

IMPLEMENTASI ALGORITMA A* MENGGUNAKAN GRAPH PADA APLIKASI ROUTE AT LOCATION BERBASIS WEB

Rizal Ahmad Fauzi¹, Rizal Rachman²

¹Universitas Adhirajasa Reswara Sanjaya
Jl.Sekolah Internasional No.1-2 Antapani, Bandung, (022) 7100124
e-mail: creations.raf@gmail.com

² Universitas Adhirajasa Reswara Sanjaya
Jl.Sekolah Internasional No.1-2 Antapani, Bandung, (022) 7100124
e-mail: rizalrachman@ars.ac.id

Abstract

The application of informed search methods for the search process for routes and directions has been implemented in many digital maps, including Google Maps. But not a few people experience problems when arriving at destination locations such as office buildings, hospitals, airports, stadiums and even recreational areas that have such a large area, namely, lack of floor plan information, to find out the fastest route and direction in order to complete their destination at that location. with optimal travel time. Route at Location is a web-based digital map application that applies the A (A Star) algorithm using Graph in searching for routes and directions at the Bandung Zoo recreation area. The results of this study include, the A* (A Star) algorithm can find the optimal route by using heuristics to rearrange the nodes, so that the search process for the destination node can be found faster and the direction generated from the route search process makes it easier for users to go to. to the destination.*

Keywords: Algoritma A*, Graph, Digital Map

Abstrak

Penerapan *informed search method* (metode pencarian informasi) untuk proses pencarian rute (*route*) dan arah (*direction*) sudah banyak diimplementasikan pada peta digital, diantaranya adalah *Google Maps*. Namun tidak sedikit orang yang mengalami permasalahan ketika tiba di lokasi tujuan seperti gedung perkantoran, rumah sakit, bandara, stadion bahkan tempat rekreasi yang memiliki area yang begitu luas yaitu, kurangnya informasi denah, untuk mengetahui rute dan arah tercepat agar bisa menyelesaikan tujuannya pada lokasi tersebut dengan waktu tempuh yang optimal. *Route at Location* merupakan aplikasi peta digital berbasis *web application* yang menerapkan algoritma A* (*A Star*) menggunakan *Graph* dalam pencarian rute dan arah di tempat rekreasi Bandung Zoo. Hasil dari penelitian ini diantaranya, algoritma A* (*A Star*) dapat menemukan rute yang optimal dengan menggunakan heuristik untuk menyusun ulang *node*, sehingga proses pencarian *node* tujuan dapat ditemukan lebih cepat dan arah (*direction*) yang dihasilkan dari proses pencarian rute mempermudah pengguna untuk menuju ke lokasi tujuan.

Keywords: Algoritma A*, Graph, Peta Digital

1. Pendahuluan

Di era globalisasi, penggunaan peta atau maps sudah tidak menggunakan kertas maupun buku atlas. Kini, maps sudah terdigitalisasi untuk mempermudah penggunaan. Karenanya, diluncurkan Google Maps sebagai alat penunjuk arah dan tempat yang mampu mencakup hampir seluruh daerah di dunia. Dalam penggunaannya, Google Maps menggunakan algoritma untuk menentukan

rute terpendek agar pengguna dapat tiba di lokasi tujuan dengan waktu yang lebih singkat (Prasetyo et al. 2019:37).

Peta digital yang terdapat pada perangkat Smartphone maupun Tablet ini lebih mendapatkan sorotan karena sangat berfungsi ketika kita dalam perjalanan. Bukankah Google Maps akan sangat berguna jika kita ingin mengetahui letak suatu daerah atau tempat ataupun jalan menuju tempat tersebut ketika kita sedang

dalam perjalanan? Bahkan aplikasi Maps ini juga dapat memperhitungkan waktu yang kita perlukan untuk mencapai lokasi tersebut (Setiawan et al. 2018:70).

Didalam kehidupan sehari-hari perpindahan dari suatu tempat ke tempat lain sangat mempertimbangkan efisiensi waktu dan biaya, maka diperlukan pengetahuan untuk menentukan suatu jalur terpendek antar tempat yang ingin dituju. Algoritma pencarian jalur terpendek atau lebih dikenal sebagai Shortest-Path dipakai dalam menentukan rute dalam sebuah Graph. Penerapan konektifitas Graph dalam kehidupan nyata biasa ditemukan di berbagai bidang diantaranya penerapan di bidang telekomunikasi, bidang jaringan komputer, bidang transportasi, jaringan listrik dan saluran air (Dalem, 2018:41).

