

## ALGORITMA A\* UNTUK Mencari KOSAN TERDEKAT DENGAN UMMI SUKABUMI BERBASIS WEB

**Didik Indrayana, Anthony Suryadjie, Asril Adi Sunarto**  
Teknik Informatika, Universitas Muhammadiyah Sukabumi  
Jl.R.Syamsudin, S.H. No. 50 Sukabumi, Indonesia  
*didik.ind@ummi.ac.id*

### ABSTRAK

Di era teknologi yang berkembang pesat, kemajuan teknologi memberikan dampak signifikan terhadap berbagai aspek kehidupan manusia, termasuk komunikasi transportasi, dan pencarian informasi. Internet dan teknologi berbasis web telah menjadi bagian kebiasaan dari kehidupan sehari-hari, memungkinkan akses informasi dengan mudah dan cepat. Salah satu sektor yang merasakan dampak positif dari perkembangan teknologi ini adalah pendidikan, dimana mahasiswa sering memanfaatkan teknologi untuk mencari hunian sementara seperti kosan. Mencari kosan dekat dengan kampus menjadi perhatian utama bagi mahasiswa karena berbagai keuntungan yang ditawarkannya, seperti menghemat waktu dan biaya transportasi serta memudahkan partisipasi dalam kegiatan akademik dan akses ke fasilitas kampus. Pada fenomena ini penggunaan metode algoritma A\* merupakan metode yang cocok untuk mencari kosan terdekat dengan area kampus. Algoritma A\* dipilih karena kemampuannya mencari rute terdekat secara meluas dengan menghitung nilai heuristik nya. Untuk menentukan bobot pada setiap graf, digunakan algoritma haversine untuk mengukur jarak antar graf yang saling berhubungan. Dengan dilakukannya Penelitian ini mahasiswa dapat dengan mudah untuk menemukan kosan yang strategis dan lebih efisien, mengurangi waktu dan tenaga yang diperlukan dalam proses pencarian, serta membantu mahasiswa baru menemukan kosan serta dapat memberikan pelayanan tambahan untuk mahasiswa UMMI secara tidak langsung dan diharapkan aplikasi ini bisa menjadi salah satu bagian dari fitur dari siak UMMI Sukabumi.

**Kata kunci :** Algoritma, Kosan, A\*(star), GIS

### 1. PENDAHULUAN

Pada era teknologi yang berkembang pesat, sangat mempengaruhi kehidupan manusia. Kemajuan dari teknologi ini memberikan dampak kemudahan untuk berbagai aspek kehidupan, seperti hal nya komunikasi, transportasi, hingga pencarian informasi, internet dan teknologi berbasis web khususnya, telah menjadi bagian integral dari kehidupan sehari-hari, memungkinkan akses informasi dengan mudah dan cepat. Salah satu faktor yang mendapatkan respon positif dari perkembangan teknologi yaitu pendidikan. Mahasiswa, dalam dunia pendidikan seringkali memanfaatkan teknologi untuk memenuhi berbagai kebutuhan, termasuk mencari hunian/kosan/kontrakan. Mendapatkan informasi yang valid menjadi kebutuhan penting bagi mahasiswa, terutama mereka yang menempuh pendidikan di luar kota atau biasa disebut merantau ke kota lain.

Pada fenomena yang terjadi terkait mencari kosan dekat dengan kampus semakin menjadi perhatian utama bagi mahasiswa, mengingat berbagai keuntungan yang ditawarkannya. Kemudahan akses menuju kampus tanpa perlu menempuh jarak jauh merupakan salah satu daya tarik utama, yang tidak hanya menghemat waktu tetapi juga biaya transportasi. Tinggal dekat kampus juga memungkinkan mahasiswa untuk lebih aktif berpartisipasi dalam kegiatan akademik dan unit kegiatan kampus, serta mempermudah akses ke fasilitas kampus seperti perpustakaan dan pusat studi. Selain itu, lingkungan yang dekat dengan kampus biasanya dikelilingi oleh

berbagai fasilitas pendukung seperti tempat makan, toko kebutuhan sehari-hari, dan pusat Kesehatan, yang semuanya menunjang kehidupan sehari-hari mahasiswa. Mengingat pentingnya aspek-aspek ini, dalam hal yang mendesak untuk menemukan kosan yang strategis semakin meningkat, terutama di awal tahun ajaran baru ketika permintaan tempat tinggal membludak.

