

Analisa dan Penerapan Algoritma Floyd Warshal Untuk Optimalisasi Jalur Berbasis GPS

Rizki Darmawan¹, Rajib Ghaniy²

STIKOM Binaniaga/Sistem Informasi

¹Email: rizkidarmawanrpl@gmail.com

²Email: rajib@stikombinaniaga.ac.id

ABSTRACT

Makalah ini membuat tentang masalah dalam menentukan jalur terpendek atau terdekat menuju lokasi pelanggan yang memberikan keluhan, kemudian teknisi akan melakukan penugasan maintenance ke lokasi pelanggan. Dalam pelaksanaan penugasan manajemen perlu kontrol dan monitoring proses pelaksanaan penugasan oleh teknisi yang berada dilapangan. Pokok permasalahan dalam pengembangan ini adalah belum optimalnya penentuan jalur lokasi pelanggan yang membutuhkan maintenance secara cepat dan tepat. Algoritma yang digunakan dalam aplikasi adalah algoritma floyd-warshall. Pelaksanaan pengembangan ini menggunakan metode RUP (Rational Unified Process), adapun langkah-langkahnya : 1) Permulaan (inception), 2) Perluasan/Perencanaan (elaboration), 3) Konstruksi (construction), dan 4) Transisi (transition). Hasil pengembangan menunjukkan bahwa : 1) Aplikasi yang dibuat sudah dapat menghasilkan solusi optimal dengan mendapatkan dan menentukan jalur terpendek menuju lokasi pelanggan bagi operasional maintenance oleh teknisi, 2) Aplikasi mampu memonitoring proses pelaksanaan penugasan, 3) Nilai jarak tempuh yang dihasilkan dari algoritma floyd-warshall dapat menghitung prediksi biaya pengeluaran bahan bakar kendaraan, dan 4) Hasil unjuk kerja aplikasi masuk dalam kategori "Sangat Layak". Nilai persentase hasil pengujian pada aplikasi untuk setiap faktor yaitu : Functionality sebesar 88,8 %, Efficiency sebesar 86 %, dan Usability sebesar 83 %. Persentase total dari kualitas aplikasi adalah 86,2 %

Keywords: GPS, A-GPS, RUP, Algoritma, Floyd Warshal.

ABSTRAK

This paper makes a problem about determining the shortest or closest path to the location of the customer who is giving the complaint, then the technician will assign maintenance to the customer's location. In the implementation of management assignments, it is necessary to control and monitor the process of carrying out assignments by technicians who are in the field. The main problem in this development is the not yet optimal determination of the customer's location path that requires maintenance quickly and precisely. The algorithm used in the application is the floyd-warshall algorithm. The implementation of this development uses the RUP (Rational Unified Process) method, as for the steps: 1) Beginning (inception), 2) Expansion / Planning (elaboration), 3) Construction (construction), and 4) Transition. The development results show that: 1) Applications that have been made can produce optimal solutions by obtaining and determining the shortest path to the customer's location for maintenance operations by technicians, 2) The application is able to monitor the assignment implementation process, 3) Mileage values generated from the floyd-algorithm warshall can calculate the prediction of vehicle fuel expenses, and 4) the results of the performance of the application fall into the "Very Worthy" category. The percentage value of the test results on the application for each factor is: Functionality of 88.8%, Efficiency of 86%, and Usability of 83%. The total percentage of application quality is 86.2%.

Kata Kunci: GPS, A-GPS, RUP, Algorithm, Floyd Warshal

PENDAHULUAN

Bonet atau disebut juga Bogor Internet merupakan perusahaan yang bergerak dibidang jasa

penyediaan akses *internet*, selain itu juga pembuatan aplikasi dan *website*, *data center* dan *hosting*. Bonet melakukan pelayanan salah satunya dengan datang langsung ke lokasi pelanggan untuk *maintenance* perangkat yang ada di lokasi pelanggan. Sebelumnya para teknisi melakukan pertemuan untuk menentukan siapa saja yang akan berangkat dan menuju lokasi mana saja.

Setiap hari biasanya satu teknisi mendapat bagian *maintenance* lebih dari satu pelanggan, hal tersebut tidak termasuk dengan keluhan pelanggan yang belum terdata sebelumnya. Oleh karena itu keluhan pelanggan yang baru masuk/mendadak tersebut harus ditentukan kembali siapa teknisi yang mampu menanganinya dari sisi jarak lokasi teknisi dengan lokasi pelanggan.