Teknik pencarian yang sering digunakan untuk menentukan jalur terpendek yaitu pencarian buta (Blind Search) dan pencarian heuristik (Heuristic Search). Pencarian buta cenderung lebih mudah dipahami dibandingkan pencarian heuristik, tetapi hasil pencarian yang diperoleh pencarian heuristik lebih variatif dan waktu pencarian solusi lebih cepat. Salah satu metode pencarian jalur terpendek yang termasuk dalam pencarian heuristik adalah Algoritma A*. Untuk mencapai tujuan dengan jarak tempuh terdekat, Algoritma A* memiliki suatu nilai heuristik yang digunakan sebagai dasar pertimbangan dimana estimasi nilai atau biaya terkecil yang akan menentukan jarak tempuh terdekat (Yamin et al. 2015:1065).

Penerapan informed search method (metode pencarian informasi) untuk proses pencarian rute (route) dan arah (direction) sudah banyak diimplementasikan pada peta digital, diantaranya adalah Google Maps. Namun tidak sedikit orang yang mengalami permasalahan ketika tiba di lokasi tujuan seperti gedung perkantoran, rumah sakit, bandara, stadion bahkan tempat rekreasi yang memiliki area yang begitu luas yaitu, kurangnya informasi denah, untuk mengetahui rute dan arah tercepat agar bisa menyelesaikan tujuannya pada lokasi tersebut dengan waktu tempuh yang optimal.

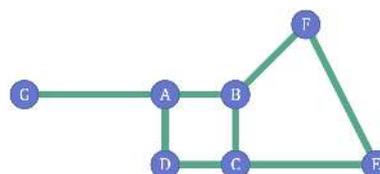
2. Metode Penelitian

Pada tahap pembangunan, penulis menggunakan model SDLC (*Software Development Life Cycle*). Model *waterfall* menjadi pilihan bagi penulis dikarenakan sesuai jika diterapkan pada penelitian ini.

Classic Life Cycle atau Waterfall Model merupakan model yang tidak asing dan banyak digunakan. Dinamakan waterfall karena step by step yang akan dilalui harus menunggu proses keberhasilan step sebelumnya dan bergerak beruntun yang meliputi, tahap-tahap: perencanaan, analisis, desain, implementasi, uji coba dan pemeliharaan.

2.1. Perencanaan Sistem (System Initiation)

Penulis menerapkan graph pada peta digital untuk memberikan jalur yang saling terhubung dengan node untuk keseluruhan rute yang ada pada objek penelitian, dan mengimplementasikan algoritma A Star (A*) untuk penentuan jalur terpendek agar mencapai tujuan dengan cepat dan waktu tempuh yang efisien.



Gambar 2.1. Graph

2.2. Analisis Sistem (System analysis)

Langkah ini merupakan proses analisa untuk kebutuhan aplikasi. Analisis permasalahan, fungsional dan kebutuhan data. Pada tahap ini penulis melaksanakan penelitian dan studi literatur.

2.2.1. Analisis Permasalahan

Langkah-langkah yang dilaksanakan agar dapat menganalisis permasalahan dalam melakukan perancangan sistem adalah:

A. Mengidentifikasi Masalah

Untuk mengidentifikasi masalah, langkah yang dilaksanakan diantaranya melakukan penelitian, menerapkan wawasan yang telah diperoleh, memperhitungkan ruang lingkup dan batasan masalah, serta menjabarkan hasil berlandaskan studi literatur.

B. Menentukan Kebutuhan Informasi Pengguna

Untuk mengidentifikasi kebutuhan informasi pengguna, penulis melihat ada kekurangan dalam fasilitas Bandung Zoo yaitu peta digital dan tidak tersedianya fitur navigasi pada suatu tempat di Google Maps.

C. Menggambarkan Kebutuhan Sistem

Menganalisis kebutuhan sistem navigasi pada peta digital Bandung Zoo dilakukan dengan menggambarkan dan menyusun node hingga saling terhubung dan membentuk graph.

2.2.2. Analisis Operasional (Functionalitas Analysis)

Jika tahap analisis permasalahan selesai maka selanjutnya tahap analisis fungsional. Setelah mendapatkan rumusan permasalahan, ringkasan tujuan dan kebutuhan sistem, terdapat tahap-tahap yang akan dilaksanakan untuk mendapatkan informasi mengenai aplikasi yang akan dibangun, dengan target mampu menangani kebutuhan pengelola Bandung Zoo dan pengguna. Langkah-langkah tersebut antara lain:

- Menentukan kordinat (x dan y) masing-masing node, sesuai dengan lokasi dan jalur yang ada di Bandung Zoo.
- Menghubungkan node yang ada menggunakan edge hingga terbentuk graph.
- Menjabarkan berbagai fungsi yang ada, kemudian objek dan alur mana yang akan beroperasi dalam fungsi yang akan dirancang.
- Mengimplementasikan algoritma A Star (A*) untuk penentuan jalur.