Untuk mencari kosan dekat dengan kampus seringkali menjadi tantangan tersendiri bagi mahasiswa, terutama karena tingginya permintaan yang menyebabkan persaingan ketat. Banyak mahasiswa kesulitan menemukan kosan yang sesuai dengan kebutuhan mereka, baik dari segi lokasi dan hal lainnya. Proses pencarian juga memakan waktu dan tenaga, karena mahasiswa harus mengunjungi banyak tempat yang belum tentu valid informasi keberadaan tempat kosan nya untuk membandingkan berbagai opsi yang ada. Situasi ini seringkali membuat mahasiswa khawatir tidak mendapatkan hunian sebelum perkuliahan dimulai. Dalam permasalahan ini penulis memiliki ide untuk membuat pencarian kosan terdekat dengan kampus dengan bantuan Algoritma. Ada beberapa metode algoritma untuk menyelesaikan permasalahan tersebut, setelah penulis teliti dari beberapa jurnal yang menghadapi fenomena yang serupa, ada beberapa algoritma yang ditawarkan untuk menyelesaikan fenomena ini yaitu Algoritma yang ditawarkan untuk fenomena ini yang penulis temukan yaitu menggunakan A-star(A\*), Bellman-Ford, Floyd-warshall, Dijkstra, dan yang lainnya.

Dalam menghadapi masalah ini penulis akan menggunakan algoritma A\* karena algoritma ini merupakan pembaruan dari algoritma dijkstra bukan hanya itu saja algoritma ini dapat memproses secara efektif dalam mencari rute terdekat dengan memilih secara langsung graf yang tersambung pada graf lainnya dengan bobot yang minim [1], untuk menentukan bobot pada setiap graf menggunakan bantuan dari algoritma haversine untuk mengukur jarak antar graf yang saling berhubungan, hasil perhitungan jarak tersebut akan menjadi bobot untuk graf yang saling berhubungan saja dan nantinya akan dieksekusi oleh dijkstra untuk pemilihan rute dengan bobot terendah. pada proyek pencarian rute ini membutuhkan dokumentasi yang mendetail seperti pengumpulan data hingga hasil akhir sehingga membantu dalam pelacakan dan verifikasi, selain itu dokumentasi secara rinci juga membantu menciptakan transparansi dan menyediakan rekam jejak yang lengkap. Metode waterfall merupakan metode yang pas untuk membantu penyelesaian masalah terkait, metode ini juga menyediakan keterbacaan proses yang baik, memungkinkan semua pemangku kepentingan untuk memahami dan mengikuti alur proyek dengan mudah.

Terdapat penelitian yang serupa seperti penelitian oleh Fairuz Eka Andiany dan Wiwien Hadikurniawati yang membahas tentang implementasi algoritma dijkstra untuk mencari rute terdekat antar kantor dan estimasi penggunaan bahan bakar kendaraan, lalu penelitian dari Aldy Cantona membahas implementasi algoritma dijkstra untuk mencari rute terdekat ke museum di Jakarta, Muhmmad Taufiq Ismail yang membahas implementasi algoritma dijkstra untuk mencari rute menuju rumah sakit pada aplikasi C-Hos, Maulana Syepanda membahas perancangan aplikasi pencarian rute wisata kuliner dengan algoritma dijkstra, penelitian tentang penerapan algoritma dijkstra untuk menentukan rute terdekat pada jasa pengiriman barang di palangkaraya yang diteliti oleh Edward Christian Rufus, dan terakhir penelitian tentang Sistem informasi untuk pemetaan dan geolokasi COVID-19 di Kota Sukabumi yang diteliti oleh Asril Adi Sunarto dan Yuli Noviawan. Selain itu juga adapun platform seperti mamikos yang menyediakan informasi kosan namun paltform tersebut masih ada beberapa data kosan yang belum update dan letak lokasi yang kurang akurat. Dari Penelitian ini akan dibentuk hasil akhir nya menjadi aplikasi dalam bentuk web agar mudah diakses.

