

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Pada bagian ini disajikan beberapa penelitian terdahulu yang relevan dengan topik penelitian sebagai landasan dalam penyusunan skripsi. Referensi yang digunakan berasal dari publikasi ilmiah, seperti jurnal, prosiding, maupun hasil penelitian sebelumnya, yang memiliki keterkaitan dengan objek, metode, atau permasalahan yang diteliti. Dalam penelitian ini digunakan lima referensi ilmiah yang dipilih berdasarkan tingkat relevansi dan keterkaitannya dengan penelitian yang dilakukan.

1. Maxsi Ari (2022), dalam penelitian berjudul *Optimasi Vehicle Routing Problem pada Rute Pendistribusian Menggunakan Metode Ant Colony Optimization*, membahas permasalahan penentuan rute distribusi pada perusahaan jasa pengiriman barang dan dokumen yang belum memiliki metode optimasi rute. Penelitian tersebut menerapkan metode Ant Colony Optimization (ACO) untuk menyelesaikan permasalahan Vehicle Routing Problem (VRP). Hasil penelitian menunjukkan bahwa rute distribusi paling optimal adalah A–C–E–B–D–F–A dengan total jarak tempuh sebesar 18,9 km, sehingga proses distribusi menjadi lebih efisien.
2. Nova Sepadyati (2022), melalui penelitian berjudul *Optimalisasi Rute Pengiriman dengan Meminimasi Jarak Menggunakan Saving Matrix: Sebuah Studi*, mengkaji proses distribusi produk pada CV. X di Surabaya. Permasalahan yang dihadapi berupa rendahnya tingkat pemanfaatan kapasitas kendaraan dan terjadinya pengiriman berulang pada hari yang sama. Penelitian menggunakan metode Saving Matrix untuk menyelesaikan masalah Capacitated Vehicle Routing Problem (CVRP). Hasil penelitian menunjukkan adanya penghematan jarak tempuh sebesar 12,11 km serta pengurangan penggunaan kendaraan hingga dua unit dibandingkan dengan sistem distribusi sebelumnya.

3. Novria Fatmawati Lakutu (2023), dalam penelitian berjudul *Algoritma Dijkstra dan Algoritma Greedy untuk Optimasi Rute Pengiriman Barang pada Kantor Pos Gorontalo*, membandingkan kinerja Algoritma Dijkstra dan Algoritma Greedy dalam menentukan rute pengiriman barang yang paling efisien. Berdasarkan hasil penelitian, Algoritma Dijkstra menghasilkan total jarak tempuh sebesar 304,90 km, sedangkan Algoritma Greedy menghasilkan 441,60 km. Dengan demikian, Algoritma Dijkstra dinilai lebih optimal dalam menentukan jalur distribusi.
4. Hasyrani Windyatri (2023), melalui penelitian berjudul *Optimasi Rute Pengiriman BBM dengan Heterogeneous Vehicle Routing Problem with Multi-Trips*, mengoptimalkan distribusi bahan bakar minyak (BBM) dari depot menuju beberapa SPBU di wilayah Yogyakarta. Penelitian menerapkan metode Heterogeneous Vehicle Routing Problem (HVRP) dengan mempertimbangkan konsep multi-trip, pembagian cluster, serta klasifikasi jalan menggunakan Branch and Bound pada perangkat lunak LINGO Solver. Hasil optimasi menunjukkan penurunan total jarak tempuh sebesar 6,55% pada Cluster 1, 23,84% pada Cluster 2, dan 9,57% pada Cluster 3.
5. Ulfa Is. Abdul (2023), dalam penelitian berjudul *Optimasi Rute Pendistribusian BBM Menggunakan Algoritma Tabu Search dan Cheapest Insertion Heuristic*, membandingkan dua algoritma heuristik dalam menentukan rute distribusi BBM di PT Pertamina (Persero) Unit Pemasaran VII Gorontalo. Hasil penelitian menunjukkan bahwa Algoritma Tabu Search menghasilkan rute yang lebih optimal dengan total jarak tempuh 815,2 km, sedangkan Cheapest Insertion Heuristic menghasilkan total jarak tempuh 1.039,7 km, sehingga Tabu Search dinilai lebih efektif dalam mengoptimalkan proses distribusi.

