

PENCARIAN LOKASI FASILITAS UMUM TERDEKAT DILENGKAPI DENGAN RUTE KENDARAAN UMUM LYN

Esther Irawati S.¹, Gunawan², Indra Maryati³, Joan Santoso⁴, Rossy P.C.⁵

^{1,3,4,5}Jurusan Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Surabaya 60284

E-mail : esther@stts.edu, maryati@stts.edu, joan@stts.edu, -

²Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya, Surabaya 60111
E-mail : gunawan@stts.edu

ABSTRAK

Terbatasnya informasi tentang lokasi suatu fasilitas umum menyebabkan kesulitan bagi seseorang saat hendak menggunakan fasilitas umum tersebut. Kesulitan akan semakin terasa ketika kebutuhan untuk menggunakan fasilitas tersebut bersifat mendesak. Seseorang yang terbiasa menggunakan kendaraan pribadi dalam aktifitas sehari-hari akan mengalami kesulitan dalam mencari suatu fasilitas pada saat menggunakan kendaraan umum. Hal ini disebabkan banyaknya jenis kendaraan umum yang ada dewasa ini. Pada penelitian ini akan dibuat sebuah aplikasi yang dapat menyediakan informasi tentang lokasi dari suatu fasilitas umum dan rute kendaraan umum yang dapat digunakan. Aplikasi yang dibuat terdiri dari dua bagian, yaitu aplikasi server dan aplikasi mobile. Aplikasi server digunakan untuk melakukan proses pencarian lokasi fasilitas terdekat. Selain melakukan proses pencarian lokasi terdekat, aplikasi server juga melakukan pencarian rute yang dapat digunakan serta menentukan rute kendaraan umum yang dapat digunakan. Algoritma yang digunakan dalam proses pencarian lokasi terdekat menggunakan dua buah algoritma yang sering digunakan yaitu A. Hasil informasi dari server berupa detail lokasi terdekat, rute menuju lokasi fasilitas yang ditampilkan dalam peta, dan jenis kendaraan umum yang dapat digunakan. Hasil informasi akan dikirimkan kembali menjadi output pada aplikasi mobile.*

Kata kunci : Pencarian Lokasi Terdekat; Kendaraan Umum; Lyn; Lokasi Fasilitas Umum; Algoritma A*

1. PENDAHULUAN

Dewasa ini daya jelajah masyarakat semakin meluas. Dengan semakin meluasnya daya jelajah masyarakat, maka kemampuan untuk mengingat lokasi terdekat dari suatu fasilitas umum akan menurun. Hal ini menjadi salah satu alasan mengapa masyarakat memerlukan suatu informasi tentang lokasi-lokasi fasilitas umum yang sering diperlukan dalam kehidupan sehari-hari.

Alasan lain yang mendasari dibutuhkannya suatu informasi tentang lokasi-lokasi fasilitas umum adalah tuntutan untuk dapat menggunakan waktu secara efektif. Pada era modern saat ini, penggunaan waktu yang tidak efektif dapat memberi dampak negatif dalam menjalankan kehidupan sehari-hari, terutama dalam dunia bisnis. Dengan adanya informasi yang sesuai dengan keperluan, seseorang akan dapat mempersingkat waktu yang dapat digunakan untuk menjalankan aktifitas lain yang tidak kalah penting.

Kendaraan umum lyn atau biasa disebut bemo merupakan salah satu jenis kendaraan yang banyak digunakan oleh masyarakat. Hampir semua kalangan pernah menggunakan kendaraan umum jenis ini. Meskipun kendaraan umum jenis lyn banyak dipakai, sering kali banyak pengguna tidak mengetahui rute-rute yang ditempuh dari masing-masing bemo. Sebagian besar pengguna hanya mengetahui rute lyn yang biasa mereka gunakan.

