



IMPLEMENTASI DIJKSTRA PADA APLIKASI PEMESANAN TIKET WISATA UNTUK REKOMENDASI LOKASI WISATA TERDEKAT

Alif Widiyanto¹, Riad Sahara², Cian Ramadhona Hassolthine³

^{1,2,3}Program Studi PJJ Informatika, Universitas Siber Asia Jakarta

Jln. Harsono RM No.1, Ragunan, Ps.Minggu, Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12550

¹alif.ti441@gmail.com, ²riadsahara@lecturer.unsia.ac.id, ³cianhassolthine@lecturer.unsia.ac.id

Abstract

Indonesia is a developing country with great potential in tourism, especially in natural and cultural attractions. Alongside the advancement of information systems, the tourism industry utilizes them to enhance interaction between tourism managers and tourists. Despite the implementation of information systems, there are still regions in Indonesia lacking information on tourist attractions, particularly regarding the distance between locations. This research aims to develop a dynamic tourism ticket booking application that can be managed by individual tourist site operators across various regions. This application utilizes the Dijkstra algorithm to recommend the nearest tourist attractions from the tourist's location. The research outcome presents several tourist attractions on a map interface, displaying the recommended sequence of locations to visit, optimized for the most efficient route from start to finish. Travelers can directly purchase digital tickets, exchangeable at their intended destinations. These tickets contain one-time usable QR codes.

Keywords : Dijkstra, Ticket, Mapbox, Web, QR Code.

Abstrak

Indonesia merupakan negara berkembang dengan potensi pariwisata yang besar, khususnya wisata alam dan budaya. Seiring dengan perkembangan sistem informasi, industri pariwisata memanfaatkannya untuk meningkatkan interaksi antara pengelola wisata dengan para wisatawan. Meskipun penerapan sistem informasi telah dilakukan, masih terdapat daerah di Indonesia yang kurang informasi objek wisata. Terutama informasi jarak lokasi wisata. Penelitian ini bertujuan untuk mengembangkan aplikasi pemesanan tiket wisata yang dinamis dan dapat dikelola oleh masing-masing pengelola objek wisata di berbagai daerah. Aplikasi ini menggunakan algoritma *Dijkstra* untuk merekomendasikan objek wisata terdekat dari titik lokasi wisatawan. Hasil penelitian ini ialah menampilkan beberapa objek wisata yang disajikan pada tampilan peta dengan menampilkan hasil rekomendasi urutan lokasi wisata dengan rute yang paling optimal dari paling awal sampai akhir untuk dikunjungi. Wisatawan dapat langsung membeli tiket secara digital yang nanti dapat ditukar ke tempat wisata yang dituju. Tiket ini berisi *Qr Code* yang dapat digunakan satu kali.

Kata kunci : Dijkstra, Tiket, Mapbox, Web, QR Code.

1. PENDAHULUAN

Seperti yang kita ketahui, Indonesia adalah salah satu negara berkembang yang memiliki banyak potensi pariwisata, terutama untuk wisata alam dan budaya karena keanekaragaman suku, adat istiadat, dan

kebudayaan serta keindahan alam dan fauna yang ditawarkan oleh iklim tropisnya [1]. Seiring berjalannya pemanfaatan sistem informasi yang ada, hal ini dimanfaatkan oleh industri pariwisata untuk meningkatkan interaksi antara pengelola wisata dengan konsumen agar lebih mudah di



akses oleh para wisatawan. Namun, informasi tentang objek wisata yang ada masih kurang di beberapa wilayah Indonesia, dalam jurnal Justin peran pariwisata kabupaten Bengkayang, Kalimantan Barat, Indonesia. Kabupaten ini memiliki banyak potensi untuk wisata, namun kurangnya informasi publik tentang objek wisata yang ada [2]. Upaya promosi sektor pariwisata yang ada di Pemkab Nganjuk masih terbatas seperti penyebaran brosur dari dinas pariwisata di pusat kota saat kirab hari jadi Pemkab Nganjuk [3]. Banyaknya objek wisata di kota Semarang menyebabkan kebingungan bagi wisatawan dalam memilih destinasi yang akan dikunjungi [4]. Adapun masalah kurangnya informasi dan perhatian terhadap beberapa objek wisata yang mungkin tidak terindeks dengan baik oleh mesin pencari [5].

Dari hasil penelitian sebelumnya masalah masalah utama yang ada adalah kurangnya informasi tentang objek wisata di daerah, peneliti sebelumnya mencoba untuk membuat solusi untuk mengatasi masalah itu semua seperti untuk menginformasikan lokasi wisata, info detail di objek wisata tersebut dan juga info estimasi biaya tiket masuk. Dari *literatur review* yang peneliti sudah rangkum kebanyakan hanya memberikan informasi mengenai objek wisata tersebut. Maka dari itu hasil penelitian dari studi literatur yang ada peneliti akan menerapkan algoritma *Dijkstra* dan mengintegrasikannya ke dalam sistem pemesanan tiket wisata yang dinamis dan dapat dikelola oleh masing-masing pengelola objek wisata di berbagai daerah, sehingga wisatawan bisa untuk langsung membeli tiket secara digital yang nantinya dapat di tukarkan ke tempat wisata yang di tuju. Tiket ini berisikan QR Code yang dapat digunakan satu kali, apabila sudah ditukarkan barcode akan tidak aktif dan tidak bisa di gunakan.

