

Rekomendasi Angkutan Umum

Public Transport Recommendation

AINIL MARDHATILLAH¹, NABIL HAMZAH ASH-SHIDDIQ², RIZKICA DEVIYANTI³, VIRAGITA ATHAYA HAURA⁴

Abstrak

Perjalanan antar kota dengan jarak yang jauh pastinya membutuhkan waktu yang banyak dan dapat menjadi tantangan tersendiri bagi banyak orang. Tak sedikit yang memilih untuk menggunakan angkutan umum sebagai sarana transportasi utama. Tak sedikit yang memilih untuk menggunakan angkutan umum sebagai sarana transportasi utama. Oleh karena itu, dirancang suatu program manajemen rekomendasi angkutan umum yang dapat memberikan informasi terkait angkutan umum yang paling tepat bagi pengguna. Ruang lingkup penelitian ini terbatas pada implementasi struktur data algoritma dijkstra, graph, array dan vektor, queue, dan unordered map (hash map) dalam pembuatan program. Program ini memungkinkan pengguna mendapatkan informasi harga berdasarkan jenis kendaraan, serta menampilkan seluruh data yang tersimpan dengan format yang rapi dan mudah dibaca. Dengan demikian, program *Public Transport Recommendation* dapat membantu pengguna angkutan umum untuk memilih kendaraan dengan jarak dan waktu tempuh yang efisien.

Kata Kunci: dijkstra, graf, jarak, perjalanan, queue, transportasi umum

Abstract

Travelling across cities with long distances is definitely a long time and can be a personal challenge for many people. Not a few have chosen to use public transportation as their primary means of transportation. Therefore, a general transport recommendation management program has been designed that can provide the most appropriate public transport information to the user. The scope of this research is limited to the implementation of the data structures of the algorithm dijkstra, graph, array and vector, queue, and unordered maps (hash maps) in program creation. The program allows users to obtain price information based on type of vehicle, as well as display all stored data in a neat and easy-to-read format. Thus, the Public Transport Recommendation program can help public transport users to choose vehicles with efficient distances and travel times.

Keywords: dijkstra, distance, graph, public transport, queue, travel

PENDAHULUAN

Perjalanan antar kota dengan jarak yang jauh pastinya membutuhkan waktu yang banyak dan dapat menjadi tantangan tersendiri bagi banyak orang. Sebuah kota tidak akan bisa hidup tanpa angkutan umum, bagaimanapun tidak semua masyarakat mampu memenuhi kebutuhannya sendiri pasti ada yang bergantung pada angkutan umum untuk memenuhi kebutuhan mobilitasnya dengan alasan tidak ada atau tidak mempunyai kendaraan pribadi (Haqie *et al.* 2020). Tak sedikit yang memilih untuk menggunakan angkutan umum sebagai sarana transportasi utama. Adapun alasan lainnya seperti, keinginan untuk menghindari kemacetan serta faktor kenyamanan dan efisiensi. Oleh karena itu, angkutan umum menjadi solusi utama yang banyak dipertimbangkan. Dalam memilih angkutan umum, orang cenderung mempertimbangkan aksesibilitasnya. Aksesibilitas sering dikaitkan dengan jarak, waktu tempuh dan biaya perjalanan (Riawan & Ahyudanari, 2020). Efisiensi waktu sangat dipengaruhi oleh rute perjalanan yang optimal dan frekuensi keberangkatan, sementara

¹Ainil Mardhatillah, G6401221045, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia, ainilmardhatillah@apps.ipb.ac.id;

²Nabil Hamzah Ash-Shiddiq, G6401221901, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia, nbilhznabil@apps.ipb.ac.id;

³Rizkica Deviyanti, G6401221007, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia, rizkikadeviyanti@apps.ipb.ac.id;

⁴Viragita Athaya Haura, G6401211116, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia, viragitaathayahaura@apps.ipb.ac.id;

keterjangkauan biaya berkaitan erat dengan jenis kendaraan yang digunakan, seperti bus, kereta api, atau angkutan massal lainnya. Dengan mempertimbangkan kedua aksesibilitas ini, angkutan umum tidak hanya menawarkan alternatif yang praktis, tetapi juga solusi yang ekonomis bagi para pelaku perjalanan jarak jauh.