Saat ini, proses untuk menentukan teknisi yang harus menangani keluhan pelanggan yang baru masuk sedangkan semua teknisi berada di lokasi pelanggan yaitu dengan menghubungi teknisi satu persatu kemudian menanyakan *progress* dari *maintenance* yang dikerjakan. Hal tersebut kurang efektif dan efisien karena bagian *customer service* harus menghubungi teknisi berulang kali dan kemudian memberitahukan kepada pelanggan mengenai keluhannya itu yang akan *dimaintenance* selanjutnya.

Monitoring teknisi yang berada di lapangan dengan menggunakan pesan singkat (*sms*) dan telepon (*call*). Kedua hal tersebut lebih memakan banyak biaya dan tidak *realtime monitoring*. Dewasa ini, perangkat *smartphone* sudah sangat canggih, salah satunya adanya teknologi *GPS*. *Global Positioning System (GPS)* yaitu sistem navigasi berbasis satelit yang terdiri dari 24 jaringan satelit yang ada di orbit. *GPS* pada awalnya diperuntukan bagi penggunaan militer, namun pada tahun 1980 pemerintah membuat sistem yang tersedia untuk penggunaan sipil. Satelit *GPS* berputar mengelilingi bumi selama 12 jam sehari di dalam orbit dan mengirimkan sinyal yang akurat ke bumi. *GPS Receiver* mengambil informasi itu dengan perhitungan *triangulation* menghitung lokasi user dengan tepat. *GPS Receiver* membandingkan waktu sinyal dikirim dengan sinyal tersebut diterima. Dari informasi tersebut di dapatlah jarak satelit. Dengan perhitungan jarak, *GPS Receiver* dapat melakukan perhitungan dan menentukan posisi user dan menampilkannya dalam *maps*.

Dalam melakukan *maintenance* ke lokasi-lokasi pelanggan tentu akan adanya biaya perjalanan yang diperlukan oleh teknisi. Biaya perjalanan dihitung per kilometer dari titik awal teknisi berada menuju titik lokasi pelanggan, secara pulang-pergi. Biaya perjalanan mencakup biaya bahan bakar kendaraan motor. Dikarenakan menghitung secara manual dengan melihat *odometer* pada kendaraan cukup menyulitkan karena harus selalu mengukur kilometer titik awal dan kilometer titik akhir dan sering terlupakan oleh teknisi dan akhirnya tetap menggunakan data kira-kira untuk menentukan jarak awal dan jarak akhir yang telah ditempuh. Akhirnya selama ini dalam menentukan biaya bahan bakar kendaraan dihitung secara patokan, artinya untuk tiga kali perjalanan dikenai biaya Rp 20.000 per teknisi/per kendaraan.

Berdasarkan permasalahan di atas maka dilakukan penelitian dengan menerapkan perhitungan untuk menghasilkan solusi jalur terpendek yang lebih optimal yaitu dengan menerapkan Algoritma *Floyd-Warshall*. Algoritma ini dapat menentukan jalur terpendek bagi teknisi dalam *me-maintenance* perangkat menuju lokasi-lokasi pelanggan dengan lebih akurat dan diharapkan dari pengembangan dengan aplikasi mampu memberikan implikasi positif dari segi biaya perjalanan yaitu prediksi biaya bahan bakar kendaraan yang efisien dihitung dari jarak optimal per penugasan.

Dalam menyelesaikan masalah penentuan jarak terpendek dapat menggunakan beberapa algoritma, seperti *Algoritma Dijkstra*, *Algoritma Bellman-Ford*, *Algoritma Greedy*, *Algoritma Ant Colony*, dan lain-lain. Merujuk pada jurnal/penelitian yang dilakukan oleh Ahmad Adityo Anggoro dimana, pemanfaatan *Algoritma Floyd-Warshall* dapat disimpulkan sebagai berikut “*Setelah diimplementasikan Algoritma Floyd-Warshall mampu mengoptimalkan pencarian rute terdekat untuk pencarian tempat*”. (Anggoro, 2015)

Dari penelitian tersebut dapat dilihat bahwa, algoritma tersebut memiliki kelebihan dan kekurangannya masing-masing, inilah yang menjadi dasar dan alasan mengapa *Algoritma Floyd-Warshall* yang digunakan untuk membantu menyelesaikan masalah dalam penelitian ini, karena algoritma ini melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Maksudnya, solusi-solusi dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu. Sehingga dapat mempermudah pengguna dalam menentukan jalur terpendek yang akan digunakan dalam *maintenance* perangkat ke lokasi pelanggan dengan bantuan teknologi *GPS* yang mampu menampilkan secara *realtime* proses *monitoring* yang berjalan.