Perbedaan dari penelitian sebelumnya hanya merekomendasikan objek wisata terdekat dari satu daerah saja, maka dari itu gambaran dari permasalahan yang ada dan melatarbelakangi penelitian ini adalah belum adanya rekomendasi objek wisata sesuai titik lokasi wisatawan berada maka dari itu penelitian yang akan di lakukan ialah untuk mengcover semua daerah sesuai titik lokasi wisatawan. Nantinya pengelola wisata di setiap daerah bisa ikut serta mendaftarkan objek wisatanya pada sistem yang akan dibuat oleh peneliti.

2. TINJAUAN PUSTAKA

Pada penelitian yang dilakukan oleh peneliti-peneliti sebelumnya, seperti yang dilakukan oleh beberapa peneliti dengan menggunakan berbagai macam algoritma dan metode. Penelitian sebelum yang berjudul "Rancangan Sistem *E-itinerary* wisata di kawasan Jakarta Utara dengan penerapan Algoritma *Dijkstra*" [6]. Dengan permasalahan luasnya daerah wisata di kawasan tersebut tidak cukup di kunjungi hanya dalam satu hari. Bagi wisatawan luar kota maupun luar negeri akan kesulitan menentukan waktu yang efektif dalam menentukan wisata mana yang akan di kunjungi terlebih dahulu. Pada penelitian ini penyelesaian masalah yang ada ialah dengan pencarian lintasan terpendek dari suatu tempat ke tempat lain dengan menerapkan algoritma *Dijkstra*, hasil dari penelitian ini pengguna akan memasukan beberapa lokasi wisata yang akan di kunjungi pada sistem dan algoritma *Dijkstra* akan mencari lintasan yang memiliki panjang paling minimum.

Berikutnya, penelitian yang berjudul "Perancangan Aplikasi Rekomendasi Objek Pariwisata Berbasis Android Menggunakan Location Based Service Untuk Dinas Pariwisata Kota Pekanbaru" [7], pada saat itu Dinas Pariwisata Kota Pekanbaru memiliki situs web resmi yang berfungsi memberikan informasi tentang pariwisata Kota Pekanbaru. Website tersebut menampilkan daftar tempat wisata di Pekanbaru. Namun, informasi tentang tempat wisata yang dapat dikunjungi oleh wisatawan ditampilkan secara statis dalam bentuk teks dan tidak lengkap, sehingga agak sulit bagi wisatawan untuk mengetahui secara menyeluruh tentang objek wisata dan lokasi yang dapat wisatawan kunjungi. Penelitian yang dilakukan menghasilkan aplikasi yang akan digunakan oleh wisatawan yang mengunjungi objek wisata Kota Pekanbaru. Wisatawan akan melihat rekomendasi tempat wisata di sekitar aplikasi. Aplikasi yang dikembangkan juga dilengkapi dengan fitur rute terpendek dari posisi pengguna terhadap objek wisata yang dipilih.

Pada penelitian selanjutnya Yang berjudul "Aplikasi Pemandu Pencarian Wisata Terdekat Berbasis GIS Android Dengan Algoritma *Dijkstra*" [8], penelitian ini berfokus pada daerah di Kabupaten Sumenep. Kabupaten Sumenep memiliki daya tarik wisata alam dan budaya, sehingga menjadi tujuan para wisatawan. Tidak hanya daya tarik budaya kerapan sapinya, tetapi

juga memiliki wisata alam yang indah. Pemandangan alam, bukit tinggi, pantai, asta tinggi, dan keanekaragaman potensi kelautan ada ciri-ciri wisata alam. Lalu tujuan dari penelitian yang dilakukan adalah membuat sistem informasi geografis berbasis android yang menggunakan algoritma *Dijkstra*. Hasilnya adalah aplikasi ini dapat membantu, mempermudah, dan mempercepat pencarian lokasi wisatawan. Berdasarkan uji coba pencarian wisata terdekat menggunakan aplikasi ini menghasilkan akurasi sebesar 95%.

2.1 Algoritma *Dijkstra*

Dijkstra dijelaskan sebagai algoritma efektif untuk menentukan jarak terdekat dari satu tempat ke tempat lain[9]. Algoritma *Dijkstra* digunakan untuk menentukan rute terpendek pada graph berarah atau graph tak berarah yang memiliki bobot tanpa mengenumerasi secara spesifik semua rute yang mungkin menjadi solusi rute optimal[10]. prinsip dasar dari algoritma ini yaitu dengan mencari jarak lintasan yang terkecil. Algoritma *Dijkstra* ini di terapkan pada sebuah aplikasi *Location Based Service* dengan *platform* website yang memanfaatkan Google Map sebagai peta navigasi.

2.2 Mapbox

Mapbox adalah program atau *library* yang memungkinkan menggabungkan fitur ke dalam aplikasi seluler dengan mudah. Untuk meningkatkan pengalaman pengguna, banyak pengembang menambahkan peta ke dalam aplikasi mereka.

Salah satu kelebihan *Mapbox* adalah dapat membuat peta menggunakan teknologi teknologi sumber terbuka secara gratis, yang diberikan kepada pengembang untuk digunakan ke dalam aplikasi mereka[11].