Dari alasan-alasan tersebut, salah satu solusi yang dapat digunakan adalah penyediaan suatu aplikasi yang dapat memberikan informasi mengenai lokasi fasilitas umum yang tepat kepada pengguna. Lokasi fasilitas umum yang dapat diketahui adalah lokasi rumah sakit, apotek, bank, dan ATM. Selain itu, aplikasi ini juga dilengkapi dengan rute dan jenis lyn yang dapat digunakan untuk menuju lokasi tersebut. Untuk dapat meningkatkan efisiensi dan efektifitas pengguna dalam mendapatkan informasi yang dibutuhkan, maka pembuatan aplikasi ini dilakukan pada *Handphone* / telepon genggam yang telah menjadi suatu bagian yang tidak terpisahkan dalam kehidupan sehari-hari. Adapun target dari pengguna dari aplikasi ini adalah masyarakat umum di Indonesia, khususnya di wilayah Surabaya.

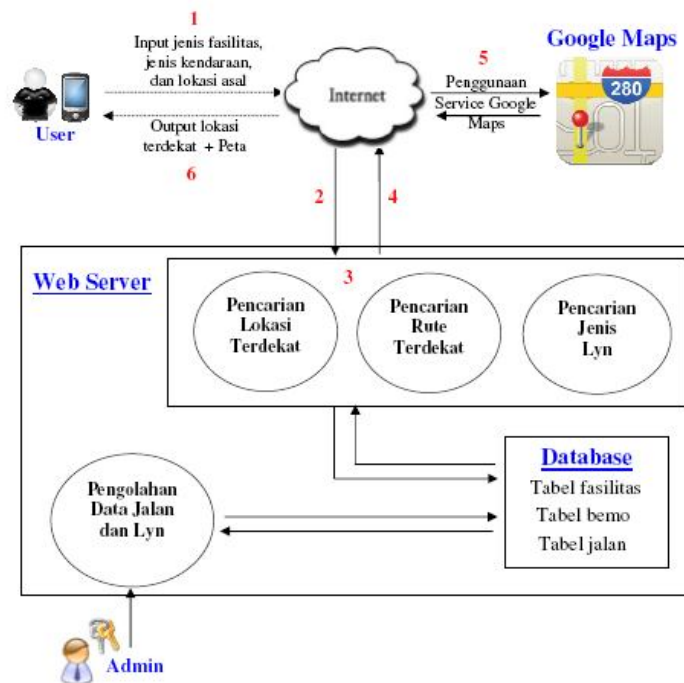
2. INPUT DAN OUTPUT

Pada sisi client, sistem ini memerlukan input dan akan mengembalikan output. Input dilakukan oleh pengguna melalui aplikasi yang telah ada pada Handphone. Data input yang digunakan adalah Fasilitas umum yang ingin dicari, jenis kendaraan yang hendak digunakan, dan lokasi asal user. Lokasi user hanya perlu diinputkan jika tidak terdapat fasilitas GPS pada Handphone user. Jenis kendaraan dibagi menjadi 2 macam, yaitu kendaraan pribadi dan kendaraan umum.

Hasil output yang pertama kali keluar adalah 5 lokasi terdekat. Pada saat user memilih salah satu dari 5 lokasi tersebut, maka akan tampil peta yang menunjukkan jalan dari lokasi user saat menggunakan aplikasi menuju posisi lokasi yang dipilih. Khusus untuk jenis kendaraan umum, akan diberi tambahan output berupa jenis kendaraan umum yang dapat digunakan user untuk menuju lokasi tersebut. Pada sisi server hanya terdapat proses input, yaitu input yang dilakukan oleh administrator tentang lokasi berbagai fasilitas umum seperti rumah sakit, apotek, bank, dan atm, serta rute kendaraan umum.

3. ARSITEKTUR SISTEM

Sistem baru yang dibuat akan menghasilkan suatu informasi tentang lokasi suatu fasilitas umum di kota Surabaya. Hasil informasi yang akan diberikan merupakan letak dari lokasi terdekat. Selain itu, sistem juga memberikan informasi tentang rute dan jenis kendaraan umum lyn yang dapat digunakan untuk menuju lokasi tersebut.



