

Desain Aplikasi Pencarian Kontrakan Kota Tarakan Berbasis Mobile Menggunakan Metode Algoritma Dijkstra

Evi Marliyani, Moh. Masduki Syahlan, Obert, Roman Gusmana
Program Studi Sistem Informasi, STMIK PPKIA Tarakanita Rahmawati

Abstraksi— Masa sekarang ini pencarian kontrakan di kota Tarakan baik untuk mahasiswa, pegawai, dan masyarakat umum, dalam pencariannya saat ini kebanyakan masih menggunakan sistem dari mulut ke mulut serta media sosial. Dengan sistem tersebut akan sulit dalam pencarian informasinya seperti, dalam perjalanan pencarian alamatnya juga sangat lama dalam melakukan pencarian kontrakan di kota Tarakan biasanya mahasiswa, pegawai, maupun masyarakat umum biasanya hanya mendengarkan atau mengetahui informasi yang disampaikan dari masyarakat satu ke masyarakat lainnya sehingga informasi yang didapatkan tidak akurat.

Algoritma Dijkstra adalah suatu algoritma untuk menentukan rute dengan jarak yang pendek. Diasumsikan bahwa semua jarak yang ditempuh bernilai positif. Ide algoritma ini berdasarkan fakta setiap jarak minimum terdapat lebih dari satu, tetapi sebenarnya hanya ada satu jarak untuk ditempuh. Hal ini terjadi karena semua jarak bernilai positif.

Menurut hasil analisa yang didapatkan oleh penulis dalam melakukan penelitian pada, Metode Algoritma Dijkstra ialah bahwa metode yang digunakan masih sangat belum efisien dalam hal menentukan rute terpendek karena metode ini tidak menghitung dari keseluruhan jalur yang ada namun hanya menghitung node yang terdekat dari titik awal dan akan melakukan perhitungan ketika node memiliki percabangan dan akan memilih nilai yang terkecil dari node yang memiliki percabangan.

Kata Kunci— Aplikasi, Mobile, Rumah Kontrakan, Metode Algoritma Dijkstra

I. PENDAHULUAN

Dalam era globalisasi saat ini, teknologi terus berkembang pesat untuk mempermudah berbagai aktivitas manusia, baik dalam bidang transportasi, komunikasi, kesehatan, pendidikan, dan lainnya. Teknologi seperti internet dan berbagai aplikasi telah memungkinkan manusia berkomunikasi lebih mudah dan memperoleh informasi dari berbagai belahan dunia. Melalui internet, orang dapat menyampaikan dan mengakses informasi kapan saja dan di mana saja, termasuk dalam dunia bisnis, seperti penggunaan ponsel untuk transaksi atau komunikasi.

Di kota Tarakan, pencarian kontrakan untuk mahasiswa, pegawai, dan masyarakat umum masih banyak dilakukan melalui sistem tradisional, seperti dari mulut ke mulut atau menggunakan media sosial. Sistem ini seringkali kurang efisien karena informasi yang didapatkan bisa kurang akurat, alamat sulit ditemukan, dan kontrakan yang ditawarkan

sering kali tidak sesuai dengan ekspektasi. Selain itu, pemilik kontrakan tidak selalu memperbarui informasi secara berkala, sehingga menyulitkan calon penyewa dalam pencarian.

Untuk mengatasi masalah ini, diperlukan media informasi yang mudah diakses dan dapat memudahkan proses pencarian kontrakan. Oleh karena itu, penulis mengembangkan sebuah aplikasi yang menyediakan informasi lengkap mengenai kontrakan, termasuk alamat, nomor telepon pemilik, dan detail lainnya. Aplikasi ini juga dilengkapi dengan fitur pencarian alamat menggunakan Google Maps untuk mempermudah navigasi. Selain membantu pencari kontrakan, aplikasi ini juga diharapkan dapat mendukung bisnis pemilik kontrakan.

Dalam menyelesaikan masalah pencarian rute terpendek menuju lokasi kontrakan, penulis menggunakan Algoritma Dijkstra. Algoritma ini terkenal efektif dalam menemukan rute tercepat. Aplikasi ini dirancang menggunakan Eclipse, PHP, dan MySQL sebagai basis data untuk mendukung fungsionalitas aplikasi secara optimal.

II. TINJAUAN PUSTAKA

Desain dan perancangan perangkat lunak bertujuan untuk membangun sistem yang dapat memenuhi kebutuhan fungsional, target kinerja, serta mematuhi batasan biaya, waktu, dan sumber daya yang tersedia. Kualitas perangkat lunak biasanya dinilai berdasarkan tingkat kepuasan pengguna terhadap perangkat yang digunakan [1].