2.3 QR Code

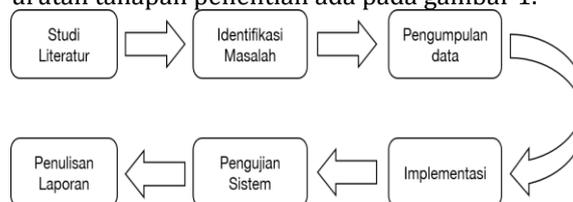
Pada penelitian sebelumnya (Dedi Susanto, 2023) merupakan jenis kode QR yang dibuat oleh Danso Wave, yang dimiliki oleh perusahaan Denso Cooperation di Jepang. Pada tahun 1994, kode QR baru diperkenalkan ke publik, Qr code sendiri merupakan generasi kedua dari kode batang atau barcode [12]. Kode QR menjadi lebih populer karena lebih mudah dan cepat digunakan daripada kode bar (kode batang). Oleh karena itu, pengaplikasian kode QR tersimpan berbagai

informasi yang dapat di intergrasikan sesuai kebutuhan pembuatnya[13].

3. METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian ini yaitu diantaranya studi pustaka, pengumpulan data, identifikasi masalah, penerapan algoritma dan implementasi urutan tahapan penelitian ada pada gambar 1.



Gambar 1. Tahapan Penelitian

Pada tahap pertama adalah membuat studi literatur. Studi literatur dibuat bersumber dari jurnal dan karya ilmiah dari penelitian sebelumnya mengenai objek yang akan diteliti. Teknik ini umum digunakan ialah mengumpulkan, mengevaluasi dan merangkum penelitian sebelumnya tentang topik yang relevan.

Pada tahap kedua, mengidentifikasi masalah dari studi literatur untuk dijadikan rumusan masalah dalam penelitian ini untuk di lakukan pengembangan.

Setelah rumusan masalah dibuat, pada tahap ketiga dilakukan pengumpulan data, proses pengumpulan data dilakukan menggunakan metode observasi untuk mengumpulkan data titik-titik koordinat lokasi wisata. Pengumpulan data koordinat menggunakan layanan yang ada pada *Google Maps*.

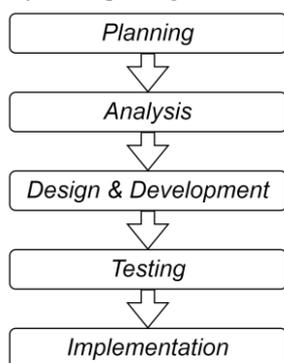
Pada tahap implementasi ialah tahapan yang meliputi pengembangan sistem yang dilakukan dengan menulis kode pemrograman tertentu.

Tahap pengujian sistem, menguji sistem yang telah dibuat untuk melihat apakah sudah sesuai dengan kebutuhan atau tidak serta menguji fitur telah berfungsi dengan baik.

3.2. Tahapan Pengembangan Sistem

Tahapan pengembangan sistem yang digunakan dalam mengembangkan aplikasi ini adalah dengan menggunakan metode *Web Development Life Cycle* (WDLC). Metode ini merupakan sebuah proses pengembangan atau perancangan dalam sebuah website dan dapat menggambarkan proses yang akan disampaikan

kepada pengguna[14]. Tahapan-tahapan dalam WDLC diantaranya ada pada gambar 2.



Gambar 2. Tahapan Pengembangan Sistem

Berikut ini adalah penjelasan dari tahapan-tahapan yang digunakan pada metode *Web Development Life Cycle* (WDLC) :

1. *Planning*

Dalam metode *Web Development Life Cycle* (WDLC), tahapan perencanaan merupakan tahapan awal dalam perancangan sebuah website. Sebelum pengembang membuat sistem, tahapan ini sangat penting. Identifikasi tujuan dan sasaran dari website yang akan dibangun adalah bagian dari tahapan ini, sehingga proses pembuatan sistem berjalan sesuai dengan rencana. Proses identifikasi ini sangat penting karena pengembang harus memahami tujuan pembuatan sistem dan memahami kriteria pengguna sistem[15].

2. *Analysis*

Pada tahap ini akan dilakukan identifikasi analisis kebutuhan apa saja yang akan digunakan dan serta menganalisa penelitian dari peneliti – peneliti terdahulu. Mulai dari data yang akan digunakan, bahasa pemrograman yang digunakan dan juga library untuk implementasi *Dijkstra*. Serta analisis fungsi dari sistem yang dibuat dengan mempertimbangkan proses yang dibutuhkan untuk mendukung fitur website [16].

3. *Design & Development*

Dalam tahap ini peneliti menyiapkan berbagai representasi diagram dari komponen fisik dan logis yang akan dibuat selama tahap pembangunan.

Seperti *use case* diagram serta mengimplementasikan ke dalam program.

4. *Testing*

Tahap *testing* merupakan Proses uji coba menunjukkan hasil pembuatan web. Tahap ini menentukan apakah situs web yang dibuat memenuhi kebutuhan pengguna, mulai dari informasi yang dibutuhkan hingga kinerja yang dihasilkan. Pada tahap ini, elemen seperti konten, fungsi, kemudahan penggunaan, dan akurasi sistem diuji. Tahap metode *testing* menggunakan *Black Box Testing*, Dalam hal ini akan dicek kelayakan secara penggunaan tanpa melihat struktur coding[17].