Dalam upaya mencapai tujuan tersebut, penerapan ilmu teknologi dan struktur data sangat penting. Proyek ini mengimplementasikan struktur data untuk membuat sistem manajemen rekomendasi angkutan umum yang memberikan rute dan angkutan umum yang optimal berdasarkan posisi asal dan tujuan pengguna. Terdapat beberapa fitur fungsi dalam program, yaitu: fungsi untuk menerima lokasi asal dan tujuan pengguna, fungsi untuk menentukan rute terdekat dan angkutan umum yang dapat dinaiki, fungsi untuk menyajikan informasi jadwal dan ketersediaan angkutan umum, fungsi untuk memberikan rute alternatif, serta fungsi untuk penambahan, pencarian, dan penghapusan data angkutan umum. Untuk menentukan rute terdekat, digunakan algoritma Dijkstra yang bertujuan untuk menemukan lintasan terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya (Budihartono 2016). Pada fitur penjadwalan, menggunakan struktur data queue dengan algoritma FIFO, penggunaan algoritma ini dimaksudkan jadwal tetap pada urutan yang benar tanpa merubah urutan yang sudah ada (Gunawan *et al.* 2023). Struktur data seperti vektor, graph, dan map juga digunakan dalam sistem program. Struktur data map digunakan untuk mengatur status pengguna sebagai admin atau user dalam program. Setiap pengguna memiliki kunci unik yang dipetakan ke nilai yang menentukan statusnya. Dengan penggunaan struktur data yang tepat dalam proses pemrograman akan menghasilkan sebuah program yang secara keseluruhan menjadi lebih sederhana (Pratama 2020). Hal ini memungkinkan sistem program untuk mengolah informasi dengan lebih cepat dan akurat, serta mengurangi waktu yang diperlukan untuk pencarian dan pengambilan keputusan. Dengan demikian, tujuan dari proyek ini dapat tercapai guna memudahkan pengguna dalam mendapatkan rekomendasi angkutan umum dengan jarak dan waktu yang efisien.

METODE

Ruang lingkup penelitian ini terbatas pada implementasi struktur data algoritma dijkstra, graph, array dan vektor, queue, dan unordered map (hash map). Struktur data tersebut bertujuan untuk mengelola data angkutan umum, rute antar kota, jadwal keberangkatan, dan autentikasi pengguna.

Algoritma Dijkstra

Algoritma Dijkstra menurut Edsger W. Dijkstra sendiri merupakan algoritma yang bertujuan untuk menentukan jalur terpendek antara dua titik dalam graf. Cara kerja algoritma ini untuk menemukan jalur yang paling efektif dan efisien dengan melakukan perbandingan bobot terkecil dari node awal (titik keberangkatan) ke node terakhir (titik tujuan). Fungsi dimulai dengan menginisialisasi jarak ke semua simpul dengan nilai maksimum ('INT_MAX_VAL'), kecuali untuk kota awal ('start_city') yang disetel ke 0 dan menggunakan priority queue untuk memproses simpul dengan jarak terpendek. Selama antrian tidak null, maka simpul dengan jarak terpendek akan diproses dan jika simpul tersebut adalah kota tujuan, maka proses dihentikan. Pada program ini, fungsi utama untuk algoritma Dijkstra adalah 'dijkstra' yang bertujuan untuk mengimplementasikan keseluruhan logika pencarian jalur terpendek. Fungsi pendukung seperti 'get_vertex_index' ditambahkan untuk mendapatkan indeks simpul dari nama kota.

```
PathResult dijkstra(const string& start_city, const string& end_city) {
    int start_index = get_vertex_index(start_city);
    int end_index = get_vertex_index(end_city);
    vector<int> dist(cities.size(), INT_MAX_VAL);
    dist[start_index] = 0;
```

```

priority_queue<Node, vector<Node>, greater<Node>> pq;
pq.push({0, start_index});
while (!pq.empty()) {
    int u = pq.top().second;
    pq.pop();
    if (u == end_index) break;
    for (int v = 0; v < cities.size(); ++v) {
        if (adj_matrix[u][v] != 0) {
            int weight = adj_matrix[u][v];
            if (dist[u] + weight < dist[v]) {
                dist[v] = dist[u] + weight;
                pq.push({dist[v], v});
            }
        }
    }
}
// Build and reverse the route
for (int v = end_index; v != -1; v = parent[v]) {
    result.route.push_back(cities[v]);
}
reverse(result.route.begin(), result.route.end());
return result;
}

