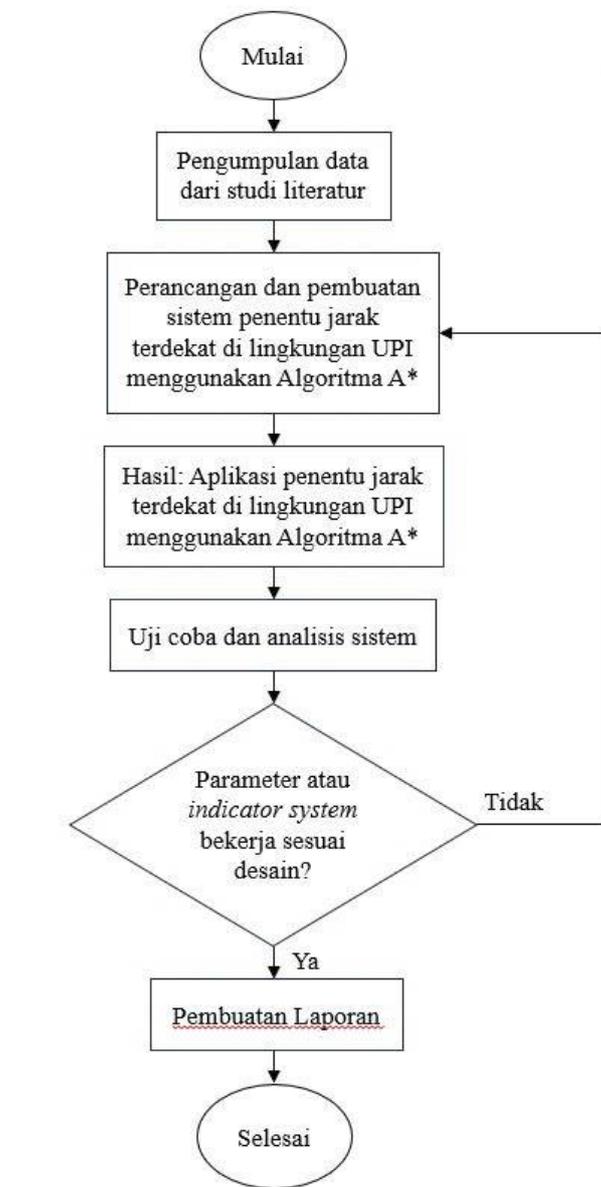


BAB III

METODE PENELITIAN

3.1 Prosedur Penelitian

Penelitian diawali dengan melakukan observasi, pengumpulan data dari studi literatur. Dilanjutkan dengan melakukan perancangan dan pembuatan sistem penentu jarak terdekat di lingkungan UPI menggunakan Algoritma A*. Uji coba dan analisis sistem hingga analisis hasil penelitian dan penulisan laporan. Rincian ilustrasi alur pengerjaan aplikasi penulis ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Perancangan Aplikasi Penentu Jarak Terdekat

1. Studi Literatur

Hal pertama yang dilakukan penulis sebelum melakukan perancangan model simulasi adalah dengan mengumpulkan informasi, menganalisis masalah, fungsi, dan kebutuhan sistem. Pada tahap ini penulis memulai penelitian dengan studi literatur untuk mengkaji informasi lebih dalam serta mendapatkan teori mengenai sistem penentu jalur terpendek di UPI Bumi Siliwangi menggunakan Algoritma A*.

2. Pengumpulan Data

Langkah selanjutnya adalah dengan melakukan pengumpulan data. Salah satu permasalahan orang baru di lingkungan UPI yaitu dikarenakan luasnya lingkungan UPI maka orang baru akan kebingungan dalam melakukan pengambilan rute dari satu tempat ke tempat lain di sekitar lingkungan UPI. Oleh karena itu, dengan adanya permasalahan ini penulis ingin membuat aplikasi berbasis web ini untuk membantu orang baru dalam menentukan pengambilan rute terdekat dari satu tempat ke tempat lain di lingkungan UPI. Aplikasi yang akan penulis buat akan dikhususkan untuk para pejalan kaki, dikarenakan mayoritas pengguna jalan di lingkungan UPI adalah pejalan kaki.

