

**KOMBINASI ALGORITMA ELECTRE DAN *DIJKSTRA* DALAM  
SISTEM PEMILIHAN *GUESTHOUSE* DI KOTA MEDAN**

**SKRIPSI**

**WILBERT CAPRI**

**181401050**



**PROGRAM STUDI S-1 ILMU KOMPUTER**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS SUMATERA UTARA**

**MEDAN**

**2023**

**KOMBINASI ALGORITMA ELECTRE DAN *DIJKSTRA* DALAM SISTEM  
PEMILIHAN *GUESTHOUSE* DI KOTA MEDAN**

**SKRIPSI**

**WILBERT CAPRI**

**181401050**



**PROGRAM STUDI S-1 ILMU KOMPUTER**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS SUMATERA UTARA**

**MEDAN**

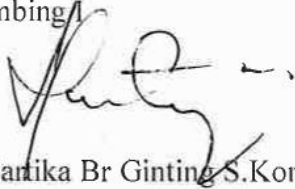
**2023**

## PERSETUJUAN

Judul : Kombinasi Algoritma Electre dan Dijkstra dalam  
Sistem Pemilihan *Guesthouse* di Kota Medan  
Kategori : Skripsi  
Nama : Wilbert Capri  
Nomor Induk Mahasiswa : 181401050  
Program Studi : SARJANA (S-1) ILMU KOMPUTER  
Fakultas : ILMU KOMPUTER DAN TEKNOLOGI  
INFORMASI UNIVERSITAS SUMATERA  
UTARA

Komisi Pembimbing :

Pembimbing I



Dewi Sanika Br Ginting S.Kom., M.Kom

NIP. 199005042019032023

Pembimbing II



Dr. T. Henny Febriana Harumy, S.Kom.,  
M.Kom

NIP.198802192019032016

Diketahui/ Disetujui oleh

Program Studi S-1 Ilmu Komputer



Dr. Amalia S.T., M.T

NIP.197812212014042001

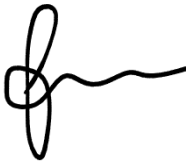
**PERNYATAAN**

**KOMBINASI ALGORITMA ELECTRE DAN *DIJKSTRA* DALAM SISTEM  
PEMILIHAN *GUESTHOUSE* DI KOTA MEDAN**

**SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing – masing telah disebutkan sumbernya.

Medan, 10 Desember 2023



Wilbert Capri

## **KATA PENGANTAR**

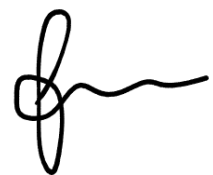
Kita panjatkan puji syukur kepada Tuhan YME karena berkat dan rahmatnya , penulis bisa mampu menyelesaikan skripsi sebagai salah satu syarat kelulusan atau menyelesaikan Program Sarjana (S1) Program Studi Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Sumatera Utara.

Dengan tulus, penulis ingin mengungkapkan rasa syukur dan Terima Kasih dan hormat sebesar – besarnya pada berbagai pihak yang memberikan bantuan selama penulis menyelesaikan skripsi ini, baik itu merupakan dukungan doa, semangat, dan bimbingan. Penulis menyampaikan banyak terima kasih untuk pihak berikut:

1. Bapak Dr. Muriyanto Amin, S.Sos., M.Si selaku Rektor dari Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc., selaku Dekan dari Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
3. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dewi Sartika Br Ginting S.Kom., M.Kom. Selaku Dosen Pembimbing I yang telah memberikan banyak bimbingan, arahan, kritik dan saran kepada penulis sehingga skripsi ini bisa selesai.
5. Ibu Dr.T.Henny Febriana Harumy S.kom., M.Kom. selaku Dosen Pembimbing II yang telah memberikan banyak bimbingan, arahan, kritik dan saran kepada penulis sehingga skripsi ini bisa selesai.
6. Seluruh Bapak dan Ibu Dosen program Studi S-1 Ilmu Komputer yang telah memberikan saran dan kritik dalam penyusunan skripsi penulis ini.
7. Bapak Handrizal, S.Si, M.Comp.Sc selaku dosen Pembimbing Akademik yang telah memberikan nasihat dan bimbingan dalam mengikuti dan menyelesaikan studi di S-1 Ilmu Komputer Universitas Sumatera Utara.
8. Seluruh Staf Pegawai di Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara yang telah banyak memberikan bantuan kepada penulis selama masa perkuliahan sampai kepada tahap penyusunan skripsi ini.