5. *Implementation*

Ini adalah tahapan terakhir dari metode WDLC. Di sini, website diberikan kepada pengguna dan untuk berinteraksi langsung dengan sistem. Ini memungkinkan pengguna untuk menggunakan website yang telah dibuat untuk pertama kalinya.

4. HASIL DAN PEMBAHASAN

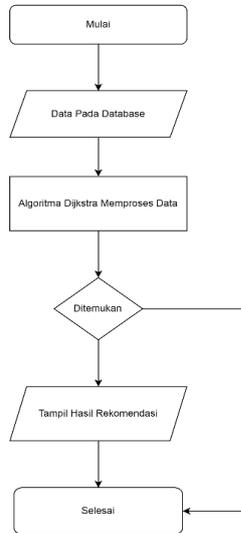
4.1 *Planning*

Pada tahap *planning* atau perencanaan ada beberapa hal yang dilakukan diantaranya

1. Menentukan maksud dari pembuatan website
2. Pahami siapa yang akan menggunakan website
3. Pahami teknologi website seperti apa yang akan digunakan dalam membangun
4. Menentukan sumber dan lokasi informasi yang akan digunakan dalam website

4.2 *Analysis*

Pengembangan menggunakan algoritma *Dijkstra* sebagai *logic* yang akan digunakan untuk menentukan rekomendasi lokasi wisata, proses ini dilakukan pada titik lokasi yang disiapkan pada database, struktur alur sistem pada gambar 2 merupakan logic sistem yang berjalan pada *platform*, dimana algoritma *Dijkstra* bekerja dengan data data yang ada pada database.



Gambar 3. Flowmap Diagram

Pada penelitian ini penulis mencoba mengumpulkan data untuk sample. Data yang dikumpulkan ialah data lokasi wisata yang terdekat dengan lokasi peneliti yang berada di kabupaten Bogor yang di representasikan dalam bentuk koordinat geografis (*latitude* dan *longitude*) serta mengumpulkan data lokasi yang nanti nya tidak termasuk ke dalam kategori rekomendasi lokasi wisata terdekat yang di tampilkan pada gambar 4.

NO	Nama Lokasi	Longitude	Latitude
1	Telaga Saat Puncak Bogor	106.99460997531408	-6.688288913284524
2	Lembang Wonderland Bandung	107.61220987301	-6.8170209224416
3	Kebu Raya Cibodas Bogor	107.00741318730165	-6.741034240109861
4	Pantai Karang Potong Cianjur	107.1678695176	-7.4676518872615
5	Taman Safari Bogor	106.95064688762552	-6.7196830877130775
6	Taman Wisata Matahari Bogor	106.9159412339	-6.6528854519356
7	Leuwiharjo Treking Sentul Bogor	106.95636710401686	-6.5915849386172845
8	Taman Langit Pangalengan Bandung	107.52641472107764	-7.231338188289899
9	Situ Cileunca Pangalengan Bandung	107.55312371315956	-7.189168213220012
10	Nimo Highland Bandung	107.57888097085807	-7.220827878283251

Gambar 4. Sample Data Lokasi

Dari data sample yang sudah dikumpulkan dilakukan proses digitasi dengan *logic* algoritma *Dijkstra*. Dengan bahasa pemrograman PHP, Javascript dan juga bantuan *library* *MapBox* untuk menampilkan graf rute lokasi.

```

<?php
// Assuming $koneksi is your database connection
$sql = mysqli_query($koneksi, "SELECT id, latitude, longitude, nama_lokasi FROM lokasi");
$locations = array();

while ($sql = mysqli_fetch_array($sql)) {
    $locations[] = array(
        'lat' => (double)$sql['latitude'],
        'lon' => (double)$sql['longitude'],
        'name' => $sql['nama_lokasi'],
        'id' => $sql['id']
    );
}
    
```

Gambar 5. Script Sisi Server

Pada *script* ini menunjukkan fungsi untuk mengambil data lokasi dari server yang kan membentuk *array* lokasi wisata yang berisi latitude, longitude.

```

// Fungsi untuk menemukan semua Lokasi terdekat dalam jarak 50 km
function findLocationsWithinDistance(currentLocation, locations, maxDistance) {
    var nearbyLocations = [];

    locations.forEach(function(location) {
        var distance = calculateDistance(currentLocation, location);
        if (distance <= maxDistance) {
            location.distance = distance;
            nearbyLocations.push(location);
        }
    });

    // Urutkan Lokasi berdasarkan jarak
    nearbyLocations.sort(function(a, b) {
        return a.distance - b.distance;
    });

    return nearbyLocations;
}

// Fungsi untuk menghitung jarak antara dua titik
function calculateDistance(point1, point2) {
    var lat1 = point1.lat;
    var lon1 = point1.lon;
    var lat2 = point2.lat;
    var lon2 = point2.lon;

    // Formula untuk menghitung jarak antara dua titik
    var R = 6371; // Radius bumi dalam kilometer
    var dLat = (lat2 - lat1) * (Math.PI / 180);
    var dLon = (lon2 - lon1) * (Math.PI / 180);
    var a =
        Math.sin(dLat / 2) * Math.sin(dLat / 2) +
        Math.cos(lat1 * (Math.PI / 180)) * Math.cos(lat2 * (Math.PI / 180)) *
        Math.sin(dLon / 2) * Math.sin(dLon / 2);
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    var distance = R * c;

    return distance;
}
    
```