## 2. TINJAUAN PUSTAKA

### 2.1. Rumah kost

Kost bisa disebut juga sebagai indekos didalam kamus Bahasa Indonesia dapat dikenal sebagai menumpang tinggal di rumah orang lain dengan makan atau tidak[2].

Rumah kost ini banyak diminati untuk kalangan mahasiswa karena harga yang terjangkau dengan fasilitas sesuai dengan kebutuhan.

### 2.2. WebGIS

Pengertian webGIS merupakan system informasi geografis dikembangkan dari aplikasi GIS yaitu sebuah system pemetaan secara digital dengan memanfaatkan sebuah internet untuk menyimpan, mempublikasi dan mengumpulkan data spasial dalam bentuk teks[3].

### 2.3. Graph

Graph merupakan susunan data umum yang tidak berdekatan dengan satu sama lain[4]. Graph juga merupakan Kumpulan simpul berhingga untuk menyimpan data antara simpul yang berhubungan.

### 2.4. Algoritma A\*(star)

Algoritma A\*(star) merupakan jenis algoritma yang sering digunakan untuk mencari rute terdekat atau permasalahan jarak terpendek[5].

Algoritma ini menerapkan pencarian dengan melebar ke setiap *node* pada tingkatan yang sama, sehingga mendapatkan rute terbaik dari titik awal ke tujuan. Algoritma ini disertai dengan fungsi heuristic, fungsi ini digunakan untuk optimasi dalam memutuskan pemilihan *node* tujuan yang akan dilalui.

### 2.5. Flowchart

Flowchart merupakan sebuah diagram visual untuk menggambarkan runutan proses atau sebuah alur kerja untuk memudahkan proses pengambilan Keputusan[6].

Diagram alur ini dapat membantu mendeskripsikan Langkah-langkah dengan detail untuk mencapai hasil yang optimal.

### 2.6. Algoritma Haversine

Algoritma ini adalah sebuah formula untuk menghitung jarak koordinat diantara dua titik pada bumi[7].

Algoritma ini banyak dipakai untuk sistem informasi geografis karena perhitungannya yang akurat, keakuratan formula ini membutuhkan garis lintang dan garis bujur untuk menghitung jarak sebenarnya pada lingkaran maka dari itu algoritma ini banyak digunakan pada sistem informasi geografis.

### 2.7. Leaflet JS

Leaflet JS merupakan library javascript untuk map yang interaktif dan mudah digunakan oleh perangkat seluler[8].

Ukuran dari kerangka kerja ini hanya 42 KB sudah banyak memiliki fitur yang dibutuhkan oleh para pengembang.

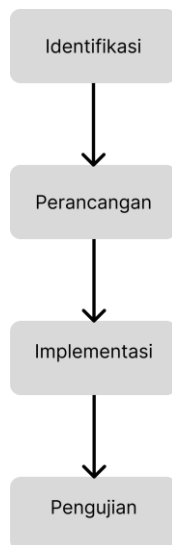
### 2.8. Laravel

Laravel merupakan kerangka kerja aplikasi web dari bahasa pemrograman PHP yang gratis untuk membangun *website* dan aplikasi[9].

Laravel ini banyak diminati untuk pengembangan web karena dapat dilakukan dengan mudah dan eifisien.

### 3. METODE PENELITIAN

Pada penelitian ini menggunakan metode Algoritma A\*(star), berikut merupakan alur dari metode penelitian ini.



Gambar 1. Alur penelitian

#### 3.1. Identifikasi

Pada tahap penelitian ini yaitu melakukan studi literatur terkait permasalahan yang dihadapi dengan cara melakukan pengkajian dari penelitian-penelitian sebelumnya untuk memahami bagaimana algoritma A\* diimplementasikan kedalam sistem serta memahami Langkah-langkah setiap perhitungannya.

#### 3.2. Perancangan

Pada perancangan penelitian ini yaitu menyusun desain graf yang nantinya akan digunakan seperti simpul (*vertex*) dan tepi (*edges*), lalu menentukan rumus untuk mencari nilai heuristik, selanjutnya Menyusun algoritma A\* menjadi bentuk pseudocode atau diagram alur untuk menggambarkan proses pencarian.

