

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem dapat didefinisikan sebagai suatu kerangka kerja yang terdiri dari beberapa komponen atau elemen yang saling berhubungan dan berinteraksi secara sinergis untuk mencapai tujuan bersama. Komponen-komponen ini, baik berupa entitas fisik maupun konseptual, saling bergantung dan terintegrasi dalam suatu struktur yang teratur. Dalam konteks yang lebih luas, sistem dapat dipahami sebagai suatu kesatuan yang holistik, di mana setiap bagiannya berkontribusi terhadap fungsi keseluruhan secara efektif dan efisien. Dengan demikian, sistem tidak hanya sekadar kumpulan elemen yang terpisah, tetapi juga merupakan suatu entitas yang kompleks dan dinamis dengan tujuan yang terdefinisi dengan jelas.

2.1.1 Pengertian Aplikasi

Secara umum, aplikasi dapat didefinisikan sebagai perangkat lunak yang dirancang untuk memenuhi kebutuhan spesifik pengguna dalam menjalankan tugas tertentu. Aplikasi ini terdiri dari serangkaian jendela dan objek yang terintegrasi untuk memfasilitasi interaksi pengguna, mulai dari input data, pemrosesan, hingga penyajian hasil dalam bentuk laporan. Dengan kata lain, aplikasi merupakan alat bantu yang siap pakai dan disesuaikan dengan kemampuan yang dimilikinya untuk meningkatkan efisiensi dan produktivitas pengguna dalam menyelesaikan berbagai aktivitas. Aplikasi bisa berisi: -Suatu antarmuka: Menu, window, dan control window di mana user berinteraksi langsung dengan aplikasi. -Proses logika aplikasi: Kejadian (event) dan fungsi skrip yang dibuat sebagai logika aplikasi, validasi dan proses lainnya [3].(2.7)

2.1.2. Konsep Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu kesatuan yang terdiri dari individu, proses, dan teknologi yang bekerja secara terpadu untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan data menjadi informasi yang relevan dan berguna bagi suatu organisasi atau perusahaan. Sistem ini tidak hanya melibatkan komponen teknologi semata, tetapi juga interaksi antara manusia dan komputer dalam menjalankan fungsinya. Dalam pandangan Mulyani (2016:16), sistem informasi modern telah terkomputerisasi, di mana kolaborasi antara manusia dan komputer menjadi kunci dalam mengoptimalkan pengelolaan dan pemanfaatan informasi untuk mendukung pengambilan keputusan dan pencapaian tujuan organisasi.

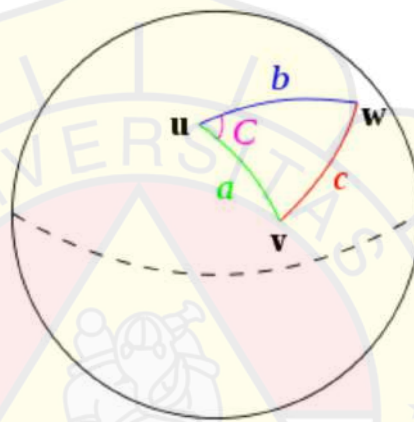
2.1.3. Rancang Bangun Sistem Informasi

Menurut Yuntari (2017), "rancang" merujuk pada proses transformasi hasil analisis sistem menjadi instruksi pemrograman yang rinci, menjelaskan implementasi setiap komponen sistem. Proses ini menghasilkan spesifikasi mendetail yang menjadi acuan dalam pengembangan sistem baru. Sementara itu, Ladjamuddin (2002) mendefinisikan "bangun" sebagai aktivitas menciptakan sistem baru atau melakukan modifikasi pada sistem yang sudah ada, baik secara menyeluruh maupun sebagian. Dengan demikian, "rancang" berfokus pada perencanaan dan spesifikasi, sedangkan "bangun" mengacu pada tindakan implementasi dan realisasi sistem tersebut.