2.2.2. Analisis Kebutuhan Data

Analisis kebutuhan data diperlukan agar memenuhi informasi yang dibutuhkan dalam proses pemetaan digital. Berikut tahapan yang akan dilaksanakan, antara lain:

- Meninjau data lokasi dan posisi satwa maupun fasilitas yang ada di Bandung Zoo.
- Melakukan pengumpulan data yang relevan berdasarkan object dalam sistem navigasi peta digital

2.3. Desain Sistem (System design)

Tahap ini dikerjakan sebelum dilakukannya tahap implementasi pada aplikasi, untuk membuat gambaran yang akan dilakukan dan seperti apa tampilannya. Pada langkah ini dapat membantu untuk menganalisis kebutuhan perangkat keras dan perangkat lunak juga mendefinisikan software architecture secara keseluruhan.

Tujuan dari proses desain ini untuk memudahkan proses pengkodean aplikasi yang akan dibangun, agar sesuai dengan kerangka sistem yang diinginkan.

2.4. Implementasi Sistem (System Implementation)

Tahap ini merupakan penerapan dari tahap sebelumnya, untuk dapat proses oleh komputer, maka tahap desain sistem kemudian di konversi melalui proses coding untuk membangun sebuah aplikasi berdasarkan penelitian yang sedang di kaji, pada kasus ini menggunakan bahasa pemrograman JavaScript dan untuk pertukaran data menggunakan JSON.

2.5. Uji Coba (Testing)

Sesuatu yang dibangun harus dilakukan uji coba. Begitupun dengan aplikasi. Semua fungsi pada aplikasi harus dilakukan uji coba, agar meminimalisir error pada aplikasi, dan hasilnya mesti selaras dengan kebutuhan yang telah direncanakan.

Aplikasi ini diuji dengan metode Black Box untuk mengetahui tahap kesuksesan dari semua fungsi yang terdapat pada aplikasi yang telah dibangun.

2.6. Pemeliharaan (Maintenance)

Penulis belum mencapai tahap ini, oleh karena itu tahap ini belum dilaksanakan. Namun pada dasarnya pemeliharaan suatu aplikasi sangat diperlukan, termasuk tahap pengembangan, karena aplikasi yang dibuat tidak akan terus menerus menggunakan data yang sama pada saat membangun aplikasi. Mungkin disaat dioperasikan masih memiliki error yang sebelumnya tidak terdeteksi, atau ada penambahan fitur yang belum terdapat pada aplikasi sebelumnya. Pengembangan diperlukan ketika terjadi perubahan data dan untuk mengembangkan tentunya mengikuti perubahan waktu atau zaman, serta ketika ada pergantian posisi satwa, fasilitas dan lain-lain.

3. Hasil dan Pembahasan

3.1. Analisis Kebutuhan User

Penulis menganalisis apa saja yang dibutuhkan user pada saat mengakses aplikasi Route at Location, diantaranya:

- Pengguna dapat memilih lokasi awal.
- Pengguna dapat memilih lokasi tujuan.
- Pengguna dapat memulai proses pencarian rute.
- Pengguna dapat menghentikan proses pencarian rute.
- Pengguna dapat menghapus data atau rute yang sebelumnya dicari.
- Pengguna dapat melakukan zoom in sebanyak 3 kali pada peta digital.

- g) Pengguna dapat melakukan zoom out sebanyak 3 kali pada peta digital.
- h) Pengguna dapat melihat jumlah node yang dicari ketika melakukan proses pencarian rute.
- i) Pengguna dapat melihat progres keberhasilan pencarian rute dengan range 0.00% - 100.00%.

3.2. Desain Aplikasi

Pada pembangunan aplikasi ini. Penulis terlebih dahulu merancang desain agar aplikasi yang dibangun sesuai dengan yang diharapkan. Berikut beberapa tahap desain yang penulis lakukan dalam pembuatan aplikasi ini.

3.2.1. Rancangan Graph

Algoritma pencarian pada graph seperti Algoritma A Star (A*) bekerja pada graph berarah yang memiliki bobot, juga perlu tahu apa lokasi itu dan lokasi mana yang terhubung dengan yang lainnya.

