasi dan siapa yang berhak menggunakannya. Penamaan *use case* dibuat semudah mungkin dan mudah dipahami. Aktor dan *use case* adalah dua komponen utama dalam *use case* yaitu sebagai berikut :

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri; simbol aktor berbentuk orang, tetapi aktor tidak selalu orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar aktor atau unit.

Berikut ini adalah lambang yang digunakan didalam membuat *use case* yang dapat dilihat pada tabel 2.2.

Tabel 2. 2. Lambang Use Case Diagram

Simbol	Nama	Keterangan
	<i>Use Case</i>	Fungsionalitas yang disediakan oleh sistem berfungsi sebagai unit yang saling berkomunikasi melalui pertukaran pesan antara unit atau aktor.
	<i>Aktor</i>	Sistem lain yang berinteraksi dengan sistem yang akan di kembangkan diluar sistem
	<i>Subject</i>	Berfungsi untuk mendeklarasikan lingkup suatu objek

Simbol	Nama	Keterangan
	<i>Association Relationship</i>	Sebagai penghubung antara aktor dan use case
	<i>Include Relationship</i>	Hubungan use case tambahan yang memerlukan use case utama untuk dijalankan terlebih dahulu, dengan panah menunjuk ke use case yang harus dijalankan lebih dulu.
	<i>Exclude Relationship</i>	Hubungan use case tambahan yang dapat berdiri sendiri, dengan anak panah menuju use case yang dituju.
	<i>Generalization relationship</i>	Menunjukkan generalisasi dari use case ke yang lebih umum atau general

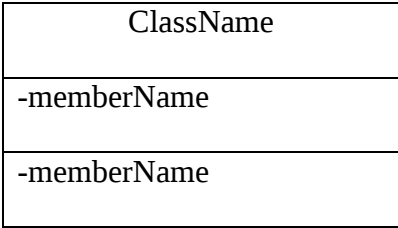
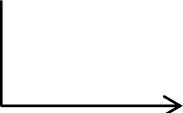
Sumber : (Ryanda, 2023)

10. Class Diagram

Class Diagram adalah kumpulan objek yang memiliki struktur, perilaku, dan semantic/kata yang sama. *Class Diagram* ini menggambarkan struktur sistem secara detail, termasuk kelas-kelas yang ada dalam sistem, atribut, metode, dan hubungan antar objek (Pranoto et al., 2024). Kelas memiliki atribut dan metode (operasi). Atribut merujuk pada variabel yang menggambarkan properti-properti dalam kelas tersebut, biasanya berupa teks dalam satu baris, sedangkan metode adalah fungsi yang dimiliki oleh kelas yang digambarkan dengan simbol-simbol dalam diagram kelas. Penggunaan *Class Diagram* terkait dengan struktur data sistem atau bisa juga menggantikan ERD dalam menggambarkan diagram rekayasa

perangkat lunak secara tradisional (Ryanda, 2023). Berikut ini adalah lambang yang digunakan untuk membuat *class diagram* yang dapat dilihat pada tabel 2.3.

Tabel 2. 3. Lambang Class Diagram

Simbol	Keterangan
	Hubungan antar kelas dengan pengertian umum. kelas pada sistem
	
	interface
	Hubungan antar kategori yang memiliki arti umum
	Hubungan antar kelas artinya satu kelas digunakan oleh yang lainnya
	Hubungan antar kelas dengan pengertian yang umum
	Hubungan antar kelas yang saling ketergantungan
	Hubungan antar class yang semua memiliki bagian tersebut.

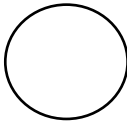
Sumber : (Ryanda, 2023)

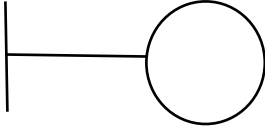
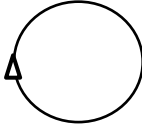
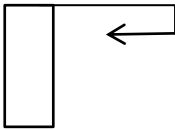
11. Sequence Diagram

Sequence Diagram adalah jenis diagram dalam pemodelan sistem yang digunakan untuk menggambarkan interaksi antar objek dalam urutan waktu tertentu. Diagram ini menggambarkan bagaimana objek-objek yang terlibat dalam suatu *use case* saling berkomunikasi melalui pesan. Pesan ini bisa berupa permintaan atau respons yang dikirimkan antar objek untuk melaksanakan fungsi tertentu.

