IMPLEMENTASI ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK KE WISATA PANTAI PULAU NIAS

SKRIPSI

OLEH:

JEPRIN TALUNOHI

178160020

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

MEDAN

2023

IMPLEMENTASI ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK KE WISATA PANTAI PULAU NIAS

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh

Gelar Sarjana (S1) di Fakultas Teknik

Universitas Medan Area

Oleh:

JEPRIN TALUNOHI

178160020

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

MEDAN

2023

LEMBAR PENGESAHAN

Judul	: Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek ke Wisata Pantai Pulau Nias
Nama	: Jeprin Talunohi
NPM	: 178160020
Program Studi	: Sarjana (S1) Teknik Informatika
Fakultas	: Teknik

Disetujui Oleh

Komisi Pembimbing

Zulfikar Sembiring, S. Kom, M. Kom
Pembimbing I

Nurul Khairina, S. Kom, M.Kom
Pembimbing II

Dr. Rahmad Syah, S.Kom, M.Kom
Dekan Fakultas Teknik

Rizki Muliono, S. Kom, M.Kom
Ka. Prodi

Tanggal Lulus : 11 April 2023

HALAMAN PERNYATAAN

Saya menyatakan dengan sungguh-sungguh bahwa skripsi ini yang saya susun, sebagai syarat memperoleh gelar serjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Apabila dikemudian hari terdapat kejanggalan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya sesuai dengan peraturan yang berlaku di Universitas Medan Area.

Demikian pernyataan ini saya buat.

Medan, April 2023

Yang membuat pernyataan,

Jeprin Talunohi

178160020

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIK

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama	: Jeprin Talunohi
NPM	: 178160020
Program Studi	: Sarjana (S1) Teknik Informatika
Fakultas	: Teknik
Jenis karya	: Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

Implementasi Algoritma *Dijkstra* untuk Menentukan Rute Terpendek ke Wisata Pantai Pulau Nias

Beserta dengan perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-eksklusif ini Universitas Medan Area berhak menyimpan, mengalihkan media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir/skripsi/tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan, Sumatera Utara

Pada Tanggal : 11 April 2023

Yang menyatakan :

(Jeprin Talunohi)

RIWAYAT HIDUP

Jeprin Talunohi, dilahirkan di desa Hilizalootano kecamatan Mazino kabupaten Nias Selatan. Pada tanggal 3 Maret 1998, anak ke pertama dari tujuh bersaudara pasangan dari ayah Zanaha Talunohi dan ibu Isalia harita.

Penulis menyelesaikan pendidikan Sekolah Dasar di SD Perguruan Nasional Gultom di kecamatan Medan Perjuangan kabupaten Kota Medan pada tahun 2011, pada tahun itu penulis melanjutkan pendidikan Sekolah Menengah Pertama di SMP HKBP Sidorame Kec. Medan Perjuangan Kab. Kota Medan dan tamat pada tahun 2014. Penulis melanjutkan Pendidikan Sekolah Menengah Atas di SMA Negeri 8 Medan Kec. Medan Area Kab. Kota Medan dan lulus pada tahun 2017.

Pada tahun 2017, penulis melanjutkan pendidikan pada perguruan tinggi swasta tepatnya di Universitas Medan Area (UMA) Fakultas Teknik pada Program Studi Teknik Informatika.

ABSTRAK

Dalam menyelesaikan masalah jarak terpendek untuk menghasilkan satu tujuan pada sebuah lintasan diperlukan sebuah algoritma. Algoritma *Dijkstra* merupakan salah satu algoritma yang bisa digunakan untuk menyelesaikan masalah jarak terpendek dengan satu tujuan pada sebuah lintasan yang tidak memiliki sisi *cost negative*. Penelitian ini bertujuan untuk mengoptimalkan jalur lokasi wisata menggunakan Algoritma *Dijkstra* sehingga menghasilkan jalur yang efektif bagi wisatawan untuk mengunjungi seluruh pantai yang ada di Pulau Nias. Pantai tersebut adalah Pantai Hoya Gunungsitoli, Pantai Indah Fofola, Pantai Saiti, Pantai Indah Tureloto, Pantai Toyolawa, Pantai Merah Afulu, Pantai Sirombu, Pantai Ladeha, Pantai Sorake, Pantai Lagundri, Pantai Blessing dan Pantai Pulau Tello. Hasil dari penelitian ini adalah Algoritma *Dijkstra* dapat diimplementasikan untuk mendapatkan jalur alternatif berupa rute terpendek bagi wisatawan untuk berkunjung ke pantai yang ada di daerah kepulauan Nias. Berdasarkan pengolahan data rute perjalanan secara manual yang dilakukan, jalur terpendek yang bisa dilalui adalah A-B-C-D-E-F-G-H-I-J-K-L dengan panjang lintasan adalah 430,7 km. Dari pengujian sistem yang dilakukan, jarak rute dan waktu tempuh terbaik yang bisa dilalui oleh wisatawan menuju lokasi pantai yaitu rute dari lokasi awal Gunungsitoli menuju lokasi tujuan Pantai Hoya Gunungsitoli dengan jarak rute 35.84 km dan waktu tempuh 149,89 menit.

Kata Kunci: Algoritma *Dijkstra*, Pantai Nias, Jalur Terpendek, Objek Wisata, Sistem.

Abstract

In solving the shortest distance problem to produce one destination on a path, an algorithm is needed. Dijkstra's algorithm is one of the algorithms that can be used to solve the shortest distance problem with one goal on a path that has no negative cost. This study aims to optimize the path of tourist sites using Dijkstra's algorithm so as to produce an effective path for tourists to visit all the beaches on Nias Island. The beaches are Hoya Gunungsitoli Beach, Indah Fofola Beach, Saiti Beach, Indah Tureloto Beach, Toyolawa Beach, Red Afulu Beach, Sirombu Beach, Ladeha Beach, Sorake Beach, Lagundri Beach, Blessing Beach, and Tello Island Beach. The result of this research is that Dijkstra's algorithm can be implemented to get an alternative path in the form of the shortest route for tourists to visit beaches in the Nias archipelago. Based on manual travel route data processing, the shortest path that can be traversed is A-B-C-D-E-F-G-H-I-J-K-L, with a track length of 430.7 km. From the system testing carried out, the best route distance and travel time that can be traveled by tourists to the beach location is the route from the initial location of Gunungsitoli to the destination location of Hoya Gunungsitoli Beach, with a route distance of 35.84 km and a travel time of 149.89 minutes.

Keywords: Dijkstra Algorithm, Nias Beach, Shortest Path, Tourist Attractions, System.

KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa atas berkat, rahmat, dan karunia-Nya, saya bisa menyelesaikan penyusunan skripsi dengan judul "Implementasi Algoritma *Dijkstra* untuk Menentukan Rute Terpendek ke Wisata Pantai Pulau Nias".

Pada kesempatan ini, penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah memberikan banyak dukungan seperti dukungan motivasi, dukungan informasi materi, dan arahan, sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya:

1. Tuhan Yang Maha Esa, berkat rahmat dan karunia-Nya skripsi ini dapat terselesaikan,
2. Orang Tua Bapak dan Ibu penulis yang telah mendukung, memberikan semangat, motivasi, dan banyak perhatian serta memenuhi segala kebutuhan penulis selama masa penyusunan skripsi ini,
3. Bapak Prof. Dr. Dadan Ramdan, M.Eng., M.Sc., selaku Rektor Universitas Medan Area,
4. Bapak Dr. Rahmad Syah, S.Kom., M.Kom., selaku Dekan Fakultas Teknik Universitas Medan Area,
5. Ibu Susilawati, S.Kom., M.Kom., selaku Wakil Dekan Fakultas Teknik Universitas Medan Area,
6. Bapak Rizki Muliono, S.Kom., M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Medan Area,

7. Bapak Zulfikar Sembiring, S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah memberikan banyak masukan, kritik, saran, dan motivasi kepada penulis serta membimbing penulis dalam menyelesaikan skripsi ini,
8. Ibu Nurul Khairina, S.Kom., M.Kom., selaku Dosen Pembimbing II yang telah memberikan arahan, bimbingan, semangat, dan motivasi kepada penulis hingga penyusunan skripsi ini terselesaikan,
9. Bapak/Ibu karyawan pengelola pantai blessing terkhususnya kepada bapak Marselinus Fau, selaku Kepala Pengelola Pantai Blessing yang telah memberikan waktu untuk tanya jawab terkait penelitian saya selama di Blessing Beach,
10. Serta semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini, yang Namanya tidak bisa disebutkan satu persatu. Terimakasih banyak.

Penulis menyadari bahwa Tugas Akhir/Skripsi ini masih jauh dari kata sempurna, untuk itu penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk pengembangan selanjutnya.

Medan, 11 April 2023

Jeprin Talunohi

178160020

DAFTAR ISI

Halaman

LEMBAR PENGESAHAN.....**Error! Bookmark not defined.**

HALAMAN PERNYATAAN.....**Error! Bookmark not defined.**

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIK..... **Error! Bookmark not defined.**

RIWAYAT HIDUP v

ABSTRAK..... vi

KATA PENGANTAR..... viii

DAFTAR ISI x

DAFTAR TABEL xiii

DAFTAR GAMBAR..... xiv

BAB I PENDAHULUAN..... 1

1.1 Latar Belakang 1

1.2 Perumusan Masalah 4

1.3 Batasan Penelitian 4

1.4 Tujuan Penelitian 5

1.5 Manfaat Penelitian 5

1.6 Sistematika Penulisan..... 6

BAB II TINJAUAN PUSTAKA 7

2.1 Algoritma 7

2.2 Pengertian Graf 8

2.3 Jalur Lintasan Terpendek (*Shortest Path*) 10

2.4 *Google Maps API* 10

2.5 *Website* 12

2.6 Bahasa Pemrograman *PHP* 13

2.7 Database *MySQL* 14

2.8 Algoritma *Dijkstra* 16

2.9 Tahapan Algoritma *Dijkstra* 17

2.10 Penelitian Terdahulu 19

BAB III ANALISIS DAN PERANCANGAN SISTEM 25

3.1 Objek Penelitian 25

3.2 Metode Penelitian 26

3.3 Data Penelitian 28

3.4 Tahapan Penelitian Menentukan Rute Dengan Algoritma *Dijkstra* 31

3.5 Pengolahan Data Rute Perjalanan Secara Manual 33

3.6 Perancangan Sistem 42

3.7 Perancangan *User Interface* 48

BAB IV HASIL DAN PEMBAHASAN 54

4.1 Hasil 54

4.2 Pembahasan 63

BAB V KESIMPULAN DAN SARAN 66

5.1 Kesimpulan 66

5.2 Saran..... 67

DAFTAR PUSTAKA..... 68

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu 19

Tabel 3.1 Objek Penelitian 25

Tabel 3.2 Koordinat Lokasi Pantai 28

Tabel 3.3 Jarak Antar Setiap *Node* Dari *Maps* 29

Tabel 3.4 Hasil Pengujian Pencarian Rute Terpendek Secara Manual 41

Tabel 4.1 Perangkat Pengujian Sistem Aplikasi 54

Tabel 4.2 Pengujian Tampilan Antarmuka Pengguna Aplikasi 55

Tabel 4.3 Pengujian Implementasi Algoritma *Dijkstra* 61

DAFTAR GAMBAR

Gambar 2.1 Flowchart Proses Algoritma *Dijkstra* 18

Gambar 3.1 Diagram Alir Tahapan Penelitian 26

Gambar 3.2 Graf Rute Perjalanan Menuju Wisata Pantai Nias 30

Gambar 3.3 Flowchart Tahapan Penelitian Menentukan Rute Terpendek 31

Gambar 3.4 Contoh Perhitungan Manual Pencarian Rute Terpendek 32

Gambar 3.5 Langkah Pertama Pengolahan Data Algoritma *Dijkstra* 34

Gambar 3.6 Langkah Kedua Pengolahan Data Algoritma *Dijkstra* 35

Gambar 3.7 Langkah Ketiga Pengolahan Data Algoritma *Dijkstra* 35

Gambar 3.8 Langkah Keempat Pengolahan Data Algoritma *Dijkstra* 36

Gambar 3.9 Langkah Kelima Pengolahan Data Algoritma *Dijkstra* 36

Gambar 3.10 Langkah Keenam Pengolahan Data Algoritma *Dijkstra* 37

Gambar 3.11 Langkah Ketujuh Pengolahan Data Algoritma *Dijkstra* 37

Gambar 3.12 Langkah Kedelapan Pengolahan Data Algoritma *Dijkstra* 38

Gambar 3.13 Langkah Kesembilan Pengolahan Data Algoritma *Dijkstra* 38

Gambar 3.14 Langkah Kesepuluh Pengolahan Data Algoritma *Dijkstra* 39

Gambar 3.15 Langkah Kesebelas Pengolahan Data Algoritma *Dijkstra* 39

Gambar 3.16 Langkah Keduabelas Pengolahan Data Algoritma *Dijkstra* 40

Gambar 3.17 Langkah Ketigabelas Pengolahan Data Algoritma *Dijkstra* 40

Gambar 3.18 Use Case Diagram 43

Gambar 3.19 Flowchart Sistem Pencarian Rute 44

Gambar 3.20 Flowchart *Login Admin* 45

Gambar 3.21 Flowchart Halaman Data Informasi Pantai 46

Gambar 3.22 Flowchart Halaman Data Informasi *Node* 47

Gambar 3.23 Flowchart Halaman Data Informasi *Admin* 48

Gambar 3.24 Tampilan Menu Utama 49

Gambar 3.25 Tampilan *Form Login Admin* 50

Gambar 3.26 Tampilan Menu Utama *Admin* 50

Gambar 3.27 Tampilan *Form* Tambah Data Pantai 51

Gambar 3.28 Tampilan *Form* Tambah Data *Node* 52

Gambar 3.29 Tampilan *Form* Tambah Data *Admin* 53

Gambar 4.1 Tampilan Menu Utama 57

Gambar 4.2 Tampilan Daftar Pantai Pada Sistem Aplikasi 58

Gambar 4.3 Tampilan Halaman *Form Login Admin* 58

Gambar 4.4 Tampilan Halaman Menu Utama *Admin* 59

Gambar 4.5 Tampilan Halaman *Form* Data Pantai 59

Gambar 4.6 Tampilan Halaman *Form* Tambah Data *Node* 60

Gambar 4.7 Tampilan Halaman *Form* Tambah Data *Admin* 60

Gambar 4.8 Tampilan Pencarian Rute Berhasil 64

Gambar 4.9 Tampilan Pencarian Rute Gagal Versi 1 65

Gambar 4.10 Tampilan Pencarian Rute Gagal Versi 2 65

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam sektor pariwisata, Indonesia dikenal memiliki banyak destinasi wisata potensial yang menarik banyak wisatawan untuk berkunjung ke lokasi tersebut. Pulau Nias merupakan salah satu daerah pariwisata yang berpotensi karena keindahan alamnya yang luar biasa dibuktikan sejak tahun 2017 - 2019 adanya kunjungan wisatawan mancanegara ke Pulau Nias sebanyak 9.086 orang dan kunjungan wisatawan domestik sebanyak 111.139 orang. Pulau Nias merupakan sebuah pulau di bagian barat Sumatera Indonesia yang dikelilingi oleh laut yang luas dan pantai yang banyak (Ompusunggu, 2022).

Pulau Nias mempunyai keindahan pesona pantai yang luar biasa dan terdapat banyak pantai yang indah di ujung pulau. Salah satu pantainya yang terkenal yaitu Pantai Sorake Nias Selatan, memiliki ombak yang besar dan mencapai ketinggian hingga 15 kaki (m). Oleh karena itu, ini dianggap sebagai tempat selancar terbaik kedua setelah Hawaii. Selain Sorake *Beach*, masih banyak pantai lain yang dapat dikunjungi di Pulau Nias oleh wisatawan mancanegara maupun domestik di antaranya Pantai Hoya Gunungsitoli, Pantai Indah Fofola, Pantai Saiti, Pantai Indah Tureloto, Pantai Toyolawa, Pantai Merah Afulu, Pantai Sirombu, Pantai Ladeha, Pantai Lagundri, Pantai Blessing, dan Pantai Pulau Tello. Pengembangan pariwisata lokal dapat memberikan banyak manfaat ekonomi, sosial dan budaya bagi penduduk setempat. (Lahagu, 2018).

Karena banyaknya pantai yang ada di Pulau Nias, tentunya wisatawan memerlukan informasi yang jelas mengenai lokasi pantai serta jalur yang harus

ditempuh untuk menuju lokasi pantai yang ingin dikunjungi dan waktu yang dibutuhkan agar wisatawan dapat memaksimalkan perjalanan wisatanya (Masri, Kiswanto, & Kusuma, 2019). Ada beberapa hal yang perlu diperhatikan saat melakukan perjalanan, untuk semua wisatawan harus memilih jarak terpendek ke tujuan mereka karena agar bisa menghemat waktu, energi dan bahkan biaya bahan bakar. Bepergian tanpa mengetahui rute terpendek atau berapa lama waktu yang dibutuhkan untuk mencapai pantai yang mau dikunjungi dapat melebihi anggaran perjalanan dan mengurangi suasana kehangatan liburan kalian.

Dari permasalahan di atas maka penulis membuat sistem pencarian jalur terpendek yang dapat digunakan oleh wisatawan sebagai bahan pertimbangan untuk menentukan alternatif lokasi objek wisata pantai yang satu arah atau lokasinya berdekatan, sehingga menghemat waktu perjalanan ke lokasi pantai yang ingin dikunjungi.

Pada penelitian ini menggunakan metode Algoritma *Dijkstra* untuk menentukan rute terpendek bagi wisatawan dalam mengunjungi seluruh pantai yang ada di Pulau Nias. Algoritma itu sendiri adalah proses komputasi yang mengubah kuantitas *input* menjadi kuantitas *output*. Algoritma dikatakan "benar" jika menghasilkan *output* yang benar untuk setiap *input*. Setiap algoritma memiliki perbedaan dalam mencapai tujuan (Ginting & Barus, 2018). Sederhananya, algoritma adalah formula untuk menemukan solusi dari masalah yang ada. Menurut para ahli, algoritma didefinisikan sebagai proses langkah demi langkah yang sistematis, logis dan komprehensif untuk memecahkan suatu masalah. (Atmojo & Susiana, 2019).

Ada banyak algoritma yang bisa digunakan untuk menentukan rute terpendek pada sebuah graf yang terarah dan salah satunya Algoritma *Dijkstra* ini. Algoritma *Dijkstra* akan mencari jalur terpendek jika terdapat dua *node* atau lebih yang menjadi *node* awal dan *node* tujuan. Algoritma ini bekerja dengan cara membandingkan bobot terkecil berdasarkan *node* awal dan akhir untuk menemukan jalur paling efisien. (Masri, Kiswanto, & Kusuma, 2019).

Berdasarkan penelitian sebelumnya yang dilakukan oleh (Hamdi & Prihandoko, 2018) Algoritma *Dijkstra* dapat menyelesaikan masalah pencarian jalur terpendek dari simpul A ke simpul Z dalam graf berbobot. *Dijkstra* adalah suatu jenis algoritma umum yang memecahkan masalah optimisasi untuk menemukan jalur terpendek. Jalur terpendek dari simpul a ke z pada graf berbobot memiliki bobot atau bilangan positif. Oleh karena itu, tidak dapat ditransfer sebagai bilangan negatif. Saat mencari jalur terpendek, algoritma *Dijkstra* bekerja dengan cara mencari bobot minimum dari graf berbobot yang diseleksi antara dua titik atau lebih pada sebuah graf. Jumlah jalur yang diperoleh akan menjadi nilai yang paling terendah.

Berdasarkan penjelasan di atas terkait penggunaan metode Algoritma *Dijkstra*, penulis mencoba menerapkannya pada pencarian lintasan terpendek ke wisata pantai pulau Nias. Dengan judul "Implementasi Algoritma *Dijkstra* untuk Menentukan Rute Terpendek ke Wisata Pantai Pulau Nias" bisa menentukan jalur lokasi wisata yang efektif bagi wisatawan dengan jarak rute terkecil dan waktu tempuh yang sedikit untuk menuju lokasi pantai yang ada di pulau Nias.

1.2 Perumusan Masalah

Berdasarkan informasi yang diberikan oleh penulis di atas, maka rumusan masalah dalam penelitian ini berbunyi antara lain:

1. Bagaimana menggunakan Algoritma *Dijkstra* dalam menentukan jalur lokasi wisata yang efektif bagi penggunanya menuju lokasi pantai yang ada di Pulau Nias,
2. Bagaimana cara melakukan pengujian pencarian rute terpendek menggunakan Algoritma *Dijkstra* terhadap 12 titik lokasi pantai Pulau Nias yang akan dikunjungi oleh wisatawan agar menghasilkan rute terpendek yang memiliki jarak rute terkecil dan waktu tempuh yang sedikit.

1.3 Batasan Penelitian

Adapun batasan-batasan masalah dalam pengerjaan skripsi ini, antara lain:

1. Titik awal lokasi dimulai dari kota Gunungsitoli Pulau Nias
2. Titik lokasi tujuan yaitu destinasi wisata pantai Pulau Tello Nias Selatan
3. Terdapat 12 titik lokasi pantai yang menjadi objek penelitian yaitu Pantai Hoya Gunungsitoli, Pantai Indah Fofola, Pantai Saiti Nias Utara, Pantai Indah Tureloto Lahewa, Pantai Toyolawa Nias Utara, Pantai Merah afulu Nias Utara, Pantai Sirombu, Pantai Ladeha Nias Selatan, Pantai Sorake, Pantai Lagundri, Pantai Blessing, dan Pantai Pulau Tello
4. Output pada system yang dibuat berupa rute terpendek
5. Aplikasi pencarian rute terpendek ini dibuat berbasis *website*
6. Menggunakan program *PHP* dan menggunakan *database MySQL*

7. Penelitian dilakukan di Pulau Nias & data diambil di Pantai Blessing Nias Selatan.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang penulis sampaikan di atas, adapun tujuan dari pengerjaan skripsi ini, antara lain:

1. Mengoptimalkan jalur lokasi wisata menggunakan Algoritma *Dijkstra* sehingga menghasilkan jalur yang efektif bagi wisatawan untuk mengunjungi seluruh pantai yang ada di Pulau Nias.
2. Melakukan pengujian pencarian rute terpendek terhadap 12 titik lokasi pantai Pulau Nias yang akan dikunjungi oleh wisatawan, untuk mengetahui jarak rute yang terkecil dan waktu tempuh yang sedikit.

