

# Aplikasi Navigasi Supermarket Berbasis Web Menggunakan Algoritma *Dijkstra*

Muhammad Rayhan Yovi dan Kemal Ade Sekarwati

Informatika, Fakultas Teknologi Industri,  
Universitas Gunadarma  
Jl. Margonda Raya No. 100 Pondok Cina Depok Jawa Barat  
E-mail : yovihan@gmail.com, ade@staff.gunadarma.ac.id

## Abstrak

Supermarket merupakan salah satu tempat belanja yang populer bagi masyarakat modern karena menyediakan berbagai macam produk kebutuhan sehari-hari. Pada supermarket dengan tata letak yang luas dan kompleks, pelanggan sering kali mengalami kesulitan dalam menemukan barang-barang yang dibutuhkan yang mengakibatkan pengalaman berbelanja menjadi kurang efisien. Masalah ini menjadi lebih signifikan jika pelanggan harus mencari banyak produk di lokasi yang berbeda sehingga dibutuhkan solusi untuk membantu pelanggan dalam menentukan rute berbelanja terpendek agar pengalaman belanja menjadi lebih efisien. Aplikasi Navigasi *Supermarket* dirancang dengan tujuan memberikan solusi untuk masalah tersebut. Dengan menggunakan algoritma *Dijkstra*, aplikasi ini dapat memuat rute terpendek untuk mendapatkan seluruh barang yang diinginkan oleh pengguna. Aplikasi ini merupakan aplikasi berbasis website dan dikembangkan menggunakan bahasa pemrograman *JavaScript* dan *framework React*. Aplikasi ini dibuat menggunakan metode SDLC (*System Development Life Cycle*) dengan beberapa tahapan, yaitu tahap perencanaan, tahap desain, tahap implementasi dan pembuatan kode program, dan tahap uji coba. Algoritma *Dijkstra* berhasil diimplementasikan pada Aplikasi Navigasi *Supermarket* dan menyediakan rute terpendek untuk penggunaannya. Berdasarkan hasil pengujian terhadap 25 responden, aplikasi dapat berjalan pada berbagai jenis perangkat dengan berbagai resolusi.

**Kata kunci** : Algoritma *Dijkstra*, Navigasi, Rute Terpendek, Supermarket, Website.

## Pendahuluan

Supermarket merupakan salah satu tempat belanja yang populer bagi masyarakat modern. Supermarket menawarkan berbagai macam barang dan produk yang memenuhi kebutuhan sehari-hari [1]. Walaupun masyarakat sudah dapat berbelanja secara daring melalui ponsel dari rumah, berbelanja secara langsung ke supermarket tetap memiliki daya tarik sendiri. Selain dapat melihat produk secara langsung, pembeli juga dapat terekspos ke barang-barang lain yang sekiranya mereka butuhkan.

Daya tarik dari berbelanja di supermarket tersebut hadir dengan masalah lain. Pengunjung sering kali kesulitan dalam mencari barang yang diinginkan. Keadaan ini disebabkan oleh ukuran supermarket yang besar, tata letak yang kompleks, dan penempatan produk yang sejenis pada rak-rak yang terpisah.

Dalam upaya untuk meningkatkan penjualan dan interaksi pengunjung dengan berbagai produk, banyak supermarket yang menempatkan barang-

barang yang sejenis pada rak yang berjauhan agar pengunjung melihat barang-barang lain selama berbelanja [2]. Hal tersebut didukung oleh teori Dr. Paul Harrison dari Deakin University [3]. Teori tersebut membahas tentang tata letak barang di supermarket yang diatur sedemikian rupa agar pembeli berbelanja lebih lama dan dapat melihat-lihat produk lain selagi mereka mencari produk yang mereka inginkan [3]. Hal ini sering kali menyebabkan pengunjung harus berkeliling dan mencari barang-barang yang diinginkan dengan susah payah. Pengunjung harus menghabiskan waktu dan tenaga hanya untuk menemukan beberapa barang, terlebih lagi jika daftar belanjanya panjang.

Fenomena tersebut mendorong penelitian dan pengembangan aplikasi bernama “Aplikasi Navigasi Supermarket” yang dirancang agar dapat membantu pengunjung supermarket dalam mencari barang yang diinginkan dengan lebih efisien dari segi waktu maupun tenaga. Aplikasi ini dirancang menggunakan algoritma *Dijkstra* yang bekerja dengan menelusuri semua simpul secara sistem-

atis hingga menemukan jalur dengan bobot terendah. Dalam konteks Aplikasi Navigasi Supermarket, tata letak supermarket direpresentasikan sebagai grid di mana setiap simpul mewakili titik lokasi di dalam supermarket, dan setiap sisi merepresentasikan jarak antar simpul. Pemilihan algoritma Dijkstra didasarkan pada keunggulannya dalam menemukan rute terpendek yang memberikan solusi untuk meningkatkan pengalaman berbelanja pelanggan, yang selama ini sering kali terganggu oleh kesulitan dalam menemukan produk tertentu di supermarket.