Gambar 6. Script Perhitungan Jarak

Di fungsi "*findLocationsWithinDistance*" berfungsi untuk mencari lokasi-lokasi terdekat dalam jarak tertentu dari lokasi peneliti, dan pada fungsi "*calculateDistance*" ialah fungsi untuk menghitung jarak antara dua titik koordinat menggunakan formula perhitungan.

```

var locations = <?php echo json_encode($locations); >;
// Jarak maksimum dalam kilometer
var maxDistance = 50;
// Temukan semua Lokasi terdekat dalam jarak 50 km
nearbyLocations = findLocationsWithinDistance(currentLocation, locations, maxDistance);
    
```

Gambar 7. Script Implementasi Fungsi

Pada *script* diatas berfungsi untuk mengambil data array lokasi yang diambil dari dari *script* sisi server sebelumnya,

peneliti menghitung lokasi-lokasi terdekat dari lokasi peneliti saat ini dengan batas jarak maksimum 50 kilometer menggunakan fungsi "findLocationsWithinDistance".

```
// Get the user's current location using the Geolocation API
navigator.geolocation.getCurrentPosition(function(position) {
  var currentUserLocation = {
    lat: position.coords.latitude,
    lon: position.coords.longitude
  };
});
```

Gambar 8. Script Posisi Geografis Pengguna

'navigator.geolocation.getCurrentPosition': Metode ini digunakan untuk mendapatkan posisi geografis saat ini dari perangkat pengguna melalui geolocation API

```
// Create a map
var mapbox = new mapboxgl.Map({
  container: 'map',
  style: 'mapbox://styles/mapbox/streets-v11',
  center: [106.8467, -6.2088], // Centered around Jakarta
  zoom: 9 // Zoom Level for better visibility
});
```

Gambar 9. Script Peta Mapbox

Pada script gambar 7 yaitu berfungsi untuk membuat peta dengan menggunakan Library Mapbox, script yang dibuat untuk menentukan properti peta seperti container, style, pusat peta, dan zoom level.

```
nearbyLocations.forEach(function (location) {
  new mapboxgl.Marker()
    .setLngLat([location.lon, location.lat])
    .addTo(mapbox);
});
```

Gambar 10. Script Perhitungan Jarak

Menggunakan perulangan untuk iterasi melalui setiap lokasi terdekat dan membuat marker untuk setiap lokasi menggunakan Mapbox dan juga menambahkan Marker ke peta.

```
// Extract coordinates from Locations
var coordinates = nearbyLocations.map(function (location) {
  return [location.lon, location.lat];
});

// Add a Line route
mapbox.on('load', function () {
  mapbox.addSource('route', {
    'type': 'geojson',
    'data': {
      'type': 'Feature',
      'properties': {},
      'geometry': {
        'type': 'LineString',
        'coordinates': coordinates
      }
    }
  });

  mapbox.addLayer({
    'id': 'route',
    'type': 'line',
    'source': 'route',
    'layout': {
      'line-join': 'round',
      'line-cap': 'round'
    },
    'paint': {
      'line-color': '#3887be',
      'line-width': 8
    }
  });
});
```

Gambar 11. Script Rute Pada Peta

Pada variable coordinates untuk membuat kumpulan koordinat dari lokasi-lokasi terdekat dan setiap elemen array merupakan pasangan koordinat yang berisi '[longitude, latitude]'. Ketika telah dimuat ('mapbox.on('load', ...)'), script ini menambahkan rute pada peta, menentukan sumber data untuk rute dengan tipe GeoJSON yang berisi informasi tentang linestring dan koordinatnya. Lalu menambahkan layer pada peta untuk menampilkan rute tersebut.

Proses digitasi yang dilakukan dengan 10 data sample lokasi menghasilkan jarak yang dimulai dari lokasi peneliti ke beberapa lokasi yang direkomendasikan berdasarkan jarak yang sudah di tentukan, untuk hasil proses digitasi ditunjukkan pada tabel 1. Peneliti mendefinisikan sebagai (CL) yaitu titik lokasi peneliti, dan juga lokasi yang di rekomendasikan antara lain yaitu Leuwihejo treking sentul Bogor (L1), Taman wisata matahari (L2), Telaga saat puncak Bogor (L3), Taman safari Bogor (L4), Kebun raya cibodas Bogor (L5).

Tabel 1. Hasil Perhitungan Jarak

No	Lokasi 1	Lokasi 2	Jarak
1	CL	L1	15.4
2	CL	L2	17.64
3	CL	L3	25.83
4	CL	L4	26

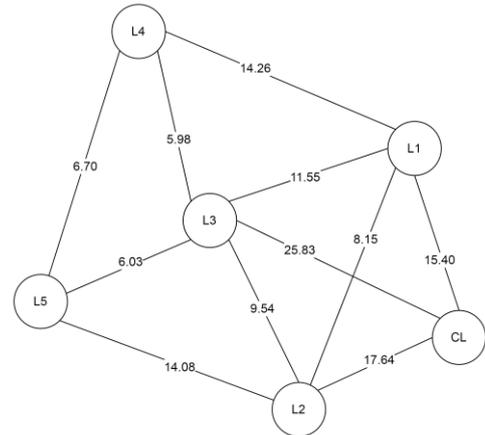
5	CL	L5	45.77
6	L1	L2	8.15
7	L1	L3	11.55
8	L1	L4	14.26
9	L1	L5	17.55
10	L2	L3	9.54
11	L2	L4	8.36
12	L2	L5	14.08
13	L3	L4	5.98
14	L3	L5	6.03
15	L4	L5	6.7