1.5 Manfaat Penelitian

Berdasarkan uraian tujuan penelitian tersebut, adapun manfaat dari pembuatan skripsi ini, antara lain:

1. Membantu wisatawan dalam menentukan rute perjalanan wisata dengan jarak terpendek dari kota Gunungsitoli ke destinasi wisata pantai yang di pulau Nias,
2. Menjadi sumber pengetahuan bagi penelitian yang sejenis dan memperoleh rute perjalanan dengan jarak rute terpendek dan waktu tempuh terbaik dalam mengunjungi lokasi pantai di pulau Nias.

1.6 Sistematika Penulisan

Sistematika penulisan dalam proposal skripsi ini terdiri dari 5 bab, dengan penjelasan sebagai berikut:

1. Bab I Pendahuluan, bab ini berisi tentang penjelasan terkait latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat, hipotesis, dan sistematika penulisan
2. Bab II Tinjauan pustaka, bab ini menjelaskan tentang dasar teori mengenai algoritma, graf, google maps, website, bahasa pemrograman *PHP*, *database MySQL*, Algoritma *Dijkstra*, tahapan Algoritma *Dijkstra*, dan penelitian terdahulu
3. Bab III Metodologi penelitian, bab ini menjelaskan mengenai metode penelitian yang digunakan, objek penelitian, teknik pengumpulan data, tahapan penelitian yang dilakukan, rancangan system aplikasi, dan rancangan *user interface*.
4. Bab IV Hasil dan pembahasan, bab ini menjelaskan mengenai Algoritma *Dijkstra* dalam pencarian rute terpendek, rute perjalanan menuju destinasi wisata pulau nias, analisa data rute perjalanan, pengolahan data rute perjalanan, pengujian system aplikasi, pengujian implementasi Algoritma *Dijkstra* dan pembahasan.
5. Bab V Kesimpulan dan Saran, bab ini menjelaskan mengenai kesimpulan dan saran dari penelitian yang telah dilakukan.

BAB II
TINJAUAN PUSTAKA

2.1 Algoritma

Menurut (Qomaruddin, Bismi, & Hariyanto, 2022) Kata "algoritma" (*algorithm*) itu berasal dari nama sarjana Arab terkenal Abu Jafar Muhammad ibnu Musa al-Khawarizmi. Orang Barat menyebutnya "*algorism*". Algoritma juga merupakan instruksi yang sistematis dan tepat untuk memecahkan masalah dengan logis. Secara umum, metode pemecahan masalah dapat dibagi menjadi beberapa kategori.

1. Strategi solusi langsung;
2. Strategi pencarian pada ruang status;
3. Strategi solusi atas ke bawah
4. Strategi solusi dari bawah ke atas.

Menurut (Grace, Tanciga, & Nurdin, 2018) Algoritma adalah teknik komputasi yang dapat menghasilkan sejumlah input dan output. Jika algoritma menghasilkan output yang benar untuk setiap inputnya, maka algoritma tersebut dianggap "benar (*correct*)." Dalam kasus ini, algoritma memberikan deskripsi langkah-langkah yang relevan untuk menyelesaikan masalah. Deskripsi langkah-langkah tersebut bisa dituliskan ke dalam bentuk notasi algoritma. Notasi algoritma adalah rancangan penyelesaian masalah yang ditulis dalam notasi (tata cara penulisan). Dua bentuk notasi algoritma yang paling umum adalah diagram alir (*flowchart*) dan *pseudo-code*. Teks algoritma harus mudah dibaca dan dipahami, dan tidak ada aturan baku untuk itu.

Menurut (Muharrom, 2020) Algoritma memiliki lima elemen kunci, antara lain keterbatasan, kepastian, masukkan, keluaran, dan kemanjuran. Dengan demikian, pengembangan suatu algoritma membutuhkan tiga elemen:

1. Elemen *input*, bagian ini biasanya mencakup parameter variabel, jenis variabel, konstanta, dan paramenter (dalam fungsi).
2. Elemen *output*, bergantung pada algoritma dan rancangan program. Bagian ini mencantumkan algoritma dan tugas yang diselesaikan oleh program.
3. Elemen *processing*, bagian ini sangat penting untuk pengembangan algoritma, karena berkaitan dengan logika tugas, logika algoritma (sintaksis, semantik), jenis dan metode (rekursi, perbandingan, penambahan, pengurangan, dll.).

Berdasarkan kondisinya, komponen algoritma terdiri dari dua kondisi, antara lain: 1. kondisi awal, adalah kondisi suatu program ketika algoritma siap dijalankan, atau ketika algoritma siap dijalankan. Kondisi awal ditentukan melalui input algoritma. 2. kondisi akhir, adalah kondisi setelah algoritma selesai dijalankan.

Menurut (Hamdi & Prihandoko, 2018) Algoritma secara informal didefinisikan sebagai serangkaian langkah komputasi yang mengubah input menjadi output. Mereka mengambil nilai atau sekumpulan nilai sebagai masukan dan menghasilkan nilai atau kumpulan nilai sebagai keluaran.

2.2 Pengertian Graf

Menurut (Grace, Tanciga, & Nurdin, 2018) Graf adalah bidang ilmu yang luas. Batuan graf dapat menyelesaikan banyak masalah, dan graf dapat mewakili banyak struktur. Cara umum untuk merepresentasikan sebuah jaringan adalah

melalui grafik. Dalam sistem navigasi, kota biasanya ditempatkan sebagai *node* (*vertex*/*node*) dan jalan antar kota sebagai sisi (*edge*). Panjang jalan diwakili oleh bobot (*weight*) dari setiap sisi, dan dalam model persoalan tertentu, bobot dari setiap sisi dapat bernilai negatif. Dalam kasus ini, simpul mewakili kota, dan sisi mewakili perjalanan yang memungkinkan. Biaya yang dikeluarkan untuk perjalanan atau jarak diwakili oleh bobot dari setiap sisi.

Menurut (Qomaruddin, Bismi, & Hariyanto, 2022) Graf adalah struktur diskrit dengan simpul (*vertex*) dan sisi (*edge*). Sejumlah grafik dibuat oleh pasangan V dan E; di mana E adalah himpunan sisi yang menghubungkan dua simpul dari grafik dan V terdiri dari simpul yang tidak kosong. Dua jenis umum graf dapat dibedakan jika dilihat dari arah pada sisi, yaitu:

1. Graf tak berarah; Jenis graf yang arah sisinya tidak ditentukan dan urutan simpul yang dihubungkan oleh sisi tidak diperhitungkan. Jadi (vj, vk) = (vk, vj) memiliki satuan yang sama.
2. Graf berarah; Jenis graf yang sisinya memiliki arah tertentu dan tujuan yang jelas. Busur (arc) adalah istilah untuk sisi yang berarah. Pada graf berarah (VJ, VK) dan (VK, VJ) direpresentasikan oleh dua sisi dengan arah yang berbeda. Oleh karena itu, mereka tidak sama. Busur (VJ, VK), simpul VJ disebut simpul awal dan simpul VK disebut simpul akhir.

Menurut (Hamdi & Prihandoko, 2018) meskipun teori graf adalah topik yang lama, graf memiliki banyak aplikasi modern dalam kehidupan sehari-hari. Graf menunjukkan objek diskrit dan hubungan mereka. Contoh representasi visual dari grafik adalah peta, yang dapat digunakan untuk menyelesaikan banyak masalah di

dunia nyata. Peta dapat digunakan untuk banyak hal, seperti menentukan jalur terpendek dari satu tempat ke tempat lainnya, menentukan tata letak jalur transportasi, membangun jaringan telekomunikasi atau internet, dan banyak lagi.

Dalam teori graf, menentukan jalur optimal dari satu tempat ke tempat lain merupakan masalah umum dalam kehidupan sehari-hari. Untuk mendapatkan pencarian jalur yang optimal tersebut bisa menggunakan salah satu algoritma dengan pendekatan *greedy* ialah menggunakan Algoritma *Dijkstra*.

2.3 Jalur Lintasan Terpendek (*Shortest Path*)

Menurut (Muharrom, 2020) Sejak akhir 1950-an, pembicaraan dan penelitian telah berfokus pada pencarian rute terpendek. Banyak sektor yang menggunakan pencarian rute terpendek untuk meningkatkan kinerja sistem, mengurangi biaya, dan mempercepat prosedur. Pengoptimalan dengan graf berbobot berarti menemukan jalur terpendek antara dua atau lebih simpul dalam jaringan berbobot dengan bobot sisi total paling kecil atau minimum. Bobot dapat diwakili oleh berbagai faktor, seperti jarak antar tempat, waktu yang dibutuhkan untuk pengiriman pesan, dan biaya pengembangan. Proses meminimumkan bobot pada suatu lintasan graf ditunjukkan dengan permasalahan kata terpendek pada suatu lintasan. Berikut ini adalah beberapa jenis masalah jalur terpendek, yaitu: jalur terpendek antara dua node, antara semua pasangan simpul, dari simpul tertentu ke semua node lainnya, dan dari dua node yang melewati simpul tertentu.

2.4 *Google Maps API*

Menurut (Ahdan & Setiawansyah, 2020) *Google Maps* awalnya dikembangkan oleh dua bersaudara dari Denmark, Lars dan Jens Rassmusen

sebagai solusi pemetaan, dan akhirnya dibeli oleh perusahaan *Google* pada bulan Oktober 2004. *Google Maps* adalah program pemetaan yang canggih dengan fitur seperti profil ketinggian dan tampilan citra satelit selain kemampuan untuk menjelajahi tempat, mencari alamat, dan menerima petunjuk arah mengemudi.

Menurut (Folaimam, Rosihan, & Mubarak, 2018) *Google Maps* menawarkan teknologi pemetaan yang mudah digunakan oleh *user* (*userfriendly*). Layanan *Google Maps* untuk wilayah Indonesia dapat digunakan dengan cara mengakses situs http://maps.google.com atau http://maps.google.co.id. Selain itu, *Google Maps* menawarkan petunjuk arah untuk mengemudi serta peta yang dapat ditarik dengan gambar satelit dari seluruh dunia. Setiap gambar yang ada di *Google Maps* diunggah ke server *Google* menggunakan berbagai gambar peta, *database*, dan objek interaktif yang dibuat dalam berbagai bahasa pemrograman, seperti *HTML*, *JavaScript*, dan *AJAX*.

Menurut (Setiawan, Prakoso, & Suryaningrum, 2019) Perkembangan *Google Maps* melahirkan *Google Maps API* yang memungkinkan seorang *programmer web* dapat menggunakan *Google Maps* di dalam *website* yang dibuatnya. Pada awalnya hanya *JavaScript API*, *Maps API* telah diperluas untuk mendukung aplikasi *Adobe Flash*. Karena popularitas *Google Maps API*, banyak saingan muncul, termasuk *OpenLayers*, *MapQuest Development Platform*, *Bing Maps Platform*, dan *Yahoo*! *Maps API*.

Menurut (Yulianto, Ramadiani, & Kridalaksana, 2018) API atau *Application Programming Interface*, tidak hanya sekedar kumpulan *class*, metode, fungsi, dan tanda tangan (*signature*). sebaliknya, *API* berusaha memecahkan masalah yang tidak jelas "*clueless*" yang muncul saat mengembangkan perangkat lunak

(*software*) dengan ukuran yang besar, dengan perilaku komponen samar dan berbagai tingkat kompleksitas, dari yang sederhana hingga yang rumit. Jika seseorang mempertimbangkan kesulitan yang muncul saat mengganti protokol atau *database XML*, batuan *API* dapat membuat penyesuaian ini mudah dilakukan. *API* merupakan kumpulan dari instruksi, prosedur, kelas, dan protokol, memungkinkan perangkat lunak (*software*) tertentu berkomunikasi dengan perangkat lunak lainnya. *API* bertujuan untuk menyingkirkan masalah atau instruksi yang tidak jelas pada *system* dengan membangun blok besar dari perangkat lunak (*software*) di seluruh dunia dan menggunakan kembali perintah, fungsi, kelas, protokol, atau *API* yang mereka punya. Ini meningkatkan efisiensi dan produktivitas seorang *programmer* karena mereka tidak perlu menghabiskan banyak waktu untuk merancang dan menulis infrastruktur.

Menurut (Paunsyah, Mubarok, & Shofa, 2019) *Google Maps API* adalah layanan (*service*) yang diberikan oleh *Google* kepada penggunanya untuk memungkinkan mereka menggunakan dan mengembangkan aplikasi peta digital dengan baik. Fitur *Google Maps API* bisa dimasukkan ke *website* yang mereka kembangkan untuk menghemat waktu dan sumber daya, sehingga pembuatan aplikasi bisa lebih berfokus pada data-data yang diolah.

2.5 *Website*

Menurut (Rumondor, Sentinuwo, & Sambul, 2019) *WWW* (*World Wide Web)* merupakan salah satu layanan yang dapat diakses oleh pengguna komputer yang terhubung ke internet dan memiliki jaringan. *WWW* atau lebih dikenal dengan nama *website*, ini mencakup berbagai informasi, mulai dari informasi yang berguna, berguna dan sekaligus penting, informasi berguna dan serius juga ada,

hingga informasi yang tidak berguna juga tersedia. Selain itu, informasi gratisan dan informasi komersial juga banyak disini. *Website* atau situs adalah kumpulan halaman *web* yang mengandung teks, foto, animasi, musik, dan kombinasi elemen yang berbeda, baik statis maupun dinamis. Halaman-halaman ini saling terhubung dan membentuk serangkaian bangunan yang terkait dan terhubung satu sama lain diciptakan oleh jaringan halaman (*hyperlink*).

Menurut (Nugraha & Syarif, 2018) *Website* adalah sumber informasi yang bisa diakses oleh banyak orang yang berada dalam jaringan, baik mereka yang memiliki koneksi internet atau tidak. *Website* merupakan kumpulan *hyperlink* yang dibuat dengan *HTML* (*HyperText Markup Language*) dan sangat banyak digunakan di internet. *Hypertext Markup Language* (*HTML*) adalah bahasa pengkodean umum yang digunakan untuk menata dokumen teks. Tujuan utama *HTML* adalah untuk memproses berbagai data dan informasi sehingga dokumen dapat dilihat dan ditampilkan secara online melalui layanan *web*.

2.6 Bahasa Pemrograman *PHP*

Menurut (Putra, 2018) *PHP* (dahulu dikenal sebagai *Personal Home Page*, sekarang *PHP*: *Hypertext Preprocessor* adalah program yang dikembangkan bersama oleh *programmer* dari berbagai negara yang tertarik dengan perangkat lunak *open source*. *PHP* Dirancang khusus untuk mengakses dan mengubah data pada server *database open source* seperti *MySQL*.

Menurut (Rumondor, Sentinuwo, & Sambul, 2019) *PHP*, juga dikenal sebagai *Personal Hypertext Preprocessor* merupakan bahasa *script* berbasis server yang dihosting. *PHP* adalah salah satu bahasa pemrograman yang digunakan untuk membuat *HTML* (*Hypertext Markup Language*) bekerja. Ini awalnya dibuat oleh

Rasmus Lerdorf sebagai proyek pribadi, kemudian disempurnakan oleh tim yang terdiri dari enam pengembang (*six of developer*) dan dirilis ulang sebagai *PHP*. *PHP* adalah bahasa *scripting* yang bekerja dengan *HTML* untuk membuat halaman *web* yang dinamis. Dengan kata lain, tidak hanya sintaks yang kita berikan berfungsi di server, tetapi juga ada dalam teks *HTML*, dan browser menerima hasilnya. Kode *PHP* dimulai dengan tanda lebih kecil "<" dan diakhiri dengan tanda lebih besar ">".

2.7 *Database MySQL*

Menurut (Ramadhan & Mukhaiyar, 2020) *Database* adalah kumpulan data digital yang diorganisasikan untuk satu atau lebih pengguna dan digunakan untuk mempermudah pengorganisasian, penyimpanan, dan akses data. *Database Management System* (*DBMS*) mengelola database digital ini, yang memiliki berbagai kemampuan untuk membuat konten *database*, memelihara, mencari, dan mendapatkan akses tambahan ke data. Saat ini, beberapa database yang dapat diakses termasuk *Oracle*, *PostgreSQL*, *Microsoft Access*, *MySQL*, dan *SQL* Server, antara lain. *Database* memiliki fitur, yaitu:

1. Mempermudah identifikasi data dengan mengelompokkan data, misalnya dengan membuat beberapa tabel atau field yang berbeda-beda
2. Mengurangi jumlah data ganda
3. Mempermudah penggunaan oleh user dalam berbagai situasi, seperti ketika mereka memasukkan data baru
4. Menyimpan secara digital; dan
5. Berfungsi sebagai solusi alternatif untuk masalah penyimpanan ruang dalam aplikasi.

Menurut (Rumondor, Sentinuwo, & Sambul, 2019) *Database MySQL* adalah program *database client*-server berbasis *console* yang tersedia dalam bentuk teks atau kode-kode. *MySQL* tergolong sebagai *DBMS* (*Database Management System*), tersedia secara *open source* di bawah *GPL* (*General Public License*). Menurut lisensi sumber terbuka (*open source*), siapa pun dapat menggunakan *MySQL* tanpa batasan, tetapi tidak dapat diubah menjadi produk yang dapat dijual. Struktur *Query Language* (*SQL*) adalah komponen utama dalam memilih dan memasukkan data, dan membuat tindakan pada data menjadi sederhana dan mudah. *MySQL* biasanya digunakan saat membuat aplikasi berbasis *website* dengan bahasa pemrograman *PHP*.

Menurut (Ramadhan & Mukhaiyar, 2020) *MySQL* merupakan salah satu kategori *database* server yang paling umum digunakan karena mengakses database melalui bahasa *SQL*. Baik versi komersial maupun *FOSS License Exception* adalah lisensi *MySQL*. Menjadi "*database opensource* yang paling banyak digunakan di dunia" adalah kalimat singkat sebagai *branding* atau *tagline* dari *MySQL*. *MySQL* didukung oleh banyak sistem operasi, termasuk *Windows* dan *Linux*. Kita dapat menggunakan software seperti *PhpMyAdmin* atau *mysql yog* untuk mempermudah pengelolaan *database MySQL*. *PhpMyAdmin* adalah program atau alat *open source* gratis untuk pengkodean dan pemeliharaan *database MySQL*, yang menyediakan aktivitas *MySQL* seperti manajemen basis data, tabel, bidang (*fields*), hubungan (*relations*), indeks, pengguna (*users*), dan izin. *PhpMyAdmin* menggunakan bahasa pemrograman *PHP*, dengan begitu *PhpMyAdmin* dan *MySQL* tidaklah sama.

2.8 Algoritma *Dijkstra*

Menurut (Serdano, Zarlis, & Hartama, 2019) Algoritma *Dijkstra* merupakan salah satu bentuk dari algoritma *greedy*, menggunakan daftar yang berdekatan untuk menunjukkan jalur yang akan dilalui dan menemukan jalur terpendek dengan satu tujuan pada jalur tanpa sisi *cost* negatif dan dengan biaya serendah mungkin. Seorang ilmuwan komputer Belanda bernama Edsger *Dijkstra* menemukan metode ini pada tahun 1956, dan diterbitkan pada tahun 1959. Menjelang tahun 2000, sudah banyak algoritma yang memiliki kemampuan untuk menemukan jalur terpendek. Salah satu cara yang dapat menyelesaikan masalah pencarian rute terpendek adalah dengan menggunakan Algoritma *Dijkstra* karena cocok untuk penyelesaian kasus yang menggunakan graf alternatif.

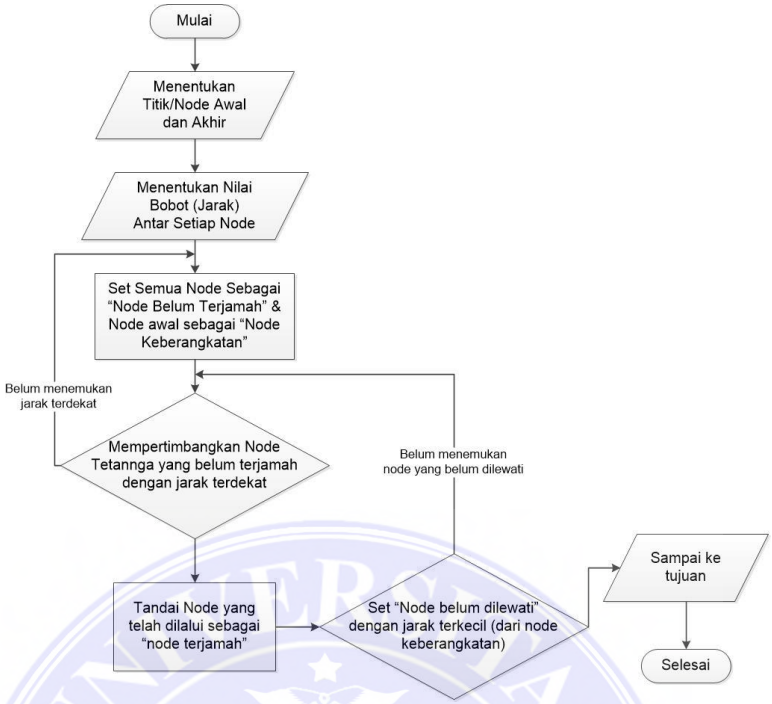
Menurut (Deepa, Kumar, Manimaran, Rajakumar, & Krishnamoorthy, 2018) Ada banyak teori untuk memecahkan masalah jalur terpendek dan salah satu solusi cara yang banyak digunakan adalah Algoritma *Dijkstra* (AD) yang juga banyak digunakan dalam banyak pekerjaan perhitungan teknik juga. Cara kerja dari algoritma *Dijkstra* adalah diberikan sebuah graf dengan semua simpul, pilih salah satunya sebagai simpul sumber, dan kemudian temukan jalur terpendek (*Shortest Path*). Pertama, kita harus membuat "*Shortest Path Tree* (SPT), atau pohon jalur terpendek dengan sumber sebagai akar." Kita juga harus mempertahankan "dua set pertama berisi simpul yang termasuk dalam SPT dan set lainnya berisi simpul yang belum termasuk dalam SPT." Dalam setiap langkah kita harus menemukan simpul yang merupakan himpunan terakhir dan memiliki jarak minimum dari sumber. Algoritma ini memecahkan masalah jalur terpendek (*Shortest Path*) di bawah

batasan bahwa seharusnya tidak ada siklus berbobot *negatif* yang ada dalam graf. Jika ada siklus berbobot negatif maka jalur terpendek tidak akan terdeteksi.