9. Kepada kedua orang tua penulis, Bapak dan Ibu yang selalu nasihat dan selalu mendoakan yang terbaik bagi penulis agar bisa menyelesaikan skripsi ini.
10. Untuk teman – teman saya yang berada di kost Barat, saya ingin memberikan rasa terima kasih karena sudah memberi banyak dukungan, motivasi dan penghiburan selama menempuh pendidikan S-1.
11. Kepada teman – teman seperjuangan yang selalu memberikan dukungan pada penulis, Tegar Sinaga, Rubenhard Tobing, Dicky Manurung, Pius Febryo, Ricky Berlando, Paskal, Gessel Leon, Joe Sitinjak, Benny Sinaga, Johannes Karo, Pangeran, Erick Gultom, Nicholas Nivarel, Adi dan teman-teman lainnya yang belum dapat disebutkan.
12. Teman – teman semasa berkuliah di program studi S-1 Ilmu Komputer, khususnya kom B Angkatan 2018 yang sangat banyak memberi motivasi, bantuan, serta semangat disaat perkuliahan.
13. Kakak – Abang senior yang telah memberi pengalaman serta nasehat dalam menjalani masa-masa perkuliahan.
14. Keluarga Besar IMILKOM USU (Himpunan Mahasiswa S-1 Ilmu Komputer USU). Dimana ini sebagai tempat di mana penulis mendapat pembelajaran untuk berorganisasi disaat masa kuliah.
15. Berbagai pihak baik langsung ataupun tidak langsung terlibat dalam penyelesaian pengerjaan skripsi ini.

Medan, 10 Desember 2023



Penulis

## ABSTRAK

Kota Medan merupakan salah satu kota terbesar di Indonesia dan juga merupakan salah satu kota terbesar di Indonesia dan juga merupakan pintu gerbang utama perdagangan serta merupakan salah satu pusat perekonomian di Sumatera Utara sehingga kota Medan sering dikunjungi oleh para wisatawan-wisatawan dengan berbagai kepentingan. Wisatawan yang berwisata ke kota Medan kesulitan dalam menentukan guesthouse atau akomodasi yang sesuai dengan kriteria wisatawan atau pengunjung, oleh karena itu diperlukan sistem pendukung keputusan yang dapat memudahkan pengunjung dalam memilih guesthouse. Dalam penelitian disini penulis memuat pembahasan proses pemilihan lokasi guesthouse dengan menggunakan kombinasi algoritma Dijkstra dan metode Elimination Et Choix Traduisant la Realit  (ELECTRE). Dengan metode ELECTRE, keputusan akan diambil berdasarkan kriteria yang dimasukkan pengunjung dengan menggunakan pencarian multi kriteria, dan dengan algoritma Dijkstra akan memberikan solusi pencarian jarak terdekat yang mengacu pada kriteria masukan pengunjung. Hasil implementasi sistem ini berupa informasi lokasi penginapan yang sesuai dengan kriteria yang diinput dengan ketepatan 95% yang diperoleh dari pengujian user.

**Kata Kunci :** Sistem Pendukung Keputusan , *Guesthouse*, *Dijkstra Algorithm*, *ELECTRE Method*

## ABSTRACT

The city of Medan is one of the largest cities in Indonesia and is also one of the largest cities in Indonesia and is also the main gateway to trade and is one of the economic centers in North Sumatra so that the city of Medan is often visited by tourists with various interests. Tourists who travel to the city of Medan find it difficult to determine a guesthouse or accommodation that suits the criteria of tourists or visitors, therefore a decision support system is needed to make it easier for visitors to choose a guesthouse. In the research here the author includes a discussion of the process of selecting guesthouse locations using a combination of the Dijkstra algorithm and the Elimination Et Choix Traduisant la Realit  (ELECTRE) method. With the ELECTRE method, decisions will be made based on the criteria entered by the visitor using a multi-criteria search, and with the Dijkstra algorithm it will provide the closest distance search solution which refers to the visitor's input criteria. The resulting implementation of this system is guesthouse location information that matches the input criteria with 95% accuracy rate from user acceptance testing.

**Title :** Guesthouse Selection In Medan City Using Combination Of ELECTRE Method And Dijkstra Algorithm

**Kata Kunci :** Decision Support System, *Guesthouse*, *Dijkstra Algorithm*, *ELECTRE Method*



## DAFTAR ISI

PERSETUJUAN .....	i
PERNYATAAN .....	ii
KATA PENGANTAR .....	iii
ABSTRAK.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN.....	1
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah .....	3
1.3    Batasan Masalah.....	3
1.4    Tujuan Penelitian.....	4
1.5    Manfaat Penelitian.....	4
1.6    Metodologi Penelitian .....	4
1.7    Sistematika Penulisan.....	5
BAB II LANDASAN TEORI.....	7
2.1    Sistem Pendukung Keputusan .....	7
2.2    Metode Elimination Et Choix Traduisant La Realite (ELECTRE) .....	8
2.3    Graf.....	15
2.4    Algoritma Djikstra.....	18
BAB III ANALISIS DAN PERANCANGAN .....	26
3.1    Analisis Sistem .....	26
3.2    Pemodelan Sistem .....	29
3.3    Flowchart.....	35
3.4    Perancangan Antarmuka ( <i>Interface</i> ) Sistem.....	38
3.5    Perancangan Database .....	45
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	48
4.1 Implementasi Sistem .....	48