Aplikasi adalah solusi teknologi untuk pemrosesan data yang diharapkan pengguna. Desain aplikasi pencarian, khususnya, adalah perangkat lunak yang dikembangkan untuk memenuhi kebutuhan fungsional guna memudahkan pengguna dalam melakukan pencarian data yang diinginkan [2].

Desain Aplikasi Pencarian adalah sebuah pembangunan perangkat lunak untuk mengkonstruksi sistem yang memberikan kepuasan akan spesifikasi kebutuhan fungsional. Dimana aplikasi adalah program yang di buat untuk mempermudah dalam melakukan pencarian data yang diinginkan bagi penggunanya.

Android merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat mobile, seperti smartphone dan tablet. Sistem ini bersifat open source, sehingga memungkinkan pihak lain untuk mengembangkannya lebih lanjut. Perkembangan Android cukup pesat dan kini mendukung berbagai perangkat dengan spesifikasi beragam, dari low-end hingga high-end [3].

Salah satu algoritma yang digunakan dalam pemecahan masalah rute terpendek adalah Algoritma Dijkstra, yang ditemukan oleh Edsger Wybe Dijkstra pada tahun 1959. Algoritma ini berfungsi untuk menentukan rute terpendek antara dua titik pada graf yang terhubung oleh jalur dengan bobot positif

Algoritma Dijkstra, dinamai menurut penemunya Edsger Dijkstra adalah sebuah algoritma rakus (*greedy algorithm*) dalam memecahkan permasalahan jarak terpendek dengan bobot-bobot sisi (*edge weights*) melambangkan jarak antara lokasi-lokasi tersebut, maka algoritma Dijkstra dapat digunakan untuk menemukan jarak terpendek antara dua lokasi.

Algoritma Dijkstra adalah algoritma "rakus" (*greedy*) yang berfokus pada solusi terbaik di setiap langkah tanpa memikirkan konsekuensi di masa depan. Prinsipnya adalah memilih jalur dengan bobot terendah yang tersedia pada setiap langkah, namun semua bobot dalam graf harus bernilai positif. Sebagai contoh, jika ada node A sebagai titik awal dan B sebagai tujuan, jalur dari A ke C dengan bobot 3, dan dari C ke B dengan bobot 2 menghasilkan total jarak 5. Namun, jika ada jalur langsung dari A ke B dengan bobot 4, maka rute terpendek adalah A langsung ke B.

III. ANALISA DAN DESAIN SISTEM

A. Analisa Sistem

Dalam penerapan algoritma djikstra ini metode yang digunakan yaitu untuk menentukan rute terpendek oleh karena itu, aplikasi ini digunakan untuk menemukan lintasan terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik yang menggambarkan lokasi kontrakan. Garis menggambarkan jalan dan bobot menggambarkan jarak, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. Dengan kata lain algoritma ini menghitung lintasan berdasarkan jarak terpendek yang ditempuh di tiap-tiap jalan, sehingga dapat mencari tempat tujuan wisata dengan jarak terpendek.

Desain aplikasi yang Penulis buat, terdapat beberapa tampilan menu pencarian, peng input-an data bagi pemilik kontrakan, lokasi terdekat, petunjuk, bantuan, dan tentang. pada saat *client* melakukan pencarian kontrakan maka akan muncul data kontrakan dan pada saat menekan lokasi terdekat maka, yang dilakukan sistem pertama kali yaitu mendapatkan titik koordinat keberadaan client, selanjutnya yang dilakukan client memilih tempat kontrakan yang akan didatangi, sistem membaca titik koordinat tempat kontrakan yang telah disimpan sebelumnya di dalam database, lalu menampilkan icon tempat kontrakan di peta. selanjutnya client menekan tombol cari untuk menampilkan rute ke tempat tujuan, sistem melakukan pengujian dengan menggunakan algoritma djikstra untuk rute terpendek ke tempat tujuan, setelah di temukan rutenya. sistem akan menampilkan garis dari titik client ke tempat kontrakan yang telah di pilih.

Peta atau map yang digunakan adalah sebagai fungsi yang menampilkan letak dari lokasi atau koordinat titik si pencari kontrakan ke lokasi kontrakan yang akan dituju. sehingga memberi kemudahan dalam mengakses sebuah informasi dalam melakukan pencarian kontrakan.

