

APLIKASI PENGIRIMAN PESANAN MAKANAN MENGGUNAKAN ALGORITMA DJIKSTRA

Luqman Affandi¹, Ridwan Rismanto², Muhammad Mustofa Firmansyah³

^{1,2,3} Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang
¹laffandi@polinema.ac.id, ²rismanto@polinema.ac.id, ³tofa_muz12@yahoo.com

Abstrak

Pelanggan restoran Prada masih sering terlambat dalam mendapat respon ketika melakukan layanan pesan antar. Layanan pesan antar di restoran tersebut menggunakan bantuan sosial media seperti whatsapp, line dan BBM, oleh karena itu dibutuhkan seorang admin yang selalu sedia untuk membalas percakapan ketika ada pesanan pelanggan. Selain itu, kurir juga sering kebingungan untuk menentukan urutan rute pengiriman saat ada beberapa pemesanan makanan yang harus diantar pada satu waktu. Pada Skripsi ini, Peneliti membangun sebuah aplikasi dua aplikasi yang dapat memenuhi kebutuhan dari pelanggan dan kurir. Aplikasi yang pertama berfungsi untuk melayani pemesanan makanan untuk pelanggan. Aplikasi yang kedua berfungsi untuk menerima pesanan pelanggan, mengatur pesanan mana yang siap diantar, dan melihat rute pengiriman makanan. Dari hasil ujicoba, gabungan dari layanan google api dan algoritma djikstra dinilai mampu untuk memberikan hasil yang cukup akurat untuk menentukan beberapa urutan jalur pengiriman. selain itu, aplikasi juga mampu menangani pemesanan pelanggan.

Kata kunci : jalur terpendek, urutan rute, restoran prada, algoritma djikstra

1. Latar Belakang

Prada Fried Chicken merupakan salah satu restoran cepat saji yang sedang berkembang di Malang. Prada Fried Chicken menyajikan Ayam Goreng lokal yang dapat dinikmati untuk semua kalangan, dengan menawarkan menu ayam goreng, hamburger, pasta dan sebagainya. Beberapa fasilitas pelayanan kepada konsumennya, yaitu makan di tempat (Dine-In), dibawa pulang (Take-Away), dan layanan pesan antar (Delivery Order). Layanan pesan antar, yaitu konsumen memesan secara online atau melalui telepon kemudian pesanan diantar menuju tempat konsumen.

Proses Pemesanan yang dilakukan melalui online yaitu dengan menggunakan aplikasi sosial media seperti Whatsapp, Line dan Blackberry Messenger. Pelanggan menuliskan daftar pesannya dan mengirimkan lokasi dimana pelanggan berada, lalu pihak prada menanggapi pemesanan tersebut dan mengantarkan pesannya. Setelah pesanan diterima oleh pelanggan, kurir memberikan kuitansi mengenai berapa biaya pemesanan sekaligus pengiriman makanan tersebut, lalu pelanggan membayar tagihan tersebut.

Selama ini penggunaan aplikasi chatting untuk delivery order dinilai kurang efektif karena kurangnya tingkat kompleksitas. Admin diharuskan untuk memberitahukan daftar menu setiap pelanggan melakukan pemesanan melalui aplikasi sosial media chatting, sehingga pelanggan baru (khususnya) tidak mengetahui daftar menu sebelum dia bertanya kepada admin tentang menu. Hal tersebut menyebabkan admin diharuskan untuk melayani banyak pesanan dalam satu waktu, baik saat berkomunikasi dengan pelanggan melalui chatting maupun saat melihat jalur pengiriman yang menampilkan beberapa lokasi pelanggan dalam satu waktu.

Pencarian rute terpendek yang saat ini digunakan oleh Restoran Prada yaitu dengan memanfaatkan fasilitas share lokasi pada smartphone melalui aplikasi sosial media chatting seperti Whatsapp, Line dan Blackberry Messenger. Admin menanyakan pesanan pelanggan dan meminta pelanggan untuk melakukan share lokasi. Dari situ, admin mendapatkan lokasi pelanggan beserta rute terpendek dari fasilitas google map. Kekurangannya lokasi peta dan rute tidak bisa menampilkan lebih dari satu lokasi pelanggan dalam satu waktu, maksudnya adalah admin hanya dapat peta lokasi pelanggan dan rute yang ditempuh untuk masing-masing pelanggan, sehingga tidak tersedia rute menuju dua atau lebih pelanggan sekaligus.