Gambar 1: Arsitektur Sistem

Aplikasi yang dibuat ada dua macam, aplikasi server dan aplikasi mobile. Aplikasi server berupa web server. Aplikasi mobile berupa aplikasi yang digunakan melalui handphone berbasis Android. Input yang diminta sistem adalah lokasi asal pengguna, jenis fasilitas yang dicari, dan rute kendaraan umum. Detail dari proses yang ada pada sistem dijabarkan sebagai berikut :



Gambar 2: Aplikasi

3.1. Proses Penerimaan Permintaan

Tahap pertama yang dilakukan oleh server untuk mendapatkan lokasi suatu fasilitas umum terdekat adalah penerimaan permintaan dari user. Proses ini akan dijalankan pada saat server mendapat request. Proses request dilakukan oleh user dengan menggunakan aplikasi mobile. Pada saat request sampai pada server, server akan melakukan proses pengambilan detail data yang diperlukan untuk proses selanjutnya. Data-data yang diambil adalah posisi asal pengguna baik berupa latitude / longitude atau berupa teks nama jalan, jenis kendaraan yang dipilih, dan juga jenis fasilitas yang dicari oleh user.

Pada saat melakukan request kepada server, inputan yang dikirimkan oleh aplikasi mobile berupa parameter-parameter. Parameter-parameter yang diterima oleh server ini digunakan untuk mendapatkan jenis data yang diperlukan sebagai inputan. Jenis inputan yang diinginkan adalah posisi awal user, jenis fasilitas yang dicari, dan jenis kendaraan umum yang akan dipakai oleh user. Hasil inputan ini akan digunakan untuk menjalankan proses selanjutnya.

3.2. Proses Pencarian Lokasi Terdekat dan Rute

Proses yang terjadi setelah penerimaan request dari user adalah melakukan pencarian lokasi fasilitas umum yang ada. Lokasi fasilitas yang dicari adalah lokasi yang ada dekat dengan posisi user. Pada proses ini server akan menggunakan inputan data posisi awal dan jenis fasilitas yang dicari oleh user. Posisi awal digunakan sebagai parameter untuk menentukan apakah lokasi dari suatu jenis fasilitas yang dicari user ada pada jarak yang dekat atau pada jarak yang jauh. Posisi awal yang akan digunakan dalam proses pencarian berupa koordinat latitude dan longitude.

Jika user menggunakan fasilitas GPS pada handphone, maka posisi awal yang dikirimkan pada server berupa latitude dan longitude. Jika tidak terdapat fasilitas GPS pada handphone user, maka posisi awal yang dikirimkan pada server berupa nama jalan. Nama jalan yang didapat ini akan dirubah menjadi koordinat latitude dan longitude oleh server menggunakan service dari GeoCode

Tahap selanjutnya yang akan dilakukan oleh server setelah memperoleh koordinat posisi user adalah melakukan pencarian letak lokasi dari fasilitas yang dicari. Jenis fasilitas yang dicari oleh user akan digunakan sebagai parameter untuk mendapatkan semua daftar suatu jenis fasilitas yang telah disimpan pada database. Daftar jenis fasilitas yang dicatat sebagai hasil adalah jenis fasilitas yang berada pada radius tertentu dari posisi pengguna. Pada penelitian ini, radius yang digunakan adalah seluas 20 kilometer. Pemberian ruang lingkup dengan radius tertentu dimaksudkan agar proses pencarian lokasi yang terdekat tidak terlalu lama. Semakin kecil ruang lingkup akan berakibat semakin cepat proses pencarian, tetapi kecilnya ruang lingkup juga memiliki resiko untuk tidak menemukan daftar jenis fasilitas yang dicari pada radius tersebut. Ruang lingkup yang besar memiliki kemungkinan menemukan daftar fasilitas yang dicari lebih tinggi, tetapi memiliki resiko proses pencarian menjadi lebih lama. Radius dapat diubah pada web server jika dirasa kurang luas atau terlalu luas.

Daftar fasilitas yang telah ditemukan akan diproses satu persatu untuk mendapatkan urutan lokasi terdekat mulai dari yang paling dekat dengan posisi user hingga yang paling jauh. Proses untuk mendapatkan jarak tempuh dari posisi user hingga posisi masing-masing fasilitas dilakukan dengan algoritma A* sedangkan untuk pengurutannya digunakan algoritma pengurutan Merge Sort.

