APLIKASI MOBILE PENCARIAN RUTE TERPENDEK PADA PENGIRIMAN ORDER CV. ALFA FRESH DENGAN ALGORITMA *BRUTE FORCE*

Ulsa Oktari[1], Dr. Indri Ariyanti, S.E.,M.Si[2], M. Aris Ganiardi, S.Si.,MT[3]

Program Studi D4 Manajemen Infromatika
Jurusan Manajemen Infromatika, Politeknik Negeri Sriwijaya
Jl. Srijaya Negara Bukit Besar, Bukit Lama, Ilir Barat 1, Palembang 30139

Email : ulsaoktari25@gmail.com[1], indri3673@yahoo.com[2], marisg2010@gmail.com[3]

Abstrak. *Traveling Salesman Problem* merupakan pemecahan masalah yang digunakan dalam mencari rute terpendek untuk mengunjungi semua node tepat satu kali dan kemudian kembali ke node awal. Pemecahan masalah *Traveling Salesman Problem* dengan menggunakan algoritma *Brute Force*. Objek penelitian ini adalah kurir di CV. Alfa Fresh. Algoritma *Brute Force* mampu memberikan solusi untuk *Traveling Salesman Problem* untuk memilih dan menentukan rute-rute terpendek untuk antar orderan dari kantor menuju ke tempat tujuan. Algoritma Brute Force adalah algoritma yang digunakan untuk mencocokan pola dengan semua rute yang akan dilalui untuk menemukan pola rute yang terpendek .Algoritma *Brute Force* bekerja dengan meng-enumerasi semua kandidat kemungkinan yang ada. Dengan adanya aplikasi ini dapat mempermudah kurir dalam menentukan rute terdekat dari posisi kurir tersebut.

Kata Kunci : *Traveling Salesman Problem*, Algoritma *Brute Force*, Rute Terpendek.

1. PENDAHULUAN

CV. Alfa Fresh adalah perusahaan asal Indonesia yang menyediakan layanan pembelian kebutuhan dapur secara *online* seperti penjualan buah-buahan, sayur-sayuran, daging, dan bumbu dapur lainnya. CV. Alfa Fresh ini didirikan pada tahun 2017 di kota Palembang. Perusahaan ini menyediakan aplikasi berbasis Web dan Mobile untuk pemasarannya. CV. Alfa Fresh sendiri mempunyai banyak karyawan dengan berbagai macam bagian yang ada, salah satunya adalah kurir. Kurir dalam CV. Alfa Fresh bertugas mengantar pesanan seperti halnya pada perusahan lain, kurir yang membantu pembeli untuk mengantarkan pesanannya yang di *order* secara *online*.

Kurir mengunjungi tempat pelanggannya, dimana hal ini tempat atau tujuan kurir tidak hanya satu tempat melainkan beberapa tempat setiap harinya sesuai dengan pesanan yang diterima. Dan cukup diantar satu kali untuk setiap pelanggan. Apabila terdapat 5 tempat/tujuan yang akan dikunjungi oleh kurir, maka terdapat beberapa rute yang mungkin dilewati. Hal ini yang menyebabkan kurir harus dapat mengatur urutan kunjungannya agar memperoleh jarak yang seefesien mungkin. Permasalahan yang sering terjadi adalah kurir terkadang menempu jarak yang terlalu jauh dan terlalu berputar-putar untuk mengunjungi satu tempat ke tempat lainnya, ini yang menyebabkan kurir menjadi kurang efisien dalam mengunjungi urutan tempat yang ingin ditujunya.

Penyelesaian persoalan *travelling salesman problem* dapat diterapkan mengunjungi sejumlah tempat. Untuk menyelesaikan persoalan tersebut ada berbagai metode yang dapat digunakan salah satunya dengan menggunakan algoritma *Brute Force*. Algoritma *Brute Force* adalah algoritma yang digunakan untuk mencocokan pola dengan semua rute yang akan dilalui untuk menemukan pola rute yang Optimal. *Brute Force* adalah teknik pemecahan masalah yang sangat umum dan dapat digunakan untuk menyelesaikan masalah dengan menemukan rute terpendek. *Brute Force* bekerja dengan meng-enumerasi semua kandidat kemungkinan yang ada, sehingga menghasilkan solusi yang terbaik.

Agar aplikasi mobile in dapat dimanfaatkan oleh kurir pada perusahaan tersebut maka diperlukan sebuah aplikasi mobile yang dapat melakukan penggantar orderan berdasarkan alamat pelanggan tersebut kemudian menentukan urutan rute terpendek yang akan dilalui dari data yang telah berhasil dihimpun. Untuk itu penulis bermaksud untuk melakukan sebuah penelitian dengan judul **"Aplikasi Mobile Pencarian Rute Terpendek Pada Pengiriman Order CV. Alfa Fresh Dengan Algoritma *Brute Force*"** Aplikasi Mobile tersebut diharapkan dapat membantu petugas kurir pada CV. Alfa Fresh untuk menentukan rute terpendek dalam mengunjungi pelanggan sehingga dapat mempersingkat jarak yang ditempuh dan menghemat waktu.