Sequence Diagram bersifat dinamis karena menggambarkan perubahan dan aliran pesan yang terjadi secara berurutan, sesuai dengan urutan waktu. Setiap objek biasanya diwakili dengan garis vertikal, dan pesan antar objek digambarkan dengan panah horizontal yang menunjukkan aliran komunikasi atau interaksi. Urutan pesan yang dikirimkan menggambarkan bagaimana objek-objek berinteraksi dalam konteks proses atau skenario yang spesifik, sehingga diagram ini sangat berguna untuk memvisualisasikan alur eksekusi dalam sistem perangkat lunak. Berikut ini adalah lambang yang digunakan dalam sequence diagram yang ditampilkan pada tabel 2.4.

Tabel 2. 4. Lambang Sequence Diagram

Simbol	Nama	Kegiatan
	<i>Entity Class</i>	Kumpulan kelas-kelas

Simbol	Nama	Kegiatan
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi interface
	<i>Control Class</i>	Objek yang berisi logika bertanggung jawab atas entitas
	<i>Recurisive</i>	Pengiriman pesan
	<i>Message</i>	Mengirim pesan
	<i>Activation</i>	Saat melakukan aksi, panjang bidang sama dengan durasi aksi
	<i>Lifeline</i>	Garis titik yang terhubung dengan objek

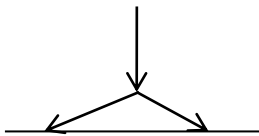
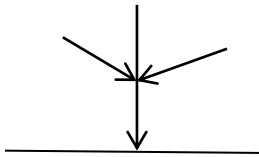
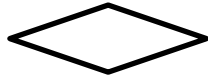
Sumber : (Ryanda, 2023)

12. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan alur kerja (*workflow*) atau aktivitas dalam sebuah sistem, proses bisnis, atau menu pada perangkat lunak. Hal yang perlu diperhatikan adalah bahwa diagram ini menggambarkan aktivitas yang dilakukan oleh sistem, bukan oleh aktor. Jadi, aktivitas yang ditampilkan adalah yang dilakukan oleh sistem itu sendiri (Malius, Apriyanto, & Ali Hakam Dani, 2021). *Activity Diagram* berfungsi sebagai representasi dari serangkaian alur kerja yang

menggambarkan setiap proses yang terjadi dalam sistem (Ryanda, 2023). Berikut ini adalah lambang *Activity Diagram* yang dapat dilihat pada tabel 2.5.

Tabel 2. 5. Lambang Activity Diagram

Simbol	Nama	Keterangan
	<i>Start Point</i>	Mulainya suatu proses
	<i>End Point</i>	Berakhirnya suatu proses
	<i>Activity</i>	prosesnya
	<i>Fork</i>	Proses paralel dan menggabungkan dua proses paralel menjadi satu
	<i>Join</i>	dekomposisi
	<i>Decision Point</i>	Pilihan untuk mengambil sebuah keputusan sesuai kondisi
	<i>Swimlane</i>	Menunjukkan pelaku

Sumber : (Ryanda, 2023)

13. RAD

RAD merupakan kombinasi dari berbagai teknik yang terorganisir . RAD menggabungkan metode *prototyping* dan teknik terstruktur lainnya guna memahami kebutuhan pengguna dan merancang sistem informasi.