3.3. Proses Pencarian Jenis Lyn

Proses pencarian jenis lyn hanya akan dijalankan pada saat user memilih jenis kendaraan umum. Proses pencarian jenis lyn baru akan dilakukan setelah proses pencarian lokasi fasilitas umum dan rute selesai dilakukan. Jenis lyn yang dicari berdasarkan pada rute jalan yang telah dihasilkan sebelumnya dengan menggunakan algoritma A*. Pada awal, semua data lyn akan diambil dari database.

Proses selanjutnya adalah mencari daftar lyn yang dapat melewati setiap titik-titik jalan (node) yang telah dihasilkan pada proses pencarian lokasi dan rute. Masing-masing jalan akan mencatat setiap lyn yang lewat. Setelah semua jalan selesai mencatat daftar lyn yang lewat, akan dilakukan proses pemberian suatu nilai pada lyn yang lewat di masing-masing jalan. Pemberian nilai ini dilakukan untuk mendapatkan jenis lyn yang dinilai paling menguntungkan. Dengan pemberian nilai diharapkan user nantinya tidak perlu sering berganti lyn untuk mencapai lokasi tujuan.

Setelah mendapatkan daftar lyn yang lewat pada masing-masing jalan beserta nilai dari tiap lyn, akan dilakukan proses pengurutan lyn. Proses pengurutan ini menggunakan algoritma Merge Sort. Proses pengurutan ini dilakukan pada masing-masing jalan. Setiap lyn yang mempunyai nilai tertinggi akan berada pada posisi teratas. Selesai pengurutan dapat diasumsikan bahwa lyn yang posisinya paling atas merupakan lyn yang paling menguntungkan. Dari posisi awal akan diambil lyn yang berada pada posisi teratas. Jika nilai lyn tersebut lebih dari satu (n), maka jalan selanjutnya akan dilewati dan langsung menuju jalan ke n. Hal ini dilakukan karena jalan-jalan yang dilewati dapat ditempuh dengan menggunakan lyn yang sama. Pada jalan selanjutnya juga akan diambil lyn yang berada pada posisi teratas. Proses ini akan terus diulang hingga mencapai posisi tujuan.

4. ALGORITMA A*

Algoritma A*[1] merupakan salah satu jenis algoritma yang digunakan untuk menyelesaikan kasus yang berhubungan dengan path finding (pencarian jalan). Dalam hasil pencariannya Algoritma A* dikatakan complete dan optimal. Algoritma A* menggunakan cara pencarian secara Best First Search, dimana pencarian dilakukan dengan cara melebar ke setiap node pada level yang sama, dan nantinya akan menemukan rute terbaik dari titik awal sampai titik tujuan. Algoritma A* juga dilengkapi dengan suatu fungsi heuristik. Fungsi heuristik yang terdapat pada algoritma A* digunakan sebagai optimasi dalam menentukan node tujuan yang akan dipilih.

Algoritma A* memiliki lima komponen utama, yaitu : node awal, node goal, open list, closed list, dan cost. Node awal merupakan titik awal dari posisi saat ini, sedangkan node goal merupakan titik akhir atau dapat juga disebut titik tempat tujuan. Cost merupakan nilai dari jarak yang telah ditempuh untuk sampai ke tempat tujuan. Open list digunakan sebagai penampung alternatif node yang tersedia untuk dipilih sebagai node selanjutnya. Open list ini berupa sebuah priority queue, dimana setiap node yang masuk pertama akan dikeluarkan pertama dengan syarat tertentu. Closed list digunakan sebagai penampung node yang telah dilewati selama proses berlangsung. Closed list ini berupa sebuah stack, dimana node yang terakhir dimasukkan akan dikeluarkan pertama kali. Selain sebagai penampung node yang telah dilewati, closed list ini juga digunakan untuk mendapatkan rute terdekat saat node goal sudah dicapai. Isi dari open list dan closed list adalah node-x, parent node-x, dan $F(x)$. Node-x merupakan node tempat berada saat ini. Parent node-x merupakan node asal dari node saat ini. $F(x)$ merupakan nilai cost yang didapat dari perhitungan fungsi heuristik. Cara perhitungan fungsi heuristik adalah dengan menggunakan rumus $F(x) = G(x) + H(x)$. $G(x)$ merupakan cost yang telah didapat dari node awal hingga node yang akan dituju. $H(x)$ merupakan nilai estimasi dari suatu node tempat berada hingga node goal. Nilai $H(x)$ didapat dari jarak antara suatu node hingga node goal tanpa perlu melewati node-node lain yang tersedia. Nilai $F(x)$ digunakan sebagai acuan dalam memilih node yang akan dituju selanjutnya. Node yang memiliki nilai $F(x)$ terkecil yang akan dipilih sebagai node tujuan.