2. TINJAUAN PUSTAKA

2.1 Rute Terpendek

(Andayani dan Wulan. 2014) Rute terpendek adalah suatu persoalan untuk mencari lintasan antara dua atau lebih simpul pada graf

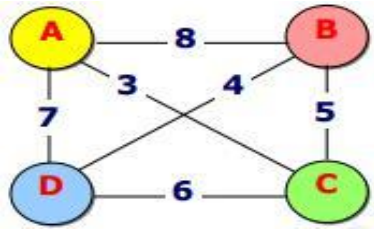
berbobot yang gabungan bobot sisi graf yang dilalui berjumlah paling minimum. Pada graf berbobot terdapat optimasi yang dapat dinyatakan dalam jarak antar kota, waktu pengiriman pesan, biaya dan sebagainya. Dalam hal ini bobot harus bernilai positif, walau dalam hal lain dapat bernilai negatif. Lintasan terpendek dengan verteks awal s dan verteks tujuan t didefinisikan sebagai lintasan terpendek dari s dan t dengan bobot minimum dan berupa lintasan sederhana (simple path).

2.2 Konsep *Travelling Salesman Problem* (TSP)

(Tri, 2013) Apa yang dilakukan dalam TSP adalah membentuk sebuah rute perjalanan. Operator yang bisa digunakan untuk masalah TSP adalah pencarian urutan semua lokasi untuk memilih lokasi yang belum pernah terpilih satu demi satu sehingga dihasilkan satu rute kunjungan yang lengkap dari lokasi awal kemudian mengunjungi semua lokasi yang lain tepat satu kali dan akhirnya kembali ke lokasi awal. Sehingga dengan definisi tersebut dapat dikatakan bahwa konsep permasalahan TSP memiliki aturan sebagai berikut:

1. Harus mengunjungi setiap tempat tepat satu kali, tidak boleh kurang ataupun lebih.
2. Semua tempat harus dikunjungi dalam satu kali perjalanan (*tour*).
3. Dimulai dan diakhiri pada kota yang sama.

Sebagai ilustrasi dengan Gambar 2.1, diasumsikan bahwa simpul awal dan simpul akhir adalah 1. Suatu graf TSP dengan 4 simpul tersebut dikonversi menjadi sebuah pohon pencarian yang menghasilkan (4-1)! = 6 kemungkinan urutan kunjungan.



Gambar 2.1 Contoh Graf Rute Perjalanan

Berdasarkan (Suyanto, 2010) TSP dikatakan ada 2 jenis, yaitu:

1. TSP asimetris

TSP jenis asimetirs ini, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1. Dengan n kota, besarnya ruang pencarian adalah ( ) jalur yang mungkin.

2. TSP simetris

TSP jenis simetris, biaya dari kota 1 ke kota 2 adalah sama dengan biaya dari kota 2 ke kota 1. Apabila dengan n kota, jumlah jalur yang mungkin adalah jalur yang mungkin.

2.3 Algoritma *Brute Force*

(Wilson, Yuliant, Izzatul. 2015) Metode *Brute force* merupakan pendekatan yang langsung dalam memecahkan suatu masalah, yang didasarkan pada pernyataan masalah serta mendefinisikan terhadap konsep yang dilibatkan secara langsung. Pemecahan masalah menggunakan Metode *Brute force* sangat sederhana, langsung dan jelas. Karakteristik pada Metode *Brute force* :

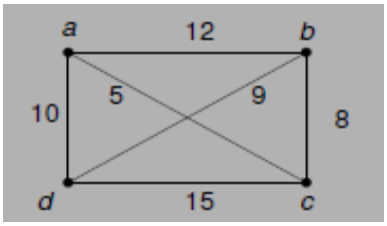
1. Jumlah langkah yang di perlukan besar.
2. Dipakai sebagai dasar dalam menemukan suatu solusi yang lebih efisien atau kreatif.
3. Hampir semua masalah dapat diselesaikan dengan metode ini.
4. Digunakan sebagai dasar dalam perbandingan kualitas suatu algoritma.