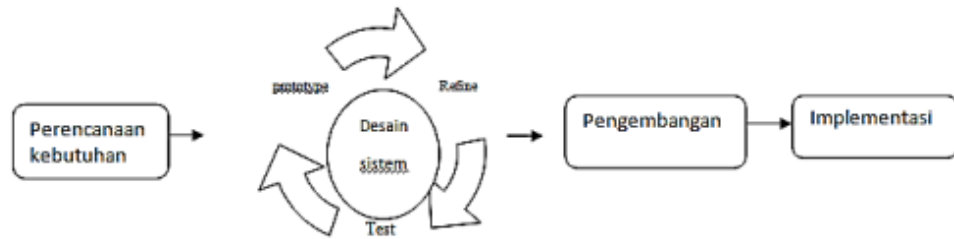
Pendekatan ini juga fokus pada siklus pengembangan yang cepat, umumnya berlangsung antara 60 hingga 90 hari, dengan pembangunan sistem yang berbasis komponen. RAD yaitu metode yang memanfaatkan pendekatan pengembangan bertahap dan berulang, namun menekankan pada batas waktu dan efisien biaya sesuai kebutuhan (Utami & Zein, 2023). Metode RAD dalam pengembangan dikenal dengan kecepatan prosesnya. Ini terjadi karena semua pihak terkait, baik pengguna maupun pengembang, terlibat aktif dalam setiap tahap hingga hasil akhir tercapai. Menurut (Pricillia & Zulfachmi, 2021) RAD adalah pendekatan pengembangan sistem yang berfokus pada objek, mencakup metode pengembangan dan perangkat lunak yang mendukung proses tersebut.

RAD dirancang untuk mempercepat waktu yang dibutuhkan dalam tahapan pengembangan sistem yang masih tradisional, terutama antara fase perancangan dan implementasi. Tujuan akhirnya adalah untuk memenuhi kebutuhan bisnis yang cepat berubah. Model RAD memiliki 3 tahapan yaitu :

- a. Perencanaan Kebutuhan (*Requirement Planning*), pengguna dan analis sistem berdiskusi untuk membahas tujuan sistem dan kebutuhan data yang diperlukan. Tahap ini krusial karena melibatkan kerjasama yang erat antara kedua belah pihak.
- b. Desain Sistem (*System Design*), partisipasi aktif dari pengguna diperlukan untuk memastikan desain sistem sesuai dengan ekspektasi mereka. Pengguna dapat memberikan *feedback* langsung jika terdapat ketidaksesuaian dalam rancangan, dan sistem dirancang berdasarkan dokumentasi kebutuhan yang telah disusun sebelumnya. Output dari

tahap ini adalah spesifikasi perangkat lunak yang mencakup struktur keseluruhan sistem, struktur data, dan elemen-elemen lainnya.

- c. Pengembangan, di mana prototipe yang telah disetujui dikembangkan menjadi aplikasi yang lebih lengkap dan fungsional. Pada tahap ini, pengembang menggunakan berbagai alat bantu seperti *framework*, generator kode, atau *library* untuk mempercepat proses pengembangan dan meningkatkan efisiensi kerja. Pengujian dilakukan secara paralel dengan pengkodean untuk memastikan bahwa setiap fungsi sistem berjalan dengan baik dan sesuai dengan spesifikasi yang telah disepakati. Selain itu, masukan dari pengguna yang diterima selama tahap ini dapat langsung diintegrasikan ke dalam pengembangan tanpa mengganggu alur kerja utama. Pendekatan ini memungkinkan aplikasi berkembang secara dinamis sambil tetap menjaga kualitas dan memenuhi kebutuhan pengguna dengan lebih cepat.
- d. Implementasi (*Implementation*), pengembang mulai membangun program sesuai dengan rancangan yang telah disetujui oleh pengguna dan analis. Pada awalnya diterapkan, program diuji untuk memastikan fungsionalitasnya sesuai dengan apa yang sudah disepakati sebelumnya. Pengguna kemudian dapat memberikan masukan akhir dan memberikan persetujuan kepada sistem yang telah dibuat.



Gambar 2. 1. Model Rapid Application Development (RAD)

Sumber: (P, Muhammad Dedi Irawan, & Ahyat Perdana Utama, 2022)

14. Sampel Penelitian

Sampel merujuk pada sebagian elemen atau karakteristik yang ada dalam suatu populasi, yang dipilih menggunakan metode tertentu, dengan tujuan agar data yang diperoleh dapat mencerminkan keseluruhan populasi tersebut (N. Wulandari & Wahyudi, 2022). Penentuan jumlah responden dapat dihitung dengan mengaplikasikan persamaan slovin. Rumus Slovin yaitu suatu persamaan yang digunakan untuk menentukan jumlah sampel yang diperlukan agar dapat mewakili populasi secara keseluruhan (N. Wulandari & Wahyudi, 2022). Berikut ini adalah rumus Slovin:

$$n = \frac{N}{1 + Ne^2}$$

Keterangan:

n = total sampel

N= total populasi

e = persentase pelanggaran dengan persentase 0,1

15. *Functional Testing*

Pengujian fungsional adalah bagian dari proses jaminan kualitas yang menekankan pengujian berdasarkan spesifikasi komponen perangkat lunak yang diuji (Susanto, Nugraheni, & Widyarini, 2021). Proses ini melibatkan

pemberian data masukan dan pengecekan hasil keluaran, tanpa banyak memperhatikan struktur internal program. Pengujian ini berfokus pada evaluasi layanan yang harus disediakan oleh sistem, cara sistem berinteraksi dengan data masukan, serta perilaku sistem dalam kondisi atau situasi tertentu. Umumnya dalam tes fungsionalitas menggunakan metode *blacbox testing*.

Black box testing adalah metode untuk menguji sistem yang menilai fungsi dari sistem tanpa mempertimbangkan cara kerja dari perangkat lunak tersebut (Wijaya & Astuti, 2021). Tujuannya untuk menemukan ketidaksesuaian fungsi, masalah pada antarmuka, kesalahan dalam struktur data, masalah performa, serta kesalahan pada proses awal dan akhir. Metode ini dilakukan berdasarkan spesifikasi kebutuhan sistem, di mana pengujian dilakukan tanpa perlu mengakses atau menulis kode program (McLaughlin dan Sarjoughian, 2020:2196-2207). *Black box* juga disebut sebagai pengujian fungsional berdasarkan spesifikasi dari pengguna dan penguji sistem tidak memiliki akses untuk ke kode program dari perangkat lunak yang dibuat (Praniffa et al., 2023).

Pengujian ini dirancang berdasarkan persyaratan sistem yang telah ditentukan. Semakin kompleks persyaratan yang ada, semakin banyak skenario pengujian yang perlu dikembangkan untuk memastikan model berfungsi sesuai harapan. Dalam pengujian *black box*, jumlah data uji bisa diperkirakan dengan pertimbangan jumlah inputan data yang akan diuji, untuk input yang harus dipatuhi, serta batasan input, baik batas maksimum maupun minimum, yang sesuai dengan spesifikasi yang telah ditentukan (Halawa & Saifudin, 2023).

Sehingga dapat ditarik kesimpulan jika bahwa metode ini berfokus pada pengujian fungsionalitas perangkat lunak berdasarkan spesifikasi yang telah ditetapkan, tanpa melibatkan akses atau peninjauan kode program dengan menggunakan metode *blackbox testing*. Pengujian ini dirancang berdasarkan kebutuhan sistem dan semakin kompleks persyaratan, semakin banyak skenario pengujian yang harus dikembangkan.

16. Usability Accepten Testing

User Acceptance Testing adalah metode tes yang melibatkan pengguna langsung untuk berinteraksi dengan sistem dan memvalidasi apakah setiap fungsi yang ada beroperasi sebagaimana mestinya (Wahyudi, Fahrullah, Alameka, & Haerullah, 2023). UAT (*User Acceptance Testing*) adalah metode tes aplikasi yang dilakukan oleh pengguna untuk memastikan aplikasi memenuhi kebutuhan dan harapan pengguna (W. Wulandari, Nofiyani, & Hasugian, 2023). *Acceptance testing* adalah pengujian yang dilakukan oleh pengguna dengan pendekatan *black box* untuk memastikan sistem memenuhi spesifikasi yang telah ditetapkan. Tujuan *User Acceptance Testing* (UAT) untuk memastikan setiap tahapan pengembangan perangkat lunak yang dilakukan memenuhi kebutuhan pengguna (Pillai & Hemamalini, 2022). Kerangka kerja yang diusulkan juga mendukung penggunaan alat otomatisasi untuk pengujian perangkat lunak. UAT ini akan mengoptimalkan penggunaan kembali kode yang ada guna meningkatkan kualitas perangkat lunak secara keseluruhan.

