ISSN 2540 – 7902

SISTEM PENCARI ANJUNGAN TUNAI MANDIRI TERDEKAT DENGAN METODE KARTESIUS DAN RUTE TERPENDEK DENGAN ALGORITMA DIJKSTRA BERBASIS GEOGRAPHIC INFORMATION SYSTEM

Ragil Satra Wicaksana [*,1], Addy Suyatno, M.Kom[2], Indah Fitri Astuti, M.Cs[3]

[1,2,3]Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Mulawarman
Kampus Gunung Kelua Barong Tongkok Samarinda, Kalimantan Timur
E-Mail : : satriaragil46@gmail.com [1], addysuyatno@yahoo.com [2] , indahfitriastuti@yahoo.com [3]

ABSTRAK

Automatic Teller Machine (ATM) atau dalam bahasa Indonesia disebut Anjungan Tunai Mandiri diciptakan untuk mempermudah transaksi perbankan. ATM biasanya diletakkan di tempat-tempat yang strategis dan banyak terjadi transaksi uang seperti pusat perkantoran, pusat hiburan, pusat bisnis dan mall. Luas kota Samarinda yang mencapai 717,4 km² membuat letak ATM tersebar di banyak tempat, sedangkan informasi tentang lokasi ATM di samarinda masih terbatas. Pemanfaatan sistem pencari Lokasi ATM yang berbasis *Geographic Information System* (GIS) digunakan dapat untuk mempermudah dalam pencarian lokasi ATM. Sistem dikembangkan dengan metode Kartesius untuk pencarian lokasi terdekat dan Algoritma Dijkstra untuk pencarian rute terpendek. Input pada sistem ini berupa koordinat lokasi *user* yang kemudian diproses dengan metode Kartesius untuk menentukan tiga rekomendasi ATM terdekat. Rekomendasi yang dipilih kemudian diproses dengan algoritma Dijkstra untuk mencari rute terpendek menuju ke ATM tersebut. Penelitian ini menghasilkan suatu sistem pencarian yang berbasis GIS yang dapat menentukan lokasi ATM terdekat menggunakan metode Kartesius dan rute terpendek menggunakan algoritma Dijkstra.

Kata Kunci : *Automatic Teller Machine*, *Geographic Information System*, Pencarian Lokasi Terdekat, Pencarian rute terpendek, Kartesius, Dijkstra.

1. PENDAHULUAN

Laju pertumbuhan ekonomi semakin pesat mendorong dunia perbankan terus meningkatkan pelayanannya guna mempermudah aktivitas transaksi uang bagi nasabahnya. Salah satu upaya yang dilakukan bank ialah dengan membangun Automatic Teller Machine (ATM) atau dalam bahasa Indonesia disebut Anjungan Tunai Mandiri. ATM diciptakan untuk mempermudah transaksi perbankan. ATM dimaksudkan untuk menggantikan peran dari teller yang sering dijumpai di bank. ATM biasanya diletakkan di tempat-tempat yang strategis dan banyak terjadi transaksi uang seperti pusat perkantoran, pusat hiburan, pusat bisnis dan mall.

Samarinda adalah ibu kota propinsi Kalimantan Timur yang menjadi gerbang keluar dan masuknya barang dan jasa, menjadikan transaksi perbankan di samarinda cukup tinggi, begitu juga dengan transaksi ATM. Luas kota Samarinda yang mencapai 71.800 Ha (www.samarindakota.go.id), membuat letak ATM tersebar di banyak tempat, sedangkan informasi tentang lokasi ATM di Samarinda masih terbatas.

Solusi yang dapat dilakukan adalah membuat suatu sistem yang berbasis Geograpic Information System (GIS) yang dapat menetukan lokasi ATM terdekat serta menemukan jalur tercepat untuk menuju lokasi ATM tersebut. Salah satu cara mencari lokasi ATM terdekat adalah dengan menggunakan metode Kartesius. Kartesius merupakan suatu persamaman matematika untuk mencari jarak antara dua titik dalam bidang datar dan dapat juga digunakan pada sistem koordinat geografis untuk mencari jarak antar dua koordinat.