4.2 Pengujian.....	55
4.3 User Acceptance Test.....	89
BAB V KESIMPULAN DAN SARAN .....	91
5.1 Kesimpulan .....	91
5.2 Saran.....	91
DAFTAR PUSTAKA .....	92
LISTING PROGRAM .....	A-1

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Graf Sederhana (Munir, 2012) .....	15
<b>Gambar 2.2</b> Graf Ganda (Munir, 2012) .....	16
<b>Gambar 2.3</b> Graf Semu (Munir, 2012) .....	16
<b>Gambar 2.4</b> Graf Tak-Berarah (Munir, 2012) .....	17
<b>Gambar 2.5</b> Graf Berarah (Munir, 2012) .....	17
<b>Gambar 2.6</b> Graf Berbobot (Munir, 2012) .....	17
<b>Gambar 2.7</b> Graf tidak berbobot (Munir, 2012) .....	18
<b>Gambar 3.1</b> Diagram Ishikawa .....	27
<b>Gambar 3.2</b> Diagram Umum Sistem .....	30
<b>Gambar 3.3</b> Diagram Use Case .....	31
<b>Gambar 3.4</b> Activity Diagram Sistem Pemilihan Guesthouse.....	35
<b>Gambar 3.5</b> Sequence Diagram Sistem Pemilihan Guesthouse.....	35
<b>Gambar 3.6</b> <i>Flowchart</i> Sistem Pemilihan Guesthouse .....	36
<b>Gambar 3.7</b> <i>Flowchart</i> Metode ELECTRE .....	37
<b>Gambar 3.8</b> <i>Flowchart</i> Algoritma Dijkstra .....	38
<b>Gambar 3.9</b> Rancangan Halaman <i>Splash Screen</i> .....	39
<b>Gambar 3.10</b> Rancangan Halaman Login Akun.....	40
<b>Gambar 3.11</b> Rancangan Halaman Daftar Akun .....	41
<b>Gambar 3.12</b> Rancangan Halaman Beranda.....	42
<b>Gambar 3.13</b> Rancangan <i>Bottomsheet</i> Filter Pemilihan Kriteria .....	43
<b>Gambar 3.14</b> Halaman Hasil Pencarian Sesuai Kriteria .....	44
<b>Gambar 3.15</b> Halaman Detail Informasi Guesthouse .....	45
<b>Gambar 4.1</b> Halaman <i>Splash Screen</i> .....	48
<b>Gambar 4.2</b> Halaman Daftar .....	49
<b>Gambar 4.3</b> Halaman Login .....	50
<b>Gambar 4.4</b> Halaman Beranda.....	51
<b>Gambar 4.5</b> Halaman <i>Bottomsheet</i> Pemilihan Kriteria.....	52
<b>Gambar 4.6</b> Halaman Hasil Pencarian Sesuai Kriteria .....	53
<b>Gambar 4.7</b> Halaman Detail Informasi Lokasi Guesthouse .....	54
<b>Gambar 4.8</b> Pengujian Berupa Input Kriteria dan Lokasi <i>User</i> .....	55

<b>Gambar 4.9</b> Hasil Rekomendasi Guesthouse .....	56
<b>Gambar 4.12</b> Hasil Pengujian Kesesuaian Sistem dengan Kriteria.....	90

## DAFTAR TABEL

<b>Tabel 3.1</b> Narrative Use Case Daftar / Login Akun .....	32
<b>Tabel 3.2</b> Narrative Use Case Rekomendasi Guesthouse.....	32
<b>Tabel 3.3</b> Narrative Use Pencarian Guesthouse dari Filter Kriteria .....	33
<b>Tabel 3.4</b> Narrative Use Informasi Guesthouse Terdekat .....	33
<b>Tabel 3.5</b> Tabel List Guesthouse .....	46
<b>Tabel 3.6</b> Tabel Data User .....	46
<b>Tabel 3.7</b> Tabel Data Jarak .....	47
<b>Tabel 4.1</b> Data Lokasi <i>User</i> .....	57
<b>Tabel 4.2</b> Data Jarak .....	58
<b>Tabel 4.3</b> Data Bobot Jarak .....	65
<b>Tabel 4.4</b> Data Jarak dari USU ke Seluruh Guesthouse .....	65
<b>Tabel 4.5</b> Data Bobot Harga Guesthouse per Malam .....	66
<b>Tabel 4.6</b> Data Bobot Rating Guesthouse Terdekat .....	66
<b>Tabel 4.7</b> Data Bobot Total Fasilitas.....	67
<b>Tabel 4.8</b> Bobot Pengambilan Keputusan .....	67
<b>Tabel 4.9</b> Data Bobot Setiap Alternatif dihitung dari lokasi user di USU.....	67
<b>Tabel 4.10</b> Matriks Ternormalisasi R.....	70
<b>Tabel 4.11</b> Pembobotan pada matriks yang telah dinormalisasi .....	72
<b>Tabel 4.12.1</b> Menentukan himpunan <i>concordance</i> baris 1.....	73
<b>Tabel 4.12.2</b> Menentukan himpunan <i>concordance</i> baris 2.....	73
<b>Tabel 4.12.3</b> Menentukan himpunan <i>concordance</i> baris 3.....	73
<b>Tabel 4.12.4</b> Menentukan himpunan <i>concordance</i> baris 4.....	74
<b>Tabel 4.12.5</b> Menentukan himpunan <i>concordance</i> baris 5.....	74
<b>Tabel 4.12.6</b> Menentukan himpunan <i>concordance</i> baris 6.....	75
<b>Tabel 4.12.7</b> Menentukan himpunan <i>concordance</i> baris 7.....	75
<b>Tabel 4.12.8</b> Menentukan himpunan <i>concordance</i> baris 8.....	76
<b>Tabel 4.12.9</b> Menentukan himpunan <i>concordance</i> baris 9.....	76