#### 3.3. Implementasi

Di tahapan implementasi melakukan pengolahan data seperti menyusun data input sesuai format yang dibutuhkan oleh algoritma, Adapun tahap pengkodean untuk mengimplementasikan algoritma A\* kedalam Bahasa pemrograman yaitu PHP dan menguji bagian kecil dari sebuah kode tersebut agar memastikan algoritma berfungsi dengan benar.

#### 3.4. Pengujian Aplikasi

Bagian pengujian penulis melakukan pengujian aplikasi mandiri dengan pengujian *black box testing* pengujian tersebut yaitu menguji setiap *input output* dari perangkat lunak tanpa mengetahui struktur kode[10].

#### 3.5. Lokasi dan Objek Penelitian

Objek penelitian pada penelitian ini difokuskan kepada mahasiswa baru Universitas Muhammadiyah

Sukabumi maupun lama yang sedang mencari kostan atau pindah dari kostan lama ke kostan yang lain (lebih dekat dengan kampus), untuk Lokasi penelitian ini yaitu di daerah kota sukabumi yang tidak jauh dengan kampus berikut merupakan pengambilan layar dari google maps beserta titik persimpangan jalan yang nantinya berguna untuk pengukuran jarak antara kostan dengan kampus UMMI.

### 4. HASIL DAN PEMBAHASAN

#### 4.1. Perhitungan Jarak

Untuk perhitungan jarak akan dihitung terlebih dahulu dengan algoritma haversine yang nantinya jarak tersebut akan menjadi bobot dari jarak ke jarak yang sudah ditentukan kemudian bobot tersebut akan ditentukan oleh algoritma A\* untuk penentuan rute. Berikut merupakan rumus dari haversine dan algoritma A\*

##### a. Rumus haversine

$$d = 2R \cdot \arcsin(\sqrt{\sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2)}) \quad \dots(1)$$

d = merupakan jarak antara dua titik diukur

R = jari-jari bumi

$\phi$  = garis lintang latitude dalam radian

$\lambda$  = garis bujur dalam radian

$\Delta\phi$  = perbedaan garis lintang

$\Delta\lambda$  = perbedaan garis bujur

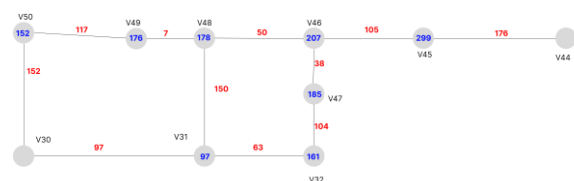
##### b. Rumus A\*

$$f(n) = g(n) + h(n) \quad \dots(2)$$

f(n) = nilai total untuk simpul n

g(n) = biaya jalur dari titik awal hingga simpul n

h(n) = perkiraan biaya dari simpul n ke tujuan



Gambar 2. Graph

Pada titik awal dan titik tujuan diberikan nilai heuristik kosong karena titik awal dan titik tujuan tidak perlu diberi nilai, kemudian pada nilai heuristik node yang lain didapatkan dengan menggunakan rumus haversine serta simpul atau graph yang saling terhubung satu sama lain juga didapatkan menggunakan rumus haversine. Lalu pencarian rute terpendek akan menggunakan rumus A\*

##### a. Jika dihitung dari titik awal yaitu V44 menuju V30 yaitu

$$f(V45) = 176 + 299 = 475$$

- b. diketahui bahwa V44 hanya terhubung ke V45 jadi bisa dilanjutkan langsung ke perhitungan selanjutnya.

$$f(V46) = (176 + 105) + 207 = 488$$

- c. pada perhitungan selanjutnya menghitung fungsi  $g(n)$  yang telah dilalui yaitu jalur yang sudah dilalui V44 ke V45 dan V45 menuju V46.

$$f(V47)=(176+105+38)+185=504$$

$$f(V48) = (176 + 105 + 50) + 178 = 509$$

- d. perhitungan diatas karena V46 terhubung ke 2 vertex yang berbeda maka dilakukanlah perhitungan untuk kedua cabang untuk menentukan biaya yang lebih murah namun karena algoritma ini bersifat melebar kesetiap node nya perhitungan pada f(48) masih di evaluasi dan akan masih berlanjut.

$$f(V32) = (176 + 105 + 38 + 104) + 161 = 584$$

$$f(V31) = (176 + 105 + 50 + 150) + 97 = 578$$

- e. Terlihat perhitungan berikut setelah evaluasi melalui jalur V48 memiliki biaya lebih rendah untuk ke V30.