Pencarian rute terpendek dapat dilakukan dengan memafaatkan salah satu model matematika yaitu Graph. Graph merupakan model matematika yang sangat kompleks dan rumit, tetapi dapat juga menjadi solusi yang sangat baik terhadap beberapa kasus tertentu. Banyak sekali aplikasi menggunakan graph sebagai alat untuk mempresentasikan atau memodelkan persoalan sehingga persoalan itu dapat diselesaikan dengan baik. Aplikasi-aplikasi tersebut misalnya menentukan rute terpendek (the shortest path problem), persoalan pedagang keliling (travelling salesperson problem), persoalan tukang pos China (chinese postman problem), pewarnaan graph (graph colouring), pembuatan sistem jalan raya satu arah (Making a road system one way), menentukan peringkat peserta sebuah turnamen (rangking the participans in a tournament), dan masih banyak lagi (Pradana, 2009).

*Corresponding Authors
Email : satriaragil46@gmail.com

Lintasan dengan bobot minimum disebut sebagai rute terpendek. Bobot disini dapat berupa jarak, waktu tempuh, atau ongkos transportasi dari satu simpul ke simpul yang lain yang membentuk rute tertentu. Solusi untuk persoalan rute terpendek ini sering disebut juga sebagai pathing algorithm. Salah satu shortest path algorithm yang sering digunakan, yaitu algoritma Dijkstra. Algoritma Dijkstra merupakan sebuah graph search algorithm yang menyelesaikan singlesource shortest path problem di mana Dijkstra akan mencari jalur terpendek dari simpul awal dengan cara memeriksa dan membandingkan setiap jalur (Susani, 2012).
Penelitian sebelumnya yang berjudul "Sistem Informasi Geografis ATM (Automatic Teller Machine) dan Mini Market Terdekat Berbasis Android 2.2 (Studi Kasus : Surabaya Timur)" yang dilakukan oleh Abdul Fatah tahun 2014. Penelitian ini menghasilkan aplikasi untuk pencarian lokasi ATM dan Minimarket di Surabaya Timur, sedangkan pada penelitian ini akan dibangun sistem pencari lokasi ATM berbasis GIS dengan menggunakan metode Kartesius untuk mencari lokasi ATM terdekatnya dan algoritma Dijkstra untuk mencari rute terpendek.

2. TINJAUAN PUSTAKA

2.1 Persamaan Kartesius

Sistem koordinat kartesian bidang datar dua dimensi, jarak antara dua titik dapat dicari melalui persamaan:

$$D = \sqrt{(x_2 - x_1) + (y_2 - y_1)} \quad (1)$$

di mana :
D : Jarak linier antara dua titik
xi : Posisi titik i (1,2,…n) pada sumbu x
yi : Posisi titik i (1,2,…n) pada sumbu y

Kesebandingan antara beda antara beda sudut lintang dan bujur dengan beda jarak lurus, maka dapat beda posisi pada koordinat kartesian sebanding dengan beda posisi pada garis lintang dan bujur.

$$D = \frac{D \approx D\emptyset}{\sqrt{(\lambda_2 - \lambda_1)2(\varphi_2 - \varphi_1)2}} \quad (2)$$

di mana :
$D\emptyset$: Jarak sudut antar dua titik
λ_1 : Posisi titik I (1,2,…n) pada garis lintang dalam derajat
φ_1 : Posisi titik I (1,2,…n) pada garis bujur dalam derajat

Tujuan rumus ini adalah untuk mencari suatu titik pencarian, bukan menghitung berapa jarak diantaranya. Sehingga yang dibandingkan adalah nilai jarak sudut antara titik koordinat yang ditentukan dengan titik-titik koordinat yang dibuat dalam *database* (Bernadus, 2012).

2.2 Algoritma Dijkstra

Algoritma dijkstra merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Prinsip dari algoritma Dijkstra adalah dengan pencarian dua lintasan yang paling kecil. Algoritma Dijkstra memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Jarak titik yang diketahui (dari titik awal) disetiap iterasi diperbarui bila ternyata didapat titik yang baru yang memberikan jarak terpendek. Syarat algoritma ini adalah bobot sisinya yang harus nonnegatif, (Satyananda, 2012).