```

Graph

Graph merupakan himpunan simpul (*vertices/nodes*) dan himpunan sisi (*edges*) yang merupakan himpunan beranggotakan dua elemen dari simpul (Cormen *et al.* 2009). Graph ini bertujuan untuk merepresentasikan jaringan kota dan rute antar kota dengan biaya dan menggunakan adjacency matrix untuk menyimpan koneksi antar kotanya. Pada bagian 'private' dari kelas 'Graph', terdapat deklarasi variabel 'vector<vector<int>> adj_matrix;' yang merepresentasikan adjacency matrix untuk graf. Melalui kode 'adj_matrix.resize(100, vector<int>(100, 0));' menunjukkan ketika objek 'Graph' dibuat maka adjacency matrix akan diinisialisasi dengan ukuran 100 x 100 yang berarti maksimal terdapat 100 kota dalam graf. Kemudian terdapat 'void add_edge' yang bertujuan untuk menambahkan sebuah edge antara dua kota dalam graf dengan biaya tertentu. Ketika edge ditambahkan, biayanya akan disimpan dalam adjacency matrix.

Array dan Vektor

Array atau larik adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama. Menurut definisinya, array adalah sebuah variabel yang dapat menyimpan lebih dari satu nilai sejenis dengan tipe data yang sama. Suatu array mempunyai jumlah komponen yang banyaknya tetap. Banyaknya komponen dalam suatu array ditunjukkan oleh suatu indeks yang disebut tipe indeks (Hindriani *et al.* 2014). Sedangkan vektor merupakan array dinamis yang ukurannya dapat disesuaikan secara dinamis. Array dan vector memiliki peran krusial dalam implementasi program ini. Array digunakan untuk menyimpan adjacency matrix di dalam kelas Graph. Adjacency matrix, yang direpresentasikan oleh 'adj_matrix' sebagai array dua dimensi, menunjukkan hubungan antarkota dalam graf. Setiap elemen array mewakili relasi antara dua kota yang terhubung melalui jalur tertentu. Array ini telah ditetapkan dengan ukuran maksimum 100 kota, mengindikasikan batasan pada jumlah kota yang dapat direpresentasikan.

Sedangkan, vektor digunakan untuk menyimpan daftar kota dalam kelas 'Graph', diwakili oleh variabel 'cities'. Nama-nama kota disimpan dalam vektor ini, dan karena sifatnya yang dinamis, vektor dapat berkembang seiring penambahan kota baru ke dalam graf. Vektor menjadi pilihan yang lebih fleksibel daripada array karena memungkinkan ukurannya disesuaikan secara dinamis, tanpa batasan ukuran yang telah ditetapkan sebelumnya. Selain

itu, terdapat pula vektor 'Kendaraan', yang terisi oleh objek graf untuk setiap jenis kendaraan yang ada, baik bus maupun kereta.

Queue

Queue atau antrian adalah suatu jenis struktur data yang dapat diproses dengan sifat FIFO (First In First Out), dimana elemen yang pertama kali masuk ke antrian akan keluar pertama kali. Ada dua jenis operasi yang bisa dilakukan di antrian: enqueue (memasukkan elemen baru ke dalam elemen) dan dequeue (adalah mengeluarkan satu elemen dari suatu antrian) (Gunawan *et al.* 2023). Queue dalam program ini digunakan sebagai struktur data untuk menyimpan jadwal keberangkatan kendaraan seperti bus dan kereta. Dalam konteks ini, queue dimanfaatkan untuk berbagai tujuan. Pertama, queue digunakan untuk menambahkan jadwal baru dengan menggunakan fungsi 'tambahkanJadwal', yang menerima parameter berupa queue dan jadwal baru yang ingin ditambahkan. Selanjutnya, queue juga digunakan untuk menghapus jadwal yang sudah ada dengan menggunakan fungsi 'hapusJadwal', yang juga menerima parameter berupa queue. Terakhir, queue digunakan untuk menampilkan jadwal keberangkatan kendaraan melalui fungsi 'tampilkanJadwal', yang menerima parameter berupa queue dan menampilkan jadwal-jadwal tersebut. Dengan demikian, queue menjadi elemen kunci dalam pengelolaan jadwal keberangkatan kendaraan dalam program tersebut.