<b>Tabel 4.12.10</b> Menentukan himpunan <i>concordance</i> baris 10.....	77
<b>Tabel 4.12.11</b> Menentukan himpunan <i>concordance</i> baris 11.....	77
<b>Tabel 4.12.12</b> Menentukan himpunan <i>concordance</i> baris 12.....	78
<b>Tabel 4.12.13</b> Menentukan himpunan <i>concordance</i> baris 13.....	78
<b>Tabel 4.12.14</b> Menentukan himpunan <i>concordance</i> baris 14.....	78
<b>Tabel 4.12.15</b> Menentukan himpunan <i>concordance</i> baris 15.....	79
<b>Tabel 4.12.16</b> Menentukan himpunan <i>concordance</i> baris 16.....	79
<b>Tabel 4.12.17</b> Menentukan himpunan <i>concordance</i> baris 17.....	79
<b>Tabel 4.12.18</b> Menentukan himpunan <i>concordance</i> baris 18.....	80
<b>Tabel 4.12.19</b> Menentukan himpunan <i>concordance</i> baris 19.....	81
<b>Tabel 4.12.20</b> Menentukan himpunan <i>concordance</i> baris 20.....	81
<b>Tabel 4.13.1</b> Menentukan himpunan <i>discordance</i> baris 1.....	81
<b>Tabel 4.13.2</b> Menentukan himpunan <i>discordance</i> baris 2.....	82
<b>Tabel 4.13.3</b> Menentukan himpunan <i>discordance</i> baris 3.....	82
<b>Tabel 4.13.4</b> Menentukan himpunan <i>discordance</i> baris 4.....	83
<b>Tabel 4.13.5</b> Menentukan himpunan <i>discordance</i> baris 5.....	83
<b>Tabel 4.13.6</b> Menentukan himpunan <i>discordance</i> baris 6.....	84
<b>Tabel 4.13.7</b> Menentukan himpunan <i>discordance</i> baris 7.....	84
<b>Tabel 4.13.8</b> Menentukan himpunan <i>discordance</i> baris 8.....	84
<b>Tabel 4.13.9</b> Menentukan himpunan <i>discordance</i> baris 9.....	85
<b>Tabel 4.13.10</b> Menentukan himpunan <i>discordance</i> baris 10.....	85
<b>Tabel 4.13.11</b> Menentukan himpunan <i>discordance</i> baris 11.....	86
<b>Tabel 4.13.12</b> Menentukan himpunan <i>discordance</i> baris 12.....	86
<b>Tabel 4.13.13</b> Menentukan himpunan <i>discordance</i> baris 13.....	87
<b>Tabel 4.13.14</b> Menentukan himpunan <i>discordance</i> baris 14.....	87
<b>Tabel 4.13.15</b> Menentukan himpunan <i>discordance</i> baris 15.....	88
<b>Tabel 4.13.16</b> Menentukan himpunan <i>discordance</i> baris 16.....	88
<b>Tabel 4.13.17</b> Menentukan himpunan <i>discordance</i> baris 17.....	89
<b>Tabel 4.13.18</b> Menentukan himpunan <i>discordance</i> baris 18.....	89
<b>Tabel 4.13.19</b> Menentukan himpunan <i>discordance</i> baris 19.....	90
<b>Tabel 4.13.20</b> Menentukan himpunan <i>discordance</i> baris 20.....	90
<b>Tabel 5.1</b> Pertanyaan UAT .....	101
<b>Tabel 5.2</b> Hasil UAT .....	102
<b>Tabel 5.3</b> Pembobotan Hasil UAT .....	103

## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang

Ibukota Provinsi Sumatera Utara adalah Kota Medan, merupakan salah satu kota terbesar di Indonesia dan juga menjadi pintu gerbang utama perdagangan dan menjadi salah satu pusat perekonomian di Sumatera dan Kota Medan juga terkenal dengan kuliner dan beberapa objek wisata yang ada di dalam kota maupun sehingga dengan beberapa faktor tersebut kota Medan sering dikunjungi wisatawan dengan berbagai kepentingan dan tujuan liburan, bisnis dan sebagainya. Oleh karena itu *guesthouse* menjadi fasilitas pendukung untuk tempat tinggal sementara bagi orang yang bepergian dengan harga yang ekonomis di kota Medan.