A. Pengaturan Graph

Graph adalah kumpulan node dan edge. Node (simpul atau objek) dihubungkan bersama oleh edge (tautan, koneksi, panah atau busur). Terdapat dua hal dasar yang perlu diketahui mengenai graph, yaitu:

- a) Mengatur posisi node dalam graph
- b) Mengatur edge dari setiap node

Contoh kasus graph sederhana, sebagai berikut:

Atur node:

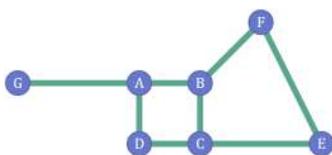
A B C D E F G

Gambar 3.1. Contoh Nodes

Atur edge dari setiap node:

A: A→B A→D A→G
 B: B→A B→C B→F
 C: C→B C→D C→E
 D: D→C D→A
 E: E→C E→F
 F: F→B F→E
 G: G→A

Gambar 3.2. Contoh Nodes



Gambar 3.3. Contoh Nodes

B. Implementasi Graph Pada Peta Digital

Berdasarkan data yang tersedia pada Bandung Zoo diperoleh node sebanyak 324, pada tahap ini penulis membagi 2 kelompok dari total nodes yang ada, sebagai berikut:

- a) Untuk kelompok pertama berisikan data satwa, fasilitas dan gedung berjumlah 40 nodes, yang ditandai dengan warna kuning.



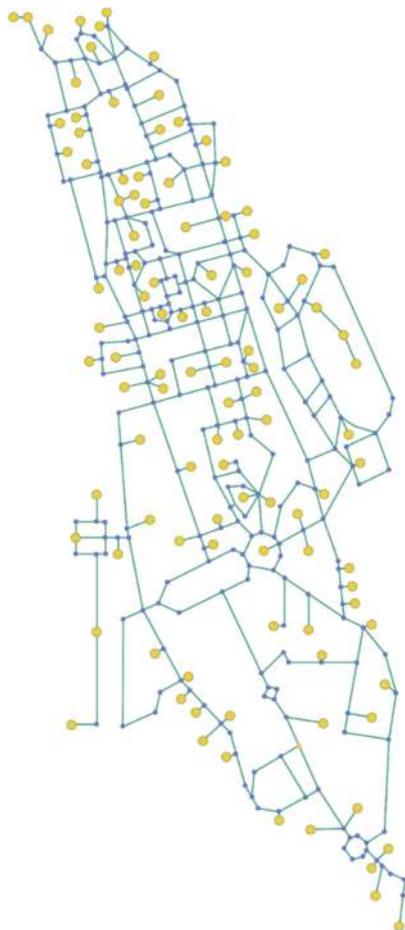
Gambar 3.4. Nodes Kelompok 1

- b) Sedangkan kelompok kedua berisikan data jalur atau rute yang ada pada Bandung Zoo berjumlah 284 nodes, yang ditandai dengan warna biru.



Gambar 3.5. Nodes Kelompok 2

Berikut adalah graph yang diatur sesuai data yang tersedia pada Bandung Zoo:



Gambar 3.6. Graph Bandung Zoo

3.2.2. Rancangan Algoritma A Star (A*)

Dalam sistem pemilihan jalur terdekat pada peta digital Bandung Zoo ini diterapkan algoritma A Star (A*), dengan cara mengikuti jalur yang paling kecil yang dikenal dengan biaya (cost), sehingga antrian prioritas terpisah dari bagian rute alternatif sepanjang jalur yang dilalui.

Jika pada node manapun, terdapat bagian dari rute yang dilalui memiliki cost lebih besar dibandingkan dengan bagian rute pertemuan lainnya, maka A* akan mengabaikan bagian rute yang memiliki cost lebih besar dan akan memilih melintasi bagian rute dengan cost yang lebih kecil. Proses ini berlanjut hingga tujuan tercapai.

- a) Tahap pertama, dibuat sebuah fungsi yang membungkus seluruh perintah yang akan menjalankan fungsi dari astar.

```
function astar(start, goal) {
```

Gambar 3.7. Rancangan A Star Tahap 1

- b) Kedua memasukan perintah tambah marker pada kordinat node awal dan kordiant node tujuan (posisi awal dan posisi tujuan), dilanjutkan dengan mendeklarasikan variabel dasar yang dibutuhkan untuk menampung data atau nilai yang diperlukan.

```
  addStartFlag(nodeCoords(start));
  addGoalFlag(nodeCoords(goal));
  var closedSet = {};
  var openSet = {};
  openSet[start] = true;
  var openSetCount = 1;
  var totalNodesSeen = 1;
  var cameFrom = {};
  var progress = 0;
```

Gambar 3.8. Rancangan A Star Tahap 2

- c) Tahap ketiga mendeklarasikan notasi algoritma A Star (A*) yaitu, $f(n) = g(n) + h(n)$.

```
var gScore = {};
gScore[start] = 0;

var fScore = {};
fScore[start] = gScore[start] + distNodes(start, goal);
```

Gambar 3.9. Rancangan A Star Tahap 3

- $g(n)$ dideklarasikan sebagai `var gScore = {};` dengan nilai awal `gScore[start] = 0;`, $g(n)$ merupakan biaya atau jarak dari node awal ke node(n) atau node berikutnya.
- Kemudian $f(n)$ dideklarasikan sebagai `var fScore = {};` yang bernilai sesuai notasi algoritma A Star (A*).

