OPTIMASI RUTE DAN BIAYA OPERASIONAL DALAM EKSPEDISI DARAT MENGGUNAKAN ALGORITMA *TABU SEARCH* BERBASIS ANDROID
(Studi Kasus: PT. ANTESS)

SKRIPSI

Oleh:
AKMAL AFIF
NIM. 09650010

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014

OPTIMASI RUTE DAN BIAYA OPERASIONAL DALAM EKSPEDISI DARAT MENGGUNAKAN ALGORITMA *TABU SEARCH* BERBASIS ANDROID
(Studi Kasus: PT. ANTESS)

SKRIPSI

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S. Kom)

Oleh:
AKMAL AFIF
NIM. 09650010

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2014

OPTIMASI RUTE DAN BIAYA OPERASIONAL DALAM EKSPEDISI DARAT MENGGUNAKAN ALGORITMA *TABU SEARCH* BERBASIS ANDROID (Studi Kasus: PT. ANTESS)

SKRIPSI

Oleh:

AKMAL AFIF

NIM. 09650010

Telah Disetujui untuk Diuji

Malang, April 2014

Dosen Pembimbing I	Dosen Pembimbing II
Fachrul Kurniawan, M. MT	**Yunifa Miftachul Arif, M.T**
NIP. 19771020 200912 1 001	**NIP. 19830616 201101 1 004**

Mengetahui
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

OPTIMASI RUTE DAN BIAYA OPERASIONAL DALAM EKSPEDISI DARAT MENGGUNAKAN ALGORITMA *TABU SEARCH* BERBASIS ANDROID (Studi Kasus: PT. ANTESS)

SKRIPSI

Oleh:
AKMAL AFIF
NIM. 09650010

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal : 10 April 2014

Susunan Dewan Penguji :			**Tanda Tangan**
1. Penguji Utama	**:**	**Hani Nurhayati, M.T** **NIP. 19780625 200801 2 006**	**( )**
2. Ketua Penguji	**:**	**Fresy Nugroho, M.T** **NIP. 19710722 201101 1 001**	**( )**
3. Sekretaris Penguji	**:**	**Fachrul Kurniawan, M.MT** **NIP. 19771020 200901 1 001**	**( )**
4. Anggota Penguji	**:**	**Yunifa Miftachul Arif, M.T** **NIP. 19830616 201101 1 004**	**( )**

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama Lengkap	:	Akmal Afif
NIM	:	09650010
Fakultas/Jurusan	:	SAINS DAN TEKNOLOGI / TEKNIK INFORMATIKA
Judul Skripsi	:	**OPTIMASI RUTE DAN BIAYA OPERASIONAL DALAM EKSPEDISI DARAT MENGGUNAKAN ALGORITMA TABU SEARCH BERBASIS ANDROID (Studi Kasus: PT. ANTESS)**

Dengan ini menyatakan bahwa:

1. Isi dari skripsi yang saya buat ini adalah benar-benar karya saya sendiri dan tidak terdapat unsur-unsur penjiplakan karya orang lain, selain nama-nama termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila di kemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya bersedia untuk mempertanggung jawabkan, dan menanggung segala resiko, serta diproses sesuai peraturan yang berlaku.

Demikian pernyataan ini saya buat dengan penuh kesadaran.

Malang, 31 Maret 2014
Yang Membuat Pernyataan,

Akmal Afif
NIM. 09650010

MOTTO

- Tidak ada mimpi yang terlalu tinggi untuk diraih -

PERSEMBAHAN

Buat:

Abahku, Ibuku, Kakakku dan seluruh keluargaku

yang selalu mendukungku secara Moral, Material, dan Spiritual.

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Syukur Alhamdulillah penulis haturkan kehadirat Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul Optimasi Rute dan Biaya Operasional dalam Ekspedisi Darat Menggunakan Algoritma Tabu Search Berbasis Android (Studi Kasus: PT. ANTESS). Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Ucapan terima kasih penulis sampaikan kepada:

1. Abahku Makinudin, Ibuku Nurul Hidayati, Kakakku M. Asnal Mahfud dan seluruh keluarga besar penulis yang selalu memberikan dukungan, do'a, dan motivasi dalam penyelesaian skripsi ini.
2. Prof. Dr. H. Mudjia Rahardjo, M.Si, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Bayyinatul Muchtaromah, drh. MSi, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak Dr. Cahyo Crysdian selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
5. Bapak Fachrul Kurniawan, M. MT selaku pembimbing dalam skripsi ini yang telah memberikan bimbingan dan pengarahan dalam proses penyelesaian skripsi ini.

6. Bapak Yunifa Miftachul Arif, M.T selaku pembimbing dalam skripsi ini yang telah memberikan bimbingan dan pengarahan dalam proses penyelesaian skripsi ini.
7. Ibu Ririn Kusumawati, M.Kom selaku dosen wali yang telah memberikan nasehat dan pengarahan dalam skripsi ini.
8. Seluruh Dosen Universitas Islam Negeri Maulana Malik Ibrahim Malang, khususnya dosen Teknik Informatika beserta seluruh staf yang telah memberikan ilmu dan membantu dalam penyelesaian skripsi ini.
9. Seluruh teman-teman Jurusan Teknik Informatika khususnya kelas a angkatan 2009.
10. Sahabat-sahabatku di kontrakan dan seluruh sahabat penulis yang telah memotivasi, dan membantu dalam proses penyelesaian skripsi ini.
11. Dan kepada seluruh pihak yang membantu penulisan skripsi ini, yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan. Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan khususnya bermanfaat bagi penulis secara pribadi.

Wassalamu'alaikum Wr. Wb.

Malang, 31 Maret 2014

Penulis,

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Afif, Akmal. 2014. 09650010. **Optimasi Rute dan Biaya Operasional Dalam Ekspedisi Darat Menggunakan Algoritma *Tabu Search* Berbasis Android (Studi Kasus: PT. ANTESS).** Jurusan Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) Fachrul Kurniawan, M. MT, (II) Yunifa Miftachul Arif, M.T

Kata Kunci : *optimasi, estimasi biaya, tabu search*

PT.ANTESS (Antaran Express) merupakan sebuah perusahaan swasta nasional, yang difokuskan bergerak dalam bidang jasa transportasi logistik, *cargo management*, jasa pergudangan dan distribusi yang sesuai dengan kebutuhan pelanggan. Dengan memiliki tujuan pengiriman di berbagai kota di pulau jawa, PT. ANTESS membutuhkan suatu mekanisme pengiriman barang melalui jalur darat yang bisa diandalkan dengan estimasi waktu seefisien mungkin dengan memilih rute pengiriman yang paling cepat dan tepat. Sehingga biaya operasional yang akan dikeluarkan dalam proses pengiriman barang tersebut dapat dikalkulasikan secara standar. Maksimus merupakan aplikasi optimasi yang bertujuan untuk membantu pihak PT. ANTESS dalam menentukan rute tempuh dan besar biaya operasional dalam distribusi logistik, khususnya kota-kota tujuan di pulau jawa dengan berdasarkan jenis kendaraan tertentu.

Penggunaan Aplikasi Optimasi Maksimus sangat membantu dalam menentukan estimasi biaya operasional dalam proses pendistribusian logistik melalui jalur darat. Berdasarkan beberapa kota tujuan dan keperluan biaya lainnya, pengguna dapat mengetahui rute tempuh kendaraan yang paling efektif dari beberapa kota tujuan menggunakan algoritma tabu *search*, total jarak tempuh kendaraan dan informasi tentang besaran biaya yang dibutuhkan selama proses distribusi logistik tersebut. Biaya-biaya tersebut meliputi biaya timbang, biaya perbaikan, dan biaya untuk bahan bakar kendaraan. Informasi rute tempuh kendaraan divisualisasikan dengan menggunakan bantuan *driving direction* di Google Map, dengan mengambil koordinat dari masing-masing kota tujuan.

Berdasarkan fungsinya, aplikasi ini dapat digunakan oleh pihak dari PT. ANTESS (Antaran Express), sebagai bahan pertimbangan dalam menentukan biaya operasional distribusi logistik melalui jalur darat, khususnya kota-kota di pulau jawa.

ABSTRACT

Afif, Akmal. 2014. 09650010. **Route and Operational Cost Optimization in Overland Expedition Using *Tabu Search* Algorithm with Android Based (Case Study: PT. ANTESS).** Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Supervisor (I) Fachrul Kurniawan, M. MT, (II) Yunifa Miftachul Arif, M.T

Keywords : *optimization, cost estimation, tabu search*

PT. ANTESS (Antaran Express) is a national private company, which focuses on engaging logistic transportation services, freight management, warehousing, and distribution services based on customers' demand. By having the purpose of sending in various cities in the Java island, PT. ANTESS requires a mechanism for the delivery of goods by road which can be relied upon to estimate the time as efficient as possible by choosing the quickest and precisely delivery. Because of this process, the operational cost to delivery process can be calculated by standart of procedure. Maksimus is an optimized application that aims to help the PT. ANTESS in determining these great mileage and operational costs while distribution logistics, especially cities on the island of Java based of certain types of vehicle.

Maksimus Optimization Application is helpful in determining the estimated operational costs in the logistics distribution process through the land. Based on several destinations and other expenses purposes, the user can determine the most effective vehicle mileage from several destinations using a tabu search algorithm, the total mileage of the vehicle and the amount of information about the required cost for the logistics distribution process. These costs include the cost of stations, repairing costs, and the cost of fuel for vehicles. Vehicle mileage information is visualized by driving direction on Google Maps, by taking the coordinates of each destination.

Based on this research, the application can run properly on all versions of the Android platform above 4.0.3 (Ice Cream Sandwich) version, and the use of tabu search algorithm can be applied to determine routes and logistics distribution vehicle mileage which can be applied.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Optimasi merupakan suatu proses pencarian satu atau lebih penyelesaian layak yang berhubungan dengan nilai-nilai ekstrim dari satu atau lebih nilai objektif pada suatu permasalahan sehingga tidak terdapat solusi ekstrim yang masih ditemukan. Optimasi sangat berguna dihampir segala bidang dalam rangka melakukan usaha secara efektif dan efisien untuk mendapatkan target dan hasil yang ingin dicapai. Seperti dalam prinsip ekonomi yang berorientasikan untuk senantiasa menekan pengeluaran untuk menghasilkan output dan hasil yang maksimal.

PT. ANTESS (Antaran Express) merupakan salah satu anak perusahaan Sarana Group, yaitu sebuah perusahaan swasta nasional, yang difokuskan bergerak dalam bidang jasa transportasi logistik, *cargo management,* jasa pergudangan dan distribusi yang sesuai dengan kebutuhan pelanggan. Perusahaan yang beralamat di Graha Sarana Puri Surya Jaya J01-20 Vancouver, Gedangan Sidoarjo ini juga menerima jasa pengiriman logistik melalui jalur darat dan penyeberangan dengan tujuan Pulau Jawa, Bali, Sumatera, Bangka, Kalimantan & Sulawesi. PT. ANTESS memiliki pilihan armada yang bervariasi sehingga dapat disesuaikan dengan kebutuhan angkutan dari *customer* lain. Pada kasus ini, difokuskan pada pengangkutan melalui jalur darat, yaitu dengan menggunakan truk-truk untuk mengangkut barang atau bertindak sendiri dengan kendaraan milik perusahaan tersebut.

Perusahaan ekspedisi sebagai perusahaan transportasi mempunyai peranan yang sangat penting dalam mata rantai arus barang-barang dari suatu tempat ke tempat yang lain. Dengan kapasitasnya, jasa ekspedisi dapat mengorganisasikan pelaksanaan pengiriman melalui jalur darat, laut, dan udara. Dengan demikian ekspedisi lebih memberikan jasa bagi pertukaran (pengangkutan dan penerimaan) barang-barang dengan tarif dan biaya yang didasarkan kepada suatu persetujuan terlebih dahulu dengan langganannya. Dengan memiliki tujuan pengiriman di berbagai daerah dari beberapa kota di pulau jawa, PT. ANTESS membutuhkan suatu mekanisme pengiriman barang melalui jalur darat yang bisa diandalkan dengan estimasi waktu seefisien mungkin dengan memilih rute pengiriman yang paling cepat dan tepat. Sehingga biaya operasional yang akan dikeluarkan dalam proses pengiriman barang tersebut dapat dikalkulasikan secara standar.

Hal ini juga sesuai dengan Firman Allah SWT dalam Al Qur'an Surat Al Isra' ayat 27:

إِنَّ ٱلْمُبَذِّرِينَ كَانُوٓاْ إِخْوَٰنَ ٱلشَّيَٰطِينِ ۖ وَكَانَ ٱلشَّيْطَٰنُ لِرَبِّهِۦ كَفُورًا ﴿٢٧﴾

Artinya : *"Sesungguhnya orang-orang yang pemboros itu adalah saudara setan dan setan itu sangat ingkar kepada Tuhannya".*

Ayat di atas menjelaskan bahwa Allah menyuruh manusia agar senantiasa mensyukuri atas apa yang telah Allah karuniakan kepada mereka, tidak menyalahgunakannya, seperti menafkahkannya dengan boros, tidak mensyukurinya, dan memanfaatkannya kepada kemaksiatan. Apabila mereka

ingkar terhadap rezeki yang telah diberikan oleh Allah SWT, maka Allah akan menimpakan siksa kepadanya. Menggunakan suatu rezeki, harta/biaya tertentu dengan tidak boros dan tepat merupakan gambaran dari ayat tersebut, dimana optimasi dan efisiensi biaya dalam operasional suatu kegiatan usaha menjadi penerapannya.

Aplikasi ini berbasis pada platform Android. Menurut (Safaat 2012:3) Android merupakan generasi baru *platform mobile, platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Android merupakan *platform mobile* pertama yang Lengkap, Terbuka, dan Bebas.

1. Lengkap (*Complete Platform*): para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform*): *Platform* Android disediakan melalui lisensi *open source.* Pengembang dapat dengan bebas untuk mengembangkan aplikasi.
3. Bebas (*Free Platform*): Android adalah *platform/aplikasi* yang bebas untuk develop. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak diperlukan biaya keanggotaan, tidak diperlukan biaya pengujian, dan tidak ada kontrak yang diperlukan.

Aplikasi untuk android dapat didistribusikan dan diperdagangkan dalam bentuk apa pun.

Algoritma *Tabu Search* digunakan dan diimplementasikan dalam aplikasi ini untuk menentukan rute terbaik dan jarak tempuh tercepat yang akan dikunjungi oleh kendaraan pengirim. Sehingga dari panjang rute yang telah didapatkan tadi, dapat diketahui berapa perhitungan biaya yang diperlukan selama pengiriman barang tersebut. *Tabu Search* adalah salah satu prosedur metaheuristik tingkat tinggi untuk penyelesaian permasalahan optimasi kombinatorial. TS ini dirancang untuk mengarahkan metode-metode lain (atau komponen-komponen TS itu sendiri) untuk keluar atau menghindari dari masuk dalam solusi optimal yang bersifat lokal. Kemampuan TS dalam menghasilkan klasik dan praktis dari berbagai bidang mulai bidang penjadwalan hingga bidang telekomunikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, dapat diambil suatu permasalahan yaitu, "Bagaimana membuat aplikasi untuk optimasi rute dan biaya operasional pada perusahaan ekspedisi barang menggunakan algoritma *tabu search* berbasis android?".

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Visualisasi peta menggunakan Google Maps API.

2. Penyimpanan data menggunakan database SQLite.
3. Pada penelitian ini tidak membahas bagaimana menangani database.
4. Pengiriman barang dengan sistem eceran, yaitu satu kendaraan untuk beberapa tujuan maksimal hanya 4 kota tujuan.
5. Pengiriman barang hanya melalui darat dan khusus pada pulau Jawa.
6. Kendaraan yang digunakan dalam kondisi layak dan baik.
7. Rute/ jalan yang ditempuh kendaraan dalam kondisi normal.
8. Durasi waktu perjalanan distribusi logistik tidak diperhitungkan.
9. Rute yang dipilih dalam *driving direction* di Google Map, diasumsikan dapat dilewati oleh truk distribusi.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat aplikasi optimasi rute dan biaya operasional pada perusahaan ekspedisi barang PT. ANTESS menggunakan algoritma *Tabu Search* Berbasis Android.

1.5 Manfaat Penenlitian

1.5.1 Bagi Peneliti

Menambah wawasan keilmuan dan ketrampilan dalam membangun sebuah aplikasi optimasi menggunakan *platform* android serta dapat mengetahui penerapan dari metode yang digunakan.

1.5.2 Bagi Perusahaan

Membantu perusahaan ekspedisi barang PT. ANTESS khususnya dalam hal efektifitas pengiriman barang sehingga bisa dihasilkan rute dengan biaya operasional standar.

1.6 Sistematika Penulisan

Penulisan penelitian ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan, dan metode penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang teori-teori yang terkait dengan permasalahan yang diambil.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan analisa dari kebutuhan sistem untuk membuat aplikasi yang meliputi spesifikasi kebutuhan *software* dan langkah-langkah pembuatan aplikasi optimasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang implementasi dari aplikasi yang dibuat secara keseluruhan. Serta melakukan pengujian terhadap aplikasi yang dibuat untuk mengetahui aplikasi tersebut telah dapat berjalan dengan baik dan dapat memberikan solusi dari permasalahan yang dihadapi.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat untuk pengembangan dari aplikasi selanjutya.

1.7 Metode Penelitian

Pembuatan skripsi ini terbagi menjadi beberapa tahap pengerjaan, yaitu sebagai berikut:

1. Pengumpulan data-data yang diperlukan

   Berikut merupakan beberapa metode yang digunakan dalam pengumpulan data:

   a. Studi Literatur

      Pada metode ini penulis akan melakukan pencarian pembelajaran dari berbagai macam literatur dan dokumen yang menunjang pengerjaan skripsi ini khususnya yang berkaitan dengan aplikasi optimasi menggunakan metode *tabu search*. Serta implementasi Google Map API pada *platform* android.

   b. Observasi

      Melakukan pengamatan terhadap data yang diteliti, melakukan *interview* dengan pihak yang berkaitan (staff PT. ANTESS) dengan pembuatan aplikasi optimasi ini.

   c. *Browsing*

      Melakukan pencarian ke berbagai macam website di internet yang menyediakan informasi dan literatur yang relevan dengan permasalahan dalam pembuatan aplikasi ini.

2. Analisis data yang telah dikumpulkan

   Membuat analisa data yang diperoleh dari hasil observasi, sesuai dengan kebutuhan dalam proses pembuatan aplikasi optimasi ini.

3. Perancangan Sistem

Memahami rancangan kerja optimasi rute dan biaya operasional pada android sesuai data yang ada dan mengimplementasikan modela yang diinginkan oleh pengguna.

4. Pembuatan Aplikasi

Tahap ini merupakan tahap pembuatan dan pengembangan aplikasi sesuai dnegan sistem yang sudah didesain sebelumnya. Aplikasi ini dibangun pada *platform* Android dengan menggunakan bahasa pemrograman XML dan Java.

5. Ujicoba dan Evaluasi

Tahap ini sampai pada pengujian seluruh spesifikasi terstruktur dan sistem secara keseluruhan. Uji coba sistem dilakukan setelah aplikasi selesai dibangun dengan menggunakan kuesioner. Proses ini dilakukan guna untuk memastikan, apakah aplikasi sudah berjalan dengan baik dan sesuai dengan sistem yang telah dirancang atau belum.

6. Penyusunan Buku Skripsi

Tahap ini merupakan tahap terakhir, yaitu melakukan dokumentasi terhadap pelaksanaan penelitian ini. Diharapkan, buku skripsi ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sistem optimasi rute dan biaya menggunakan algoritma *tabu search* berbasis android ini lebih lanjut maupun pada kasus yang lain.

BAB II
TINJAUAN PUSTAKA

2.1 Optimasi

Menurut Intan & Arifin (2010:9-10), Optimasi adalah proses pencarian satu atau lebih penyelesainan layak yang berhubungan dengan nilai-nilai ekstrim dari satu atau lebih nilai objektif pada suatu masalah sampai tidak terdapat solusi ekstrim yang ditemukan. Ketika satu masalah model optimasi dalam proses pencariannya hanya mempertimbangkan satu nilai objektif semata, maka proses pencarian nilai optimalnya dikenal sebagai *single objective optimization problems* atau SOOPs. Sementara suatau masalah pencarian bentuk optimum dengan melibatkan *objective* lebih dari satu, maka penentuan nilai solusi-solusi optimalnya dikenal sebagai *multi-objective optimization problems* atau MOOPs. Tantangan MOOPs adalah berhubungan dengan nilai *goodness of fit* dari suatu solusi, karena semua solusi mempunyai daerah sendiri sebagai nilai *fitness value.*

Masalah optimasi dapat dikategorikan ke dalam dua kelas besar, yaitu optimasi tanpa pembatas (*unconstrained optimization*) dan optimasi dengan pembatas (*constrained optimization*). Dari namanya kita bisa mengetahui bahwa optimasi tanpa pembatas hanya melibatkan fungsi tujuan, tidak ada pembatas (*constraint*), sedangkan optimasi dengan pembatas, selain fungsi tujuan kita juga mempunyai tambahan pembatas yang membuat permasalahan lebih rumit. Dalam *constrained optimization*,

dengan adanya pembatas diperlukan algoritma yang berbeda untuk menyelesaikannya.

Prosedur pemecahan masalah optimasi adalah memodelkan persoalannya ke dalam sebuah program matematis dan kemudian memecahkannya dengan menggunakan teknik-teknik atau metode optimasi seperti program linier, program nonliniar, program tujuan ganda, dan metode-metode lainnya yang sudah berkembang saat ini. Dalam penelitian ini akan menggunakan algoritma tabu search untuk menyelesaikan persoalan optimasi.