Untuk menerapkan algoritma *Dijkstra*, logika kode program digunakan untuk menentukan jika jarak lebih kecil dari jarak sebelumnya akan menghapus data lama dan menyimpan ulang jarak yang baru. Pada tabel 2 merupakan hasil dari mengikuti langkah perhitungan logika algoritma *Dijkstra*.

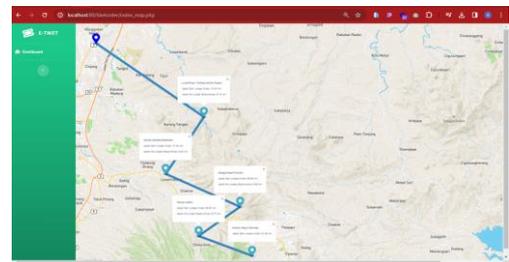
Tabel 2. Hasil Perhitungan Algoritma *Dijkstra*

Iterasi	Lokasi Tertentu	Jarak dari CL	L1	L2	L3	L4	L5
Awal	CL	0	15.4	17.6	25.8	26	45.77
1	L1	15.4	0	8.15	11.6	14.26	17.6
2	L2	17.64		0	9.54	8.4	14.1
3	L3	25.83			0	5.98	6.03
4	L4	26				0	6.7
5	L5	31.31					0

Tiap titik lokasi ditampilkan nilai bobot sesuai dengan jarak antar titik ke titik lainnya dalam kilometer, digambarkan pada gambar 10 berupa graf antara jarak dari CL ke L5 atau jarak terjauh dari lokasi yang di rekomendasikan.



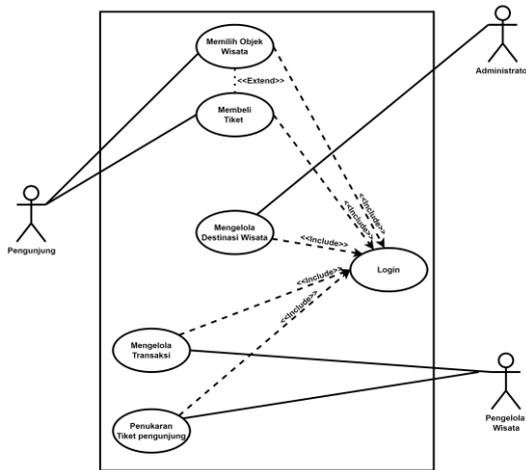
Gambar 12. Graf Analisa Algoritma *Dijkstra* Hasil rekomendasi pada halaman website disajikan pada gambar 13, dimana hasil yang di tampilkan ada jalur garis yang di rekomendasikan untuk mencapai lokasi yang terjauh. Terdapat informasi nama lokasi dan juga jarak dari lokasi peneliti dan juga jarak dari lokasi 1 ke lokasi selanjutnya.



Gambar 13. Hasil Rekomendasi Pada Website ([Link Website](#))

4.3 Design & Development

Pada tahap *Design&Development*, peneliti melakukan perancangan dan pembangunan sitem pada proses pemesanan tiket, dilakukannya identifikasi aktor yang akan berinteraksi dan berperan dalam pengoperasian sistem yang dibangun. Interaksi dari aktor aktor ini akan di gambarkan dalam *use case diagram* pada gambar 14.



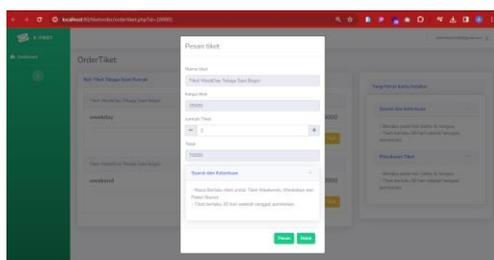
Gambar 14. Use Case Diagram

Di dalam use case diagram ditunjukkan aktor pengunjung memilih objek wisata, di activity tersebut pengunjung memilih objek wisata yang di rekomendasikan dari implementasi *Dijkstra* dengan mengklik pin lokasi yang ada pada peta. Pemilihan objek wisata di tampilkan pada halaman *Home* website yang ada pada gambar 15.



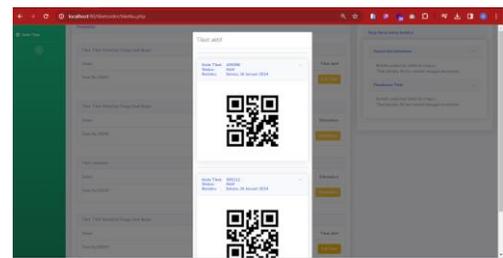
Gambar 15. Halaman Home

Setelah memilih lokasi wisata, website akan mengarahkan ke halaman pemesanan tiket. Halaman ini digunakan untuk pengunjung memesan tiket masuk objek wisata yang dipilih. Halaman pemesanan tiket ditampilkan pada gambar 16.