Menurut Alfred V. Aho (1974) dalam Satyananda (2012) menjelaskan rincian algoritma Dijkstra sebagai prosedur:

Input: graf terhubung dan berarah G=(V,E), matriks bobot C, titik awal v0.
Output: jarak terpendek dari titik V0 ke titik lain dalam D.
Prosedur Dijkstra (G,C, v0)
Mulai
S ← { v0 } ;
D[v0] ← 0 ;
Untuk masing-masing v dalam V- { v0 } lakukan
D[v] ← C[v0,v] ;
Selama S ≠ V lakukan
Mulai
Pilih salah satu titik w di v – s dimana D[w] adalah minimum;
S ← S _ {w} ;
Untuk masing-masing v di V – S lakukan
D[v] ← min (D[w], D[w] + C[w , v]) ;
Selesai
Selesai

Rincian prosedur tersebut *v0* merupakan titik awal yang ditentukan. D[v] adalah jarak terpendek dari *v0* ke titik v. C adalah matrik bobot, dan C[w , v] adalah jarak (bobot) dari titik w ke titik v. Min adalah fungsi untuk mencari nilai terkecil dari dua nilai, himpunan S digunakan untuk mencatat titik-titik yang terpilih pada setiap iterasi dan himpunan V berisi semua titik dalam *graph*. Menurut Aldous dan Wilson (2000), *graph* merupakan diagram yang memuat titik yang disebut *vertex* dan dihubungkan oleh garis yang disebut sisi, dengan masing-masing sisi tepat menghubungkan dua titik.

Menurut Fauzi (2011), terdapat tiga elemen utama yang menggambarkan kondisi status dari setiap simpul yang sedang ditelusuri, yaitu :
1. Kondisi node yang belum ditemukan dan belum dikunjungi.
2. Kondisi node yang sudah ditemukan tetapi belum dikunjungi
3. Kondisi node yang telah ditemukan dan sudah dikunjungi.

Node yang dikunjungi merupakan node yang terpendek dari setiap tahap algoritma Dijkstra. Jadi jalur atau rute yang dibentuk oleh algoritma

Dijkstra tersusun dari node yang telah ditemukan dan telah dikunjungi. Langkah-langkah dari algoritma Dijkstra yaitu:

1. Langkah Pertama yaitu menetapkan node awal sebagai status ditemukan (*found*) dan kemudian dikunjungi atau ditangani (*handle*)
2. Langkah kedua yaitu dilakukan pencarian terhadap setiap node yang dapat dicapai secara langsung dari node yang sedang dikunjungi.
3. Langkah ketiga yaitu:
   a. Apabila node yang didapatkan pada langkah kedua belum pernah ditemukan maka rubah statusnya menjadi ditemukan.
   b. Apabila node yang didapatkan sudah pernah ditemukan maka lakukan update pada bobotnya, ambil bobot yang lebih kecil.
4. Langkah keempat yaitu dilakukan pencarian terhadap node yang memiliki bobot paling kecil dari semua node yang berada pada status ditemukan kemudian mengunjunginya.
5. Lakukan *looping* secara berurutan pada langkah kedua ketiga dan keempat sampai semua node ditemukan.

2.3 *Geographic Information System* (GIS)

Menurut Aronoff (Prahasta, 2009), GIS adalah sistem yang berbasiskan komputer (CBIS) yang digunakan untuk menyimpan dan memanipulasi informasi-informasi Geografis. GIS dirancang untuk mengumpulkan, menyimpan dan menganalisis objek-objek dan fenomena dimana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis. Dengan demikian, GIS merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografis: (a) masukan, (b) manajemen data (penyimpanandata dan pemanggilan data), analisis manipulasi data, dan (d) keluaran.

Sedangkan menurut Star Jeffrey & Esten John (Prahasta, 2009), GIS adalah sistem informasi yang dirancang untuk bekerja dengan data yang tereferensi secara spasial atau kordinat geografis. Atau dengan kata lain, GIS adalah sistem basis data dengan kemampuan-kemampuan khusus menangani data yang tereferensi secara spasial; selain merupakan sekumpulan operasi-operasi yang dikenakan terhadap data tersebut.