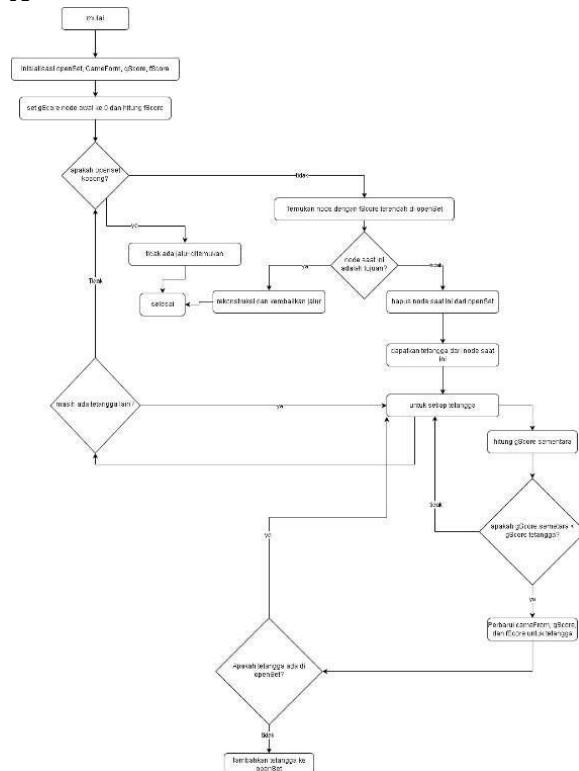
$$f(30)=(176+105+50+150+97)+0=578$$

$$f(31)=(176+105+38+104+63)+97=583$$

- f. Pada tahap ini terjadilah *closed list* untuk jalur  $V44 \rightarrow V45 \rightarrow V46 \rightarrow V47 \rightarrow V32 \rightarrow V31$  karena memiliki biaya yang lebih besar dibanding jalur yang ditempuh yaitu  $V44 \rightarrow V45 \rightarrow V46 \rightarrow V48 \rightarrow V31 \rightarrow V30$

#### 4.2. Diagram alur algoritma A\*

Berikut merupakan diagram alur dari algoritma A\*



Gambar 3. Diagram Alur Algoritma A\*

### 4.3. Pengimplementasian kode

Berikut merupakan pengimplementasian Algoritma A\* menjadi bentuk kode kedalam bahasa pemrograman PHP.

```
function astar($start, $goal, $vertices,
$edges) {
    $openSet = [$start];
    $cameFrom = [];
    $gScore = [];
    $fScore = [];

    foreach ($vertices as $vertex =>
$coords) {
        $gScore[$vertex] = INF;
        $fScore[$vertex] = INF;
    }

    $gScore[$start] = 0;
    $fScore[$start] =
haversine($vertices[$start]['lat'],
, $vertices[$start]['lon'],
$vertices[$goal]['lat'],
$vertices[$goal]['lon']);
    while (!empty($openSet)) {
        $current = null;
        $lowestFScore = INF;
        foreach ($openSet as $vertex) {
            if ($fScore[$vertex] <
$lowestFScore) {
                $lowestFScore =
$fScore[$vertex];
                $current = $vertex;
            }
        }
    }
}
```

#### 4.4. Implementasi Basis data

Berikut dilampirkan sebuah implementasi database untuk penampungan data pada penelitian ini

#### 4.5. Tabel graph

Gambar 3. Implementasi tabel graph

Pada basisdata tersebut merupakan data yang disimpan dalam basisdata dengan data yang diperlukan yaitu nama dari kedua *vertex*, *line string* pada data line string dibutuhkan untuk membuat garis yang nantinya untuk fitur navigasi dan data jarak antar dua *vertex* yang berguna untuk membantu proses perhitungan menggunakan Algoritma A\*.

#### 4.10. Halaman utama

Gambar 7. Halaman utama aplikasi

Gambar diatas merupakan implementasi tampilan untuk halaman utama yang langsung memberikan informasi terkait sebaran kostan yang dekat dengan UMMI Sukabumi dengan radius 1km atau kurang dari UMMI.