2.9 Tahapan Algoritma *Dijkstra*

Menurut (Sidhu & Krishan, 2022) Tahapan yang dilalui Algoritma *Dijkstra* untuk menemukan jalur terpendek adalah dimulai dengan memilih simpul awal dan simpul tujuan, kemudian mengatur jarak antar setiap simpul yang ada, dan mencari jarak terpendek dengan menyeleksi semua simpul untuk menemukan simpul tujuan. Algoritma *Dijkstra* bekerja dengan cara menghitung semua jalur terpendek dari satu titik ke semua titik lainnya. Adapun tahapannya, sebagai berikut:

1. Tetapkan simpul awal dan simpul tujuan, dengan simpul awal = 0
2. Tetapkan jarak antar setiap simpul yang ada
3. Hitung jarak terpendek antara simpul awal dengan semua tetangga
4. Jika jarak simpul telah dihitung, perbarui jarak terpendek
5. Perbarui simpul sebelumnya ke daftar simpul yang telah dikunjungi.
6. Untuk simpul saat ini, periksa tetangganya.
7. Ulangi 3,4,5,6 sampai menemukan simpul tujuan dan didapat jarak terpendek.



Gambar 2.1 Flowchart Proses Algoritma *Dijkstra*

Menurut (Cantona, Fauziah, & Winarsih, 2020) Tahapan-tahapan penerapan Algoritma *Dijkstra* untuk menentukan rute terpendek pada sebuah kasus antara lain:

1. Menentukan titik awal sebagai *node* pertama dan titik akhir sebagai *node* tujuan,
2. Menentukan nilai bobot, atau jarak antara setiap *node* yang ada,
3. Menetapkan *node* awal sebagai "*node* keberangkatan" dan semua *node* "belum terjamah",
4. Mulai mencari rute dengan jarak terkecil, dimulai dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum terjamah dan hitung jarak dari titik keberangkatan,

5. Tandai node yang telah dilalui sebagai "*node* terjamah" dan *node* terjamah tidak akan pernah di cek kembali,

6. Set "*node* belum terjamah" dengan jarak terkecil (dari *node* keberangkatan) sebagai "*node* keberangkatan" selanjutnya dan lanjutkan dengan kembali ke tahapan ke-4.

7. Setelah mempertimbangkan semua jarak terhadap node tetangga dan menemukan *node* tujuan, maka proses dinyatakan selesai. Jarak yang disimpan adalah jarak terakhir dengan bobot terkecil.

2.10 Penelitian Terdahulu

Penelitian ini tidak dapat dilakukan tanpa jurnal atau sumber pendukung. Berikut adalah beberapa topik yang relevan dengan judul penelitian yang digunakan oleh peneliti sebagai bahan referensi selama penulisan skripsi ini.

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	Judul	Hasil
1.	Wahyu Tisno Atmojo dan Susiana, 2019,	Implementasi Algoritma *Dijkstra* Untuk Mencari Rute Tujuan Wisata Terpendek Dalam Aplikasi Berbasis *Web*	Sangat sulit untuk menghitung jarak yang harus ditempuh untuk mencapai lokasi wisata. Masalah besar lainnya adalah para pelancong harus memilih rute mereka secara manual, yang mereka anggap kurang efisien. Dalam situasi seperti ini, rute yang memiliki jalur terpendek harus ditempuh untuk mengurangi waktu tempuh dan biaya.

			Metode *waterfall* digunakan untuk mengembangkan sistem aplikasi yang akan dibuat dalam penelitian ini. Selain itu, untuk memodelkan sistem yang akan berjalan pada aplikasi, menggunakan *Unified Model Language* (UML), termasuk *Use Case Diagram* dan *Activity Diagram*. Pengumpulan data menggunakan studi literatur dan observasi. Aplikasi pencarian rute perjalanan terpendek ini dibuat berbasis *web* dan menggunakan algoritma *Dijkstra* untuk menghitung jarak antara lokasi awal dan akhir tujuan wisata tanpa mengulangi tujuan wisata lainnya.
2.	(Rumondor, Sentinuwo, & Sambul, 2019)	Perancangan Jalur Terpendek Evakuasi Bencana di Kawasan Boulevard Manado Menggunakan Algoritma *Dijkstra*	Provinsi Sulawesi Utara sangat rentan terhadap gempa bumi dan tsunami, yang dapat mencapai ketinggian lima meter. Oleh karena itu, algoritma *Djikstra* digunakan untuk membuat jalur evakuasi terpendek untuk bencana gempa dan tsunami. Ini akan membantu mitigasi bencana dengan memberikan informasi tentang jalur terpendek yang akan dilalui dan mengajarkan orang bagaimana menyelamatkan diri jika terjadi bencana gempa dan tsunami.

			Pada penelitian ini, metode *waterfall* digunakan sebagai dasar perancangan aplikasi, dan *use case* diagram digunakan dalam proses perancangan dan pengembangan sistem untuk menemukan jalur terpendek menuju evakuasi dari bencana gempa bumi dan tsunami. Penelitian ini juga menggunakan bahasa pemrogaraman *HTML* dan *PHP*, dan basis datanya disimpan di *database MySQL*. Sebuah sistem informasi grafis (SIG) berbasis *web* akan digunakan untuk menentukan rute evakuasi terpendek yang dapat ditempuh seorang pengunjung dari *Boulevard* Manado ke area evakuasi aman (*shelter*). Selain itu, GIS menghitung jumlah waktu yang dibutuhkan, jarak, dan arah dari lokasi bencana ke lokasi evakuasi.
3.	Mahrizal Masri, Andika Pratomo Kiswanto, dan Budhi Santri Kusuma, 2019	Implementasi Algoritma *Dijkstra* dalam Perancangan Aplikasi Penentuan Rute Terpendek Pada Objek Pariwisata	Karena banyaknya tempat wisata di sekitaran Danau Toba, para wisatawan ingin mengetahui lokasi tempat-tempat wisata tersebut dan jalur yang harus mereka tempuh untuk bisa sampai kesana. Algoritma *Djikstra* mengatasi masalah pencarian jalur terpendek dengan

		Danau Toba dan Sekitarnya	mencari jalur terpendek dari simpul awal hingga simpul akhir. Untuk membuat aplikasi pencarian jalur terpendek yang diinginkan, penelitian ini menggunakan Data *Flow* Diagram (DFD), *Use Case Diagram*, dan desain *database*. Aplikasi pencarian jalur terpendek yang akan dibangun menggunakan database *MySQL* dan berbasis *web*. Dari hasil penelitian, menunjukkan bahwa metode *Dijkstra* dapat digunakan dalam sistem informasi geografis untuk menemukan rute terpendek yang akan ditempuh wisatawan untuk mencapai tujuan. Hasil pencarian rute terpendek menunjukkan jarak tempuh dan waktu tempuh dari lokasi awal ke titik tujuan.
4.	Aldy Cantona, Fauziah, dan Winarsih, 2020	Implementasi Algoritma *Dijkstra* Pada Pencarian Rute Terpendek ke Museum di Jakarta	Banyak museum di Jakarta yang bersejarah, sebagian besar menyimpan sejarah hingga kemerdekaan Indonesia saat ini. Metode *Djikstra* dapat digunakan untuk menentukan jarak paling dekat dari satu titik ke museum yang dipilih sebagai tujuan. Aplikasi ini dibuat untuk mencari museum dengan jarak paling pendek di kota Jakarta.

			Piranti *Flutter* digunakan untuk membuat aplikasi dengan bahasa pemrograman *Dart* dalam penelitian ini, yang menghasilkan keluaran (output) yang sesuai dengan desain yang telah dibuat. Selain itu, menggunakan *flowchart* untuk menjelaskan alur proses seluruh tahapan dalam mencari rute terdekat menuju museum. Untuk mempersingkat proses mencari museum yang ada di Jakarta, aplikasi pencarian rute terpendek dipasang di *smartphone*. Pengujian menunjukkan bahwa algoritma *Djikstra* menghasilkan rute terpendek dengan nilai efektif untuk dilalui dengan mobil, karena hanya memerlukan 7 bobot, atau setara 35% dari total 20 bobot, untuk mencapai Museum Nasional. Dengan menghiraukan kemacetan dan kondisi ganjil genap di Jakarta.
5.	Jenni Veronika Ginting, Ertina	Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma *Dijkstra*	Jika seseorang membutuhkan perhatian medis, mereka harus pergi ke rumah sakit. Dalam keadaan darurat, jalur terpendek adalah yang paling penting. Karena letak rumah sakit di kota Kisaran berbeda-beda, sehingga ada banyak rute yang dapat dipilih sesuai dengan kebutuhan pasien. Oleh karena itu, algoritma *Dijksta* digunakan

	Sabarita Barus. 2018		untuk membuat aplikasi penentuan rute terpendek rumah sakit yang melakukan perhitungan untuk menemukan rute terpendek dari titik awal ke titik akhir tujuan. Pada penelitian ini menggunakan studi literatur dalam mencari teori yang relevan dan melakukan wawancara ke pihak terkait untuk mendapatkan data-data yang diperlukan. Pada pembuatan aplikasi menggunakan bahasa pemrograman *PHP*, *database MySQL* dan *coding editor Dreamweaver CS3*. Dari hasil penelitian, pada sebuah website, algoritma *Djikstra* digunakan untuk membangun aplikasi yang mencari jalur terpendek menuju rumah sakit di Kota Kisaran. Jalur terpendek dengan bobot paling sedikit dapat ditemukan dengan menggunakan algoritma ini. Untuk meningkatkan kontrol internal dan mencegah pihak yang tidak berhak mengakses data, sistem aplikasi membatasi akses setiap pengguna.

BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Objek Penelitian

Pada penelitian ini, objek yang akan diteliti ialah tempat destinasi wisata pantai. Pantai-pantai ini berlokasi di pulau Nias, salah satu pantai yang terkenal di Pulau Nias ialah pantai sorake letaknya berada di Kabupaten Nias Selatan, Kecamatan Teluk Dalam. Selain pantai sorake, terdapat juga pantai sirombu di pulau Nias letaknya di Kabupaten Nias Barat, Kecamatan Sirombu. Terdapat juga pantai lagundri dan pantai blessing di pulau Nias letaknya tidak jauh dari pantai sorake dan masih ada pantai lainnya.

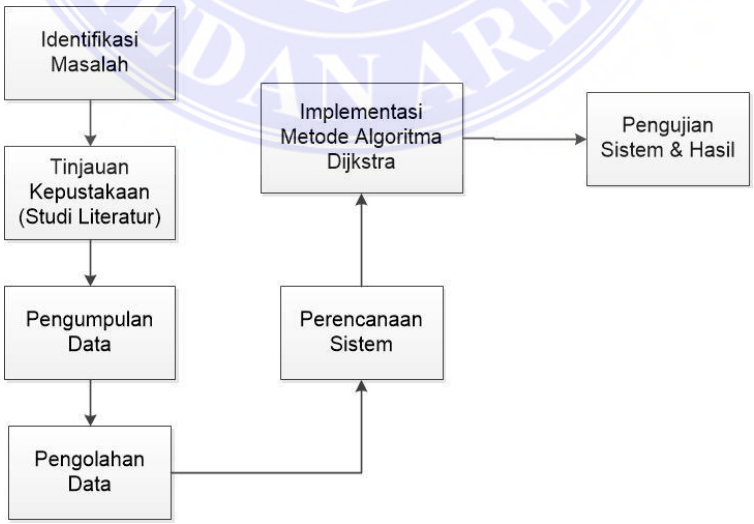
Tabel 3.1 Objek Penelitian

No	Pantai Nias	Kecamatan	Lokasi
1	Pantai Hoya Gunungsitoli	Gunungsitoli Utara	Jl. Pantai Tlk. Belukar, Gunungsitoli
2	Pantai Indah Fofola	Tuhemberua	Desa Banuagea, Nias Utara
3	Pantai Saiti Nias Utara	Sawo	Desa Seriwau, Nias Utara
4	Pantai Indah Tureloto	Lahewa	Jl. Tureloto, Balefadoro, Nias Utara
5	Pantai Toyolawa	Lahewa	Jl. Raya Toyolawa, Hili Gawolo, Nias Utara
6	Pantai Merah Afulu	Afulu	Jl. Lahewa-Afulu, Harewakhe, Nias Utara
7	Pantai Sirombu	Sirombu	Jl. Tetesua, Sirombu, Nias Barat

8	Pantai Ladeha	Amandraya	Desa Lolomoyo, Nias Selatan
9	Pantai Sorake	Fanayama	Jl. Sorake, Botohilitano, Nias Selatan
10	Pantai Lagundri	Teluk Dalam	Jl. Hiliamaetaniha, Lagundri, Nias Selatan
11	Pantai Blessing	Teluk Dalam	Desa Hilitobara, Nias Selatan
12	Pantai Pulau Tello	Pulau-pulau Bata	Desa Bawodobara, Nias Selatan

3.2 Metode Penelitian

Pada metode penelitian berisi tentang tahapan-tahapan dilakukannya penelitian yang dituangkan dalam bentuk diagram alir dan menggambarkan proses penelitian yang ditempuh secara menyeluruh, mulai dari identifikasi masalah hingga pengujian system dan hasil penelitian. Berikut tahapan-tahapan penelitian yang dilakukan, antara lain:



Gambar 3.1 Diagram Alir Tahapan Penelitian

Pada tahap awal, melakukan identifikasi masalah dan diteruskan dengan tinjauan kepustakaan atau studi literatur, dengan mencari referensi dari beberapa buku, jurnal, dan karya tulis yang berhubungan dengan Algoritma *Dijkstra*. Setelah tinjuan kepustakaan seterusnya di lanjutkan dengan pengumpulan data yang dilakukan dengan cara observasi atau pengamatan secara langsung, melakukan wawancara terhadap pengelola salah satu pantai di Nias. Dan dengan studi pustaka melalui google maps untuk mendapatkan rute perjalanan sementara, jarak rute, dan waktu tempuh menuju lokasi pantai yang ada di Nias.

Setelah pengumpulan data dilakukan seterusnya dilanjutkan dengan pengolahan data secara manual untuk memodelkan rute yang akan menjadi titik awal dan titik akhir yang akan dilalui untuk mendapatkan rute terpendek dengan Algoritma *Dijkstra*.

Setelah pengolahan data dilakukan seterunya di lanjutkan dengan perencanaan system meliputi perancangan system yang di dalamnya terdapat (*use case* diagram, flowchart system pencarian rute, flowchart *login admin*, *flowchart* halaman data informasi pantai, flowchart halaman data informasi *node*, dan flowchart data informasi *admin*), dan perancangan *interface* mulai dari tampilan menu utama, tampilan *form login admin*, tampilan menu utama *admin*, tampilan *form* tambah data pantai, tampilan *form* tambah data *node*, dan tampilan form tambah data *admin*.

Setelah itu di lanjutkan dengan tahapan implementasi metode Algoritma *Dijkstra* untuk menentukan rute terpendek ke dalam system dan tahapan terakhir melakukan pengujian system pencarian rute terpendek apakah system bekerja

dengan baik untuk menghasilkan jalur alternatif berupa rute terpendek menggunakan metode Algoritma *Dijkstra*.

3.3 Data Penelitian

Pada penelitian ini, data yang digunakan sebagai bahan penelitian adalah nama-nama pantai yang ada di kepulauan Nias, informasi mengenai pantai tersebut, informasi mengenai rute perjalanan dimulai dari lokasi keberangkatan yaitu kota Gunugsitoli menuju wisata pantai Nias untuk penggambaran graf manual, jarak antar lokasi pantai untuk melakukan proses pengolahan data dalam pencarian rute terpendek secara manual, dan titik koordinat lokasi pantai untuk membuat penanda titik lokasi ke dalam sistem aplikasi pencarian rute terpendek. Data penelitian ini diambil dari sumber *Google Maps*.

Tabel 3.2 Koordinat Lokasi Pantai

No	*Node*	Pantai Nias	Latitude	Longitude
1	A	Pantai Hoya Gunungsitoli	1.397673	97.541077
2	B	Pantai Indah Fofola	1.471234	97.448191
3	C	Pantai Saiti Nias Utara	1.521130	97.413059
4	D	Pantai Indah Tureloto	1.429592	97.142684
5	E	Pantai Toyolawa	1.406904	97.082300
6	F	Pantai Merah Afulu	1.287441	97.205642
7	G	Pantai Sirombu	0.949433	97.415004
8	H	Pantai Ladeha	0.758540	97.636602
9	I	Pantai Sorake	0.570293	97.732000
10	J	Pantai Lagundri	0.580437	97.741593

11	K	Pantai Blessing	0.549983	97.809650
12	L	Pantai Pulau Tello	-0.040738	98.276879

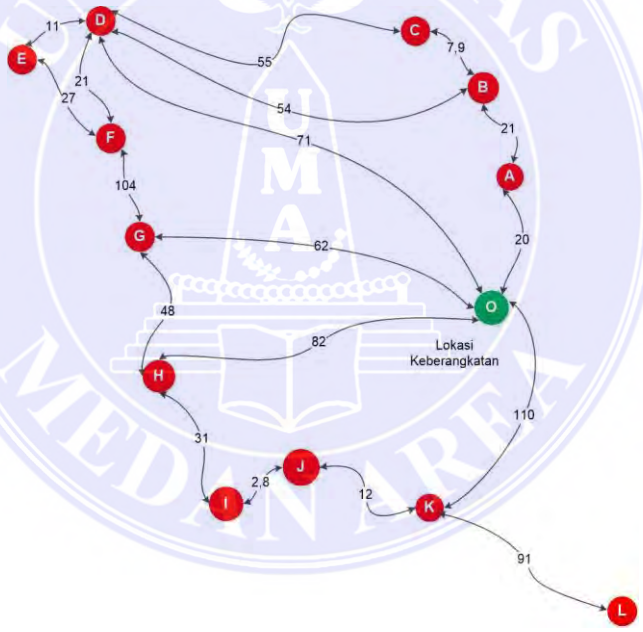
Tabel 3.2 merupakan keterangan titik koordinat lokasi pantai yang ada di pulau Nias sekaligus menjadi data untuk bahan penelitian. Titik koordinat lokasi pantai tersebut bertujuan untuk memudahkan saat pembuatan penanda titik lokasi pantai ke dalam system aplikasi pencarian rute terpendek.

Tabel 3.3 Jarak Antar Setiap *Node* Dari *Maps*

No	Keterangan Lokasi	*Node* awal	*Node* Tujuan	Jarak antar *node* (km)
1	Kota Gunungsitoli	O	A	20
			D	71
			G	62
			H	82
2	Pantai Hoya Gunungsitoli	A	B	21
3	Pantai Indah Fofola	B	C	7,9
			D	54
4	Pantai Saiti Nias Utara	C	D	55
5	Pantai Indah Tureloto	D	E	11
			F	21
6	Pantai Toyolawa	E	F	27
7	Pantai Merah Afulu	F	G	104
8	Pantai Sirombu	G	H	48
9	Pantai Ladeha	H	I	31

10	Pantai Sorake	I	J	2,8
11	Pantai Lagundri	J	K	12
12	Pantai Blessing	K	L	91
13	Pantai Pulau Tello	L	-	-

Tabel 3.3 merupakan keterangan jarak antar setiap *node* dalam satuan km, dimana jarak antar setiap *node* tersebut digunakan untuk memudahkan saat penggambaran graf manual dan melakukan proses pengolahan data dalam pencarian rute secara manual.



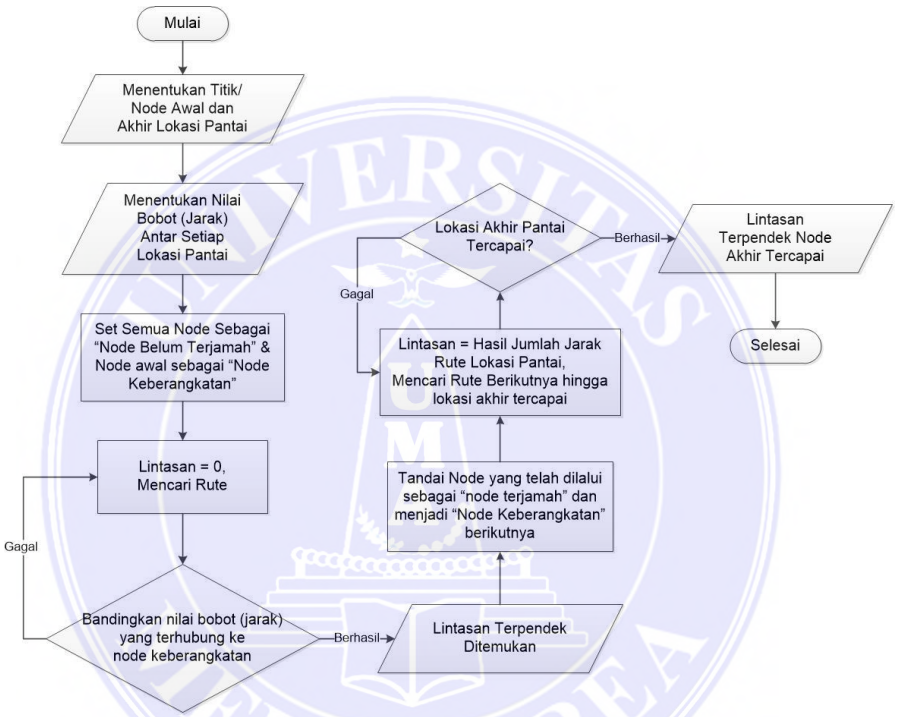
Gambar 3.2 Graf Rute Perjalanan Menuju Wisata Pantai Nias

Gambar 3.2 merupakan gambaran rute perjalanan yang akan dilewati oleh wisatawan menuju destinasi pantai pulau Nias, dimana lokasi keberangkatannya adalah *node* O atau kota Gunungsitoli. Untuk bisa mencapai lokasi pantai, para wisatawan harus melawati rute yang ada. Terdapat beberapa rute untuk mencapai

lokasi pantai-pantai ini, dari beberapa rute tersebut memiliki jarak tempuh yang berbeda-beda.

3.4 Tahapan Penelitian Menentukan Rute Dengan Algoritma *Dijkstra*

Tahapan penelitian untuk menentukan rute terpendek ke wisata pantai Pulau Nias dengan Algoritma *Dijkstra* dapat dilihat pada gambar 3.3.