Restoran Prada membutuhkan aplikasi yang dapat memberikan kemudahan bagi pelanggan dan pemilik restoran. Keuntungan bagi pelanggan adalah kemudahan untuk melakukan pemesanan, melihat daftar menu. Sedangkan kemudahan yang didapat oleh pihak prada adalah data pemesanan tersusun rapi, dapat melayani beberapa pemesanan dalam satu waktu tanpa perlu menanggapi atau memberi balasan pada pelanggan, petugas pengiriman makanan dapat mengetahui urutan rute pengiriman terpendek dari beberapa pemesanan.

Berdasarkan hasil uraian masalah diatas, maka perlu dibangun Aplikasi Pengiriman Pesanan Makanan Menggunakan Algoritma Dijkstra.

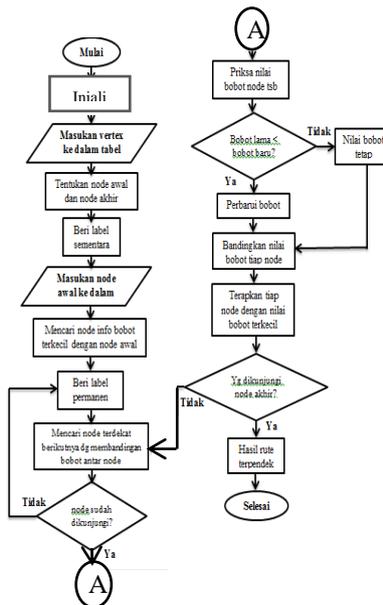
2. Landasan Teori

2.1. Algoritma Dijkstra

Pada dasarnya, algoritma ini merupakan salah satu bentuk algoritma greedy. Algoritma ini termasuk algoritma pencarian graf yang digunakan untuk menyelesaikan masalah lintasan terpendek dengan satu sumber pada sebuah graf yang tidak memiliki cost sisi negatif, dan menghasilkan sebuah pohon lintasan terpendek. Algoritma ini sering digunakan pada routing. Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah [1].

Algoritma Dijkstra merupakan salah satu bentuk algoritma greedy. Algoritma ini termasuk algoritma pencarian graf yang digunakan untuk menyelesaikan masalah lintasan terpendek dengan satu sumber pada sebuah graf yang tidak memiliki cost sisi negatif dan menghasilkan sebuah pohon lintasan terpendek [1].

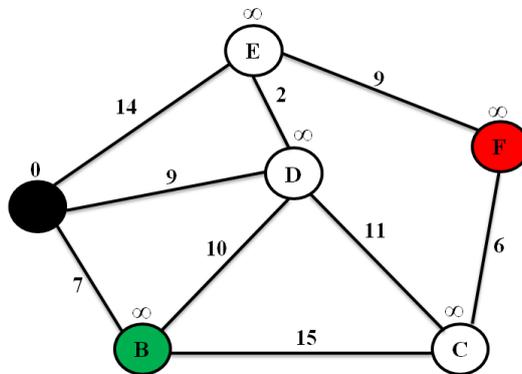
Algoritma Dijkstra menggunakan adjacent list untuk merepresentasikan sebuah jaringan. Secara garis besar algoritma dijkstra membagi semua node menjadi dua, kemudian dimasukkan ke dalam tabel yang berbeda, yaitu tabel permanen dan tabel temporal. Tabel permanen berisi node awal dan node-node yang telah melalui proses pemeriksaan dan labelnya telah diubah dari temporal menjadi permanen. Tabel temporal berisi node-node yang berhubungan dengan node pada tabel permanen [2].



Gambar 1. Diagram Alir Algoritma Dijkstra [3]

Pemilihan rute dalam algoritma Dijkstra dilakukan dengan Best First Search (BFS), best first search merupakan kombinasi dari metode depth first search dan breadth first search dimana pencarian diperbolehkan mengunjungi node pada level lebih rendah jika node pada level lebih tinggi memiliki nilai heuristik lebih buruk. Dari uraian di atas algoritma Dijkstra dapat digambarkan ke dalam diagram alir seperti Gambar 1.