## Metode Penelitian

Penelitian ini dilakukan dengan menggunakan metode studi pustaka, yaitu serangkaian kegiatan mencari dan mengumpulkan informasi melalui referensi dari internet dan buku yang berkaitan dengan penulisan. Penelitian ini juga menggunakan metode *System Development Life Cycle* (SDLC), yaitu metode yang digunakan untuk merancang, membangun, dan memelihara suatu sistem dengan tahapan sebagai berikut:

1. Tahap Perencanaan Pada tahap ini dilakukan proses pengumpulan data untuk membuat aplikasi berbasis web menggunakan framework React dengan bahasa pemrograman JavaScript. Pada tahap ini juga dilakukan proses persiapan hardware dan software, serta analisis algoritma yang akan digunakan.
2. Tahap Desain Pada tahap ini, dilakukan perancangan model dari sistem, seperti membuat diagram *Unified Modeling Language* (UML) dan membuat rancangan tampilan *User Interface* (UI) dari aplikasi menggunakan Figma.
3. Tahap Implementasi dan Pembuatan Kode Program Pada tahap ini, rancangan aplikasi yang sebelumnya dibuat dengan menggunakan Figma diimplementasikan ke dalam bentuk kode program berbahasa pemrograman JavaScript dengan framework React menggunakan Visual Studio Code.
4. Tahap Uji Coba Tahap ini merupakan tahap pengujian aplikasi yang telah dibuat dengan menggunakan berbagai browser yang ada di ponsel maupun komputer pengguna untuk memastikan program berjalan dengan baik pada berbagai device.

## Hasil dan Pembahasan

Aplikasi Navigasi Supermarket merupakan aplikasi berbasis web yang diperuntukkan untuk memudahkan pengguna dalam menentukan rute tercepat

untuk mendapatkan semua barang saat berbelanja di supermarket. Aplikasi ini dikembangkan menggunakan bahasa pemrograman JavaScript dan framework React. Data barang supermarket disimpan dalam bentuk file JSON.

### Analisis Kebutuhan Pengguna

Aplikasi Navigasi Supermarket dikembangkan dan dirancang untuk dapat memudahkan pengguna dalam mencari rute terpendek saat berbelanja di supermarket. Aplikasi Navigasi Supermarket harus dapat mengakomodir kebutuhan pengguna seperti: Menampilkan peta supermarket, menampilkan daftar barang yang dijual pada supermarket, dan menunjukkan rute terpendek yang bisa diikuti untuk mendapatkan seluruh barang yang akan dibeli.

### Analisis Kebutuhan Hardware dan Software

Untuk menunjang kebutuhan dalam membuat aplikasi dibutuhkan hardware dan software agar aplikasi dapat berjalan secara optimal. Berikut ini merupakan hardware dan software yang digunakan untuk merancang dan membuat aplikasi :

1. Analisis kebutuhan perangkat keras (hardware)

Hardware yang digunakan dalam pembuatan Aplikasi Navigasi Supermarket sebagai berikut:

- (a) Laptop Asus TUF FX504 GE
- (b) Processor : Intel Core i7-8750H CPU 2.20GHz
- (c) Memory : RAM 8 GB
- (d) Storage : SSD 512 MB

2. Analisis kebutuhan perangkat lunak (software)

Software yang digunakan dalam pembuatan Aplikasi Navigasi Supermarket sebagai berikut:

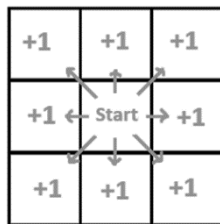
- (a) Operating System Windows 11 (64 bit)
- (b) Visual Studio Code V 1.81.1
- (c) React V 18.2
- (d) . Node JS V 20.9.0

### Analisis Algoritma

Algoritma yang digunakan pada aplikasi ini adalah algoritma Dijkstra, sebuah algoritma yang digunakan untuk mencari jalur terpendek dari satu titik ke semua titik lain dalam sebuah graf berbobot [4]. setiap edge memiliki bobot yang mewakili jarak atau biaya. Algoritma Dijkstra membutuhkan poin-poin berikut untuk dapat bekerja:

1. Graf berbobot yang terdiri dari simpul (*vertex*) dan sisi (*edge*) yang memiliki nilai bobot sebagai representasi jarak antar dua simpul.
2. Sebuah titik simpul (*node*) untuk menjadi titik awal dari rute jalur terpendek.
3. Kumpulan titik simpul (*nodes*) yang menjadi tujuan akhir dari rute jalur terpendek.
4. Sebuah array atau *priority queue* yang digunakan untuk menyimpan jarak sementara dari proses pencarian jalur terpendek

Pada aplikasi ini, node pada peta digambarkan dalam bentuk grid seperti Gambar 1



Gambar 1: Visualisasi Jarak Antar Node

### Analisis Kebutuhan Data

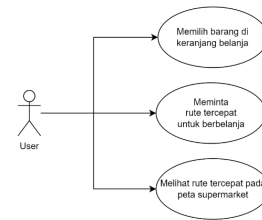
Analisis kebutuhan data merupakan analisis terhadap kebutuhan data apa saja yang dibutuhkan oleh Aplikasi Navigasi Supermarket. Berikut ini merupakan analisis kebutuhan data dari Aplikasi Navigasi Supermarket:

1. Titik koordinat barang.
2. Nama barang.
3. Titik koordinat letak obstacle atau hambatan-hambatan seperti lemari, rak, tembok, ataupun pilar pada supermarket.

### Rancangan Diagram Unified Model Language

Diagram *Unified Model Language* (UML) digunakan untuk memudahkan visualisasi dalam proses perancangan sistem. Ada Beberapa jenis diagram UML yang digunakan pada proses perancangan Aplikasi Navigasi Supermarket seperti use case diagram, activity diagram, class diagram, dan sequence diagram [5].

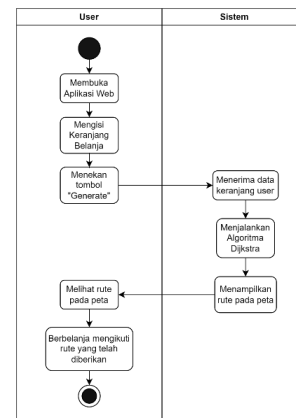
*Use case* diagram adalah diagram UML yang menggambarkan interaksi antara pengguna dan sistem [6]. *Use case diagram* pada Aplikasi Navigasi Supermarket dapat dilihat pada Gambar 2.



Gambar 2: Use Case Diagram Aplikasi Navigasi Supermarket

Pada *use case diagram*, yang berperan sebagai aktor adalah pengunjung supermarket yang akan berbelanja. Pengunjung akan dihadapkan langsung dengan denah supermarket dan daftar barang yang dijual pada supermarket tersebut. Jika sudah memilih barang yang akan dibeli, pengunjung dapat menekan tombol untuk meminta sistem melakukan generate rute tercepat untuk mendapatkan seluruh barang yang akan dibeli. Sistem akan menampilkan rute tersebut melalui denah supermarket yang ditampilkan pada halaman utama.

*Activity diagram* merupakan model proses yang terjadi dari sebuah sistem. Tujuan dari activity diagram yaitu untuk menjelaskan alur interaksi antara pengguna dan sistem pada aplikasi [7]. Activity diagram Aplikasi Navigasi Supermarket dapat dilihat pada Gambar 3.

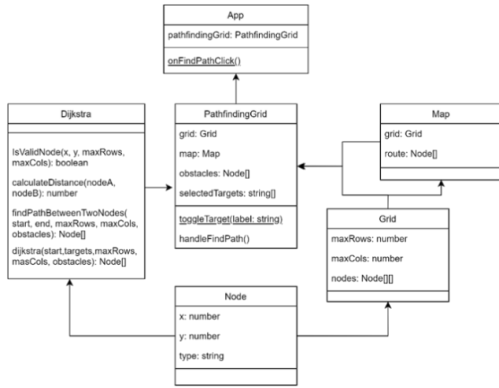


Gambar 3: Activity Diagram Aplikasi Navigasi Supermarket

Aplikasi ini membantu pengguna berbelanja di supermarket dengan lebih efisien. Pengguna dapat melihat daftar barang yang tersedia dan menambakkannya ke keranjang belanja. Sistem kemudian akan menghitung rute tercepat untuk mengunjungi semua item di keranjang belanja dan menampilkannya di peta. Pengguna dapat mengikuti rute ini untuk berbelanja dengan lebih cepat dan mudah.