2.2 Sistem Pencarian Rute (Route Finding System)

2.2.1. Pengertian Pencarian Rute

Pencarian rute adalah proses menentukan lintasan terbaik antara dua atau lebih titik di jaringan jalan. Menurut Delleng et al. (2009), pencarian rute optimal bertujuan untuk menemukan jalur dengan kriteria tertentu seperti jarak terpendek, waktu tempuh tercepat, atau biaya transportasi paling efisien

2.2.2. Kriteria Optimasi Rute

Beberapa kriteria umum dalam optimasi rute sistem informasi geografis meliputi:

1. Jarak terpendek (Shortest Path) – mencari rute dengan total jarak paling minimal.
2. Waktu tempuh tercepat (Fastest Route) – memperhitungkan kecepatan rata-rata atau kondisi lalu lintas.
3. Biaya minimum (Least Cost Route) – memperhitungkan faktor biaya seperti bahan bakar atau tarif tol.

2.2.3. Algoritma Pencarian Rute

Beberapa algoritma yang populer dalam pencarian rute sistem informasi geografis adalah:

1. Algoritma Dijkstra

Menemukan jarak terpendek dari satu titik ke semua titik lain pada graf berbobot non-negatif.

2. Algoritma A*

Merupakan pengembangan dari Dijkstra dengan menggunakan fungsi heuristik untuk mempercepat pencarian.

3. Algoritma Floyd-Warshall

Menghitung jarak terpendek antara semua pasangan titik dalam graf. Dalam penelitian ini, algoritma Dijkstra atau A* sering dipilih karena efisiensi dan akurasi dalam sistem navigasi berbasis GIS.

2.3 Optimalisasi Pencarian Rute

Optimalisasi rute merupakan proses untuk meningkatkan efisiensi jalur berdasarkan parameter tertentu (misalnya jarak, waktu, atau biaya). Menurut Kumar & Singh (2020), optimalisasi dapat dilakukan dengan menggabungkan algoritma pencarian dengan teknik heuristik atau metaheuristik, seperti *Genetic Algorithm* (GA), *Ant Colony Optimization* (ACO), atau *Simulated Annealing* (SA). Tujuan optimalisasi adalah:

1. Mengurangi waktu tempuh dan konsumsi bahan bakar.
2. Meningkatkan akurasi hasil rekomendasi.
3. Menyediakan rute alternatif yang adaptif terhadap kondisi dinamis (macet, jalan rusak, dll).

2.4 Android

Android merupakan sistem operasi untuk telepon seluler yang berbasis linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Android umumnya digunakan di smartphone dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di nokia, iOS, Apple, dan BlackBerry OS. (Mulyati, 2017)

Android adalah sebuah *toolkit software* yang baru untuk perangkat bergerak yang dibuat oleh Google dan *Open Handset Alliance*. Dalam beberapa tahun, *android* diharapkan dapat ditemukan dalam jutaan *handphone* dan berbagai perangkat bergerak, membuat *android* menjadi *platform* utama untuk pengembang aplikasi.

Sudah ada banyak *platformmobile* di pasar saat ini, termasuk Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiMo). Meskipun beberapa fitur telah muncul sebelumnya, *android* adalah *platform* yang menggabungkan beberapa hal berikut :

1. *Android* merupakan sebuah *platform* yang berbasis linux dan *open source*. Pembuat *handset* menyukai hal ini karena mereka dapat menggunakan dan menyesuaikan *platform* tanpa membayar *royalti*.
2. Sebuah arsitektur berbasis komponen. Bagian dari *aplikasi android* dapat digunakan sebagai bahan lain oleh *developer*.
3. Banyak *built-in service* yang tidak biasa. Servis berdasarkan lokasi menggunakan GPS atau *cell tower triangulation* yang membuat pengalaman pemakai terjadi bergantung lokasi.

Aplikasi *android* dijalankan di atas VM bernama *Dalvik Virtual Machine*. Di dunia ini terdapat dua jenis distributor sistem operasi *Android*. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD) (Kasman, 2016).