Di karenakan banyaknya *guesthouse* dan kamar yang tersedia sehingga pengguna yang datang ke kota Medan akan sulit memilih *guesthouse* dan beragam fasilitas yang ditawarkan tiap hotel sehingga diperlukannya sebuah sistem untuk mempermudah pengguna dalam menentukan pilihan melalui sistem dengan beberapa kriteria seperti fasilitas, harga, rating dan jarak yang dapat diisi oleh pengguna. Ada pun jarak merupakan salah satu kriteria yang sangat dipertimbangkan untuk mencari *guesthouse* terdekat bagi pengguna. Semua masalah ini dapat diselesaikan dengan sistem pendukung keputusan dengan penerapan beberapa algoritma.

Sistem pendukung keputusan merupakan bagian dari sistem informasi berbasis komputer yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. SPK juga bisa dibilang sebagai sistem komputer yang mengolah data menjadi informasi dalam mengambil keputusan atas masalah semi-terstruktur yang spesifik. Terdapat beberapa algoritma yang dapat dipakai dalam sistem pendukung keputusan, salah satunya adalah Elimination Et Choix Traduisant la RealitÃ (ELECTRE). Kelebihan dari metode ini adalah mempermudah pengambilan keputusan dari ketidakjelasan dan ketidakpastian pada kasus yang mempunyai banyak alternatif (Novendra Adiputra Sinaga, Arifin Tua Purba, 2019). Metode ini digunakan untuk menangani hubungan outranking dengan menggunakan perbandingan berpasangan antara alternatif di bawah masing – masing kriteria secara terpisah.

Dalam sistem pendukung keputusan pemilihan *guesthouse* ini terdapat beberapa kriteria yang digunakan yaitu fasilitas, harga, review dan jarak. Dimana kriteria yang diberikan akan membantu pengguna dalam mencari tau seberapa jauh jarak dari lokasi

*guesthouse* dari titik lokasi user. Maka untuk membantu penelitian digunakan lah algoritma pencarian jarak salah satunya adalah algoritma *Dijkstra*.

Algoritma *Dijkstra* merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Prinsip algoritma *Dijkstra* adalah pencarian dua lintasan yang paling kecil. Algoritma *Dijkstra* memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Beberapa penelitian yang telah mengembangkan Sistem Informasi Geografis (SIG), salah satunya dilakukan oleh Priyantoro, penelitian ini berhasil membangun aplikasi pencarian rute terbaik berdasarkan jadwal keberangkatan, jarak terdekat dan jumlah pergantian armada berbasis *Google Map* pada sistem operasi Android dengan studi kasus Trans Jogja. Selain itu pada penelitian yang dilakukan oleh Tirastitam mengenai sistem perencanaan transportasi umum berbasis algoritma *Dijkstra* dengan studi kasus Bangkok Metropolitan Area dengan berfokus pada bus, BTS dan MRT jadwal/rute untuk memberikan informasi kepada penumpang. Penumpang dapat memilih jalan dan rute dengan mudah menggunakan *Dijkstra* STAR Algoritma, dan diperoleh tarif perjalanan. Berdasarkan latar belakang yang telah dijelaskan, maka penulis ingin mengkombinasikan algoritma Elimination Et Choix Traduisant la Realita (ELECTRE) dan algoritma *Dijkstra* ke dalam sebuah Sistem Pendukung Keputusan Pemilihan *guesthouse* di Kota Medan. Dimana algoritma ELECTRE digunakan untuk memberi rekomendasi *guesthouse* sesuai kriteria yang akan dipilih oleh pengambil keputusan dan algoritma *Dijkstra* digunakan untuk mengetahui jarak antara lokasi yang diinginkan pengguna dengan *guesthouse* yang dijadikan alternatif pilihan. Selain itu kinerja dari masing – masing algoritma ini juga akan sangat mempengaruhi hasil akhir dari penyelesaian masalah sehingga penulis juga perlu untuk mengukur kinerja algoritma berdasarkan kompleksitas algoritma.

## 1.2 Rumusan Masalah

Berdasarkan permasalahan yang ada di latar belakang yaitu proses pemilihan *guesthouse* dengan berbagai jenis pelayanan yang diberikan tiap *guesthouse* yang berbeda akan membuat para pengguna cukup sulit menentukan pilihan *guesthouse* yang diinginkan. Oleh sebab itu sistem ini dibuat dimana diterapkannya algoritma Elimination Et Choix Traduisant la Realita (ELECTRE) pada sistem pendukung keputusan untuk memberikan rekomendasi *guesthouse* di kota Medan dan diterapkan



algoritma *Dijkstra* untuk mendapatkan jarak dari lokasi pengguna ke *guesthouse* yang dijadikan rekomendasi.