*Class diagram* merupakan gambaran, struktur, dan hubungan antara class, atribut, dan metode dari setiap objek pada aplikasi [8]. *Class diagram* juga menggambarkan interaksi dan hubungan dari setiap kelas yang ada dalam suatu sistem. *Class diagram* Aplikasi Navigasi Supermarket dapat dilihat

pada Gambar 4.



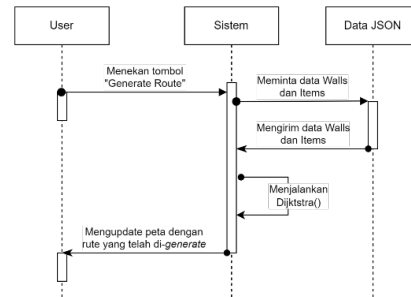
Gambar 4: Class Diagram Aplikasi Navigasi Supermarket

Terdapat 6 objek utama yang menjalankan aplikasi ini:

1. *Class App* menjadi fungsi utama dari aplikasi ini yang menerima property dari class PathfindingGrid. Terdapat method onFindPathClick() pada class ini yang berfungsi untuk menampilkan hasil perhitungan kepada pengguna.
2. *Class Node* berfungsi untuk menentukan jarak antara node pada aplikasi. Pada aplikasi ini, jarak antar node adalah satu poin untuk segala arah.
3. *Class Pathfinding Grid* merupakan class yang memiliki property *grid*, *map*, *obstacles*, dan *selectedTargets*. Properties tersebut menampilkan informasi untuk menentukan ukuran peta, rute yang tersedia, penghalang, dan target atau kumpulan target yang dituju. Terdapat *method toggleTarget* dan *handleFindPath()* yang berfungsi untuk memvisualisasikan rute terpendek yang dapat dilalui.
4. *Class Map* dan *Grid* merupakan class yang berfungsi sebagai objek peta pada aplikasi. Class Grid menentukan ukuran dari grid yang akan digunakan dan class Map menentukan node yang tersedia untuk dilalui. Kedua class ini memiliki fungsi yang serupa, namun perlu dipisah untuk keberlanjutan aplikasi ke depannya jika ingin menambahkan denah untuk supermarket yang ukurannya berbeda.
5. *Class Dijkstra* merupakan class yang berfungsi untuk melakukan kalkulasi rute terpendek dari barang-barang yang dipilih. Terdapat beberapa method di dalamnya. *Method isValidNode()* berfungsi untuk menentukan apakah node tersebut dapat dilalui atau tidak. *Method findPathBetweenNodes()* berfungsi untuk mencari jalan penghubung antara dua

node. *Method calculateDistance()* berfungsi untuk mengukur jarak antar node. *Method dijkstra()* berfungsi untuk mengkalkulasi kembali dan menentukan rute terpendek antara pilihan jalur yang dapat dilalui antara dua node.

6. Sequence diagram merupakan gambaran untuk menampilkan interaksi antar objek-objek dalam sistem berdasarkan waktu pelaksanaannya [9]. Terdapat tiga objek utama pada aplikasi ini, yaitu pengguna, sistem, dan database. Database yang digunakan aplikasi ini berbentuk data JSON sehingga tidak memerlukan autentikasi khusus pada proses request maupun response data. Sequence diagram pada aplikasi ini dapat dilihat pada Gambar 5.

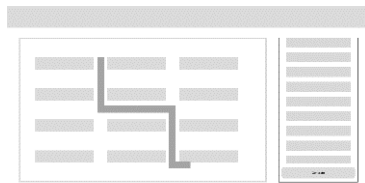


Gambar 5: 5 Sequence Diagram Aplikasi Navigasi Supermarket

Diagram tersebut menggambarkan proses yang terjadi ketika aplikasi dijalankan. Pengguna menekan tombol untuk melakukan generate rute setelah memilih daftar belanjaan. Sistem akan meminta data obstacles dan target points pada database. Database akan mengirimkan kembali data tersebut kepada sistem. Sistem akan terus menjalankan *function Dijkstra* untuk menemukan rute tercepat yang dapat dilalui oleh pengguna. Setelah rute didapatkan, sistem akan menampilkan rute tersebut kepada pengguna dan pengguna dapat berbelanja mengikuti rute yang telah disediakan.

## Rancangan Tampilan

Rancangan tampilan dibuat dengan tujuan menggambarkan desain tampilan website yang akan dibuat. Tampilan halaman utama dari Aplikasi Navigasi Supermarket dapat dilihat pada Gambar 6.