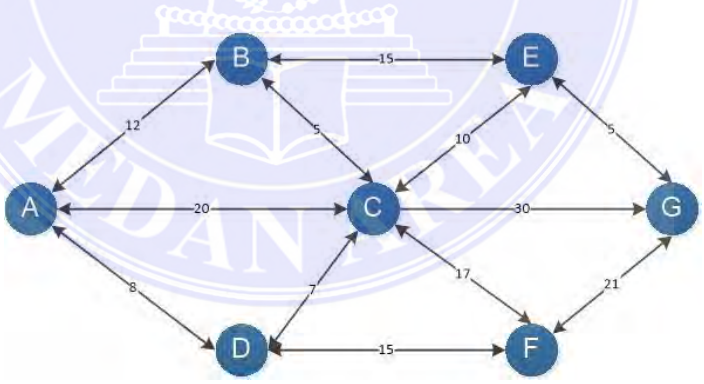
Gambar 3.3 Flowchart Tahapan Penelitian Menentukan Rute Terpendek

Dalam menentukan rute terpendek menggunakan Algoritma *Dijkstra*, terdapat beberapa tahapan yang harus dilakukan agar rute terpendek yang diinginkan bisa tercapai dengan baik, antara lain:

1. Menentukan titik atau *node* awal dan *node* akhir lokasi pantai,
2. Menentukan nilai bobot atau jarak antar setiap lokasi pantai yang ada,
3. Lalu, set semua *node* sebagai "*node* belum terjamah" dan *node* awal sebagai "*node* keberangkatan",

4. Nilai lintasan = 0, mencari rute terpendek pertama dengan Algoritma *Dijkstra* yaitu membandingkan nilai bobot (jarak) yang terhubung ke *node* keberangkatan,
5. Jika berhasil maka lintasan terpendek ditemukan dan jika gagal maka akan mengulangi langkah untuk mencari rute,
6. *Node* yang telah dilalui diset menjadi "*node* terjamah" dan menjadi "*node* keberangkatan" berikutnya, nilai jarak lintasan sama dengan hasil dari jumlah jarak lintasan yang telah dilewati,
7. Mencari rute berikutnya hingga menuju *node* akhir lokasi pantai, jika berhasil maka lintasan terpendek *node* akhir tercapai dan proses pencarian rute dinyatakan selesai.

Contoh perhitungan manual pencarian rute terpendek menggunakan metode Algoritma *Dijkstra* adalah sebagai berikut:



Gambar 3.4 Contoh Perhitungan Manual Pencarian Rute Terpendek

1. Mulai *Node* A, Akhiri *Node* G. Setiap tepi yang menghubungkan *node* diberi nilai,
2. Berdasarkan *node* awal, *Dijkstra* menghitung *node* yang berdekatan dan hasilnya adalah *node* D karena nilai bobot terkecil dari hasil perhitungan dan nilainya adalah 8 (0 + 8),

3. *Node* D ditunjuk sebagai *node* sumber dan terjamah. Dari semua *node* yang tidak berpasangan dan terhubung langsung ke *node* D adalah *node* C dan *node* F. *Dijkstra* menghitung dan hasilnya adalah *node* C. Hal ini karena bobot minimum hasil perhitungan adalah 15. (8+7),
4. *Node* C ditunjuk sebagai *node* sumber dan terjamah. *Node* B, E, F, dan F adalah *node* yang berdekatan terhubung langsung ke *node* C. *Dijkstra* menghitung dan hasilnya adalah *node* B. Nilainya adalah 20 (15 + 5),
5. *Node* B ditunjuk sebagai *node* sumber dan ditandai sebagai *node* yang terpengaruh. *Dijkstra* menghitung ulang dan menemukan *node* E. Hal ini dikarenakan *node* B tidak terhubung secara otomatis dengan *node* lain. *Node* E terhubung langsung ke *node* B, yang memiliki nilai 35 (20 + 15),
6. *Node* E menjadi *node* terjamah dan sebagai *node* sumber, Dijkstra menghitung ulang dan mencapai *Node* G (*destination node*) melalui *node* E. Bobot yang dihasilkan adalah 40 (35 + 5).
7. Maka rute terpendek yang bisa dilewati adalah A – D – C – B – E – G dengan total nilai bobot atau nilai lintasan 40.

3.5 Pengolahan Data Rute Perjalanan Secara Manual

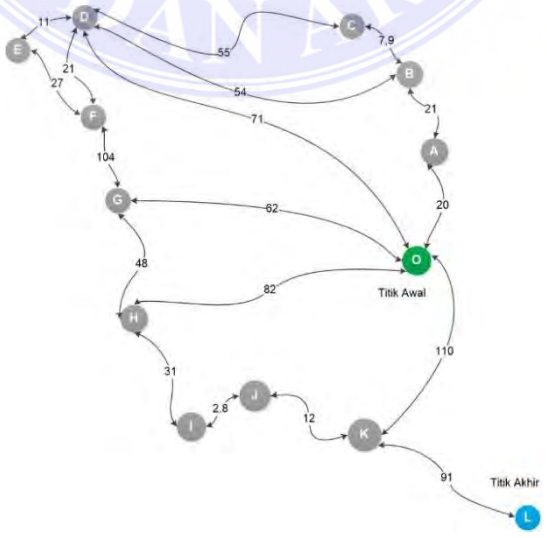
Pengolahan data dilakukan dengan memodelkan rute yang mencakup titik awal dan titik akhir yang akan dilewati untuk menemukan jalur terpendek. *Node* O dipilih untuk titik awal dan *node* L untuk titik akhir. *Node* mendefinisikan tujuan, garis mendefinisikan jalur, dan algoritma *Dijkstra* menghitung bobot minimum untuk setiap *node*.

Menurut (Ginting & Barus, 2018) Tujuan dari metode *Dijkstra* yaitu mencari jalur terpendek mulai dari satu titik ke titik berikutnya dengan bobot paling kecil. Berikut adalah beberapa cara untuk mencari jalur terpendek dengan *Dijkstra*:

1. Tentukan titik referensi dari setiap sudut yang tersedia,
2. Pilih titik awal dan,
3. Tandai posisi akhir,
4. Kemudian ukur jarak dari titik pertama ke titik terdekat satu per satu,
5. *Dijkstra* secara bertahap akan memperluas pencarian dari satu titik ke titik berikutnya.

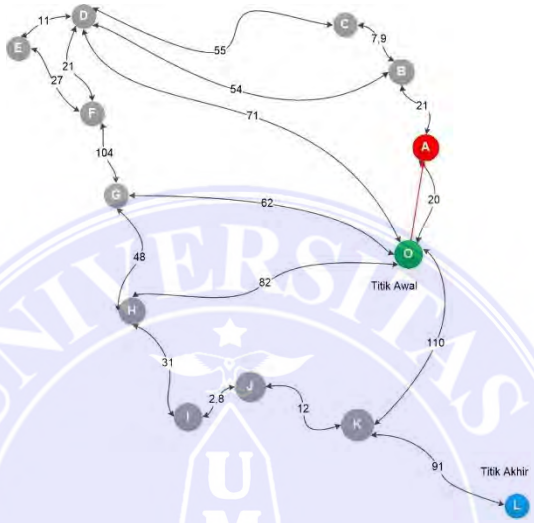
Untuk menemukan rute terpendek dari *node* awal ke *node* tujuan dengan jarak minimum, berikut adalah penjelasan langkah demi langkah:

1. *Node* pertama adalah O, dan *node* terakhir adalah L. Setiap tepi yang menghubungkan *node* diberi nilai,



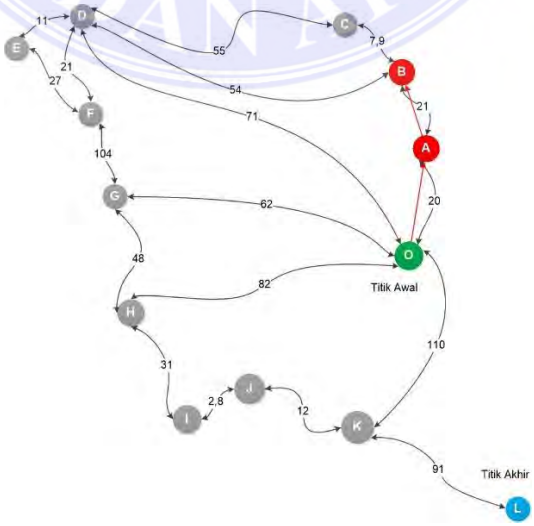
Gambar 3.5 Langkah Pertama Pengolahan Data Algoritma *Dijkstra*

2. *Dikstra* mulai menghitung bobot minimum, hasil perhitungan pada *node* yang berdekatan yang terhubung langsung dengan *node* pertama, yaitu *node* A. Bobot minimum adalah 20 (0 + 20), dan *node* D, G, H, dan K tidak akan dimasukkan dalam jalur terpendek karena bobotnya melebihi bobot *node* A,



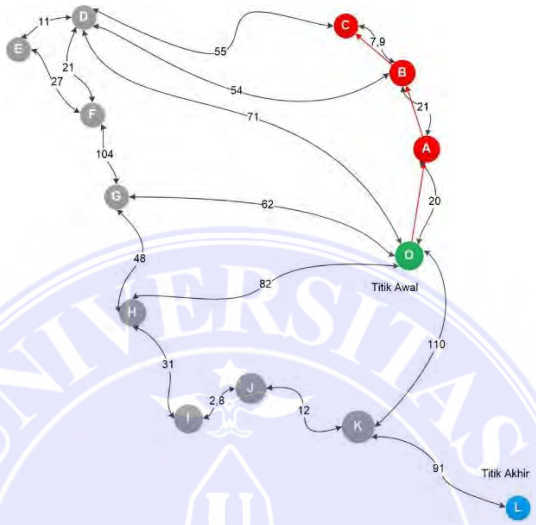
Gambar 3.6 Langkah Kedua Pengolahan Data Algoritma *Dijkstra*

3. *Node* A ditunjuk sebagai node sumber dan terjamah. *Dijkstra* menghitung dan menemukan *node* B. *Node* A terhubung langsung ke *node* B, karena *node* A tidak terhubung ke *node* lain. Nilainya adalah 41 (20 + 21),



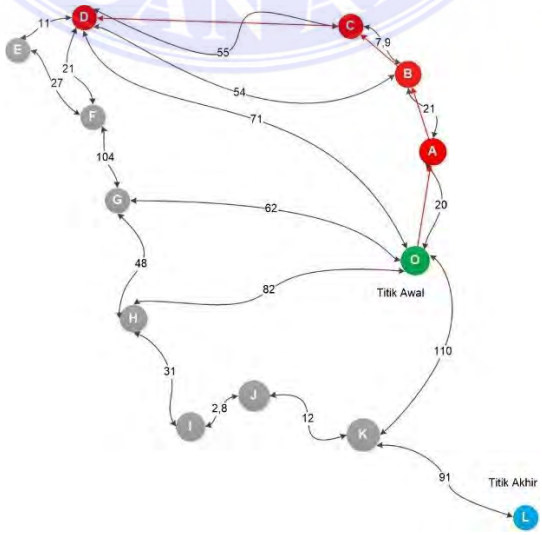
Gambar 3.7 Langkah Ketiga Pengolahan Data Algoritma *Dijkstra*

4. *Node* B ditunjuk sebagai *node* sumber dan terjamah. *Dijkstra* menghitung dan menemukan *node* C, nilai bobot minimum yang diperoleh adalah 48,9 (41 + 7,9). Sedangkan *node* D tidak termasuk rute terpendek karena nilai bobotnya lebih besar dibandingkan *node* C yaitu 95 (41+54),



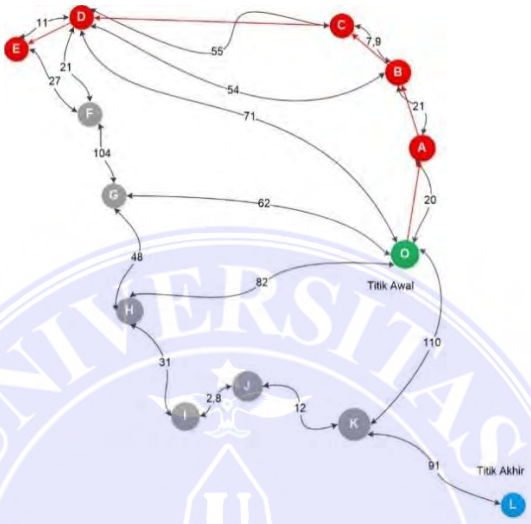
Gambar 3.8 Langkah Keempat Pengolahan Data Algoritma *Dijkstra*

5. *Node* C ditunjuk sebagai node sumber dan terjamah. Berikutnya *Dijkstra* melakukan perhitungan kembali dan menemukan *node* D. *Node* C terhubung langsung ke *Node* D karena *Node* C tidak terhubung ke *node* lainnya. Nilainya adalah 103,9 (48,9 + 55),



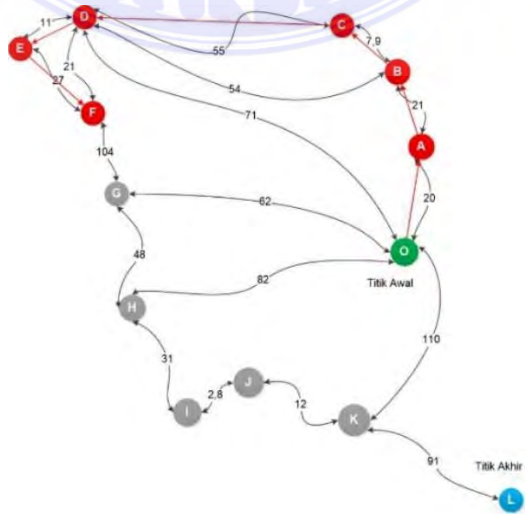
Gambar 3.9 Langkah Kelima Pengolahan Data Algoritma *Dijkstra*

6. *Node* D sebagai *node* sumber dan *node* terjamah. *Dijkstra* melakukan perhitungan dan mendapatkan *node* E. Hasil perhitungan adalah 114,9 (103,9 +11). Sedangkan *node* F tidak termasuk rute terpendek karena nilai bobotnya lebih besar dibandingkan *node* E yaitu 124,9 (103,9+21),



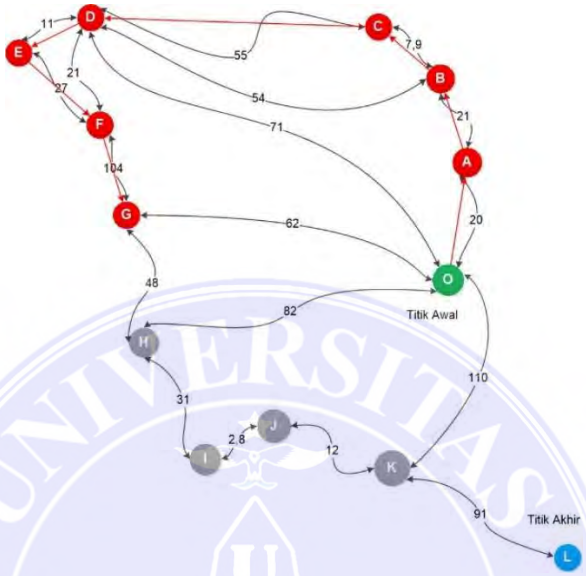
Gambar 3.10 Langkah Keenam Pengolahan Data Algoritma *Dijkstra*

7. *Node* E sebagai *node* sumber dan *node* terjamah. *Dijkstra* menghitung ulang dan menemukan *node* F. Karena *node* E tidak terhubung ke *node* lainnya maka *node* E terhubung langsung ke *node* F, nilainya adalah 141,9 (114,9 + 27),



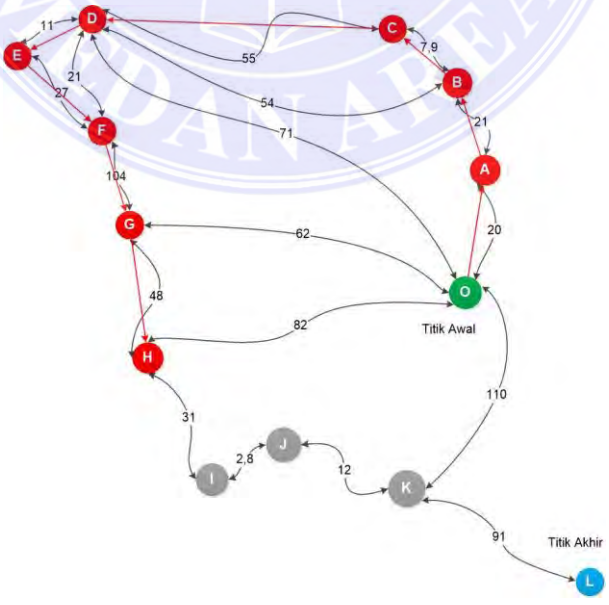
Gambar 3.11 Langkah Ketujuh Pengolahan Data Algoritma *Dijkstra*

8. *Node* F ditunjuk sebagai *node* sumber dan terjamah. *Dijkstra* menghitung ulang dan menemukan *node* G. *Node* F terhubung langsung ke *node* G dengan nilai 245,9 (141,9 + 104) karena tidak terhubung ke *node* lainnya,



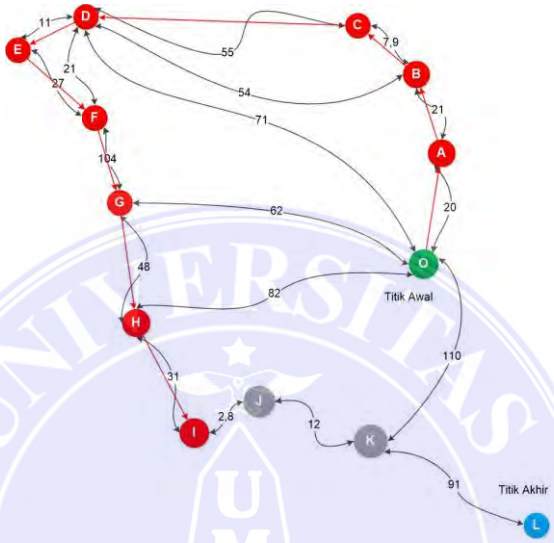
Gambar 3.12 Langkah Kedelapan Pengolahan Data Algoritma *Dijkstra*

9. *Node* G ditunjuk sebagai *node* sumber dan terjamah. *Dijkstra* menghitung ulang dan menemukan *node* H dengan nilai 293,9 (245,9 + 48),



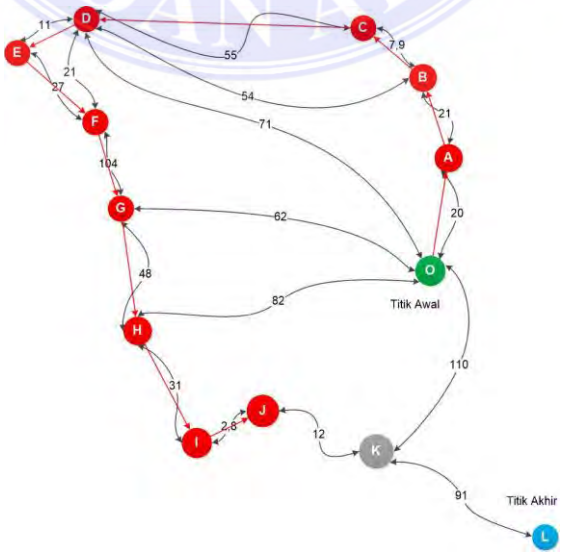
Gambar 3.13 Langkah Kesembilan Pengolahan Data Algoritma *Dijkstra*

10. *Node* H ditunjuk sebagai *node* sumber dan terjamah. *Dijkstra* menghitung ulang dan mendapatkan *node* I. Karena *node* H tidak terhubung ke *node* lainnya, secara otomatis *node* H terhubung langsung ke *node* I, nilai bobotnya 324,9 (293,9 + 31).



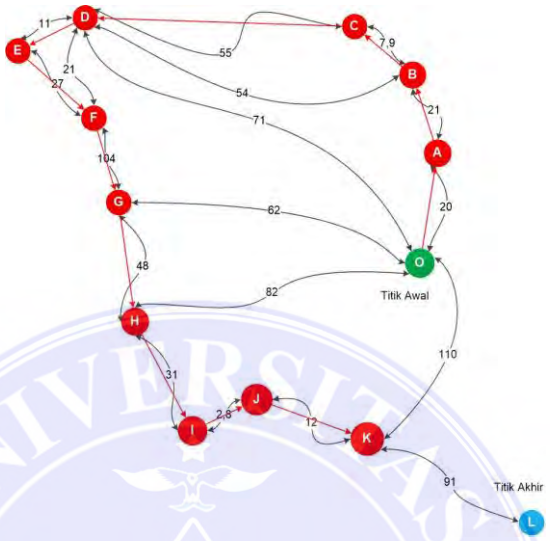
Gambar 3.14 Langkah Kesepuluh Pengolahan Data Algoritma *Dijkstra*

11. *Node* I menjadi *node* sumber dan terjamah. *Dijkstra* melakukan perhitungan lagi dan mendapatkan *node* J dengan nilai bobot 327,7 (324,9 + 2,8),



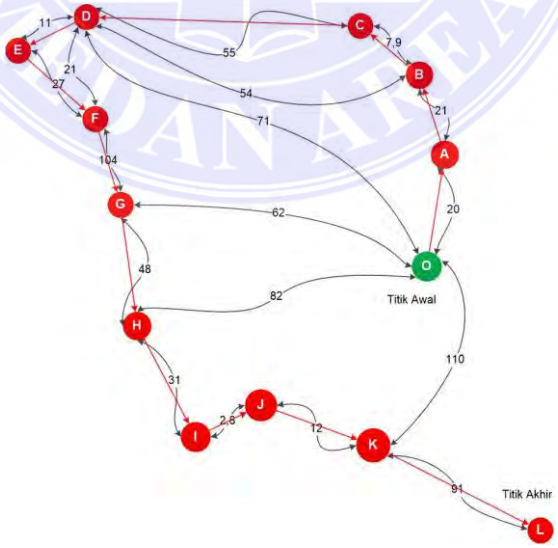
Gambar 3.15 Langkah Kesebelas Pengolahan Data Algoritma *Dijkstra*

12. *Node* J ditunjuk sebagai *node* sumber dan terjamah. *Dijkstra* menghitung ulang dan mendapatkan *node* K. *Node* J terhubung langsung dengan *node* K, sehingga nilainya adalah 339,7 (327,7+12).



Gambar 3.16 Langkah Keduabelas Pengolahan Data Algoritma *Dijkstra*

13. *Node* K menjadi *node* sumber dan terjamah. *Dijkstra* menghitung ulang dan mendapatkan *node* L (node akhir). Nilai bobot yang dihasilkan adalah 430,7 (339,7 + 91).