Sebuah graf berarah yang berbobot (weighted directed graph) G dan sumber vertex dalam G dan V merupakan himpunan semua vertices dalam graph G , dalam hal ini graf berarah yang berbobot dan sumber vertex dapat disimpulkan sebagai inputan dari algoritma Dijkstra. Setiap sisi dari graf ini adalah pasangan vertices (u, v) yang melambangkan hubungan dari vertex u ke vertex v . Himpunan semua tepi disebut E [4].



Gambar 2. Graf Penentuan Jalur Terpendek [5]

Tabel 1. Penyelesaian Graf Dengan Menggunakan Algoritma Dijkstra

Langkah	Vertex Sumber	Jarak Vertex Lain	Jalur Pada Graf
1.	A	A → B = 7 A → D = 9 A → E = 14	
2.	B	B → C 7 + 15 = 22 B → D 7 + 10 = 17	
3.	D	D → C 9 + 11 = 20 D → E 9 + 2 = 11	
4.	E	E → F 11 + 9 = 20	

Untuk menghitung bobot dari semua sisi dapat menggunakan fungsi pada rumus 1.

$$W: E \rightarrow [0, \infty]$$

1

Contoh bentuk pengimplementasian algoritma Dijkstra pada graf untuk menentukan jalur terpendek dari node A ke F pada Gambar 2 dan penyelesaiannya pada Tabel 1.

3. Metodologi

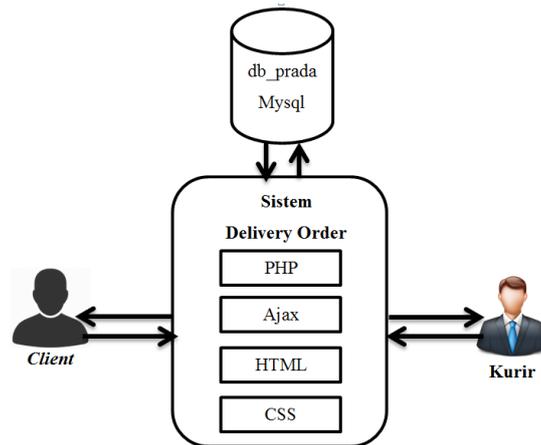
Metode yang digunakan dalam pengembangan sistem delivery order adalah metode waterfall. Alasan menggunakan metode ini adalah karena metode waterfall melakukan pendekatan secara sistematis dan berurutan dalam membangun suatu sistem. Proses metode waterfall yaitu pada pengerjaan dari suatu sistem dilakukan secara berurutan. Sistem yang dihasilkan berkualitas baik, dikarenakan pelaksanaannya secara bertahap sehingga tidak terfokus pada tahapan tertentu.

Tahapan dari metode waterfall adalah:

1. Pengumpulan Data
2. Perancangan Sistem
3. Implementasi
4. Pengujian Sistem

3.1. Perancangan Sistem

Dalam merancang sistem ini, peneliti membuat rancangan arsitektur sistem dari sistem yang dibangun. Berikut adalah detail rancangan dari arsitektur sistem tersebut.



Gambar 3 Perancangan arsitektur sistem.

Client memesan makanan melalui pada website Sistem Delivery Order. Data client diproses oleh sistem untuk disimpan pada database. Data yang tersimpan di database akan diproses kembali oleh sistem untuk ditampilkan pada halaman kurir. Kurir mengakses data tersebut melalui website. Data yang diterima kurir berupa data pesanan client yang ditampilkan dalam bentuk daftar pemesanan. kurir juga dapat melihat detail pesanan dari client. Selain itu, kurir dapat mengatur pesanan yang siap untuk dikirim. Pesanan yang siap dikirim akan diproses oleh sistem, agar dihitung menggunakan algoritma djikstra, sehingga sistem dapat menampilkan urutan rute pengiriman berdasarkan pesanan yang siap dikirim.