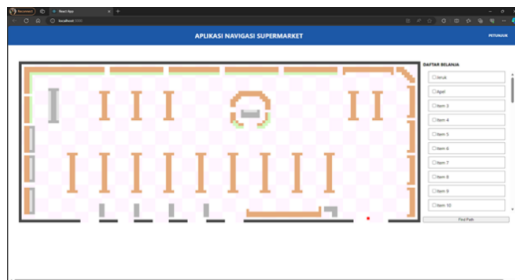


Gambar 6: Tampilan Halaman Utama Aplikasi Navigasi Supermarket

Pengguna akan langsung dihadapkan kepada peta dan daftar belanja. Aplikasi ini memiliki tampilan yang didesain sesederhana mungkin untuk memudahkan pengguna dalam memahami cara kerja aplikasi ini.

### Pengembangan Tampilan Halaman Utama

Proses pengembangan tampilan halaman utama dapat dilakukan dengan mengubah kode HTML dan CSS dari halaman. Framework React memungkinkan developer untuk mengembangkan halaman menjadi komponen terpisah yang dapat dipanggil di komponen lain. Pada halaman utama, penulis menggunakan dua komponen bernama *Navbar* dan *PathFindingGrid*. Komponen *Navbar* merupakan komponen yang berisi *Navigation Bar* dan komponen *PathFindingGrid* merupakan komponen yang berisi kode-kode untuk menampilkan denah, daftar barang yang dijual, dan implementasi logic program. Tampilan halaman utama dari Aplikasi Navigasi Supermarket dapat dilihat pada Gambar 7.



Gambar 7: Tampilan Halaman Utama Aplikasi Navigasi Supermarket

### Pengembangan dan Penerapan Algoritma Dijkstra

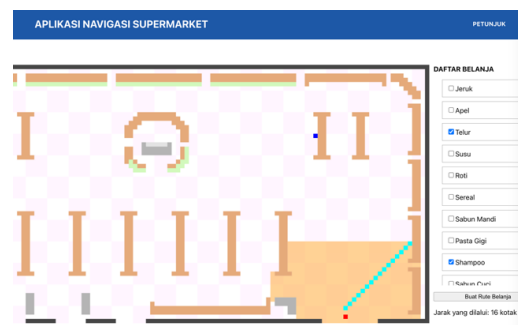
Algoritma Dijkstra yang dikembangkan untuk Aplikasi Navigasi Supermarket menggunakan pendekatan graf untuk menilai bobot dari jarak antar node dalam sebuah grid yang memiliki hambatan-hambatan tertentu. Aplikasi ini memiliki beberapa fungsi utama untuk menjalankan algoritma, seperti *isValidNode* untuk validasi node, *CalculateDistance* untuk menghitung jarak antara dua titik, dan *findPathBetweenTwoNodes* untuk mencari jalur antara dua node dengan mempertimbangkan adanya penghalang dalam grid.

Fungsi Dijkstra merupakan implementasi dari algoritma Dijkstra dalam aplikasi ini. Algoritma ini mencari jalur terpendek antara sebuah node awal dan sekelompok node target dalam lingkungan berbasis grid dengan mempertimbangkan adanya rintangan. Proses algoritma ini melibatkan permutasi node-node target, pencarian jalur terpendek, dan penghitungan total jarak.

Pada tampilan aplikasi, halaman utama menampilkan peta supermarket dalam bentuk grid melalui komponen *PathfindingGrid*. Sebelum menampilkan peta, komponen ini menetapkan parameter peta, titik awal, lokasi gambar supermarket, dan menyimpan koordinat barang yang akan dibeli. Komponen ini juga memanfaatkan hook *useState* untuk menyimpan state dari item-item yang dipilih oleh pengguna, termasuk rute yang dihasilkan oleh algoritma Dijkstra.

Berbagai fungsi di dalam komponen *PathfindingGrid*, seperti *toggleTarget* untuk menampilkan lokasi barang yang dipilih dan *handleFindPath* untuk menjalankan algoritma Dijkstra, membantu pengguna dalam menavigasi peta supermarket. Selain itu, *getNodeColor* memberikan warna pada setiap grid pada peta sesuai dengan peran dan statusnya. Setelah menjalankan fungsi-fungsi tersebut, komponen *PathfindingGrid* menghasilkan dokumen HTML yang menampilkan visualisasi dari algoritma Dijkstra pada Aplikasi Navigasi Supermarket. Secara teoritis, algoritma akan menghasilkan rute terpendek untuk mengunjungi seluruh node yang dipilih karena algoritma Dijkstra merupakan algoritma greedy yang mengunjungi seluruh node sekitar hingga menemukan node target. Rute yang dimunculkan berdasar kepada node-node yang telah dikunjungi dalam proses scanning.