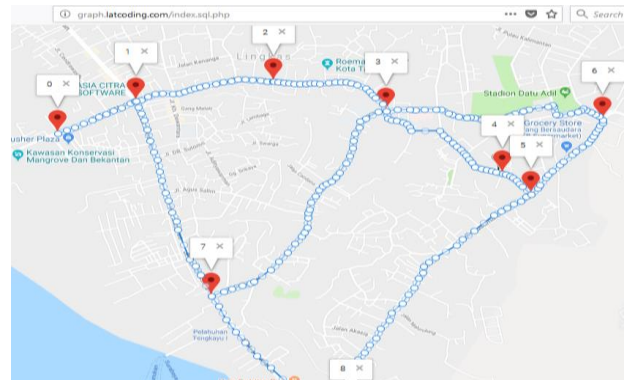
Berikut ini adalah data kontrakan yang telah dihimpun

Tabel 3.1. Data Kontrakan

Nama Lengkap	Kelurahan	Alamat Lengkap
Ibu Eva	Pamusian	Markoni Gang 5 RT 22 No 91-92
Bapak Budianto	Juata Permai	Perumahan PNS Blok B 242

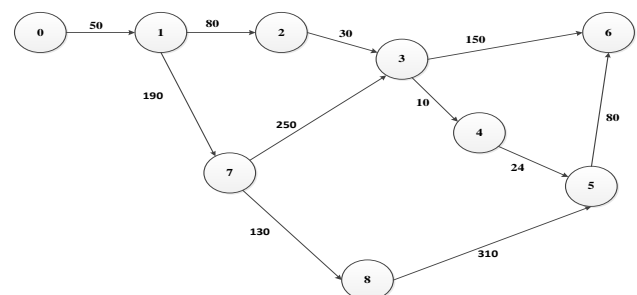
Ibu Yuni	Pamusian	Jl. Ladang Dalam Rt 26
SBapak Sukiran	Kampung 1Skip	Jl. Matahari
Bapak Ady	Kampung 1 Skip	Jl. Matahari
Ibu Ita Masitah	Pamusian	Jl. Jendral Sudirman
Bapak Fery	Pamusian	Kampung Baru
Ibu Dwi Lesmidyaerti	Juata Permai	Persemaian Belakang Kandang Buaya
Bapak Andanu	Karang Anyar Pantai	Jl. Sesanip Depan SMK N 2 Tarakan
Ibu Dewi Novita Sari	Kampung 4	Kampung 4 Gang Ayam

Studi kasus pada peta kontrakan yang di kota Tarakan adalah mencari rute terpendek dari titik awal berada di Jl. Gajah Mada menuju kontrakan Ibu Eva yang berada di Jl Imam Bonjol, Markoni, menggunakan Algoritma Dijkstra, seperti Gambar 3.1.



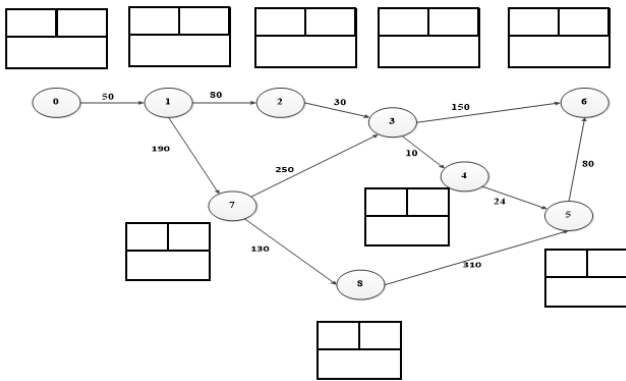
Gambar 3.1. Studi kasus Algoritma Dijkstra

Untuk mempermudah penjelasan dibentuk seperti Gambar 3.2 dibawah ini.



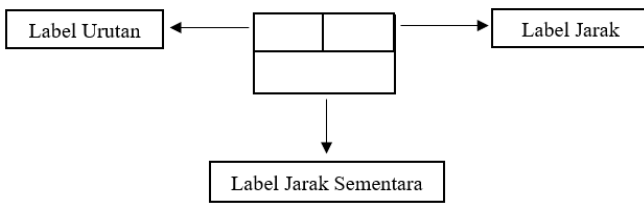
Gambar 3.2. Studi kasus Algoritma Dijkstra

Nilai jarak yang ada pada grafik berdasarkan satuan kilometer dan meter, dimulai dari node 0 yaitu Jl Gajah Mada adalah titik awal dan node 4 adalah kontrakan Ibu Eva adalah tujuan, kemudian mencari rute manakah yang harus dilalui dan melewati jarak yang paling pendek. Untuk mendapatkan rute tersebut, maka grafik diatas ditambahkan beberapa tabel untuk mengisi nilai rute yang akan dilalui oleh pencari kontrakan. Seperti Gambar 3.3 berikut.