### 1.3 Batasan Masalah

Beberapa batasan masalah dalam penelitian ini adalah:

1. Algoritma yang digunakan pada sistem pendukung keputusan untuk memilih *guesthouse* adalah algoritma Elimination Et Choix Traduisant la Realit   (ELECTRE).
2. Algoritma yang digunakan untuk mengetahui jarak antara lokasi yang diinginkan user dengan *guesthouse* yang dijadikan alternatif pilihan adalah Algoritma Dijkstra.
3. Kriteria yang digunakan dalam menentukan keputusan *guesthouse* adalah fasilitas, harga, rating dan jarak.
4. Penentuan bobot kriteria alternatif *guesthouse* diperoleh dari data yang dikumpul dari survei langsung ke lokasi.
5. Jenis graf yang digunakan adalah graf berarah (directed graph) dan graf berbobot di mana bobot menyatakan jarak.
6. Implementasi sistem berbasis mobile dan menggunakan bahasa pemrograman Java dan menggunakan MySQL sebagai sistem database manajemen.

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membangun sebuah sistem pendukung keputusan dalam pemilihan *guesthouse* di kota Medan dengan metode Electre
2. Mengetahui jarak antara lokasi yang diinginkan / dimasukkan oleh *user* dengan lokasi *guesthouse* lain dengan menggunakan algoritma Dijkstra.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah menghasilkan sebuah sistem pendukung keputusan yang dapat digunakan pengguna atau calon pengunjung dalam pemilihan *guesthouse* berdasarkan kriteria yang diinginkan menggunakan metode Elimination Et Choix Traduisant la Realit   (ELECTRE) dan algoritma Dijkstra

## 1.6 Metodologi Penelitian

Metode penelitian yang dilakukan dalam penelitian ini adalah:

### 1. Studi Pustaka

Pada tahap ini dilakukan pencarian dan pengumpulan referensi yang dibutuhkan untuk penulisan penelitian. Sumber referensi berupa jurnal, buku, artikel ilmiah, makalah dan juga situs resmi internet yang berhubungan dengan metode sistem pengambilan keputusan dan referensi lainnya.

### 2. Analisis dan Perancangan

Pada tahap ini penulis melakukan analisis terhadap berbagai keperluan yang dibutuhkan dalam penelitian untuk dirancang dalam sebuah diagram alur (flowchart), UML dan diagram Ishikawa.

### 3. Implementasi Sistem

Pada tahap ini dilakukan pengkodean dalam Bahasa pemrograman JAVA dan juga menggunakan Sqlite sebagai basis datanya.

### 4. Pengujian Sistem

Pada tahap ini, sistem atau aplikasi yang telah dibuat akan dilakukan pengujian.

### 5. Dokumentasi

Pada tahap ini dilakukan pembuatan laporan dari hasil analisa dan perancangan sistem dalam format penulisan berbentuk skripsi

## 1.7 Sistematika Penelitian

### **BAB 1            Pendahuluan**

Di bab pendahuluan menjelaskan latar belakang penelitian yang dilakukan berjudul “Kombinasi Algoritma Electre dan Dijkstra dalam Sistem Pemilihan Guesthouse di Kota Medan”, rumusan masalah , batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penelitian.

### **BAB 2            LANDASAN TEORI**

Di bab ini menjelaskan tentang teori – teori yang berhubungan dengan sistem pendukung keputusan, *Guesthouse*, Algoritma Dijkstra, Metode Electre dan sebagainya.

**BAB 3                    ANALISIS DAN PERANCANGAN**

Bab ini berisi analisis terhadap alur kerja program yang menggunakan metode Electre dan juga Algoritma Dijkstra terhadap data – data yang didapat dari pakar,serta perancangan sistem yang akan dibuat.

**BAB 4                    IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi tahapan pembuatan sistem pakar yang berisi implementasi dari metode Electre dan juga Algoritma Dijkstra sesuai dengan analisis dan perancangan, dilanjutkan dengan pengujian terhadap sistem pakar yang dibangun.

**BAB 5                    KESIMPULAN DAN SARAN**

Di bab ini berisi kesimpulan dari hasil pengujian sistem yang dibangun serta saran dari penulis untuk penelitian selanjutnya berdasarkan hasil pengujian sistem.

## BAB 2

### LANDASAN TEORI

#### 2.1 Sistem pendukung keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) adalah sistem yang akan memberikan dukungan keputusan yang bertujuan untuk memecahkan masalah baik dalam kondisi yang terstruktur maupun tidak terstruktur. Menurut Raymond McLeod, sistem pendukung keputusan adalah sistem penghasil informasi spesifik yang bertujuan memecahkan masalah yang harus dipecahkan oleh manajer pada berbagai tingkatan.