- Dan untuk $h(n)$ adalah `distNodes(start, goal);`, $h(n)$ merupakan perkiraan biaya atau jarak garis lurus dari node awal (start) atau node(n) ke node tujuan (goal).

- d) Tahap selanjutnya mendeklarasikan `var whileLoop;` dan dilanjutkan dengan `var algo = function() {` untuk membungkus perintah untuk mengeksekusi notasi dari algoritma A Star (A*), sebagai berikut:

- Perintah pertama merupakan proses pencarian node yang akan dilalui dan memasukannya kedalam openSet (Open List), dan menampilkannya pada peta digital Bandung Zoo.

```
if (openSetCount < 1) {
  clearInterval(whileLoop);
  looping = false;
  noPathFound();
  return;
}
var openSetUnsorted = [];
for (var k in openSet) openSetUnsorted.push(k);
var openSetSortedF = openSetUnsorted.sort(
  function(a, b) { return fScore[a] - fScore[b]; }
);
var current = openSetSortedF[0];
```

```
displayNode(nodeCoords(current)); //Tampilkan node y.
```

Gambar 3.10. Rancangan A Star Tahap 4A

- Perintah kedua untuk menghitung progres pencarian rute dan menampilkannya pada info box dan jika progres sudah mencapai 100% maka garis rute (direction) ditampilkan pada peta digital Bandung Zoo.

```
var currProgress = (
  1 - distNodes(current, goal) / distNodes(start, goal)
) * 100;
if (currProgress > progress) {
  progress = currProgress;
}
updateLegend(totalNodesSeen, progress);
if (current == goal) {
  clearInterval(whileLoop);
  looping = false;
  var path = reconstructPath(cameFrom, goal);
  var pathCoords = path.map(nodeCoords);
  pathFound(pathCoords);
  updateLegend(totalNodesSeen, 100);
  return;
}
```

Gambar 3.11. Rancangan A Star Tahap 4B

- Perintah ketiga menghapus node yang berada di openSet (Open List) jika node tersebut mempunyai biaya atau jarak yang kecil, kemudian memasukan node dengan biaya atau jarak kecil ke dalam closedSet (Close List), dan melanjutkan proses perbandingan antara node parent saat ini dengan node parent tetangga atau rute lainnya yang ada di openSet (Open List). Jika pada proses ini ditemukan node parent dengan biaya

atau jarak yang kecil maka akan dimasukkan ke dalam `closedSet` (Close List).

```
delete openSet[current];
openSetCount--;
totalNodesSeen++;
closedSet[current] = true;
var adj = adjNodes(current);
for (var i = 0; i < adj.length; i++) {
  var neighbor = adj[i];
  if (neighbor in closedSet) {
    continue;
  }
  tentativeGScore = gScore[current] + distNodes(current, neighbor);
  if (!(neighbor in openSet) || tentativeGScore < gScore[neighbor]) {
    cameFrom[neighbor] = current;
    gScore[neighbor] = tentativeGScore;
    fScore[neighbor] = gScore[neighbor] + distNodes(neighbor, goal);
    if (!(neighbor in openSet)) {
      openSet[neighbor] = true;
      openSetCount++;
    }
  }
}
```

Gambar 3.12. Rancangan A Star Tahap 4C

e) Tahap terakhir yaitu melakukan pengulangan perintah sebelumnya yang ada di `var algo = function() {` hingga proses selesai atau rute sudah ditemukan.

```
};
if (searchDelay < 0) {
  var looping = true;
  whileLoop = 0;
  while (looping) {
    algo();
  }
} else {
  whileLoop = setInterval(algo, searchDelay);
}
}
```

Gambar 3.13. Rancangan A Star Tahap 5

3.2.3. Penyimpanan dan Pertukaran Data

JSON biasa digunakan untuk format standar pada saat pertukaran data antar aplikasi, namun seharusnya tidak hanya itu saja, masih terdapat fungsi lainnya dari JSON. Pada kasus ini penulis menggunakannya untuk penyimpanan data node dan edge untuk ditampilkan pada aplikasi Route at Location.

