

Sistem Informasi Pencarian Service Ac Mobil Menggunakan Algoritma Dijkstra

Fandi Saputra^a, Yulvia Nora Marlim^b

^aSekolah Tinggi Ilmu Komputer Pelita Indonesia, Jl. Ahmad Yani No, 78-88, fandri.saputra@student.pelitaindonesia.ac.id

^bSekolah Tinggi Ilmu Komputer Pelita Indonesia, Jl. Ahmad Yani No, 78-88, yulvianoramarlim@lecturer.pelitaindonesia.ac.id

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 19 Oktober 19

Revisi Akhir: 28 November 19

Diterbitkan Online: 30 Desember 19

KATA KUNCI

Algoritma Dijkstra, Jasa Service Ac Mobil, Rute Terpendek, Google Maps API, Sistem Informasi Pencarian

KORESPONDENSI

E-mail:

yulvianoramarlim@lecturer.pelitaindonesia.ac.id

ABSTRACT

Di kota pekanbaru belakangan ini banyak sekali keluhan pada beberapa orang mengenai pencarian jasa service ac mobil, keluhan yang sering didapati oleh pelanggan adalah susah menemukan lokasi service jasa ac mobil sebab tempatnya sulit dijangkau yang terletak di jalan gang, dan kebanyakan tidak adanya papan nama yang menandai adanya tempat jasa service ac mobil tersebut. Akibatnya masalah yang sering timbul seperti berhenti dipinggir jalan untuk bertanya dimana lokasi jasa service ac mobil kepada penduduk/masyarakat sekitar sehingga calon konsumen memerlukan biaya tambahan dan waktu yang terbuang karena belum adanya sebuah sistem yang dapat mendukung proses pencarian rute terpendek untuk menemukan lokasi jasa service ac mobil. Maka dibutuhkanlah sebuah sistem informasi pencarian service jasa ac mobil dengan rute terpendek. Rute terpendek dicari dengan menggunakan sebuah algoritma dijkstra yang diterapkan didalam sebuah website. Metode dijkstra digunakan untuk membantu menghitung rute terpendek untuk menuju tujuan. Pembobotan untuk masing-masing node diambil berdasarkan jarak tempuh antar bengkel. Pada penelitian objek yang digunakan adalah tempat jasa service ac mobil di pekanbaru. Untuk node awal(sumber) tidak ditentukan sedangkan node tujuan(akhir) ditentukan titik koordinatnya(longitude dan latitude). Sistem informasi ini dibangun menggunakan web dan memanfaatkan *google maps api* sebagai petanya. Tujuan penelitian ini untuk mempermudah calon konsumen mencari alamat jasa service ac mobil. Diharapkan dengan adanya sistem ini dapat membantu calon konsumen untuk lebih mudah menemukan lokasi dan dapat menghemat waktu dan biaya.

1. PENDAHULUAN

Perkembangan teknologi pada saat ini tumbuh dengan sangat pesat. Terutama perkembangan teknologi di bidang pemrograman website, pemrograman website didesain untuk dapat mempermudah menyelesaikan pekerjaan manusia dalam berbagai bidang. Pembangunan sistem menggunakan website juga diharapkan dapat membantu manusia dalam mengolah data dengan cepat namun tepat dan dapat diakses dimana saja sehingga hasilnya dapat digunakan manusia untuk mengambil keputusan yang tepat berdasarkan sistem yang dibuat.

Sistem informasi merupakan suatu kumpulan data yang terorganisasi beserta tata cara penggunaannya yang mencakup lebih jauh dari pada sekedar penyajian. Istilah tersebut menyiratkan suatu maksud yang ingin dicapai dengan jalan memilih dan mengatur data serta menyusun tata cara penggunaannya. Salah satu contoh sistem informasi digunakan untuk membantu mencari rute terpendek dalam mencari jasa service ac mobil [1].

Rute terpendek yaitu cara yang digunakan untuk menemukan lintasan terpendek antara dua atau beberapa simpul yang berhubungan, persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi persoalan ini biasanya direpresentasikan dalam bentuk graf, graf yang digunakan dalam pencarian lintasan terpendek atau *shortest path* adalah graf berbobot untuk dapat menggunakan graf ini biasanya dilakukan sebuah perhitungan menggunakan algoritma dijkstra [2].