User Acceptance Testing (UAT) adalah metode tes yang dilakukan oleh *end user*. Hasil dari pengujian ini berfungsi sebagai bukti bahwa sistem mampu mendukung kebutuhan pengguna secara efektif. UAT diterapkan

pada tahap pengembangan perangkat lunak untuk memastikan bahwa sistem tidak hanya sesuai dengan spesifikasi teknis, tetapi juga memenuhi kebutuhan nyata pengguna. Pengujian sistem berfokus pada evaluasi fungsionalitas sistem, memastikan bahwa sistem bekerja sesuai dengan perannya dan menghasilkan output yang diharapkan. Pengujian perangkat lunak dilakukan melalui *User Acceptance Testing* (UAT) menggunakan metode pengujian *black box* untuk memastikan jika sistem berjalan sesuai dengan kriteria yang telah ditetapkan. Pengujian ini juga dilengkapi dengan pembagian kuesioner kepada pengguna yang nantinya hasil dari kuisisioner tersebut dihitung persentasenya dan akan dicocokkan dengan nilai yang ada di skala likert.

B. Kajian Empiris

Penulisan skripsi ini, peneliti merangkum informasi dari penelitian terdahulu untuk sumber referensi yang mendukung kajian dan analisis yang dilakukan, perbandingan. Berikut adalah tinjauan pustaka yang saya gunakan sebagai berikut:

1. Menurut (Puspitasari & Rakhma Devi, 2021), penelitian mereka membahas bahwa dalam membuat SPD (Surat Perjalanan Dinas) yang masih dilakukan secara konvensional cenderung memakan waktu lama. Penelitian ini menghasilkan rancangan sistem terkomputerisasi untuk SPD yang bertujuan memudahkan pegawai untuk mengelola dan merencanakan perjalanan dinas. Sistem ini mendukung pembuatan surat dinas, pengelolaan biaya, administrasi, serta pencetakan surat dan laporan tugas. Dengan adanya sistem ini, administrator dapat lebih mudah menginput data pegawai yang akan menjalankan perjalanan dinas.

2. Menurut (Dwi Yatma & Fikri Zulfikar, 2023) dalam penelitiannya, membahas mengenai permasalahan mekanisme pengurusan perizinan perjalanan dinas terkendala oleh atasan yang sedang tidak berada di tempat sehingga perizinan bisa tertunda dan proses *reimbursement* belum terintegrasi dengan sistem. Adapun hasil dalam penelitian ini yaitu pembuatan aplikasi *E-Reimbursement* berbasis *website* dengan metode *Rapid Application Development* (RAD) berjalan dengan baik.
3. Menurut (Aji & Santoso, 2022), penelitian mereka membahas masalah pada pengelolaan surat perjalanan dinas di BPS Kabupaten Sragen, yang mencakup waktu pembuatan surat yang lama, ketiadaan pengarsipan data surat perjalanan dinas serta laporan, dan tidak adanya rekapitulasi biaya perjalanan dinas. Penelitian ini menghasilkan suatu sistem yang mampu membuat dan mencatat data surat serta menyediakan rekapitulasi jumlah biaya perjalanan dinas. Dari hasil uji coba, sistem ini membutuhkan rata-rata waktu 50,24 detik untuk pembuatan surat tugas dalam 10 kali percobaan, dengan skor SUS sebesar 83,5, menunjukkan bahwa sistem berfungsi dengan baik.
4. Menurut (Jansen Sinla, 2019), penelitiannya membahas bahwa sistem yang belum menggunakan digital menyulitkan petugas ketika melakukan pengelolaan, penginputan, dan pencarian data, sehingga membutuhkan waktu cukup lama. Penelitian ini menghasilkan aplikasi Surat Perintah Perjalanan Dinas (SPD) berbasis *website*, dirancang untuk membantu pegawai dalam pembuatan dan pengelolaan SPD.
5. Menurut (Widiyansa, 2020), penelitiannya membahas bahwa pembuatan surat perjalanan dinas masih memanfaatkan metode tradisional dengan

aplikasi *Microsoft Word* dan *Microsoft Excel*. Proses ini mengharuskan pengguna untuk mengedit setiap saat pegawai akan melakukan perjalanan dinas, yang rentan terhadap kesalahan manusia. Hasil penelitiannya adalah rancangan aplikasi surat perjalanan dinas yang memudahkan pegawai dalam melakukan permohonan perjalanan dinas secara efektif dan efisien, mengurangi penggunaan kertas, serta meminimalkan risiko hilangnya surat perjalanan dinas.