Gambar 3.3. Membuat tabel pada setiap node

Penjelasan tabel seperti pada Gambar 3.4

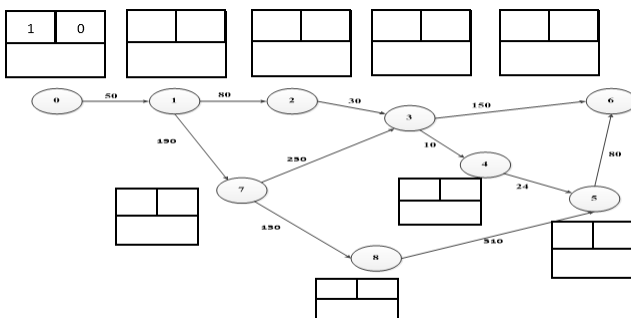


Gambar 3.4. Penjelasan tabel

Setelah itu ada beberapa langkah yang harus dilakukan yaitu :

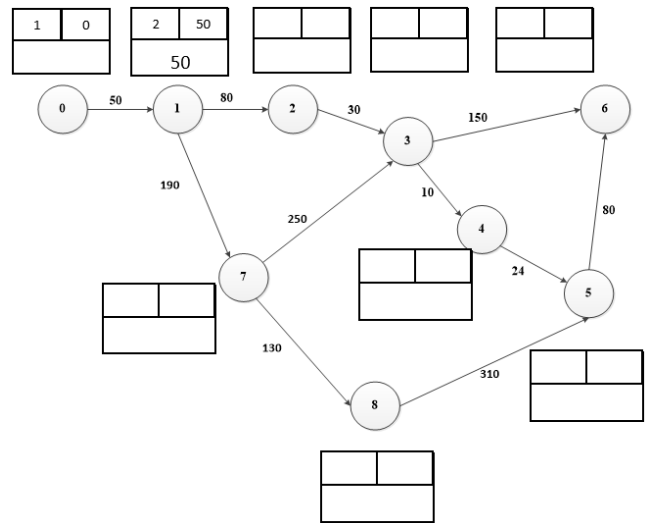
1. Mengisi kotak label pada node awal dengan label urutan 1 dan label jarak 0.
2. Menetapkan label jarak sementara untuk semua node yang dapat dihubungkan langsung dari awal.
3. Pilih node dengan label jarak sementara terkecil dan menuliskan nilainya dilabel jarak, serta tambahkan label urutannya.
4. Masukkan label jarak sementara pada setiap node yang belum memiliki label urutan dan label jarak yang dapat dihubungkan langsung dari node yang baru saja ditulis label jarak dan label urutannya. Nilainya diisi dengan total dari label jarak dari node sebelumnya dan jarak dari node tersebut. Jika label jarak sementara dinode tersebut sudah memiliki nilai, maka harus diganti hanya jika nilai yang baru lebih kecil.
5. Pilih node dengan label jarak sementara terkecil dan menggunakan label semmentaranya sebagai label jarak dari node tersebut, serta tambahan label urutannya.
6. Ulangi langkah 4 dan 5 hingga node tujuan memiliki label jarak dan label urutan.

Maka pada langkah pertama dalam mengisi kotak label pada node awal dengan label urutan 1 dan label jarak 0. Seperti Gambar 3.5.



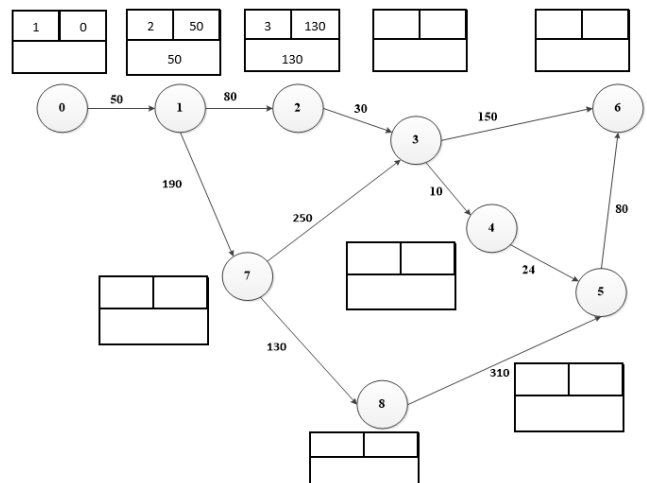
Gambar 3.5. Mengisikan label jarak dan label urutan pada node 0

Kemudian carilah node yang berhubungan langsung dengan node 0 hanya node 1 maka langsung saja isikan label urutan beserta jaraknya seperti Gambar 3.6.