Menurut Raper J., Green H. (Prahasta, 2009), GIS salah satu sistem yang kompleks dan pada umumnya juga (selain yang *stand-alone*) terintegrasi dengan lingkungan sistem komputer lainnya di tingkat fungsional dan jaringan (*network*). Jika diuraikan, GIS sebagai sistem terdiri dari beberapa komponen dengan berbagai karakteristiknya.

1. Perangkat Keras/*Hardware*

Saat ini GIS sudah tersedia bagi berbagai *platform* perangkat keras; mulai dari kelas PC *dekstop, workstations*, hingga *multi-user host* yang bahkan dapat digunakan oleh banyak orang secara bersamaan (simultan) dalam jaringan komputer yang luas, tersebar, berkemampuan tinggi, memiliki penyimpanan (*harddisk*) yang besar, dan mempunyai kapasitas memori (RAM) yang besar. Walaupun demikian, fungsionalitas GIS tidak terkat secara ketat pada karakteristik-karakteristik fisik perangkat kerasnya sehingga keterbasatasan memori pada suatu PC-pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk aplikasi GIS adalah komputer (PC), *mouse*, *monitor* (plus VGA-*card* grafik) yang beresolusi tinggi, *digitizer*, *printer*, *plotter*, *receiver* GPS, dan *scanner*.

2. Perangkat Lunak/*Software*

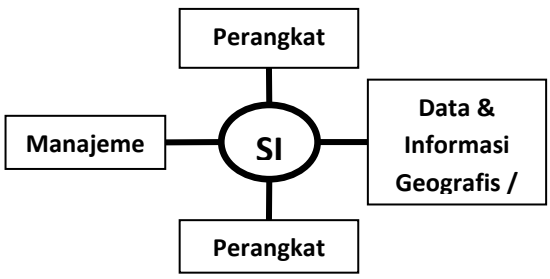
GIS bisa juga merupakan sistem perangkat lunak yang tersusun secara modular dimana sistem basis datanya memegang peranan kunci. Pada kasus perangkat GIS tertentu, setiap subsistem diimplementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul hingga tidak mengherankan ada beberapa perangkat GIS yang terdiri dari ratusan modul program (*.exe) yang masing-masing dapat dieksekusi tersendiri.

3. Data & Informasi Geografi

GIS dapat mengumpulkan dan menyimpan data atau informasi yag diperlukan baik secara tidak langsung (dengan cara meng-*import*-nya dari format-format perangkat GIS yang lain) maupun secara langsung dengan cara melakukan dijitasi data spasialnya (dijitasi *on-screen* atau *head-ups* diatas tampilan layar monitor dengan menggunakan *digitizer*) dari peta analog dan kemudian memasukan data atributnya dari tabel atau laporan dengan menggunakan *keyboard.*

4. *Management*

Suatu proyek GIS akan berhasi jika dikelola dengan baik dan dikerjakan oleh orang-orang yang memiliki keahlian (kesesuaian dengan *job-description* yang bersangkutan) yang tepat pada semua tingkatan.



Gambar 1. Komponen-Komponen GIS
(Sumber : Raper J., Green H., 1994. Yang ditulis dalam buku Prahasta, 2009)

3. HASIL DAN PEMBAHASAN

Sistem pada penelitian ini memiliki lima tahapan proses. Tahapan pertama, sistem akan mendeteksi lokasi *user* menggunakan *library geolocation* Google Maps API, jika browser tidak mendukung untuk mengakses *library geolocation* Google Maps API atau posisi *user* tidak terbaca oleh sistem maka sistem akan meminta *user* untuk menginputkan secara manual posisi *user* sehingga didapatkan koordinat *user*. Tahap kedua, sistem akan menghitung jarak linier antara koordinat ATM dengan koordinat *user* dan menggunakan persamaan Kartesius kemudian mencari nilai terkecil. *Sample* koordinat ATM yang ada pada sistem dapat dilihat pada tabel 1.