Unordered Map

Unordered map menurut Stroustrup (2013) dan Sutter (2000), merupakan struktur data yang efisien dan mudah untuk digunakan, yang disediakan oleh STL untuk menyimpan pasangan key-value. Struktur data ini sangat cocok untuk case di mana kecepatan akses dan manipulasi data menjadi hal yang penting. Dengan mengimplementasikan tabel hash, 'unordered_map' memastikan bahwa operasi-operasi seperti penambahan, pencarian dan penghapusan dapat dilakukan dengan waktu rata-rata $O(1)$. Tidak seperti 'map', yang mengatur elemen-elemen dalam urutan tertentu berdasarkan kunci, 'unordered_map' tidak memastikan urutan tertentu pada elemen yang disimpannya.

Pada kelas 'UserManager', 'unordered_map' digunakan untuk menyimpan informasi pengguna yang terdaftar. deklarasinya adalah 'unordered_map<string, User> users;'. Saat pengguna baru ditambahkan ke 'unordered_map' ini, maka pada saat login, 'unordered_map' akan memeriksa apakah nama pengguna tersedia dan memverifikasi kata yang diberikan. Sedangkan, pada kelas 'Graph', 'unordered_map' digunakan untuk memetakan nama kota ke indeks dalam adjacency matrix. Struktur data ini membantu mengelola graf yang berisi kota-kota sebagai simpul dan rute antar kota sebagai edges ('unordered_map<string, int> vertex_map;'). Pada kelas 'Graph' tersebut 'unordered_map' memungkinkan penambahan kota baru dan pemetaan nama kota ke indeks secara cepat. Ketika sebuah kota baru ditambahkan ke graf, kota tersebut akan dimasukkan ke 'vertex_map' untuk mempermudah akses ke indeks terkait adjacency matrix.

Int Main

Fungsi 'main' menjadi titik masuk dari program. Fungsi ini yang bertugas memanggil 'session()' yang merupakan loop utama dari aplikasi. Pada loop tersebut, program akan menampilkan menu pilihan untuk pengguna yang belum login agar dapat mendaftar maupun login. Setelah login, pengguna dapat menambahkan jadwal, menghapus jadwal, menampilkan rute, menampilkan jadwal atau logout, hal ini bergantung pada status login mereka apakah sebagai admin atau pengguna biasa (user). Fungsi 'session' juga digunakan untuk memisahkan tugas administrasi (seperti menambahkan dan menghapus jadwal) dan tugas pengguna (seperti menampilkan rute dan jadwal), serta memelihara sesi pengguna saat ini. Hal ini untuk memastikan bahwa hanya admin yang dapat melakukan operasi-operasi kritis/penting sementara pengguna biasa hanya dapat melihat informasinya saja.

```

int main() {
    session();
    return 0;
}

```

HASIL DAN PEMBAHASAN

Pengembangan proyek struktur data *Public Transport Recommendation* melibatkan program yang cukup kompleks sehingga memerlukan perhatian yang mendalam dalam berbagai aspek. Beberapa aspek yang menjadi fokus utama pada proyek ini antara lain manajemen data yang melibatkan pengelolaan informasi tentang jadwal perjalanan, rute antar kota dan pengguna. Hal ini mencakup penyimpanan data yang tepat dan penggunaan struktur data yang efisien. Selain itu, implementasi algoritma yang tepat juga sangat berperan penting agar dapat memproses dan menganalisis data dengan efisien. Dalam proyek ini algoritma Dijkstra digunakan untuk membantu menemukan rute terpendek antara dua kota, sedangkan algoritma pencarian yang efisien digunakan dalam menentukan destinasi wisata yang sesuai dengan preferensi dari pengguna. Kemudian, aspek penting lainnya adalah interaksi pengguna, di mana pengguna memiliki kemampuan untuk menerima input, baik dalam hal registrasi, login, atau penambahan jadwal perjalanan. Jadi, program harus dapat berinteraksi dengan pengguna secara baik dan memberikan respons yang tepat sesuai dengan tindakan yang diambil pengguna. Dalam konteks manajemen pengguna, keamanan data pengguna juga menjadi fokus utama, sebuah program harus mampu melindungi informasi sensitif seperti kata sandi pengguna dan mencegah akses yang tidak sah. Fleksibilitas dan skalabilitas pun tidak kalah pentingnya, karena proyek ini dapat digunakan oleh banyak pengguna sehingga program harus dirancang dengan fleksibilitas dan skalabilitas yang memadai. Hal ini termasuk kemampuan untuk menambahkan dan menghapus jadwal perjalanan dengan mudah.