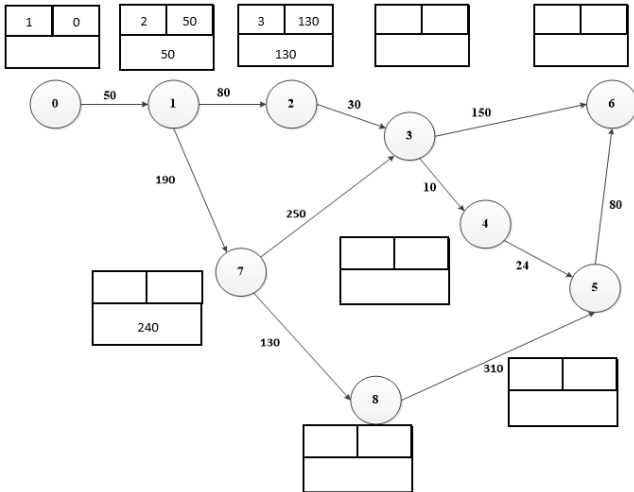
Gambar 3.6. Mengisikan label jarak dan label urutan pada node 1

Selanjutnya cari node yang langsung berhubungan dengan node 0, belum memiliki label jarak dan urutan hanya node 2, jarak node 2 diisi dengan total jarak node 0 sampai ke 2, melalui node 2 jadi $50+80 = 130$ dan isikan label jarak dan urutannya. Seperti pada Gambar 3.7.

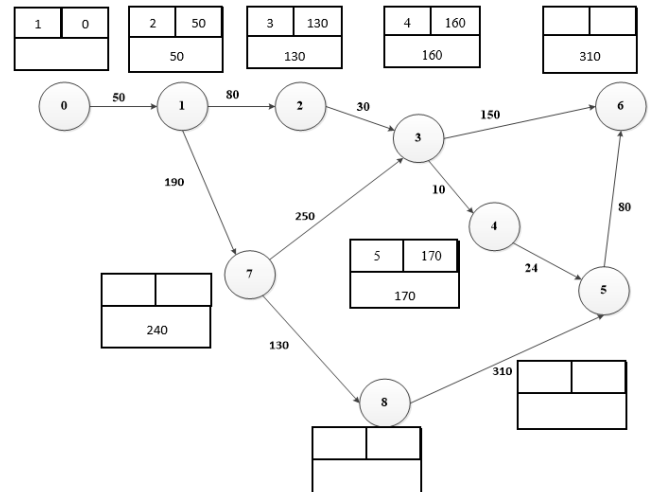


Gambar 3.7. Mengisikan label jarak dan label urutan pada node 2

Node 7 dan 2 yang terhubung langsung dengan node 1, sedangkan node 2 telah di tentukan label jarak dan label urutannya maka node 1, dan node 7 belum memiliki label jarak dan label urutan, jadi node 7, $50+190=240$ maka yang terpilih adalah node 2 karena memiliki jarak yang lebih rendah, kemudian isikan label jarak dan label urutan, seperti pada Gambar 3.8.



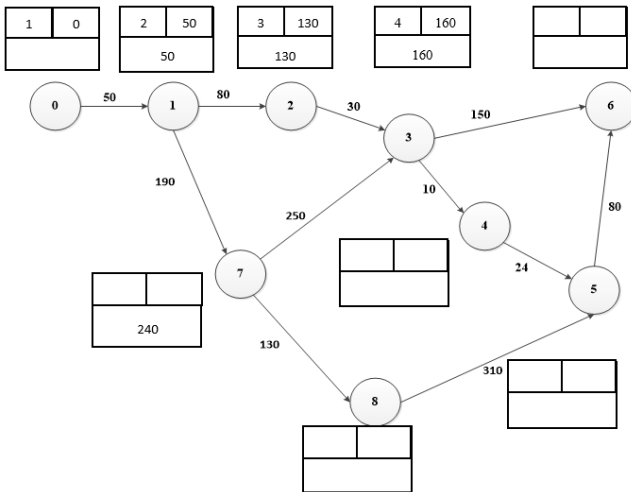
Gambar 3.8. Mengisikan label jarak dan label urutan pada node 7