1. Identifikasi Masalah

- a. Tidak tercapainya dari sisi efisiensi dalam penugasan dikaitkan dengan faktor biaya
- b. *Monitoring* koordinasi antar teknisi dalam *maintenance* membutuhkan waktu lama

2. Tujuan Penelitian

- a. Menentukan jalur terpendek menuju lokasi pelanggan yang optimal bagi operasional *maintenance* teknisi
- b. Membuat aplikasi yang mampu *me-monitoring* proses *maintenance* oleh teknisi di lapangan dengan *real time* berbasis *GPS*
- c. Mengetahui prediksi pengeluaran biaya bahan bakar kendaraan
- d. Mengetahui nilai persentase kelayakan aplikasi

3. Tinjauan Pustaka

a. Pencarian Rute Terpendek

Lintasan terpendek yaitu lintasan minimum yang diperlukan untuk mencapai dari suatu titik(tempat) ke titik(tempat) yang lain. Lintasan terpendek (*short path*) merupakan salah satu masalah yang dapat diselesaikan dengan menggunakan graf. Graf adalah sekumpulan titik (*node*) di dalam bidang dua dimensi yang dihubungkan dengan sekumpulan garis (*edge*). Sebuah *graph* dibentuk dari sekumpulan titik yang dihubungkan dengan garis-garis.

Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak-kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul.

Jika diberikan graf berbobot, masalah lintasan terpendek adalah bagaimana kita mencari sebuah jalur pada graf yang meminimumkan jumlah bobot sisi pembentuk jalur tersebut. Terdapat bermacam lintasan persoalan lintasan terpendek antara lain (Yusaputra, 2013) :

- 1) Lintasan terpendek antara dua buah simpul tertentu. (*a pair shortest path*)
- 2) Lintasan terpendek antara semua simpul. (*all pairs shortest path*)
- 3) Lintasan terpendek dari simpul tertentu ke semua simpul yang lain. (*single-source shortest path*)
- 4) Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu. (*intermediate shortest path*).

b. Algoritma Floyd Warshal

Algoritma Warshall menggunakan matriks *boolean* untuk mencari ada tidaknya jalan dari titik satu ke titik yang lain. Algoritma ini kemudian dikembangkan untuk mengurangi waktu komputasi sehingga hanya menggunakan satu perkalian matriks. Bentuk algoritma untuk mencari R^+ adalah sebagai berikut :

```
Step 1: T = R
Step 2: For k=1 to n do
            For i=1 to n do
                For j=1 to n do
                    T[i, j] = T[i, j] + (T[i, k] . T[k, j])
Step 3:      R* = T.
```

Algoritma Warshall dikembangkan oleh R.W. Floyd sehingga matriks merupakan graf berbobot dan bukan lagi matriks *boolean*. Algoritma ini digunakan untuk mencari jarak terpendek antara semua titik dalam graf. Ketiadaan garis yang menghubungkan sebuah pasangan dilambangkan dengan tak-hingga. Dalam pengertian lain algoritma Floyd-Warshall adalah algoritma yang akan memilih satu jalur terpendek dan teraman dari beberapa alternatif jalur yang dihasilkan dari proses kalkulasi.

Algoritma Floyd-Warshall mencoba untuk memberikan solusi optimum yang memiliki pemikiran terhadap konsekuensi yang ditimbulkan dari pengambilan keputusan pada suatu

tahap. Algoritma Floyd-Warshall mampu mengurangi pencarian keputusan yang tidak mengarah ke solusi. Prinsip yang dipegang oleh algoritma ini adalah prinsip optimalitas, yaitu jika solusi per-tahap optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke- i) juga optimal (Yusaputra, 2013). Bentuk algoritma untuk mencari R^+ adalah sebagai berikut :

Step 1: Jika antara i - j merupakan jalan (*edge*), $T[i, j]$ berisi panjang jalan. Jika tidak, $T[i, j] = \infty$

Step 2: For $k=1$ to n do

For $i=1$ to n do

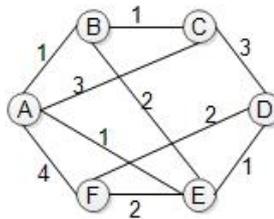
For $j=1$ to n do

$T[i, j] = \min \{ T[i, j], T[i, k] + T[k, j] \}$

Step 3: $R^+ = T$.