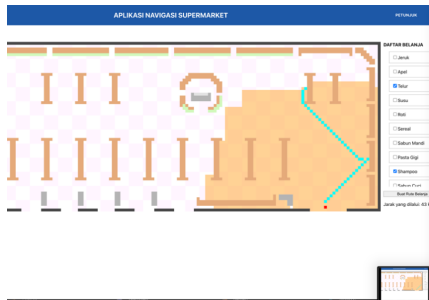
Visualisasi pada Aplikasi Navigasi Supermarket bekerja dengan melakukan scanning ke segala arah untuk menemukan node target terdekat. Area yang dipindai ditandai dengan warna oranye seperti pada Gambar 8.



Gambar 8: Visualisasi proses scanning pada Aplikasi Navigasi Supermarket

Algoritma kemudian dijalankan kembali setelah menemukan node terdekat dengan menjadikan node tersebut sebagai node awal. Algoritma akan melakukan scanning kembali ke berbagai arah hingga menemukan node selanjutnya. Proses terse-

but dilakukan secara berulang hingga seluruh node yang dipilih berhasil dikunjungi. Setelah node target ditemukan, algoritma akan memunculkan rute yang ditandai dengan garis berwarna biru seperti pada Gambar 9.

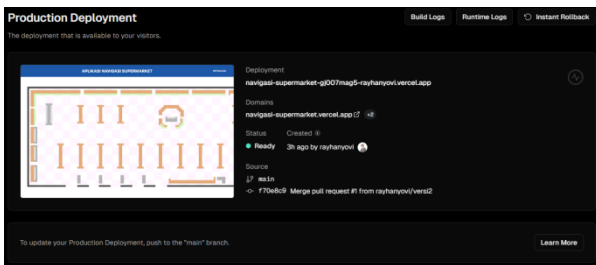


Gambar 9: Visualisasi Rute Aplikasi Navigasi Supermarket

### Deployment Aplikasi

Deployment adalah proses untuk membuat sebuah aplikasi atau website menjadi dapat diakses secara langsung oleh pengguna di Internet. Aplikasi Navigasi Supermarket dilakukan melalui platform Vercel. Proses deployment ini dimulai dengan membuat repository di Github sebagai tempat penyimpanan kode aplikasi.

Setelah proses deployment selesai, informasi status deployment akan ditampilkan, termasuk URL aplikasi yang telah dilakukan deployment. Status deployment Aplikasi Navigasi Supermarket dapat dilihat pada Gambar 10.



Gambar 10: Status Deployment pada Vercel

### Uji Coba

Tahap uji coba merupakan tahap pengujian pada aplikasi web. Uji coba aplikasi ini dilakukan dengan metode *Black Box Testing*. Pengujian dengan metode ini berfokus pada perspektif end-user aplikasi [10].

Aplikasi ini masih dalam tahap prototipe sehingga data yang digunakan untuk tahap uji coba masih berupa data dummy dan tidak merepresentasikan data pada pusat perbelanjaan sesungguhnya.

Uji coba dibagi menjadi beberapa bagian diantaranya uji coba fungsi, uji coba perangkat, dan uji coba pengguna. Tujuan dilakukannya uji coba

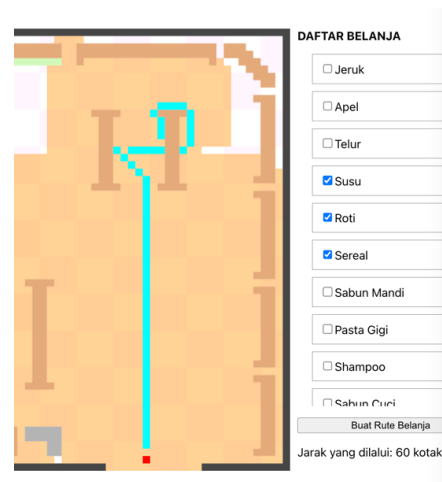
adalah untuk mengetahui apakah aplikasi berjalan sesuai ketentuan atau tidak.

Uji coba fungsi merupakan uji coba fungsionalitas fitur-fitur yang terdapat pada aplikasi web. Skenario uji coba fungsi pertama adalah checkbox pada daftar belanjaan dapat dipilih dan menampilkan titik lokasi barang pada peta. Hasil yang diharapkan dapat dilihat pada Gambar 11.