Algoritma A* dimulai dengan mempersiapkan open list dengan node awal sebagai isi dan mengosongkan closed list. Langkah berikutnya adalah mengambil isi dari open list. Jika open list kosong, proses akan berakhir dengan hasil rute tidak ditemukan. Node yang diambil dari open list dimasukkan ke dalam closed list. Jika node tersebut merupakan node goal, proses akan berakhir dengan hasil rute ditemukan. Jika node tersebut bukan merupakan node goal, open list akan diisi dengan node baru yang merupakan successor node tersebut. Successor node adalah node yang dapat dituju dari node saat ini. Successor node yang akan dimasukkan ke dalam open list hanya berupa node yang tidak terdapat dalam closed list.

Pada proses berikutnya, algoritma A* akan mengambil kembali isi dari open list, dan mengulangi langkah yang sama hingga node goal ditemukan atau isi dari open list adalah kosong.

Algoritma 1: Algoritma A*

1. Persiapkan sebuah Priority Queue sebagai OPEN (start, null, 0)
2. Persiapkan sebuah Stack kosong sebagai CLOSED
3. IF OPEN kosong, berhenti dengan kondisi GAGAL
4. ELSE
5. DELETE state pertama (state-x) dari OPEN, kemudian PUSH ke CLOSED
6. IF state-x tujuan adalah GOAL, berhenti dengan kondisi SUKSES, telusuri masing-masing state pada CLOSED
7. ELSE
8. EXPAND state-x untuk membentuk daftar suksesor
9. INSERT suksesor ke OPEN, lakukan SORTING sehingga state dengan cost minimal terletak pada posisi terdepan
10. Kembali ke step 3.

5. UJI COBA

Uji coba dilakukan pada aplikasi server dan aplikasi telepon genggam. Pada aplikasi server digunakan menu testing yang telah dibuat untuk proses uji coba. Menu testing ini akan menghasilkan daftar fasilitas terdekat, peta yang berisi rute yang dapat ditempuh, serta rute kendaraan umum jika user memilih menggunakan kendaraan umum.

Pada uji coba menggunakan aplikasi server, user diminta untuk mengisikan semua data yang diperlukan secara manual. Posisi user yang pada aplikasi mobile dapat diambil secara otomatis, pada aplikasi server harus dimasukkan secara manual karena pada aplikasi server tidak terdapat fasilitas GPS.

Tabel 1: Uji Coba Pertama

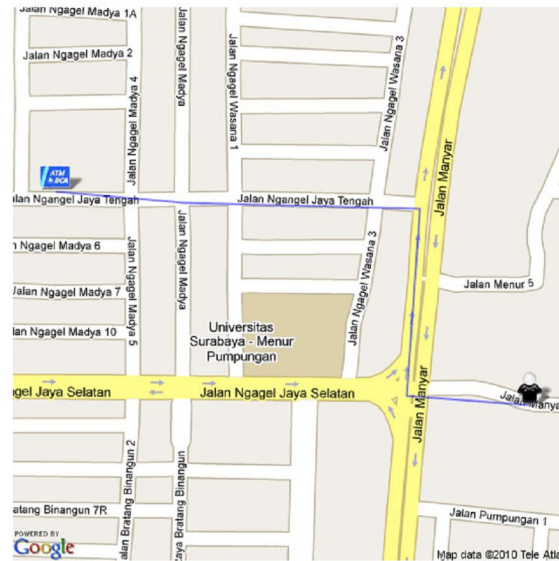
Input	Output
Jenis kendaraan = pribadi	1. ATM BCA RMI – 0.449 km
Lokasi asal = Jalan Manyar Rejo	2. ATM BCASTTS – 0.691 km
Fasilitas yang dicari = ATM BCA	3. ATM BCA Ubaya Ngagel – 0.897 km
Aplikasi yang digunakan = Server	4. ATM BCA Pucang – 2.02 km