Program pada proyek ini juga dirancang untuk dapat membuat pengguna mengetahui biaya perjalanan. Sebuah program yang mampu untuk memberikan informasi mengenai biaya, terutama biaya transportasi ke suatu lokasi, akan sangat berguna bagi wisatawan dalam merencanakan perjalanan mereka dengan lebih baik. Menurut Parassa *et al.* (2018) dengan akses ke perkiraan biaya transportasi, wisatawan dapat mengelola anggaran mereka dengan lebih efisien sehingga mereka dapat menyiapkan dana yang dibutuhkan dan membuat keputusan yang tepat saat memilih rute kota tujuan yang sesuai dengan kondisi finansial mereka. Kode program yang kami analisis ini mencakup juga implementasi dari berbagai fitur yang relevan dengan pengelolaan jadwal dan rute angkutan umum. Sistem yang bangun meliputi fitur registrasi pengguna, manajemen jadwal keberangkatan, dan pemetaan rute antarkota. Sistem juga memungkinkan untuk menambahkan dan menghapus jadwal perjalanan, serta menampilkan jadwal rute berbagai jenis kendaraan umum seperti bus dan kereta api.

	jakarta	bandung	semarang	surabaya
jakarta	0	30	40	0
bandung	30	0	0	0
semarang	40	0	0	50
surabaya	0	0	50	0

Gambar 1 Output untuk show route berdasarkan jarak

```

Pilih Kota Asal:jakarta
Pilih Kota Tujuan:
surabaya
Best path: jakarta -> semarang -> surabaya
Total price/distance: 90

```

Gambar 2 Output untuk rute terdekat

```

Jadwal Bus:
Tidak ada jadwal yang tersedia

Jadwal Kereta:
=====
Asal Tujuan Keberangkatan Kedatangan Harga
=====
jakarta bandung 09.00 10.00 Rp 3000
=====
Logged in as: admin

```

Gambar 3 Tabel jadwal transportasi dan biaya

Public Transport Recommendation ini dirancang dengan tampilan interface yang mudah dibaca dan digunakan, sehingga diharapkan para pengguna dapat dengan mudah memilih transportasi dengan rute yang diinginkan. Pengguna dapat melihat tabel rute kendaraan dan tabel untuk jadwal transportasi. Ketika pengguna login dan memilih menu 'show route', maka program *Public Transport Recommendation* akan mengolah data menggunakan struktur data yang sesuai untuk mengumpulkan data yang berkaitan dengan informasi biaya transportasi, jenis transportasi, dan faktor-faktor lainnya yang bersangkutan. Setelah itu, berdasarkan informasi-informasi yang telah dikelola, program akan memberikan biaya transportasi untuk setiap transportasi untuk perjalanan dari kota asal ke kota yang dituju. Sedangkan untuk melihat tabel yang berisikan informasi terkait jam keberangkatan dan jam kedatangan, pengguna dapat memilih menu 'show schedule'.

Program rekomendasi angkutan ini memiliki kelebihan untuk dapat memberikan estimasi jam keberangkatan dan kedatangan serta dilengkapi dengan biaya transportasi yang dapat membantu pengguna dalam mengatur jadwal dan anggaran perjalanan mereka dengan mudah. Namun pada implementasinya terdapat beberapa bagian yang dapat ditingkatkan dalam pengembangan sistem angkutan umum ini. Salah satunya adalah pengembangan lebih lanjut terhadap manajemen pengguna dan otorisasi. Karena dalam implementasinya, sistem saat ini hanya membedakan antara pengguna biasa dan administrator, tanpa memberikan detail lebih lanjut mengenai peran dan hak akses masing-masing pengguna. Pengembangan lebih lanjut dalam hal tersebut dapat meningkatkan keamanan dan fleksibilitas pengguna sistem.