Tabel 1. Koordinat ATM

Nama ATM	Latitude	Longitude
ATM BNI SPBU PM Noor	-0.4525590568368827	117.16307669878006
ATM BNI Kantor Cabang Unmul	-0.467612469546452	117.15587764978409
ATM BNI Pramuka	-0.46429602777196644	117.15439438819885
ATM BNI AS MART M. Yamin	-0.463094437484553755	117.15021014213562
ATM BNI Lulu Mart M. Yamin	-0.4654547039271663	117.14896559715271
ATM BNI Samarinda Square	-0.4701323205393252	117.1472704410553
ATM BNI Drive Thru	-0.47444516610776566	117.1448028087616
ATM BNI Mall Lembusuwana	-0.47509960263882267	117.14722752571106
ATM BNI Auto Swalayan Wahid Hasyim	-0.4607516040509344	117.15019404888153
ATM BNI Wahid Hasyim	-0.45741370208226517	117.1514493227005
ATM BNI Kantor Cabang AW Syahrani	-0.4510570664879729	117.14733481407166
ATM BNI Swalayan AW Syahrani	-0.4650255646325943	117.13930963446991
ATM BNI KK. Ahmad Yani	-0.4788840684095304	117.15522050857544
ATM BNI Ahmad Yani 1	-0.4784200624419675	117.15608954429626
ATM BNI Ahmad Yani 2	-0.4768510246893224	117.15846866369247
ATM BNI Ahmad Yani 3	-0.47629314451325416	117.16008335351944
ATM BNI Ahmad Yani 4	-0.4697595060270742	117.17059761285782
ATM BNI P.M Noor	-0.4611928131538183	117.17439562208229

(Data diambil pada Bulan Maret 2016)

Data pada Tabel 1 didapatkan dengan observasi langsung dilapangan dan mencatat koordinat lokasi dari ATM tersebut. Tahap ketiga, sistem akan menambahkan posisi *user* dan posisi ATM kedalam *graph* sebagai simpul *temporary*.

Tahap keempat sistem melakukan pencarian rute terpendek antara simpul *user* dan simpul ATM dengan menjalankan langkah-langkah yang terdapat pada Algoritma Dijkstra. *Graph* yang dipakai pada sistem dapat dilihat pada tabel 2.

Tabel 4.2 *Graph* Sistem

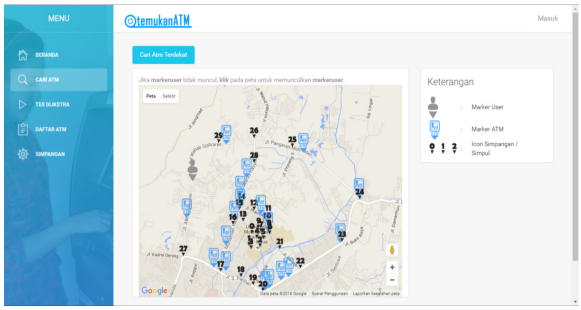
Simpul	Simpul Relasi	Bobot
0	4	159.30647999102
	1	265.41944926755
1	3	80.481252901204
	0	265.41944926755
	16	700.38545096822
	18	751.99673294051
2	3	451.30960313009
	4	201.83998066555
	5	189.75022355654
3	2	451.30960313009
	1	80.481252901204
	4	323.50600713514
4	3	323.50600713514
	0	159.30647999102
	2	201.83998066555
5	6	179.31365479686
	7	159.99775235798
	2	189.75022355654
	7	158.58538543023
6	8	76.722297014647
	21	406.18373502619
	5	179.31365479686
7	5	159.99775235798
	5	158.58538543023
	9	97.222234569346

	8	186.67824022117
8	10	156.27045440643
	6	76.722297014647
	7	186.67824022117
9	10	265.34246744395
	7	97.222234569346
10	11	158.8090245705
	8	156.27045440643
	9	265.34246744395
11	12	341.72817491975
	25	1646.3476780189
	10	158.8090245705
12	13	459.53771242556
	14	314.25418275051
	11	341.72817491975
13	16	359.94407193388
	15	363.23109295073
	12	459.53771242556
14	28	973.08116434546
	12	314.25418275051
	15	137.95620550872
15	14	137.95620550872
	16	344.45725830264
	13	363.23109295073
16	15	344.45725830264
	13	359.94407193388
	17	1045.4194288223
	1	700.38545096822
17	27	936.67533094303
	18	489.59021697164
	16	1045.4194288223
18	17	489.59021697164
	19	322.75619139214
	1	751.99673294051
19	18	322.75619139214
	20	260.80393308583
	21	979.11909082764
20	19	260.80393308583
	22	979.98286890896
21	6	406.18373502619
	22	623.18516754992
	19	979.11909082764
22	23	1103.9801044087
	20	979.98286890896
	21	623.18516754992
23	24	1041.5591980805
	22	1103.9801044087
24	25	1935.2169976415
	23	1041.5591980805
25	26	888.37574788306
	11	1646.3476780189
	24	1935.2169976415
26	28	527.85684074511
	25	888.37574788306
	29	823.78192483664
27	29	2932.9555198997
	17	936.67533094303
28	26	527.85684074511
	14	973.08116434546
	29	1587.0217636915
29	28	1587.0217636915
	26	823.78192483664
	27	2932.9555198997