Optimasi rute didasarkan pada jarak paling efektif (jarak terpendek) yang akan ditempuh oleh kendaraan dari satu lokasi ke lokasi selanjutnya sehingga semua titik dapat dilayani. Kendaraan harus berangkat dari suatu depot dengan rute tertentu untuk memenuhi *demand* (permintaan) node-node dalam rute tersebut dan kembali ke depot semula. Total *demand* dari kota-kota dalam suatu rute tidak boleh melebihi kapasitas kendaraan yang bertugas melayani rute tersebut. Selain itu setiap kendaraan mempunyai waktu operasional maksimum tertentu dimana waktu tersebut merupakan waktu maksimal bagi suatu kendaraan untuk kembali ke depot.

Dalam optimasi, parameter biaya yang digunakan dalam menentukan besar biaya operasional adalah panjang rute pengiriman, jumlah dan jenis kendaraan yang digunakan, serta kapasitas tangki maksimal setiap kendaraan. Sehingga akan diketahui kebutuhan bahan bakar kendaraan berdasarkan rasio kebutuhan bahan bakar kendaraan

dengan jenis tertentu serta harga bahan bakar yang digunakan kendaraan tersebut per liter. Biaya perjalanan akan dihitung berdasarkan persamaan berikut:

$$C = \frac{Total\ Jarak\ Tempuh\ Kendaraan}{Rasio\ Kebutuhan\ Bahan\ Bakar}\ x\ Harga\ Bahan\ Bakar/liter$$

Ketika semakin panjang jarak yang ditempuh kendaraan dalam proses pendistribusian barang, maka semakin besar pula biaya operasional yang dikeluarkan. Biaya tersebut juga akan kalkulasikan dengan biaya tambahan menurut standar operasi pada PT. Antaran Express.

2.2 Perusahaan Ekspedisi Barang PT. ANTESS (Antaran Express)

Perusahaan ekspedisi barang PT. ANTESS merupakan suatu perusahaan yang bergerak dibidang transportasi logistik, *cargo management*, jasa pergudangan dan distribusi yang sesuai dengan kebutuhan pelanggan. Dalam perusahaan ini pengiriman barang di konsolidasi berdasarkan daerah tujuan dengan menggunakan kontainer dan diangkut menggunakan truk yang sesuai dengan dimensi, besar dan keperluan dari kontainer tersebut. Di Indonesia kontainer dikenal dengan nama peti kemas yang terbuat dari bahan logam dan memiliki beberapa macam ukuran dan tipe. Peti kemas atau kontainer dapat dikatakan sebagai "*the moving gedown*" yaitu gudang mini yang bergerak dari suatu tempat ke tempat lain sebagai akibat dari adanya pengangkutan.

Kontainer dibuat untuk memuat dan mengangkut semua jenis barang produksi industri maupun agraris, dan menciptakan daya tampung muatan dalam 1 (satu) peti kemas yang memiliki volume besar, dan dapat diangkut dengan cepat dan mudah dari kapal ke truk atau sebaliknya. Untuk jenis-jenis barang agraris yang masih belum diolah dan masih rendah nilainya kurang tepat menggunakan kontainer, kecuali kopi, teh, kulit hewan dan lain-lain; mengingat bahwa harga sewa kontainer cukup mahal. Kontainer dapat juga dipergunakan untuk mengangkut barang-barang kering atau muatan kelontong (*general cargo*) seperti teh, tekstil, alat-alat radio, optik, *personal effect*, juga dapat melakukan angkutan muatan jenis (*bulk cargo*), muatan cair atau muatan dingin.

Peti kemas dapat menyederhanakan proses bongkar muat, menurunkan biaya dan meningkatkan penggunaan kapasitas peralatan angkutan. Peti kemasdiatakan sebagai peralatan sistem pengangkutan (*working tool*) yang bersifat padat modal. Melalui penggunaan peti kemas dapat terjadi kerja sama dan bentuk gabungan berbagai jenis alat angkutan. Alat angkutan pada kenyataannya berfungsi menjadi penggerak peti kemas.

Dalam pengiriman barang dengan kontainer terdapat dua cara angkutan, yaitu (Nasution, 2004):

1. CY System, dalam cara CY (*Container Yard*) sistem, para *consignee* (penerima barang yang tertulis di dalam dokumen perjalanan) dapat menerima barangnya langsung dari tempat alamat/gudang penerima terutama bagi muatan golongan *household effects*, yaitu barang-

barang alat rumah tangga yang dimasukkan ke dalam kontainer di tempat/alamat rumah si pemilik barang, kemudian diangkut dengan *trailler truck* ke pelabuhan muat. Kemudian setelah kontainer dikunci atau disegel dan telah dibukukan pada perusahaan pelayaran, maka kontainer dimuat ke atas kapal.

Pihak pelayaran dengan sistem ini tidak berhak melakukan *stripping* (pembongkaran) di manapun atas kontainer dan isinya kecuali sudah memperoleh izin dari pemilik barang, kemudian diangkut dengan *trailler truck* kontainer yang telah kosong. Dalam kasus lain, kontainer baru dapat dibuka jika terdapat kecurigaan instansi yang berwenang, misalnya petugas bea cukai pelabuhan yang bersangkutan.

2. CFS System, dengan CFS (*Container Freight Station*) sistem, muat kontainer diangkut dengan kondisi "*from godown port till godown port*". Pengiriman dengan cara ini berarti si penerima barang hanya dapat menerima barangnya di gudang pelayaran di pelabuhan, maka *stripping* dapat dilakukan perusahaan pelayaran untuk disimpan di gudang dan kemudian akan diambil dari gudang yang bersangkutan oleh *consignee.*

Kemudian dalam cara pengangkutan barang melalui jalur darat & penyeberangan, perusahaan ini menggunakan dua sistem sebagai berikut:

3. Sistem *Full Container Loaded*, dengan sistem ini, kontainer harus dimasukkan atau dipadatkan menjadi 1 (satu) partai atau lebih, akan tetapi untuk hanya satu alamat penerima di pelabuhan tujuan. Dengan sistem ini seorang *shipper* menyewa kontainer dan mengirim ke alamat untuk 1 penerima, yaitu untuk 1 atau beberapa partai barang yang dipadatkan dalam 1 kontainer. Jadi, untuk cara kondisi pengiriman tersebut, kontainer yang telah berisi muatan dari *bonded warehouse*, *shipper* akan menuju CY dan menunggu pengkapalannya saja.
4. Sistem *Less Than Container Loaded*, dengan sistem ini maka kontainer hanya dapat dipadatkan barang-barang yang terdiri dari beberapa *shipper* dan atau *consignee* dengan cara dimana *shipper* mengirim barangnya ke CFS yaitu lapangan timbun yang dekat dengan dermaga dimana kapal yang bersangkutan akan bertambat dengan menggunakan truk angkutan lainnya. Setelah semua kontainer terkumpul di CFS dan kemudian melalui prosedur bea cukai maka barang tersebut dimuat (*stuffing*) ke dalam kontainer sesuai dengan *destination* dari barang tersebut. Dengan sistem ini "*door to door service*" dapat terlaksana dengan baik.

2.3 Algoritma *Tabu Search*

Tabu Search (TS) pertama kali diperkenalkan oleh Glover sekitar tahun 1986. Glover menyatakan bahwa TS adalah salah satu prosedur metaheuristik tingkat tinggi untuk penyelesaian permasalahan optimasi kombinatorial. TS ini dirancang untuk mengarahkan metode-metode lain (atau komponen proses TS itu sendiri) untuk keluar atau menghindari dari masuk dalam solusi optimal yang bersifat lokal. Kemampuan TS dalam menghasilkan solusi yang mendekati optimal telah dimanfaatkan dalam beragam permasalahan klasik dan praktis dari berbagai bidang mulai bidang penjadwalan hingga bidang telekomunikasi.

Untuk menunjang sistematis dari tujuan *Tabu Search* tersebut, maka digunakanlah dua macam *tools,* yaitu *adaptif memory* dan *responsive exploration*. Dengan adanya *adaptif memory* pada *Tabu Search* ini menuntun suatu prosedur yang mampu dalam melakukan pencarian solusi yang diinginkan dengan lebih ekonomis dan efektif, sedangkan *responsive exploration* lebih menekankan pada tahapan tiap proses yang harus dilalui selama proses pencarian itu berlangsung, dimana pada setiap tahapan tersebut dipunyai suatu variabel keputusan yang akan menuntun pada tahapan selanjutnya sampai akhir proses pencarian dihentikan.

2.3.1 Konsep Dasar *Tabu Search*

Memori pada *Tabu Search* ini mempunyau dua sifat, yaitu *explicit* dan *attributive* (Intan & Arifin, 2010).

a. *Explicit memory* menyimpan *complete solution* yang umumnya menghabiskan alokasi ruang memory dan waktu, sehingga untuk menghindari hal ini, *complete solution* dikurangi sehingga hanya terdiri dari *elite solution* yang dikunjungi selama pencarian.
b. *Attributive memory* menyimpan informasi tentang atribut dari solusi yang ditemukan yang mungkin dapat berubah dari satu solusi ke solusi lain. Sedangkan penggunaan dari memori *Tabu search* ini mengacu pada *adaptif memory* seperti yang telah dijelaskan sebelumnya.

Struktur memori dalam *Tabu search* menggunakan empat prinsip utama yaitu:

a. *Recency* atau lebih lengkapnya *recency based memory*, memori yang tetap menjaga struktur/solusi yang lebih baik maka solusi ini akan tetap dipertahankan sampai ditemukannya solusi baru yang lebih baik lagi. Agar *recency* dapat bekerja dengan baik, maka didukung oleh *Tabu list*, dan dalam *tabu list* ini terdapat *tabu active*. *Tabu List* merupakan suatu set memory yang memberikan informasi tentang struktur dari solusi awal yag pernah diambil, sehingga *tabu list* akan menghindari digunakannya struktur yang sama untuk menghasilkan suatu solusi jika struktur tersebut pernah dipakai. Sedangkan *tabu active* adalah *tabu list* yang berada paling akhir, artinya bahwa solusi itu merupakan solusi dari iterasi yang paling akhir dikunjungi. Untuk membatasi *range* atau jumlah dari *tabu list* sendiri dinamakan dengan *Tabu Tenure*.

b. *Frequency* menyediakan sebuah tipe informasi yang merupakan kumpulan informasi yang telah direkam oleh *recency based memory*. Sehingga *frequency* dan *recency* dapat saling melengkapi untuk membentuk suatu informasi permanen guna mengevaluasi pergerakan/ *move* yang terjadi.
c. *Quality* adalah kemampuan untuk membedakan solusi terbaik yang dikunjungi selama pencarian atau iterasi berlangsung.
d. *Influence* mempertimbangkan efek yang terjadi dari pemilihan solusi yang dipilih selama pencarian berlangsung, tidak hanya kualitas saja dipertimbangkan melainkan juga strukturnya.

Kelebihan TS terletak pada struktur memori yang fleksibel. Struktur memori itu akan membolehkan pencarian terus dilakukan meskipun solusi yang diperoleh saat ini tidak ada yang lebih baik dari solusi terbaik yang telah diperoleh . Struktur memori tersebut juga mampu menjaga agar proses pencarian tidak jatuh pada lokal optimal yang pernah muncul pada pencarian sebelumnya. TS umumnya tidak menggunakan pembentukan kandidat solusi secara acak sebagaimana *simulated annealing* dan *genetic algorithm.* Pemilihan kandidat solusi dala TS juga tidak dilakukan secara probabilistik sebagaimana *ant colony, simulated annealing* dan *genetic algorithm*.

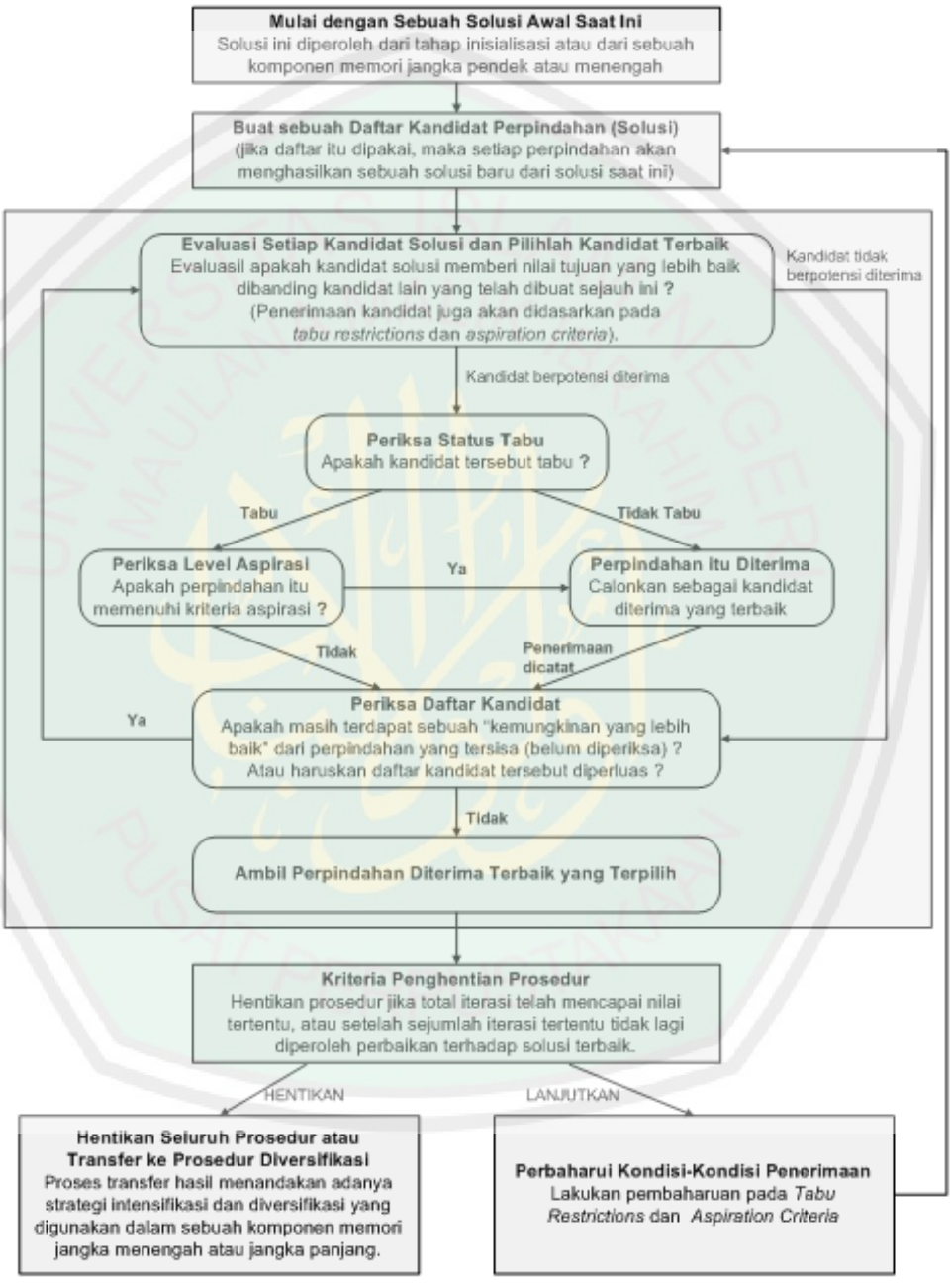
Struktur memori fundamental dalam *tabu search* dinamakan *tabu list. Tabu list* menyimpan atribut dari sebagian *move* (transisi solusi) yang telah diterapkan pada iterasi-iterasi sebelumnya. *Tabu search* menggunakan *tabu list* untuk menolak solusi-solusi yang memenuhi atribut tertentu guna

mencegah proses pencarian mengalami *cycling* pada daerah solusi yang sama, dan menuntun proses pencarian menelusuri daerah solusi yang belum dikunjungi. Tanpa menggunakan strategi ini, *local search* yang sudah menemukan solusi optimum lokal dapat terjebak pada daerah solusi optimum lokal tersebut pada iterasi-iterasi berikutnya.

Pada tiap iterasi, dipilih solusi baru yang merupakan solusi terbaik dalam *neighborhood* dan tidak tergolong sebagai tabu. Kualitas solusi baru ini tidak harus lebih baik dari kualitas solusi sekarang. Apabila solusi baru ini memiliki nilai fungsi objektif lebih baik dibandingkan solusi terbaik yang telah dicapai sebelumnya, maka solusi baru ini dicatat sebagai solusi terbaik yang baru (Priyandari, 2009).

Skema umum *Tabu Search* disajikan pada gambar dibawah ini. Pemilihan kandidat terbaik didasarkan nilai fungsi tujuan. Pemeriksaan nilai fungsi tujuan lebih didahulukan sebelum pemeriksaan status *tabu*. Apabila nilai fungsi tujuan sebuah kandidat lebih baik dari yang lain, maka kandidat tersebut berpotensi untuk diterima sehingga perlu diperiksa status tabunya. Urutan pemeriksaan nilai fungsi tujuan kemudian status tabu memberikan kemungkinan proses penyelesaian yang lebih cepat. Pemilihan kandidat solusi terbaik yang dilakukan oleh TS menggunakan prinsip *global-best strategy* (GB) bukan *first-best strategy* (FB). GB adalah strategi dimana algoritma akan mengganti solusi terbaik saat ini dengan solusi terbaik yang ada pada *negihborhood*. Adapun FB adalah strategi dimana algoritma akan

mengganti solusi terbaik saat ini secara langsung jika solusi yang lebih baik ditemukan.



Gambar 2.1. Proses Algoritma *Tabu Search* (Sumber: Priyandari, 2009)

2.3.2 Mekanisme *Tabu Search*

Sebagai sebuah algoritma, *tabu search* dalam penyelesainnya harus melewati setiap tahapan tertentu yang telah diatur, adapun tahapan-tahapan tersebut adalah:

1. Membangkitkan solusi awal

   Membangkitkan solusi awal disini berarti sebelum kita memulai tahapan *tabu search*, kita mempunyai acuan awal sebagai pembanding ketika proses *tabu search* dimulai.

2. Menentukan kriteria aspirasi (*aspiration criteria*)

   Kriteria aspirasi ini fungsinya sebagai fungsi tujuan atau *goal* yang akan dicapai, contohnya untuk penelitian ini kriteria aspirasinya adalah minimasi jarak tempuh kendaraah.

3. Melakukan *Move*

   Ada beberapa *move* yang dapat dipilih selama proses pencarian ini berlangsung:

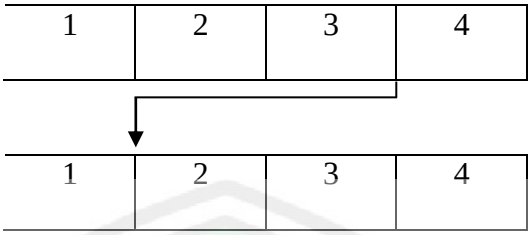
   1) *Local Search*, yang terdiri dari dua macam yaitu:

      a. Insertion: memilih secara acak satu bagian struktur untuk dipindah ke bagian yang lain.

      Contoh: Struktur awal

1	2	3	4

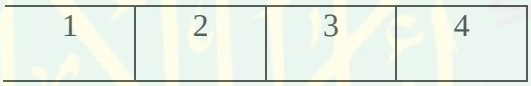
      Jika dengan proses random didapat atribut ke-3, maka struktur dapat berubah menjadi:



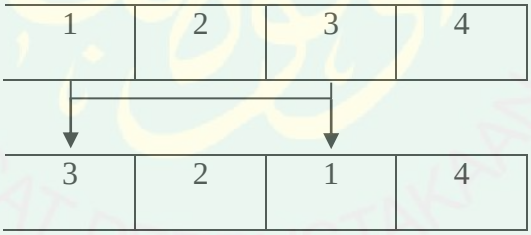
Gambar 2.2 Ilustrasi *Insertion Move*

b. Swap: memilih secara acak dua bagian struktur untuk selanjutnya ditukar posisinya.

Contoh: Struktur awal



Jika random menghasilkan 1 dan 3, maka struktur menjadi



Gambar 2.3 Ilustrasi *swap move*

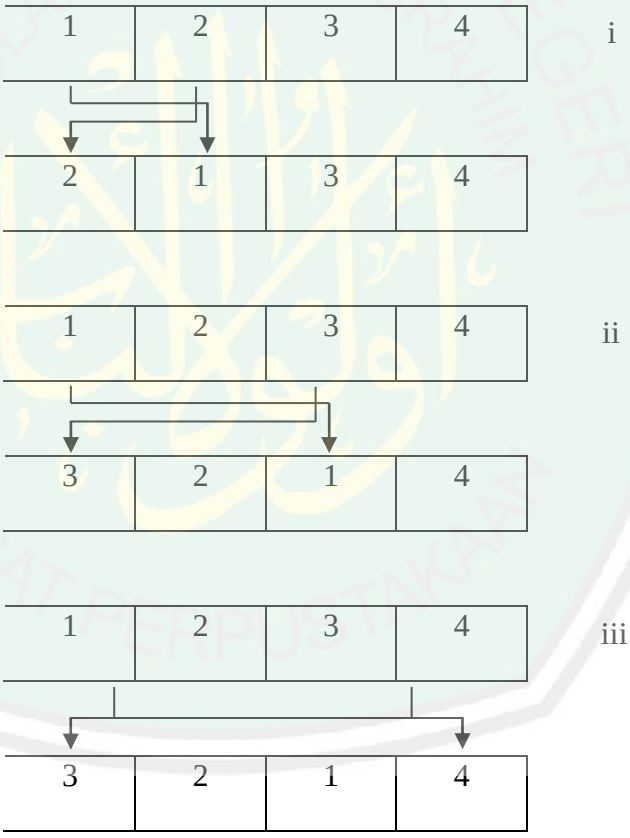
2) Neighborhood Search

Untuk pencarian dengan teknik ini setiap kemungkinan atribut dari struktur dapat dipindah-pindah. Permutasi *n-change neighborhood* mengambil n elemen dari matrik solusi (berhubungan dengan jarak masing-masing kota tujuan). Perubahan yang dipakai oleh dua *neighborhood* dengan

melakukan *swap* elemen matrik atau kombinasi elemen itu dengan menukar elemen lain dalam matrik.