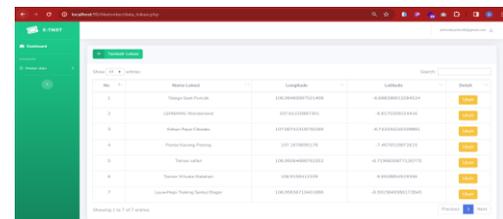
Gambar 16. Halaman Pemesanan Tiket

Pengunjung yang sudah memesan tiket dan melakukan pembayaran akan ditampilkan halaman tiket pengunjung. Yang berisi tiket aktif yang berupa *Qr Code* yang nantinya untuk ditukarkan pada aktor pengelola wisata. Halaman tiket pengunjung ditampilkan pada gambar 17.



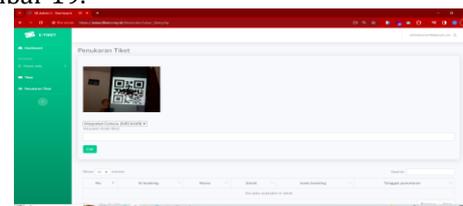
Gambar 17. Halaman Tiket Pengunjung

Pada aktor administrator, interaksi pada sistem untuk mengelola data master yang meliputi informasi data lokasi destinasi wisata dan juga data user dari masing-masing pengelola wisata. Halaman administrator ditujukan pada gambar 18.



Gambar 18. Halaman Administrator

Pada aktor pengelola wisata, interaksi pada sistem untuk mengelola transaksi dan juga melayani pengunjung yang ingin menukarkan tiket masuk wisata, pengunjung memperlihatkan *Qr code* untuk di validasi pada sistem halaman penukaran tiket. Halaman penukaran tiket ditujukan pada gambar 19.



Gambar 19. Halaman Penukaran Tiket

4.4 Testing



Pada tahapan pengujian ini menggunakan metode *Black Box Testing*. Dalam hal ini sistem diuji coba fitur dan fungsional sistem. Pengujian *Black Box Testing* untuk aktor pengunjung, administrator dan pengelola wisata ditujukan pada tabel 3.

Tabel 3. Pengujian *Black Box Testing*

No	Skenario Pengujian	Hasil
1	Pengujian aktor memasukkan "Username" dan "Password" yang benar, lalu login. Akan menampilkan halaman sesuai role aktor	Valid
2	Pengujian Aktor Pengunjung, Menampilkan rekomendasi lokasi wisata terdekat dan rute pada halaman utama website	Valid
3	Pengujian Aktor Pengunjung memilih lokasi wisata dan memesan tiket sesuai lokasi wisata yang dipilih	Valid
4	Pengujian Aktor Pengunjung menampilkan QR Code tiket wisata yang di pesan	Valid
5	Pengujian Aktor Administrator menambah, hapus, ubah master data lokasi dan user pengelola wisata	Valid
6	Pengujian Aktor Pengelola Wisata melakukan penukaran QR code tiket pengunjung	Valid

4.5 Implementation

Implementasi *Dijkstra* telah berhasil di kembangkan berbasis website. Website tersebut telah diunggah secara online pada salah satu web hosting dan dapat diakses melalui url: <https://www.lifserv.my.id/tiketorder/>.

5. KESIMPULAN

Pada hasil penelitian ini mengenai implementasi *Dijkstra* pada aplikasi pemesanan tiket wisata untuk rekomendasi lokasi wisata terdekat, peneliti menarik kesimpulan diantaranya :

1. Aplikasi yang dibangun berbasis website dengan bahasa pemrograman php versi 7 dan menggunakan database MYSQL versi 7.4.6, dan juga menggunakan *library MapBox* untuk mendukung implementasi *Dijkstra*.
2. Lokasi wisata yang di rekomendasikan adalah lokasi wisata dengan jarak 50 km dari lokasi pengguna.
3. Hasil rekomendasi lokasi wisata berupa graf urutan lokasi wisata yang di rekomendasikan untuk di kunjungi, dan juga terdapat informasi jarak dari lokasi 1 ke lokasi selanjutnya.
4. Hasil uji dengan *Black Box Testing*, sesuai dengan kebutuhan sistem sesuai akses aktor pengguna.
5. Tiket masuk wisata berupa Qr Code yang hanya dapat digunakan 1 kali, setelah ditukar tiket menjadi tidak aktif.

DAFTAR PUSTAKA:

[1] U. Munir, "Konsep Kebijakan Pariwisata Berbasis Kerakyatan (Studi Pariwisata Di Pulau Lombok)," *Yust. MERDEKA J. Ilm. Huk.*, vol. 8, no. 2, pp. 125–134, 2022, doi: 10.33319/yume.v8i2.199.

[2] S. P. G. Setiawan, H. Sujaini, and M. A. Irwansyah, "Sistem Informasi Objek Wisata dengan Algoritma Djisktra untuk Rute Terdekat dan Metode Analytical Hierarchy Process (AHP) untuk Rekomendasi. (Studi Kasus Kabupaten Bengkayang)," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 2, p. 191, Apr. 2020, doi: 10.26418/justin.v8i2.36804.