Algoritma *dijkstra* dikenal sukses dalam membantu dalam pencarian rute terpendek dalam mencari suatu tempat, dan tidak bisa dipungkiri dijkstra masih menjadi salah satu yang populer dari sekian banyak algoritma yang ada [3]. Algoritma ini ditemukan oleh *edsgar W. Dijkstra* dan di publikasikan pada tahun 1959 pada sebuah jurnal *Numerische Mathematik* yang berjudul "A Note on Two Problems in Connexion with Graphs". [4] Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian

yang diberikan adalah *infiniti* (tak terhingga). Pada algoritma dijkstra, node digunakan karena algoritma dijkstra menggunakan graf berarah untuk penentuan rute lintasan terpendek [5].

Salah satu contoh *platform* yang digunakan untuk mencari rute terpendek adalah menggunakan *Google Maps API*. *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript*. Cara membuat *Google Maps* untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi Internet yang sangat stabil. Dengan menggunakan *Google Maps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik *Google* sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi [6].

2. TINJAUAN PUSTAKA

2.1. Konsep Dasar Sistem Informasi

2.1.1. Pengertian Sistem

Menurut Wikipedia [2] Indonesia sistem berasal dari bahasa Latin yaitu *Systema* yang berarti suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi dan energi. Ada beberapa pendapat yang menjelaskan definisi sistem, yaitu:

Sistem adalah kumpulan komponen yang saling terikat dan mempunyai suatu tujuan yang ingin dicapai [6].

Sistem adalah sekumpulan elemen yang saling berhubungan dan saling berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan [8].

Sistem dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Sistem didefinisikan oleh Fredrick H.W.U sebagai berikut: "Sistem adalah suatu sistem yang beroperasi dan berinteraksi dengan lingkungannya untuk mencapai sasaran tertentu, suatu sistem menunjukkan tingkah lakunya melalui interaksi diantara komponen-komponen di dalam sistem dan diantara lingkungannya". Terdapat dua kelompok dasar pendekatan dalam mendefinisikan sistem yaitu berdasarkan pendekatan pada prosedurnya dan yang berdasarkan pendekatan komponennya [7].

1. Pendekatan sistem pada prosedurnya

Suatu sistem adalah jaringan dan prosedur yang saling berkaitan, dan bekerjasama untuk melakukan suatu pekerjaan atau menyelesaikan suatu masalah tertentu.

2. Pendekatan sistem pada komponennya

Suatu sistem adalah sekumpulan dari beberapa elemen yang saling berinteraksi dengan teratur sehingga membentuk suatu totalitas untuk menyelesaikan suatu masalah tertentu.

2.1.2. Karakteristik Sistem

Menurut Yonni Aris (2012:2) [9], sistem mempunyai karakteristik sebagai berikut:

1. Mempunyai Komponen Sistem (*Components System*)

Suatu sistem tidak berada dalam lingkungan yang kosong, tetapi sebuah sistem berada dan berfungsi di dalam lingkungan yang berisi sistem lainnya. Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk kesatuan. Apabila suatu sistem merupakan salah satu dari komponen sistem lain yang lebih besar, maka akan disebut dengan *subsystem*, sedangkan sistem yang lebih besar tersebut adalah lingkungannya.

2. Mempunyai Batasan Sistem (*Boundary*)

Batas sistem merupakan pembatas atau pemisah antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Mempunyai Lingkungan (*Environment*)

Lingkungan luar adalah apa pun di luar batas dari sistem yang dapat mempengaruhi operasi sistem, baik yang menguntungkan ini tentunya harus dijaga sehingga akan mendukung kelangsungan operasi sebuah sistem. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan sebuah sistem.

4. Mempunyai Penghubung (*interface*) Antar Komponen

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Penghubung inilah yang akan menjadi media yang digunakan data dari masukan (*input*) hingga keluaran (*output*). Dengan adanya penghubung, suatu subsistem dapat berinteraksi dan berintegrasi dengan subsistem yang lain membentuk satu kesatuan.