2.4 Algoritma *Brute Force* Dengan Teknik *Exhaustive Search* Pada *Traveling Salesman Problem*

Exhaustive search adalah teknik pencarian solusi secara *brute force* untuk masalah yang melibatkan pencarian elemen dengan sifat khusus. Biasanya di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan (Munir, Rinaldi. 2005). Penggunaan *Brute force* pada permasalahan TSP dapat di ilustrasikan sebagai berikut:

Diberikan banyak *(n)* tempat tujuan pada sebuah kota serta diketahui jarak antara setiap kota satu sama lain. Temukan perjalanan (*tour*) terpendek yang melalui setiap kota lainnya hanya sekali dan kembali lagi ke kota asal keberangkatan. Permasalahan TSP ini tidak lain adalah menemukan sirkuit Hamilton dengan bobot minimum. Langkah-langkah teknik *exhaustive search:*

1. Buat list (Enumerasi) setiap solusi yang mungkin dengan cara yang sistematis.
2. Evaluasi setiap kemungkinan solusi satu per satu, terlebih solusi yang *mirror*, bisa di keluarkan. Kemudian simpan solusi terbaik yang ada sampai saat ini.
3. Setelah pencarian berakhir, umumkan solusi terbaik.
4. Sumber daya yang dibutuhkan dalam pencarian solusi menggunakan *Exhaustive Search* sangat besar, meskipun teknik *exhaustive* secara teoritis menghasilkan solusi.



Gambar 2.2 Graph ABCD

R1 = (A,B,C,D,A) = 12 + 8 + 15 + 10 = 45

R2 = (A,B,D,C,A) = 12 + 9 + 8 + 5 = 41

R3 = (A,C,B,D,A) = 5 + 8 + 9+ 10 = 32

R4 = (A,C,D,B,A) = 5 + 15 + 9+ 12 = 41

R5 = (A,D,B,C,A) = 10 + 9 + 8+ 5 = 32

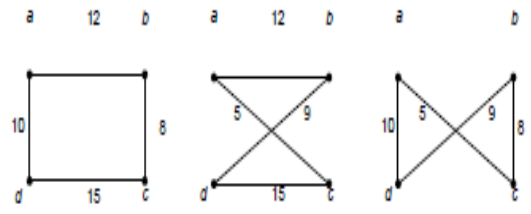
R6 = (A,D,C,B,A) = 10 + 15 + 8+ 12 = 45

Jika contoh diselesaikan dengan teknik *exhaustive search*, maka kita harus mengenumerasi sebanyak (*n* – 1)! buah sirkuit Hamilton, menghitung setiap bobotnya, dan memilih sirkuit Hamilton dengan bobot terkecil.

Teknik *exhaustive search* di atas dapat dilakukan perbaikan mengenai banyaknya pencarian rute, yakni dengan diketahui bahwa setengah dari rute perjalanan adalah hasil pencerminan dari setengah rute yang lain, yaitu dengan mengubah arah rute perjalanan, rute tersebut adalah :

1. R1 dan R6
2. R2 dan R4
3. R3 dan R5

Maka dapat dihilangkan setengah dari jumlah permutasi (dari 6 menjadi 3). Ketiga buah sirkuit Hamilton yang dihasilkan adalah seperti gambar 3.2.



Gambar 2.3 Sirkuit Hamilton

Untuk graf dengan banyaknya (*n)* buah simpul, hanya perlu mengevaluasi sirkuit Hamilton sebanyak (*n* – 1)/2 buah.

3. METODOLOGI PENELITIAN

3.1 Tahapan Perumusan Masalah

Tahap ini merupakan proses perumusan masalah dan membatasi masalah yang akan diteliti. Hal ini dibutuhkan agar dapat lebih mengarahkan peneliti dalam membuat aplikasi sehingga yang dikerjakan tidak keluar dari batasan yang telah ditetapkan sebelumnya.

3.2 Tahapan Pengumpulan Data

Dalam tahap pengumpulan data ini penulis mengacu pada pendapat (Suryabrata. 39:2015), dimana pendapat tersebut menjelaskan tahap pengumpulan data yang terbagi menjadi dua macam, yaitu:

1. Data Primer

Data primer adalah data yang pertama kali dicatat dan dikumpulkan dalam penelitian. Data primer dimana Penulis melakukan survey secara langsung ke pihak yang bersangkutan, yaitu pihak yang memiliki otoritas terhadap pengumpulan data tersebut. Penulis melakukan wawancara dan observasi dengan pegawai CV.Alfa Fresh Palembang. Wawancara tersebut menghasilkan sebuah kendala yang dihadapi pihak CV. Alfa Fresh Palembang dimana kurir sebagai pengantar orderan ke pelanggan masih terkendala dalam menentukan rute terefektif untuk mengantarkan pesanan ke pelanggan, sehingga waktu pengiriman orderan masih kurang efektif.

2. Data Sekunder

Data sekunder yaitu data yang dikumpulkan dari sumber-sumber yang ada. Disini, disini penulis mengumpulkan data secara tidak langsung, yaitu dengan cara mencari informasi melalui jurnal penelitihan, buku, dan sumber dokumen lainnya, sebagai referensi dari pembangunan aplikasi mobile yang akan dibuat

.