Gambar 3.17 Langkah Ketigabelas Pengolahan Data Algoritma *Dijkstra*

Hasil Pengujian Rute:

Tabel 3.4 Hasil Pengujian Pencarian Rute Terpendek Secara Manual

No	*Node* awal	*Node* Tujuan	Jarak antar *node* (km)	Rute Perjalanan	Jumlah Jarak Rute (km)
1	O	A*	20	O-A	20
		D	71		
		G	62		
		H	82		
2	A	B*	21	O-A-B	41
3	B	C*	7,9	O-A-B-C	48,9
		D	54		
4	C	D*	55	O-A-B-C-D	103,9
5	D	E*	11	O-A-B-C-D-E	114,9
		F	21		
6	E	F*	27	O-A-B-C-D-E-F	141,9
7	F	G*	104	O-A-B-C-D-E-F-G	245,9
8	G	H*	48	O-A-B-C-D-E-F-G-H	293,9
9	H	I*	31	O-A-B-C-D-E-F-G-H-I	324,9
10	I	J*	2,8	O-A-B-C-D-E-F-G-H-I-J	327,7
11	J	K*	12	O-A-B-C-D-E-F-G-H-I-J-K	339,7

12	K	L*	91	O-A-B-C-D-E-F-G-H-I-J-K-L	430,7

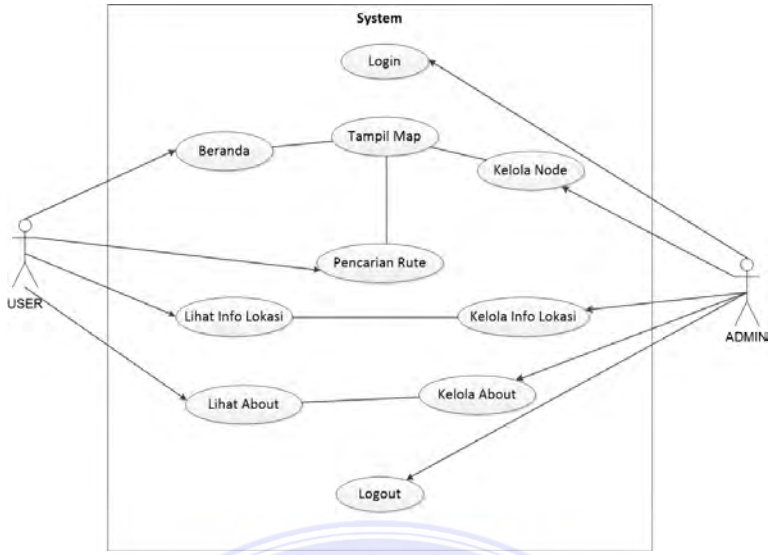
Bila *node* akhir telah diperoleh maka pencarian jalur menggunakan *Dijkstra* dinyatakan berakhir. Dari hasil pengujian yang dilakukan menuju lokasi tujuan yaitu destinasi wisata pantai Pulau Tello Nias Selatan, dengan titik awal keberangkatan di Kota Gunungsitoli didapatkan bahwa jalur alternatif berupa rute terpendek yang bisa dilalui oleh para wisatawan untuk mengunjungi 12 pantai yang ada di pulau Nias adalah Pantai Hoya Gunungsitoli (A)-Pantai Indah Fofola (B)-Pantai Saiti Nias Utara (C)-Pantai Indah Tureloto Lahewa (D)-Pantai Toyolawa Nias Utara (E)-Pantai Merah, Afulu (F)-Pantai Sirombu Nias Barat (G)-Pantai Ladeha Nias Selatan (H)-Pantai Sorake (I)-Pantai Lagundri (J)-Pantai Blessing (K)-Pantai Pulau Tello (L) dengan jarak lintasan 20-21-7,9-55-11-27-104-48-31-2,8-12-91 dengan total nilai bobotnya 430,7 atau panjang lintasan adalah 430,7 km.

3.6 Perancangan Sistem

Pada bagian ini dilakukan perancangan proses-proses yang akan terjadi di dalam sistem agar hasil yang akan berjalan sesuai dengan rancangan aplikasi yang telah dibuat.

3.6.1 *Use Case* Diagram

Use case diagram menunjukkan bagaimana sistem berinteraksi dengan lingkungannya. Dalam *use case* diagram ini, pengguna (*user*) dan administrator dapat berinteraksi dengan sistem dalam berbagai cara, dan terdapat batasan untuk memastikan bahwa interaksi yang terjadi tidak sama atau berbeda antara mereka.

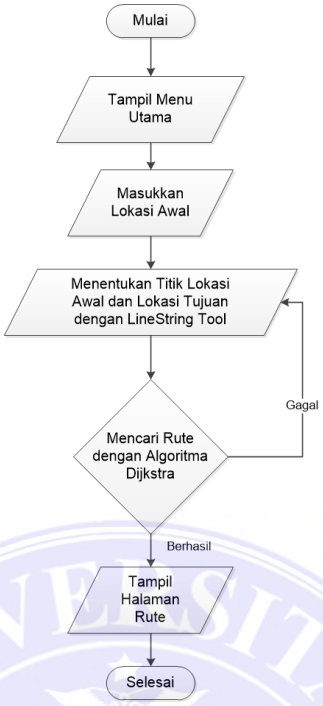


Gambar 3.18 *Use Case* Diagram

Seperti yang ditunjukkan pada Gambar 3.18 *use case* diagram, pengguna dapat berinteraksi dengan sistem dengan mengunjungi halaman menu utama (*home page*), mencari rute, melihat informasi lokasi yang diberikan oleh pengguna, dan mengakses halaman *about* yang terhubung ke *website* pencarian rute ini. Di sisi lain, administrator dapat berinteraksi ke sistem dengan masuk (*login*) ke situs *web* untuk mengelolanya, seperti mengelola *node*, informasi lokasi, kelola halaman *about*, lalu *logout* untuk mengakhiri aktivitasnya.

3.6.2 *Flowchart* Sistem Pencarian Rute

Dalam mencari rute terdekat menuju lokasi yang ingin dikunjungi, terjadi interaksi antara *user* dengan sistem dan alur proses tahapan-tahapan untuk mencari mencari rute tersebut dijabarkan pada *flowchart* berikut ini.

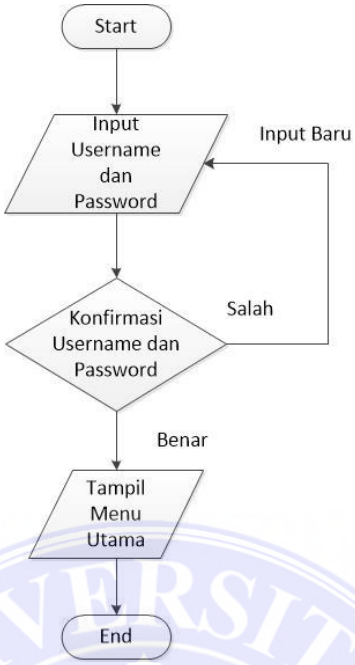


Gambar 3.19 *Flowchart* Sistem Pencarian Rute

Diawali dengan *user* masuk ke dalam website, lalu website tersebut akan menampilkan menu utamanya. Pada halaman menu utama websitenya, *user* memasukkan lokasi awal (keberangkatan) dan menentukan titik lokasi awal *user* dan lokasi tujuan dengan *LineString Tool*, kemudian system akan mencari rute dengan menggunakan Algoritma *Dijkstra*. Jika system gagal mencari rute maka *user* harus mengulangi kembali menentukan lokasi awal dan lokasi tujuannya dengan *LineString Tool*. Jika berhasil, system akan menampilkan rute yang dicari oleh *user*.

3.6.3 *Flowchart Login Admin*

Admin memiliki akses untuk mengelola seluruh isi website pencarian rute terpendek, sebelum itu terlebih dahulu seorang *admin login* ke dalam website dan alur proses tahapan-tahapan untuk *login* dijabarkan pada *flowchart* berikut ini.

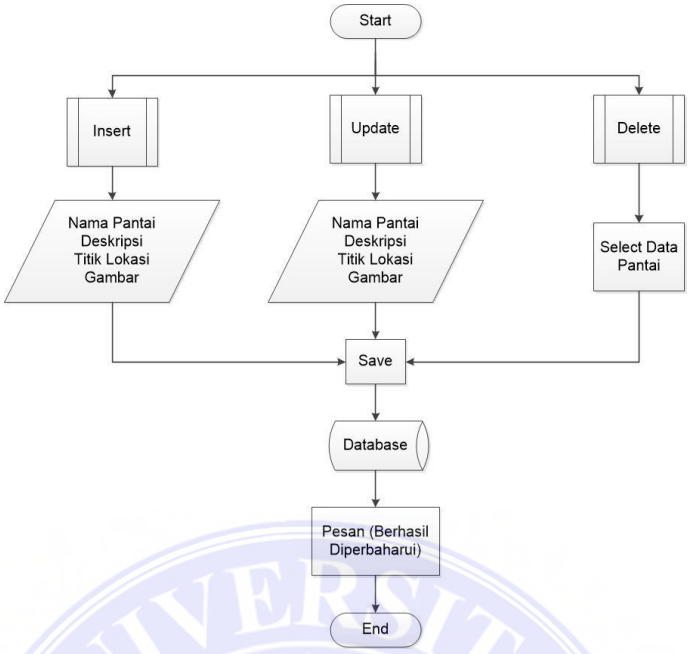


Gambar 3.20 *Flowchart Login Admin*

Untuk mengakses situs *web* pencarian rute tercepat, administrator harus memasukkan *username* dan *password* mereka, yang kemudian diproses oleh sistem. Jika *admin* memasukkan *username* atau *password* yang salah, mereka tidak akan dapat mengakses menu utama situs *web* dan harus memasukan ulang *username* dan *password* mereka. Jika *username* dan *password* yang dimasukkan benar, halaman menu utama akan muncul.

3.6.4 *Flowchart* Halaman Data Informasi Pantai

Data informasi pantai diakses oleh *admin* untuk dikelola, interaksi yang dilakukan *admin* terhadap sistem adalah melakukan *insert*, *update*, dan *delete*, dan alur proses tahapan-tahapan untuk mengelola data informasi pantai tersebut dijabarkan pada *flowchart* berikut ini.

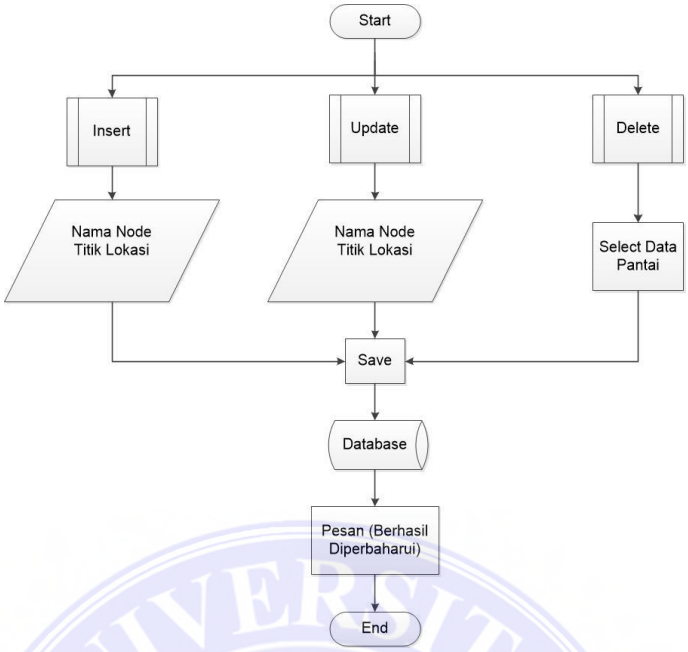


Gambar 3.21 *Flowchart* Halaman Data Informasi Pantai

Admin berada di halaman data informasi pantai untuk dikelola sesuai dengan kebutuhan yang diperlukan, pada bagian ini *admin* bisa melakukan beberapa tindakan yaitu *insert* data pantai, *update*, dan melakukan *delete* pada data yang tidak diinginkan. Diawali dengan start, setelahnya tindakan *insert* atau *update* atau *delete*, kemudian *save* agar bisa diproses oleh sistem di dalam *database* dan jika tindakan tersebut berhasil dilakukan akan muncul pesan "berhasil diperbaharui" dan selesai. Data yang dikelola oleh admin pada halaman data informasi pantai berupa nama pantai, deskripsi, titik lokasi, dan gambar.

3.6.5 *Flowchart* Halaman Data Informasi *Node*

Data informasi *node* diakses oleh *admin* untuk dikelola, interaksi yang dilakukan *admin* adalah melakukan *insert*, *update*, dan *delete*, dan alur proses tahapan-tahapan untuk mengelola data informasi *node* tersebut dijabarkan pada *flowchart* berikut ini.

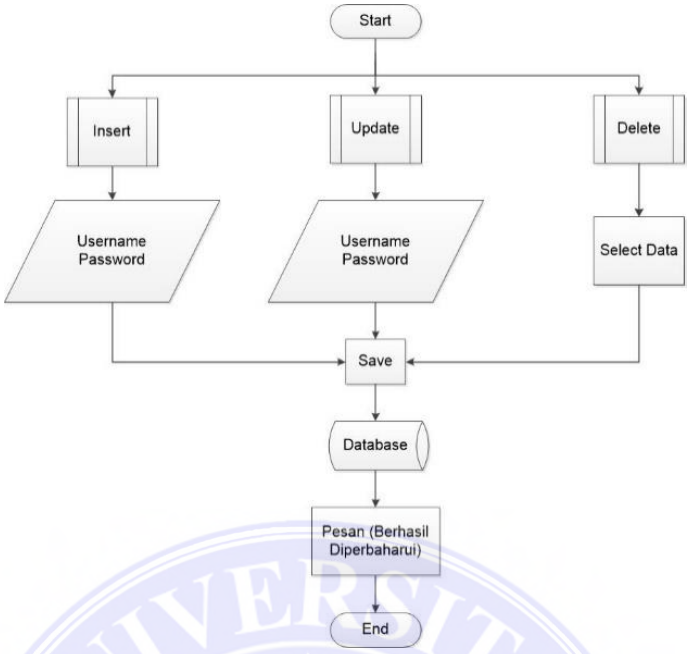


Gambar 3.22 *Flowchart* Halaman Data Informasi *Node*

Admin berada di halaman data informasi *node* untuk dikelola sesuai dengan kebutuhan yang diperlukan, pada bagian ini *admin* bisa melakukan beberapa tindakan yaitu *insert* data *node*, *update*, dan melakukan *delete* pada data *node* yang tidak diinginkan oleh *admin*. Diawali dengan *start*, setelahnya tindakan *insert* atau *update* atau *delete*, kemudian *save* agar bisa diproses oleh sistem di dalam *database* dan jika tindakan tersebut berhasil dilakukan akan muncul pesan "berhasil diperbaharui" dan selesai. Data yang dikelola oleh *admin* pada halaman data informasi *node* berupa nama *node* dan titik lokasi.

3.6.6 *Flowchart* Halaman Data Informasi *Admin*

Data informasi *admin* dapat diakses oleh *admin* itu sendiri untuk dikelola, interaksi yang dilakukan *admin* adalah melakukan *insert*, *update*, dan *delete*, dan alur proses tahapan-tahapan untuk mengelola data informasi admin baru atau admin lama tersebut dijabarkan pada *flowchart* berikut ini.



Gambar 3.23 *Flowchart* Halaman Data Informasi *Admin*

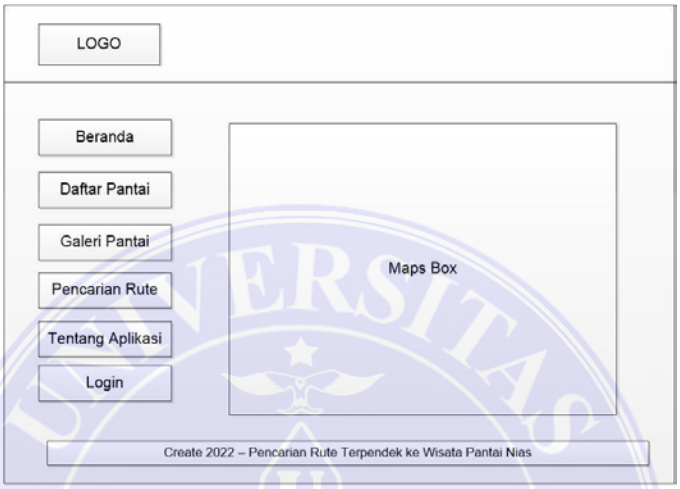
Admin berada di halaman data informasi *admin* untuk dikelola sesuai dengan kebutuhan, pada bagian ini *admin* bisa melakukan tindakan yaitu *insert* data *admin* baru, *update* data *admin* lama, dan melakukan *delete* pada data yang tidak diinginkan. Diawali dengan *start*, setelahnya tindakan *insert* atau *update* atau *delete*, kemudian *save* agar bisa diproses oleh sistem di dalam *database* dan jika tindakan tersebut berhasil dilakukan akan muncul pesan "berhasil diperbaharui" dan selesai. Data yang dikelola oleh *admin* pada halaman data informasi *admin* berupa *username* dan *password*.

3.7 Perancangan *User Interface*

Untuk memungkinkan seorang pengguna (*user*) atau *admin* menggunakan sistem yang sedang dibuat, perancangan antarmuka pengguna (*user interface*) dilakukan untuk membuat tampilan bagian dalam perangkat lunak yang berfokus pada tampilan atau gaya.

3.7.1 Tampilan Menu Utama

Gambaran tampilan menu utama *website* pencarian rute terpendek ke wisata pantai yang ada di Pulau Nias. Pada menu utama *website* terdapat *header*, *sidebar*, *content*, dan *footer*.

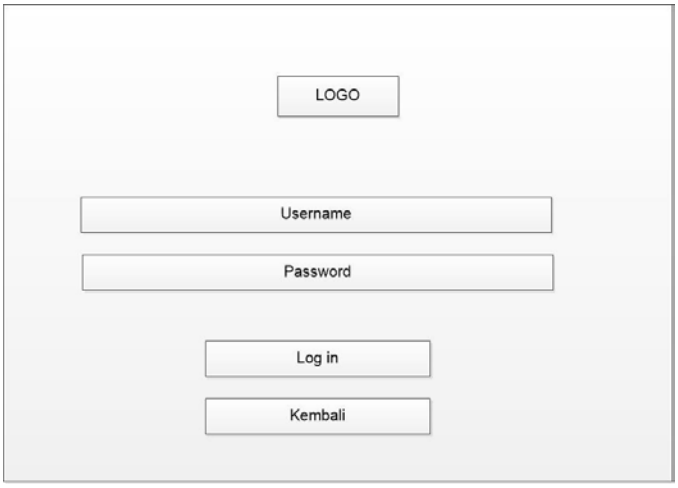


Gambar 3.24 Tampilan Menu Utama

Seperti yang terlihat pada gambar 3.24 tampilan menu utama, terdapat logo *website*, berisikan tombol *button* sub-sub menu seperti beranda, daftar pantai untuk melihat pantai-pantai yang terdaftar di dalam *website*, galeri pantai untuk melihat bentuk atau gambaran pantai yang terdaftar, *button* pencarian rute, tentang aplikasi, *login* untuk *admin*, dan terdapat maps box untuk mencari rute yang diinginkan. Selain itu, terdapat tulisan hak cipta "Pencarian Rute Terpendek ke Wisata Pantai Nias" pada bagian bawah *website*.

3.7.2 Tampilan *Form Login Admin*

Gambaran tampilan *form login admin website* pencarian rute terpendek ke wisata pantai yang ada di Pulau Nias. Pada bagian *form login admin website* pencarian rute terpendek terdapat logo *website*, terdapat kolom *input username* dan *password* admin, *button log in*, dan kembali.

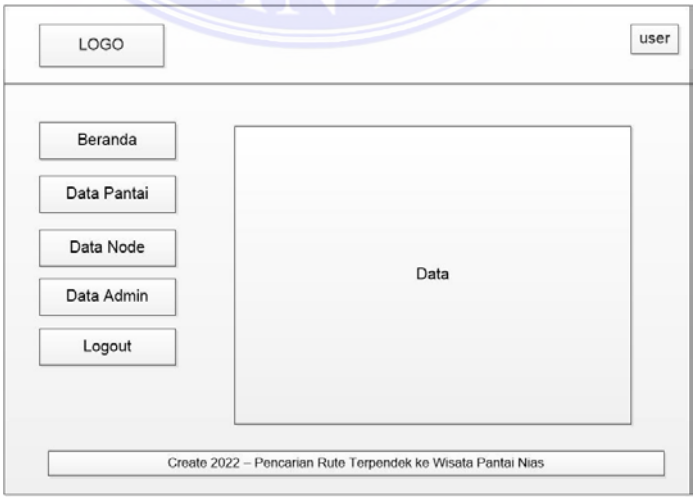


Gambar 3.25 Tampilan *Form Login Admin*

Seperti yang ditunjukkan pada Gambar 3.25 tampilan *form login admin*, seorang administrator harus memasukkan *username* dan *password* untuk mengakses halaman menu utama administrator. Kemudian, Anda dapat kembali ke menu utama dengan mengklik *button* "kembali".

3.7.3 Tampilan Menu Utama *Admin*

Gambaran tampilan menu utama *admin website* pencarian rute terpendek ke wisata pantai yang ada di Pulau Nias. Posisi *header*, *sidebar*, *content*, dan *footer* pada menu utama *admin* tidak berbeda jauh dengan menu utama *user*.

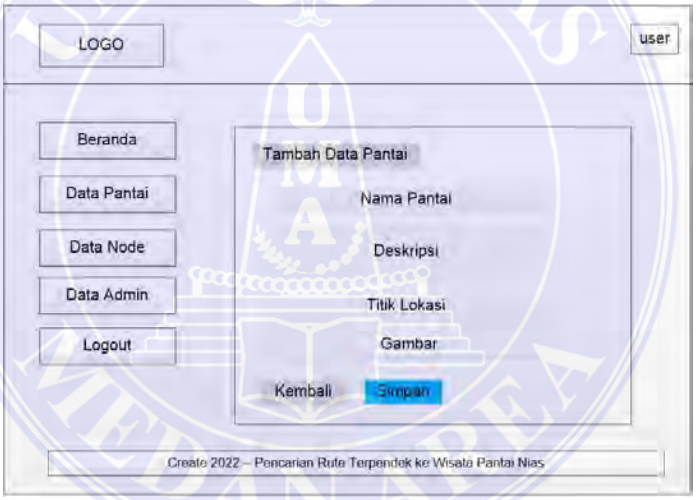


Gambar 3.26 Tampilan Menu Utama *Admin*

Seperti yang terlihat pada gambar 3.26 tampilan menu utama *admin*, terdapat logo *website* dan nama *admin* yang sedang *login*, berisikan tombol *button* sub-sub menu seperti beranda, data pantai, data *node*, dan data *admin*, selain itu terdapat juga tombol *logout* untuk keluar dari halaman menu utama *admin*.