5. Mempunyai Masukan (*input*)

Masukan atau input merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*), yaitu bahan yang dimasukkan agar sistem tersebut dapat beroperasi dan masukan sinyal (*signal input*), yaitu masukan yang diproses untuk mendapatkan keluaran.

6. Mempunyai pengolahan (*processing*)

Pengolahan (*process*) merupakan bagian yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan.

7. Mempunyai Sasaran (*Objective*) dan Tujuan

Suatu sistem pasti memiliki sasaran (*objective*) atau tujuan (*goal*). Apabila sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Tujuan inilah yang mengarahkan suatu sistem. Tanpa adanya tujuan, sistem menjadi tidak terarah dan terkendali.

8. Mempunyai Keluaran (*Output*)

Keluaran (*output*) merupakan hasil dari pemrosesan. Keluaran dapat berupa informasi sebagai masukan pada sistem lain atau hanya sebagai sisa pembuangan.

9. Mempunyai Umpan Balik (*Feed Back*)

Umpan balik diperlukan oleh bagian kendali (*Control*) sistem untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya ke dalam kondisi normal.

2.1.3. Syarat-Syarat Sistem

Ada beberapa syarat dalam sistem diantaranya sebagai berikut [9] :

1. Sistem harus dibentuk untuk menyelesaikan masalah.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

2.1.4. Klasifikasi Sistem

secara umum sistem dapat diklasifikasikan menjadi beberapa jenis [9], yaitu:

1. Sistem dapat diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi, dan lain sebagainya.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran

bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*. Sistem informasi merupakan contoh *man-machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis, sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

2.2. Algoritma Dijkstra

Algoritma *dijkstra* (sesuai penemunya edsgar *Dijkstra*), adalah sebuah algoritma yang dipakai dalam memecahkan permasalahan jarak terpendek (shortest path problem) untuk sebuah graf berarah (directed graph). [7]
langkah-langkah algoritma *dijkstra* dapat dilakukan dengan langkah-langkah berikut: [7]

1. Tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, *dijkstra* akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap.
2. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi) 2.
3. Set semua node yang belum dilalui dan set node awal sebagai "node keberangkatan"
4. Dari node keberangkatan, pertimbangkan node tetangga yang belum dilalui dan hitung jaraknya dari titik keberangkatan. jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru
5. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah dilalui sebagai "node dilewati". node yang dilewati tidak akan pernah dicek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
6. Set "node belum dilewati" dengan jarak terkecil (dari node keberangkatan) sebagai "node keberangkatan" selanjutnya dan ulangi langkah e.

3. METODOLOGI

Metodologi pengembangan sistem merupakan proses standar yang digunakan tim pengembang untuk menghubungkan semua

langkah yang diperlukan untuk menganalisa, merancang, mengimplementasi dan memelihara sistem.

3.1. Metode System Development Life Cycle (SDLC)

SDLC (*System Development Life Cycle*) adalah Proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem informasi. Dalam metode ini terdapat 7 *phase/* langkah yang harus dilakukan peneliti. Adapun output yang dihasilkan di tiap *phase* SDLC yaitu :

1. Identifikasi dan Seleksi (*project identification and selection*)
Pada tahap ini dilakukan pengamatan terhadap lokasi bengkel jasa service ac mobil dan mencatat hasil survey terhadap bengkel-bengkel tersebut. Adapun hal tersebut dilakukan untuk mendapatkan kesimpulan kemungkinan perlunya perancangan aplikasi tersebut. Sehingga dapat dijadikan ruang lingkup penelitian. Setelah dilakukan penelitian maka dapat beberapa fakta tentang lokasi bengkel jasa service ac mobil sehingga dapat disimpulkan bahwa akan dilakukan perancangan aplikasi berbasis web menggunakan algoritma *dijkstra*.
2. Inisialisasi dan Perencanaan (*project initiation and planning*)
Setelah ditetapkan sebuah proyek perancangan aplikasi sesuai dengan langkah pertama, maka akan dilakukan kegiatan awalan atau inisialisasi yaitu dengan lebih terfokus pada kegiatan mengumpulkan fakta-fakta yang berhubungan dengan pencarian bengkel jasa service ac mobil. Untuk kemudian dilakukan perancangan aplikasi baru berbasis web berdasarkan banyaknya pengguna internet disaat ini.
3. Analisa atau Penganalisaan (*Analysis*)
Pada langkah ketiga ini terdiri tiga sub yaitu *Requirement Determination*, *Requirement Structuring*, dan *Alternative Generating Design*.