#### 4.11. Halaman informasi

[illegible]

Gambar 8. *Pop-up* informasi kosten

Gambar diatas menunjukan ketika mengklik salah satu kostan yang berada di peta penyebaran kostan tersebut akan menampilkan informasi seperti digambar dan bisa langsung menunjukan navigasi untuk menuju ketempat dari lokasi kita saat ini atau yang sedang berlangsung.

#### 4.12. Halaman login



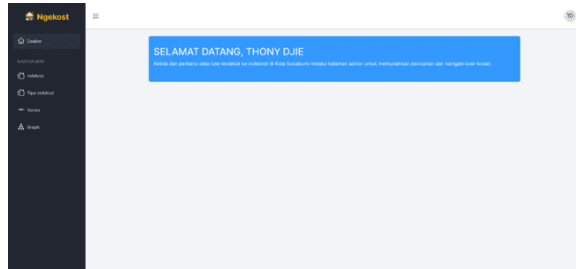
Gambar 9. Halaman Login

Gambar tersebut merupakan tampilan halaman login yang khusus admin untuk menginput data.

Gambar 9. Halaman Login

Gambar tersebut merupakan tampilan halaman login yang khusus admin untuk menginput data.

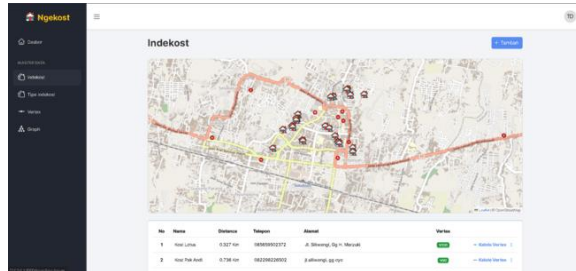
#### 4.13. Halaman utama admin



Gambar 10. Halaman utama admin

Diatas merupakan tampilan halaman utama untuk halaman admin.

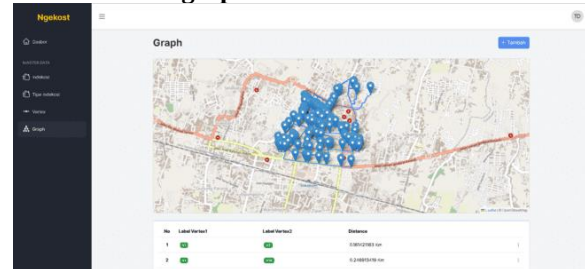
#### 4.14. Halaman Kost



Gambar 11. Halaman admin kost

Halaman tersebut merupakan halaman yang menampilkan data kosten.

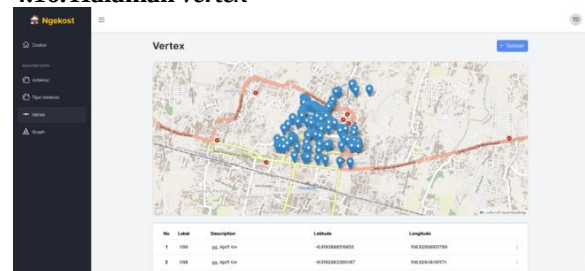
#### 4.15. Halaman graph



Gambar 12. Halaman admin graph

Halaman yang menunjukkan penhubungan per-vertex yang berguna untuk fitur navigasi.

#### 4.16. Halaman vertex



Gambar 13. Halaman admin vertex

Yang terakhir yaitu halaman *vertex* untuk mengatur titik koordinat setiap simpang jalan dan lokasi kosten berada.

#### 4.17. Pengujian

Di tahap pengujian pada penelitian ini dilakukan menggunakan metode *black-box testing* dengan melakukan pengetesan seperti yang sudah dibuat tabel skenario test berikut ini

Tabel 1. Skenario pengujian aplikasi

| No | Jenis pengujian                | deskripsi                  | Hasil yang di harapkan                                          | status |
|----|--------------------------------|----------------------------|-----------------------------------------------------------------|--------|
| 1  | Tes tampilan penyebaran kosten | <i>Functional testing</i>  | Menampilkan penyebaran kosten                                   | Lulus  |
| 2  | Tes fitur informasi kosten     | <i>UI Testing</i>          | Menampilkan informasi kosten                                    | Lulus  |
| 3  | Tes tampilan rute              | <i>UI Testing</i>          | Menampilkan rute ke tujuan yang dipilih                         | Lulus  |
| 4  | Tes integrasi ke database      | <i>Integration Testing</i> | Berhasil mengubah, mengedit dan menambah data kedalam basisdata | Lulus  |
| 5  | Tes login                      | <i>Functional Testing</i>  | Berhasil login dan sesuai dengan yang ada di basisdata          | Lulus  |

Berdasarkan pengujian dari tabel diatas aplikasi sudah berjalan sesuai dengan yang diharapkan dari tes fungsional, integrasi dan tampilan.