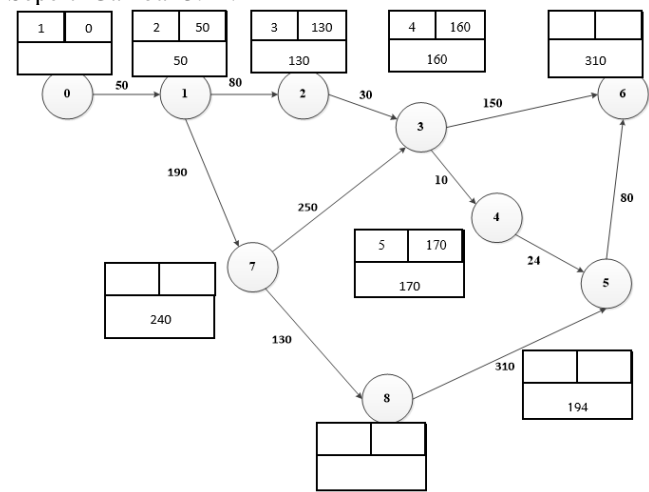
Gambar 3.10. Mengisikan label jarak dan label urutan pada node 4

Selanjutnya node yang terhubung dengan node 2 langsung ialah hanya node 3, jadi $130+30=160$, maka isikan lah label jarak sementara dan label urutannya, seperti Gambar 3.9.

Selanjutnya menghitung label jarak dan label urutan pada node 5 yang terhubung langsung pada node 4 maka node 5, ialah $170+24=194$, isikan label jarak dan label urutannya Seperti Gambar 3.11.



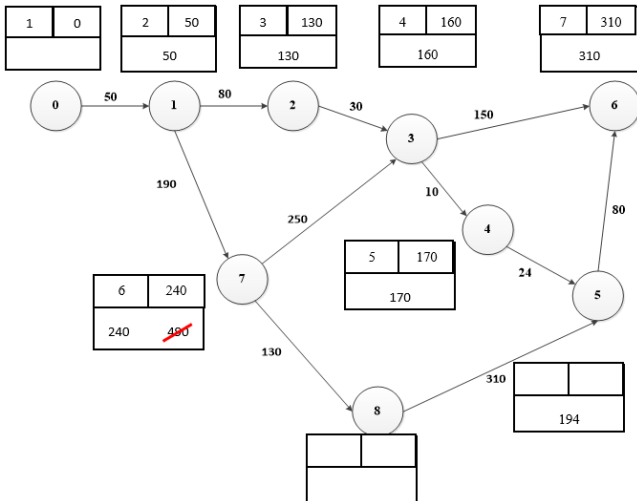
Gambar 3.9. Mengisikan label jarak dan label urutan pada node 3



Gambar 3.11. Mengisikan label jarak dan label urutan pada node 5

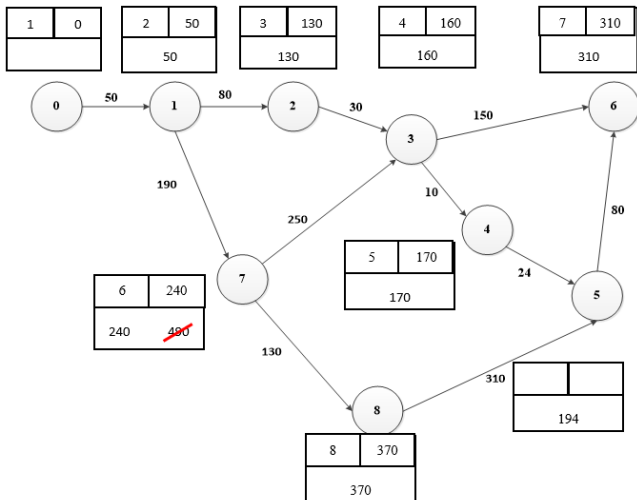
Selanjutnya menghitung label jarak dan label urutan pada node 6 dan node 4 yang terhubung langsung ke node 3 maka node 6, jadi $160+150=310$ hasil dari node 6 telah di temukan sekarang menghitung label jarak dan label urutan node 4, jadi $160+10=170$ dan hasil dari node 4 telah di temukan maka yang terpilih adalah node 4 dengan total hasil lebih kecil di banding dengan node 6, isikan label jarak dan label urutan seperti Gambar 3.10.

Node 7 yang terhubung langsung dengan node 3 memiliki label jarak yang telah di peroleh dari hasil perhitungan sebelumnya dan node 7 belum memiliki label jarak maka node 7, $240+250=490$ hasil lebih besar dari nilai yang sudah ada yaitu 240, maka nilai yang diambil adalah yang kecil yaitu 240. Seperti pada Gambar 3.12.