3.3 Tahapan Perancangan Penelitian

Dalam mengerjakan tugas akhir ini, perancangan sistem yang digunakan meliputi *usecase diagram, activity diagram, class diagram, dan sequence diagram.*

3.4 Penyelesaian Algoritma *Brute Force* Dengan Teknik *Exhaustive Search* Pada *Traveling Salesman Problem*

Contoh studi kasus penerapan dari algoritma *brute force* dengan teknik *Exhaustive Search* Pada *Traveling Salesman Problem* itu sendiri adalah sebagai berikut, dengan posisi start awal di CV.Alfa Fresh lalu titik pertama ke Jalan Putri Rambut Selako no.4 Ilir barat 1, titik kedua Jalan Joko no 10 Bukit kecil, dan titik ke tiga Jalan Bali Blok C3 Komplek Pusri Borang sako lalu kembali lagi ke CV. Alfa Fresh.

Tabel 3.1 List Order

Posisi	Latitude	Longitude
Cv. Alfa Fresh (A)	-2.9439079	104.7327026
Jalan Putri Rambut Selako No.4 Ilir barat 1 (B)	-2.9934064	104.7300526
Jalan Joko No 10 Bukit Kecil (C)	-2.9930219	104.7480037
Jalan Bali Blok C3 Komplek Pusri Borang sako (D)	-2.9252864	104.7794028

Pertama kita mencari jarak antara posisi start awal ke tujuan dengan rumus perhitungan jarak euclidean yaitu :

$$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}$$

Keterangan
x_i = Koordinat x untuk titik i awal (latitude)
y_i = Koordinat y untuk titik j tujuan (longitude)
d_{ij}= Jarak Antara dua titik i dan j

Mencari jarak antara start awal yaitu CV. Alfa Fresh ke tujuan Jalan Putri Rambut Selako No. 4 Ilir barat 1 (A-B) dengan rumus sebagai berikut :

$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}$
$= \sqrt{((-2.9439079) - (-2.9934064)^2) + ((104.7327026) - (104.7300526)^2)}$
$= \sqrt{(0.0494985)^2 + (0.00265)^2}$
$= \sqrt{0.002450101502 + 0.0000070225}$
$= \sqrt{0.002457124002}$
$= 0.049569385733 \times \frac{111.322}{1000}$
$= 0.0005518163159$

Mencari jarak antara start awal yaitu CV. Alfa Fresh ke tujuan Jalan Joko No. 10 Bukit kecil (A-C) dengan rumus sebagai berikut :

$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}$
$= \sqrt{((-2.9439079) - (-2.9930319)^2) + ((104.7327026) - (104.7480037)^2)}$
$= \sqrt{(0.049124)^2 + (0.0153011)^2}$
$= \sqrt{0.002413167376 + 0.0002341236612}$
$= \sqrt{0.0026472910372}$
$= 0.0514518322045 \times \frac{111.322}{1000}$
$= 0.005727720865$

Mencari jarak antara start awal yaitu CV. Alfa Fresh ke tujuan Jalan Bali Blok C3 Komplek Pusri Borang sako (A-D) dengan rumus sebagai berikut :

$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}$
$= \sqrt{((-2.9439079) - (-2.9252864)^2) + ((104.7327026) - (104.7794028)^2)}$
$= \sqrt{(-0.0186215)^2 + (-0.0467002)^2}$
$= \sqrt{0.00034676026225 + 0.00218090868004}$
$= \sqrt{0.00252766894229}$
$= 0.05027592805996 \times \frac{111.322}{1000}$
$= 0.00559681686349$

Mencari jarak antara start awal yaitu Jalan Putri Rambut Selako No. 4 Ilir barat 1 ke tujuan Jalan Joko No.10 Bukit kecil (B-C) dengan rumus sebagai berikut :

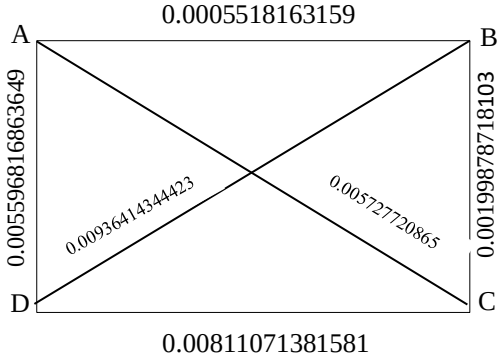
$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}$
$= \sqrt{((-2.9934064) - (-2.9930319)^2) + ((104.7300526) - (104.7480037)^2)}$
$= \sqrt{(-0.0003745)^2 + (-0.0179511)^2}$
$= \sqrt{0.00000014025025 + 0.00032224199121}$
$= \sqrt{0.00032238224146}$
$= 0.01795500602785 \times \frac{111.322}{1000}$
$= 0.00199878718103$

Mencari jarak antara start awal yaitu Jalan Putri Rambut Selako No. 4 Ilir barat 1 ke tujuan Jalan Bali Blok C3 Komplek Pusri Borang sako (B-D) dengan rumus sebagai berikut :