## 5. KESIMPULAN DAN SARAN

Pada penelitian ini setelah melalui beberapa tahapan dari identifikasi masalah, menemukan solusi, melakukan perancangan dan membangun sebuah aplikasi sehingga dapat menghasilkan aplikasi sesuai dengan ekspektasi yaitu mengadakan fitur informasi lokasi letak atau Alamat kosten dan informasi lainnya terkait kosten yang ingin dituju, selain itu fitur

navigasi dapat berjalan dengan lancar, selengkapnya dapat dilihat pada tabel skenario pengujian.

Adapun saran untuk pengembangan dari aplikasi ini yaitu menambah fitur *booking* dan penambahan *user*, untuk para mahasiswa teknik informatika UMMI Sukabumi yang ingin melanjutkan ide ini bisa dimasukan menjadi fitur pada siak UMMI agar para mahasiswa lebih mudah mengaksesnya dan bisa juga merekomendasikan agar UMMI Sukabumi dapat membuka kostannya sendiri.

## DAFTAR PUSTAKA

- [1] E. Christian Rufus, R. Rizkyaka Riyadi, D. Nugraha Hasibuan, E. Christian, V. Handrianus Pranatawijaya, and T. Nyaho Jl Yos Sudarso Palangka Raya, "Penerapan Algoritma Dijkstra Dalam Menentukan Rute Terpendek Untuk Jasa Pengiriman Barang Di Palangka Raya," *J. Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3387–3391, 2024.
- [2] M. R. Anwar and S. Purnama, "Boarding House Search Information System Database Design," *Int. J. Cyber IT Serv. Manag.*, vol. 2, no. 1, pp. 70–81, 2022, doi: 10.34306/ijcitsm.v2i1.89.
- [3] U. Schulze, "'GIS works!'—But why, how, and for whom? Findings from a systematic review," *Trans. GIS*, vol. 25, no. 2, pp. 768–804, 2021, doi: 10.1111/tgis.12704.
- [4] M. Stauffer, *Graph theory*, no. 9783319530031. 2023. doi: 10.1007/978-3-319-53004-8\_3.
- [5] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment," *IEEE Access*, vol. 9, pp. 59196–59210, 2021, doi: 10.1109/ACCESS.2021.3070054.
- [6] A. Syukur and E. Wahyono, "The Use of Flowchart in Improving Students' Ability in Writing Paragraph Corresponding Email Article's History The Use of Flowchart in Improving Students' Ability in Writing Paragraph," *Ethical Ling.*, vol. 8, no. 1, pp. 269–275, 2021.
- [7] N. Islam Hasanah and A. Sujjada, "Implementasi Formula Haversine Dalam Webgis Perguruan Tinggi Swasta Di Jawa Barat Dan Banten," *Idealis Indones. J. Inf. Syst.*, vol. 7, no. 2, pp. 178–188, 2024, [Online]. Available: <http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index>
- [8] B. Nosiyanto, N. M. Faizah, and ..., "Design of The Employee Presence Application CV. Pramana Mukti Jaya Utilizes Extreme Programming Methods and Leaflet. JS Technology," ... *Inf. Syst. ...*, vol. 1, no. 1, pp. 16–25, 2024, doi: 10.35870/jwis.v1i1.136.
- [9] M. Stauffer, *Laravel: Up & Running*. 2023.
- [10] M. D. Gustinov *et al.*, "Analysis of Web-Based E-Commerce Testing Using Black Box and White Box Methods," *Int. J. Inf. Syst. Innov. Manag.*, vol. 1, no. 1, pp. 20–31, 2023, [Online]. Available: [www.toko.ghaniib.my.id](http://www.toko.ghaniib.my.id).