A. Struktur Data

Pada tahap ini penulis membagi 3 file untuk masing-masing data, sebagai berikut:

a) File `nodes.json` untuk menyimpan seluruh data nodes dan koneksi (edges) antara node yang ada pada peta Bandung Zoo.

```
{
  "node1": {"lat":16.8466, "lon":84.4848, "adj":["node2", "node3"]}
}
```

Gambar 3.14. Struktur Data File `nodes.json`

Pada gambar 3.14 terdapat:

- Key `"node1"` berisi ID dari sebuah node.

- `"lat"` dan `"lon"` merupakan nilai object pada ID node, yang selanjutnya di masukan value sesuai data latitude dan longitude node. Tahap ini untuk menentukan posisi koordinat node pada peta digital Bandung Zoo.

- Object `"adj"` untuk menentukan nilai array sebagai koneksi pada node, yang berisi value ID node lain yang terhubung dengan key. Tahap ini mengatur edges, untuk key ini berisikan dua ID lain yaitu `node2` dan `node3` yang terhubung dengan key `node1`. Jika diartikan bahwa `node1` bisa terhubung ke `node2` dan `node3`.

b) File `locs.json` untuk menampilkan data node yang ada di file `nodes.json` sesuai dengan pengelompokannya, yaitu kelompok 1 untuk menampilkan data satwa, fasilitas dan gedung pada peta digital Bandung Zoo.

```
[
  {"name": "Harimau", "node": "3_harimau"}
]
```

Gambar 3.15. Struktur Data File `locs.json`

Pada gambar 3.15. terdapat:

- Object `"name"` berisi nama atau inisial ID node.
- `"node"` merupakan object dengan value ID node.

c) File `locs_path.json` untuk menampilkan data node yang ada di file `nodes.json` sesuai dengan pengelompokannya, yaitu kelompok 2 untuk menampilkan data jalur atau rute yang ada pada Bandung Zoo.

```
[
  {"name": "Node 1", "node": "node1"}
]
```

Gambar 3.16. Struktur Data File `locs_path.json`

Pada gambar 3.16. terdapat:

- Object `"name"` berisi nama atau inisial ID node.
- `"node"` merupakan object dengan value ID node.

3.2.4. User Interface

User Interface adalah tampilan visual yang dilihat langsung oleh pengguna (user). User Interface ini untuk memperlihatkan bagaimana tampilan dari aplikasi yang telah dibangun oleh penulis. User Interface pada aplikasi web ini meliputi:

A. Tampilan Halaman Introduction

Halaman Introduction merupakan halaman yang pertama kali yang akan tampil pada saat pengguna membuka atau mengakses aplikasi Route at Location. Halaman Introduction ini merupakan halaman yang berisikan tentang aplikasi Route at Location.



Gambar 3.17. Halaman Introductions

B. Tampilan Halaman Peta Digital Bandung Zoo

Halaman peta digital merupakan tempat interaksi pengguna untuk mengeksplorasi tempat wisata Bandung Zoo, menemukan berbagai lokasi yang ada dengan mudah dan menentukan lokasi awal untuk menuju lokasi yang diinginkan.



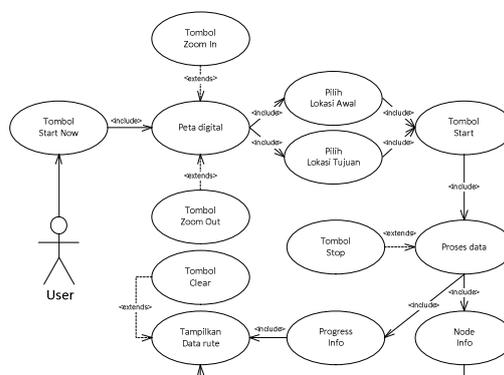
Gambar 3.18. Halaman Peta Digital

3.3. Implementasi Aplikasi

Implementasi merupakan tahapan dimana segala rancangan yang dibangun akan diuraikan, dan untuk mendapatkan gambaran mengenai aplikasi Route at Location yang akan dibuat, dilakukan pemodelan dengan menggunakan functional modeling. Proses dari aplikasi Route at Location dimodelkan dengan Use Case Diagram.

3.3.1. Use Case Diagram

Melalui use case diagram mampu mengetahui proses yang beroperasi pada aplikasi. Berikut merupakan use case diagram pada aplikasi Route at Location.



Gambar 3.19. Use Case Diagram

Gambar 3.19. Use Case Diagram Route at Location dijelaskan pada gambar dibawah ini.