3. Perancangan dan Pembuatan

Pada tahap ini penulis dapat memulai merancang sistem penentu jalur terpendek di UPI Bumi Siliwangi menggunakan Algoritma A* ini dengan beberapa tambahan maupun perubahan komponen agar aplikasi penulis dapat berjalan sesuai keinginan penulis. Dengan melakukan uji coba simulasi algoritma terlebih dahulu maka penulis dapat menentukan kode program beserta algoritma yang akan digunakan dalam pembuatan aplikasi ini. Dalam melakukan perancangan aplikasi ini, penulis menggunakan bahasa pemrograman JavaScript. Pembuatan aplikasi dimulai dari perancangan *software* terlebih dahulu, dengan menentukan apa saja yang akan digunakan untuk *back-end* maupun *front-end* pada aplikasi penulis yang akan dibuat. Selain itu penulis harus melakukan analisis kebutuhan *software*. Aplikasi yang akan dibuat oleh penulis adalah aplikasi berbasis web yang akan digunakan untuk para pengguna *mobile* maupun desktop yang baru menginjakkan kaki dan masih asing di lingkungan UPI untuk menentukan pengambilan jalur terdekat dari satu tempat ke tempat lain di lingkungan UPI. Selanjutnya melakukan pengembangan, penulis mengembangkan program dengan membangun code program dan *user interface* serta menguji coba program. Penulis

merancang *front-end* terlebih dahulu untuk bagian code programnya, kemudian penulis merancang *back-end* untuk bagian *user interface*-nya.

4. Pengujian dan Evaluasi

Tahap berikut adalah tahap untuk menguji beserta mengevaluasi sistem yang dibuat. Keberhasilan sistem diuji dengan kriteria yang diharapkan sebelumnya. Jika terdapat kesalahan pada sistem (*error*, *bug*, dan lain-lain), maka penulis harus melakukan evaluasi untuk memperbaiki sistem agar sistem dapat berjalan sesuai dengan yang diinginkan.

5. Dokumentasi

Pada tahap terakhir ini penulis melakukan dokumentasi penelitian berupa penulisan laporan dengan menjelaskan hasil dari penelitian yang telah dilakukan sesuai dengan diagram alir yang sudah ditentukan.

3.2 Alat dan Bahan Penelitian

Dalam pelaksanaan studi, peneliti menggunakan berbagai jenis perangkat untuk melakukan penelitian, yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Selain itu, dalam penelitian ini, data yang digunakan merupakan *dataset* sebagai bahan penelitian.

Program yang digunakan untuk melakukan simulasi perbandingan algoritma adalah sebuah aplikasi yang bernama Pathfinding Visualization. Aplikasi tersebut ditulis oleh seorang programmer yang bernama CodeNoodles. Aplikasi ini dirancang dengan bahasa pemrograman JavaScript dan boleh digunakan oleh siapa saja. Setelah melakukan perbandingan algoritma, penulis melakukan perbandingan aplikasi penulis dengan Google Maps.

Algoritma yang akan digunakan pada saat simulasi adalah Algoritma Breadth-First Search, Algoritma Greedy Best-First Search, dan Algoritma A*. Algoritma Dijkstra tidak termasuk karena pada simulasi ini setiap titik memiliki bobot yang sama, hal tersebut mengartikan bahwa Algoritma Dijkstra dan Algoritma BFS akan mendapatkan hasil yang sama. Percobaan dilakukan sebanyak sepuluh kali pada setiap algoritma lalu dirata-ratakan hasilnya untuk mendapatkan kesimpulan simulasi. Algoritma yang diimplementasikan pada simulasi dapat melakukan *pathfinding* secara vertikal dan horizontal. Setiap perjalanan dua titik yang bersebelahan secara vertikal maupun horizontal memiliki *cost* sebanyak satu.