2.1.4 Metode Haversine

Metode Haversine merupakan sebuah metode yang digunakan dalam sistem navigasi dimana metode ini akan menghasilkan sebuah perhitungan jarak antara dua

titik dari garis bujur (longitude) dan garis lintang (latitude) [1]. Metode Haversine merupakan suatu cara penentuan jarak dari titik koordinat berdasarkan posisi garis lintang dan garis bujur atau dalam aplikasinya kini menggunakan Latitude dan Longitude pada Google map, hasil dari perhitungan dengan metode Haversine Formula adalah jarak dari kedua titik yang dapat digambarkan dalam peta menggunakan fasilitas API atau Application Programming Interface pada Google map. Bentuk pola Haversine diperlihatkan seperti gambar :



Gambar 1.1 Bentuk Pola Haversine

Gambar 1.1 mengilustrasikan prinsip dasar rumus Haversine, sebuah persamaan trigonometri bola yang esensial dalam navigasi untuk menghitung jarak terpendek antara dua titik di permukaan bumi. Awalnya, rumus ini diterapkan dalam astronomi kelautan untuk mengukur jarak antar bintang. Josef de Mendoza y Rios pertama kali menggunakan rumus ini pada tahun 1801, sedangkan penemunya adalah James Andrew pada tahun 1805. Istilah "Haversine" sendiri baru muncul pada tahun 1835, dicetuskan oleh Profesor James Inman.

Penerapan rumus Haversine didasarkan pada asumsi bahwa bumi berbentuk bola sempurna dengan jari-jari R sebesar 6.3671 km. Dengan mengetahui koordinat bola (lintang dan bujur) dari dua titik, yaitu lon_1 , lat_1 , dan lon_2 , lat_2 , kita dapat

menghitung jarak terpendek antara kedua titik tersebut menggunakan rumus Haversine.

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2}) / 2); \quad \text{Persamaan 1}$$

$$y = (\text{lat2} - \text{lat1}); \quad \text{Persamaan 2}$$

$$d = \text{sqrt}(x * x + y * y) * R \quad \text{Persamaan 3}$$

Keterangan :

Lat1 = Derajat latitude pengguna

Lon1 = Derajat longitude pengguna

Lat2 = Derajat latitude tempat wisata

Lon2 = Derajat longitude tempat wisata

x = Longitude (Bujur)

y = Latitude (Lintang)

d = Jarak (km)

1 derajat = 0,0174532925 radian

R = 6371 km.

2.2 Peralatan Analisa (Tool System)

Alat bantu analisa sistem dan perancangan dalam pendekatan sistem, penulis menggunakan alat bantu untuk proses analisis dan perancangan terstruktur, yang membantu menunjang pembuatan tugas akhir diantaranya :

2.2.1 Basis Data (*Database*)

Menurut Indrajani (2015:70), “basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.”

1. Postgresql

PostgreSQL adalah sistem manajemen basis data relasional yang bersifat open-source dan didistribusikan secara bebas di bawah lisensi BSD. PostgreSQL merupakan salah satu sistem database yang populer dan banyak digunakan selain MySQL dan Oracle. Keunggulan PostgreSQL terletak pada fitur-fitur replikasi basis data yang dimilikinya, seperti DB Mirror, PGPool, Slony, PGCluster, dan lain sebagainya. Fitur-fitur ini memungkinkan pengguna untuk membuat salinan basis data yang identik dan menjaga konsistensi data di berbagai lokasi, meningkatkan ketersediaan dan keandalan sistem. (2.3)

2. JavaScript (Bahasa Pemrograman)

JavaScript adalah bahasa pemrograman yang digunakan untuk pengembangan web dan berjalan di sisi klien (client-side). Artinya, kode JavaScript diproses dan dieksekusi langsung oleh peramban web pengguna, seperti Google Chrome, Mozilla Firefox, atau Opera Mini. Hal ini berbeda dengan bahasa pemrograman sisi server (server-side) yang diproses di server sebelum hasilnya dikirimkan ke peramban pengguna. (2.4)

3. React Native

React Native adalah sebuah kerangka kerja (framework) JavaScript yang digunakan untuk membangun aplikasi mobile. Pengembangan aplikasi menggunakan React Native melibatkan penulisan kode dalam campuran JavaScript dan JSX, yaitu sebuah sintaks yang menyerupai XML. Keunggulan React Native adalah kemampuannya untuk menyediakan antarmuka JavaScript yang dapat mengakses fitur-fitur perangkat seperti kamera dan lokasi, sehingga aplikasi yang dibangun dapat memanfaatkan fitur-fitur tersebut secara optimal (2.5).