Aktor	Nama Use Case	Deskripsi Use Case
User	Tombol Start now	Use case ini digunakan untuk masuk ke dalam sistem peta digital Bandung Zoo
User	Pilih lokasi awal	Use case ini digunakan untuk memilih lokasi awal yang diinginkan oleh user
User	Pilih lokasi tujuan	Use case ini digunakan untuk memilih lokasi tujuan yang diinginkan oleh user
User	Tombol Start	Use case ini digunakan untuk memulai proses pencarian rute pada peta digital Bandung Zoo
User	Tombol Stop	Use case ini digunakan untuk menghentikan proses pencarian rute pada peta digital Bandung Zoo
User	Tombol Clear	Use case ini digunakan untuk menghapus data rute yang telah di proses sebelumnya
User	Tombol Zoom in	Use case ini digunakan untuk memperbesar peta digital Bandung Zoo
User	Tombol Zoom out	Use case ini digunakan untuk memperkecil peta digital Bandung Zoo
User	Node info	Use case ini digunakan untuk melihat jumlah node yang diproses dalam pencarian rute
User	Progress info	Use case ini digunakan untuk melihat presentase progres pencarian rute

Gambar 3.20. Use Case Diagram

3.4. Pengujian Aplikasi

Testing atau pengujian sistem adalah sesuatu yang wajib dilakukan untuk mendeteksi kesalahan (error) pada aplikasi yang akan diuji. Pengujian dilakukan untuk mengetahui kinerja suatu aplikasi yang telah dibangun agar sesuai dengan tujuan.

3.5. Black Box Testing

Setelah melewati tahap-tahap sebelumnya, maka pada proses akhir ini penulis melaksanakan uji coba untuk aplikasi Route at Location menggunakan metode black box.

Skenario Uji	Test Case	Hasil yang Diharapkan	Keterangan
Pilih tombol <i>Start now</i>	Menampilkan halaman peta digital	Tampil halaman peta digital	Valid
Pilih lokasi awal	Memunculkan marker berwarna hijau pada lokasi yang dipilih	Muncul marker berwarna hijau sesuai dengan lokasi yang dipilih	Valid
Pilih lokasi tujuan	Memunculkan marker berwarna merah pada lokasi yang dipilih	Muncul marker berwarna merah sesuai dengan lokasi yang dipilih	Valid
Pilih tombol <i>Start</i>	Memproses rute sesuai dengan lokasi awal dan tujuan yang dipilih dan menampilkannya pada peta digital	Muncul rute tercepat yang diatur sesuai lokasi awal dan tujuan	Valid
Pilih tombol <i>Stop</i>	Menghentikan proses pencarian rute	Proses pencarian berhenti	Valid
Pilih tombol <i>Clear</i>	Menghapus data rute sebelumnya	Data rute yang sudah diproses terhapus dari peta digital	Valid
Pilih tombol <i>Zoom in</i>	Memperbesar tampilan peta digital	Tampilan peta digital membesar	Valid
Pilih tombol <i>Zoom out</i>	Memperkecil tampilan peta digital	Tampilan peta digital mengecil	Valid
Lihat info <i>Node</i>	Menampilkan jumlah node yang diproses	Jumlah node yang diproses tampil pada <i>info box</i>	Valid
Lihat progres pencarian	Menampilkan persentase keberhasilan progres pencarian rute	Jumlah persentase proses pencarian tampil pada <i>info box</i>	Valid

Gambar 3.21. Black Box Testing

4. Kesimpulan

Berdasarkan penjabaran pada bab sebelumnya mengenai implementasi algoritma A* (A Star) menggunakan Graph pada aplikasi Route at Location berbasis Web Application, baik itu secara teori maupun implementasi, maka penulis memperoleh kesimpulan, diantaranya:

- Telah dibangun aplikasi Route at Location (Peta digital berbasis web application) dengan menerapkan algoritma A Star (A*) untuk proses pencarian lokasi dan Graph sebagai lintasan penghubung bagi setiap lokasi yang ada di Bandung Zoo.
- Algoritma A* (A Star) dapat menemukan rute yang optimal dengan menggunakan heuristik untuk menyusun ulang node, sehingga proses pencarian node tujuan dapat ditemukan lebih cepat.