3.7.4 Tampilan *Form* Tambah Data Pantai

Gambaran tampilan *form* tambah data pantai *website* pencarian rute terpendek ke wisata pantai yang ada di Pulau Nias. Pada halaman *form* tambah data pantai *website* pencarian rute terpendek, seorang *admin* bisa menambahkan data pantai baru.



Gambar 3.27 Tampilan *Form* Tambah Data Pantai

Seperti yang terlihat pada gambar 3.27 tampilan *form* tambah data pantai, jika seorang *admin* ingin menambahkan data pantai baru harus memasukkan data berupa nama pantai, deskripsi, titik lokasi, dan gambar. Setelah itu, klik simpan agar bisa diproses oleh sistem dan masuk ke *database*.

3.7.5 Tampilan *Form* Tambah Data *Node*

Gambaran tampilan *form* tambah data *node website* pencarian rute terpendek ke wisata pantai yang ada di Pulau Nias. Pada halaman *form* tambah data *node website* pencarian rute terpendek, seorang *admin* bisa menambahkan data *node* baru.

Gambar 3.28 Tampilan *Form* Tambah Data *Node*

Seperti yang terlihat pada gambar 3.28 tampilan *form* tambah data *node*, jika seorang *admin* ingin menambahkan data *node* baru harus memasukkan data berupa nama *node* dan titik lokasi. Setelah itu, klik simpan agar bisa diproses oleh sistem dan masuk ke *database*.

3.7.6 Tampilan *Form* Tambah Data *Admin*

Bentuk desain *form* penambahan *admin* baru ke dalam sistem pada *website* pencarian rute terpendek menuju wisata pantai Pulau Nias. Pada halaman *form* tambah data *admin website* pencarian rute terpendek, seorang *admin* bisa menambahkan data *admin* baru untuk mengelola *website*.

Gambar 3.29 Tampilan *Form* Tambah Data *Admin*

Seperti yang terlihat pada gambar 3.29 tampilan *form* tambah data *admin*, jika seorang *admin* ingin menambahkan data *admin* baru harus memasukkan data berupa *username* dan *password*. Setelah itu, klik simpan agar bisa diproses oleh sistem dan masuk ke *database*.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan perumusan masalah, hasil uji coba, dan pembahasan yang telah penulis buat, maka dapat disimpulkan:

1. Jalur terpendek yang dapat dilalui untuk mengunjungi 12 titik lokasi pantai yang telah diteliti dengan titik awal lokasi kota gunungsitoli dan tujuan akhir adalah pantai pulau tello yaitu O - A - B - C - D - E - F - G - H - I - J - K - L dengan jarak lintasan 20 - 21 - 7,9 – 55 – 11 – 27 – 104 – 48 – 31 - 2,8 – 12 - 91 dengan total nilai bobotnya 430,7 atau panjang lintasan ialah 430,7 km.
2. Dari 12 titik lokasi pantai yang telah diuji dengan titik lokasi awal yaitu kota gunungsitoli menunjukkan bahwa rute berhasil ditampilkan dan mendapatkan rute ke 7 titik lokasi tujuan dengan jarak rute dan waktu tempuh yang berbeda-beda, terdapat 4 titik lokasi tujuan yang berhasil ditampilkan juga tapi rute yang didapatkan terlalu jauh dan 1 titik lokasi tujuan yang tidak berhasil ditampilkan karena jarak rute dan waktu tempuhnya tidak terdeteksi. Jarak rute dan waktu tempuh terbaik yang bisa dilalui oleh wisatawan menuju destinasi wisata pantai pulau Nias yaitu rute dari lokasi awal gunungsitoli menuju lokasi tujuan Pantai Hoya Gunungsitoli dengan jarak rute 35.84 km dan waktu tempuh 149.89 menit.

5.2 Saran

Adapun saran untuk pengembangan penelitian selanjutnya agar lebih sempurna sebuah sistem yang telah dibangun adalah sebagai berikut:

1. Pada penelitian selanjutnya bisa mengganti objek yang diteliti selain daerah wisata pantai, bisa dengan museum atau hotel pada suatu daerah
2. Penelitian selanjutnya bisa menggunakan metode pencarian rute lain seperti *A-Star* dan *Bellman-Ford* untuk melihat perbandingan akurasi pencarian
3. Penelitian selanjutnya dapat menggunakan titik lokasi pantai yang lain yang jarak rute dan waktu tempuhnya bisa terdeteksi oleh system yang menggunakan metode Algoritma *Dijkstra*.

DAFTAR PUSTAKA

Ahdan, S., & Setiawansyah. (2020). *Pengembangan Sistem Informasi Geografis Untuk Pendonor Darah dengan Algoritma Dijkstra berbasis Android*. Jurnal Sains Dan Informatika (Research of Science and Informatic)*, 6 (12)*, 67-77.

Atmojo, W. T., & Susiana. (2019). *Implementasi Algoritma Dijkstra Untuk Mencari Rute Tujuan Wisata Terpendek Dalam Aplikasi Berbasis Web*. Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK), 15-21.

Cantona, A., Fauziah, & Winarsih. (2020). *Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta*. Jurnal Teknologi dan Manajemen Informatika, 27-34.

Deepa, G., Kumar, P., Manimaran, A., Rajakumar, K., & Krishnamoorthy, V. (2018). *Dijkstra Algorithm Application: Shortest Distance between Buildings*. International Journal of Engineering & Technology, 974-976.

Folaimam, B., Rosihan, & Mubarak, A. (2018). *Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas*. JIKO (Jurnal Informatika dan Komputer)*, 2 (2)*, 61-69.

Ginting, J. V., & Barus, E. S. (2018). *Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra*. Jurnal Mantik Penusa, 1-8.

Grace, D., Tanciga, M. S., & Nurdin. (2018). *Sistem Informasi Letak Geografis Penentuan Jalur Tercepat Rumah Sakit Di Kota Palu Menggunakan Algoritma Greedy Berbasis Web*. Jurnal Elektronik (STMIK) BINA MULIA*, 4 (2)*, 59-76.

Hamdi, S., & Prihandoko. (2018). *Analisis Algoritma Dijktra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota)*. Jurnal Ilmiah Ilmu-Ilmu Teknik, 26-32.

Lahagu, E. (2018). *Strategi Pengelolaan Obyek Wisata Pantai Sorake Sebagai Wisata Alam Bawomataluo Di Nias Selatan Provinsi Sumatera Utara*. Yogyakarta: Sekolah Tinggi Pariwisata Ambarrukmo (STIPRAM).

Masri, M., Kiswanto, A. P., & Kusuma, B. S. (2019). *Implementasi Algoritma Dijkstra Dalam Perancangan Aplikasi Penentuan Rute Terpendek Pada Objek Pariwisata Danau Toba Dan Sekitarnya*. SEMNASTEK UISU, 221-225.

Muharrom, M. (2020). *Implementasi Algoritma Dijkstra Dalam Penentuan Jalur Terpendek Studi Kasus Jarak Tempat Kuliah Terdekat*. Indonesia Journal Of Business Intelligence (IJUBI), 25-30.

Nugraha, W., & Syarif, M. (2018). *Penerapan Metode Prototype Dalam Perancangan Sistem Informasi Penghitungan Volume dan Cost Penjualan*

Minuman Berbasis Website. Jurnal Sistem Informasi Musirawas (JUSIM), 94-101.

Ompusunggu, V. M. (2022). *Kontribusi Wisata Pantai Lagundri Dan Pantai Sorake Dalam Meningkatkan Ekonomi Keluarga Di Kecamatan TelukDalam, Nias Selatan, Sumatera Utara*. Pendidikan, Saintek, Sosial dan Hukum (PSSH), 1-13.

Paunsyah, H., Mubarok, H., & Shofa, R. N. (2019). *Penentuan Jalur Terpendek Menggunakan Google Maps API pada Sistem Informasi Geografis (GIS) Panti Sosial di Kota Tasikmalaya*. Innovation in Research of Informatics (INNOVATICS), 1-6.

Putra, R. R. (2018). *Penerapan Web Promosi Pada Bagan Deli Medan Belawan Menggunakan Pemrograman PHP Database MYSQL*. Jurnal Teknik Dan Informatika, 45-48.

Putri, T. D., Sugeng, W., & Safitri, E. (2020). *Algoritma Dijkstra untuk Penentuan Jarak Tempuh Terpendek Pengantaran Katering Pabrik*. MIND (Multimedia Artificial Intelligent Networking Database) Journal, 5(2), 108-120.

Qomaruddin, M., Bismi, W., & Hariyanto, D. (2022). *Pewarnaan Graf Pada Peta Provinsi Jawa Barat Menggunakan Algoritma Welch-Powell*. Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 10 (2)*, 258-263.

Ramadhan, R. F., & Mukhaiyar, R. (2020). *Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi*. Jurnal Teknik Elektro Indonesia (JTEIN)*, 1(2)*, 129-134.

Rumondor, A. G., Sentinuwo, S. R., & Sambul, A. M. (2019). *Perancangan Jalur Terpendek Evakuasi Bencana di Kawasan Boulevard Manado Menggunakan Algoritma Dijkstra.* Jurnal Teknik Informatika*, 14 (2)*, 261-268.

Serdano, A., Zarlis, M., & Hartama, D. (2019). *Perbandingan Algoritma Dijkstra Dan Bellman-Ford Dalam Pencarian Jarak Terpendek Pada SPBU*. Seminar Nasional Sains & Teknologi Informasi (SENSASI), 259-264.

Setiawan, J., Prakoso, R. S., & Suryaningrum, K. M. (2019). *Penentuan Rute Terpendek Menuju Pusat Perbelanjaan Di Jakarta Menggunakan Algoritma Dijkstra*. Jurnal Ilmiah MATRIK*, 21 (3)*, 156-165.

Sidhu, H. S., & Krishan, G. (2022). *Research On Dijkstra's, A*, Bellman-Ford And Floyd-Warshall Path Finding Algorithms*. International Research Journal of Modernization in Engineering Technology and Science*, 4*(01), 886-893.

Yulianto, Ramadiani, & Kridalaksana, A. H. (2018). *Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal*. Jurnal Ilmiah Ilmu Komputer Informatika Mulawarman*, 13(1)*, 14-21.

LAMPIRAN

Source Code

[Style.php]

```
<meta charset="utf-8" />

<title>PRTWPN - 2022</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta content="Premium Multipurpose Admin & Dashboard Template"
name="description" />

<meta content="Themesdesign" name="author" />

<!-- App favicon -->

<link rel="shortcut icon" href="<?= base_url() ?>/assets/faviconpantai.png">

<link href="<?= base_url() ?>assets/libs/select2/css/select2.min.css"
rel="stylesheet" type="text/css" />

<!-- Bootstrap Css -->

<link href="<?= base_url() ?>assets/css/bootstrap.min.css" id="bootstrap-style"
rel="stylesheet" type="text/css" />

<!-- Icons Css -->

<link href="<?= base_url() ?>assets/css/icons.min.css" rel="stylesheet"
type="text/css" />

<!-- App Css-->

<link href="<?= base_url() ?>assets/css/app.min.css" id="app-style"
rel="stylesheet" type="text/css" />

<link href="https://api.mapbox.com/mapbox-gl-js/v2.4.1/mapbox-gl.css"
rel="stylesheet">


<link rel="stylesheet" href="https://unpkg.com/leaflet@1.2.0/dist/leaflet.css" />

  <script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.2/mapbox-
gl.js'></script>

    <link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.2/mapbox-gl.css'
rel='stylesheet' />

    <script src='https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-
geocoder/v2.3.0/mapbox-gl-geocoder.min.js'></script>

    <link rel='stylesheet' href='https://api.mapbox.com/mapbox-gl-
js/plugins/mapbox-gl-geocoder/v2.3.0/mapbox-gl-geocoder.css' type='text/css' />
```

```
    <script src='https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-draw/v1.0.0/mapbox-gl-draw.js'></script>
    <link rel='stylesheet' href='https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-draw/v1.0.0/mapbox-gl-draw.css' type='text/css'/>
  <!-- <link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.1/dist/leaflet.css" />
    <link rel="stylesheet" href="assets/leaflet-routing-machine/dist/leaflet-routing-machine.css" /> -->


<style>
        h3.logo-sm {
                position: relative;
                bottom: -9px;
                color: white;
                font-size: 12px;
        }
        h3.logo-lg {
                position: relative;
                bottom: -9px;
                color: white; }
</style>
<script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.49.0/mapbox-gl.js'></script>
<link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.49.0/mapbox-gl.css' rel='stylesheet' />
<style type="text/css">
.marker { display: block; border: none; border-radius: 50%; cursor: pointer; padding: 0; background-position: top; }
.mapboxgl-popup-content {
/* background-color: #030303b5; */
color: white;
padding: 0; }
</style>
```

[header.php]

```
<header id="page-topbar">
<div class="navbar-header">
<div class="d-flex">
<!-- LOGO -->
<div class="navbar-brand-box">
<a href="<?= site_url() ?>" class="logo logo-dark">
        <span class="logo-lg">
            <img src="<?= base_url() ?>assets/logopantai.png" alt=""
        height="90">  </span>
        <span class="logo-lg">
            <img src="<?= base_url() ?>assets/prtwpn.jpg" alt=""
        height="28">  </span>
</a>
<a href="<?= site_url() ?>" class="logo logo-light">
        <span class="logo-sm">
            <img src="<?= base_url() ?>assets/logopantai.png" alt=""
        height="70"> </span>
        <span class="logo-sm">
            <img src="<?= base_url() ?>assets/logopantai.png" alt=""
        height="70"> </span>
</a> </div>
<button type="button" class="btn btn-sm px-3 font-size-24 header-item waves-
effect" id="vertical-menu-btn">
<i class="mdi mdi-menu"></i> </button> </div>
<!-- Search input -->
<div class="search-wrap" id="search-wrap"> <div class="search-bar">
<input class="search-input form-control" placeholder="Search" />
<a href="#" class="close-search toggle-search" data-target="#search-wrap">
        <i class="mdi mdi-close-circle"></i>
</a> </div> </div> <div class="d-flex">
<div class="dropdown d-none d-lg-inline-block">
```

```
<button type="button" class="btn header-item toggle-search noti-icon waves-effect" data-target="#search-wrap">
        <i class="mdi mdi-magnify"></i>
</button> </div>
<?php if ($this->session->has_userdata('user')) { ?>
<div class="dropdown d-inline-block">
<button type="button" class="btn header-item waves-effect" id="page-header-user-dropdown" data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        <i class="mdi mdi-account-circle-outline font-size-20 align-middle me-1"></i>
        <span class="d-none d-xl-inline-block ms-1"><?= $this->session->userdata('user')->username ?></span>
        <i class="mdi mdi-chevron-down d-none d-xl-inline-block"></i>
</button>
<div class="dropdown-menu dropdown-menu-end">
<a class="dropdown-item d-block" href="#"><span class="badge badge-success float-end">11</span><i class="mdi mdi-cog-outline font-size-16 align-middle me-1"></i> Settings</a>
<div class="dropdown-divider"></div>
<a class="dropdown-item text-danger" href="<?= site_url('logout') ?>" onclick="return confirm('Apakah kamu yakin ingin keluar?');"><i class="mdi mdi-power font-size-16 align-middle me-1 text-danger"></i> Logout</a>
</div> </div> <?php } ?> </div> </div>
</header>
```

[sidebar.php]

```
<!-- ========== Left Sidebar Start ========== -->
<div class="vertical-menu">
<div data-simplebar class="h-100">
<!--- Sidemenu --> <div id="sidebar-menu">
<!-- Left Menu Start -->
<ul class="metismenu list-unstyled" id="side-menu">
        <li class="menu-title">Menu</li>
```

```
<?php if ($this->session->has_userdata('user')) { ?>
<li> <a href="<?= site_url('admin/dashboard') ?>" class="waves-effect">
<i class="dripicons-view-thumb"></i> <span>Beranda</span>
</a> </li> <li>
<a href="<?= site_url('admin/pantai') ?>" class="waves-effect">
        <i class="mdi mdi-map-marker-radius-outline mdi-18px"></i>
        <span>Data Pantai</span>
</a> </li> <li>
        <a href="<?= site_url('admin/node') ?>" class="waves-effect">
                <i class="mdi mdi-google-maps mdi-18px"></i>
                <span>Data Node</span> </a>
</li>
<!--<li>
<a href="<?= site_url('admin/graph') ?>" class="waves-effect">
        <i class="mdi mdi-map-marker-distance mdi-18px"></i>
        <span>Crud Graph</span>
</a> </li> -->
<li>
        <a href="<?= site_url('admin/user') ?>" class="waves-effect">
        <i class="mdi mdi-account-group-outline mdi-18px"></i>
        <span>Data Admin</span> </a>
</li> <li>
        <a href="<?= site_url('logout') ?>" class=" waves-effect" onclick="return
        confirm('Apakah kamu yakin ingin keluar?');">
        <i class="dripicons-exit"></i>
        <span>Logout</span> </a>
</li> <?php } else { ?>
<li>
        <a href="<?= site_url('dashboard') ?>" class="waves-effect">
        <i class="dripicons-view-thumb"></i>
        <span>Beranda</span> </a>
```

```
</li> <li>
        <a href="<?= site_url('pantai') ?>" class="waves-effect">
                <i class="mdi mdi-map-marker-radius-outline mdi-18px"></i>
                <span>Daftar Pantai</span> </a>
</li> <li>
        <a href="<?= site_url('galeri') ?>" class="waves-effect">
                <i class="mdi mdi-image-multiple-outline mdi-18px"></i>
                <span>Galeri Pantai</span> </a>
</li> <li>
        <a href="<?= site_url('djikstra') ?>" class="waves-effect">
                <i class="mdi mdi-map-marker-path mdi-18px"></i>
                <span>Pencarian Rute</span> </a>
</li> <li>
        <a href="<?= site_url('about') ?>" class="waves-effect">
                <i class="mdi mdi-information-outline mdi-18px"></i>
                <span>Tentang Aplikasi</span> </a>
</li> <li>
        <a href="<?= site_url('login') ?>" class=" waves-effect">
                <i class="mdi mdi-18 mdi-login"></i>
                <span>Login Admin</span>
        </a>
</li> <?php } ?> </ul> </div>
<!-- Sidebar --> </div> </div>
<!-- Left Sidebar End -->
```

[footer.php]

```
<footer class="footer">
<div class="container-fluid"> <div class="row">
<div class="col-sm-6"> Create
        <script> document.write(new Date().getFullYear())
```

```
        </script> by © Jeprin Talunohi - Pencarian Rute Terpendek ke Wisata
        Pantai Nias
</div> <div class="col-sm-6">
<div class="text-sm-end d-none d-sm-block">
</div> </div> </div> </div>
</footer>
```

[js.php]

```
<!-- JAVASCRIPT -->
<script src="<?= base_url() ?>assets/libs/jquery/jquery.min.js"></script>
<script src="<?= base_url()
?>assets/libs/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="<?= base_url()
?>assets/libs/metismenu/metisMenu.min.js"></script>
<script src="<?= base_url() ?>assets/libs/simplebar/simplebar.min.js"></script>
<script src="<?= base_url() ?>assets/libs/node-waves/waves.min.js"></script>
<script src="<?= base_url() ?>assets/js/app.js"></script>
```

Admin Dashboard [index.php]

```
<!doctype html>
<html lang="en">
<head>
<?php $this->load->view('_parts/style') ?>
</head>
<body>
        <!-- Begin page -->
                <div id="layout-wrapper">
                <?php $this->load->view('_parts/header') ?>
                <?php $this->load->view('_parts/sidebar') ?>
                <!--
                ================================================
                ============== -->
        <!-- Start right Content here -->
```

```
<!--
==========================================================
========= -->

<div class="main-content" style="margin-top:100px;">

<div class="page-content"> <div class="container-fluid">

<div class="page-content-wrapper"> <div class="mt-3">

<h3 class=""><strong><?= $title ?></strong></h3>

</div> <div class="row"> <div class="col-lg-4"> <div class="card">

<div class="card-body"> <div class="p-4">

<div class="d-flex justify-content-between"> <div>

<h5 class="text-muted">Total</h5>

<h4><strong>Pantai</strong></h4>

</div>

<h1 class=""><span class=""><?= $countRS ?></span> <span
class="text-muted h3"></span></h1>

</div> </div> </div> </div> </div>

<div class="col-lg-4"> <div class="card"> <div class="card-body"> <div
class="p-4">

<div class="d-flex justify-content-between"> <div>

<h5 class="text-muted">Total</h5>

<h4><strong>Simpul</strong></h4>

</div>

<h1 class=""><span class=""><?= $countSimpul ?></span> <span
class="text-muted h3">Unit</span></h1>

</div> </div> </div> </div> </div>

<div class="col-lg-4"> <div class="card"> <div class="card-body">

<div class="p-4">

<div class="d-flex justify-content-between"> <div>

<h5 class="text-muted">Total</h5>

<h4><strong>Graph</strong></h4> </div>

<h1 class=""><span class=""><?= $countGraph ?></span> <span
class="text-muted h3">Unit</span></h1>
```

```
</div> </div> </div> </div> </div> </div> </div> </div> <!-- container-fluid --> </div>

<!-- End Page-content -->

<?php $this->load->view('_parts/footer') ?> </div>

<!-- end main content--> </div>

<!-- END layout-wrapper -->

<?php $this->load->view('_parts/js') ?>

</body>

</html>
```