Gambar 11: Skenario memilih barang pada daftar belanja

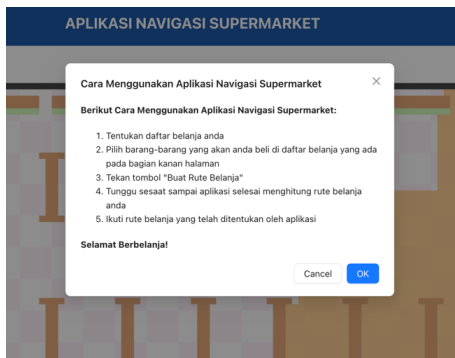
Selanjutnya, pengguna akan menekan tombol "Buat rute belanja" untuk menjalankan algoritma. Setelah algoritma selesai berjalan, aplikasi akan memvisualisasikan rute untuk mendapatkan seluruh barang yang telah dipilih. Rute yang didapat divisualisasikan dengan bentuk garis berwarna biru. Hasil yang diharapkan dapat dilihat pada Gambar 12.



Gambar 12: Skenario generate dan visualisasi rute belanja

Aplikasi ini juga memiliki fitur petunjuk untuk memudahkan pengguna dalam mencari tau cara penggunaan aplikasi. Skenario uji coba fungsi untuk fitur ini adalah dengan menekan tombol petunjuk pada sudut kanan atas. Sebuah popup message

yang berisi cara penggunaan aplikasi akan muncul. Hasil yang diharapkan untuk scenario ini dapat dilihat pada Gambar 13.



Gambar 13: Skenario melihat petunjuk cara penggunaan aplikasi

Hasil uji coba fungsi pada aplikasi dapat dilihat pada Tabel 1.

Tabel 1: Hasil Uji Coba Fungsi Aplikasi Navigasi Supermarket

No	Fungsi Fitur	Skenario Uji coba	Hasil yang diharapkan	Keterangan
1	Memilih barang pada menu "Daftar Belanja"	Menekan <i>checkbox</i> pada bagian kiri barang.	<i>Checkbox</i> berhasil ditekan dan warna <i>grid</i> barang berubah menjadi warna biru pada peta.	Sukses
2	Melakukan <i>generate</i> untuk mendapatkan rute berbelanja.	Menekan tombol "Buat Rute Belanja"	Rute belanja berhasil didapatkan dan ditampilkan pada peta dengan visualisasi berwarna biru muda.	Sukses
3	Melihat petunjuk cara penggunaan aplikasi.	Menekan tombol "Petunjuk" pada <i>navbar</i> aplikasi.	Menampilkan <i>popup menu</i> yang berisi tata cara penggunaan aplikasi.	Sukses

Uji coba perangkat merupakan pengujian aplikasi web di beberapa perangkat yang berbeda guna mengetahui apakah aplikasi web mempunyai kelancaran dan tampilan yang sama. Hasil pengujian perangkat dapat dilihat di Tabel 2

Tabel 2: Hasil Uji Coba Perangkat

No	Nama Perangkat	Spesifikasi	Hasil Pengujian
1	Asus TUF FX 504 GE	Intel Core i7 8 <sup>th</sup> Gen, 8GB RAM, Windows 11, 1920x1080px resolution	Aplikasi berjalan dengan baik dan sesuai dengan resolusi layar
2	Apple iPhone XR	Apple A12 Bionic (10 nm), 3GB RAM, IOS 16.6.1, 828 x 1792px resolution	Aplikasi berjalan dengan baik dan sesuai dengan resolusi layar
3	Samsung Galaxy Tab S6 Lite	Snapdragon 732G, 4GB RAM, Android 13, 1200x2000px resolution	Aplikasi berjalan dengan baik dan sesuai dengan resolusi layar

Uji Coba pengguna atau *User Acceptance Test (UAT)* merupakan pengujian aplikasi terhadap calon pengguna aplikasi seperti mahasiswa, dan masyarakat umum yang biasa berbelanja di supermarket untuk mencari tau tingkat kepuasan responden terhadap pengalaman berbelanja di supermarket. Responden akan memberikan penilaian terhadap pertanyaan terkait fungsionalitas dan tampilan dari aplikasi. Sebaran poin dan klasifikasi pilihan untuk jawaban responden terdapat pada Tabel 3.

Tabel 3: Klasifikasi Berdasarkan Bobot

No	Keterangan	Presentase Nilai
1	STS : Sangat Tidak Setuju	1
2	TS : Tidak Setuju	2
3	N: Netral	3
4	S : Setuju	4
5	SS : Sangat Setuju	5

Terdapat lima pernyataan yang diajukan kepada 25 orang responden. Setiap jawaban memiliki bobot poin yang berbeda. Jawaban dari responden akan menentukan tingkat persetujuan pengguna terhadap Aplikasi Navigasi Supermarket.