Android merupakan sebuah sistem operasi telepon *seluler* dan komputer tablet layar sentuh (*touchscreen*) yang berbasis linux. Sedangkan menurut Murya (2014) android adalah sistem operasi berbasis linux yang digunakan untuk telepon *seluler* (*mobile*) seperti telepon pintar (*smarphone*) dan komputer tablet (PDA). Android menyediakan *platform* terbuka bagi prara pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh bermacam-macam piranti bergerak. (Kasman, 2016)

2.5 PHP

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML. PHP merupakan singkatan dari “PHP : *Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen HTML, sekaligus bekerja di sisi *server* (*server-side HTML-embedded scripting*). Artinya sintaks dan perintah di *server* tetapi disertakan pada halaman HTML biasa, sehingga *script* tak tampak disisi *client* yang dapat mengakses database menjadi begitu mudah. (Setiawan, 2014)

Tujuan dari bahasa ini adalah untuk membuat aplikasi di mana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. Kelebihan PHP dari bahasa pemrograman web, antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dari mulai IIS, *apache*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman *open source* yang banyak digunakan untuk mengembangkan aplikasi berbasis web. PHP dapat dijalankan pada berbagai sistem operasi, seperti Linux, Windows, dan macOS, serta diproses di sisi server (*server-side scripting*) sehingga kode program dieksekusi sebelum hasilnya ditampilkan kepada pengguna. PHP menyediakan beberapa tipe data yang digunakan untuk menyimpan dan mengolah informasi, yaitu sebagai berikut :

a. *Integer*

Integer merupakan tipe data yang digunakan untuk menyimpan bilangan bulat tanpa nilai desimal, misalnya 1, 25, -10, dan 100.

b. *Double*

Double atau *float* merupakan tipe data yang digunakan untuk menyimpan bilangan pecahan atau desimal, seperti 0,5; 3,14; dan 12,75.

c. Object

Object merupakan tipe data yang digunakan untuk merepresentasikan objek yang dibuat dari suatu kelas (*class*). Tipe data ini berisi atribut dan metode yang dapat digunakan dalam pemrograman berorientasi objek (*Object-Oriented Programming/OOP*).

d. Boolean

Boolean merupakan tipe data yang hanya memiliki dua nilai, yaitu *true* dan *false*. Tipe data ini umumnya digunakan untuk proses pengambilan keputusan atau pengecekan kondisi, seperti pada pernyataan *if*, *else*, maupun *while*.

e. String

String merupakan tipe data yang digunakan untuk menyimpan data berupa teks, karakter, angka, maupun kombinasi keduanya yang dituliskan di dalam tanda kutip.

f. Array

Array merupakan tipe data yang dapat menyimpan lebih dari satu nilai dalam satu variabel. Setiap elemen dalam array dapat diakses menggunakan indeks atau kunci (*key*)

g. Null

Null merupakan tipe data yang menunjukkan bahwa suatu variabel tidak memiliki nilai atau bernilai kosong. Nilai Null biasanya digunakan untuk menginisialisasi variabel yang belum memiliki isi atau setelah nilainya dihapus.

h. Resource

Resource merupakan tipe data khusus yang digunakan untuk menyimpan referensi terhadap sumber daya eksternal, seperti koneksi basis data, file, atau layanan lain yang dibuat oleh fungsi tertentu pada PHP

2.6 SQL

Menurut Yahya (2015), SQL (*Structured Query Language*) merupakan bahasa standar yang digunakan untuk mengelola dan memanipulasi data pada sistem manajemen basis data relasional (*Relational Database Management System* atau RDBMS). Salah satu RDBMS yang banyak digunakan adalah MySQL, yang didistribusikan secara gratis di bawah lisensi *General Public License* (GPL) sehingga dapat digunakan dan dikembangkan oleh siapa saja (Biasta, 2011).

MySQL dikembangkan berdasarkan konsep *Structured Query Language* (SQL), yaitu bahasa yang digunakan untuk melakukan berbagai operasi pada basis data, seperti membuat, mengubah, menghapus, dan mengambil data. Kinerja suatu sistem manajemen basis data dapat dilihat dari kemampuan *optimizer* dalam memproses dan mengeksekusi perintah SQL yang diberikan, baik melalui masukan pengguna maupun dari aplikasi yang menggunakan basis data tersebut.