3.2.1 Alat Penelitian

Perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan pada penelitian ini adalah sebagai berikut:

1. Perangkat Keras (*Hardware*)
 - a. Komputer
 - b. Monitor
 - c. Papan tombol (*keyboard*)
 - d. Tetikus (*mouse*)
2. Perangkat Lunak (*Software*)
 - a. JavaScript
 - b. Express JS
 - c. MongoDB
 - d. Mongo Atlas
 - e. MongoDB Compass
 - f. Mongoos
 - g. Mapbox
 - h. Node JS
 - i. React JS
 - j. Vite JS
 - k. Mapbox GL

3.2.2 Spesifikasi Perangkat Penulis

Diperlukan juga sebuah device untuk membuat dan menguji hasil dari penelitian ini, berikut adalah spesifikasi dari perangkat desktop dan juga *mobile* yang akan digunakan oleh penulis yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Penulis

Perangkat Desktop	Perangkat <i>Mobile</i>
-------------------	-------------------------

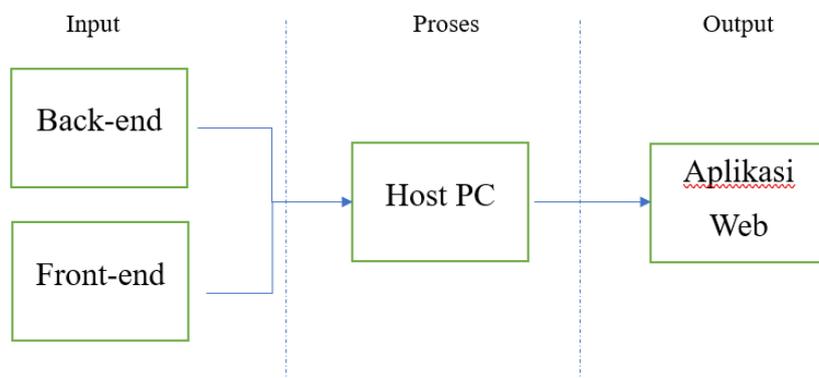
<ul style="list-style-type: none"> • NVIDIA GTX 950M 4GB VRAM • 16 GB RAM • Intel Core i7 6700HQ @ 2.60GHz <p style="text-align: center;">Perangkat Desktop</p>	<ul style="list-style-type: none"> • Android Version 7.1.1 • Octa Core 2.5 GHz Cortex-A53 • 6GB RAM <p style="text-align: center;">Perangkat <i>Mobile</i></p>
<ul style="list-style-type: none"> • Windows 10 Home 64-bit 	<ul style="list-style-type: none"> • Model CPH1727

3.3 Metode Penelitian

Dalam penelitian ini, metode penelitian mencakup diskusi mengenai langkah-langkah dan tahapan perancangan yang akan digunakan. Metode penelitian ini terdiri dari beberapa komponen yang akan dibahas, seperti perancangan sistem model, perancangan *database*, dan juga pengujian sistem.

3.4 Prinsip Kerja Sistem

Prinsip kerja dari sistem pencari rute terpendek dari satu tempat ke tempat lain di lingkungan UPI Bumi Siliwangi ini direpresentasikan ke dalam diagram block seperti yang tertera pada Gambar 3.2.



Gambar 3.2 Diagram Blok Prinsip Kerja Sistem

Pada Gambar 3.2 Diagram Blok Prinsip Kerja Sistem ini menerangkan bahwa sistem aplikasi penentu rute dari satu tempat ke tempat lain di lingkungan UPI Bumi Siliwangi ini terbagi menjadi tiga bagian yaitu bagian *input*, *proses*, dan *output*. Pertama pada bagian *input* terdapat *back-end* dan *front-end*. Semua data-data penting seperti data kode, data gambar, data tempat atau lokasi, dan data deskripsi disimpan di bagian *input* ini. Pada bagian *input* ini para