Contoh: Struktur awal

Dipilih 3 *change neighborhood*, maka struktur atribut di atas dapat berubah menjadi:



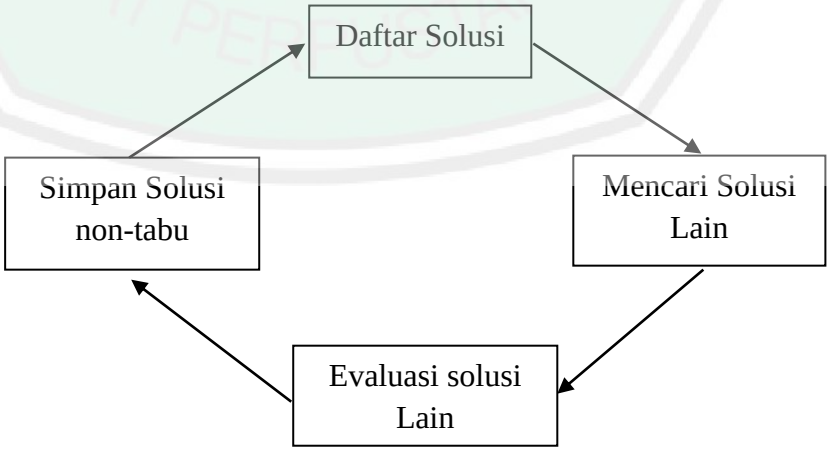
Gambar 2.4 Ilustrasi *n-Change Neighborhood Move*

4. Untuk menghindari terulangnya langkah yang diambil, maka dilakukan tabu test, Tabu test memanfaatkan tabu list yang sudah ada. Tabu list berisi atribut solusi-solusi yang telah dikunjungi

sebelumnya. Tujuan sebenarnya dari tabu list bukan untuk mencegah terulangnya langkah yang telah diambil, tetapi lebih kepada agar tidak mundur. Untuk mencegah perulangan, daftar solusi yang telah dicapai disimpan.

5. *Alternative move* yang lolos tabu test masih harus melewati *aspiration criteria* yang lebih baik daripada aspiration *treshold* atau tidak, jika tidak maka teruskan iterasi berikutnya.
6. Jika *alternative move* mempunyai *aspiration criteria* yang lebih baik daripada *aspiration threshold* maka dilakukan eksekusi terhadap *alternative move* tersebut dan memperbaharui memori yang tidak relevan.
7. Jika aturan pemberhentian sudah memenuhi syarat pemberhentian, maka pencarian berhenti.

Secara umum mekanisme *Tabu Search* dapat digambarkan sebagai berikut:



Gambar 2.5 Mekanisme Umum *Tabu Search*

2.4 Android

Menurut Safaat (2012:1), Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Android diakusisi oleh *Google* pada Juli 2005, dan baru dirilis perdana pada 5 November 2007. Android berlisensi di bawah *GNU, General Public Lisensi Versi 2 (GPLv2),* yang memperbolehkan pihak ketiga untuk mengembangkannya dengan menyertakan term yang sama. Pendistribusiannya di bawah Lisensi *Apache Software* (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya.

Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Appple pada sistem operasi Tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan di atas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, aplikasi dan *tool* pengembangan, market aplikasi android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

Gambar 2.6. *Home Screen* Android

Menurut Safaat (2012:3), Android merupakan *platform* masa depan. Android dikatakan sebagai "*platform mobile* pertama yang Lengkap, Terbuka, dan Bebas".

1. Lengkap (*Complete Platform*): Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang mengembangkan aplikasi.
2. Terbuka (*Open Source Platform*): Platform Android disediakan melalui lisensi *open source.* Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
3. *Free* (*Free Platform*): Android adalah *platform/*aplikasi yang bebas untuk develop. Tidak ada lisensi atau biaya royalti untuk dikembangkan

pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk android dapat didistibusikan dan diperdagangkan dalam bentuk apa pun.

Android merupakan generasi baru *platform mobile, platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, *General Public License* Versi 2 (GPLv2), yang sering dikenal dengan istilah "*copyleft*" lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh di bawah terms. Android didistribusikan di bawah Lisensi Apache *Software* (ASL/Apache2), yang memungkinkan untuk didistribusi kedua dan seterusnya. Komersialisasi pengembang (produsen *handset* khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka ke masyarakat *open source.* Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka dibawah lisensi apa pun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah lisensi apa pun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis android. Sebagian besar pengembang menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android. *Eclipse* adalah IDE paling populer untuk pengembangan Android,

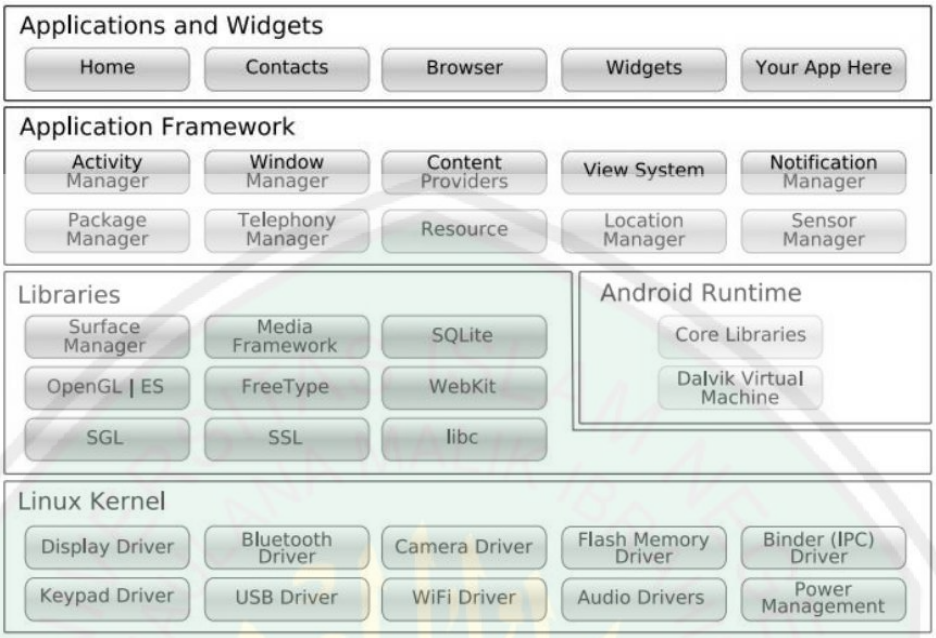
karena memiliki Android *plug-in* yang tersedia untuk memfasilitasi pengembangan Android. Selain itu, *Eclipse* juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi Android, ini terbukti dengan adanya penambahan *plugins* untuk *eclipse* untuk membuat *project* android dimana *source software* langsung dari situs resminya Google. Akan tetapi hal di atas tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti Netbeans untuk melakukan pengembangan android.

2.4.1 Arsitektur Android

Android dirancang dengan berbagai macam arsitektur sebagai komponen yang terdapat pada perangkat tersebut, sebagai berikut (Safaat, 2012:6-9):

1. *Application*dan *Widgets*, merupakan layer dimana manusia berhubungan dengan aplikasi saja, seperti aplikasi untuk browsing. Selain itu, fungsi-fungsi seperti telepon dan sms juga terdapat pada layer ini.
2. *Application Frameworks*, merupakan layer dimana para pembuat aplikasi melakukan pengembangan/ pembuatan aplikasi yang akan dijalankan di sistem operasi *Android*. Beberapa komponen yang terdapat pada layer ini adalah, *Views*, *Content Provider*, *Resource Manager*, *Notification Manager* dan *Activity Manager*.

3. *Libraries*, merupakan layer dimana fitur-fitur *Android* berada yang dapat digunakan untuk menjalankan aplikasi. *Library* yang disertakan seperti *library* untuk pemutaran audio dan video, tampilan, grafik, *SQLite*, *SSL* dan *Webkit*, dan 3D.
4. *Android Run Time*, merupakan layer yang berisi *Core Libraries* dan *Dalvik Virtual Machine* (DVK). *Core libraries* berfungsi untuk menerjemahkan bahasa Java/C. Sedangkan DVK merupakan sebuah virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.
5. *Linux Kernel*, merupakan layer yang berfungsi sebagai *abstraction*/ pemisah antara *hardware* dan *software*. *Linux kernel* inilah yang merupakan inti sistem operasi dari Android yang berfungsi untuk mengatur sistem proses, *memory*, *resouce*, dan *driver*. *Linux kernel* yang digunakan *Android* adalah *linux kernel* release 2.6.



Gambar 2.7. Arsitektur Android (Sumber: Safaat, 2012: 9)

2.4.2 Fitur Android

Android memiliki beberapa fitur baik itu fitur perangkat kerasmaupun fitur perangkat lunak yang dipakai oleh developer.

2.4.2.1 Fitur Perangkat Keras Android

Perangkat Android memiliki bebrapa fitur perangkat keras di dalamnya, yang dapat dimanfaatkan *developer* dalam membangun aplikasi.

a. *Touchscreen*

Perangkat android memiliki fitur layar sentuh (*touchscreen*) yang memberikan bebeapa kemungkinan bagi pengguna untuk berinteraksi dengan aplikasi menggunakan jari.

b. GPS

Sistem operasi android mendukung GPS yang memungkinkan *developer* untuk mengakses lokasi pengguna. Contoh aplikasi yang memanfaatkan GPS adalah aplikasi Peta (*Map*) yang menunjukkan lokasi pengguna dan memberikan petunjuk untuk menuju suatu lokasi.

c. *Accelerometer*

Android mendukung *accelerometer*, yaitu perangkat yang digunakan untuk mengukur percepatan. *Accelerometer* dapat memberitahukan apabila suatu perangkat android bergerak, terguncang, atau berbalik arah posisinya.

d. SD *Card*

Android memiliki fitur yang memungkinkan pengguna atau aplikasi untuk mengakses (menyimpan dan membuka) file pada SD *Card.*

2.4.2.2 Fitur Perangkat Lunak Android

Android memiliki banyak fitur perangkat lunak yang dapat digunakan oleh *developer* dalam mengembangkan aplikasi. Beberapa fitur yang sering dipakai antara lain:

a. *Internet*

Kemampuan akses internet pada android memberikan banyak keunggulan. Berbagai informasi secara *real-time* dapat diperoleh dengan mudah dari internet. Contoh: pengguna dapat menggunakan internet di android untuk melihat cuaca suatu area, peta, dan bahkan *download* musik.

b. *Audio and Video Support*

Sistem operasi android memungkinkan *developer* menyertakan *audio* dan *video* dalam aplikasi dengan mudah karena sistem operasi android telah mendukung beberapa standar format *audio* dan *video.*

c. *Contact*

Android memungkinkan akses ke *contacts* yang tersimpan dalam perangkat android. *Developer* dapat menggunakan fitur ini untuk menampilkan *contacts* dengan cara baru yang berbeda.

d. *Security*

Android memungkinkan aplikasi untuk melakukan banyak hal. Akan tetapi android juga menyiapkan mekanisme keamanan berupa *permission* berkaitan dengan beberapa tugas. Contoh: *download image* dan menyimpannya di SD *Card,* maka harus disetujui terlebih dahulu untuk dapat mengakses SD *Card*.

e. *Google APIs*

Sistem android memungkinkan pengguna untuk membuat panggilan telepon, mengorganisasi *contacts* atau meng-*install* aplikasi. *Developer* dapat mengintegrasikan peta (*map*) ke dalam suatu aplikasi dengan menggunakan *Maps API* yang mendukung *Map Widgets*. Berbagai fitur dapat ditambahkan dengan *Maps API,* antara lain: menampilkan suatu lokasi di peta, mendapatkan panduan navigasi, komunikasi data antara aplikasi dengan *clouds.*

2.4.3 *The Dalvik Virtual Machine* (DVM)

Menurut Safaat (2012:4), salah satu elemen kunci dari android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di dalam *Dalvik Virtual Machine* (DVM) bukan di *Java Virtual Machine* (JVM), sebenarnya banyak persamaan nya dengan *Java virtual machine* (VM) seperti Java ME (*Java Mobile Edition*), tetapi Android menggunakan *Virtual Machine* sendiri yang dikustomisasi dan dirancang untuk memastikan bahwa beberapa *feature-feature* berjalan lebih efisien pada perangkat *mobile.*

Dalvik Virtual Machine (DVM) adalah "register bebas" sementara *Java Virtual Machine* (JVM) adalah "stack based", DVM didesain dan ditulis oleh Dan Bornsten dan beberapa *engineers* Google lainnnya. Jadi, bisa kita katakan "*Dalvik equals (Java) == False*". *Dalvik Virtual Machine* menggunakan kernel Linux untuk

menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C/ C+ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataanya kita harus banyak memahami Arsitektur dan proses sistem dari kernel linux yang digunakan dalam Android tersebut.

Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* untuk mengeksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

2.4.4 Android SDK (*Software Development Kit*)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google.

Saat ini disediakan Android SDK (*Software Develeopment Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, Android memberi Anda kesempatan untuk membuat Aplikasi yan kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone.*

Beberapa fitur-fitur Android yang paling penting adalah:

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable.*
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile.*
3. *Integrated browser* berdasarkan *engine open source Webkit.*
4. Grafis yang dioptimalkan dan didukung oelh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi *hardware*).
5. SQLite untuk menyimpan data.
6. *Media Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM *Telephony* (Tergantung *hardware*).
7. *Bluetooth,* EDGE, 3G, dan WiFi (tergantung *hardware*).
8. Kamera, GPS, kompas, dan *accelerometer* (tergantung *hardware*).

9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE *Eclipse*.

2.4.5 ADT (*Android Development Tools*)

Android Development Tools adalah penghubung antara IDE Eclipse dengan Android SDK (Safaat, 2012:17). *Android Development Tools* (ADT) adalah plugin yang didesain untuk IDE *Eclipse* yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE *Eclipse*. Dengan menggunakan ADT akan memudahkan kita dalam membuat aplikasi *project* android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita juga dapat melakukan *running* aplikasi menggunakan Android SDK melalui *Eclipse*. ADT juga memungkinkan kita untuk dapat melakukan pembuatan *package* android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Eclipse dapat digunakan untuk mengembangkan aplikasi Android dengan menggunakan *Android Development Tools*, yang mempunyai fungsi:

1. Membuat, menguji, dan menyusun aplikasi Android yang berjalan di *smartphone* Android.

2. Mensimulasikan seluruh pengalaman pengguna *online* dan *offline* untuk berbagai jenis *smartphone* Android melalui desktop.
3. Memungkinkan untuk memanfaatkan keunggulan pengembangan unik untuk *platform* Android.

Gambar 2.8. Emulator Android

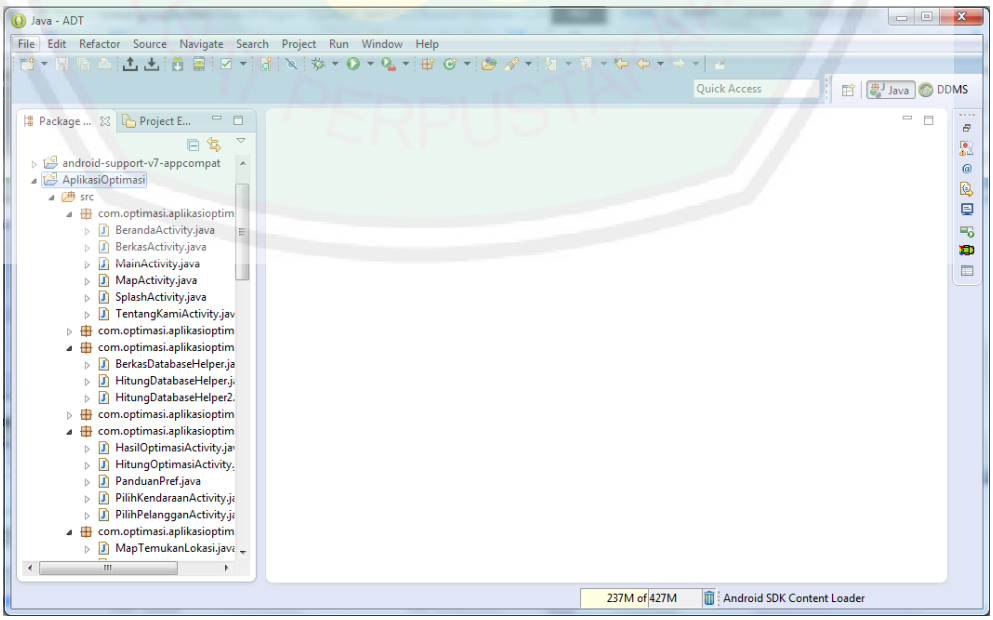
2.4.6 IDE Eclipse

Integrated Development Environment (IDE) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Eclipse adalah sebuah IDE (*Integrated Development Environtment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*).

Eclipse mempunyai beberapa sifat, antara lain:

1. *Multi-platform:* Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Multi-language:* Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role:* Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

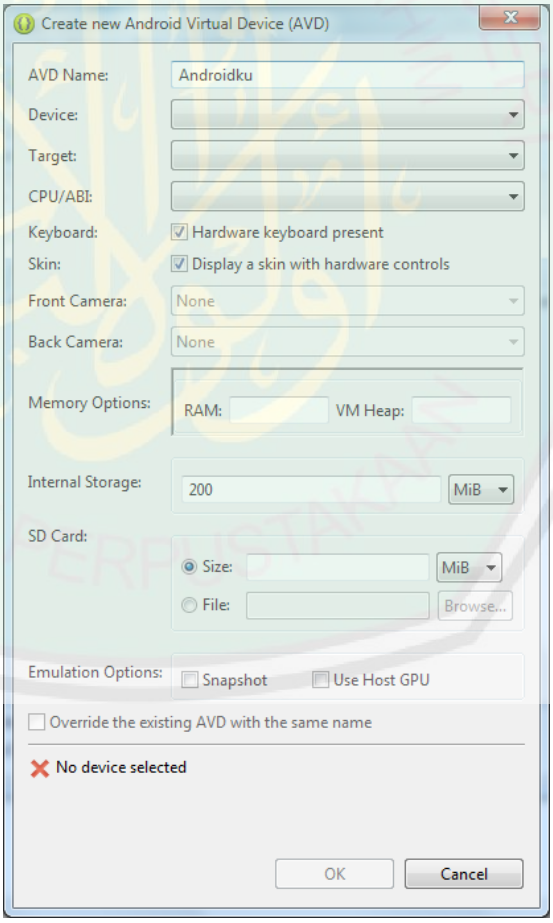
Pada pembuatan aplikasi tersebut, menggunakan *Eclipse* terbaru bawaan dari *Android Developer Tool* (ADT) versi 22, karena semakin tinggi versi IDE maka semakin lengkap fitur-fiturnya.



Gambar 2.9. ADT (*Android Development Tool*) Eclipse

2.4.7 *Android Virtual Device* (AVD)

Android Virtual Device (AVD) merupakan *emulator* untuk menjalankan program aplikasi Android yang dibuat, AVD ini nantinya yang dijadikan sebagai temapt *test* dan menjalaankan aplikasi *Android* yang dibuat, AVD berjalan di *Virtual Machine*. Untuk membuat AVD dapat dilakukan dari IDE Eclipse dengan cara menekan menu Window – Android Virtual Device Manager – kemudian pilih New.



Gambar 2.10. Setting Awal AVD Baru

Pada gambar terdapat beberapa pilihan untuk membuat *simulator* AVD, yang pertama ada kolom nama yang harus diisi. Kemudian terdapat *device*, ini harus diisi sesuai keinginan, karena banyak pilihan untuk membuat *simulator* AVD. Target di sini maksudnya adalah bahwa AVD yang dibuat harus menyesuaikan kebutuhan, dan disesuaikan dengan SDK serta versi ADT eclipse yang sudah di instalasi sebelumnya. *Size* merupakan penyedia memori untuk menjalankan AVD, karena *simulator* AVD membutuhkan memori untuk menjalankannnya, dan diisi sesuai kebutuhan dalam pembuatan aplikasi.

2.4.8 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh *tools* yang dinamakan "apt tools" ke dalam paket Android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang disebut dengan aplikasi, dan nantinya dapat di *install* di perangkat *mobile* (Safaat, 2012:9).

Ada enam jenis komponen pada aplikasi Android yaitu:

a) *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*,

tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari *activity* ke *activity* lain kita dapat melakukannya dengan satu *even*, misalnya *click* tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan *method* Activity.setContentView(). ContentView adalah objek yang berada pada root hirarki.

b) *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada pada kelas induknya. Misalnya, *media player* sedang memutar lagu dari *list* yang ada, aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu misalnya, atau menulis sms ketika *media player* sedang berjalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service. Service* dijalankan pada *thread* utama dari proses aplikasi.

c) *Broadcast Receiver*

Broadcast receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil dari *camera*, atau pengubahan referensi bahasayang digunakan. Aplikasi juga dapat menginisiasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan siap untuk digunakan.

Broadcast receiver tidak memiliki*user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

d) *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file* sistem seperti *database* SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhan oleh suatu *activity,* misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (*Map*), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka di sinilah fungsi *content provider*.