Pada program struktur data ini menggunakan queue dan algoritma Dijkstra yang mana program ini dirancang untuk mengoptimalkan perjalanan dengan mencari rute terpendek antara dua titik dalam suatu graf yang terhubung. Program ini menggunakan struktur data queue untuk mengatur antrian destinasi yang akan dikunjungi dan algoritma Dijkstra untuk mencari rute terpendek. Informasi tambahan yang mengindikasikan jarak atau antar destinasi dapat divisualisasikan menggunakan graf (Purwananto *et al.* 2005). Dengan demikian, pengguna dapat dengan mudah melihat dan membandingkan jarak antara destinasi wisata dan merencanakan perjalanan mereka dengan lebih efisien.

Algoritma Dijkstra pada kode ini digunakan untuk menemukan jalur terpendek dan harga termurah antara dua kota dalam graf yang mewakili rute perjalanan antarkota. Pertama, algoritma menginisialisasi matriks adjacency untuk merepresentasikan keterhubungan antar kota dengan memberi bobot pada setiap edge yang menghubungkan kota. Kemudian, dimulai dari kota asal, algoritma menghitung jarak terpendek ke setiap kota lainnya dengan mempertimbangkan bobot edge antar kota. Proses ini dilakukan dengan menggunakan pendekatan Greedy, di mana algoritma selalu memilih kota dengan jarak terpendek yang telah diketahui sebagai kandidat berikutnya untuk dieksplorasi. Selama eksplorasi, algoritma memperbarui jarak terpendek ke setiap kota yang terhubung dengan kandidat saat ini, jika jarak baru yang dihitung lebih kecil dari jarak yang telah diketahui sebelumnya. Proses ini berlanjut hingga semua kota dieksplorasi dan jarak terpendek dari kota asal ke tujuan terakhir ditemukan. Hasilnya adalah jalur terpendek antara kedua kota tersebut, yang direpresentasikan sebagai serangkaian kota yang harus dilalui serta total bobot atau jaraknya.

SIMPULAN

Public Transport Recommendation antar kota dapat memudahkan pengguna dalam mendapatkan informasi mengenai rute dan jadwal angkutan umum. Program ini memungkinkan pengguna mendapatkan informasi harga berdasarkan jenis kendaraan, serta

menampilkan seluruh data yang tersimpan dengan format yang rapi dan mudah dibaca. Dengan demikian, pengguna dapat membuat keputusan yang lebih baik terkait pemilihan angkutan umum. Sistem program ini tidak hanya meningkatkan efisiensi manajemen data tetapi juga memberikan kemudahan akses informasi yang sangat penting untuk perencanaan perjalanan yang optimal.

DAFTAR PUSTAKA

- Budihartono E. 2016. Penerapan Algoritma Dijkstra Untuk Sistem Pendukung Keputusan Bagi Penentu Jalur Terpendek Pengiriman Paket Barang Pada Travel. Mataram: Politeknik Harapan Bersama.
- Dijkstra EW. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, (1) : 269-271. doi:10.1007/BF01386390.
- Gunawan R, Yuana H, Kirom S. 2023. Metode Queue Pada Sistem Antrian Online Berbasis Web Studi Kasus UPTD Puskesmas Sananwetan. 7(3):1539 . Blitar: Universitas Islam Balitar Blitar.
- Hindriani N, Narwen, Yozza H. 2014. Implementasi Antrian Dengan menggunakan Array. 3(4):147-151.
- Haqie ZA, Nadiyah RE, Ariyani OP. 2020. Inovasi Pelayanan Publik Suroboyo Bis di Kota Surabaya. 5(1). doi:10.26740/jpsi.v5n1.p23-30.
- Parassa Y, Pesik MU, Bijang NL. 2018. Model Sistem Informasi Biaya Kunjungan Wisata Di Provinsi Sulawesi Utara. *Journal of Applied Informatics and Computing*. 2 (2) : 34-37.
- Pratama MA. 2020. Struktur data array dua dimensi pada pemrograman C++. OSF Preprints [Internet]. 04 April 2020 ; [diunduh 2023 Mei 30]; doi:10.31219/osf.io/vyech.
- Riawan AR, Ahyudanari E. 2020. Analisis Pengaruh Biaya dan Waktu Tempuh Terhadap Aksesibilitas Angkutan Umum Menuju Stasiun KRL, LRT, dan BRT di Kota Bekasi. 22(2):143-156. doi:10.25104/jptd.v22i2.1593.
- Stroustrup B. 2013. The C++ Programming language (4th ed.). Addison-Wesley. Sutter H. 2000. Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solustions. Addison-Wesley.