Admin Graph [index.php]

```
<!doctype html>

<html lang="en">

<head>

<?php $this->load->view('_parts/style') ?>

<link href="<?= base_url() ?>assets/libs/datatables.net-bs4/css/dataTables.bootstrap4.min.css" rel="stylesheet" type="text/css" />

<link href="<?= base_url() ?>assets/libs/datatables.net-buttons-bs4/css/buttons.bootstrap4.min.css" rel="stylesheet" type="text/css" />

</head>

<body>

<!-- Begin page -->

<div id="layout-wrapper">

<?php $this->load->view('_parts/header') ?>

<?php $this->load->view('_parts/sidebar') ?>

<!--
============================================================
==== -->

<!-- Start right Content here -->

<!--
============================================================
==== -->

<!-- Modal -->
```

```
<div class="modal fade bs-example-modal-center" tabindex="-1" role="dialog"
aria-labelledby="mySmallModalLabel" aria-hidden="true">
<div class="modal-dialog modal-dialog-centered"> <div class="modal-content">
<div class="modal-header">
<h5 class="modal-title mt-0">Membuat Graph</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close">
</button>
</div> <div class="modal-body">
<form id="graphForm">
<input type="hidden" id="id_graph" name="id_graph">
<div class="row"> <div class="col-12 col-sm-6">
<label>Mulai</label>
<select id="start" name="start" class="form-control">
<option value="">Pilih Mulai</option>
<?php foreach ($nodeResult as $n) { ?>
<option lng="<?= $n->lat ?>" lat="<?= $n->lng ?>" value="<?= $n->id ?>"><?=
$n->name ?></option>
<?php } ?>
</select>
</div> <div class="col-12 col-sm-6">
<label>Tujuan</label>
<select id="end" name="end" class="form-control">
<option value="">Pilih tujuan</option>
<?php foreach ($nodeResult as $n) { ?>
<option lng="<?= $n->lat ?>" lat="<?= $n->lng ?>" value="<?= $n->id ?>"><?=
$n->name ?></option>
<?php } ?>
</select>
</div> </div> <div class="form-group mt-2">
<label>Jarak</label>
<input type="text" class="form-control" name="distance" id="distance"
readonly>
```

```
<small>Jarak ditampilkan dalam kilometer</small>
</div>
<!-- <div class="form-group mt-2">
        <label>Waktu tempuh</label>
        <input type="text" class="form-control" name="time" id="time">
        <small>Waktu tempuh dalam menit</small>
</div> -->
</form> </div>
<div class="modal-footer">
<button type="button" id="btnSave" class="btn btn-primary">Save
changes</button> </div>
</div><!-- /.modal-content --> </div><!-- /.modal-dialog -->
</div><!-- /.modal --> <div class="main-content" style="margin-top:100px;">
<div class="page-content"> <div class="container-fluid">
<div class="page-content-wrapper"> <div class="mt-3">
<h3 class=""><strong>Graph Jalur</strong></h3> </div>
<div class="row"> <div class="col-lg-6"> <div class="card">
<div class="card-body">
        <button type="button" class="btn btn-primary waves-effect waves-light"
        data-bs-toggle="modal" data-bs-target=".bs-example-modal-
        center">Tambah Graph</button>
<div class="p-0 table-responsive"><br>
<p>Berikut adalah data graph yang terdaftar.</p>
<?= $this->session->flashdata('statusMessage') ?>
<hr />
<table id="datatable" class="table table-sm table-bordered dt-responsive nowrap"
style="border-collapse: collapse; border-spacing: 0; width: 100%;">
<thead>
<tr>
<th>Mulai</th>
<th>Tujuan</th>
<th>Jarak</th>
```

```
<th>waktu</th>
<th>aksi</th>
</tr>
</thead>
<tbody></tbody>
</table>
</div> </div> </div> </div>
<div class="col-lg-6"> <div class="card"> <div class="card-body">
<div id="map" style="height: 450px;width: 100%;"></div>
</div> </div> </div> </div> </div>
</div> <!-- container-fluid --> </div>
<!-- End Page-content -->
<?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content--> </div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
<script src="<?= base_url()
?>assets/libs/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="<?= base_url() ?>assets/libs/datatables.net-
bs4/js/dataTables.bootstrap4.min.js"></script>
<script src='https://unpkg.com/@turf/turf@6/turf.min.js'></script>
<script type="text/javascript">
mapboxgl.accessToken =
'pk.eyJ1IjoiZWZoYWwiLCJhIjoiY2ptOXRiZ3k2MDh4bzNrbnljMjk5Z2d5aSJ9.
8dSNgeAjpdTlZ3x-b2vsog';
var map = new mapboxgl.Map({
container: 'map', // container id
style: 'mapbox://styles/mapbox/streets-v9', // stylesheet location
center: [122.514900, -3.972201], // starting position [lng, lat]
zoom: 10, // starting zoom
logoPosition: 'top-right', });
var dtb_ = $("#datatable").DataTable({ "serverSide": true,
```

```
"responsive": true,
"ajax": { "url": "<?= site_url('admin/graph/ajax/list') ?>", "type": "POST", } });
var marker = [];
$.ajax({ 'url': "<?= site_url('admin/graph/ajax/data') ?>",
'type': 'POST', success: function(e) { var data_obj = JSON.parse(e);
data_obj.forEach(function(i) { var color = i.type == '-' ? '#01f254' : '#015ff2';
marker.push(new mapboxgl.Marker({ color: color, }) .setLngLat([i.lng, i.lat])
.setPopup(new mapboxgl.Popup().setHTML(`
<div class="card" style="width: 10rem;">
<img src="<?= base_url('uploads/') ?>${i.picture}" class="card-img-top"
alt="...">
<div class="card-body">
<h6 class="card-title">${i.name}</h6>
${i.type == 'object' ? `<a href="<?= site_url('gunung/detail/') ?>${i.id}"
class="btn btn-primary">Lihat detail</a>` : "} </div> </div>
`)) // add popup
.addTo(map)); }) } });
function deleteData(id) {
var conf = confirm('Apakah anda yakin untuk menghapus data ini ?');
if (conf) {
window.location = "<?= site_url('admin/graph/delete/') ?>" + id; } }
$("#start,#end").on('change', function() {
if ($("#start").val() != '' && $("#end").val() != '') {
var distance = turf.distance(turf.point([$("#start").find(":selected").attr('lng'),
$("#start").find(":selected").attr('lat')]),
turf.point([$("#end").find(":selected").attr('lng'),
$("#end").find(":selected").attr('lat')]), {units: 'kilometers'}).toFixed(2);
$("#distance").val(distance); } })
$("#btnSave").click(function(e) { e.preventDefault();
var url = $("#id_graph").val() != '' ? "<?= site_url('admin/graph/edit/') ?>" +
$("#id_graph").val() : "<?= site_url('admin/graph/add') ?>";
$.ajax({url: url, type: 'POST', data: $("#graphForm").serialize(), success:
function(e) { $(".bs-example-modal-center").modal('hide'); dtb_.draw(); } }); })
```

```
$(".bs-example-modal-center").on('hidden.bs.modal', function(e) {
$("#id_graph").val(""); $("#start").val(""); $("#end").val("");
$("#distance").val(""); // $("#time").val(""); // $("#time").val(""); })
function showModalEditGraph(id) {
$(".bs-example-modal-center").modal('show');
$.ajax({ url: '<?= site_url('admin/graph/edit/') ?>' + id, type: 'POST',
data: { 'id': id },
success: function(e) { var Obj = JSON.parse(e); $("#id_graph").val(Obj.id);
$("#start").val(Obj.start); $("#end").val(Obj.end);
$("#distance").val(Obj.distance); // $("#time").val(Obj.time); } }) } map.on('load',
function(e) { $.ajax({ url: '<?= site_url('admin/graph/getGraphLine') ?>',
success: function(e) { var obj = JSON.parse(e);
obj.forEach(function(i) { map.addLayer({ "id": "route" + i.g_id, "type": "line",
"source": { "type": "geojson", "data": { "type": "Feature", "properties": {},
"geometry": { "type": "LineString", "coordinates": [ [i.n1_lng, i.n1_lat], [i.n2_lng,
i.n2_lat], ] } } }, "layout": { "line-join": "round", "line-cap": "round" },
"paint": { "line-color": "#2980b9", "line-width": 5 } }) }) } }); });
</script>
</body>
</html>
```

Admin Node [index.php]

```
<!doctype html>
<html lang="en">
<head>
    <?php $this->load->view('_parts/style') ?>
    <link href="<?= base_url() ?>assets/libs/datatables.net-
    bs4/css/dataTables.bootstrap4.min.css" rel="stylesheet" type="text/css" />
    <link href="<?= base_url() ?>assets/libs/datatables.net-buttons-
    bs4/css/buttons.bootstrap4.min.css" rel="stylesheet" type="text/css" />
</head>
<body>
<!-- Begin page -->
```

```
<div id="layout-wrapper">
<?php $this->load->view('_parts/header') ?>
<?php $this->load->view('_parts/sidebar') ?>
<!--
========================================================================
==== -->
<!-- Start right Content here -->
<!--
========================================================================
==== -->
<div class="main-content" style="margin-top:100px;">
<div class="page-content"> <div class="container-fluid"> <div class="page-
content-wrapper">
<div class="mt-3">
<h3 class=""><strong>Titik Node </strong></h3> </div>
<div class="row"> <div class="col-lg-6"> <div class="card"> <div class="card-
body">
<a href="<?= site_url('admin/node/add') ?>" class="btn btn-primary mb-
3">Tambah Titik Node</a>
<div class="p-0 table-responsive">
<p>Berikut adalah data node yang terdaftar.</p>
<?= $this->session->flashdata('statusMessage') ?>
<hr />
<table id="datatable" class="table table-sm table-bordered dt-responsive nowrap"
style="border-collapse: collapse; border-spacing: 0; width: 100%;">
<thead>
<tr>
<th>Node</th>
<th>lat</th>
<th>lng</th>
<th>aksi</th>
</tr>
</thead>
<tbody></tbody>
```

```
</table> </div> </div> </div> </div>
<div class="col-lg-6"> <div class="card"> <div class="card-body">
<div id="map" style="height: 450px;width: 100%;"></div>
</div> </div> </div> </div> </div>
</div> <!-- container-fluid --> </div>
<!-- End Page-content -->
<?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content--> </div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
<script src="<?= base_url()
?>assets/libs/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="<?= base_url() ?>assets/libs/datatables.net-
bs4/js/dataTables.bootstrap4.min.js"></script>
<script type="text/javascript">
mapboxgl.accessToken =
'pk.eyJ1IjoiZWZoYWwiLCJhIjoiY2ptOXRiZ3k2MDh4bzNrbnljMjk5Z2d5aSJ9.
8dSNgeAjpdTlZ3x-b2vsog';
var map = new mapboxgl.Map ({  container: 'map', // container id
style: 'mapbox://styles/mapbox/streets-v9', // stylesheet location
center: [122.514900, -3.972201], // starting position [lng, lat]
zoom: 10, // starting zoom
logoPosition: 'top-right', });
var dtb_ = $("#datatable").DataTable({
"serverSide": true, "responsive": true, "ajax": { "url": "<?=
site_url('admin/node/ajax/list') ?>", "type": "POST", } });
var marker = [];
$.ajax({
'url': "<?= site_url('admin/graph/ajax/data') ?>",
'type': 'POST',
success: function(e) { var data_obj = JSON.parse(e);
data_obj.forEach(function(i) { var color = i.type == '-' ? '#01f254' : '#015ff2';
marker.push(new mapboxgl.Marker ({ color: color, })
```

```
.setLngLat([i.lng, i.lat])
.setPopup(new mapboxgl.Popup().setHTML(`
<div class="card" style="width: 10rem;">
<img src="<?= base_url('uploads/') ?>${i.picture}" class="card-img-top"
alt="...">
<div class="card-body">
<h6 class="card-title">${i.name}</h6>
${i.type == 'object' ? `<a href="<?= site_url('gunung/detail/') ?>${i.id}"
class="btn btn-primary">Lihat detail</a>` : ''}
</div> </div>
`)) // add popup
.addTo(map));
}) } });
function deleteData(id) {
var conf = confirm('Apakah anda yakin untuk menghapus data ini ?');
if (conf) { window.location = "<?= site_url('admin/node/delete/') ?>" + id; } }
</script>
</body>
</html>
```

Admin Objects [index.php]

```
<!doctype html>
<html lang="en">
<head>
<?php $this->load->view('_parts/style') ?>
<link href="<?= base_url() ?>assets/libs/datatables.net-
bs4/css/dataTables.bootstrap4.min.css" rel="stylesheet" type="text/css" />
<link href="<?= base_url() ?>assets/libs/datatables.net-buttons-
bs4/css/buttons.bootstrap4.min.css" rel="stylesheet" type="text/css" />
</head>
<body>
<!-- Begin page -->
```

```
<div id="layout-wrapper">
<?php $this->load->view('_parts/header') ?>
<?php $this->load->view('_parts/sidebar') ?>
<!--
========================================================================
==== -->
<!-- Start right Content here -->
<!--
========================================================================
==== -->
<div class="main-content" style="margin-top:100px;">
<div class="page-content"> <div class="container-fluid">
<div class="page-content-wrapper"> <div class="mt-3">
<h3 class=""><strong>Pantai </strong></h3>
</div>
<div class="row"> <div class="col-lg-6"> <div class="card"> <div class="card-
body">
<a href="<?= site_url('admin/pantai/add') ?>" class="btn btn-primary mb-
3">Tambah Titik Pantai</a>
<div class="p-0 table-responsive">
<p>Berikut adalah data pantai yang terdaftar.</p>
<?= $this->session->flashdata('statusMessage') ?>
<hr />
<table id="datatable" class="table table-sm table-bordered dt-responsive nowrap"
style="border-collapse: collapse; border-spacing: 0; width: 100%;">
<thead>
<tr>
<th>Nama Pantai</th>
<th>lat</th>
<th>lng</th>
<th>aksi</th>
</tr>
</thead>
```

```
<tbody></tbody>
</table>
</div> </div> </div> </div>
<div class="col-lg-6"> <div class="card"> <div class="card-body">
<div id="map" style="height: 450px;width: 100%;"></div>
</div> </div> </div> </div>
</div> </div> <!-- container-fluid --> </div>
<!-- End Page-content -->
<?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content--> </div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
<script src="<?= base_url()
?>assets/libs/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="<?= base_url() ?>assets/libs/datatables.net-
bs4/js/dataTables.bootstrap4.min.js"></script>
<script type="text/javascript">
mapboxgl.accessToken =
'pk.eyJ1IjoiZWZoYWwiLCJhIjoiY2ptOXRiZ3k2MDh4bzNrbnljMjk5Z2d5aSJ9.
8dSNgeAjpdTlZ3x-b2vsog';
var map = new mapboxgl.Map({ container: 'map', // container id
style: 'mapbox://styles/mapbox/streets-v9', // stylesheet location
center: [122.514900, -3.972201], // starting position [lng, lat]
zoom: 10, // starting zoom
logoPosition: 'top-right', });
var dtb_ = $("#datatable").DataTable({ "serverSide": true, "responsive": true,
"ajax": { "url": "<?= site_url('admin/pantai/ajax/list') ?>", "type": "POST", }
}); var marker = [];
$.ajax({ 'url': "<?= site_url('admin/graph/ajax/data') ?>",
'type': 'POST', success: function(e) {
var data_obj = JSON.parse(e);
data_obj.forEach(function(i) {
```

```
var color = i.type == '-' ? '#01f254' : '#015ff2';
marker.push(new mapboxgl.Marker({
color: color, })
.setLngLat([i.lng, i.lat])
.setPopup(new mapboxgl.Popup().setHTML(`
<div class="card" style="width: 10rem;">
<img src="<?= base_url('uploads/') ?>${i.picture}" class="card-img-top"
alt="...">
<div class="card-body">
<h6 class="card-title">${i.name}</h6>
${i.type == 'object' ? `<a href="<?= site_url('pantai/detail/') ?>${i.id}" class="btn
btn-primary">Lihat detail</a>` : ''}
</div> </div>
`)) // add popup .addTo(map)); }) } });
function deleteData(id) {
var conf = confirm('Apakah anda yakin untuk menghapus data ini ?');
if (conf) {
window.location = "<?= site_url('admin/pantai/delete/') ?>" + id; } }
</script>
</body>
</html>
```

Login Admin [index.php]

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title><?= $title ?></title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta content="Premium Multipurpose Admin & Dashboard Template"
name="description" />
    <meta content="Themesdesign" name="author" />
```

```
    <!-- App favicon -->
    <link rel="shortcut icon" href="assets/faviconpantai.png">
    <!-- Bootstrap Css -->
    <link href="<?= base_url() ?>assets/css/bootstrap.min.css" id="bootstrap-style"
rel="stylesheet" type="text/css" />
    <!-- Icons Css -->
    <link href="<?= base_url() ?>assets/css/icons.min.css" rel="stylesheet"
type="text/css" />
    <!-- App Css-->
    <link href="<?= base_url() ?>assets/css/app.min.css" id="app-style"
rel="stylesheet" type="text/css" />
</head>
<body class="bg-primary mt-5" >
    <div class="home-center"> <div class="home-desc-center">
    <div class="container"> <div class="row justify-content-center mt-3">
    <div class="col-md-8 col-lg-6 col-xl-5">
    <div class="card"> <div class="card-body">
    <div class="px-2 py-3"> <div class="text-center">
        <a href="<?= site_url() ?>">
        <img src="assets/adminlogin.png" height="100" alt="logo"> </a>
        <h5 class="text-primary mb-2 mt-4">Selamat Datang Admin !</h5>
        </div>
    <form class="form-horizontal mt-4 pt-2" method="POST" action="<?=
site_url('login') ?>">
    <div class="mb-3"> <label for="username">Username</label>
    <input type="text" class="form-control" id="username" name="username"
value="<?= set_value('username') ?>" placeholder="Masukan username">
    <?= form_error('username') ?>
    </div> <div class="mb-3">
    <label for="userpassword">Password</label>
    <input type="password" class="form-control" id="userpassword"
name="password" value="<?= set_value('password') ?>" placeholder="Masukan
password">
```

```
<?= form_error('password') ?> </div> <div>

<button class="btn btn-primary w-100 waves-effect waves-light"
type="submit">Log In</button> </div>

<div class="mt-3"> <?= $this->session->flashdata('loginError') ?>

</div> </form> <div>

<button class="btn btn-warning w-100 waves-effect waves-light"><a
href="http://localhost/rutepantai/">Kembali</a></button>

</div> </div> </div> </div> </div> </div> </div>

<!-- End Log In page --> </div>

<!-- JAVASCRIPT -->

<script src="<?= base_url() ?>assets/libs/jquery/jquery.min.js"></script>

<script src="<?= base_url()
?>assets/libs/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="<?= base_url()
?>assets/libs/metismenu/metisMenu.min.js"></script>

<script src="<?= base_url()
?>assets/libs/simplebar/simplebar.min.js"></script>

<script src="<?= base_url() ?>assets/libs/node-waves/waves.min.js"></script>

<script src="<?= base_url() ?>assets/js/app.js"></script>

</body>

</html>
```

Admin User [index.php]

```
<!doctype html>

<html lang="en">

<head>

<?php $this->load->view('_parts/style') ?>

<link href="<?= base_url() ?>assets/libs/datatables.net-
bs4/css/dataTables.bootstrap4.min.css" rel="stylesheet" type="text/css" />

<link href="<?= base_url() ?>assets/libs/datatables.net-buttons-
bs4/css/buttons.bootstrap4.min.css" rel="stylesheet" type="text/css" />

</head>

<body>
```

```
<!-- Begin page -->
<div id="layout-wrapper">
<?php $this->load->view('_parts/header') ?>
<?php $this->load->view('_parts/sidebar') ?>
<!--
========================================================================
==== -->
<!-- Start right Content here -->
<!--
========================================================================
==== -->
<div class="main-content" style="margin-top:100px;">
<div class="page-content"> <div class="container-fluid">
<div class="page-content-wrapper"> <div class="mt-3">
<h3 class=""><strong><?= $title ?></strong></h3>
</div> <div class="row"> <div class="col-lg-12"> <div class="card">
<div class="card-body">
<a href="<?= site_url('admin/user/add') ?>" class="btn btn-primary mb-
3">Tambah User</a>
<div class="p-0 table-responsive">
<p>Berikut adalah data user yang terdaftar.</p>
<?= $this->session->flashdata('statusMessage') ?>
<hr />
<table id="datatable" class="table table-sm table-bordered dt-responsive nowrap"
style="border-collapse: collapse; border-spacing: 0; width: 100%;">
<thead>
        <tr>
                <th>Username</th>
                <th>password</th>
                <th>diinsert pada</th>
                <th>aksi</th>
        </tr>
</thead>
```

```
<tbody></tbody>
</table>
</div> </div> </div> </div> </div> </div>
</div> <!-- container-fluid --> </div>
<!-- End Page-content -->
<?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content--> </div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
<script src="<?= base_url()
?>assets/libs/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="<?= base_url() ?>assets/libs/datatables.net-
bs4/js/dataTables.bootstrap4.min.js"></script>
<script type="text/javascript">
var dtb_ = $("#datatable").DataTable({
"serverSide": true, "responsive": true,
"ajax": { "url": "<?= site_url('admin/user/ajax/list') ?>",
"type": "POST", } }); function deleteData(id) {
var conf = confirm('Apakah anda yakin untuk menghapus data ini ?');
if (conf) { window.location = "<?= site_url('admin/user/delete/') ?>" + id; } }
</script>
</body>
</html>
```

[pantai.php]

```
<!doctype html>
<html lang="en">
<head>
<?php $this->load->view('_parts/style') ?>
</head>
<body>
```

```
<!-- Begin page -->
<div id="layout-wrapper">
<?php $this->load->view('_parts/header') ?>
<?php $this->load->view('_parts/sidebar') ?>
<!--
================================================================
==== -->
<!-- Start right Content here -->
<!--
================================================================
==== -->
<div class="main-content" style="margin-top:100px;">
<div class="page-content">
<div class="container-fluid">
<div class="page-content-wrapper">
<div class="mt-3">
<h3 class=""><strong>Informasi Pantai-Pantai Pulau Nias<strong></h3>
</div> <div class="row mb-4"> </div>
<div class="row ">
<?php foreach ($objectResult as $gng) { ?>
<div class="col-6 col-sm-3 d-flex align-items-stretch">
<div class="card">
        <img class="card-img-top" style="height:200px;object-fit:cover;"
src="<?= base_url('uploads/' . $gng->picture) ?>" alt="Card image cap">
        <div class="card-body">
        <h5 class="card-title"><?= $gng->name ?></h5>
        <p class="card-text text-sm"><?= substr($gng->desc, 0, 50) . '...' ?><br
/><a href="<?= site_url('pantai/detail/' . $gng->id)
?>"><strong>Selengkapnya</strong></a></p>
</div> </div> </div>
<?php } ?> <div class="col-12"> <nav aria-label="Page navigation example">
<?php echo $this->pagination->create_links(); ?>
</nav> </div> </div> </div>
```

```
</div> <!-- container-fluid --> </div>
<!-- End Page-content -->
<?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content--> </div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
</body>
<script>
        $('.carousel').carousel();
</script>
</html>
```

[detailPantai.php]