Untuk mendapatkan hasil dari pengujian pengguna, hasil kuesioner diolah menggunakan rumus 1.

$$Y = \frac{\sum(N.F)}{S} \times 100\% \quad (1)$$

dengan Y adalah nilai akhir, N adalah nilai setiap jawaban, F adalah frekuensi jawaban, dan S adalah skor maksimal.

Hasil uji coba terhadap 25 orang responden dapat dilihat pada Tabel 4.

Tabel 4: Hasil uji coba terhadap responden

No	Pernyataan	Jawaban	N	F	X	Y
1	Fitur mencari rute Aplikasi Navigasi Supermarket mudah untuk digunakan	SS	5	16	116	92,8%
		S	4	9		
		N	3	0		
		TS	2	0		
		STS	1	0		
2	Tampilan Aplikasi Navigasi Supermarket mudah untuk dipahami.	SS	5	13	109	87,2%
		S	4	8		
		N	3	4		
		TS	2	0		
		STS	1	0		
3	Aplikasi Navigasi Supermarket memudahkan pengalaman berbelanja anda.	SS	5	11	101	80,8%
		S	4	4		
		N	3	10		
		TS	2	0		
		STS	1	0		
4	Rute yang dihasilkan oleh Aplikasi Navigasi Supermarket cukup optimal dan efisien.	SS	5	13	109	87,2%
		S	4	8		
		N	3	4		
		TS	2	0		
		STS	1	0		
5	Anda akan merekomendasikan Aplikasi Navigasi Supermarket ke rekan atau keluarga anda.	SS	5	14	110	88%
		S	4	7		
		N	3	4		
		TS	2	0		
		STS	1	0		

## Penutup

Penelitian yang dilakukan telah berhasil mengembangkan Aplikasi Navigasi Supermarket yang dapat menghasilkan rute terpendek dalam berbelanja dan memvisualisasikannya kepada pengguna. Berdasarkan hasil uji coba aplikasi, skor kepuasan pengguna secara keseluruhan mendapat skor 87,2%. Aplikasi ini juga berhasil meningkatkan efisiensi waktu pengguna dalam berbelanja. Hal tersebut dapat dibuktikan oleh skor spesifik kepuasan pengguna sebesar 87,2% terhadap efisiensi rute yang dihasilkan oleh Aplikasi Navigasi Supermarket.

Saran yang dapat dipertimbangkan untuk penelitian selanjutnya mengenai implementasi algoritma Dijkstra pada Aplikasi Navigasi Supermarket untuk menentukan rute terpendek dalam berbelanja di supermarket yaitu terkait optimisasi algoritma. Model grid yang digunakan pada penulisan ini berjalan cukup lambat, terlebih jika dijalankan pada sisi client. Algoritma menentukan rute terpendek dapat dilakukan dengan mempersempit jalur yang dapat dilalui dan melakukan caching untuk rute yang pernah dilalui agar sistem tidak perlu menghitung kembali rute yang pernah dilalui.

## Daftar Pustaka

- [1] Anonim, Peraturan Menteri Perdagangan Republik Indonesia Nomor 23 Tahun 2021, Direktorat Jendral Perdagangan Dalam Negeri, Departemen Perdagangan Republik Indonesia Jakarta, 2021
- [2] T. L. Urban, "The effect of one-way aisles on retail layout", Science Direct Computers & Industrial Engineering Vol. 168, Amsterdam, 2022.
- [3] Paul Harrison, "Supermarket Psychology - Entrances, layout and shelving", Paul Harrison YouTube channel, Geelong, 2009.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", The MIT Press, 2019
- [5] Rendi Juliarto, "Apa itu UML? Beserta Pengertian dan Contohnya", Dicoding, Bandung, 2021
- [6] Rendi Juliarto, "Contoh Use Case Diagram Lengkap dengan Penjelasannya.", Dicoding, Bandung, 2021
- [7] Rendi Juliarto, "Apa itu Activity Diagram? Beserta Pengertian, Tujuan, Komponen", Dicoding, Bandung, 2021.
- [8] Rony Setiawan, "Memahami Class Diagram Lebih Baik", Dicoding, Bandung, 2021
- [9] Rony Setiawan, "Apa Itu Sequence Diagram dan Contohnya", Dicoding, Bandung, 2021.
- [10] Rony Setiawan, "Black Box Testing Untuk Menguji Perangkat Lunak", Dicoding, Bandung, 2021