C. Kerangka Berpikir

RUMUSAN MASALAH

1. Bagaimana cara merancang dan membangun aplikasi Surat Perjalanan Dinas di Lingkup Dinas TPHP Kabupaten Magetan berbasis *website*?
2. Bagaimana mengimplementasikan aplikasi Surat Perjalanan Dinas di Lingkup Dinas TPHP Kabupaten Magetan berbasis *website*?
3. Bagaimana hasil pengujian sistem menggunakan metode *black box testing* dan *User Acceptance Testing*?



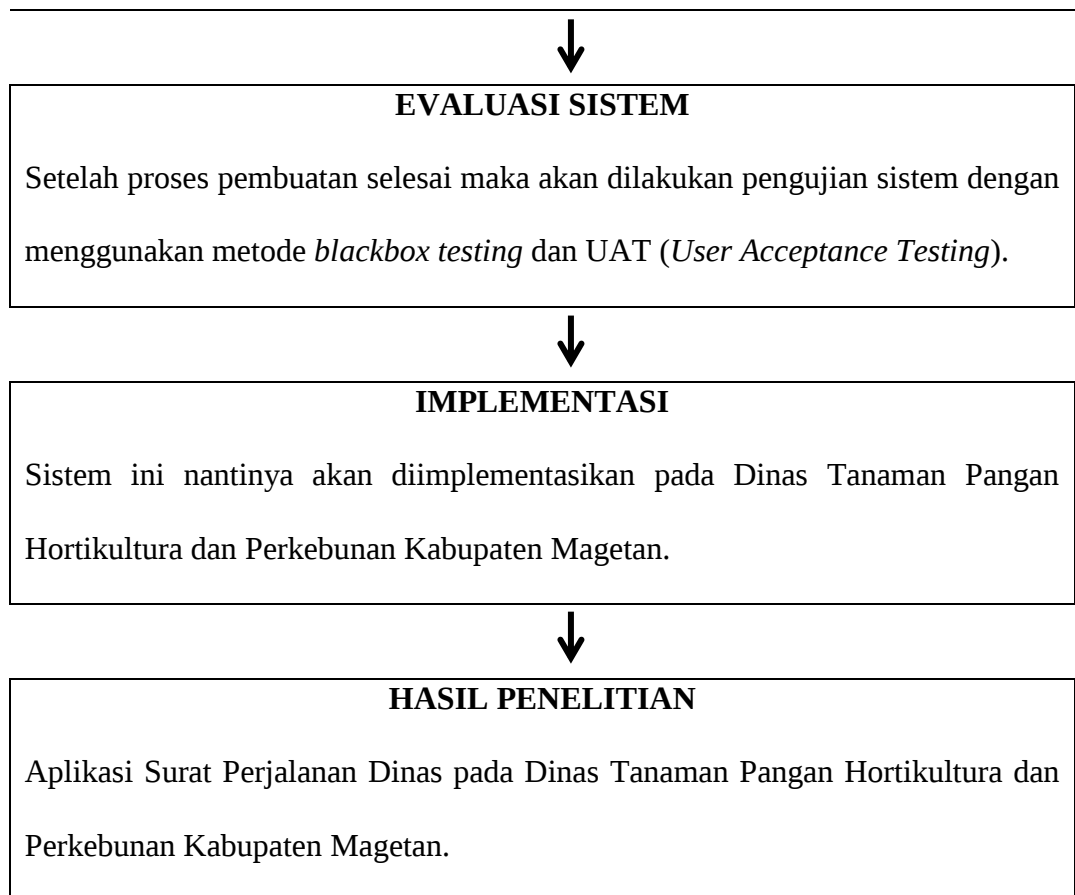
METODE PENELITIAN

Metode RAD (*Rapid Application Development*) digunakan dalam penelitian ini. Dalam RAD terdiri dari beberapa tahapan yaitu perencanaan, desain sistem, pengembangan dan implementasi.



PENGEMBANGAN SISTEM

Sistem ini dibuat dengan menggunakan *framework* laravel dan html sebagai visual sistem. Sistem *database* menggunakan MySQL



Gambar 2.2. Kerangka Berpikir