(Data Diambil Pada Bulan Maret 2016)

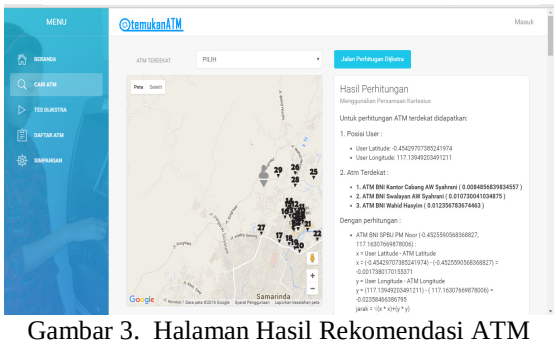
Tahap kelima, proses visualisasi hasil perhitungan Algoritma Dijkstra pada peta Google Maps API dengan didasarkan pada hasil perhitungan algoritma Dijkstra pada *graph.*

Implemetasi halaman cari ATM dapat dilihat pada gambar 2. Halaman Cari ATM dapat diakses melalui dua cara, pertama menggunakan tombol "Mulai Pencarian" yang terdapat pada halaman beranda dan melalui bagian navigasi halaman. Visualisasi peta pada halaman ini menampilkan keseluruhan marker ATM, marker simpangan dan marker *user* yang muncul setelah sistem mendeteksi posisi *user* dengan fungsi *geolocation* atau ketika *user* mengeklik pada peta untuk menentukan posisi *user* secara manual. Tombol "Cari ATM terdekat" hanya akan muncul saat marker *user* telah muncul, tombol ini berguna untuk memulai proses pencarian ATM terdekat.



Gambar 2 Halaman Cari ATM

Tombol "Cari ATM Terdekat" berfungsi untuk melakukan proses perhitungan untuk mencari posisi ATM yang mempunyai nilai terkecil dengan posisi *user*. Hasil perhitugan akan menentuakan tiga nilai terkecil dan akan ditampikan dengan data pada konten dan visualisasi peta. Peta dilengkapi dengan marker ATM untuk atm yang dipilih, marker *user* serta marker untuk posisi simpangan. Konten hasil perhitungan yang terdapat disebelah peta menunjukan detail perhitungan jarak ATM terdekat dengan *user*. Gambar hasil rekomendasi atm terdekat dapat dilihat pada gambar 3.

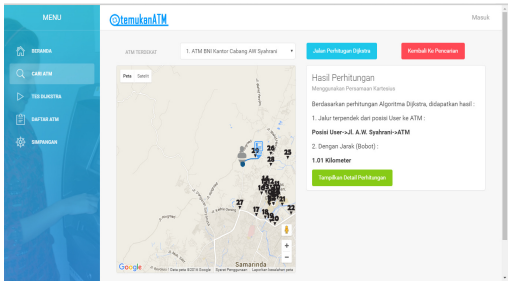


Gambar 3. Halaman Hasil Rekomendasi ATM terdekat

Terdapat *combo-box* untuk memilih tiga rekomendasi ATM terdekat pada halaman. Tombol "Jalankan Perhitungan Dijkstra" yang terdapat pada halaman hasil rekomendasi ATM berfungsi untuk memulai proses peritungan rute terdekat dari posisi user dengan posisi ATM. *User* harus memilih ATM tujuan pada *combo-box* untuk memulai perhitungan rute terpendek, jika *user* tidak melakukan tahapan tersebut maka sistem akan otomatis mengeluarkan notifikasi yang berisi perintah untuk memilih ATM tujuan. Bentuk notifikasi dapat dilihat pada gambar 4.