```
<!doctype html>
<html lang="en">
<head>
<?php $this->load->view('_parts/style') ?>
</head>
<body>
<!-- Begin page -->
<div id="layout-wrapper">
<?php $this->load->view('_parts/header') ?>
<?php $this->load->view('_parts/sidebar') ?>
<!--
==========================================================
==== -->
<!-- Start right Content here -->
<!--
==========================================================
==== -->
<div class="main-content" style="margin-top:100px;">
<div class="page-content">
```

```
<div class="container-fluid">
        <div class="page-content-wrapper">
        <div class="mt-3">
        <h3 class=""><strong><?= $objectRow->name ?></strong></h3>
        </div>
<div class="row">
        <div class="col-12 col-sm-8">
        <div class="card">
        <img class="card-img-top" style="height:400px;object-fit:cover;"
src="<?= base_url('uploads/' . $objectRow->picture) ?>" alt="Card image cap">
        </div>
<div class="card">
<div class="card-body">
<p class="card-text"><?= $objectRow->desc ?></p>
</div> </div> </div>
<div class="col-12 col-sm-4"> <div class="card"> <div class="card-body">
<div id="map" style="height: 450px;width: 100%;"></div>
</div> </div> </div> </div> </div>
</div> <!-- container-fluid --> </div>
<!-- End Page-content -->
<?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content-->
</div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
</body>
<script type="text/javascript">
mapboxgl.accessToken =
'pk.eyJ1IjoiZWZoYWwiLCJhIjoiY2ptOXRiZ3k2MDh4bzNrbnljMjk5Z2d5aSJ9.
8dSNgeAjpdTlZ3x-b2vsog';
var map = new mapboxgl.Map({
container: 'map', // container id
```

```
style: 'mapbox://styles/mapbox/streets-v9', // stylesheet location
center: [<?= $objectRow->lng ?>, <?= $objectRow->lat ?>], // starting position [lng, lat]
zoom: <?= DEFAULT_ZOOM ?>, // starting zoom
logoPosition: 'top-right', });
marker = new mapboxgl.Marker({
color: "#7d000c", })
.setLngLat([<?= $objectRow->lng ?>, <?= $objectRow->lat ?>])
.addTo(map);
</script>
</html>
```

[galeri.php]

```
<!doctype html>
<html lang="en">
<head>
<?php $this->load->view('_parts/style') ?>
</head>
<body>
<!-- Begin page -->
<div id="layout-wrapper">
<?php $this->load->view('_parts/header') ?>
<?php $this->load->view('_parts/sidebar') ?>
<!-- modal -->
<!-- Modal -->
<div class="modal fade" id="modalGaleri" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog modal-xl modal-dialog modal-dialog-centered">
<div class="modal-content"> <div class="modal-body">
<img id="imageModal" src="" style="object-fit:center;width:100%;"> </div>
<div class="modal-footer">
```

```
<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>

</div> </div> </div> </div>

<!-- /modal -->

<!--
================================================================
==== -->

<!-- Start right Content here -->

<!--
================================================================
==== -->

<div class="main-content" style="margin-top:100px;">

<div class="page-content"> <div class="container-fluid"> <div class="page-
content-wrapper"> <div class="mt-3">

<h3 class=""><strong><?= $title ?></strong></h3> </div>

<div class="row mb-4">

<div id="carouselExampleCaptions" class="carousel slide" data-bs-
ride="carousel">

<div class="carousel-indicators">

<?php $no = 0; foreach ($objectResult as $gng) { ?>

<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-
to="<?= $no++; ?>" class="active" aria-current="true" aria-label="Slide
1"></button>

<?php } ?> </div>

<div class="carousel-inner">

<?php $carsoulStatus = true;

foreach ($objectResult as $gng) { ?>

<div class="carousel-item

<?php if ($carsoulStatus) {

echo "active";

$carsoulStatus = false;

} ?> ">

<img src="<?= base_url('uploads/' . $gng->picture) ?>" style="object-
fit:cover;height:300px;" class="d-block w-100" alt="...">
```

```
<div class="carousel-caption">
<h3 class="text-light"><?= $gng->name ?></h3>
</div> </div> <?php } ?> </div>
<button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="prev">
<span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="visually-hidden">Next</span>
</button>
</div> </div> <div class="row ">
<?php foreach ($objectResult as $gng) { ?>
<div class="col-6 col-sm-3 d-flex align-items-stretch">
<div class="card">
<img class="card-img-top" style="height:200px;object-fit:cover;"
onclick="showModalGaleri('<?= $gng->picture ?>')" src="<?= base_url('uploads/'
. $gng->picture) ?>" alt="Card image cap">
<div class="card-body">
<h5 class="card-title"><?= $gng->name ?></h5>
</div> </div> </div> <?php } ?> <div class="col-12">
<nav aria-label="Page navigation example">
<?php echo $this->pagination->create_links(); ?>
</nav> </div> </div> </div>
</div> <!-- container-fluid --> </div>
<!-- End Page-content -->
<?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content--> </div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s" crossorigin="anonymous"></script>

</body>

<script>

$('.carousel').carousel()

function showModalGaleri(pic) {

$("#imageModal").attr('src', '<?= base_url('uploads/') ?>' + pic)

$("#modalGaleri").modal('show'); }

</script>

</html>
```

Pencarian Rute [djikstra.php]

```
<!doctype html>

<html lang="en">

<head>

<?php $this->load->view('_parts/style') ?>

</head>

<body>

<!-- Begin page -->

<div id="layout-wrapper">

<?php $this->load->view('_parts/header') ?>

<?php $this->load->view('_parts/sidebar') ?>

<!-- Modal detail perhitungan-->

<!-- Modal -->
```

```
<div class="modal fade" id="modalDetailPerhitungan" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">Perhitungan</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
</div>
<div class="modal-body">
<span id="resDetailPerhitungan"></span>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
</div> </div> </div> </div>
<!-- /Modal detail perhitungan-->
<!--
==========================================================
==== -->
<!-- Start right Content here -->
<!--
==========================================================
==== -->
<div class="main-content" style="margin-top:100px;">
<div class="page-content"> <div class="container-fluid">
<div class="page-content-wrapper"> <div class="mt-3">
<h3 class=""><strong><?= $title ?></strong></h3> </div>
<div class="row">
<div class="col-12 col-sm-4">
<form id="graphForm" class="mb-4 mb-sm-2">
<input type="hidden" id="id_graph" name="id_graph">
<div class="row">
```

```
<div class="col-12 col-sm-6 mb-3">
<select id="start" name="start" class="form-control select2">
<option value="">Pilih Titik Kamu</option>
<?php foreach ($nodeResult as $n) { ?>
<?php if ($n->type == '-') { ?>
<option lng="<?= $n->lat ?>" lat="<?= $n->lng ?>" value="<?= $n->id ?>"><?= $n->name ?></option>
<?php } ?> <?php } ?>
</select> </div> </div>
</form> </div>
<div class="col-12 col-sm-12 mb-5">
<div id="map" style="height: 550px;width: 100%;"></div>
<!-- left side instructions -->
<div id='keterangan' style=" position:absolute;
height: 100px; margin:9px; width: 20%; top:0; bottom:0; padding: 20px; background-color: rgba(255,255,255,0.9); font-family: sans-serif;">
<div id="calculated-line"></div> </div> </div> </div> </div>
</div> <!-- container-fluid --> </div>
<!-- End Page-content --> <?php $this->load->view('_parts/footer') ?> </div>
<!-- end main content--> </div>
<!-- END layout-wrapper -->
<?php $this->load->view('_parts/js') ?>
<script src="<?= base_url() ?>assets/libs/select2/js/select2.min.js"></script>
<script src='https://unpkg.com/@turf/turf@6/turf.min.js'></script>
<script src="https://unpkg.com/leaflet@1.2.0/dist/leaflet.js"></script>
<script src="https://unpkg.com/leaflet-routing-machine@latest/dist/leaflet-routing-machine.js"></script>
<script type="text/javascript">
mapboxgl.accessToken = 'pk.eyJ1IjoiZWZoYWwiLCJhIjoiY2ptOXRiZ3k2MDh4bzNrbnljMjk5Z2d5aSJ9.8dSNgeAjpdTlZ3x-b2vsog';
var map = new mapboxgl.Map({
container: 'map', // id wadah
```

```
style: 'mapbox://styles/mapbox/satellite-v9', // lokasi lembar gaya
center: [97.524809, 1.124700], // posisi awal [lng, lat]
zoom: 9, // mulai zoom
logoPosition: 'top-right', });
if (!navigator.geolocation) {
console.log('Geolocation is not supported by your browser');
} else {
console.log('Locating…');
navigator.geolocation.getCurrentPosition(success, error); }
var current_latitude = ''; var current_longitude = '';
function success(position) { current_latitude = position.coords.latitude;
current_longitude = position.coords.longitude; }
function error() { console.log('Geolocation error !'); }
map.addControl( new mapboxgl.GeolocateControl({ positionOptions: {
enableHighAccuracy: true },
// Saat aktif, peta akan menerima pembaruan ke lokasi perangkat saat berubah.
trackUserLocation: true,
// Gambar panah di sebelah titik lokasi untuk menunjukkan arah yang dituju perangkat.
showUserHeading: true }) );
$('#start').select2(); $('#end').select2();
var marker = [];
$.ajax({ 'url': "<?= site_url('admin/graph/ajax/data') ?>", 'type': 'POST',
success: function(e) { var data_obj = JSON.parse(e);
data_obj.forEach(function(i) { if (i.type != '-') { var color = i.type == '-' ? '#01f254' : '#015ff2'; marker.push(new mapboxgl.Marker({
color: color, })
.setLngLat([i.lng, i.lat])
.setPopup(new mapboxgl.Popup().setHTML(`
<div class="card" style="width: 10rem;">
<img src="<?= base_url('uploads/') ?>${i.picture}" class="card-img-top" alt="...">
```

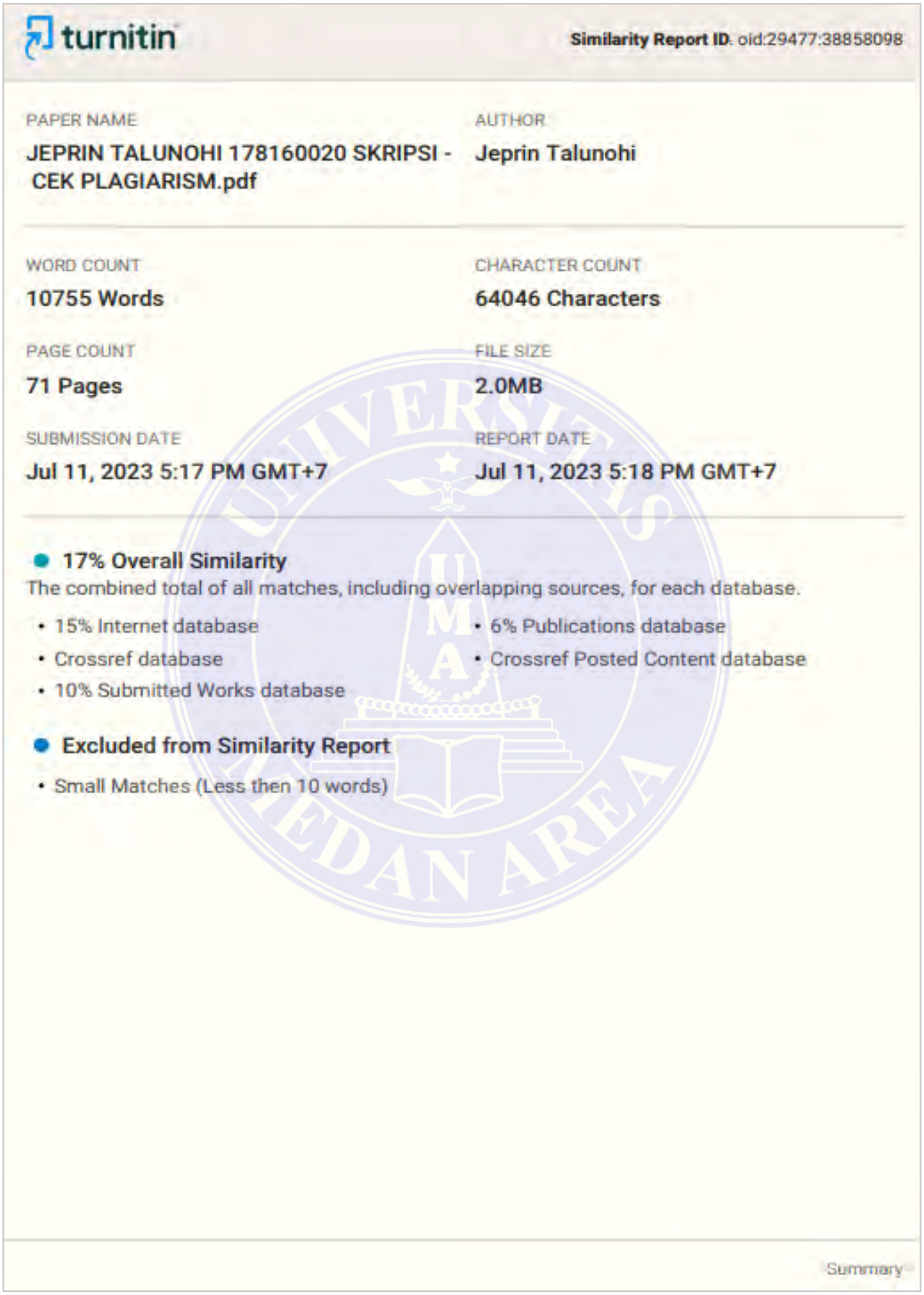
```
<div class="card-body">
<h5 class="card-title">${i.name}</h5>
<a href="<?= site_url('pantai/detail/') ?>${i.id}" class="btn btn-primary">Lihat detail</a> </div> </div>
`)) // tambahkan munculan
.addTo(map)); } }) } });
var marker_start;
$("#start").change(function(e) { e.preventDefault();
var start_atribute = $('#start').select2('data')[0].element.attributes;
if (marker_start) { marker_start.remove(); }
if ($("#start").val() == 'current_location') {
lng = current_latitude;
lat = current_longitude;
// penanda dengan posisi terdekat dari lokasi sekarang
$.ajax({
url: '<?= site_url('getMarker') ?>',
success: function(e) { var markers = JSON.parse(e);
var distance = [];
markers.forEach(function(i) { distance.push({ id: i.id,
distance: turf.distance(turf.point([lng, lat]), turf.point([i.lat, i.lng]), {
units: 'kilometers' }) }); });
distance.sort((a, b) => a.distance - b.distance);
console.log(distance);
$("#start").val(distance[0].id); } })
} else { lng = start_atribute.lng.value; lat = start_atribute.lat.value; }
map.flyTo({ center: [ lat, lng ], essential: true // animasi ini dianggap penting sehubungan dengan gerakan yang lebih disukai-dikurangi });
marker_start = new mapboxgl.Marker({
color: "#ff0000", })
.setLngLat([lat, lng])
.addTo(map) });
```

```
var draw = new MapboxDraw({
displayControlsDefault: false,
controls: { line_string: true, trash: true }, styles: [
// tarik garis
{"id": "gl-draw-line", "type": "line", "filter": ["all", ["==", "$type", "LineString"], ["!=", "mode", "static"]],
"layout": {"line-cap": "round", "line-join": "round"},
"paint": {"line-color": "#FF0000", "line-dasharray": [0.2, 2], "line-width": 4,
"line-opacity": 0.7} },
        // titik simpul
{"id": "gl-draw-polygon-and-line-vertex-halo-active", "type": "circle",
"filter": ["all", ["==", "meta", "vertex"], ["==", "$type", "Point"], ["!=", "mode", "static"]],
"paint": {"circle-radius": 10, "circle-color": "#FFF"} },
// Disini Titik
{"id": "gl-draw-polygon-and-line-vertex-active", "type": "circle",
"filter": ["all", ["==", "meta", "vertex"], ["==", "$type", "Point"], ["!=", "mode", "static"]],
"paint": {"circle-radius": 6, "circle-color": "#3b9ddd", } }, ] });
// Alat gambar peta
map.addControl(draw);
// buat, perbarui, atau hapus tindakan
map.on('draw.create', updateRoute);
map.on('draw.update', updateRoute);
map.on('draw.delete', removeRoute);
// gunakan koordinat yang baru saja Anda gambar untuk membuat permintaan arah Anda
function updateRoute() {
removeRoute(); // menimpa lapisan
var data = draw.getAll();
var lastFeature = data.features.length - 1;
var coords = data.features[lastFeature].geometry.coordinates;
```

```
var newCoords = coords.join(';');
getMatch(newCoords); }
// membuat permintaan arah
function getMatch(e) {
var url = 'https://api.mapbox.com/directions/v5/mapbox/cycling/' + e
+'?geometries=geojson&steps=true&access_token=' + mapboxgl.accessToken;
var req = new XMLHttpRequest();
req.responseType = 'json'; req.open('GET', url, true);
req.onload  = function() { var jsonResponse = req.response;
var distance = jsonResponse.routes[0].distance*0.001;
var duration = jsonResponse.routes[0].duration/60;
var steps = jsonResponse.routes[0].legs[0].steps;
var coords = jsonResponse.routes[0].geometry;
//  console.log(steps);
console.log(coords);
//  console.log(distance);
// console.log(duration);
// dapatkan jarak dan durasi
keterangan.insertAdjacentHTML('beforeend', '<p>' +  'Jarak Rute: ' +
distance.toFixed(2) + ' km<br>Durasi Waktu: ' + duration.toFixed(2) + ' minutes'
+ '</p>');
// tambahkan rute ke peta
addRoute(coords);
//  console.log(coordinates); };
req.send(); }
// menambahkan rute sebagai lapisan di peta
function addRoute (coords) {
// periksa apakah rute sudah dimuat
if (map.getSource('route')) {
map.removeLayer('route');
map.removeSource('route') } else{
```

```
map.addLayer({ "id": "route", "type": "line",
"source": { "type": "geojson", "data": { "type": "Feature", "properties": {},
"geometry": coords } }, "layout": { "line-join": "round", "line-cap": "round" },
"paint": { "line-color": "#1db7dd", "line-width": 8, "line-opacity": 0.8 } }); }; }
// hapus layer jika ada
function removeRoute () {
if (map.getSource('route')) {
map.removeLayer('route');
map.removeSource('route');
// instructions.innerHTML = '';
} else  {
   return; } }
document.getElementById('geocoder').appendChild(geocoder.onAdd(map));
</script>
</body>
</html>
```

Hasil Plagiarisme



turnitin

Similarity Report ID: oid:29477:38858098

PAPER NAME
JEPRIN TALUNOHI 178160020 SKRIPSI - CEK PLAGIARISM.pdf

AUTHOR
Jeprin Talunohi

WORD COUNT
10755 Words

CHARACTER COUNT
64046 Characters

PAGE COUNT
71 Pages

FILE SIZE
2.0MB

SUBMISSION DATE
Jul 11, 2023 5:17 PM GMT+7

REPORT DATE
Jul 11, 2023 5:18 PM GMT+7

● 17% Overall Similarity
The combined total of all matches, including overlapping sources, for each database.

- 15% Internet database
- 6% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

● Excluded from Similarity Report

- Small Matches (Less then 10 words)

Summary

SK Dosen Pembimbing

UNIVERSITAS MEDAN AREA
FAKULTAS TEKNIK
Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1☎(061) 7366878, 7360168, 7364348, 7366781, Fax.(061) 7366998 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225602, Fax. (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 257/FT.6/01.10/IX/2022 5 September 2022
Lamp : -
H a l : **Perpanjang SK Pembimbing Tugas Akhir**

Yth. Pembimbing Tugas Akhir
Zulfikar Sembiring, S. Kom, M. Kom
Nurul Khairina, S. Kom, M. Kom
di
Tempat

Dengan hormat,
Sehubungan telah berakhirnya waktu masa berlaku SK pembimbing nomor 58/FT.6/01.10/II/2022 tertanggal 7 Februari 2022 maka perlu diterbitkan kembali SK Pembimbing Skripsi baru atas nama mahasiswa berikut :

N a m a : Jeprin Talunohi
N P M : 178160020
Jurusan : Informatika

Oleh karena itu kami mengharapkan kesediaan saudara :

1. Zulfikar Sembiring, S. Kom, M. Kom (Sebagai Pembimbing I)
2. Nurul Khairina, S. Kom, M. Kom (Sebagai Pembimbing II)

Adapun Tugas Akhir Skripsi berjudul :

"Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Ke Wisata Pantai Pulau Nias"

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,

Dr. Rahmad Syah, S. Kom, M. Kom

SK Pengantar Riset

UNIVERSITAS MEDAN AREA
FAKULTAS TEKNIK
Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1☎(061) 7366878, 7360168, 7364348, 7366781, Fax (061) 7366998 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225602, Fax (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 72/FT.6/01.10/II/2022 18 Februari 2022
Lamp : -
H a l : **Penelitian Dan Pengambilan Data Tugas Akhir**

Yth. Kepala Dinas Kebudayaan, Pariwisata & Kepemudaan Olahraga
Jln. Hiliamaetaniha, Kec. Tlk. Dalam
Di
Nias

Dengan hormat,
Kami mohon kesediaan Bapak/Ibu berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	N A M A	N P M	PRODI
1	Jeprin Talunohi	178160020	Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir pada perusahaan/Instansi yang Bapak/Ibu Pimpin.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan ilmiah dan Skripsi yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana lengkap pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul penelitian :

Implementasi Algoritma *Dijkstra* untuk Menentukan Rute Terpendek Ke Wisata Pantai Pulau Nias

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

a.n. Dekan,

Dr. Rahmad Syah, S. Kom, M. Kom

Tembusan :
1. Ka. BAMAI
2. Mahasiswa
3. File

SK Selesai Riset

BLESSING BEACH
NIAS SELATAN
Jl. Baloho, Hilitobara Kec. Tlk. Dalam Kab. Nias Selatan - Sumut

SURAT KETERANGAN

02/AUU/019/2021

Berdasarkan surat dekan fakultas Teknik Universitas Medan Area nomor:72/FT.6/01.10/II/2022 tanggal **18 Februari 2022** tentang "**Penelitian dan Mengambil Data Tugas Akhir**" maka dengan ini BLESSING BEACH menerangkan bahwa:

Nama : JEPRIN TALUNOHI

Nim : 178160020

Jenjang/Jurusan : S1 – Teknik Informatika

Telah selesai melaksanakan Penelitian dan Pengambilan Data Tugas Akhir Dengan Judul:

"**Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Ke Wisata Pantai Pulau Nias**"

Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya.

Kab. Nisel, 25 Maret 2022

Pengelola Blessing Beach,

MAISELINUS FAU