2.5 Google Maps

Google Maps merupakan layanan dari google yang mempermudah penggunaannya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat. Sedangkan Google Maps API memungkinkan pengembangan untuk mengintegrasikan Google Maps ke dalam situs website. Dengan menggunakan Google Maps API memungkinkan untuk menanamkan situs Google Maps ke dalam situs eksternal, dimana situs data tertentu dapat dilakukan overlay.

Meskipun pada awalnya hanya JavaScript API, API Google Maps sejak diperluas untuk menyertakan sebuah API untuk Adobe Flash aplikasi, layanan untuk mengambil gambar peta statis, dan layanan web untuk melakukan geocoding, menghasilkan petunjuk arah mengemudi, dan mendapatkan profil elevasi.

Kelas kunci dalam perpustakaan Maps adalah MapView, sebuah subclass dari ViewGroup dalam standar perpustakaan Android. Sebuah MapView menampilkan peta dengan data yang diperoleh dari layanan Google Maps. Bila MapView memiliki fokus, dapat menangkap tombol yang ditekan dan gerakan sentuh untuk pan dan zoom peta secara otomatis.

Google juga menyediakan layanan Google Maps API yang memungkinkan para pengembang untuk mengintegrasikan Google Maps ke dalam website masing-masing dengan menambahkan data point sendiri. Dengan menggunakan Google Maps API, Google Maps dapat ditapilkan pada website terkenal. Agar aplikasi Google Maps dapatmuncul di website

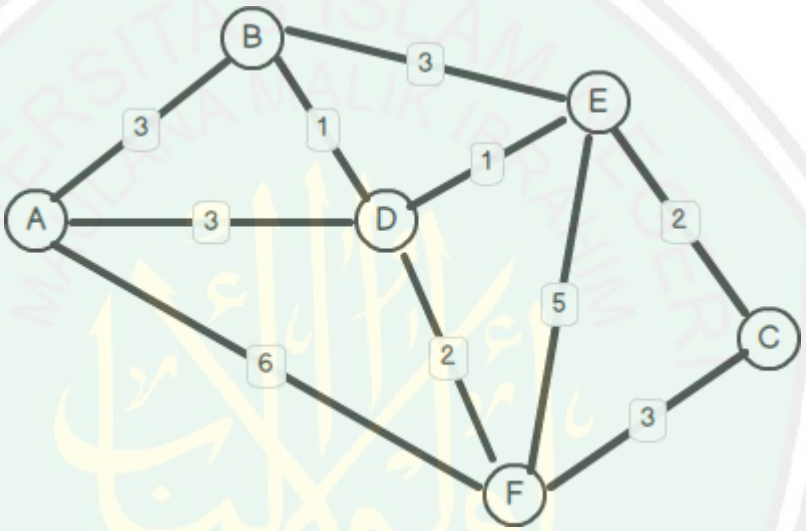
tertentu, diperlukan kode unik yang digenerasikan oleh Google untuk suatu website tertentu, agar Google Maps dapat mengenalinya.

Gambar 2.11. Tampilan Google Maps di Android

2.6 *Travelling Salesman Problem*

TSP merupakan suatu permasalahan dimana seorang *salesman* harus melewati sejumlah kota tepat satu kali dan kembali ke kota awal, yang perlu dicari adalah rute terpendek bagi *salesman* tersebut. Persoalan TSP merupakan satu persoalan optimasi kombinatorial (kombinasi permasalahan). Banyak permasalahan yang direpresentasikan dalam bentuk TSP. Diantara permasalahan yang dapat direpresentasikan dengan TSP adalah pencarian rute bis sekolah untuk mengantar siswa, pengambilan tagihan telepon, efisiensi pengiriman surat atau barang, perancangan pemasangan pipa saluran dan lain-lain.

Dalam permasalahan TSP menggunakan representasi graf untuk memodelkan persoalan yang diwakili, sehingga lebih mudah penyelesaiannya. Dengan model *weighted graph,* kota direpresentasikan sebagai verteks, dan jalurnya direresentasikan sebagai *edge.* Untuk jarak atau biaya dari suatu node ke node yang lain direpresentasikan sebagai *weight.*



Gambar 2.12. Graf Berbobot

Persoalan yang muncul adalah bagaimana cara mengunjungi node (simpul) pada graf dari titik awal ke setiap titik-titik lainnya dengan bobot minimum (biaya paling murah) dan kembali lagi ke asal node. Bobot atau biaya ini sendiri dapat mewakili berbagai hal, seperti berapa biaya minimum, jarak minimum, bahan bakar minimal, waktu minimal, kenyamanan, dan lain-lain.

2.7 Tinjauan Optimasi dari Sudut Pandang Islam

Adapun keterkaitan aplikasi yang dibuat di lihat dari sudut pandang Islam, mengenai manfaat yang terkandung dari beberapa ayat Al Qur'an dan dari Al Hadits, yaitu sebagai berikut:

Dalam Surat Al – Furqan ayat 67 :

وَٱلَّذِينَ إِذَآ أَنفَقُواْ لَمۡ يُسۡرِفُواْ وَلَمۡ يَقۡتُرُواْ وَكَانَ بَيۡنَ ذَٰلِكَ قَوَامٗا ٦٧

Artinya : "*Dan (termasuk hamba-hamba Tuhan Yang Maha Pengasih) orang-orang yang apabila membelanjakan (harta), mereka tidak berlebihan, dan tidak (pula) kikir, di antara keduanya secara wajar".*

Dalam tafsir Al Jalalain menyebutkan bahwa sifat 'ibadurrahman (hamba Allah yang beriman) adalah ketika mereka berinfak pada keluarga mereka tidak berlebihan dan tidak pelit. Mereka membelanjakan harta mereka di tengah-tengah keadaan berlebihan dan meremehkan. Intinya infak mereka bersifat pertengahan.

Ibnu Katsir menjelaskan bahwa sifat 'ibadurrahman adalah mereka tidak mubadzir (boros) kala membelanjakan harta mereka, yaitu membelanjakannya diluar hajat (kebutuhan). Mereka tidak bersifat lalai sampai mengurangi dari kewajiban sehingga tidak mencukupi. Intinya mereka membelanjakan harta mereka dengan sifat adil dan penuh kebaikan. Sikap yang paling baik adalah sifat pertengahan, tidak terlalu boros dan tidak bersifat kikir.

Hal ini juga sesuai dengan firman Allah SWT dalam QS. Al Isra': 29

وَلَا تَجۡعَلۡ يَدَكَ مَغۡلُولَةً إِلَىٰ عُنُقِكَ وَلَا تَبۡسُطۡهَا كُلَّ ٱلۡبَسۡطِ فَتَقۡعُدَ مَلُومٗا مَّحۡسُورًا ٢٩

Artinya : *"Dan janganlah kamu jadikan tanganmu terbelenggu pada lehermu dan janganlah kamu terlalu mengulurkannya karena itu kamu menjadi tercela dan menyesal."*

Maksud ayat ini adalah jangan terlalu pelit dan jangan terlalu pemurah (berlebihan). Al-Hasan al-Bashri berkata: "Tidak ada istilah berlebihan dalam berinfaq di jalan Allah." Iyas bin Mu'awiyah berkat: "Apa yang dibolehkan dalam (melaksanakan) perintah Allah Ta'ala adalah berlebihan (dalam infaq)." Selainnya berkata: "Istilah berlebih-lebihan dalam membelanjakan harta hanya untuk maksiat kepada Allah SWT." (Terjemahan Tafsir Ibnu Katsir 127-129).

Dalam ayat yang sama, Imam asy-Syaukani ketika menafsirkan ayat ini, beliau berkata: "Arti ayat ini: larangan bagi manusia untuk menahan (hartanya secara berlebihan) sehingga mempersulit dirinya sendiri dan keluarganya, dan larangan berlebihan dalam berinfak (membelanjakan harta) sampai melebihi kebutuhan, sehingga menjadikannya *musrif* (berlebih-lebihan/mubadzir). Maka ayat ini (berisi) larangan dari sikap *ifrath* (melampaui batas) dan *tafrith* (terlalu longgar), yang ini melahirkan kesimpulan disyariatkannya bersikap

moderat (dalam pengeluaran), yaitu sikap adil (seimbang) yang dianjurkan oleh Allah SWT.

Rasulullah SAW bersabda, yang artinya: "*Barang siapa hidup sederhana, niscaya Allah menjadikan kaya. Dan barang siapa boros niscaya Allah menjadikan ia miskin. Dan barang siapa mengingat kepada Allah Azza wa Jalla, niscaya Allah mencintainya*." (HR. Al Bazzar dari Thalhah).

Maka barang siapa yang menghendaki kemuliaan qana'ah (menerima/ merasa cukup apa yang ada padanya), maka seyogyanya ia menutup dari dirinya pintu-pintu keluar sepanjang ia mampu. Maka barang siapa yang banyak pengeluarannya dan lapang perbelanjaannya, niscaya ia tidak mampu akan qana'ah. Dan bersama yang demikian dibuat bagus dalam mencari rezeki dan kesederhanaan dalam penghidupan. Itu adalah pokok dalam sifat qana'ah. Dan yang kmai maksudkan dengan demikian adalah lemah lembut dalam perbelanjaan dan meninggalkan kebodohan dalam perbelanjaan. (Imam Al-Ghazali, Ihya' Ulumiddin 158).

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis dan Perancangan Sistem

3.1.1 Keterangan Umum

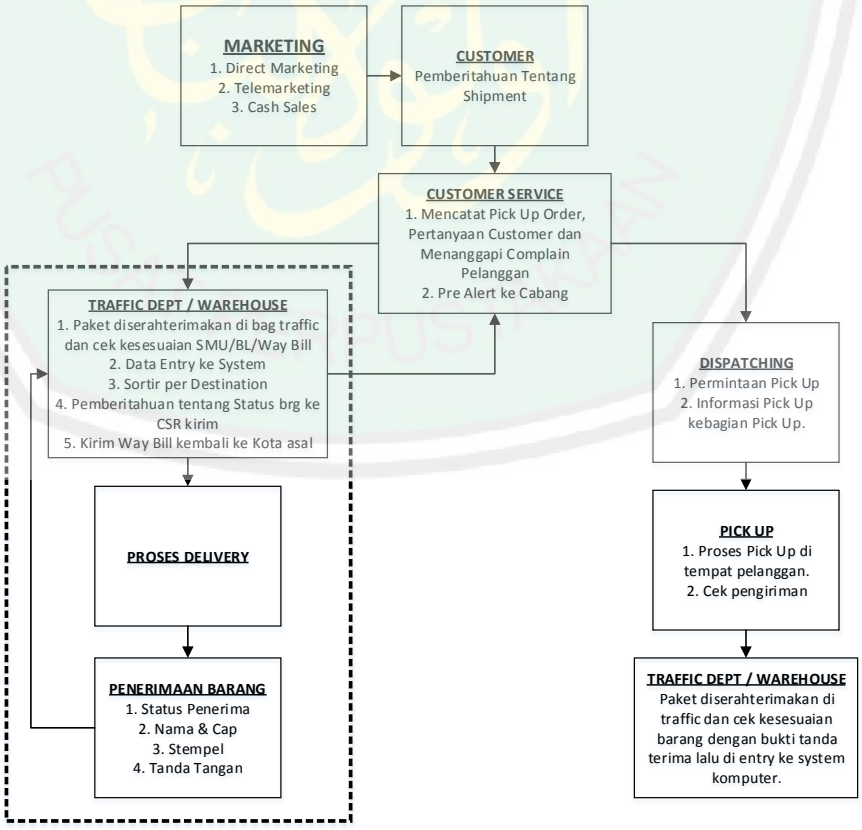
Optimasi rute menggunakan Google Maps yang khususnya dikembangkan pada perangkat *smartphone* berbasis Android bertujuan agar aplikasi ini dapat digunakan secara mudah dan bebas. Dalam artian bebas digunakan kapan saja dan dimana saja, sehingga para driver khususnya dapat mengetahui secara cepat dan tepat tentang lokasi yang akan menjadi tujuan pengiriman barang tersebut. Begitupun dengan pemilik usaha, dapat menjadikan aplikasi ini sebagai bantuan dan pertimbangan dalam hal menentukan besar biaya operasional untuk keperluan transportasi logistik tersebut.

Badan usaha yang dimaksud dalam penelitian ini yaitu PT. ANTESS (Antaran Express), merupakan perusahaan swasta nasional, yang bergerak dalam bidang transportasi logistik, *cargo management*, jasa pergudangan, dan distribusi yang sesuai dengan kebutuhan pelanggan. Efisiensi waktu merupakan hal yang penting dalam hal ini, dengan memilih dan menentukan rute mana yang akan ditempuh dengan rute pengiriman yang paling cepat, sehingga biaya operasional yang dibutuhkan pun juga dapat ditekan.

3.1.2 Gambaran Umum Aplikasi

Pada penelitian ini penulis membuat suatu aplikasi yang dapat menghitung estimasi biaya operasional yang akan dikeluarkan oleh perusahaan yang mana dalam kasus ini PT. ANTESS (Antaran Exspress) selama dalam proses pengiriman barang mulai dari *warehouse*/gudang yang merupakan lokasi awal, sampai ke berbagai tujuan pengiriman pelanggan hingga kembali lagi ke *warehouse*.

Jadi dalam prosesnya, aplikasi ini berfungsi hanya pada bagian pendistribusian barang mulai dari warehouse ke pelanggan dan kembali lagi ke warehouse atau pada proses *delivery*. Berikut merupakan alur prosedur standar operasi pada PT. ANTESS.



Gambar 3.1. Standar Operasi PT. ANTESS (Sumber: PT. ANTESS)

Perhitungan optimasi biaya ini ,berdasarkan pada panjang rute yang akan ditempuh oleh kendaraan pengangkut barang serta rasio kebutuhan bahan bakar dari setiap jenis kendaraan yang digunakan. Juga dengan perhitungan tambahan biaya lainnya, seperti biaya untuk timbang kendaraan, dan biaya perbaikan kendaraan. Pada penelitian ini penulis tidak masuk pada permasalahan berapa banyak barang yang akan diangkut dan bagaimana proses transaksinya.

Kemudian untuk optimasi rute, penulis menggunakan metode tabu search untuk mencari nilai minimum dimana dalam kasus ini nilai minimum yang dicari adalah rute tempuh terbaik. Sehingga dari nilai minimum tersebut juga akan dapat diketahui urutan lokasi tujuan distribusi yang akan dikunjungi mulai dari yang pertama hingga terakhir.

3.2 Perancangan Aplikasi

Perancangan aplikasi dilakukan untuk menggambarkan, merencanakan, dan membuat sketsa atau pengaturan dari beberapa element yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem ini merupakan hasil transformasi dari analisa ke dalam perancangan yang nantinya akan di implementasikan. Hal yang menjadi perhatian dalam perancangan sistem ini adalah bahwa sistem yang dibuat diharapkan dapat digunakan dengan mudah oleh para pengguna, sehingga sistem dapat digunakan secara menyeluruh. Selain itu aplikasi yang dibuat diharapkan dapat menjadi pertimbangan bagi perusahaan yang terkait dalam

mengkalkulasikan biaya operasional dalam proses ekspedisi darat khususnya di pulau jawa.

Aplikasi optimasi ini secara garis besar terdiri dari beberapa bagian, yaitu menghitung rute pengiriman terbaik, menghitung estimasi biaya operasional berdasarkan rute tempuh kendaraan, menampilkan/ memvisualisasikan rute terbaik hasil perhitungan tersebut.

1. Perhitungan Jarak dan Rute Distribusi

   Proses perhitungan rute tempuh kendaraan distribusi dilakukan mulai dari depot/*warehouse* PT. ANTESS ke tempat para pelanggan hingga kendaraan kembali lagi ke depot/*warehouse* tersebut dengan menggunakan algoritma *Tabu Search*. Rute yang akan ditempuh akan selalu berubah sesuai dengan lokasi tujuan mana saja dari pengiriman barang tersebut.

2. Perhitungan Estimasi Biaya Distribusi

   Estimasi biaya operasional distribusi logistik akan dikalkulasikan setelah rute pengiriman berhasil didapatkan. Parameter yang digunakan dalam menentukan besar biaya operasional adalah panjang rute pengiriman dan jenis kendaraan yang sehingga akan diketaui kebutuhan bahan bakar kendaraan berdasarkan rasio kebutuhan bahan bakar kendaraan jenis tertentu per kilometernya, serta harga bahan bakar yang akan digunakan kendaraan tersebut per liter.

3. Visualisasi Rute Distribusi

Rute terbaik pendistribusian barang yang telah didapatkan dari hasil perhitungan pada langkah pertama, akan divisualisasikan dengan menggunakan salah satu fitur dari Google Maps API yaitu *Get Direction*. Sehingga Google Maps bisa menampilkan rute tersebut sesuai dengan permintaan dan hasil perhitungan dari sistem optimasi.

3.2.1 Rancangan Penggunaan Aplikasi

Terdapat beberapa menu dalam aplikasi optimasi ini, sehingga pengguna dapat memilih salah satu dari menu tersebut sesuai dengan fungsi yang dibutuhkan. Menu tersebut antara lain, Menu Hitung Optimasi dimana pada menu ini pengguna dapat menghitung biaya operasional untuk proses distribusi barang, dengan memasukkan inputan berupa jenis kendaraan yang digunakan dan lokasi-lokasi dari distribusi logistik tersebut.

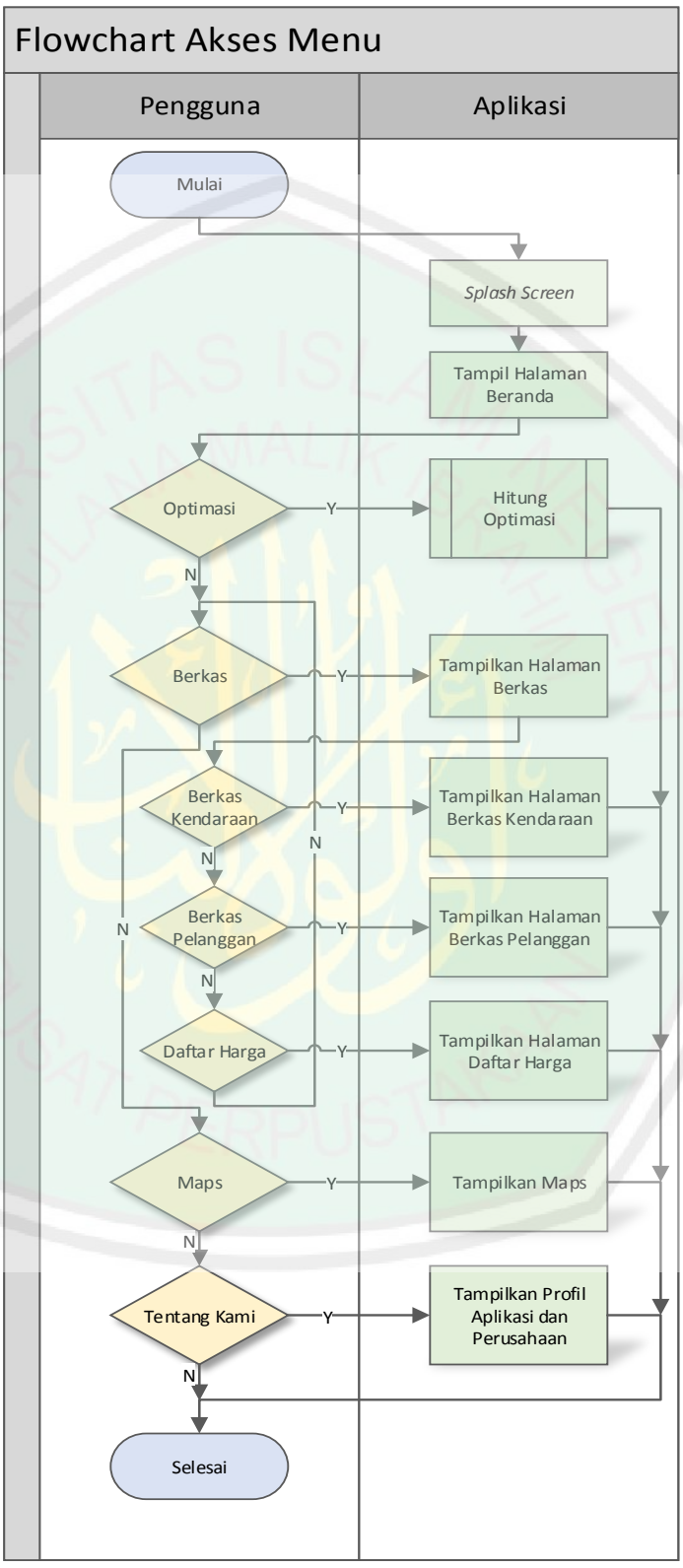
Menu Berkas, pada menu ini terdapat tiga sub menu lagi, yaitu sub menu Berkas Kendaraan, Berkas Pelanggan, dan Berkas Daftar Harga. Pada Berkas Kendaraan pengguna akan mendapatkan informasi mengenai data-data atau informasi tentang kendaraan apa saja yang dimiliki perusahaan (PT. ANTESS) untuk proses pengiriman logistik. Pada menu Berkas Pelanggan, pengguna akan mendapatkan informasi tentang data-data pelanggan yang secara garis besar berisi tentang identitas pelanggan termasuk lokasi alamat pelanggan tersebut. Pada menu berkas yang ketiga, pengguna dapat melihat daftar harga untuk biaya pengiriman logistik

berdasarkan kota tujuan dan besaran volume atau kendaraan yang akan digunakan dalam proses distribusi.

Menu selanjutnya yaitu Menu Map, fasilitas ini dimaksudkan untuk mempermudah ketika pengguna ingin melakukan pencarian rute/ jalur menuju lokasi tertentu dengan menggunakan bantuan Google Map. Jadi ketika pengguna ingin menggunakan fasilitas *driving direction*, pengguna tidak perlu membuka dua program/aplikasi sekaligus, aplikasi optimasi dan aplikasi Maps yang biasa disediakan bawaan dari android. Sehingga akan lebih efektif dan akan lebih menghemat memori dan tentu saja lebih menghemat daya.