4. Express.JS

ExpressJS adalah kerangka kerja aplikasi web yang ringan dan fleksibel untuk Node.js, yang menyediakan berbagai fitur untuk pengembangan aplikasi web dan mobile. Dikembangkan dan dikelola oleh NodeJS Foundation, ExpressJS bersifat open-source dan menawarkan antarmuka sederhana untuk membangun aplikasi. Dengan berbagai modul yang tersedia di npm, ExpressJS memberikan fleksibilitas dan kemudahan dalam pengembangan aplikasi dengan menyediakan alat-alat yang dibutuhkan secara langsung.

2.2.2 UML (Unified Modelling Language)

Menurut (Kroenke et al.,2018) UML adalah seperangkat diagram,struktur, dan teknik untuk memodelkan dan merancang program dan aplikasi berorientasi objek. Sedangkan menurut (Misriati,2015) UML adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Berikut beberapa contoh dari UML diantaranya :

2.2.3 Tujuan UML (Unified Modelling Language)

Tujuan utama dari UML (Unified Modeling Language) adalah untuk menyediakan seperangkat alat yang komprehensif bagi para pengembang perangkat lunak dalam menganalisis, merancang, dan mengimplementasikan sistem berbasis perangkat lunak. Dengan adanya alat pemodelan visual ini, industri UML dapat berkembang pesat dan menghasilkan perangkat lunak yang lebih berkualitas dan terstruktur.

2.2.4 Model-Model Diagram UML

1. Use Case Diagram

UCD adalah diagram yang digunakan untuk menunjukkan hubungan antara user/actor dengan sistem.

2. *Scenario*

Menurut (Munawar, 2005) *Scenario* adalah dokumentasi yang menjelaskan kebutuhan sistem dari sudut pandang pengguna (actor) dalam suatu Use Case yang telah dirancang sebelumnya. Skenario ini disusun dalam bentuk formulir yang berisi penjelasan tertulis tentang interaksi antara pengguna dan sistem dalam mencapai tujuan tertentu.

3. *Activity Diagram*

Adalah diagram modelling dari suatu aktivitas yang diurutkan dan secara visual menggambarkan alur kerja.

2.2.5 Metode Pengembangan Sistem

Metode Agile Software Development dipilih sebagai pendekatan utama dalam perancangan sistem informasi ini. Agile Development menekankan pada fleksibilitas dan adaptabilitas tim pengembang terhadap perubahan kebutuhan pengguna selama proses pengembangan (Fahrudin & Ilyasa, 2). Pendekatan ini memungkinkan pengembangan sistem secara bertahap dan iteratif, sehingga memungkinkan respons cepat terhadap perubahan dan umpan balik dari pengguna.

Tahapan-tahapan dalam pengembangan sistem informasi menggunakan metode Agile adalah sebagai berikut:

1. **Perencanaan (Planning):** Tahap ini dimulai dengan pengumpulan data dari pengguna melalui wawancara atau observasi langsung untuk memahami kebutuhan sistem secara komprehensif. Selanjutnya, tim pengembang merancang sistem secara menyeluruh dengan menggunakan diagram UML, ERD, dan LRS.

2. **Implementasi (Implementation):** Pada tahap ini, programmer menerjemahkan desain sistem menjadi kode program yang fungsional menggunakan bahasa pemrograman PHP dan framework Laravel untuk membangun sistem berbasis web.
3. **Pengujian (Testing):** Sistem yang telah dibangun diuji secara menyeluruh menggunakan metode black-box testing untuk mengidentifikasi dan memperbaiki bug atau kesalahan. Validasi input dan output juga dilakukan untuk memastikan kesesuaian dengan harapan pengguna.
4. **Dokumentasi (Documentation):** Selama proses pengembangan, tim pengembang membuat catatan modul dan fungsi yang ada dalam sistem informasi. Dokumentasi ini berfungsi sebagai referensi bagi tim pengembang di masa mendatang dan memfasilitasi pengembangan lebih lanjut.
5. **Penyebaran (Deployment):** Setelah sistem selesai dikembangkan dan diuji, sistem siap digunakan oleh pengguna akhir. Tahap ini melibatkan penyediaan akses dan konfigurasi sistem agar dapat diakses dan digunakan oleh pengguna.
6. **Pemeliharaan (Maintenance):** Tahap ini berlangsung secara berkelanjutan setelah sistem digunakan. Pemeliharaan berkala dilakukan untuk memastikan keamanan sistem dari bug atau celah yang mungkin muncul, serta untuk melakukan pembaruan dan peningkatan sesuai dengan kebutuhan pengguna yang berkembang.