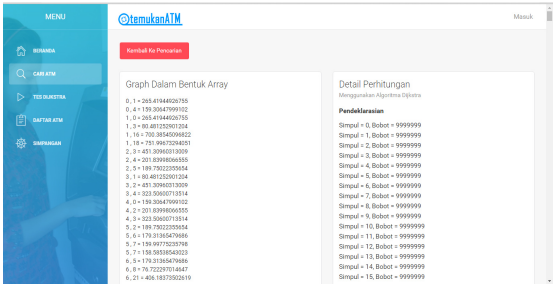
Gambar 4. Notifikasi Pada Halaman Cari ATM

Hasil pencarian rute terpendek ditampilkan dengan visualisasi peta dan konten yang berisi hasil perhitungan. Halaman hasil perhitungan dijkstra dapat dilihat pada gambar 5.



Gambar 5. Halaman Hasil Perhitungan Dijkstra

Terdapat dua tombol pada halaman hasil perhitungan Dijkastra yaitu tombol "Kembali Ke Pencarian" berguna untuk kembali ke halaman cari ATM dan tombol "Tampilkan Detail Perhitungan" yang berguna untuk melihat detail perhitungan Dijkastra. Tampilan halaman detail perhitungan Dijkstra dapat dilihat pada gambar 6.

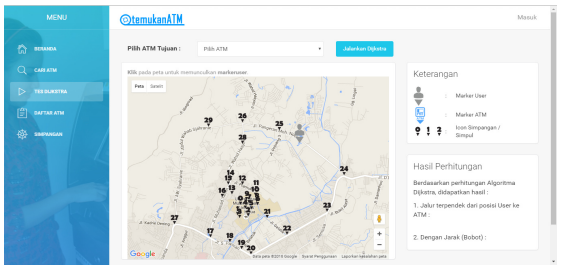


Gambar 6. Halaman Detail Perhitungan Dijkstra

Terdapat tombol "Kembali Ke Pencarian" pada halaman detail perhitungan Dijkstra yang berfungsi untuk kembali ke halaman depan cari ATM.

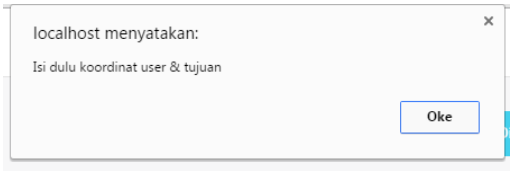
1. Halaman Tes Dijkstra

Halaman tes dijkstra merupakan halaman untuk melakukan tes pada perhitungan Dijkstra. Gambar halaman tes Dijkstra dapat dilihat pada gambar 7.



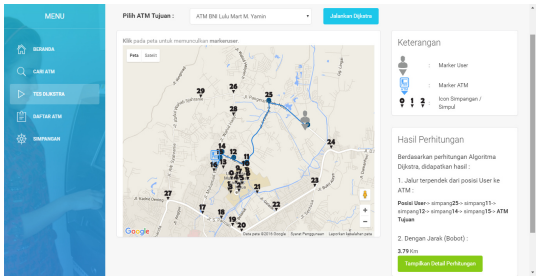
Gambar 7. Halaman Tes Dijkstra

Visualisasi peta pada halaman ini menampilkan keseluruhan marker ATM dan marker simpangan. Tombol "Jalankan Dijkstra" yang terdapat pada halaman tes dijkstra berfungsi untuk memulai proses peritungan rute terdekat dari posisi user dengan posisi ATM. *User* terlebih dahulu harus mengeklik pada peta untuk menentukan posisi *user* dan memilih ATM tujuan pada *combo-box* untuk memulai proses pencarian. Jika *user* tidak melakukan salah satu atau kedua tahapan tersebut maka sistem akan otomatis mengeluarkan notifikasi yang berisi perintah untuk mengisi keduanya. Bentuk notifikasi dapat dilihat pada gambar 8.