3.2. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) dapat digunakan sebagai alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. Juga sebagai alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Pengguna dalam sistem ini adalah *Client* dan *Kurir*. Dibawah ini merupakan *Data Flow Diagram* (DFD) dari Sistem *Delivery Order* Pada Aplikasi Pengiriman Pesanan Makanan Menggunakan Algoritma Dijkstra yang sedang berjalan adalah.

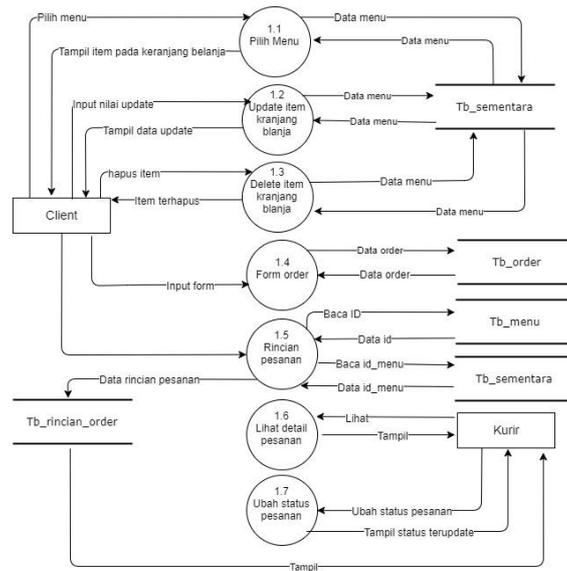
1. Data Flow Diagram level 0

Diagram level 0 merupakan diagram yang menguraikan proses apa saja yang dilakukan user terhadap sistem dan umpan balik dari sistem terhadap user. Untuk lebih jelasnya, lihat pada gambar 5.



Gambar 5. Data flow diagram level 0.

2. Data Flow Diagram level 1

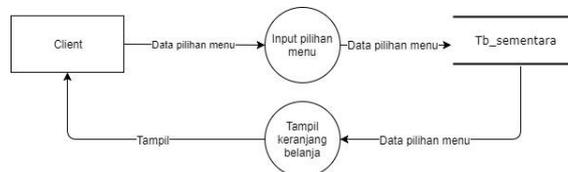


Gambar 6. Data flow diagram level 1

Diagram level 1 merupakan diagram yang mengguraikan setiap proses dari diagram level 0. Pada diagram level 1 akan diuraikan alur yang lebih jelas dan lebih detail dari level sebelumnya. Untuk lebih jelasnya, lihat pada gambar 6.

3. Data Flow Diagram level 2.1

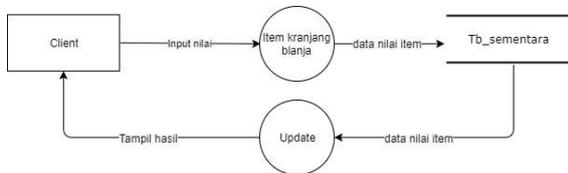
Diagram level 2.1 merupakan diagram yang mengguraikan setiap proses dari diagram level 1 pada proses 1.1. pada diagram ini diuraikan lebih rinci mengenai proses pilih menu. Untuk lebih jelasnya lihat pada gambar 7.



Gambar 7. Data flow diagram level 2.1 proses pilih menu.

4. Data Flow Diagram level 2.2

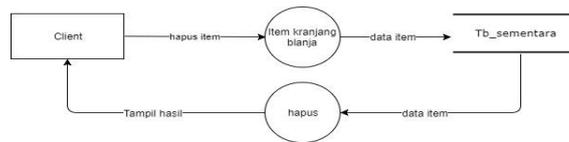
Diagram level 2.2 merupakan diagram yang mengguraikan setiap proses dari diagram level 1 pada proses 1.2. pada diagram ini diuraikan lebih rinci mengenai proses update nilai item. Untuk lebih jelasnya lihat pada gambar 8.



Gambar 8. Data flow diagram level 2.2 proses update nilai item.