2.7 Konsep Perancangan Database

Perancangan database merupakan tahapan penting dalam pengembangan sistem informasi yang bertujuan untuk merancang struktur penyimpanan data agar dapat dikelola secara efektif dan efisien. Proses ini tidak hanya berfokus pada konsep basis data, tetapi juga mencakup implementasinya sehingga mampu mendukung kebutuhan aplikasi yang dikembangkan. Menurut Ladjamudin (2005), perancangan database merupakan kombinasi dari berbagai proses yang saling berkaitan dalam membangun sistem basis data yang terintegrasi dengan aplikasi.

a. Diagram Konteks

Diagram konteks adalah diagram yang menempatkan sistem dalam konteks yang berguna sebagai identifikasi *input* dan *output* yang terlibat dalam sistem. Diagram Konteks bertujuan untuk memudahkan pengembangan sistem serta memberikan gambaran umum tentang sistem yang dibangun.

b. Diagram Alir Data (Data Flow Diagram/DFD)

Data Flow Diagram (DFD) atau Diagram Alir Data (DAD) merupakan alat pemodelan yang digunakan untuk menggambarkan aliran data dalam suatu sistem secara logis. DFD berfokus pada proses pengolahan data tanpa dipengaruhi oleh aspek fisik, seperti perangkat keras, perangkat lunak, maupun struktur penyimpanan data. Penggunaan DFD memudahkan pengembang maupun pengguna dalam memahami alur kerja dan proses yang terjadi di dalam sistem. Adapun simbol-simbol yang digunakan dalam DFD adalah sebagai berikut:

a. *External Entity (Entitas Eksternal)*

Entitas *eksternal* merupakan pihak atau objek yang berada di luar batas sistem, tetapi berinteraksi dengan sistem melalui proses pemberian masukan (*input*) maupun penerimaan keluaran (*output*).

b. *Data Flow (Arus Data)*

Arus data menunjukkan perpindahan data antarproses, penyimpanan data (*data store*), dan entitas eksternal. Arus data menggambarkan informasi yang menjadi masukan maupun keluaran dari suatu proses dalam sistem.

c. *Process (Proses)*

Proses merupakan aktivitas yang mengolah data masukan menjadi informasi atau keluaran sesuai dengan fungsi yang dijalankan oleh sistem. Proses dapat dilakukan oleh manusia, komputer, maupun kombinasi keduanya.

d. Data Store (Simpanan Data)

Penyimpanan data merupakan media yang digunakan untuk menyimpan data, baik dalam bentuk terkomputerisasi maupun nonkomputerisasi, sehingga data dapat diakses kembali saat diperlukan.

Simbol-simbol DFD yang umum digunakan mengacu pada notasi *Yourdon-DeMarco* dan *Gane & Sarson*. Kedua notasi tersebut memiliki bentuk simbol yang berbeda sehingga penggunaannya tidak dapat dicampur dalam satu diagram. Oleh karena itu, dalam perancangan sistem harus dipilih salah satu notasi agar diagram tetap konsisten. Simbol-simbol tersebut dapat dilihat pada tabel berikut.

Simbol Yourdon	Keterangan
	Entity Luar
	Data Arus
	Proses
	Simpanan Data

Tabel 2.2 Simbol DFD

c. Entity Relationship Diagram (ERD)

Menurut Ladjamudin (2005), *Entity Relationship Diagram* (ERD) merupakan model yang digunakan untuk menggambarkan struktur basis data melalui hubungan antarentitas yang saling berinteraksi. ERD berfungsi untuk memodelkan data secara konseptual sehingga memudahkan dalam perancangan basis data. Adapun komponen utama yang terdapat dalam ERD meliputi:

1. Entity 	:	Entity digambarkan menggunakan simbol berbentuk persegi panjang (<i>rectangle</i> atau <i>box</i>)..
2. Atribut 	:	elemen yang berfungsi untuk memberikan informasi atau menjelaskan sifat-sifat dari entitas tersebut.
3. Line 	:	Berfungsi untuk menghubungkan atribut dengan entitas yang saling berelasi
4. Hubungan 	:	berfungsi hubungan yang terjadi antara satu entitas atau lebih

Tabel 2.3 Simbol ERD