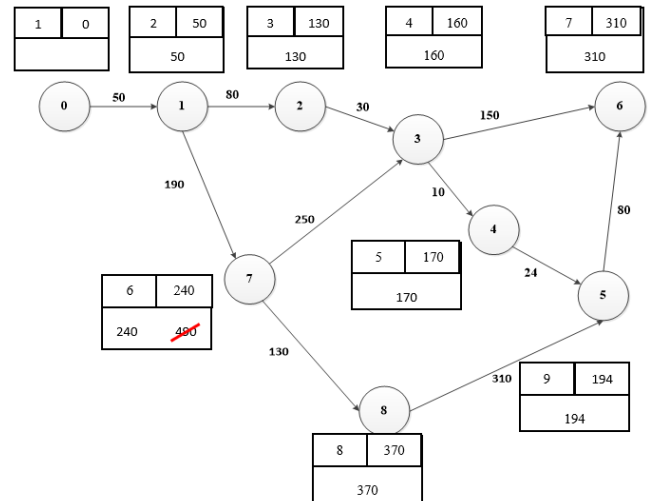
Gambar 3.12. Mengisikan label jarak dan label urutan pada node 7

Node 8 belum memiliki label jarak dan label urutan dimana node 8 terhubung langsung pada node 7 maka, $240+130=370$ isikan label jarak dan label urutannya seperti Gambar 3.13.



Gambar 3.13. Mengisikan label jarak dan label urutan pada node 8

Selanjutnya menentukan label jarak node 5 yang terhubung langsung dengan node 8, maka $370+310=680$ hasilnya lebih besar dari nilai yang sudah ada yaitu 194 yang di peroleh dari node 5 ke node 4, maka nilai 680 diabaikan maka isi label jarak dan label urutannya seperti Gambar 3.14.

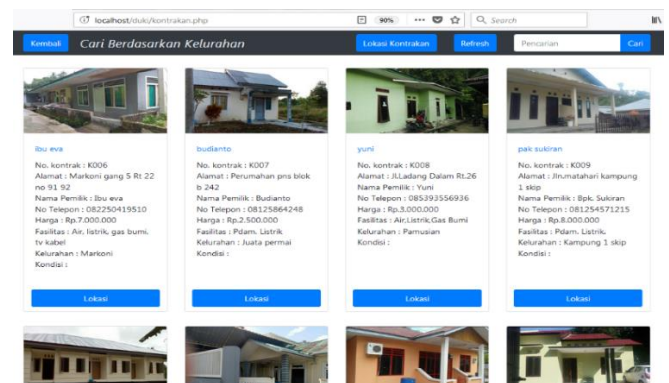


Gambar 3.14. Mengisikan label jarak dan label urutan pada node 5

Jadi rute terpendek yang akan di tempuh dari node awal yaitu 0 menuju tujuan node 4 adalah 0-1-2-3-4 dengan bobot 170.

B. Uji Coba

Uji program dilakukan saat aplikasi dijalankan. Hal ini menunjukkan bagaimana proses Algoritma Dijkstra yang diimplementasikan ke dalam program.

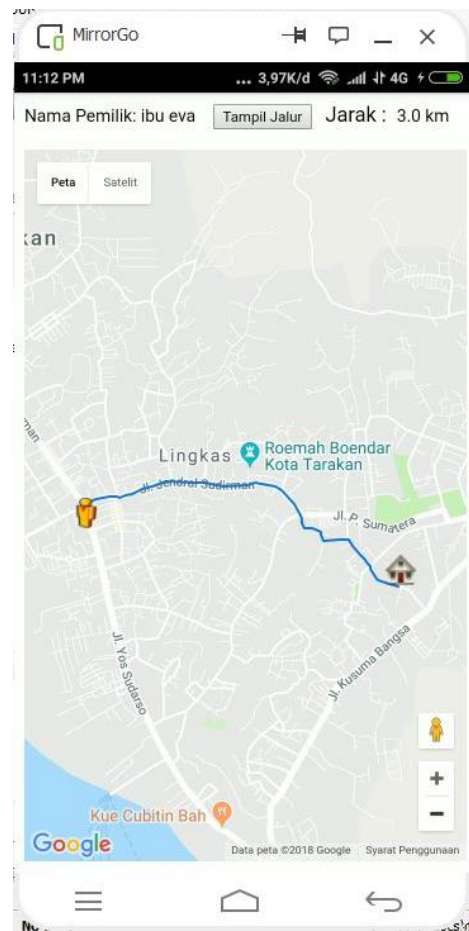


Gambar 3.15. Tampilan data kontrakan pada website

Tampilan diatas berisikan data-data kontrakan yang dapat diakses melalui aplikasi website. Sementara untuk tampilan pada android dapat dilihat pada Gambar 3.16.