##### 2.1.1 Karakteristik dari Sistem Pendukung Keputusan

Beberapa Karakteristik Sistem Pendukung Keputusan yaitu (Gholam dkk, 2009):

1. SPK menawarkan kemudahan, adaptasi yang mudah dan respon cepat.
2. Memberikan pemakai akses untuk memulai dan mengendalikan *input* dan *output*.
3. Mudah untuk dioperasikan, tanpa harus memerlukan pemrogram profesional.
4. Memberikan dukungan keputusan dan masalah yang solusinya tidak dapat ditentukan di depan.
5. Menggunakan analisis data dan perangkat pemodelan yang cukup canggih

##### 2.1.2 Tahap-Tahap dalam Pengambilan Keputusan

Berikut tahap dalam pengambilan keputusan menurut Herbert A. Simon (Kadarsah, 2002:15-16):

###### 1. *Intelligence Phace*

Pada tahap ini, adalah proses melacak dan mendeteksi sejauh mana masalah dan mengidentifikasi masalah. Data masukan diterima, diproses, dan diuji pada untuk mengidentifikasi masalah.

###### 2. *Design Phace*

Tahap ini merupakan proses mengembangkan dan menemukan alternatif tindakan/solusi yang layak. Ini adalah representasi yang disederhanakan dari kejadian sebenarnya, sehingga diperlukan proses validasi dan validasi untuk menentukan keakuratan model saat menyelidiki masalah yang ada.

### 3. *Choice Phace*

Tahap ini dilakukan dengan memilih dari berbagai alternatif solusi yang dimunculkan pada tahap perencanaan dan menentukan atau mempertimbangkan kriteria- kriteria berdasarkan tujuan yang ingin dicapai.

### 4. *Implementation Phace*

Tahap ini berlaku untuk rencana sistem yang dibuat pada tingkat perencanaan dan implementasi tindakan alternatif yang dipilih pada tingkat pemilihan.

## 2.2 Metode Electre

Menurut ELECTRE adalah salah satu metode pengambilan keputusan multi-kriteria berdasarkan konsep *superordinat* dengan menggunakan perbandingan berpasangan dari pilihan berdasarkan setiap kriteria (Kusumadewi dkk, 2006). Ini hanya digunakan jika ada banyak pilihan. Standar relevan. Jika salah satu atau lebih kriteria terlampaui (dibandingkan dengan kriteria alternatif lainnya) dan samadengan kriteria lainnya yang tersisa, alternatif tersebut dikatakan mendominasi alternatif lainnya.

Metode ELECTRE terkandung dalam metode *outranking* yang menghasilkan beberapa daftar pilihan. Metode *superordinat* diterapkan langsung ke fungsi preferensi parsial yang diterima ditentukan untuk setiap kriteria

#### 1. Normalisasi Matriks Keputusan

Tahap ini setiap atribut diubah menjadi nilai yang dapat dibandingkan. Setiap normalisasi dari  $X_{ij}$  dapat dilakukan dengan persamaan (1):

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (1)$$

Untuk  $i = 1, 2, 3, \dots, m$  dan  $j = 1, 2, 3, \dots, n$ . Sehingga didapatkan matriks R normalisasi:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}$$

R merupakan matriks hasil normalisasi, di mana  $m$  adalah alternatif,  $n$  adalah kriteria, dan  $r_{ij}$  merupakan normalisasi pengukuran terpilih dari alternatif ke  $i$  yang berhubungan dengan kriteria ke- $j$ .

## 2. Pembobotan Matriks yang telah dinormalisasi

Setelah normalisasi, setiap kolom pada matriks R dikalikan dengan bobot-bobot ( $W_j$ ) yang ditentukan pengambil keputusan. Perkalian dengan bobot matriks dapat dilakukan dengan menggunakan rumus:

$$V = R * W \quad (2)$$

Sehingga didapatkan, *weight normalized matrix* adalah:

$$V = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ V_{m1} & V_{m2} & \cdots & V_{mn} \end{bmatrix} = RW = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \cdots & w_n r_{1n} \\ w_2 r_{21} & w_2 r_{22} & \cdots & w_n r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_1 r_{m1} & w_2 r_{m2} & \cdots & w_n r_{mn} \end{bmatrix}$$

## 3. Menentukan himpunan *Concordance* dan *Discordance index*.

Untuk setiap pasang dari alternatif  $k$  dan  $l$  ( $k, l = 1, 2, 3, \dots, m$  dan  $k \neq l$ ) kumpulan kriteria terbagi menjadi 2 himpunan bagian yaitu *Concordance* dan *Discordance*. Sebuah kriteria dalam suatu alternatif termasuk *Concordance* jika sesuai dengan persamaan (3):

$$C_{kl} = \{j, y_{kj} \geq y_{lj}\}, \text{ untuk } j = 1, 2, 3, \dots, n \quad (3)$$