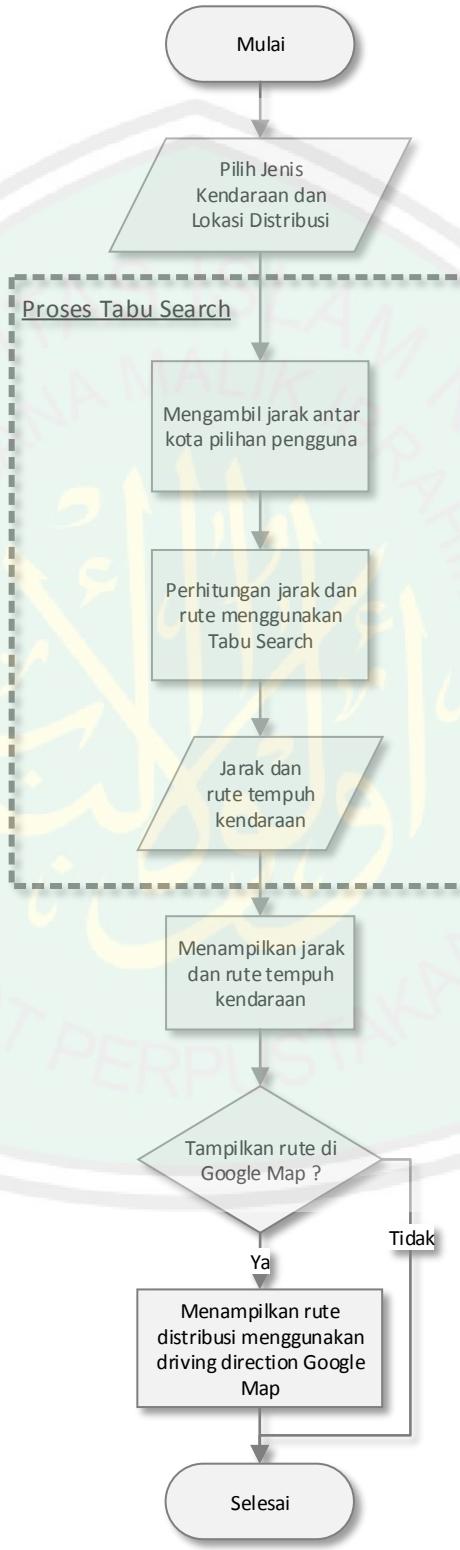
Menu terakhir dalam aplikasi ini adalah Menu Tentang Kami, pada menu ini pengguna akan mendapatkan informasi mengenai profil dari Aplikasi Maksimus, aplikasi optimasi rute dan biaya operasional ini. Selain itu, pada tab kedua terdapat informasi tentang profil singkat perusahaan jasa transportasi logistik PT. ANTESS (Antaran Express), alamat kantor pusat PT. Antaran Express (ANTESS), alamat cabang, dan alamat gudang yang mana merupakan lokasi awal dan terakhir ketika proses *delivery* logistik terjadi. Juga terdapat fasilitas temukan lokasi untuk *warehouse* PT. ANTESS sehingga pelanggan dan pengguna dapat mengetahui dengan pasti lokasi dari gudang PT. ANTESS tersebut.

Berikut ini merupakan alur pengguna ketika mengakses menu-menu yang telah disediakan pada aplikasi optimasi Maksimus:



Gambar 3.2 *Flowchart* Akses Menu Aplikasi Optimasi

3.2.2 Rancangan Perhitungan Jarak dan Rute Distribusi



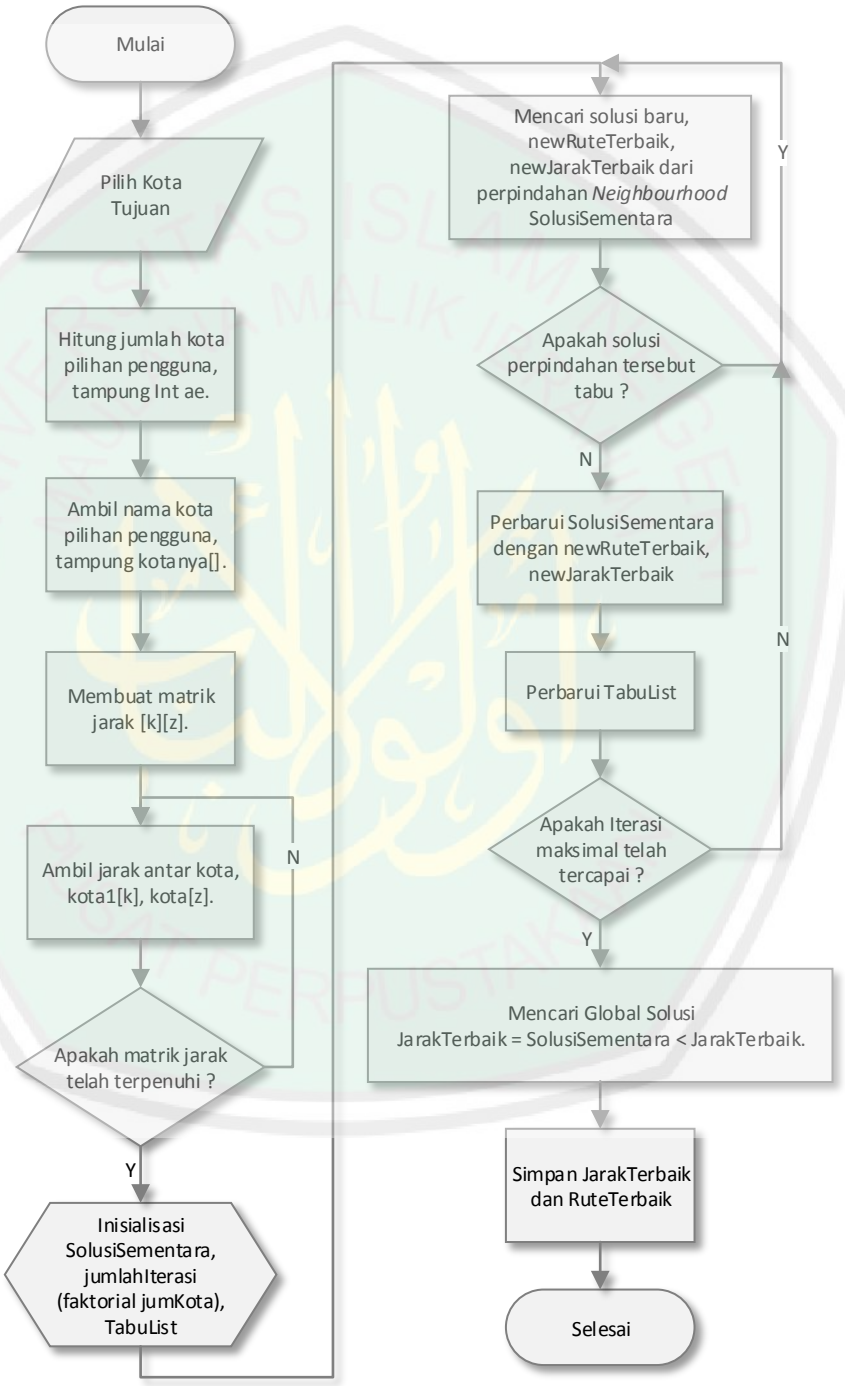
Gambar 3.3 *Flowchart* Perhitungan Jarak dan Rute Distribusi

Proses perhitungan optimasi jarak dan rute distribusi dijalankan ketika pengguna telah mengkonfirmasi data di halaman preview data, dan menekan tombol mulai hitung. Dimulai dengan pengguna memilih kota tujuan distribusi, untuk mendapatkan jarak antar kota pilihan pengguna tersebut, yang kemudian jarak-jarak tersebut akan diolah menggunakan algoritma *tabu search* untuk memperoleh jarak tempuh terpendek dengan urutan rute kota untuk dikunjungi kendaraan distribusi.

Sebelumnya jarak antar kota telah diketahui dan disimpan di dalam database. Jarak tersebut didapatkan dengan cara menggunakan bantuan *driving direction* di Google Maps, sehingga akan diketahui berapa kilometer jarak tempuh dari satu kota ke kota lainnya. Sesuai dengan kota tujuan yang didefinisikan oleh PT. ANTESS sebagai kota tujuan distribusi, terdapat 25 lokasi kota tujuan termasuk *Warehouse* PT. ANTESS, sebagai lokasi pemberangkatan kendaraan.

Sehingga akan didapatkan 625 jarak sebanyak antar 25 kota tersebut termasuk kota yang sama/ lokasi yang sama, dan tentu saja akan bernilai 0 (nol). Namun setiap kali proses perhitungan dimulai tidak semua data jarak antar kota tersebut digunakan, sistem hanya akan mengambil jarak antar kota sesuai dengan kota pilihan pengguna pada awal proses perhitungan optimasi. Jarak yang telah diambil tersebut akan dijadikan isi dari matrik 2 dimensi, dan kemudian akan dilakukan perpindahan hingga mendapatkan solusi optimal dari proses perhitungan menggunakan algoritma *tabu search*.

Berikut ini merupakan rancangan alur proses algoritma *tabu search* di aplikasi optimasi :



Gambar 3.4. *Flowchart* Proses *Tabu Search*

Perhitungan menggunakan *tabu search* dimulai dengan menerima inputan berupa kota tujuan distribusi logistik. Kemudian sistem akan menghitung jumlah kota pilihan pengguna, tujuannya yaitu nantinya akan digunakan untuk membentuk sebuah matrik jarak dengan model matrik 2 dimensi. Kemudian mengambil nama kota pilihan pengguna untuk mendapatkan kombinasi jarak antar kota tersebut dari database. Proses berikutnya yaitu membentuk matrik jarak [k][z] dan mengisinya dengan jarak yang didapat dari query database berdasarkan nama-nama kota tujuan tersebut.

Ketika matrik jarak telah terpenuhi, selanjutnya menentukan SolusiSementara dengan tujuan sebagai posisi awal untuk memulai perpindahan antar kota atau juga disebut dengan perpindahan *neighbourhood*. Iterasi ditentukan dari kombinasi perpindahan dari jumlah kotapilihan pengguna atau faktorial dari jumlah kota. Jumlah Iterasi tersebut digunakan sebagai banyaknya perulangan untuk mendapatkan nilai jarak dan solusi rute tempuh terbaik, terbaik disini berarti nilai jarak terpendek sesuai dengan urutan prioritas kota untuk dikunjungi terlebih dahulu.

Tabu list/ daftar tabu didefinisikan untuk mencegah agar proses/ perpindahan yang sedang berjalan tidak bersifat tabu, parameter tabu disini yaitu mencegah untuk tidak menghitung dengan kondisi atau perpindahan dengan lokasi dan hasil yang sama. Tabu list/ daftar tabu juga digunakan untuk memulai dari mana proses algoritma tersebut dijalankan selanjutnya,

sehingga daftar tabu tersebut lebih agar proses perpindahan dan perhitungan solusi tidak berjalan mundur atau terjebak pada solusi yang sama.

Untuk mendapatkan solusi terbaik, digunakan teknik *Neighborhood search* selama proses pencarian berlangsung. Ketika solusi baru yang ditemukan selama proses pencarian dengan teknik tersebut tidak bersifat tabu, maka SolusiSementara akan diperbarui dengan newJarakTerbaik dan newRuteTerbaik sebagai solusi terbaik saat ini. Selanjutnya status tabu daftar tabu juga diperbarui dengan solusi baru tersebut, untuk memulai proses perhitungan dengan iterasi baru yang dimulai dari solusi baru. Setelah seluruh iterasi dijalankan, sistem akan mencari Global Solusi yang didapatkan dari solusi terbaik (jarak terpendek)/ nilai terkecil dari solusi-solusi yang didapatkan dari setiap iterasi tersebut. Terakhir *tabu search* akan menyimpan solusi jarak dan rute tempuh terbaik tersebut di dalam database.

Sebagai Ilustrasi, berikut merupakan contoh kasus yang diselesaikan dengan Metode *Tabu Search* menggunakan pendekatan TSP (*Travelling Salesman Problem*) dengan input koordinat/ lokasi kota dan dimana kota asal dan kota tujuan yang telah diketahui diketahui. Jalur yang ditetapkan, dimulai dari kota ke-5, dan berakhir di kota ke-2 dengan iterasi maksimum = 6, diketahui posisi masing-masing kota:

Kota ke-	X	Y
1	10.00	0.00
2	30.00	0.00
3	15.00	20.00
4	25.00	20.00
5	10.00	30.00
6	30.00	30.00

Tabel 3.1. Ilustrasi Koordinat Masing-masing Kota

Dengan jarak antar kota dihitung dengan bentuk Euclidean:

Kota	1	2	3	4	5	6
1	0.00	20.00	20.62	25.00	30.00	36.06
2	20.00	0.00	25.00	20.62	36.06	30.00
3	20.62	25.00	0.00	10.00	11.18	18.03
4	25.00	20.62	10.00	0.00	18.03	11.18
5	30.00	36.06	11.18	18.03	0.00	20.00
6	36.06	30.00	18.03	11.18	20.00	0.00

Tabel 3.2. Ilustrasi Jarak Antar Kota

Berikut merupakan perhitungan manual menggunakan *Tabu Search* dengan maksimum iterasi = 6, diperoleh:

Iterasi ke-1 :

TABU LIST:

I : 5 1 3 4 6 2

TETANGGA (Jalur alternatif berikutnya):

* Jalur ke-1 : 5 3 1 4 6 2 = 97.98

>>> BestSoFar = 97.98

* Jalur ke-2 : 5 4 3 1 6 2 = 114.70

* Jalur ke-3 : 5 6 3 4 1 2 = 93.03

>>> BestSoFar = 93.03

* Jalur ke-4 : 5 1 4 3 6 2 = 113.03

* Jalur ke-5 : 5 1 4 3 6 2 = 112.24
* Jalur ke-6 : 5 1 3 6 4 2 = 100.44
<< BestSoFar = 93.03, jalur ke-3 - DITERIMA sebagai GlobalMin >>
<< Global Min = 93.03 >>

Iterasi ke-2 :

TABU LIST:

I : 5 1 3 4 6 2
II : 5 6 3 4 1 2

TETANGGA (Jalur alternatif berikutnya):
* Jalur ke-1 : 5 3 6 4 1 2 = 85.39
>>> BestSoFar= 85.39
* Jalur ke-2 : 5 4 3 6 1 2 = 102.11
* Jalur ke-3 : 5 1 3 4 6 2 = 101.80
>>> Jalur ada pada Tabu ke-1
* Jalur ke-4 : 5 6 4 3 1 2 = 81.80
>>> BestSoFar= 81.80
* Jalur ke-5 : 5 6 1 4 3 2 = 116.06
* Jalur ke-6 : 5 6 3 1 4 2 = 104.26
<< BestSoFar = 81.80, jalur ke-4 - DITERIMA sebagai GlobalMin >>
<< Global Min = 81.80 >>

Iterasi ke-3 :

TABU LIST :

I : 5 1 3 4 6 2
II : 5 6 3 4 1 2
III : 5 6 4 3 1 2

TETANGGA (Jalur alternatif berikutnya):
* Jalur ke-1 : 5 4 6 3 1 2 = 87.85
>>> BestSoFar= 87.85
* Jalur ke-2 : 5 3 4 6 1 2 = 88.42
* Jalur ke-3 : 5 1 4 3 6 2 = 113.03
* Jalur ke-4 : 5 6 3 4 1 2 = 93.03
* jalur ke-5 : 5 6 1 3 4 2 = 107.29

* Jalur ke-6 : 5 6 4 1 3 2 = 101.80

<< BestSoFar = 87.85, yaitu jalur ke-1 - TIDAK DITERIMA sebagai GlobalMin >>

<< Global Min = 81.80 >>

Iterasi ke-4 :

TABU LIST :

I	:	5	1	3	4	6	2
II	:	5	6	3	4	1	2
III	:	5	6	4	3	1	2
IV	:	5	4	6	3	1	2

TETANGGA (Jalur alternatif berikutnya):

* Jalur ke-1 : 5 6 4 3 1 2 = 81.80

>>> Jalurada pada Tabu ke-3

* Jalur ke-2 : 5 3 6 4 1 2 = 85.39

>>> BestSoFar= 85.39

* Jalur ke-3 : 5 1 6 3 4 2 = 114.70

* Jalur ke-4 : 5 4 3 6 1 2 = 102.11

* Jalur ke-5 : 5 4 3 6 1 2 = 111.67

* Jalur ke-6 : 5 4 6 1 3 2 = 110.88

<<BestSoFar = 85.39, jalur ke-2 - TIDAK DITERIMA sebagai GlobalMin>

<< Global Min = 81.80

Iterasi ke-5 :

TABU LIST :

I	:	5	1	3	4	6	2
II	:	5	6	3	4	1	2
III	:	5	6	4	3	1	2
IV	:	5	4	6	3	1	2
V	:	5	3	6	4	1	2

TETANGGA (Jalur alternatif berikutnya):

* Jalur ke-1 : 5 6 3 4 1 2 = 93.02

>>> Jalurada pada Tabu ke-2

* Jalur ke-2 : 5 4 6 3 1 2 = 87.85

>>> Jalur ada pada Tabu ke-4

* Jalur ke-3 : 5 1 6 4 3 2 = 112.24
>>> BestSoFar= 112.24
* Jalur ke-4 : 5 3 4 6 1 2 = 88.42
>>> BestSoFar= 88.42
* Jalur ke-5 : 5 3 1 4 6 2 = 97.98
* Jalur ke-6 : 5 3 6 1 4 2 = 110.88
<<BestSoFar = 88.42, jalur ke-4 - TIDAK DITERIMA sebagai GlobalMin>
<< Global Min = 81.80

Iterasi ke-6 :

TABU LIST :

I	:	5	1	3	4	6	2
II	:	5	6	3	4	1	2
III	:	5	6	4	3	1	2
IV	:	5	4	6	3	1	2
V	:	5	3	6	4	1	2
VI	:	5	3	4	6	1	2

TETANGGA (Jalur alternatif berikutnya):
* Jalur ke-1 : 5 4 3 6 1 2 = 102.11
>>> BestSoFar= 102.11
* Jalur ke-2 : 5 6 4 3 1 2 = 81.80
>>> Jalurada ada pada Tabu ke-3
* Jalur ke-3 : 5 1 4 6 3 2 = 109.21
* Jalur ke-4 : 5 3 6 4 1 2 = 85.39
>>> Jalurada ada pada Tabu ke-5
* Jalur ke-5 : 5 3 1 6 4 2 = 99.65
>>> BestSoFar= 99.65
* Jalur ke-6 : 5 3 4 1 6 2 = 112.24
<<BestSoFar = 99.65, jalur ke-5 - TIDAK DITERIMA sebagai GlobalMin>
<< Global Min = 81.80

Jalur terpendek yang diperoleh : 5 6 4 3 1 2
Panjang jalur terpendek : 81.80

3.2.3 Rancangan Perhitungan Estimasi Biaya Distribusi

Gambar 3.5. *Flowchart* Perhitungan Estimasi Biaya Distribusi

Untuk dapat mengkalkulasikan biaya distribusi logistik dengan tujuan kota yang telah ditentukan tersebut, dibutuhkan beberapa variabel yaitu jarak tempuh yang didapat dari hasil perhitungan menggunakan *tabu search*, rasio kebutuhan bahan bakar kendaraan sesuai pilihan pengguna, harga bahan bakar tersebut per liter, biaya timbang dan biaya perbaikan. Proses dimulai dengan menghitung biaya konsumsi bahan bakar dengan menggunakan persamaan seperti berikut:

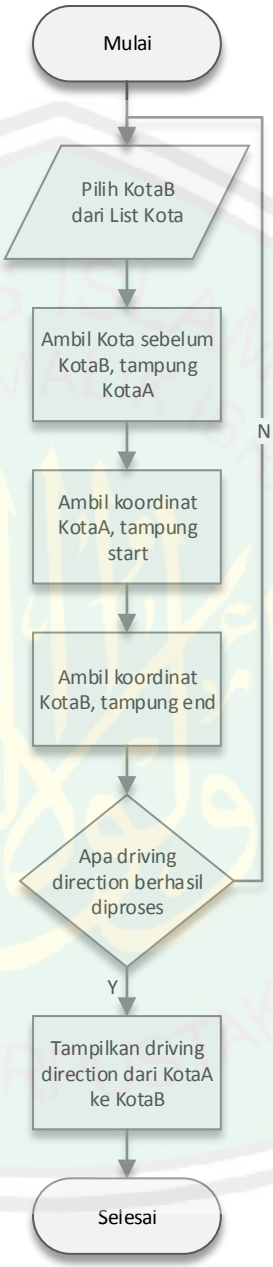
$$C = \frac{Total\ Jarak\ Tempuh\ Kendaraan}{Rasio\ Kebutuhan\ Bahan\ Bakar}\ x\ Harga\ Bahan\ Bakar/liter$$

Total jarak tempuh kendaraan akan diketahui dari hasil perhitungan dengan menggunakan algoritma *tabu search*. Rasio kebutuhan bahan bakar kendaraan sebelumnya telah didefinisikan di dalam database, karena rasio kebutuhan bahan bakar setiap kendaraan berdasarkan merknya bisa saja berbeda. Sedangkan untuk harga bahan bakar per liter disesuaikan dengan harga bahan bakar jenis solar saat ini. Biaya tambahan yaitu $Biaya\ Timbang + Biaya\ Perbaikan$ sesuai dengan pilihan pengguna. Sedangkan untuk estimasi biaya distribusi diperoleh dengan persamaan :

Estimasi biaya = $C + Biaya\ Timbang + Biaya\ Perbaikan$

Proses terakhir adalah menampilkan hasil perhitungan tersebut pada halaman Hasil Optimasi, tepat di bawah list daftar kota tujuan yang telah diurutkan sesuai dengan prioritas untuk dikunjungi oleh kendaraan distibusi terlebih dahulu.