5. Data Flow Diagram level 2.3

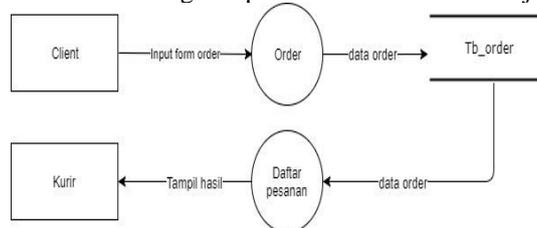
Diagram level 2.3 merupakan diagram yang mengguraikan setiap proses dari diagram level 1 pada proses 1.3. pada diagram ini diuraikan lebih rinci mengenai proses hapus item. Untuk lebih jelasnya lihat pada gambar 9.



Gambar 9. Data flow diagram level 2.3 proses hapus item.

6. *Data Flow Diagram level 2.4*

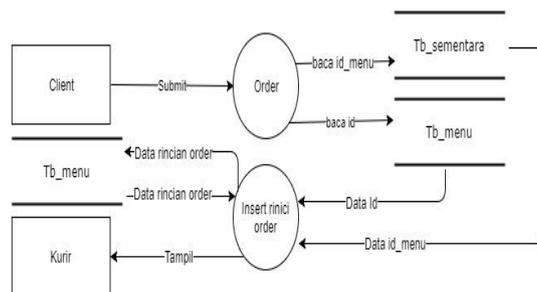
Diagram level 2.4 merupakan diagram yang mengguraikan setiap proses dari diagram level 1 pada proses 10 pada diagram ini diuraikan lebih rinci mengenai proses order. Untuk lebih jelasnya lihat pada gambar 10.



Gambar 10. Data flow diagram level 2.4 proses order.

7. *Data Flow Diagram level 2.5*

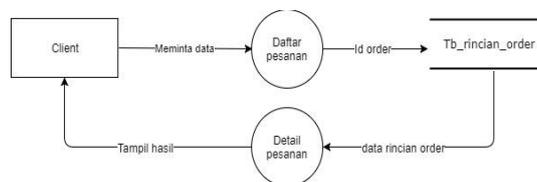
Diagram level 2.5 merupakan diagram yang mengguraikan setiap proses dari diagram level 1 pada proses 1.5 pada diagram ini diuraikan lebih rinci mengenai proses rincian pesanan. Untuk lebih jelasnya lihat pada gambar 11.



Gambar 11. Data flow diagram level 2.5 proses rincian pesanan.

8. *Data Flow Diagram level 2.6*

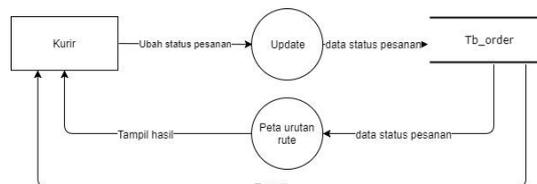
Diagram level 2.6 merupakan diagram yang mengguraikan setiap proses dari diagram level 1 pada proses 1.6 pada diagram ini diuraikan lebih rinci mengenai proses detail pesanan. Untuk lebih jelasnya lihat pada gambar 12.



Gambar 12. Data flow diagram level 2.6 Lihat pesanan.

9. *Data Flow Diagram level 2.7*

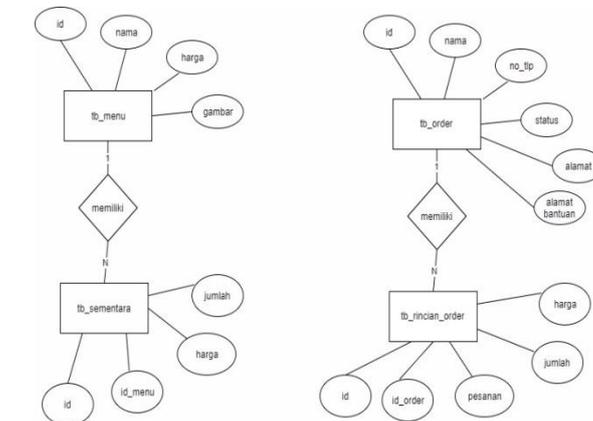
Diagram level 2.7 merupakan diagram yang mengguraikan setiap proses dari diagram level 1 pada proses 1.7 pada diagram ini diuraikan lebih rinci mengenai proses ubah pesanan. Untuk lebih jelasnya lihat pada gambar 13.