[3] A. K. Amrulloh *et al.*, "Perancangan Nganjuk E-Tourism Berbasis Android Dengan Algoritma Dijkstra Untuk Pencarian Jalur Terdekat Objek Pariwisata Abstrak," *Inov. J. Ilm. Inov.*, pp. 65–70, 2019, [Online]. Available:



- <https://ejournal.unhasy.ac.id/index.php/inovate/article/download/749/603>
- [4] B. A. Herlambang, S. O. Pradana, and R. R. Waliyansyah, "Sistem Informasi Geografis Pencarian Rute Optimum Lokasi Obyek Wisata Kota Semarang Menggunakan Algoritma Dijkstra," *J. Appl. Sci. Technol.*, vol. 2, no. 01, p. 16, 2022, doi: 10.30659/jast.2.01.16-25.
- [5] B. A. Wardana, Z. Sari, and D. R. Akbi, "Aplikasi Rekomendasi Lokasi Pariwisata Menggunakan Metode Profile Matching," *J. Repos.*, vol. 2, no. 6, p. 727, 2020, doi: 10.22219/repositor.v2i6.751.
- [6] D. Kawasan, J. Utara, D. Penerapan, A. Dijkstra, D. Widyaningrum, and C. Danuputri, "Rancangan Sistem E-itinerary Wisata Di Kawasan Jakarta Utara Dengan Penerapan Algoritma Dijkstra," *Jl. Lodan Raya*, vol. 12, no. 2, p. 6909090, 1443, [Online]. Available: <http://jurnal.umt.ac.id/index.php/jt/index>
- [7] J. Alif, I. Surya, and K. Diah, "Perancangan Aplikasi Rekomendasi Objek Pariwisata Berbasis Android Menggunakan Location Based Service Untuk Dinas Pariwisata Kota Pekanbaru," no. 28265, 2022, [Online]. Available: <https://ejournal.upbatam.ac.id/index.php/jif/article/view/4666/2533>
- [8] K. Anam and O. D. Hartono, "Aplikasi Pemandu Pencarian Wisata Terdekat Berbasis GIS Android Dengan Algoritma Dijkstra," *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 3, no. 1, p. 91, 2019, doi: 10.30645/j-sakti.v3i1.99.
- [9] T. Arta and D. Victoria, "BULLETIN OF COMPUTER SCIENCE RESEARCH Penerapan Algoritma Dijkstra dalam Pemetaan UMKM Berbasis Android," vol. 3, no. 6, pp. 420-426, 2023, doi: 10.47065/bulletincsr.v3i6.276.
- [10] I. P. W. Gautama and K. Hermanto, "Penentuan Rute Terpendek dengan Menggunakan Algoritma Dijkstra pada Jalur Bus Sekolah," *J. Mat.*, vol. 10, no. 2, p. 116, 2020, doi: 10.24843/jmat.2020.v10.i02.p128.
- [11] R. Hidayatulloh and P. Airlangga, "Integrasi Mapbox dan Google Map untuk Menunjang Fitur tambahan pada system informasi geografis," *Exact Pap. Compil.*, vol. 4, no. 1, pp. 491-496, 2022, doi: 10.32764/epic.v4i1.648.
- [12] D. Susanto, Ahmad Rabiul Muzammil, and Heriansyah, "Perancangan Aplikasi E-Voting Qr-Code Login Berbasis Web Muhammadiyah Kalimantan Barat," *J. Manaj. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 181-189, 2023, doi: 10.36595/misi.v6i2.837.
- [13] J. P. Masyarakat, "Pembimbingan Dalam Pembuatan Qr Code Kepemilikan Berbasis Web Kepada Organisasi," vol. 2, no. 1, pp. 81-88, 2023.
- [14] Suwah Yudianto and Wiwin Sulisty, "It-Explore Pengembangan Web Portal Dengan Metode Web Development Life Cycle (Wdlc) Pada Dinas Komunikasi Dan Informatika Kabupaten Bengkayang," vol. 01, pp. 145-154, 2022.
- [15] J. R. Permana, M. Muhaqiqin, and A. S. Puspaningrum, "Implementasi Metodologi Web Development Life Cycle Untuk Membangun Sistem Perpustakaan Berbasis Web (Studi Kasus: Man 1 Lampung Tengah)," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 2, no. 4, pp. 435-446, 2021, doi: 10.33365/jatika.v2i4.1407.
- [16] J. I. Informatika, R. Kaban, M. Informatika, and P. P. Medan, "PENGEMBANGAN SISTEM INFORMASI PERPUSTAKAAN DENGAN," vol. 2, no. 1, pp. 83-89, 2017.
- [17] R. Setyadi, yesaya tommy Paulus, L. Mutawalli, S. Dwiasnati, ida bagus ary indra Iswara, and erlin windia Ambrasari, "Jurnal Manajemen Informatika & Sistem Informasi (MISI) jurnal Manajemen Informatika nformatika & Sistem Informasi (MISI) ISSN : 2614-1701 (Cetak) - 2614-3739 (Online) ii," vol. 5, p. 29, 2022, [Online]. Available: https://www.google.com/search?q=jurnal+misi+stmik+lombok&rlz=1C1CHBF_enID993ID993&oq=jurn&aqs=chrome.0.69i59j69i57j0i131i433i512j0i433i512l2j69i60l2j69i61.12564j0j7&sourceid=chrome&ie=UTF-8