3.2.4 Rancangan Visualisasi Rute Distribusi



Gambar 3.6. Flowchart Perancangan Visualisasi Rute Distribusi

Hasil dari perhitungan adalah daftar lokasi yang akan dikunjungi dengan pengurutan berdasarkan lokasi pertama yang akan dikunjungi terlebih dahulu sampai lokasi terakhir, dan juga akan ditampilkan estimasi

biaya yang dibutuhkan dalam proses distribusi tersebut. Selanjutnya ketika pengguna memilih salah satu kota dari daftar tersebut, sistem akan mengambil id dari kota sebelumnya dari daftar tersebut. Setelah kota sebelumnya diambil, maka sistem akan mendapatkan koordinat dari kota tersebut, dengan tujuan digunakan sebagai posisi awal/origin untuk *driving direction.* Selanjutnya sistem akan mengambil koordinat dari kota yang dipilih oleh pengguna tadi, dan digunakan sebagai tujuan/ destinasi. Apabila driving direction berhasil diproses, maka jalur akan ditampilkan dengan *polyline* berwarna biru dengan *marker* berlabel A sebagai posisi awal dan *marker* berlabel B sebagai tujuannya.

3.2.5 Perancangan Antarmuka

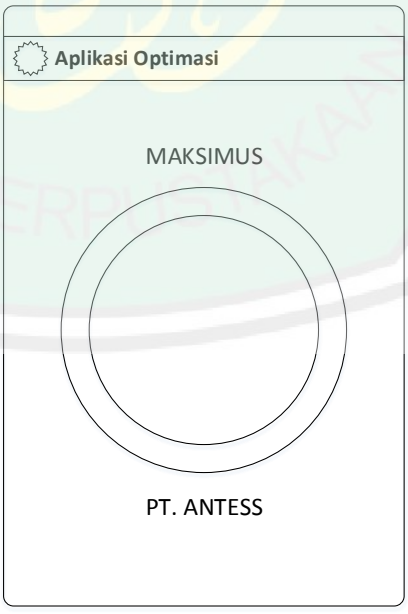
Aplikasi Optimasi rute dan biaya operasional ini memiliki beberapa halaman/ antarmuka sebagai berikut:

1. Splash Screen
2. Halaman Beranda
3. Halaman Hitung Optimasi (Panduan)
4. Halaman Hitung Optimasi (Pilih Lokasi Distribusi)
5. Halaman Hitung Optimasi (Pilih Kendaraan & Biaya)
6. Halaman Preview
7. Halaman Hitung Optimasi (Hasil Perhitungan Optimasi)
8. Halaman MapVisualisasi
9. Halaman Berkas
10. Halaman Berkas (Berkas Kendaraan)

11. Halaman Detail Berkas Kendaraan
12. Halaman Berkas (Berkas Pelanggan)
13. Halaman Detail Berkas Pelanggan
14. Halaman Berkas Daftar Harga
15. Halaman Detail Daftar Harga
16. Halaman Maps (Peta)
17. Halaman Tentang Kami (Tentang Aplikasi)
18. Halaman Tentang Kami (Tentang Profil Perusahaan)

Berikut merupakan rancangan dan penjelasan dari setiap antarmuka dalam aplikasi:

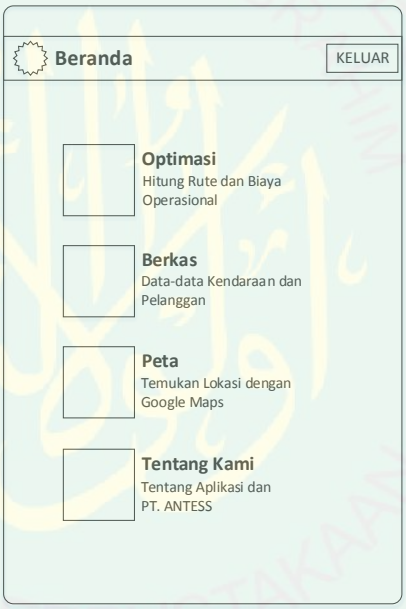
1. *Splash Screen*



Gambar 3.7. *Splash Screen*

Halaman ini merupakan halaman pertama yang akan muncul ketika aplikasi Maksimus pertama kali dijalankan. Halaman *splash screen* ditampilkan dalam beberapa detik dan secara otomatis akan langsung menuju ke halaman berikutnya, yaitu Halaman Beranda.Pada *splash screen* ini akan ditampilkan nama aplikasi kemudian terdapat logo aplikasi dan nama perusahaan PT. ANTESS (Antaran Express).

2. Halaman Beranda



Gambar 3.8. Halaman Beranda

Setelah *splah screen* selesai dijalakankan maka akan muncul halaman Beranda, halaman ini berisi empat menu utama, yaitu:

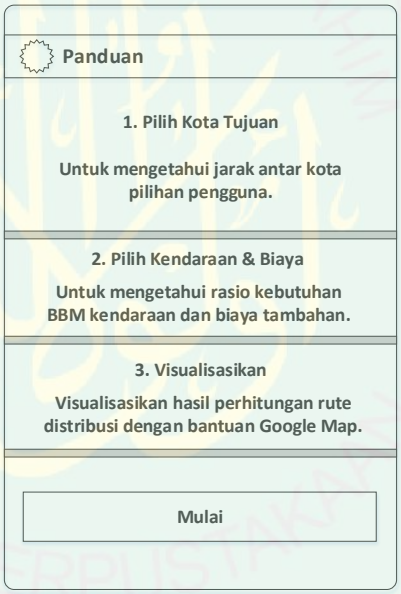
a. Menu Optimasi: untuk menghitung optimasi rute dan biaya operasional distribusi.
b. Menu Berkas: untuk mengetahui data-data tentang Kendaraan distribusi milik perusahaan dan data pelanggan.

c. Menu Peta: untuk menemukan rute/jalur menuju lokasi tertentu dengan bantuan Google Maps.

d. Menu Tentang Kami : untuk mengetahui profil dari aplikasi dan profil perusahaan jasa distribusi logistik PT. ANTESS.

Pada halaman beranda ini juga terdapat tombol KELUAR, untuk keluar dari aplikasi Maksimus.

3. Halaman Hitung Optimasi (Panduan)

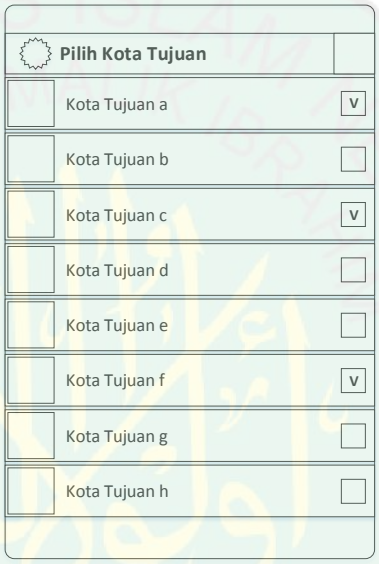


Gambar 3.9. Halaman Hitung Optimasi (Panduan)

Pada halaman panduan ini, pengguna akan disediakan panduan singkat tentang proses yang akan dilakukan oleh pengguna dalam perhitungan optimasi tersebut. Penjelasan ditampilkan dengan *TextView*, dengan setiap poin dibatasi dengan garis (*View*), terdapat tiga poin dalam panduan singkat tersebut: Pilih Kendaraan, Pilih Pelanggan, dan Visualisasikan.

Kemudian pada bagian bawah halaman terdapat tombol Mulai Hitung, untuk masuk ke halaman berikutnya, dan memulai proses memberi masukan kepada sistem untuk melakukan proses perhitungan optimasi rute dan biaya.

4. **Halaman Hitung Optimasi (Pilih Lokasi Distribusi)**



Gambar 3.10. Halaman Hitung Optimasi (Pilih Lokasi Distribusi)

Untuk proses perhitungan yang pertama, diperlukan masukkan berupa lokasi tujuan distribusi logistik. Pada halaman ini nama kota tujuan ditampilkan dengan *ListView* dengan *CheckBox*, jadi akan ada beberapa lokasi distribusi yang dapat dipilih.

Pada bagian kiri terdapat logo yang ditampilkan menggunakan *ImageView*. Pada *ActionBar* terdapat Menu Next untuk menuju ke halaman berikutnya, yaitu Halaman Pilih Kendaraan dan Biaya.

5. Halaman Hitung Optimasi (Pilih Kendaraan & Biaya)

Gambar 3.11. Halaman Hitung Optimasi (Pilih Kendaraan & Biaya)

Halaman ini pengguna dapat memilih kendaraan yang akan digunakan dalam proses distribusi logistik. Ditampilkan dengan model *Spinner ListView*, sehingga hanya akan ada satu kendaraan saja yang dapat dipilih.

Selain memilih kendaraan, pengguna juga akan menentukan biaya timbang dan biaya perbaikan kendaraan. Pada bagian ini pilihan juga ditampilkan dengan model *Spinner ListView*, dan masing-masing hanya dapat memilih satu pilihan saja.

Setelah selesai dengan pilihan kendaraan dan keperluan biaya, pengguna dapat menggunakan menu *Next* pada *ActionBar* untuk lanjut ke halaman berikutnya.

6. Halaman Optimasi (Preview Data)

Gambar 3.12. Halaman Optimasi (*Preview* Data)

Halaman Preview menampilkan lokasi distribusi pilihan pengguna pada Halaman Pilih Pelanggan, dengan model *ListView*. Pada bagian bawahnya terdapat kendaraan distribusi, jumlah biaya, dan jumlah biaya perbaikan pilihan pengguna pada halaman sebelumnya yang ditampilkan dengan *TextView*.

Halaman ini berfungsi untuk menampilkan kembali data-data masukkan yang telah dipilih oleh pengguna pada halaman-halaman sebelumnya, dan apakah telah benar atau belum.

Pada bagian bawah halaman terdapat *Button* untuk menuju ke halaman selanjutnya, dan memulai proses perhitungan optimasi rute dan biaya distribusi.

7. Halaman Optimasi (Hasil Perhitungan Optimasi)

Gambar 3.13. Halaman Optimasi (Hasil Perhitungan Optimasi)

Setelah proses perhitungan selesai, maka akan ditampilkan halaman Hasil Optimasi. Pada halaman tersebut terdapat daftar lokasi distribusi dengan menggunakan *ListView*. Tetapi telah diurutkan berdasarkan rute tempuh terbaik, dengan lokasi pertama sampai lokasi terakhir yang akan dikunjungi.

Estimasi biaya distribusi akan ditampilkan menggunakan *TextView*, di bawah daftar lokasi distribusi. Untuk memvisualisasikan rute distribusi, pengguna dapat menekan setiap item dari daftar tersebut, dan pengguna akan diarahkan ke halaman maps yang akan menampilkan rute tempuh kendaraan.

Ketika telah berada di halaman ini, pengguna tidak dapat kembali ke halaman sebelumnya, jadi pengguna harus menggunakan tombol keluar pada *ActionBar* untuk kembali ke Halaman Beranda.

8. Halaman Visualisasi Rute Distribusi

Gambar 3.14. Halaman Visualisasi Rute Distribusi

Halaman ini akan menampilkan peta dengan menggunakan bantuan Google Map, dimana pada peta tersebut akan terlihat rute tempuh dari lokasi-lokasi distribusi sesuai dengan hasil perhitungan pada bagian sebelumnya. Ketika pengguna memilih salah satu kota dari daftar rute terbaik pada Halaman Hasil Optimasi, maka kota tersebut akan menjadi tujuan dari *driving direction*, dan lokasi awal adalah kota sebelumnya.

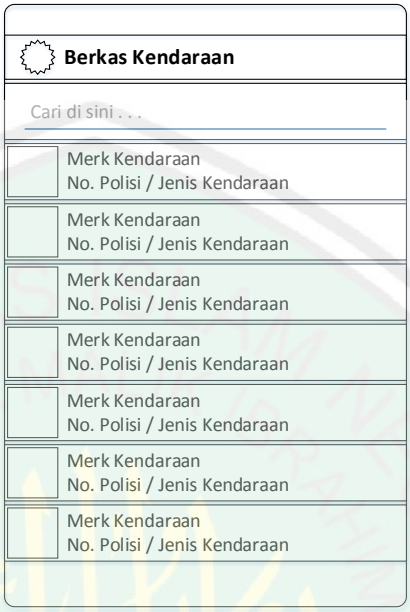
9. Halaman Berkas

Gambar 3.15. Halaman Berkas

Halaman ini akan menampilkan tiga menu dengan *ImageView* untuk logo menu dan *TextView* digunakan untuk menampilkan judul menu dan penjelasan singkat mengenai fungsi menu tersebut.

Terdapat tiga pilihan, yaitu Berkas Kendaraan yang berisi informasi tentang kendaraan transportasi logistik yang dimiliki oleh PT. ANTESS, Berkas Pelanggan yang berisi tentang informasi identitas pelanggan, dan Berkas Daftar harga berisi informasi mengenai harga/ biaya pengiriman berdasarkan kota tujuan dan jumlah barang.

10. Halaman Berkas Kendaraan



Gambar 3.16. Halaman Berkas Kendaraan

Ketika pengguna memilih Menu Berkas Kendaraan, halaman ini akan muncul dengan menampilkan data-data kendaraan distribusi dengan model *custom ListView* dengan *ImageView* berupa logo, dan *TextView* berupa merk kendaraan, nomor polisi, dan jenis kendaraan.

Pada bagian atas *ListView* terdapat menu pencarian dengan menggunakan model *EditText*. Berfungsi untuk mencari data kendaraan dengan mengetikkan merk kendaraan yang dimaksud, sehingga akan memudahkan dan mempercepat pengguna dalam menemukan data yang diperlukan.

Pengguna dapat melihat detail kendaraan dari tiap data di ListView tersebut dengan memilih/tap salah satu dari daftar tersebut, yang akan ditampilkan pada halaman berikutnya pada Detail data.

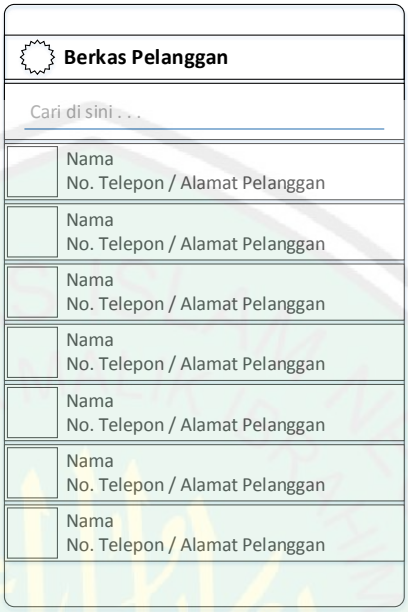
11. Halaman Detail Berkas Kendaraan

Gambar 3.17. Halaman Detail Berkas Kendaraan

Halaman ini akan muncul ketika pengguna memilih salah satu dari daftar pilihan/*ListView* pada Berkas Kendaraan. Logo ditampilkan dengan menggunakan *ImageView* dan data-data yang ada ditampilkan dengan *TextView*.

Pada halaman ini akan ditampilkan data-data yang lebih lengkap dari kendaraan yang dipilih, seperti Merk Kendaraan, Nomor Polisi, Jenis Kendaraan, Tahun, Kapasitas Tangki dan Rasio Penggunaan Bahan Bakar.

12. Halaman Berkas Pelanggan



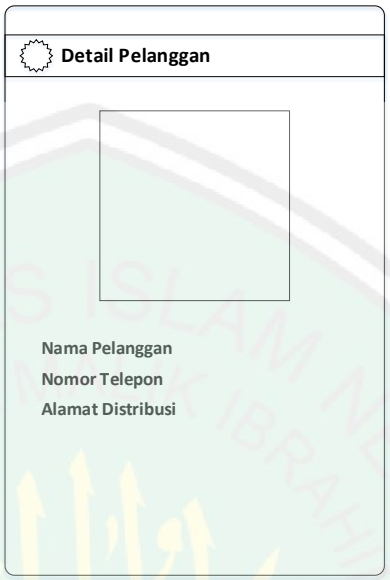
Gambar 3.18. Halaman Berkas Pelanggan

Serupa dengan Berkas Kendaraan, Berkas Pelanggan akan menampilkan data-data pelanggan dengan model logo menggunakan *ImageView* disebelah kiri, dan menggunakan *TextView* untuk menampilkan Nama, No. Telepon, dan Alamat Pelanggan.

Pada bagian atas *ListView* terdapat menu pencarian untuk mencari data pelanggan berdasarkan nama pelanggan, dan akan ditampilkan dengan model urut *ascending*.

Ketika pengguna memilih/tap salah satu dari tiap daftar tersebut, pengguna akan masuk ke halaman berikutnya yang akan menampilkan detail dari data yang dipilih tersebut.

13. Halaman Detail Berkas Pelanggan



Gambar 3.19. Halaman Detail Berkas Pelanggan

Pada halaman ini akan ditampilkan tentang data pelanggan yang lebih lengkap dari halaman sebelumnya, yaitu berkas pelanggan. Logo ditampilkan menggunakan ImageView, dan TextView disini digunakan untuk menampilkan data berupa Nama Pelanggan, Nomor Telepon, Alamat.

14. Halaman Berkas Daftar Harga

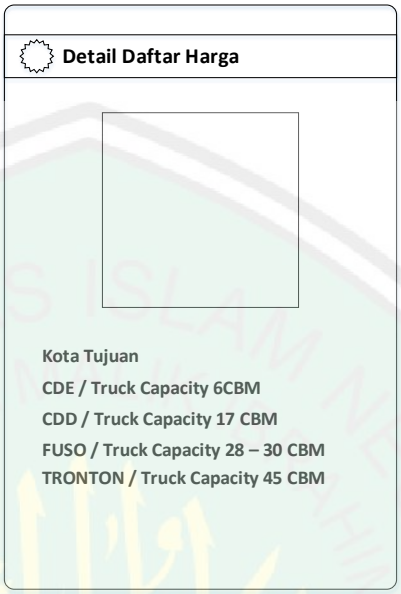
Gambar 3.20. Halaman Berkas Daftar Harga

Pada halaman Berkas Daftar Harga, ditampilkan data-data kota tujuan yang disediakan oleh PT. ANTESS, dengan model *ListView* dengan logo menggunakan *ImageView* disebelah kiri, dan menggunakan *TextView* untuk menampilkan kota tujuan.

Di bagian atas *ListView*, terdapat fasilitas pencarian data berupa kolom *EditText* untuk memudahkan para pengguna dalam mencari kota tujuan berdasarkan input nama kota.

Ketika pengguna memilih/tap salah satu dari tiap daftar tersebut, pengguna akan masuk ke halaman berikutnya yang akan menampilkan detail dari data yang dipilih tersebut.

15. Halaman Detail Daftar Harga



Gambar 3.21. Halaman Detail Daftar Harga

Halaman ini akan muncul ketika pengguna memilih salah satu dari daftar pilihan/*ListView* pada Halaman Daftar Harga. Logo ditampilkan dengan menggunakan *ImageView* dan data-data yang ada ditampilkan dengan *TextView*.

Pada halaman ini akan ditampilkan daftar harga berdasarkan kota tujuan dan kendaraan distribusi yang akan digunakan. Sehingga harga akan ditampilkan per kategori, yaitu CDE / Kapasitas 6 CBM, CDD / Kapasitas 17 CBM, FUSO / Kapasitas 28 – 30 CBM, dan TRONTON / Kapasitas 45 CBM.

16. Halaman Peta

Gambar 3.22. Halaman Peta

Menu Utama yang ketiga adalah Menu Peta, halaman seperti gambar di atas akan muncul ketika pengguna memilih Menu Peta tersebut. Fasilitas API Google Maps digunakan untuk menampilkan fungsi Google Map tersebut, dan layout *fragment* digunakan untuk menempatkan peta tersebut pada halaman/ *layout* utama tampilan tersebut.

Dengan terdapatnya menu peta ini, pengguna penguna dapat mengetahui posisinya dengan menggunakan layanan GPS. Pengguna juga dimudahkan dengan fasilitas *driving direction* yang disediakan pada peta ini. Pengguna dapa menekan posisi dimana saja pada peta sebagai lokasi tujuan, dengan lokasi awal tergantung pada koordinat pengguna saat itu, dengan menggunakan bantuan GPS.

17. Halaman Tentang Kami

Gambar 3.23. Halaman Tentang Kami (Tentang Aplikasi)

Gambar 3.24. Halaman Tentang Kami (Profil PT. ANTESS)

Pada halaman ini akan ditampilkan Informasi singkat tentang aplikasi maksimus, dan informasi tentang profil perusahaan PT. ANTESS. Tampilan pada halaman ini menggunakan *TabLayout*, dimana dalam satu halaman tersebut terdapat dua tab untuk penjelasan aplikasi dan profil perusahaan. Dimana masing-masing logo ditampilkan dengan *ImageView*, dan penjelasan berupa teks menggunakan *TextView*.

3.3 Kebutuhan Sistem

Berikut ini merupakan beberapa hal yang dibutuhkan untuk mendukung pembuatan serta uji coba aplikasi Optimasi Rute dan Biaya Operasional berbasis Android.