$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}$
$= \sqrt{((-2.9934064) - (-2.9252864)^2) + ((104.7300526) - (104.7794028)^2)}$
$= \sqrt{(-0.06812)^2 + (-0.0493502)^2}$
$= \sqrt{0.0046403344 + 0.00243544224004}$
$= \sqrt{0.00707577664004}$
$= 0.08411763572545 \times \frac{111.322}{1000}$
$= 0.00936414344423$

Mencari jarak antara start awal yaitu Jalan Joko No. 10 Bukit kecil ke tujuan Jalan Bali Blok C3 Komplek Pusri Borang sako (C-D) dengan rumus sebagai berikut :

$d_{ij} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}$
$= \sqrt{((-2.9930319) - (-2.9252864)^2) + ((104.7480037) - (104.7794028)^2)}$
$= \sqrt{(-0.064745)^2 + (-0.0313991)^2}$
$= \sqrt{0.004322405025 + 0.000985903480}$
$= \sqrt{0.005308308505}$
$= 0.07285813959332 \times \frac{111.322}{1000}$
$= 0.00811071381581$



Gambar 3.1 Ilustrasi hasil perhitungan jarak order

Setelah Mencari jarak dapatlah semua bobot semua jarak lalu kita harus menghitung setiap bobotnya, dan memilih sirkuit Hamilton dengan bobot terkecil.

R1=(A,B,C,D,A)=0.0005518163159+0.001998787 18103+0.00811071381581+0.00559681686349 =0.01625813417623

R2=(A,B,D,C,A)=0.0005518163159+0.009364143 44423+0.00811071381581+ 0.005727720865

= 0.02375439444094

R3=(A,C,B,D,A)=0.005727720865+0.0019987871
8103+0.00936414344423+0.00559681686349
=0.02268746835375

R4=(A,C,D,B,A)=0.005727720865+0.0081107138
1581+0.00936414344423+0.0005518163159
=0.02375439444094

R5=(A,D,B,C,A)=0.00559681686349+0.00936414
344423+0.00199878718103+0.005727720865
=0.02268746835375

R6=(A,D,C,B,A)=0.00559681686349+0.00811071
381581+0.00199878718103+0.0005518163159
=0.01625813417623

Jadi dengan perhitungan teknik *exhaustive search* di atas dapat dilakukan perbaikan mengenai banyaknya pencarian rute, yakni dengan diketahui bahwa setengah dari rute perjalanan adalah hasil pencerminan dari setengah rute yang lain, yaitu dengan mengubah arah rute perjalanan, rute tersebut adalah :

R1 dan R6 = .01625813417623

R2 dan R4 = 0.02375439444094

R3 dan R5 = 0.02268746835375

Maka rute yang dipilih berdasarkan total jarak yang paling terkecil atau terdekat adalah R1 dan R6.

3.5 Metode Pengembangan Sistem

Jika didalam pembuatan aplikasi penulis menggunakan algoritma *Brute Force*, maka dalam pengembangan sistem penulis menggunakan metode pengembangan sistem *Waterfall*. Metode *waterfall* dikerjakan secara bertahap secara satu-persatu mulai dari tahap yang paling atas sampai dengan tahap yang paling bawah. Tahapan yang dilalui meliputi tahap analisis,desain, pengodean, pengujian, dan tahap pendukung *(support)*. Metode pengembangan *waterfall* dipilih karena penerapan langkah-langkah dalam *waterfall* sesuai dengan metode pengembangan yang dilakukan penulis.

a. Analisis kebutuhan perangkat lunak
   Tahapan analisis kebutuhan perangkat lunka adalah proses pengumpulan kebutuhaan sistem seperti Android Studio untuk editor mengcoding aplikasi, Sublime Text berguna untuk mengcoding web, MySql untuk menyimpan database, PHP untuk mengkode program php.
b. Desain
   Tahap desain merupakan analisis kebutuhan perangkat lunak perancangan sistem dalam bentuk diagram *usecase*, diagram *activity*, diagram *class*, dan diagram *sequence*.
c. Pembuatan kode program
   Tahap pembuatan kode program adalah proses melakukan *coding system* (Pengkodean Perangkat Lunak) pada Android Studio dan Sumbile Text oleh Programmer sesuai dengan tahap analisis dan desain yang telah dibuat sebelumnya.
d. Pengujian
   Tahap pengujian penulis menggunakan metode *Black Box*, pengujian dilakukan untuk mengetahui apakah sistem yang dibuat berjalan dengan baik dan tidak. Dalam melakukan pengujian penulis membuat rencana pengujian sebagai acuan dalam melakukan pengujian.
e. Pendukung *(support)* atau pemeliharaan *(maintenance)*
   Tahapan pemeliharaan(*maintance)* ini melakukan cek pada sistem aplikasi apabila ada kesalah maka aplikasi akan diperbaiki.

4. HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan Sistem

a. Kebutuhan Fungsional

Rosa dan Shalahudin (2013:22) Menjelaskan *functional requirement* (kebutuhan fungsional) adalah sebuah kebutuhan terkait dengan fungsi produk, misalnya, sistem informasi harus mampu mencetak laporan, sistem informasi harus mampu menampilkan grafik, dan lain-lain.

Adapun kebutuhan fungsional dari sistem yang akan dibangun adalah sebagai berikut:

1. Sistem *website* dan *aplikasi* dapat melakukan otoritas dan otentifikasi penggunanya seperti admin dan kurir.
2. Sistem *website* yang dibuat dapat melakukan input data, edit data, dan menghapus data berupa data admin, customer, detail order, order, produk, user, dan kurir.
3. Sistem *aplikasi* dapat melakukan pencarian rute terdekat.
4. Sistem *aplikasi* dapat melihat *history order* yang sudah selesai dilakukan.

b. Kebutuhan Non Fungsional

Kebutuhan Non-Fungsional adalah kebutuhan tambahan yang tidak memiliki input, proses, dan output. Namun, kebutuhan non-fungsional ini sebaiknya dipenuhi karena akan sangat menentukan apakah sistem ini akan digunakan user atau tidak.

Adapun kebutuhan non fungsional yang terkait ini ada dua yaitu :

1. *Website* Admin yang akan dibangun meliputi :
a. *Website* harus dapat diakses dengan *browser*.
b. *Website* harus dapat diakses oleh pengguna yang dirumuskan.
c. Memiliki tampilan (*user interface*) yang *friendly*.

d. Sistem *website* dilengkapi *password*, sehingga hanya *user* yang terdaftar yang dapat mengakses *website*.

2. *Aplikasi* kurir yang akan dibangun meliputi :

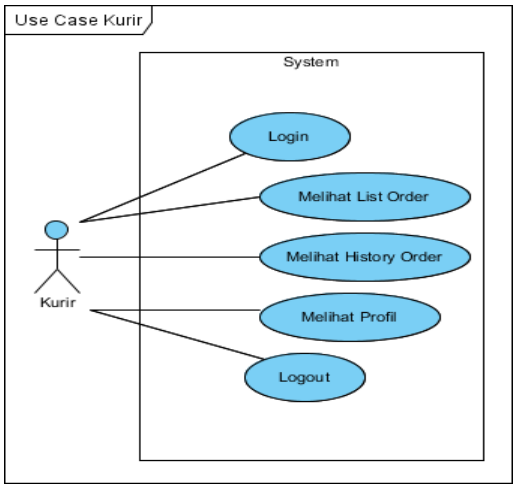
a. Kemudahan penggunaan perangkat lunak dapat diakses dimana saja dan kapan saja melalui aplikasi berbasis *mobile*.

b. Faktor keamanan berkaitan dengan penggunaan API Key (*Application Programming Interface*).

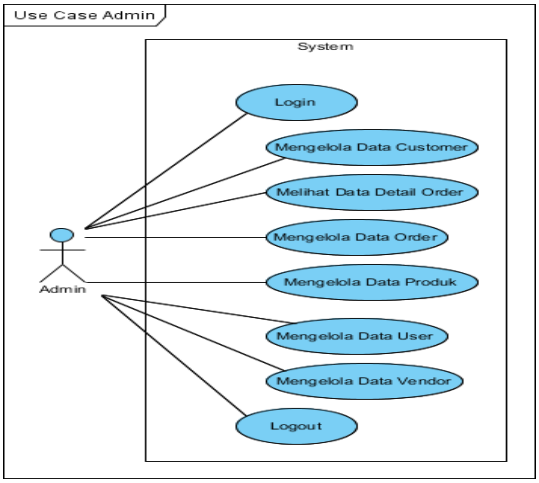
4.2 Perancangan Sistem

a. Usecase Diagram

Menurut Rosa dan Shalahuddin (2013:155) *Usecase* atau diagram *use case* merupakan pemodelan untuk kelakukan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.