pengguna tidak diperbolehkan untuk mengaksesnya, bagian *input* ini hanya dapat diakses oleh penulis saja. Kedua pada bagian proses terdapat Host PC. Pada bagian proses ini penulis menggunakan sebuah *software* yang bernama Replit untuk menghosting aplikasinya agar dapat digunakan oleh para pengguna yang membutuhkan. Replit menyimpan data-data yang terdapat pada *front-end* dan juga *back-end* di Cloud, yang artinya Replit akan mengakses semua data yang terdapat pada *front-end* maupun *back-end* secara otomatis dan akan langsung menampilkan aplikasi penulis yang siap digunakan. Ketiga pada bagian *output* terdapat aplikasi web. Pada bagian *output* ini ditujukan untuk para pengguna, pengguna tidak perlu melakukan *log-in* untuk mengakses aplikasi penulis. Cara untuk mengakses aplikasi penulis cukup mudah, dikarenakan aplikasi penulis berbentuk aplikasi web, para pengguna hanya perlu mengakses link atau tautan dengan menggunakan sebuah aplikasi browser.

3.5 Perancangan Dasar Sistem Pencari Rute Terdekat

Untuk melakukan perancangan dasar sistem pencari rute terdekat dari satu tempat ke tempat lain di lingkungan Universitas Pendidikan Indonesia (UPI) Bumi Siliwangi ini dilakukan dengan cara merancang terlebih dahulu bagian *front-end*, *database*, beserta *back-end* terlebih dahulu. Perancangan aplikasi ini menggunakan bahasa pemrograman JavaScript. Aplikasi yang digunakan pada *back-end* penulis yaitu, Express JS, MongoDB, Mongo Atlas, MongoDB Compass, Algoritma A*, Mongoos, Mapbox, Node JS. Sedangkan aplikasi yang digunakan pada *front-end* penulis yaitu, React JS, Vite JS, Mapbox GL. Model aplikasi yang ditulis oleh penulis hanya berfokus pada pengguna yang berjalan kaki saja.

Untuk menemukan sebuah jalur terpendek (yaitu jalur yang akan ditempuh oleh para pejalan kaki), kita harus memastikan bahwa rute yang akan diambil oleh algoritma adalah rute yang berbeloknya paling sedikit. Untuk ini, algoritma lebih memilih jalur lurus daripada jalur yang memerlukan belokan. Hal ini dapat diimplementasikan dengan mengurangi jarak yang diperlukan untuk pindah ke simpul berikutnya, jika simpul sebelumnya, simpul saat ini, dan simpul berikutnya berada dalam satu garis lurus. Pengimplementasian tersebut divisualisasikan pada Gambar 3.3 di bawah ini.

Hal pertama yang dilakukan penulis adalah mengumpulkan data, data berupa gambar foto gedung yang terdapat pada UPI Bumi Siliwangi, data deskripsi gedung, dan data koordinat. Data gambar diambil oleh penulis menggunakan kamera penulis untuk memfoto gedung-gedung yang ada di UPI Bumi Siliwangi. Sedangkan data deskripsi gedung, penulis mengambilnya di internet. Dan data koordinat diambil dari peta UPI Bumi Siliwangi. Contoh data gambar gedung dapat dilihat pada Gambar 3.5.



Gambar 3.5 Contoh Data Foto Gedung

Setelah selesai mengambil semua foto gedung-gedung yang ada di lingkungan UPI Bumi Siliwangi, langkah selanjutnya adalah dengan memasukkan data foto-foto tersebut, deskripsi gedung-gedung tersebut, beserta koordinat gedung-gedung tersebut ke dalam *back-end*. Untuk mendapatkan rute terdekat dari satu tempat ke tempat lainnya, maka dibutuhkanlah data rute jalan tersebut, maka buatlah kode untuk hal tersebut. Kemudian buatlah kode memberikan tampilan jalur rute terdekat dari satu tempat ke tempat lainnya yang koordinat gedungnya sudah dimasukkan ke dalam coding.