1. Perangkat Keras (*Hardware*)
   a. PC / Laptop, dengan spesifikasi minimal : *Processor* Intel(R) Core(TM) 2 Duo T6600 @ 2.20GHz (2 CPUs) dan *Memory* 2 GB RAM, digunakan untuk pembuatan aplikasi.
   b. Hand Phone: Perangkat Mobile yang berbasis Android versi 4.0.3 (*Ice Cream Sandwich*), dibutuhkan untuk melakukan uji coba aplikasi.
2. Perangkat Lunak (*Software*)
   a. Java, digunakan untuk dapat melakukan kompilasi aplikasi Android. Versi yang digunakan Sun Java SE versi 1.7 atau versi di atasnya.

b. Software Eclipse / *Android Development Tool*. Merupakan *software* yang dibutuhkan untuk melakukan coding aplikasi Android.

c. Android SDK (*Software Development Kit*), diperlukan sebagai alat bantu dan API dalam mengembangkan aplikasi Android menggunakan bahasa Java.

BAB IV

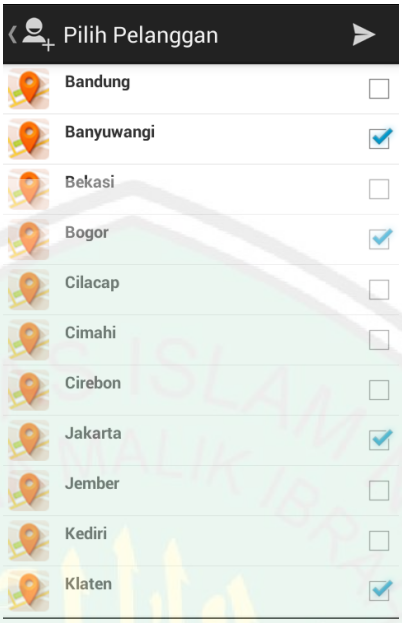
HASIL DAN PEMBAHASAN

4.1 Implementasi Algoritma *Tabu Search*

Setelah pengguna selesai memasukkan data, kemudian menekan tombol Mulai Hitung yang terdapat pada Halaman Preview Data, secara otomatis akan menjalankan perhitungan menentukan jarak tempuh kendaraan dan rute terpendek menggunakan algoritma tabu search.

Gambar 4.1. Data Pilihan Pengguna

Data yang diperlukan dalam melakukan perhitungan dengan tabu *search* adalah jarak antar kota pilihan pengguna. Ketika pengguna memilih kota tujuan di Halaman Pilih Pelanggan dengan *checkbox* maka secara otomatis dalam database, kota yang telah terpilih/ dicentang akan memiliki nilai 1, sedangkan kota yang tidak dipilih akan bernilai 0 (nol).



Gambar 4.2. Memilih Kota Tujuan Distribusi.

Kemudian pada *class* TabuSearch.java kota-kota pilihan pengguna tersebut akan diambil yaitu Banyuwangi, Bogor, Jakarta, dan Klaten, kemudian data jarak antar lokasi tersebut yang akan diolah dan dihitung panjang rute terpendek dan urutan lokasi untuk dikunjungi. Kode untuk mengambil data tersebut adalah sebagai berikut:

```
String[] dapat2 = new String[] { "COUNT (*) as jumlah" };

cursor123 = db.query(DatabasePelangganHelper.PILIH_PELANGGAN,
dapat2,"selected = 1", null, null, null, null, null);

if (cursor123.moveToNext()) {
      ae = Integer.parseInt(cursor123. getString(
      cursor123.getColumnIndex("jumlah")));
}
```

Kode tersebut digunakan untuk mendapatkan jumlah kota pilihan pengguna dari database. Termasuk *warehouse*, karena *warehouse* akan secara

otomatis terpilih dan bernilai 1 di dalam database. Sehingga jumlah kotanya adalah 5. Kemudian nilai yang didapat dari database tersebut dikonversi ke dalam tipe *Integer*, karena ketika diambil dari database data tersebut masih bertipe *String*. Jumah kota tersebut digunakan untuk membentuk matrik. Dalam kelas TabuSearch matrik digunakan untuk membandingkan jarak antar kota tujuan.

Setelah mendapatkan jumlah kota tujuan distribusi ditambah dengan *warehouse*, selanjutnya mengambil *string* nama dari setiap kota tujuan tersebut dengan menggunakan kode sebagai berikut:

```
String[] dapat = new String[] { DatabasePelangganHelper.KEY_NAME
};
cursor = db.query(DatabasePelangganHelper.PILIH_PELANGGAN, dapat,
            "selected = 1", null, null, null, null, null);

kotanya = new String[ae];
Integer isikota = 0;

while (cursor.moveToNext()) {
kotanya[isikota] = cursor.getString(
                   cursor.getColumnIndex ("nama"));
isikota++;
}
cursor.close();
```

Kode tersebut akan mengambil nama kota dari database dengan hanya kota yang bernilai 1 atau hanya yang dipilih oleh pengguna. Ketika data yang dimaksud ada di dalam database, data akan terus diambil sejumlah dengan banyaknya kota yang bernilai 1. Nama kota yang telah diambil dari database tersebut kemudian disimpan dalam bentuk array di `kotanya = new`

`String[ae];` dengan nilai `ae` sejumlah dengan banyaknya nilai yang diambil dengan kode sebelumnya dari tabel database yang sama.

```
tspEnvironment.jarak = new int[ae][ae];

for (int k = 0; k < tspEnvironment.jarak.length; k++) {
      for (int z = 0; z < tspEnvironment.jarak[0].length; z++) {

String[] s = new String[] { DatabasePelangganHelper.KEY_JARAK };
cursor2 = db.query(DatabasePelangganHelper.DESTINASI, s,
            "kota1 = '" + kotanya[k] + "' and kota2 = '"
            + kotanya[z] + "'", null, null, null, null);

      cursor2.moveToFirst();
      tspEnvironment.jarak[k][z] = Integer.parseInt(cursor2
            .getString(cursor2.getColumnIndex("jarak")));
      cursor2.close();
      }
}
```

Kode di atas akan membuat matrik jarak [k][z] sesuai dengan jumlah kota tujuan dan *warehouse*, jadi matrik tersebut akan memiliki dimensi 5x5, karena pengguna telah memiliki 5 titik lokasi, yaitu Warehouse, Banyuwangi, Bogor, Jakarta, dan Klaten.

Selanjutnya matrik jarak tersebut diisi nilainya dengan jarak dari kombinasi setiap lokasi distribusi, dengan mengambil nilainya dari database berdasarkan nama lokasi. Perulangan akan berhenti ketika semu jarak dari setiap kota telah didapat. Berikut merupakan jaraknya: 0, 273, 846, 814, 290, 273, 0, 1114, 1082, 539, 846, 1114, 0, 56, 611, 814, 1082, 56, 0, 580, 290, 539, 611, 580, 0

Sebagai gambaran matrik jarak yang terbentuk akan memiliki pola seperti berikut ini:

```
tspEnvironment.jarak = new int[][]{{0   , 273 , 846 , 814 , 290},
                                   {273 , 0   , 1114, 1082, 539},
                                   {846 , 1114, 0   , 56  , 611},
                                   {814 , 1082, 56  , 0   , 580},
                                   {290 , 539 , 611 , 580 , 0}};
```

Nilai nol tersebut mengindikasikan nilai tersebut merupakan nilai jarak dari lokasi yang sama. Dalam matrik tersebut elemen `[0][273]` merepresentasikan jarak antara *warehouse* dan Banyuwangi, dimana jarak antara *warehouse* ke Banyuwangi sama dengan jarak Banyuwangi ke *warehouse*.

Setelah matrik selesai dibentuk dan diisi dengan jarak, maka selanjutnya adalah inisialisasi solusi sementara rute distribusi. Kodenya adalah sebagai berikut:

```
int[] SolusiSementara = new int[ae + 1];

for (int p = 0; p < ae; p++) {
      SolusiSementara[p] = p;
}
SolusiSementara[ae] = 0;
Log.i("jumlah kota", "" + kotanya.length);

int faktorial = 1;
for (int ifak = 1; ifak <= kotanya.length - 1; ifak++) {
      faktorial *= ifak;
}

Log.i("Jumlah kombinasi masing2 kota", "" + faktorial);

int jumlahIterasi = faktorial;
int jumTabu = kotanya.length;

TabuList tabuList = new TabuList(jumTabu);

int[] solusiTer = new int[SolusiSementara.length];
System.arraycopy(SolusiSementara, 0, solusiTer, 0,
       solusiTer.length);
int bestCost = tspEnvironment.getObjectiveFunctionValue
               (solusiTer);
```

Dalam inisialisasi solusi sementara tersebut, nomor kota dimulai dari 0, dengan ketentuan lokasi awal dan lokasi terakhir posisinya tidak akan ditukar dengan lokasi lain. Karena lokasi awal dan terakhir adalah *warehouse* dan diinisialisasikan dengan 0. Solusi sementara dapat digambarkan sebagai berikut: `int[] SolusiSementara = new int[]{0, 1, 2, 3, 4, 0};`. Selanjutnya menentukan jumlah iterasi dan jumlah tabu list. Jumlah iterasi ditentukan dengan nilai faktorial dari jumlah kota pilihan pelanggan dan *warehouse*, dan menentukan tabu list sebagai titik mulai dan solusi awal proses perhitungan.

Solusi awal sementara, dan kemudian akan disimpan dalam tabu list :

- Rute = Warehouse-Banyuwangi-Bogor-Jakarta-Klaten-Warehouse
- Jarak = 237 + 1114 + 56 + 580 + 290 = 2313

```
int[][] tabuList;

public TabuList(int nomKota) {
	tabuList = new int[nomKota][nomKota];
}

public void tabuMove(int kota1, int kota2, int kota) {
	tabuList[kota1][kota2] += kota;
	tabuList[kota2][kota1] += kota;
}
```

Tabu list tersebut juga direpresentasikan sebagai matrik jarak yang telah dibentuk, dimana setiap nilai merepresentasikan kedudukan/ posisi tabu dalam proses tukar-menukar posisi lokasi distribusi. Misalnya, kota 1/a dan kota 2/b, ketika berjalan proses akan membuat tabu terhadap matrik tersebut

antara [1][2] dan [2][1]. Hal ini akan mencegah terjadinya langkah yang berulang dua kali atau proses tukar-menukar lokasi dengan kota dan nilai yang sama, jadi inilah yang dinamakan dengan tabu list atau daftar tabu.

```
for (int i = 0; i < jumlahIterasi; i++) {
SolusiSementara = TabuSearch.getBestNeighbour(tabuList,
                    tspEnvironment, SolusiSementara);

int NilaiSekarang = tspEnvironment.getObjectiveFunctionValue
                   (SolusiSementara);

Log.i("Hasil perhitungan saat ini = ", "" + tspEnvironment.
             getObjectiveFunctionValue(SolusiSementara));

if (NilaiSekarang < nilaiTer) {
System.arraycopy(SolusiSementara, 0, solusiTer, 0,
solusiTer.length);
nilaiTer = NilaiSekarang;
      }
}
```

Proses akan mencoba menemukan solusi terbaik pada setiap iterasi yang telah ditentukan tersebut tanpa melakukan proses atau perpindahan yang bersifat tabu. Solusi yang didapatkan dari setiap iterasi akan selalu diterima, setiap iterasi dimulai dari solusi pada iterasi sebelumnya. Ketika perhitungan pada setiap iterasi berjalan, akan selalu dicatat solusi yang dihasilkan, dan menyimpan solusi terbaik yang didapatkan. Ketika proses telah dijalankan sejumlah dengan iterasi yang telah didefinisikam tersebut maka proses akan berhenti, dan nilai jarak tempuh terkecil yang telah didapat dari proses sejumlah iterasi tersebut merupakan solusi terbaik yang ditemukan.

Berikut merupakan solusi terbaik yang ditemukan: Solusi terbaik = 2293, Urutan kota terbaik : 0, 3, 2, 4, 1, 0 (Warehouse – Jakarta – Bogor – Klaten – Banyuwangi – Warehouse).

4.2 Implementasi Aplikasi

Gambar 4.3. Halaman *Splash Screen*

Halaman pertama yang muncul ketika aplikasi berjalan adalah *splash screen*, halaman ini akan menampilkan nama aplikasi, logo aplikasi, dan nama perusahaan. Setelah ditampilkan 3 detik, halaman *splash screen* akan berganti ke halaman selanjutnya, yaitu halaman Beranda.

Halaman splash screen tersebut hanya berfungsi sebagai pengantar atau pembuka pada aplikasi Maksimus ini, sebelum menuju ke halaman utama dari aplikasi.

Gambar 4.4. Halaman Beranda

Pada Halaman Beranda ini terdapat empat menu, yaitu:

a. Menu Optimasi: menu ini merupakan inti dari aplikasi, berfungsi untuk menghitung rute terbaik sehingga akan diperoleh estimasi biaya operasional distribusi logistik.
b. Menu Berkas: untuk mengetahui data-data tentang Kendaraan distribusi milik perusahaan. Pada menu ini jua terdapat informasi mengenai lokasi distribusi / data-data tentang pelanggan serta daftar harga untuk distribusi logistik berdasarkan besaran barang dan kota tujuan pengiriman.
c. Menu Peta: untuk menemukan suatu lokasi maupun melihat rute dengan bantuan Google Maps.
d. Menu Tentang Kami: berisi informasi tentang profil dari aplikasi dan profil singkat dari PT. ANTESS.

Untuk keluar dari aplikasi Maksimus, pengguna dapat menggunakan tombol keluar yang ada pada pojok kanan atas halaman, atau pengguna juga bisa menggunakan tombol kembali di *smartphone* sebanyak dua kali.

Gambar 4.5. Halaman Hitung Optimasi (Panduan)

Sebelum pengguna memberikan masukan untuk proses perhitungan optimasi, pengguna akan disediakan panduan singkat tentang proses yang akan dilakukan dalam perhitungan optimasi tersebut. Secara umum terdapat tiga langkah utama dalam proses ini, yaitu pengguna memilih kota tujuan distribusi logistik, selanjutnya pengguna memilih kendaraan yang digunakan, biaya timbang dan biaya perbaikan, terakhir pengguna memvisualisasikan rute distribusi menggunakan bantuan *Get Direction* dengan Google Maps.

Halaman ini akan muncul setiap kali pengguna akan melakukan proses perhitungan optimasi rute dan biaya distribusi. Pada bagian bawah halaman terdapat tombol Mulai untuk masuk ke halaman berikutnya, dan

memulai proses memberikan masukan kepada sistem untuk selanjutya sistem akan melakukan perhitungan optimasi rute dan biaya.

Gambar 4.6. Halaman Hitung Optimasi (Pilih Kota Tujuan)

Untuk proses perhitungan pertama, diperlukan masukan berupa kota tujuan / lokasi distribusi. Jadi dari daftar yang telah ditampilkan tersebut, pengguna dapat memilih beberapa lokasi distribusi, jumlah kota tujuan dibatasi tidak lebih dari empat lokasi. Dalam prosedur di PT. Antaran Express untuk proses pengiriman barang dalam bentuk eceran (lebih dari satu lokasi distribusi) hanya empat lokasi distribusi.

Selain itu juga untuk lebih meringankan kinerja *smartphone* dalam melakukan perhitungan dalam menentukan rute terbaik. Karena setiap kota (termasuk *warehouse*) yang dipilih akan dibuat menjadi array 2D dengan bentuk matrik sesuai banyak kota pilihan pengguna.

Setelah selesai memilih lokasi distribusi logistik, pengguna dapat menuju ke halaman berikutnya dengan menggunakan tombol *Next* yang terdapat pada *ActionBar*.

Gambar 4.7. Halaman Hitung Optimasi (Pilih Kendaraan & Biaya)

Pada halaman ini pengguna dapat memilih satu kendaraan yang terdapat dalam daftar, yang mana kendaraan tersebut akan digunakan dalam proses distribusi logistik. Pemilihan kendaraan ini bertujuan untuk mengetahui rasio kebutuhan bahan bakar kendaraan tersebut, karena kebutuhan bahan bakar kendaraan bisa saja berbeda berdasarkan jenis dan merknya.

Selain memilih kendaraan, pengguna juga dapat menentukan biaya untuk keperluan kendaraan tersebut, yaitu biaya untuk timbang kendaraan dan biaya untuk perbaikan kendaraan tersebut. Setelah selesai memilih kendaraan, pengguna dapat menggunakan menu *Next* pada *ActionBar* untuk menuju ke halaman berikutnya, Halaman Preview.

Gambar 4.8. Halaman *Preview* Data

Halaman *Preview* Data menampilkan data-data/ *input* yang telah diberikan pengguna. Mulai dari lokasi kota tujuan distribusi logistik, pada daftar akan otomatis ditambah dengan *Warehosuse* yang merupakan lokasi awal pemberangkatan kendaraan distribusi. Kendaraan yang pilihan pengguna dan besaran biaya timbang dan biaya perbaikan untuk keperluan kendaraan tersebut.

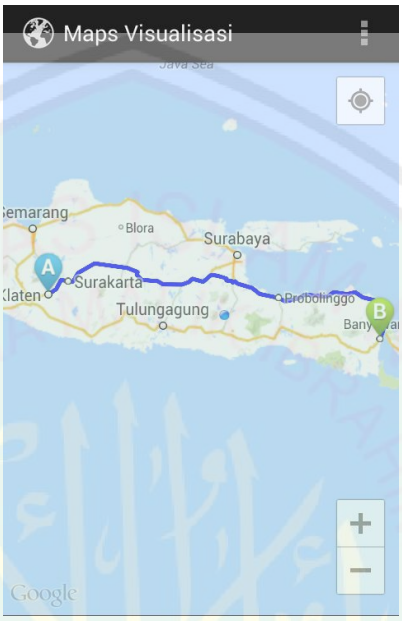
Pengguna dapat kembali ke halaman sebelumnya apabila data yang dimasukkan kurang sesuai atau belum benar. Setelah pemilihan data benar dan *preview* data sesuai, pengguna dapat memulai proses perhitungan rute terbaik dengan menekan tombol Mulai Hitung.

Gambar 4.9. Halaman Optimasi (Hasil Perhitungan Optimasi)

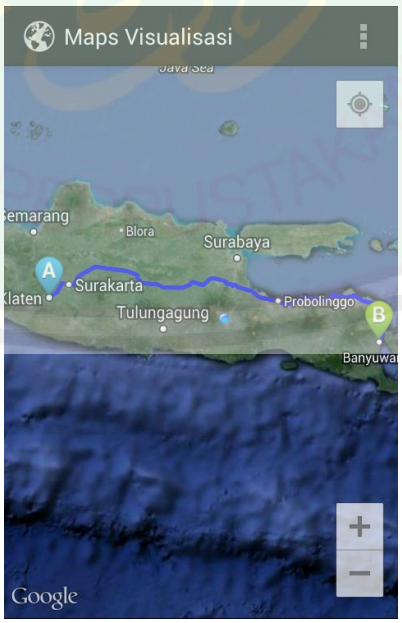
Setelah proses perhitungan selesai, maka hasil dari peritungan tersebut akan ditampilkan di halaman Hasil Optimasi. Jarak tempuh kendaraan terbik dari hasil perhitungan menggunakan tabu *search* ditampilkan di bagian atas halaman, jarak tempuh inilah yang digunakan untuk menghitung besar biaya untuk keperluan bahan bakar kendaraan yang digunakan. Selain itu, pada halaman ini terdapat daftar lokasi distribusi, sesuai dangan masukan dari pengguna tetapi telah diurutkan berdasarkan rute tempuh terpendek, dengan lokasi pertama (*warehouse*) sampai lokasi terakhir yang akan dikunjungi.

Estimasi biaya distribusi akan ditampilkan di bawah daftar lokasi distribusi. Biaya tersebut merupakan estimasi biaya untuk keseluruhan proses distribusi logistik untuk jenis kendaraan tersebut dan lokasi kota tujuan pilihan pelanggan tersebut. Untuk memvisualisasikan rute distribusi, pengguna dapat menekan setiap item dari daftar kota tujuan yang ada pada

listview tersebut, dan tombol Selesai (tanda silang) untuk keluar dan kembali ke menu beranda.



Gambar 4.10. Halaman MapVisualisasi (*Normal Mode*)



Gambar 4.11. Halaman MapVisualisasi (*Hybrid Mode*)

Halaman ini akan menampilkan peta dengan menggunakan bantuan Google Map, dimana pada peta tersebut akan menampilkan rute tempuh kendaraan berdasarkan lokasi-lokasi distribusi sesuai dengan pilihan pengguna pada bagian sebelumnya. Rute dimulai dari point A dan berakhir di point B, dimana kota yang pengguna pilih akan menjadi kota tujuan.

Untuk mendapatkan rute tersebut, masing-masing lokasi termasuk *warehouse* telah diketahui koordinatnya. Dengan menggunakan bantuan fasilitas *Driving Direction* dari Google Maps menampilkan rute tempuh kendaraan sesuai dengan yang diinginkan pengguna. Pengguna juga dapat memilih mode peta, dengan menu yang disediakan pada ActionBar yaitu *Normal Mode* dan *Hybrid Mode*.

Gambar 4.12. Halaman Berkas

Pada halaman ini terdapat tiga pilihan yaitu, Berkas Kendaraan yang berisi daftar kendaraan yang dimiliki oleh perusahaan, Berkas Pelanggan

yang berisi tentang informasi identitas pelanggan, dan Berkas Daftar Harga yang berisi informasi tentang harga/ biaya distribusi logistik berdasarkan tujuan lokasi distribusi dan besaran volume barang tersebut, sehingga bisa ditentukan akan dapat dimuat dengan menggunakan jenis kendaraan tertentu.