(Adipranata dkk., 2008)

Algoritma Floyd-Warshall sangat efisien dari sudut pandang penyimpanan data karena dapat diimplementasikan dengan hanya pengubahan sebuah matriks jarak. Berikut ini adalah contoh dari penggunaan algoritma Floyd-Warshall. Misalkan ada graf berbobot seperti pada Gambar 1.



Gambar 1. Graf berbobot untuk algoritma floyd-warshall

Graf berbobot ini dapat dijadikan sebagai matriks inisialisasi seperti pada Tabel 1.

Tabel 1. Inisialisasi Matriks T

	A	B	C	D	E	F
A	0	1	3	∞	1	4
B	1	0	1	∞	2	∞
C	3	1	0	3	∞	∞
D	∞	∞	3	0	1	2
E	1	2	∞	1	0	2
F	4	∞	∞	2	2	0

c. Global Positioning System (GPS)

GPS (Global Positioning System) adalah sistem yang digunakan untuk menentukan letak di permukaan bumi dengan bantuan penyalarsaan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit untuk mengirimkan sinyal gelombang mikro ke bumi. Sinyal tersebut kemudian diterima oleh alat penerima di permukaan bumi dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan *GPS* antara lain:

GLONASS dari Rusia, *Galileo* dari Uni Eropa, dan *IRNSS* dari India.

GPS sendiri memiliki 2 jenis, yaitu *GPS* dan *A-GPS*, dimana sistem *GPS* ini sering kita jumpai pada ponsel yang ada dipasaran sekarang ini. Berikut penjelasan dari perbedaan *GPS* dengan *A-GPS*.

1. *GPS*

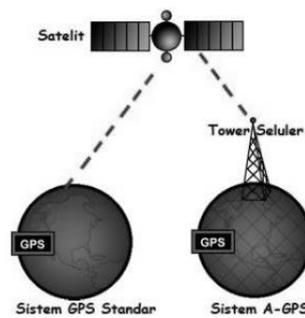
Perangkat *GPS* biasanya merupakan perangkat khusus *GPS*, bukan ponsel dan cara menentukan lokasi *GPS* tersebut adalah dibutuhkan minimal 3 satelit yang membentuk segitiga untuk mendapatkan lokasi yang akurat. Dari proses tersebut akan didapatkan hasil berupa *Time To First Fix (TTFF)* atau berapa lama sebelum posisi kita terdeteksi. Dibutuhkan waktu mulai dari 30 detik hingga 12 menit untuk menentukan lokasi ketika perangkat khusus *GPS* dinyalakan. Cukup lama namun setidaknya pengguna tidak perlu memiliki akses ke operator ponsel (pulsa) untuk mengoperasikan *GPS*.

Waktu tersebut tergantung pada lokasi Anda, jumlah gangguan dan kondisi cakrawala (cuaca). Pada daerah terbuka akan lebih cepat mendapatkan sinyal dibandingkan di daerah perkotaan yang banyak gedung-gedung tinggi atau posisi kita yang berada di dalam ruangan yang dapat mengganggu penerimaan sinyal satelit. Data waktu dari 3 satelit tersebut akan dikomputasi oleh modul *GPS* di ponsel dan akhirnya akan dihasilkan informasi tentang posisi yang berupa *latitude* (garis lintang) dan *longitude* (garis bujur) dan lokasi dalam peta.

GPS membutuhkan 3 komponen dalam proses penentuan posisi, yaitu satelit, *Receiver GPS*, dan posisi yang baik (bebas halangan). Perangkat *GPS* menangkap 3 sinyal dari 24 satelit *GPS* yang ada di luar angkasa. Supaya perangkat *GPS* dapat menangkap sinyal dengan baik, perangkat harus berada di luar ruangan, bahkan harus di bawah langit terbuka. Kekuatan sinyal dapat berkurang kalau perangkat *GPS* berada di bawah pohon, di bawah gedung-gedung pencakar langit, di dalam kendaraan dan sinyal hampir bisa dipastikan menghilang kalau perangkat *GPS* ada di dalam gedung.