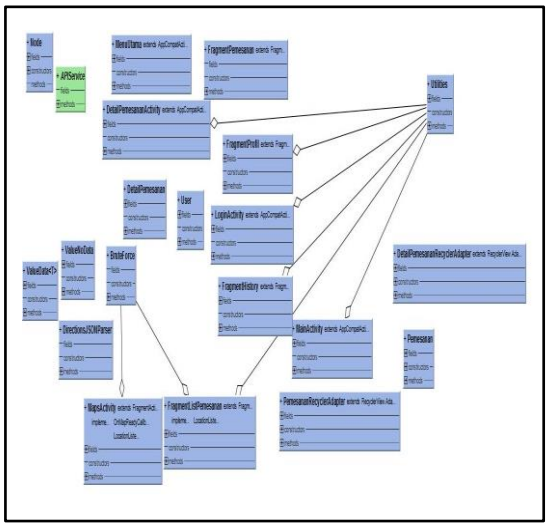
Gambar 4.1 *Usecase* Diagram Kurir



Gambar 4.2 *Usecase* Diagram Admin

b. Class Diagram

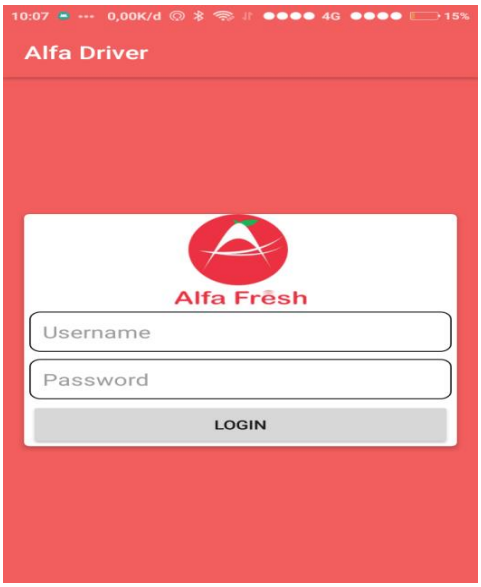
Rosa dan Shalahudin (2013 : 141-142) *Class Diagram* atau diagram kelas mengambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Sedangkan, operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



Gambar 4.3 *Class* Diagram *Aplikasi*

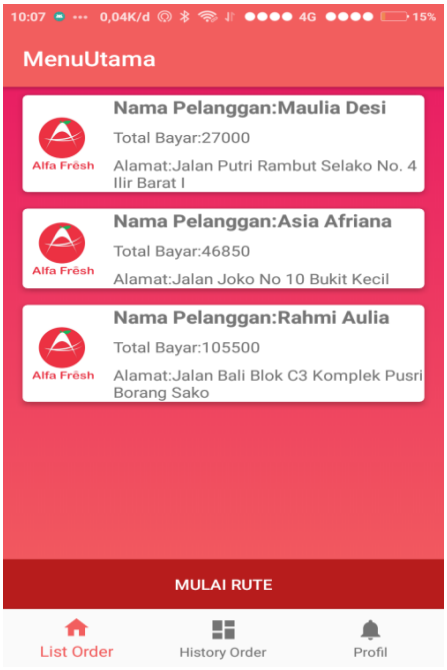
4.3 Implementasi Sistem

Berikut ini tampilan antarmuka dari aplikasi :



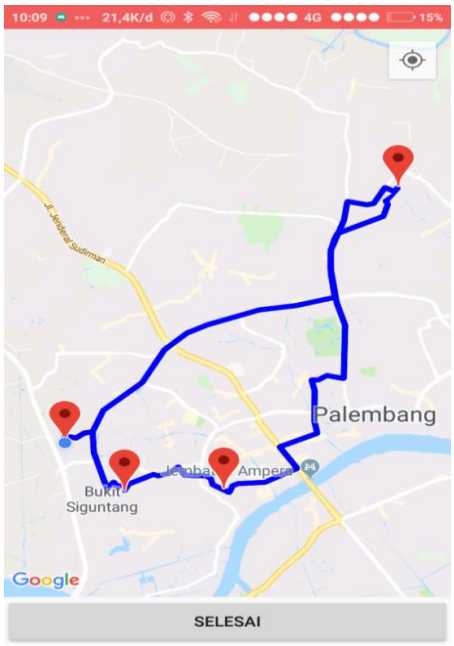
Gambar 4.4 Halaman Login Aplikasi

Pada halaman ini kurir melakukan login terlebih dahulu untuk mengakses aplikasi ini dengan mengisi username dan password lalu menekan tombol login. Jika berhasil login kurir bisa melakukan akses pada halaman selanjutnya.



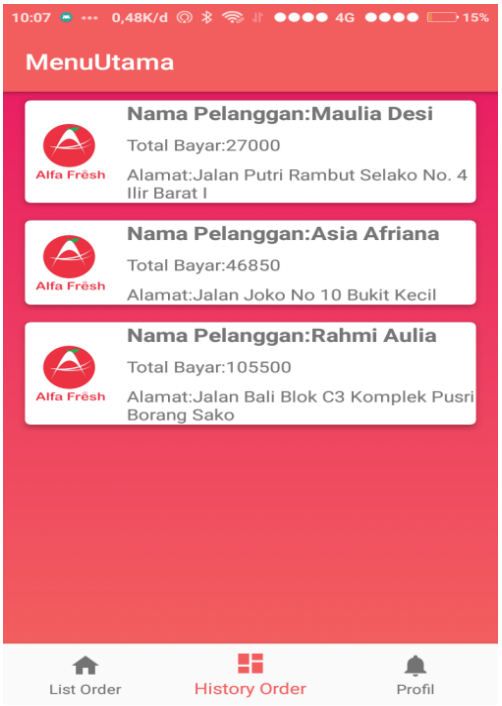
Gambar 4.5 Halaman List Order Aplikasi

Halaman utama ini berfungsi untuk melihat list orderan dan dengan menekan tombol akses Mulai Rute maka kurir dapat mengetahui rute orderan yang akan di laluinya. Selain itu, halaman ini berisi 3 panel akses berupa List Order, history order, dan profil.