Skenario uji coba yang akan dilakukan adalah mencari suatu fasilitas ATM terdekat. Dalam uji coba ini, ATM yang akan dicari adalah ATM BCA dan ATM Mandiri. Uji coba pertama yang dilakukan adalah mencari ATM BCA terdekat. Posisi user saat ini berada pada jalan Manyar Rejo, dan memilih untuk menggunakan kendaraan pribadi. Dan hasil uji coba dapat dilihat pada tabel 1, dan gambar peta yang dihasilkan dapat dilihat pada gambar 3.

Uji coba kedua yang dilakukan adalah mencari ATM Mandiri terdekat. Posisi user saat ini berada pada jalan Menur Pumpungan 5, dan memilih untuk menggunakan kendaraan umum. Hasil Uji Coba. Dan hasil uji coba dapat dilihat pada tabel 2.

Tabel 2: Uji Coba Kedua

Input	Output
Jenis kendaraan = umum	1. ATM Mandiri RMI – 1.263 km
Lokasi asal = Jalan Nginden Semolo	2. ATM Mandiri Ngagel Jaya Selatan – 2.028 km
Fasilitas yang dicari = ATM Mandiri	
Aplikasi yang digunakan = Emulator Android	



Gambar 3: Gambar Peta Hasil Uji Coba Pertama

Pada uji coba kedua ini, selain menampilkan detail informasi, aplikasi juga mengembalikan daftar kendaraan umum yang dapat digunakan untuk mencapai lokasi. Informasi tentang kendaraan umum yang dapat digunakan diberikan karena user memilih jenis kendaraan umum. Hasil dari rute kendaraan umum yang dihasilkan untuk ATM Mandiri RMI adalah sebagai berikut.

Naik Lyn RBK hingga
Perempatan Barata Jaya dan
Bratang Binangun;
Naik Lyn RT hingga Ruko RMI;

Gambar 4: Gambar Peta Hasil Uji Coba Pertama

6. SARAN DAN PENELITIAN SELANJUTNYA

Sistem yang dikembangkan saat ini memiliki beberapa keterbatasan seperti peta yang dihasilkan sebagai output tidak real time, dalam arti merupakan peta dari lokasi user saat menjalankan aplikasi pencarian pertama kali menuju lokasi dari fasilitas yang dicari. Hal ini dikarenakan untuk menghasilkan peta real time membutuhkan koneksi ke server secara terus menerus yang menyebabkan pemakaian pulsa dalam jumlah besar.

Kekurangan sistem yang lain adalah Rute jalan kendaraan pribadi yang dihasilkan dengan menggunakan service dari Google Map tidak selalu benar karena tidak semua peraturan jalan yang ada di Indonesia tercatat pada service Google Map. Kedua kekurangan tersebut sebaiknya pada penelitian selanjutnya dapat dilakukan perbaikan sehingga sistem yang dikembangkan dapat menjadi lebih baik lagi

DAFTAR PUSTAKA

- [1] Stuart Russel dan Peter Norvig, "Artificial Intelligence: A Modern Approach", Upper Saddle River, N.J.: Prentice Hall. 2003.
- [2] Donald Knuth, "Section 5.2.4: Sorting by Merging". *Sorting and Searching*. The Art of Computer Programming. 3 (2nd ed.). Addison-Wesley, 1998, pp.158-168.
- [3] James Steele dan Nelson To, "The Android Developer's Cookbook: Building Applications with the Android SDK: Building Applications with the Android SDK", Addison-Wesley, 2011.
- [4] Robert Richards, "Pro PHP XML and Web Services (Books for Professionals by Professionals)", Apress, 2006.
- [5] Gabriel Svennerberg, "Beginning Google Maps API 3 (Expert's Voice in Web Development)", Apress, 2010.