Gambar 4.13. Halaman Berkas Kendaraan

Ketika pengguna memilih Menu Berkas Kendaraan, halaman ini akan muncul dengan menampilkan daftar kendaraan distribusi. Pada daftar tersebut menampilkan merk kendaraan dengan nomor polisinya. Pengguna dapat mencari data kendaraan yang dibutuhkan dengan memasukkan nama merk kendaraan pada kolom pencarian di bagian atas halaman. Sistem secara otomatis akan mencari kata kunci yang dimasukkan pengguna tersebut dan selanjutnya akan ditampilkan tetap menggunakan model *listview* berdasarkan kecocokan dengan kata kunci.

Pengguna dapat melihat detail dari setiap data kendaraan di daftar tersebut dengan memilih (*tap*) salah satu dari daftar tersebut, kemudian data-data yang lebih lengkap akan ditampilkan pada halaman berikutnya pada halaman detail data kendaraan.

Gambar 4.14. Halaman Detail Berkas Kendaraan

Halaman ini akan muncul ketika pengguna memilih salah satu dari daftar kendaraan pada halaman Berkas Kendaraan. Pada halaman ini akan ditampilkan data-data yang lebih lengkap, seperti merk kendaraan, nomor polisi, tahun, jenis kendaraan, serta rasio penggunaan bahan bakar kendaraan tersebut.

Gambar 4.15. Halaman Berkas Pelanggan

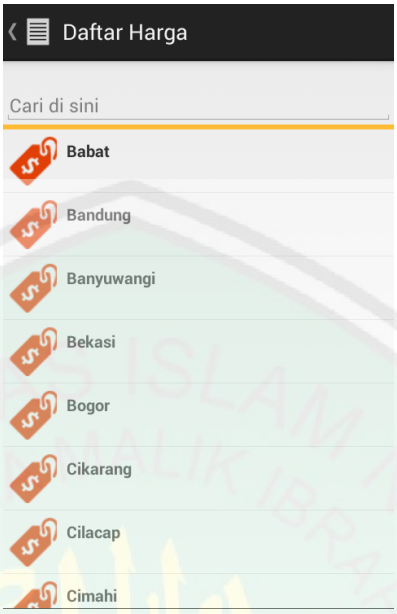
Halaman Berkas Kendaraan ini akan menampilkan daftar pelanggan. Pada bagian atas halaman terdapat fasilitas pencarian, pengguna dapat mencari lokasi distribusi logistik berdasarkan input berupa nama lokasi. Kemudian sistem secara otomatis akan mencari dari database seperti yang diinputkan oleh pengguna. Ketika ditemukan kecocokan data, akan ditampilkan dalam daftar data-data tersebut.

Ketika pengguna memilih/tap salah satu dari tiap daftar tersebut, pengguna akan masuk ke halaman berikutnya yang akan menampilkan detail dari data yang dipilih tersebut.

Gambar 4.16. Halaman Detail Berkas Pelanggan

Pada halaman ini akan ditampilkan tentang data pelanggan yang lebih lengkap dari halaman sebelumnya yaitu meliputi nama pelanggan, nomor telepon, alamat email, alamat distribusi, lokasi (latitude, longitude).

Pengguna akan dimudahkan dalam menemukan lokasi dari pelanggan tersebut, dengan menekan tombol Temukan Lokasi. Selanjutnya pengguna akan diarahkan ke halaman berikutnya yang berupa Maps dengan menampilkan *point* atau titik lokasi dimana alamat pelanggan tersebut berada.



Gambar 4.17. Halaman Daftar Harga

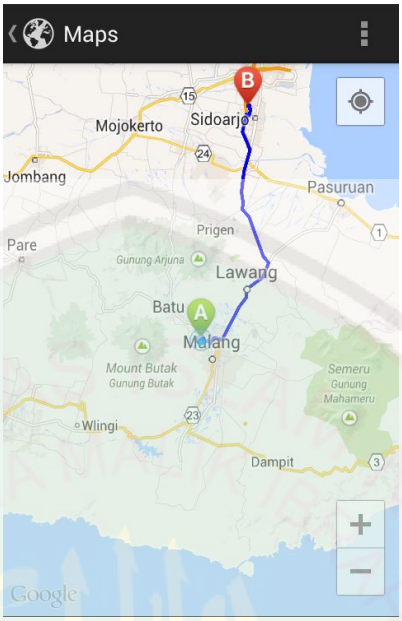
Pada halaman Daftar harga ini, pengguna akan mendapatkan informasi tentang daftar harga pengiriman barang berdasarkan kota tujuan dan banyaknya barang. Pengguna dapat melakukan proses pencarian kota tujuan dengan memanfaatkan kolom cari yang telah disediakan pada bagian atas halaman. Sistem secara otomatis akan memfilter data yang ada, dan data-data yang sesuai dengan masukkan pengguna akan ditampilkan di daftar tersebut.

Ketika pengguna memilih/tap salah satu item dari daftar tersebut, pengguna akan diarahkan pada halaman selanjutnya yaitu halaman Detail daftar harga, dan akan menampilkan detail harga sesuai dengan pilihan pengguna tersebut.

Gambar 4.18. Halaman Detail Daftar Harga

Pada halaman detail daftar harga ini, pengguna akan diberikan informasi yang lebih lengkap dari harga/ biaya pendistribusian logistik. Halaman tersebut akan menampilkan harga berdasarkan kota tujuan dan volume barang yang akan didistribusikan.

Terdapat empat kategori harga yang didefinisikan oleh PT. ANTESS, yaitu CDE dengan kapasitas truk 6 CBM, CDD dengan kapasitas truk 17 CBM, FUSO dengan kapasitas truk 28 – 30 CBM, dan TRONTON/ kapasitas truk 45 CBM.



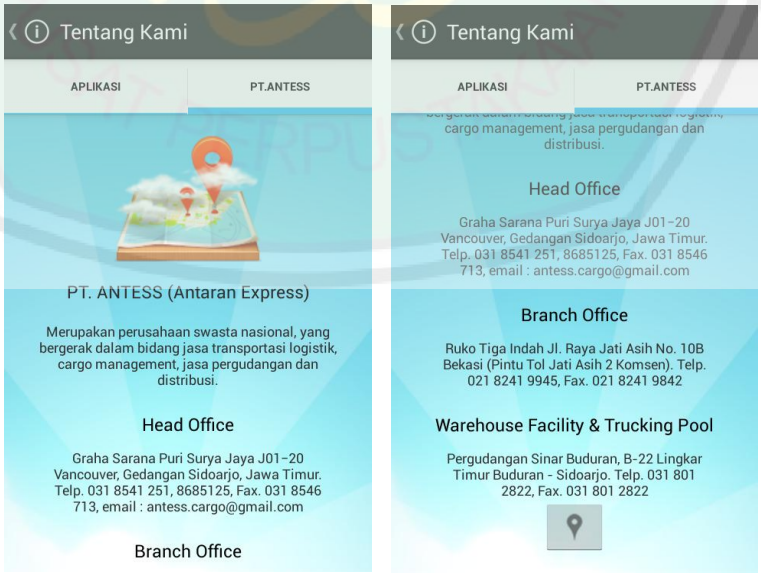
Gambar 4.19. Halaman Maps

Menu Utama yang ketiga adalah Menu Peta, halaman seperti gambar di atas akan muncul ketika pengguna memilih Menu Peta tersebut. Dengan terdapatnya menu peta ini, pengguna dapat mencari rute tempuh terhadap lokasi yang diinginkan, dengan menggunakan fasilitas *driving direction* yang telah disediakan. Pengguna hanya perlu memilih/ menyentuh lokasi tujuan pada peta untuk driving direction, sedangkan untuk lokasi awal akan diambil dari lokasi terkini dari pengguna.

Menu mode tampilan, terdapat dua mode tampilan untuk menampilkan maps, yaitu *normal mode* dan *hybrid mode*, menu ini terdapat pada bagian atas halaman. Juga terdapat menu pendukung *zoom in* dan *zoom out* serta kompas untuk menentukan arah pada peta. Pengguna juga dapat melihat posisinya terkini dalam maps, dengan menggunakan fasilitas GPS. Untuk mendapatkan fasilitas ini, pengguna terlebih dahulu harus

memperbolehkan peta melihat lokasi pengguna terkini dengan melakukan pengaturan pada *smartphone*.

Gambar 4.20. Halaman Tentang Kami (Tentang Aplikasi)



Gambar 4.21. Halaman Tentang Kami (Profil Perusahaan)

Pada halaman Tentang Kami ini terdapat dua tab dimana pada tab pertama menampilkan informasi tentang profil aplikasi optimasi maksimus ini, dan pada tab kedua berisi tentang profil singkat dari PT. ANTESS (Antaran Exspress), serta terdapat informasi tentang alamat, nomor telepon, email dan fax dari kantor dan *warehouse* PT. ANTESS.

Pada tab profil perusahaan, dibagian bawah halaman terdapat tombol temukan lokasi, ini berfungsi untuk mengarahkan pengguna pada google maps dan menampilkan lokasi dari *warehouse* dari PT. Antaran Express.

4.3 Uji Coba Aplikasi

4.3.1 Uji Coba Algoritma Tabu *Search* dalam Menentukan Prioritas Rute dan Jarak Tempuh Terpendek

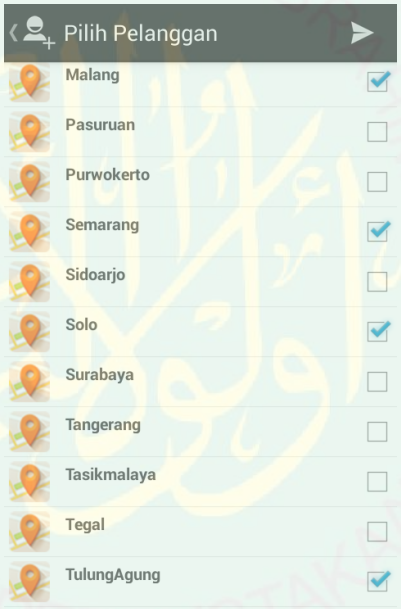
Gambar 4.22. Aplikasi Optimasi Maksimus

Pengujian aplikasi dilakukan menggunakan perangkat keras Sony Xperia L C2105 dengan spesifikasi:

CPU	: Qualcomm MSM8230 Dual-core 1 GHz
OS	: Android 4.2.2 (JellyBean)
Resolusi	: 480 x 854 Pixel (~228 ppi)
Memory Internal	: 8 Gb (5.8 GB *user available*), 1 Gb RAM
Memory Eksternal	: 8 Gb

Pengujian dilakukan terhadap rute dan jarak tempuh kendaraan distribusi logistik yang didapat. Pengujian dilakukan dua kali dengan kota tujuan sebanyak 3 dan 4 kota, dengan masing-masing kota yang berbeda pada setiap pengujian.

Pada pengujian ini ditentukan empat kota tujuan sebagai destinasi distribusi logistik, yaitu Malang, Semarang, Solo, dan Tulung Agung. Berikut ini kota yang dipilih:

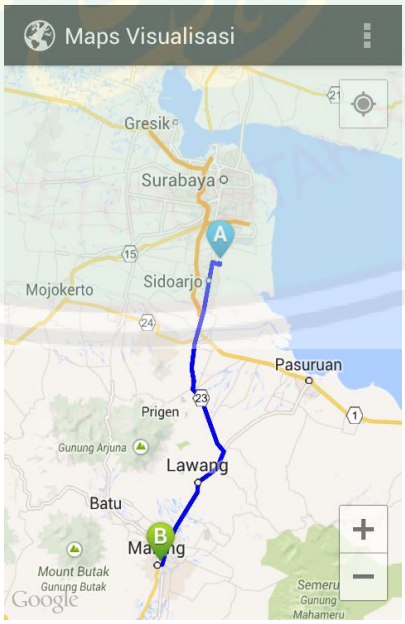


Gambar 4.23. Pengujian (Memilih Kota Tujuan)

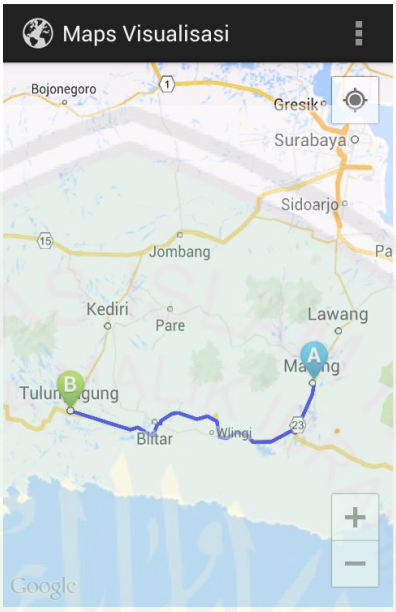
Setelah selesai menentukan kota tujuan dan biaya operasional, kemudian memulai proses perhitungan maka dihasilkan rute tempuh terbaik dan daftar kota tujuan sesuai dengan yang akan dikunjungi terlebih dahulu. Jarak tempuh kendaraan distribusi didapatkan sejauh 808 km, dengan urutan distribusi logistik dimulai dari kota Malang, Tulung Agung, Solo, dan kota terakhir Semarang, seperti berikut ini:

Gambar 4.24. Pengujian (Hasil Optimasi)

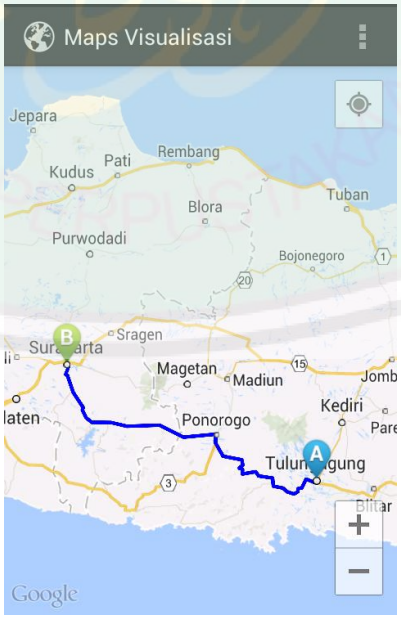
Sedangkan di bawah ini merupakan driving direction, masing-masing dari Warehouse ke kota Malang, dari kota Malang ke kota Tulung Agung, dari kota Tulung Agung ke kota Solo, dan dari kota Solo ke kota Semarang.



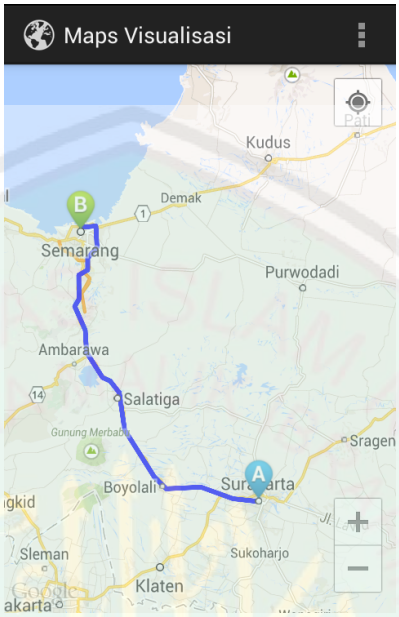
Gambar 4.25. Pengujian (*Routing Warehouse* ke Kota 1)



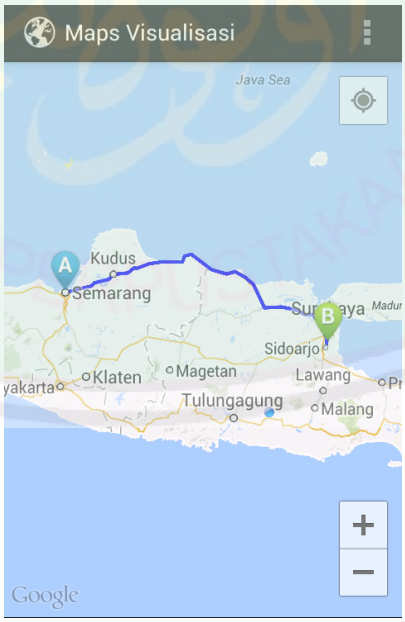
Gambar 4.26. Pengujian (*Routing* Kota 1 ke Kota 2)



Gambar 4.27. Pengujian (*Routing* Kota 2 ke Kota 3)



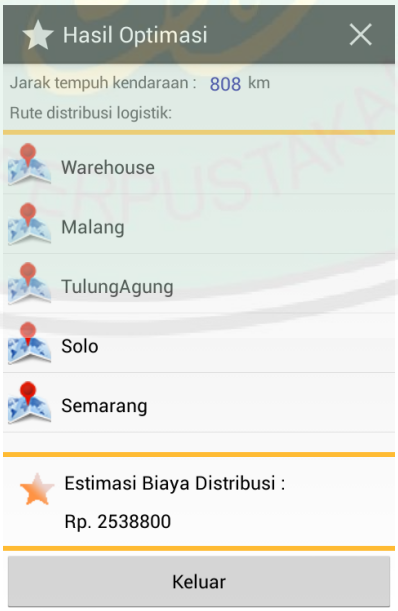
Gambar 4.28. Pengujian (*Routing* Kota 3 ke Kota 4)



Gambar 4.29. Pengujian (*Routing* Kota 4 ke *Warehouse*)

Berikut ini merupakan perhitungan estimasi biaya distribusi dengan biaya tambahan:

Gambar 4.30. Biaya Tambahan



Gambar 4.31. Hasil Optimasi Rute dan Biaya

Jarak tempuh distribusi logistik telah diketahui 808 km, dengan rasio kebutuhan bahan bakar kendaraan 5, dan harga bahan bakar jenis solar 5500, maka biaya untuk keperluan BBM sebagai berikut:

- $biayaBBM = \frac{808}{5}\ x\ 5500$ = 888800
- Total Biaya = $biayaBBM + Biaya\ Timbang + Biaya\ Perbaikan$

  = 888800 + 150000 + 1500000

  = Rp. 2.538.800

Sehingga didapatkan estimasi biaya distribusi sebesar Rp. 2.538.800, dengan jarak tempuh 808 km, rute tempuh *Warehouse* – Malang – TulungAgung – Solo – Semarang – *Warehouse*.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil implementasi dan ujicoba yang sudah dilakukan dapat disimpulkan:

1. Aplikasi dapat berjalan dengan baik karena aplikasi tersebut dapat berjalan di berbagai versi *platform* Android di atas versi 4.0.3 (*Ice Cream Sandwich*).
2. Algoritma Tabu Search yang digunakan dalam perhitungan dan penentuan rute terpendek untuk kendaraan distribusi logistik dapat diaplikasikan.
3. Dibutuhkan keneksi internet yang cepat dan stabil untuk mendapatkan fasilitas *driving direction* pada Google Map dengan baik.
4. Aplikasi dapat digunakan oleh pihak dari PT. Antaran Express (ANTESS), sebagai bahan pertimbangan dalam menentukan biaya operasional distribusi logistik melalui jalur darat, khususnya kota-kota di pulau jawa.

5.2. Saran

Masih banyak kekurangan dalam penelitian aplikasi optimasi rute dan biaya operasional ini. Oleh karena itu penulis menyarankan beberapa hal sebagai bahan pengembangan selanjutnya, diantaranya:

1. Mengembangkan aplikasi Maksimus optimasi rute dan biaya operasional dengan lebih menarik dari segi tampilan dan informasi yang diberikan, terutama fasilitas dan jasa yang diberikan oleh PT. ANTESS, sehingga pengguna dari aplikasi Maksimus ini tidak hanya dari PT. ANTESS saja tetapi juga dari pelanggan.
2. Mengembangkan aplikasi Maksimus untuk sistem operasi perangkat *mobile* yang lain seperti Windows Phone, BlackBerry, iOS dan lainnya.

DAFTAR PUSTAKA

Al- Buthoni, Abdullah bin Taslim. Mengatur dan Membelanjakan Harta. (http://www.muslim.or.id/ Diakses tanggal 14 April 2014)

Al Ghazali, Imam. 1994. *Terjemah Ihya' Ulumiddin Jilid IV*. Semarang: CV. Asy Syifa'

Al-Sheikh, Abdullah Bin Muhammad. 2003. *Tafsir Ibnu Katsir Jilid 3*. Bogor: Pustaka Imam Asy-Syafi'i

Berlianty Intan. dan Miftahol Arifin. 2010. *Teknik-teknik Optimasi Heuristik*. Yogyakarta: Graha Ilmu.

Juniarto, Susilo Dwi. Entin Martiana K. dkk. 2011. *Optimasi Distribusi Barang Berdasarkan Rute dan Daya Tampung Menggunakan Metode Simulated Annealing*. Jurusan Teknik Informatika, PENS – ITS Surabaya.

Kuncoro, Ira, dan Renga. *Penentuan Rute Pendistribusian Surat Kabar dengan Time Window, Aplikasi Algoritma Tabu Search (Studi Kasus : Koran Kompas)*. Jurusan Teknik Informatika, PENS – ITS Surabaya

Kusumadewi, Sri. dan Hari Purnomo. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu.

Nasution, M. Nur. 2004. *Manajemen Transportasi*. Jakarta : PT. Ghalia Indonesia.

Priyandari. *Tabu Search –Introduction*. Teknik Industri Universitas Sebelas Maret. (http://priyandari.staff.uns.ac.id/ Diakses tanggal 11 April 2013 18.40)

Safaat, Nazruddin. 2012. *Pemrograman Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika.

Sherly. 2010. *Usulan Penentuan Rute Pengiriman barang di PT Aries Centaurus Pada Kasus Vehicle Routing Problem dengan Algoritma Tabu Search*. Universitas Kristen Petra.

Sutapa, I Nyoman, I Gede Agus Widyadana, dan Christine. 2003. *Studi Tentang Travelling Salesman dan Vehicle Routing Problem dengan Time Windows.* Jurnal Teknik Industri, Universitas Kristen Petra.

Suyanto. 2010. *Algoritma Optimasi Determenistik atau Probabilitik.* Yogyakarta : Graha Ilmu.