Berikut uraian ketiga langkah tersebut :

- a. *Requirement Determination* (persyaratan penentuan)
Pada sub langkah ini adalah melanjutkan langkah satu sebelumnya secara lebih terperinci. Survey kembali dilakukan pada area pekanbaru sebagai sumber informasi dan yang akan diinputkan ke sistem komputerisasi yang akan dirancang.
- b. *Requirement Structuring* (persyaratan penataan)
Sub langkah kedua ini dilakukan berdasarkan sub langkah pertama, yaitu menstrukturisasi semua informasi yang telah diperoleh dari pelaksanaan sub langkah pertama tersebut.
Dalam melakukan strukturisasi model ini di gunakan alat bantu diagram ASI, *Context Diagram*, dan *Unified Modeling Language* (UML).
Adapun untuk mendukung pelaksanaan sub langkah ini adalah melakukan penelitian perpustakaan (library research) yaitu dengan melihat literature atau buku-buku yang terkait dengan pembuatan model.
- c. *Alternative Generating Design* (rancangan alternatif)
Pada sub langkah ketiga ini dilakukan untuk menghasilkan beberapa rancangan aplikasi untuk kemudian dipilih salah satunya sebagai rancangan yang akan ditetapkan. Namun dalam penelitian ini dipastikan hanya dirancang satu rancangan aplikasi baru sesuai kebutuhan.
4. Rancangan Logika (*Logical design*)
Merupakan penjabaran dari semua fungsi – fungsi dari sistem yang telah terpilih pada tahap analisa, dijabarkan secara terpisah dari spesifikasi komputer tertentu.
5. Rancangan Fisik (*Physical design*)
Tahap mengubah spesifikasi logika yang dihasilkan dari tahap rancangan logika ke teknologi tertentu diuraikan secara terperinci, yaitu semua pemrograman dan konstruksi dari sistem yang sesuai. Tahap ini diterapkan dan diuraikan pada

pembahasan dalam penelitian yang dilakukan penulis dan dijelaskan dengan lebih terperinci.

6. Implementasi (*Implementation*)

Tahap implementasi adalah dimana sistem informasi dikodekan, ditest, di-instal, prosedur pelatihan dan didukung oleh organisasi. Disini pengguna akan mencoba memakai sistem yang telah dibuat. Selama percobaan akan diawasi oleh pembuat sistem atau admin pengelola. Implementasi akan direalisasikan setelah rancangan fisik selesai dikerjakan.

7. Pemeliharaan (*Maintenance*)

Dengan dibuatnya sistem informasi secara sistematis diperlukan dan terus ditingkatkan. Pada penelitian ini tahap pemeliharaan belum diterapkan karena pemeliharaan dilakukan jika sistem telah di-implementasikan.

4. HASIL DAN PEMBAHASAN

4.1. Tabel

Untuk lebih jelasnya, dibawah ini digambarkan ilustrasi dijkstra di dalam Gmaps API dan penjelasan terperinci pada *graph* (terarah ataupun tidak) kelokasi jasa service ac mobil di kota Pekanbaru, dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

1. Tentukan node awal 1, node tujuan 5. Dimana setiap *edge* (garis/jalan) yang terhubung dengan antar node telah diberi nilai.

Contoh gambar 4.13 dibawah adalah ilustrasi implementasi dijkstra lokasi jasa service ac palapa ac yang diambil dari *ROADMAP* pada Gmap API kemudian dibuat kedalam bentuk *graph*.