- Arah (direction) yang dihasilkan dari pencarian rute mempermudah pengguna untuk menuju ke lokasi tujuan

4.1 Saran

Dalam pembuatan peta digital ini belum sepenuhnya sempurna, karena keterbatasan yang ada dari segi keilmuan dan pengetahuan penulis. Maka dapat di usulkan beberapa saran sebagai berikut:

- Dengan telah di terapkannya algoritma A Star (A*) pada peta digital Bandung Zoo, untuk pengembangan selanjutnya dapat menggunakan informed search method (metode pencarian informasi) lainnya dalam penentuan rute dari metode algoritma yang diterapkan pada saat ini, seperti algoritma BFS (Breadth First Search).
- Penentuan lokasi awal saat ini berdasarkan pada data posisi yang sudah diatur menggunakan latitude dan longitude yang disimpan pada file JSON. Untuk kedepannya bisa menerapkan lokasi sebenarnya menggunakan fitur GPS dan lainnya.

Referensi

- A.S, Rosa, and M Shalahuddin. 2015. "Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek. Bandung : Informatika." *Jurnal Pilar Nusa Mandiri*.
- Budiman, Vicky, Yogi Swara Hendro Lesmono, and Halim Agung. 2018. "Aplikasi Berbasis Android Untuk Mencari Lokasi Puskesmas Terdekat Dengan Algoritma A-Star Di Provinsi Dki Jakarta." *Jurnal Sistem Informasi, Teknologi Informatika Dan Komputer* Vol.9 (No.1): 39–48.
- Gede Wahyu Antara Dalem, Ida Bagus. 2018. "Penerapan Algoritma A* (Star) Menggunakan Graph Untuk Menghitung Jarak Terpendek." *Jurnal RESISTOR (Rekayasa Sistem Komputer)* 1 (1): 41–47. <https://doi.org/10.31598/jurnalresistor.v1i1.253>.
- Kiki Setiawan, Supriyadin, Imam Santoso, Roy Buana. 2018. "Menghitung Rute Terpendek Menggunakan Algoritma a * Dengan Fungsi Euclidean Distance." *Seminar Nasional Teknologi Informasi Dan Komunikasi* 2018 (Sentika): 70–79.
- Kom, M, M Kom, Studi Ilmu Komputer, and Universitas Pakuan. 2019. "Di Kota Bogor Dengan Menggunakan Algoritma

- a *” 16 (1): 245–53.
- Maulana, Afrizal Adam, and Wijanarto. 2019. “Implementasi Algoritma A* Dalam Aplikasi Berbasis Web Untuk Menemukan Rute Terpendek Sebagai Navigasi Peta Digital Indoor.” *Creative Information Technology Journal* 5 (1): 1. <https://doi.org/10.24076/citec.2017v5i1.129>.
- Mayadi, Mayadi, and Raisul Azhar. 2019. “Perbandingan Perhitungan Manual Dengan Algoritma A Star Dalam Pencarian Jalur Terpendek Untuk Pengiriman Pesanan Dodol Khas Lombok.” *Jurnal Informatika Dan Rekayasa Elektronik* 2 (2): 27. <https://doi.org/10.36595/jire.v2i2.114>.
- Prasetyo, Arif Cahyo, Maful Prayoga Arnandi, Harish Setyo Hudnanto, and Bayu Setiaji. 2019. “Perbandingan Algoritma Astar Dan Dijkstra Dalam Menentukan Rute Terdekat.” *SISFOTENIKA*. <https://doi.org/10.30700/jst.v9i1.456>.
- Pressman, Roger S., and Bruce R. Maxim. 2015. *Software Engineering: A Practitioner’s Approach, Eighth Edition*. ACM SIGSOFT Software Engineering Notes. <https://doi.org/10.1145/1226816.1226822>.
- Purnamasari, Popy, and Wahid Suyitno. 2016. “Pencarian Jalur Terpendek Dari Rumah Menuju Candi Jiwa Batujaya Menggunakan Algoritma A.” *KNIT-2 Nusa Mandiri*, 171–76.
- Rizky, Robby, Jurusan Sistem Informasi, Fakultas Informatika, and Universitas Mathla. 2018. “Pencarian Jalur Terdekat Dengan Metode A*(Star) Studi Kasus Serang Labuan Provinsi Banten 1),” no. November.
- Scarlet, Douglas. 2013. “Belajar Dasar Algoritma Dan Pemrograman.” *Journal of Chemical Information and Modeling* 53 (9): 1689–99. <https://doi.org/10.1017/CBO9781107415324.004>.
- Yamin, Muh., and Moh. Bandrigo Talai. 2015. “Aplikasi Pencarian Jalur Terpendek Pada Rumah Sakit Umum Bahteramas Menggunakan Algoritma A* (A-Star).” *Jurnal Informatika* 9 (2). <https://doi.org/10.26555/jifo.v9i2.a2961>.