3.5.2 Perancangan Dasar *Front-End* Sistem

Berikut adalah langkah-langkah untuk melakukan perancangan dasar *front-end* dari sistem pencari rute terdekat dari satu tempat ke tempat lain di lingkungan Universitas Pendidikan Indonesia (UPI) Bumi Siliwangi ini:

1. Membuat proyek menggunakan React.

Muhammad Habil Fachriansyah, 2023

APLIKASI PENENTU RUTE TERDEKAT DI LINGKUNGAN UNIVERSITAS PENDIDIKAN INDONESIA BUMI SILIWANGI MENGGUNAKAN ALGORITMA A*

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Hal pertama yang harus dilakukan oleh penulis adalah dengan menginisialisasi React. Untuk melakukan hal tersebut, yang harus dilakukan adalah dengan membuka *command prompt* dan mengetik sebuah *command* sama seperti yang terdapat pada Gambar 3.6.

```
C:\Users\Asus>npm create vite@latest frontend -- --template react
Need to install the following packages:
  create-vite@4.4.1
Ok to proceed? (y) y

Scaffolding project in C:\Users\Asus\frontend...

Done. Now run:

  cd frontend
  npm install
  npm run dev
```

Gambar 3.6 Instalasi React

2. Tampilkan map pada aplikasi React.

Langkah selanjutnya adalah menampilkan peta pada aplikasi React, data peta yang penulis gunakan adalah peta milik Mapbox. Mapbox menyediakan data peta mereka sebagai media *open-source* yang bisa digunakan oleh siapa saja.

3. Membuat dan tampilkan *polygon* perbatasan UPI Bumi Siliwangi.

Agar peta UPI Bumi Siliwangi dapat terlihat dengan jelas, maka penulis membuat perbatasan pada ujung peta lingkungan UPI Bumi Siliwangi. Hal ini akan memudahkan para pengguna yang tidak mengenal atau baru saja mengenal lingkungan UPI Bumi Siliwangi dan aplikasi akan langsung menyediakan tampilan peta UPI Bumi Siliwangi saat para pengguna baru saja mengakses aplikasi ini.

4. Menggunakan *sample* data sebanyak 5 gedung untuk menampilkan marker.

Sebagai pondasi, penulis mengetes aplikasi ini dengan memasukkan 5 buah data gedung-gedung yang terdapat pada lingkungan UPI Bumi Siliwangi sebelum memasukkan sisanya. Hal ini perlu dilakukan untuk memastikan apakah aplikasi penulis berjalan dengan baik atau tidak. Jika aplikasi penulis sudah berjalan dengan benar, maka hal yang harus dilakukan penulis adalah dengan memasukkan data gedung-gedung sisanya.

5. Membuat *option box* untuk mencari rute, data gedung akan dikirimkan ke *back-end*.

Untuk memudahkan para pengguna aplikasi, penulis menyediakan *option box* untuk memilih dua gedung sebagai titik awal dan titik akhir. Setelah memilih kedua gedung sebagai titik awal dan titik akhir, hal selanjutnya yang harus dilakukan pengguna adalah dengan memijit tombol “cari rute” maka rute akan langsung ditampilkan.

3.5.3 Perancangan *Database* Sistem

Berikut adalah langkah-langkah untuk melakukan perancangan *database* dari sistem pencari rute terdekat dari satu tempat ke tempat lain di lingkungan Universitas Pendidikan Indonesia (UPI) Bumi Siliwangi ini:

1. Inisiasi *database* menggunakan MongoDB.

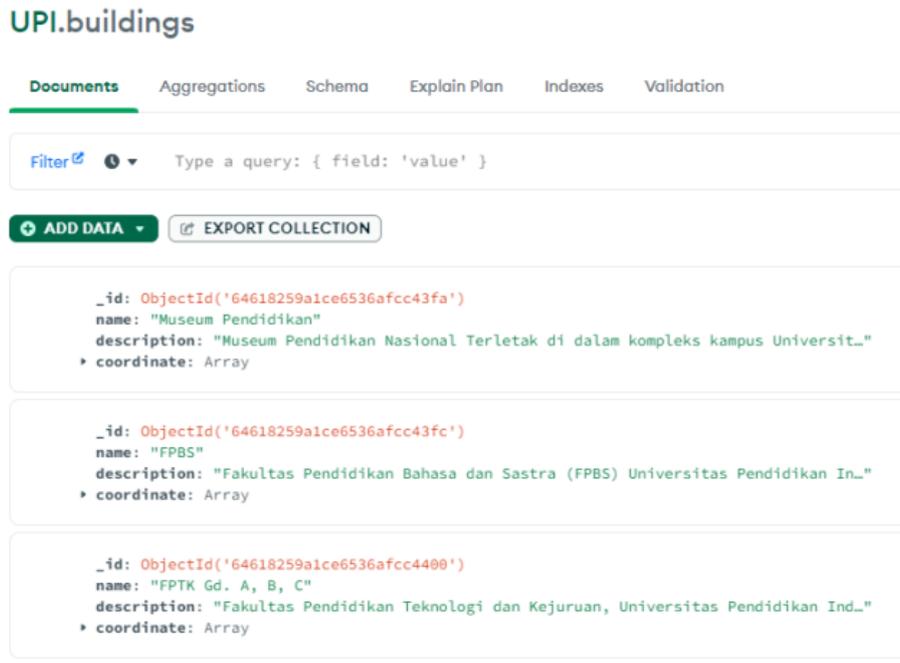
Hal pertama yang harus dilakukan dalam perancangan *database* sistem adalah dengan menginisialisasi MongoDB, MongoDB digunakan untuk menyimpan, mengelola, dan mengakses data dengan cepat dan efisien.

2. Mempersiapkan *database*, *collection*, dan *document* di MongoDB Atlas.

Hal ini dilakukan untuk mengkonfigurasi *database*, *collection*, dan *document* agar proses pengaturan atau penyesuaian suatu sistem atau perangkat agar berfungsi sesuai dengan kebutuhan atau preferensi.

3. Masukkan 5 *sample* data ke dalam *database*, *field*-nya adalah *id*, *name*, *description*, *coordinate*[lng, lat].

Hal ini dilakukan agar aplikasi dapat menampilkan koordinat, nama gedung, dan deskripsi gedung pada *front-end* seperti pada Gambar 3.7



Gambar 3.7 *Sample Data* MongoDB Atlas

4. *Database* siap diuji.

Setelah ketiga langkah tadi selesai dilaksanakan, maka *database* sudah siap untuk diuji.

3.5.4 Perancangan Dasar *Back-End* Sistem

Berikut adalah langkah-langkah untuk melakukan perancangan dasar *back-end* dari sistem pencari rute terdekat dari satu tempat ke tempat lain di lingkungan Universitas Pendidikan Indonesia (UPI) Bumi Siliwangi ini:

1. Membuat project menggunakan express js.

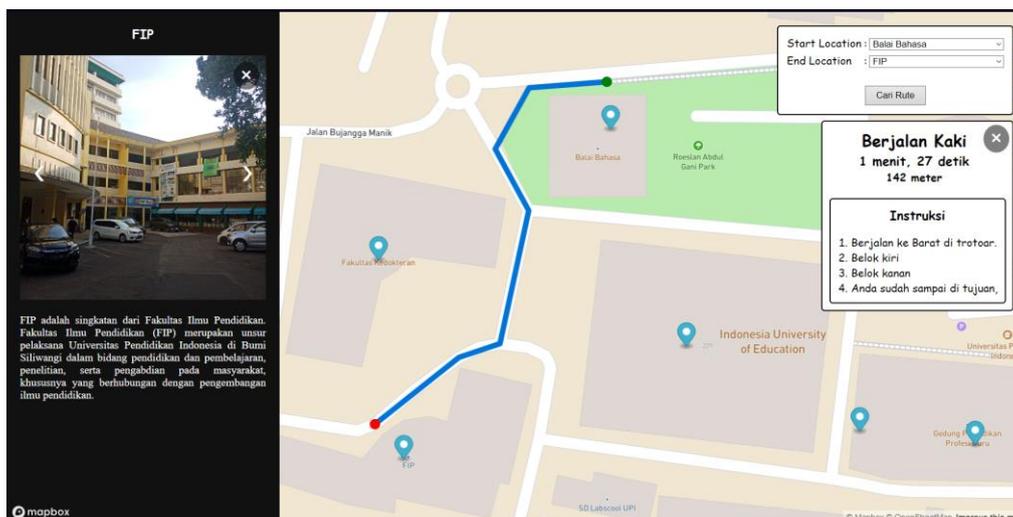
Instalasi folder *back-end* pada komputer desktop dengan memasukkan kode yang terdapat pada Gambar 3.8.

```
C:\Users\Asus>express --no-view backend
```