Keterangan :

node awal yaitu node 1 yang terletak pada lokasi ramayana panam square no 12 yang ingin menuju ke lokasi node 8 yaitu node terakhir yang terletak pada lokasi palapa ac no 5 yang dimana node 1 harus melewati node-node lainnya agar dapat menuju node 8.

- a. Node 1(node awal) : Jl. Hr Soebrantas, Ramayana Panam Square No 12
- b. Node 2 : Jl. Hr Soebrantas, Duta Carindo No 10
- c. Node 3 : Jl. Hr Soebrantas, Ateng Teknik No 15
- d. Node 4 : Jl. Arengka 2, Kusnan Ac No 30
- e. Node 5 : Jl. Soekarno Hatta, Maman Ac No 5
- f. Node 6 : Jl. Soekarno Hatta, Siyus Ac No 3
- g. Node 7 : Jl. Soekarno Hatta, II Ac No 10
- h. Node 8(node akhir) : Jl. Soekarno Hatta, Palapa Ac No 5

Tabel 1 : Data Jasa Service AC

No	Jasa Service AC	Latitude	Longitude
1	DUTA CARINDO	0.4641470497545316	101.4048668455622
2	KUSNAN AC	0.47228402614477727	101.39471998638749
3	ATENG TEKNIK AC	0.46412586020513213	101.41206207231005
4	II AC	0.45716049120155844	101.41811500887411
5	MAMAN AC	0.4711591299905028	101.41819314426914

Berikut proses perhitungan algoritma dijkstra penentuan rute terpendek jasa service ac mobil yang merujuk kepada artikel eka ismantohadi, iryanto september 2018 pada jurnal teknologi

terapan yang berjudul penerapan algoritma dijkstra untuk penentuan jalur terbaik evakuasi tsunami-studi kasus kelurahan sanur bali :

1. Semua jarak pada selain node 1 bernilai tak hingga
2. Semua node berlabel belum dikunjungi node sekarang adalah node 1
3. Terdapat 2 node yang terhubung dengan node 1 yaitu node 2 dan 4. Jarak dari 1-2 bernilai 6 jarak dari 1-4 bernilai 12.
4. Tandai node 2 terkunjungi karena merupakan bobot terkecil.
5. Tetapkan node 2 sebagai node sekarang karena jarak dari node awal, 2 terkecil.
6.
 - a. terdapat 3 node yang terhubung ke 2 yaitu 1-2-4 bernilai 16, jarak dari 1-2-3 bernilai 13, jarak dari 1-2-6 bernilai 20 dari membandingkan ketiga jarak tersebut jarak 1-2-3 terpilih sebagai jarak tersimpan dengan nilai 13.
 - b. tandai node 3 terkunjungi.
 - c. tetapkan node 3 sebagai node sekarang.
 - d. terdapat 3 node yang terhubung ke node 3 yaitu node 5, 6, 7, jarak dari 1-2-3-5 bernilai 18 jarak dari 1-2-3-6 bernilai 21 jarak dari 1-2-3-7 bernilai 16 maka terpilihlah jarak 1-2-3-7 yang bernilai 16(terkecil).
 - e. Tandai node 7 terkunjungi dan tetapkan node 7 sebagai node sekarang.
 - f. selanjutnya node yang terhubung dengan node 7 hanya 1 yaitu node 8.
 - g. tetapkan node 8 sebagai node sekarang semua node telah terkunjungi sehingga proses selesai.

Dari hasil tersebut diperoleh jarak terpendek dari node awal dengan masing-masing node, node 8 ditetapkan sebagai node tujuan sehingga jarak dari node awal 1 ke node tujuan 8 bernilai 21. $1-2=6$, $1-3=13$, $1-7=16$, $1-8=21$.