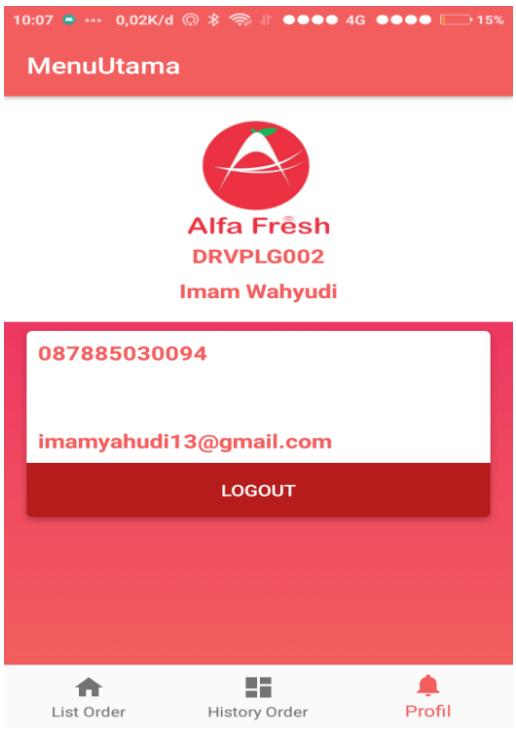
Gambar 4.6 Halaman Mulai rute Aplikasi

Merupakan halaman peta rute orderan yang ada di dalam aplikasi. Halaman ini akan membantu kurir mendapatkan rute jarak terpendek mengantar orderan dan disertai dengan arah penunjuk jalan seperti yang tersedia di dalam Google Maps.



Gambar 4.7 Tampilan Halaman *History Order* Aplikasi

Halaman ini merupakan halaman yang akan menampilkan *history orderan* jika kurir tersebut telah selesai melakukan pengantaran pesanan pelanggan.



Gambar 4.8 Halaman Tampilan Profil Aplikasi

Halaman ini merupakan halaman yang akan menampilkan profil kurir tersebut dan terdapat tombol logout untuk kurir tersebut keluar dari aplikasi.

5. KESIMPULAN DAN SARAN

1. Kesimpulan

Berdasarkan pembahasan yang telah diuraikan pada bab-bab sebelumnya, maka terdapat beberapa kesimpulan yaitu sebagai berikut :

1. Dengan adanya aplikasi ini dapat mempermudah kurir dalam menentukan rute terdekat dari posisi kurir tersebut.
2. Aplikasi ini juga dapat menjadi catatan bagi kurir untuk melihat order apa saja yang dia lakukan.
3. Algoritma brute force dapat memberikan rute terpendek dari posisi kurir dengan baik, akan tetapi proses perhitungan sedikit lebih lambat karena algoritma tersebut menghitung semua kemungkin jarak yang ada.

2. Saran

Adapun saran yang dapat diberikan oleh penulis berdasarkan pengamatan yang telah dilakukan antara lain:

1. Akan lebih berguna apabila aplikasi ini ditambahkan tentang kilometer yang telah dia tempuh guna menjadi pengingat untuk menyervis kendaraan.
2. Aplikasi mobile ini akan lebih baik jika dilengkapi dengan fiture penerima pesan dari kantor.

6. DAFTAR PUSTAKA

Andayani Sri, Wulan Endah Perwitasari, 2014, Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra, Seminar Nasional Teknologi Informasi dan Komunikasi Terapan 2014, ISBN: 979-26-0276-3, hal 165

Tri Dian Wiyanti, 2013, Algoritma Optimasi Untuk Menyelesaikan Travelling Salesman Problem, Jurnal Transformatika, vol 11, hal 2

Wilson, Yuliant, Izzatul, 2015, Analisis Penyelesaian Traveling Salesman Problem Dengan Metode *Brute Force* Menggunakan *Graphic Processing Unit*, e-Proceeding of Engineering, vol 2, hal 1875-1876

Munir, Rinaldi. Diklat Kuliah IF2251. *Strategi Algoritmik*

Suryabrat Sumadi, 2015, Metodelogi Penelitihan, Jakarta, Rajawali Pers

A.S Rosa , dan M.Shalahuddin. 2014. *Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek.* Bandung : Informatika.