2. *A-GPS*

A-GPS (Assisted-Global Positioning System) merupakan suatu teknologi penyempurnaan dari *GPS*. Teknologi ini menggunakan *server* bantuan untuk mempercepat waktu penentuan posisi lokasi *GPS* tersebut, sehingga tidak perlu lagi proses komputasi data dari 3 satelit secara langsung dari ponsel. *A-GPS* membutuhkan 3 komponen dalam proses penentuan posisi, yaitu satelit, *Assistance Server* (operator), *Receiver A-GPS*. Karena *A-GPS* dan *server* bantuan berbagi tugas dalam penentuan posisi, maka proses akan lebih cepat dan lebih efisien dibanding *GPS* biasa.

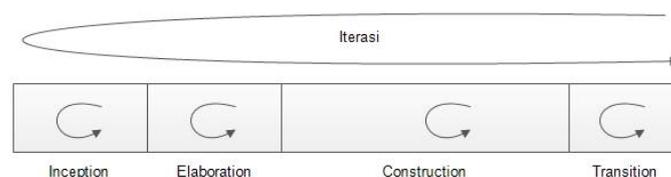


Gambar 2. Penentuan posisi *gps* dan *a-gps*

METODE

Rational Unified Process (RUP) adalah tahapan pengembangan sistem secara iteratif khusus untuk pemrograman berorientasi objek. Pendekatan pengembangan perangkat lunak pada *RUP* dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*arsitecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). *RUP* merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). *RUP* menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. *RUP* adalah sebuah produk proses perangkat lunak yang dikembangkan oleh Rational Software yang diakuisisi oleh IBM pada tahun 2003 (Pressman, 2010).

Dengan mengacu pada model pengembangan (*Unified Modeling Language*), maka penelitian ini mengambil 4 fase dalam proses ini yang dimiliki oleh *RUP*. Iteratif bisa dilakukan di dalam setiap tahap, atau iteratif tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang *inkremental* (bertambah menaik) di mana setiap interaksi akan memperbaiki interaksi berikutnya.



Gambar 3. Alur Hidup RUP

HASIL DAN PEMBAHASAN

Hasil

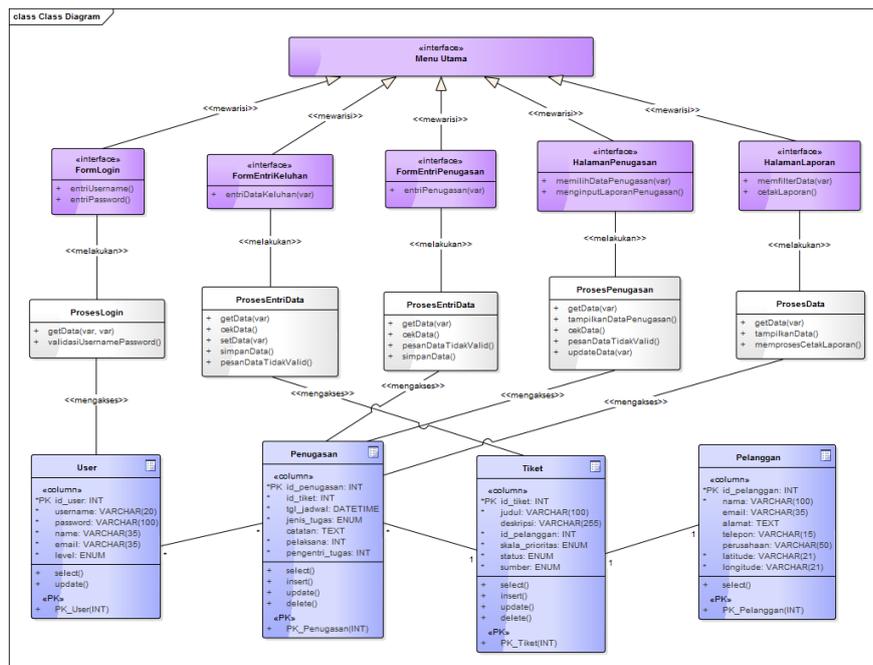
1. Permulaan (Inception)