Algoritma dijkstra dapat digunakan pencarian rute terpendek pada graf yang memiliki bobot bernilai positif jika terdapat graf yang bernilai negatif algoritma ini tidak dapat digunakan

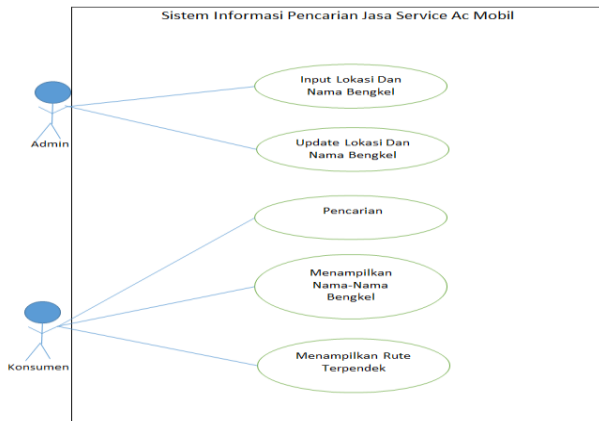
Berikut dibawah ini perhitungan algoritma dijkstra yang disimpulkan dalam bentuk tabel:

Tabel 2 : Perhitungan Dijkstra

Iterat ion	1	2	3	4	5	6	7	8
	0	∞	∞	∞	∞	∞	∞	∞
1		(6)1,2	-	(12)1,4	-	-	-	-
2		-	(13)2,3	(16)2,4	-	(20)2,6	-	-
3		-	-	-	(18)3,5	(21)3,6	(16)3,7	-
4		(12)1,4	(21)4,3	-	(18)4,5	-	-	-
5		-	-	-	-	-	(26)5,7	-
6		-	-	-	-	-	(27)6,7	(28)6,8
7		-	-	-	-	-	-	(21)7,8

4.2. Gambar

4.2.1. Use Case Diagram

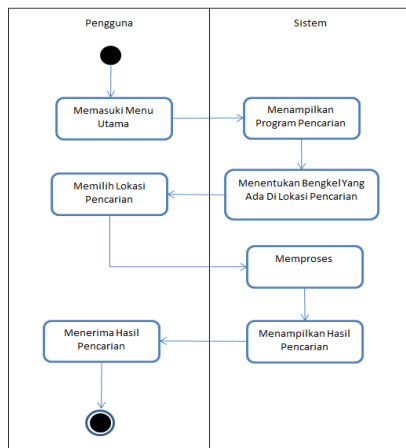


Gambar 1. Use case Baru

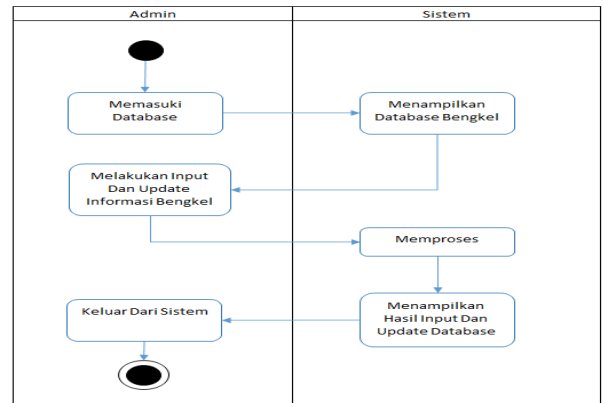
Pada gambar diatas terdapat 2(dua) aktor dalam menjalankan sebuah sistem yaitu admin dan konsumen, admin bertugas untuk menginput data nama jasa service ac mobil, lokasi jasa service ac mobil, mengupdate data-data yang ada di sistem dan konsumen melakukan pencarian lokasi jasa service ac mobil yang ada di sistem, mendapatkan tampilan nama-nama bengkel, dan mendapatkan tampilan rute terpendek admin dengan aktor yang menjalankan 2(dua) proses yaitu input lokasi dan nama bengkel, update lokasi dan nama bengkel dan konsumen dengan aktor yang menjalankan 3(tiga) proses yaitu pencarian, menampilkan nama-nama bengkel, dan menampilkan rute terpendek.

4.2.2. Activity Diagram

Activity diagram merupakan model yang menggambarkan aktivitas yang dilaksanakan oleh aktor maupun sistem informasi yang ada ataupun gabungan keduanya. Aktivitas diagram menunjukkan aliran kerja dari sistem yang dirancang.