Gambar 13. Data flow diagram level 2.7 proses ubah status pesanan.

3.3. ERD

Entity Relationship Diagram (ERD) digunakan untuk memodelkan struktur data dan hubungan (relasi) antar data menggunakan beberapa simbol / notasi. Model Entity Relationship Diagram (ERD) yang digunakan pada sistem delivery order adalah sebagai berikut.



Gambar 14. Entity Relationship Diagram (ERD) sistem dilevery order

4. Implementasi Sistem

Implementasi antarmuka atau implementasi desain user interface berpacu pada desain antarmuka yang ada pada sub-sub-bab 4.2.3. Di bawah ini dijelaskan implementasi dari antarmuka dari Aplikasi Pengiriman Pesanan Makanan Menggunakan Algoritma Dijkstra.

4.1. Halaman Utama Client

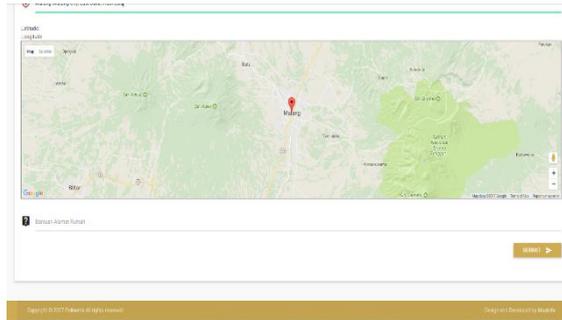


Gambar 15. Implementasi Antarmuka Halaman Utama Client

Merupakan halaman utama client yang menampilkan daftar menu. Bagian kanan adalah slidebar keranjang belanja yang isinya bertambah jika pelanggan memilih menu yang dipesan pada gambar daftar menu. Pesanan pada keranjang belanja dapat di update jumlahnya atau hapus pesanan, jika client tidak jadi memesan item tersebut. Tombol order digunakan jika client sudah selesai memilih pesanan, pesanan tersebut diproses dan client diarahkan menuju halaman berikutnya untuk mengisi form order.

4.2. Halaman Utama Order

Gambar 16. Implementasi Antarmuka Halaman Form Order Bagian Satu



Gambar 17. Implementasi Antarmuka Halaman Form Order Bagian Dua

Halaman ini merupakan halaman order. Halaman ini menampilkan rincian pesanan yang telah diisi oleh user pada sebelumnya. Pada halaman ini, client diharuskan untuk mengisikan informasi untuk keperluan order yaitu nama, nomor telepon, tanggal pemesanan, lokasi dan alamat bantuan rumah. Lokasi merupakan alamat pelanggan yang diolah menggunakan google api service sehingga dapat memunculkan lokasi pelanggan pada peta. Sedangkan bantuan alamat rumah adalah alamat lengkap pelanggan yang disimpan seperti catatan yang berguna jika google api service tidak dapat memberikan lokasi yang akurat, maka kurir dapat mencarinya dengan cara bertanya pada penduduk sekitar tentang alamat yang dicatat pada bantuan alamat rumah.

4.3. Halaman Utama Kurir

ID Pesanan	Nama Pesanan	Tanggal Pesanan	Detail Pesanan	Tanggal Pesan
3	tsbt	2017-08-01	Status	✓
4	slf	2017-08-02	Status	✓
5	stsa	2017-08-02	Status	✓
6	uadu	2017-08-02	Status	✓

Gambar 18. Implementasi Antarmuka Halaman Utama Kurir

Halaman utama kurir berisikan tentang daftar pesanan. Untuk melihat detail pesanan, kurir dapat menekan tombol link pada kolom detail. Tombol rute pengiriman mengarahkan kurir menuju halaman rute pengiriman.