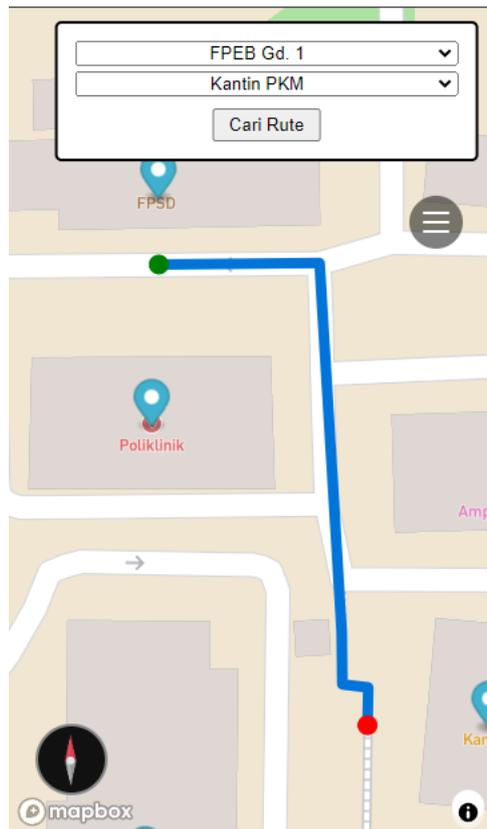
Gambar 3.8 *Command Prompt* Express js

2. Koneksi Express js dengan MongoDB Atlas menggunakan Mongoose.
3. Masukkan beberapa sample gambar ke dalam folder /public/images di root express js.

1. Kembali ke *front-end*, *fetch* data menggunakan *axios* ke *back-end* sesuai dengan rutanya, kemudian ubah *sample* 5 data yang di-*hard coding* menjadi dinamis (menggunakan *back-end*), lalu cek jika tidak terjadi *error*, maka data bisa ditambah, setelah itu *fetch* ke GET REQUEST ke `/api/buildings/direction?start=${awal_lokasi}&destination=${akhir_lokasi}` untuk mendapatkan koordinat rute pada dua gedung, kemudian buat kode rute agar dapat ditampilkan di web, juga instruksi, dan terakhir jika semua sudah benar (tidak ada *error*), maka penambahan semua data bisa dilakukan.
2. kembali ke *database*, cek penamaan data semua gedung agar hanya menggunakan huruf, angka, dan spasi, setelah itu masukkan semua data yang sudah dicek.
3. kembali ke *back-end*, copy semua images ke dalam folder `/public/images` perhatikan nama agar tidak menggunakan simbol.
4. Kembali ke *front-end*, data sudah dimasukkan semua, cek jika tidak terjadi *error*, jika sudah tidak ada *error*, menggunakan Responsive Web Design untuk kompatibel di setiap device, setelah itu cek terakhir kali untuk memastikan tidak ada *error*.



Gambar 3.10 Contoh Tampilan Aplikasi Desktop



Gambar 3.11 Contoh Tampilan Aplikasi *Mobile*

Aplikasi sudah selesai dan siap untuk diuji coba. Lihat Gambar 3.10 untuk contoh tampilan aplikasi desktop, dan Gambar 3.11 untuk contoh tampilan aplikasi *mobile*.