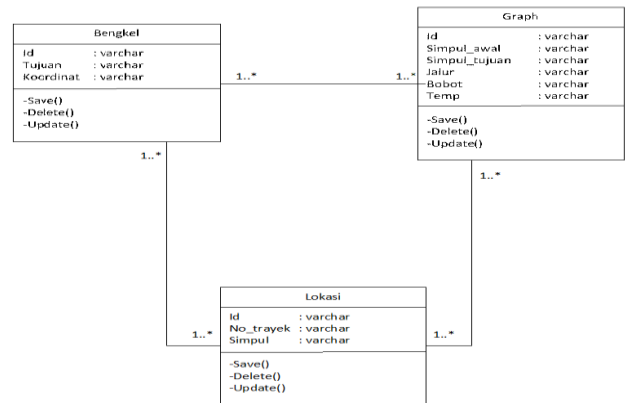
Gambar 2. Activity Diagram Baru Konsumen



Gambar 3. Activity Diagram Baru Admin

4.2.3. Class Diagram

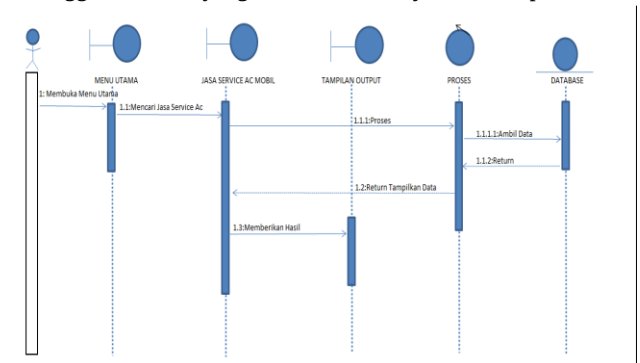
Class diagram menggambarkan struktur system dari segi pendefinisian kelas-kelas yang digunakan untuk membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua table yang dibuat di basis data dapat dijadikan kelas.



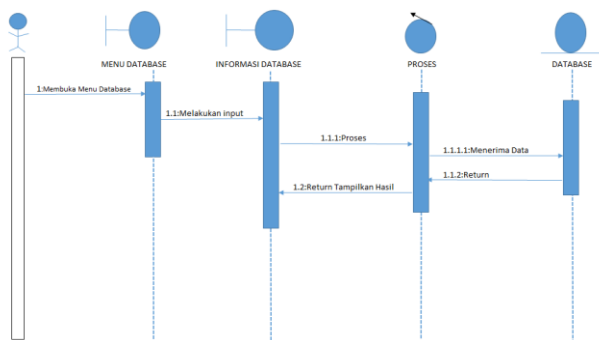
Gambar 4. Class Diagram

4.2.4. Sequence Diagram

Sequence diagram ini merupakan salah satu diagram interaction yang menjelaskan bagaimana suatu operasi itu dilakukan. Objek-objek yang berkaitan dengan proses berjalannya operasi dirurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam diagram yang terurut. Pada sequence diagram, aliran sistem sudah sangat terperinci mulai dari input, data pemrosesan nilai sehingga informasi yang dihasilkan lebih jelas dan terperinci.



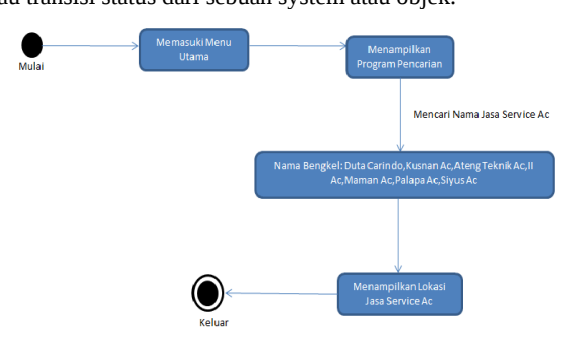
Gambar 5. Sequence Diagram Konsumen



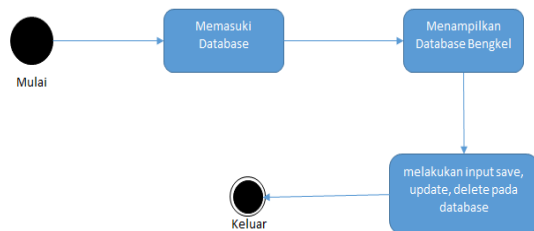
Gambar 6. Sequence Diagram Admin

4.2.5. State Diagram

State diagram menggambarkan perubahan status status atau transisi status dari sebuah system atau objek.