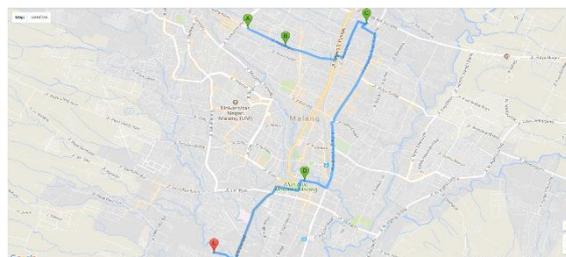
4.4. Halaman Halaman Detail Pesanan

Pesanan	Jumlah	Harga
Burger	1	\$3.00
Kemang	2	\$7
Apur	2	\$16.00
TOTAL	5	\$15.00

Gambar 19. Implementasi Antarmuka Halaman Detail Pesanan Untuk Kurir

Halaman Detail Pesanan menampilkan rincian dari apa yang dipesan oleh pelanggan. Pesanan yang ditampilkan berdasarkan id dari link yang dipilih. Jadi misal kurir ingin melihat apa saja yang dipesan oleh budi, pada halaman daftar pesanan, kurir harus memilih link pada kolom detail pesanan milik budi.

4.5. Halaman Rute Pengiriman



Gambar 20. Implementasi Antarmuka Halaman Rute Pengiriman Untuk Kurir

Halaman rute pengiriman merupakan halaman yang menampilkan urutan rute dari beberapa tujuan pengiriman yang diolah menggunakan algoritma djikstra. Banyaknya titik tujuan didapatkan dari banyaknya pesanan yang memiliki status sedang diproses. Titik A merupakan titik awal lokasi Prada Fried Chicken, titik B di Kalpataru, C di Blimbing, D di Alun-alun dan E di Sukun. Masing-masing titik tersebut adalah titik tujuan.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan analisis dan perancangan, implementasi serta pengujian sistem Aplikasi Pengiriman Pesanan Makanan Menggunakan Algoritma Dijkstra, maka didapatkan kesimpulan sebagai berikut.

1. Dengan membandingkan penerapan algoritma djikstra yang dihitung menggunakan excel dan melalui sistem, didapatkan hasil perhitungan yang sama, sehingga sistem ini dinilai berhasil mencari urutan rute terpendek dari beberapa titik tujuan dengan akurat.
2. Penerapan yang menggabungkan layanan google api yang difilter menggunakan algoritma djikstra dapat menghasilkan pencarian rute urut dari beberapa tujuan sekaligus.
3. Aplikasi yang dibangun sudah mampu mencatat pesanan dari pelanggan dan data pesanan tersebut ditampilkan pada halaman kurir, dimana kurir sudah bisa mengatur status pesanan, guna untuk menangani rute pengiriman makanan yang sudah siap dikirimkan.

5.2. Saran

Pada penelitian ini ada beberapa saran yang dapat diberikan untuk pengembangan sistem selanjutnya meliputi.

1. Hasil dari perhitungan dari sistem ini memang dinilai cukup akurat, namun memakan waktu yang cukup lama jika banyak titik tujuan yang diproses. oleh karena itu, diharapkan pada pengembang selanjutnya agar mampu membuat sistem dengan proses yang lebih cepat dan menggunakan metode yang lebih baik untuk request data dan pengolahannya.
2. Pada sistem ini hanya menampilkan urutan titik tujuan yang harus dikunjungi. diharapkan pada pengembang selanjutnya agar mampu membuat sistem yang dapat memberikan fitur tracking sehingga kurir dapat mengetahui posisi dia saat dia tersesat atau tidak tau jalan.

Daftar Pustaka

- [1] Fakhri. (2008). Penerapan Algoritma Dijkstra Dalam Pencarian Solusi Maximum Flow Problem. Makalah If2251 Strategi Algoritmik.
- [2] Purwananto, Y., & dkk. (2005). Implementasi Dan Analisis Algoritma Pencarian RuteTerpendek Di Kota Surabaya. Jurnal Penelitian dan Pengembangan TELEKOMUNIKASI.
- [3] Wibowo, A., & Wicaksono, A. (2012). Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit di Purbalingga dengan Metode Algoritma Dijkstra. JUITA ISSN : 2086-9398.
- [4] Gusmao, A., & dkk. (2013). Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra. Jurnal EECIS.
- [5] Roqib, A. (2014). Sistem Informasi Geografis Pencarian SPBU Terdekat dan Penentuan Jalur Terpendek Menggunakan Algoritma Dijkstra di Kabupaten Jember Berbasis Web.