Penugasan oleh teknisi ke tempat pelanggan dijadwalkan oleh *customer support*. Penugasan bersumber baik dari keluhan langsung oleh pelanggan kepada *customer support* melalui telepon atau aplikasi *whatsapp* atau dari admin *project* atau dari bagian *marketing* yang kemudian akan

dilanjutkan ke bagian *customer support* untuk kemudian dibuatkan jadwal penugasan. Dalam hal ini bagian *marketing* dan *admin project* hanya bisa memberikan inputan berupa keluhan pelanggan yang akan dikirimkan ke *customer support*. *Customer support* sendiri bertugas menindaklanjuti keluhan tersebut untuk dibuatkan penjadwalan penugasan kepada bagian teknisi dan juga *customer support* bisa melakukan penginputan *data master* yaitu data utama/induk dan data pendukung di dalam data penugasan dan penjadwalan. *Customer support* juga bisa membuat laporan data. Setelah *customer support* menerima keluhan pelanggan dan menginputkan data keluhan kemudian *customer support* membuat jadwal penugasan yang kemudian dikirimkan ke bagian teknisi. Bagian teknisi akan menerima notifikasi mengenai penugasan baru ke lokasi pelanggan, kemudian teknisi mengakses jadwal penugasan, termasuk detail data pelanggan dan lokasi pelanggan dalam peta. Dalam pelaksanaan tugas oleh teknisi, bagian *customer support* dan manajemen dapat memantau secara *real time monitoring* pekerjaan yang dilakukan teknisi di lokasi pelanggan, baik berupa *real time* lokasi teknisi maupun progres penugasan yang sedang dikerjakan. Kemudian teknisi dapat melakukan penginputan data laporan dari pekerjaan yang dilakukan, jika pekerjaan belum selesai maka bagian *customer support* akan membuat jadwal penugasan baru yang akan dikerjakan oleh teknisi dilain waktu, jika pekerjaan/keluhan sudah terselesaikan maka tugas selesai, selanjutnya *customer support* membuat laporan akhir untuk penugasan yang bersangkutan.

2. Perencanaan (Elaboration)

Proses perencanaan digambarkan dalam diagram kelas dimana dalam diagram tersebut menggambarkan bagaimana system berjalan dengan mengakses database yang telah dibentuk.



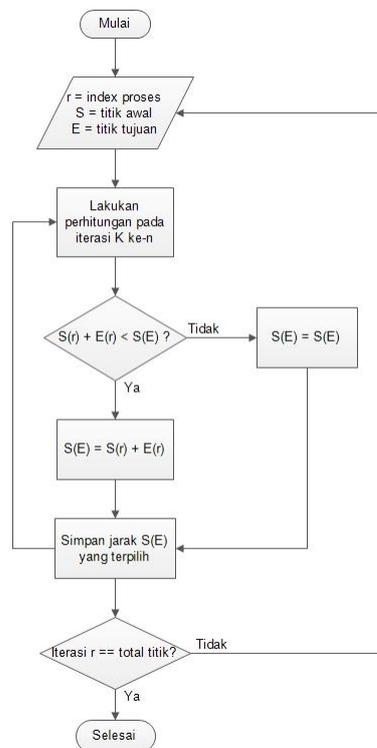
Gambar 4. Diagram Kelas

3. Konstruksi (Construction)

Alur pencarian jalur terpendek dengan menggunakan algoritma floyd-warshall, dapat dilihat pada Gambar 4.9.

Proses dari alur algoritma floyd-warshall yaitu sebagai berikut :

- Menentukan index proses (r), dan menginputkan nilai titik awal (S) dan nilai titik tujuan (E).
- Lakukan perhitungan pada iterasi/pengulangan nilai K ke- n . Setiap sel matriks dilakukan pengecekan apakah penjumlahan antara nilai titik awal $S(r)$ dengan nilai titik tujuan $E(r)$ lebih kecil dari nilai titik sebenarnya $S(E)$.
- Jika tidak, maka jarak yang dipilih adalah nilai titik sebenarnya $S(E)$.
- Jika ya, maka jarak nilai titik sebenarnya $S(E)$ diganti dengan penjumlahan antara nilai titik awal $S(r)$ dengan nilai titik tujuan $E(r)$.
- Proses dilakukan hingga iterasi terakhir yaitu nilai iterasi sama dengan total titik atau *node*.