Gambar 7. State Diagram Konsumen



Gambar 8. State Diagram Admin

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Pencarian rute terpendek jasa service ac mobil menggunakan algoritma dijkstra berbasis web ini dapat membantu konsumen dalam mencari lokasi jasa service ac mobil sehingga dapat mengurangi biaya dan menghemat waktu.
2. Pada pencarian rute terpendek jasa service ac mobil menggunakan algoritma dijkstra berbasis web ini dapat membantu para bengkel-bengkel jasa service ac mobil mendapatkan konsumen.

5.2. Saran

Saran yang dapat diberikan bagi pengembangan sistem yang telah dibangun pada pengerjaan skripsi ini adalah sebagai berikut :

1. Sistem dapat dikembangkan dengan bahasa pemrograman yang lain sesuai dengan kebutuhan pengguna dimasa depan.

2. Peneliti selanjutnya dapat mengembangkan penelitian dengan memperluas jangkauan penelitian ke daerah lain untuk mendapatkan lebih banyak informasi jasa service ac mobil.
3. Peneliti selanjutnya dapat mengembangkan sistem ini menjadi berbasis android dan membuat penginputan informasi jasa service ac mobil secara otomatis untuk kebutuhan peneliti dimasa depan.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada Sekolah Tinggi Ilmu Komputer Pelita Indonesia, dan semua pihak yang membantu dalam penelitian ini.

DAFTAR PUSTAKA

- [1] Eka Ismantohadi, Iryanto (2018). Penerapan Algoritma Dijkstra Untuk Penentuan Jalur Terbaik Evakuasi Tsunami (Studi Kasus Kelurahan Sanur Bali).
- [2] Hanif Al-Fatta (2009). Analisis Dan Perancangan Sistem Informasi Untuk Keunggulan Perusahaan Dan Organisasi Kelas Dunia.
- [3] Imron Fauzi (2011). Penggunaan Algoritma Dijkstra Dalam Pencarian Rute Tercepat Dan Rute Terpendek. Jogiyanto, H.M. (2008). Analisis dan Desain Sistem Informasi. Yogyakarta : Andi Offset.
- [4] Mega Yuda Rukmana, Fatwa Ramdani (2018). Implementasi Algoritma Dijkstra Pada Webgis Untuk Pencarian Lokasi SPBU Di Kota Malang.
- [5] Muhammad Syamsuddin Yusuf (2017), Hanifah Muslimah Az-Zahra, Diah Harnoni Apriyanti. Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Lokasi Pengguna Ke Tanaman Yang Di Tuju Berbasis Android (Studi Kasus Di Kebun Raya Purwodadi). Nugroho, Adi. (2011). Perancangan dan Implementasi Sistem Basis Data. Yogyakarta : Andi Offset.
- [6] Retyana fitaria ekasari (2017). Penerapan Algoritma Dijkstra Untuk Menemukan Lintasan Terpendek Pada Pengiriman Barang PT Kharisma Suma Jaya Sakti
- [7] Teguh Setiadi (2006). Pencarian Jalur Terpednek Menuju Pom Bensin Pada Kota Semarang Menggunakan Algoritma Dijkstra Berbasis Sistem Informasi Geografis (GIS).
- [8] Windi Eka Yulia R (2015). Pencarian SPBU Terdekat Dan Penentuan Jarak Terpendek Menggunakan Algoritma Dijkstra (Studi Kasus Di Kabupaten Jember).
- [9] Yonni Aris (2012). Penerapan Algoritma Dijkstra Pada Aplikasi Pekanbaru Taksi Guide Pada Platform Android.