Gambar 3.16. Tampilan data kontrakan pada android

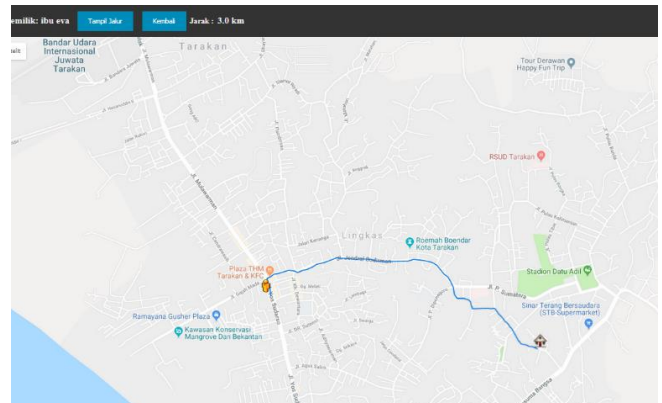


Gambar 3.18. Tampilan rute kontrakan pada android

Sementara untuk tampilan pada website dapat dilihat pada Gambar 3.19.



Gambar 3.17. Tampilan detail kontrakan pada android



Gambar 3.19. Tampilan rute kontrakan pada website

Pada tampilan berikut menampilkan map dan ketika menekan tombol tampil rute maka akan menampilkan rute menuju kontrakan, tampilan seperti pada Gambar 3.18.

Menurut hasil analisa yang didapatkan oleh penulis dalam melakukan penelitian pada, Metode Algoritma Dijkstra ialah bahwa metode yang digunakan masih sangat belum efisien dalam hal menentukan rute terpendek karena metode ini tidak menghitung dari keseluruhan jalur yang ada namun hanya menghitung node yang terdekat dari titik awal dan akan melakukan perhitungan ketika node memiliki percabangan dan akan memilih nilai yang terkecil dari node yang memiliki percabangan. Dan itu menjadi salah satu ketidakefisien dalam menentukan rute terpendek karena pada metode ini node yang kita buat berdasarkan keinginan sendiri sehingga pada penentuan jarak pun hanya berdasarkan rute yang telah dibuat melalui node itu sendiri. Sehingga pada jalur yang

tampil akan menentukan rute yang terdekat menurut node yang telah dibuat.

Jika pun ada nantinya ingin melakukan perhitungan yang lebih efisien lagi terhadap pencarian rute terpendek maka bisa dilakukan penggabungan dua metode antara algoritma djikstra dengan metode lainnya mungkin bisa dengan metode greedy ataupun metode pencarian rute terpendek yang lain

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah penulis lakukan pada studi kasus pencarian kontrakan menggunakan metode Algoritma Dijkstra dalam mencari rute terpendek, dapat disimpulkan sebagai berikut :

1. Metode yang digunakan masih belum cukup efisien dalam melakukan perhitungan jalur terdekat
2. Dalam pembuatan rute masih berdasarkan node yang bernilai kecil dari hasil perhitungan jika ada percabangan.
3. Algoritma Dijkstra mampu memberikan saran rute terpendek menuju kontrakan melalui perhitungan dan memilih angka terkecil dari node yang telah ditentukan.
4. Aplikasi ini juga masih memiliki tampilan yang kurang sempurna pada saat menampilkan gambar kontrakan ketika berada pada layar android yang memiliki ukuran 7.0 inci ke atas.

REFERENSI

- [1] Asyraaf. 2010, Pengertian Desain, Dari (<https://asyraafahmadi.com/2010/06/07/pengertian-desain/>) Diakses tanggal 25 Februari 2108 Pukul 14.30 WITA.
- [2] Dodik Suwarno. 2014, Kelebihan dan kekurangan Aplikasi, Dari (<http://www.plimbi.com/article..145431/berbagai-kelebihan-aplikasi-portable-dan-sedikit>) Diakses tanggal 25 Februari 2018,Pukul 14.57 WITA.
- [3] Nesaba Media.2018. Pengertian Android. Dari (<http://www.nesabamedia.com/pengertian-android-beserta-kelebihan-dan-kekurangannya/>) Diakses tanggal 08 Maret 2018, Pukul 20.36 WITA.

Penulis I, Evi Marliyani, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi STMIK PPKIA Tarakanita Rahmawati Tarakan, lulus tahun 2018.

Penulis II, Moh. Masduki Syahlan, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi STMIK PPKIA Tarakanita Rahmawati Tarakan, lulus tahun 2018.

Penulis III, Obert, memperoleh gelar Magister Manajemen (M.M.), Universitas Hasanuddin. Saat ini menjadi Dosen di STMIK PPKIA Tarakanita Rahmawati

Penulis IV, Roman Gusmana, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi STMIK PPKIA Tarakanita Rahmawati Tarakan, lulus tahun 2013. Saat ini menjadi Tenaga Pendidik di STMIK PPKIA Tarakanita Rahmawati