Gambar 5. Alur algoritma Floyd-Warshall

4. Transisi (Transition)

Transisi yaitu dimana aplikasi sudah diinstall pada perangkat pengguna, dan digunakan oleh masing-masing pengguna.

Pembahasan

Uji Kelayakan Perangkat Lunak Aspek *Functionality*, *Efficiency*, dan *Usability*

Tabel2 . Hasil Perhitungan Uji Kelayakan

Pengguna	Functionality	Efficiency	Usability
1	23	8	16
2	24	8	16
3	22	10	19
4	21	8	18
5	21	9	14
Total Skor	111	43	83
Skor Max	125	50	100
Persentase	88,8	86	83

Hasil perhitungan untuk mengetahui kualitas perangkat lunak dari segi *functionality*, *efficiency*, dan *usability*, kemudian dibandingkan dengan tabel persentase kelayakan menurut Arikunto Tingkat kelayakan masing-masing faktor kualitas perangkat lunak setelah membandingkan dengan tabel interpretasi persentase kelayakan perangkat lunak dapat dilihat pada Tabel 4.10.

Tabel 3. Persentase Kelayakan Tiap Faktor

No	Faktor	Persentase	Tingkat Kelayakan
1	Functionality	88,8 %	Sangat Layak
2	Efficiency	86 %	Sangat Layak
3	Usability	83 %	Sangat Layak

SIMPULAN DAN SARAN

Simpulan

Setelah melakukan tahap *RUP*, pengembangan ini berhasil membuat aplikasi pengelolaan keluhan dan penugasan teknisi berbasis *GPS* dengan algoritma floyd-warshall. Berdasarkan hasil dari implementasi sistem maka dapat disimpulkan bahwa:

1. Aplikasi yang dibuat sudah dapat menghasilkan solusi optimal yaitu mendapatkan dan menentukan jalur terpendek menuju lokasi pelanggan bagi operasional *maintenance* dari masing-masing lokasi teknisi yang berbeda yang berada di lapangan.
2. Aplikasi yang dibuat mampu *memonitoring* proses *maintenance* oleh teknisi secara *real time* dengan berbasis *GPS*.
3. Nilai jarak tempuh yang dihasilkan dari algoritma floyd-warshall dapat menghitung prediksi biaya pengeluaran bahan bakar kendaraan.

4. Hasil unjuk kerja sistem pengelolaan keluhan penugasan masuk dalam kategori “Sangat Layak”. Nilai persentase hasil pengujian pada sistem untuk setiap faktor, yaitu : *functionality* sebesar 88,8 %, *efficiency* sebesar 86 %, dan *usability* sebesar 83 %. Persentase total dari kualitas aplikasi adalah 86,2 % (Sangat Layak).

Saran

Saran yang ingin diajukan dari hasil penelitian pengembangan ini adalah dikembangkan ke versi yang lebih baik tidak hanya dari satu *platform mobile* saja seperti, menggunakan *platform IOS*, dan *windows phone*, menggunakan data lokasi pelanggan yang sudah terpetakan dengan baik secara digital, kemudian memanfaatkan *wifi operator* untuk mendapatkan titik koordinat lokasi pelanggan agar lokasi lebih mudah terdeteksi terutama pada calon pelanggan, perlu membuat algoritma lebih baik karena aplikasi ini hanya melibatkan parameter jarak saja masih belum melibatkan parameter lain seperti, jumlah penugasan yang sudah dilaksanakan per teknisi (jam kerja), waktu, beban kerja, waktu selesai pengerjaan, *reward* pelanggan, dan tingkat kesulitan penugasan agar proses rekomendasi penugasan pada aplikasi lebih fleksibel.