Gambar 8. Notifikasi Pada Halaman Tes Dijkstra

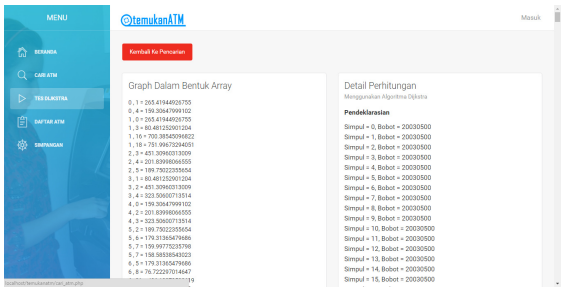
Hasil perhitungan pencarian rute terpendek ditampilkan dalam bentuk visualisasi pada peta dan konten yang berisi hasil perhitungan. Halaman hasil perhitungan dijkstra dapat dilihat pada gambar 9.



Gambar 9. Halaman Hasil Perhitungan Dijkstra

Tombol "Tampilkan Detail Perhitungan" pada konten hasil perhitungan berguna untuk melihat detail perhitungan Dijkastra. Tampilan halaman detail perhitungan Dijkstra dapat dilihat pada gambar 10.

ISSN 2540 – 7902



Gambar 10. Halaman Detail Perhitungan Dijkstra

Tombol "Kembali Ke Pencarian" pada halaman detail perhitungan Dijkstra berfungsi untuk kembali ke halaman depan cari ATM.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan analisis dan pengujian yang dilakukan pada bab sebelumnya, maka kesimpulan yang dapat diambil adalah proses perhitungan lokasi terdekat menggunakan persamaan Kartesius untuk menentukan posisi ATM pada sistem dapat diterapkan dan mampu menampilkan tiga rekomendasi ATM terdekat dan proses perhitungan rute terpendek dari simpul awal yaitu posisi user dan simpul tujuan yaitu posisi ATM dengan menggunakan algoritma Dijkstra dapat diterapkan pada sistem. Aplikasi ini mampu menunjukan rekomendasi tiga ATM terdekat dari posisi user.

4.2 Saran

Penulis menyarankan pengembangan penelitian lebih lanjut sistem sistem Pencarian ATM terdekat yaitu penambahan data graph pada sistem untuk jalan dan simpangan yang lebih luas agar cakupan peta yang dapat dijangkau pada sistem lebih maksimal, pencarian lokasi terdekat dapat dilakukan dengan metode lain agar dapat memperhitungkan jarak antara posisi user dan posisi ATM dengan berdasarkan pada jalan, bukan hanya dengan garis lurus dan penambahan variabel bobot pada graph seperti kemacetan, waktu tempuh, kepadatan jalan dan lain-lain.

5. DAFTAR PUSTAKA

[1] Bernadus, 2012. *Buat Sendiri Petamu Menggunakan Codeigniter dan Google Maps API*. Yogyakarta : Andi.

[2] Fauzi, I. 2011. *Penggunaan Algoritma Dijkstra Dalam Pencarian Rute Tercepat dan Rute Terpendek (Studi Kasus Pada Jalan Raya antara Wilayah Blok M dan Kota).* Skripsi Fakultas Sains dan Teknologi. Program Studi Teknik Informatika. UIN Syarif Hidayatullah. Jakarta.

[3] M. Aldous, Joan dan J. Wilson, Robin. 2000. *Graphs and Application an Introductory Approach*. Britania Raya: Universitas Terbuka.

[4] Pradana, B.A.. 2009. *Studi Implementasi Persoalan Lintasan Terpendek Suatu Graf dengan Algoritma Dijkstra dan Algoritma Bellman-Ford.*

[5] Prahasta, E. 2009. *Sistem Informasi Georgafis Konsep Konsep Dasar (Perspektif Geodesi & Geomatika).* Bandung: Informatika Bandung.

[6] Satyananda, D. 2012. *Struktur Data*. Modul tidak diterbitkan. Malang: Universitas Negeri Malang.

[7] Susani, I.M. 2012. *Perbandingan Algoritma Dijkstra, Bellman-Ford, dan Floyd-Warsall Untuk Mencari Rute Terpendek (The Shortest Path Problem).* Skripsi. Fakultas Sains dan Teknologi. Program Studi Matematika. Universitas Islam Negeri Sunan Kalijaga Yogyakarta.