DAFTAR RUJUKAN

- Adipranata, R., Desiree, F. J., & Handojo, A. (2008). Pencarian Rute Optimal Menggunakan Metode Transitive Closure. Surabaya: Universitas Kristen Petra.
- Anggoro, A. A. (2015). Pencarian Titik Lokasi Dengan Pemanfaatan Algoritma Floyd-Warshall Sebagai Perhitungan Jarak Terdekat Di Institut Teknologi Bandung . Bandung: STMIK LPKIA.
- Araaf, G. A. (2014). Implementasi Algoritma Bellman-Ford Dan Floyd- Warshall Untuk Mencari Rute Terpendek. Yogyakarta: AMIKOM.
- Arikunto, S. (2002). *Prosedur Penelitian Suatu Pendekatan Praktek*. Jakarta: Rineka Cipta.
- Az-Zahra, R. R. (2017). Implementasi Algoritma Floyd Warshall Untuk Pencarian Jalur Terpendek Non Player Character (NPC) Pada Game 3D Pembelajaran Kosakata Bahasa Arab.
- Bonet Utama. (2016). *Panduan Pelayanan Prima PT Bonet Utama*. Bogor: PT Bonet Utama.
- Budianto, A. (2014). Analisis Dan Perancangan Game Edukasi "Need For Safety". Yogyakarta: Universitas Negeri Yogyakarta.
- Fernandi, E., Alfandri, F., Putri, G. A., & Mathias, R. (2013). Analisis dan Desain Berorientasi Objek (OOAD) dan Diagram Activity. Depok: Universitas Gunadarma.
- Honda, C. (2017). "*Daftar Konsumsi BBM Sepeda Motor Honda*". Dipetik Oktober 20, 2017, dari <https://www.hondacengkareng.com/daftar-konsumsi-bbm-sepeda-motor-honda/comment-page-3>

- Kriswanto, R., Bendi, K. J., & Aliyanto, A. (2014). Penentuan Jarak Terpendek Rute Transmisi dengan Algoritma Floyd-Warshall. Palembang: Sekolah Tinggi Teknik Musi.
- Kusuma, R. (2014). "*Matematika Diskrit - Lintasan Terpendek (Shortest Path)*". Dipetik Mei 11, 2017, dari <https://rahadikusuma.blogspot.co.id/2014/01/matenatika-diskrit-lintasan-terpendek.html>
- Mikkonen, J. (2013, Maret 23). *PHP Floyd Warshall Algorithm*. Dipetik Nopember 21, 2017, dari <http://julmis.julmajanne.com/index.php/FloydWarshall>
- Modulmakalah. (2015). "*Pengertian dan Contoh Analisis Isi Dalam Metode Penelitian*". Dipetik Juli 11, 2017, dari <http://modulmakalah.blogspot.co.id/2015/11/pengertian-dan-contoh-analisis-isi.html>
- Munir, R. (2010). *Matematika Diskrit*. Bandung: Informatika.
- Pertamina. (2017, Oktober 10). "*Daftar Harga BBK Tmt 11 Oktober 2017*". Dipetik Oktober 20, 2017, dari <http://www.pertamina.com/id/news-room/announcement/daftar-harga-bbk-tmt-11-oktober-2017>
- Pressman, R. S. (2010). *Rekayasa Perangkat Lunak - Buku Satu* (Edisi 7 (Pendekatan Praktisi) ed.). Yogyakarta: ANDI.
- Romelta, E. (2003). *Metode Pencarian Lintasan Terpendek Dalam Graf*. Bandung: ITB.
- Rosa, & Shalaahuddin. (2014). *Rekayasa Perangkat Lunak Berstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Safaat, N. (2012). *Android Pemrograman Aplikasi Mobile Smartphone*. Bandung: Informatika.
- Setiawan, W. (2015). "*Cara Kerja GPS*". Dipetik Maret 24, 2017, dari <https://wirasetiawan29.wordpress.com/2015/05/10/cara-kerja-gps>
- Siang, J. J. (2009). *Matematika Diskrit dan Aplikasinya pada Ilmu*. Yogyakarta: Andi.
- Suhendra, Z. (2015, Juli 24). "*1 Liter Premium Bisa Tempuh 10 km, Peralite?*". Dipetik Oktober 20, 2017, dari <https://finance.detik.com/energi/2974391/1-liter-premium-bisa-tempuh-10-km-peralite>
- Supriyanto. (2010). *Perbandingan Metode Terstruktur (Tradisional) Dan Metode Object-Oriented (Oo) Pada Analisis Dan Design Sistem*. Bogor: IPB.
- Wahyunita, L. (2015). "*Algoritma Pencarian Lintasan Jalur Terpendek*". Dipetik April 22, 2017, dari https://www.slideshare.net/lailiwahyunita/algoritma-pencarian-lintasan-jalur-terpendek?from_action=save
- Yusaputra, R. (2013). *Aplikasi Mobile Pencarian Rute Terpendek Lokasi Fasilitas Umum Berbasis Android Menggunakan Algoritma Floyd-Warshall*. Pekanbaru: Universitas Islam Negeri Sultan Syarif Kasim Riau.