Sebaliknya komplementer dari himpunan bagian *Concordance* adalah himpunan *discordance* jika sesuai dengan persamaan (4):

$$D_{kl} = \{j, y_{kj} < y_{lj}\}, \text{ untuk } j = 1, 2, 3, \dots, n \quad (4)$$

#### 4. Menghitung matriks *Concordance* dan *Discordance*

Untuk menghitung matriks *Concordance* melalui persamaan:

$$C_{kl} = \sum jcw W_j \quad (5)$$

Sehingga matriks *Concordance* yang dihasilkan adalah:

$$C = \begin{bmatrix} - & C_{12} & C_{13} & \cdots & C_{1n} \\ C_{21} & - & C_{23} & \cdots & C_{2n} \\ \vdots & & & & \\ C_{m1} & C_{m2} & C_{m2} & \cdots & - \end{bmatrix}$$

Untuk menghitung matriks *Discordance* melalui persamaan:

$$d_{kl} = \frac{\{max(V_{mn}-V_{mn}-ln)\};m,n \in D_{kl}}{\{max(V_{mn}-V_{mn}-ln)\};m,n=1,2,3,...} \quad (6)$$

Sehingga matriks *Discordance* yang dihasilkan adalah:

$$D = \begin{bmatrix} - & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & - & d_{23} & \cdots & d_{2n} \\ \vdots & & & & \\ d_{m1} & d_{m2} & d_{m2} & \cdots & - \end{bmatrix}$$

#### 5. Menghitung matriks dominan *Concordance* dan *Discordance*

Menghitung matriks dominan *Concordance* dilakukan menggunakan persamaan

(7):

$$C_{kl} \geq \underline{c} \quad (7)$$

Dengan nilai *threshold* ( $\underline{c}$ ) dapat dilihat pada persamaan (8):

$$\underline{c} = \frac{\sum_{k=1}^n \sum_{l=1}^n C_{kl}}{m(m-1)} \quad (8)$$

Maka elemen matriks F dapat ditentukan menggunakan persamaan (9):

$$f_{kl} = 1, \text{ jika } C_{kl} \geq \underline{c} \text{ dan } f_{kl} = 0, \text{ jika } C_{kl} < \underline{c} \quad (9)$$

Menghitung matriks dominan *Discordance* dengan matriks G sebagai matriks dominan *Discordance* dapat ditentukan dengan bantuan nilai *threshold* ( $\underline{d}$ ) melalui persamaan (10):

$$\underline{d} = \frac{\sum_{k=1}^n \sum_{l=1}^n d_{kl}}{m(m-1)} \quad (10)$$

## 6. Menentukan agregat dominan matriks

Agregasi dari matriks dominan (E) merupakan urutan preferensi parsial dari alternatif-alternatif yang didapat dari kombinasi perkalian antara elemen matriks F dengan elemen matriks G yang bersesuaian, secara matematis dapat diperoleh dengan rumus(11):

$$E_{kl} = F_{kl} * G_{kl} \quad (11)$$

## 7. Eliminasi alternatif yang *less favourable*

Matriks E memberikan urutan pilihan dari setiap alternatif, yaitu bila  $E_{kl}=1$  sehingga alternatif  $A_k$  adalah alternatif yang lebih baik daripada  $A_l$ . Maka barisdi dalam matriks E yang berjumlah  $E_{kl} = 1$  paling sedikit dapat dieliminasi. Dengan demikian, alternatif terbaik yaitu alternatif yang mendominasi alternatif lainnya.

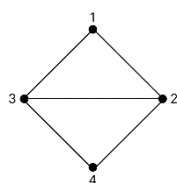
## 2.3 Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E) dan dideskripsikan dengan notasi  $G = (V, E)$ . Di mana V adalah himpunan yang tidak kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan simpul. (Rinaldi Munir, 2010).

Berdasarkan ada atau tidak adanya gelang atau tepi ganda dalam graf, maka graf dapat dibedakan menjadi dua jenis (Munir, 2012):

### 1. Graf Sederhana (*simple graph*)

Graf sederhana yaitu graf yang tidak mempunyai gelang atau sisi ganda. Gambar 2.1 merupakan contoh graf sederhana. Sisi pada graf sederhana adalah pasangan tidak terurut (*unordered pairs*) sehingga penulisan sisi (u,v) sama dengan (v,u). Lalu graf sederhana  $G=(V,E)$  terdiri dari himpunan tidak kosong simpul-simpul dan E adalah himpunan pasangan tidak terurut yang berbeda yang disebut sisi.

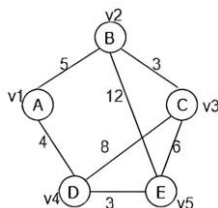


**Gambar 2.1.** Graf Sederhana (Munir, 2012)

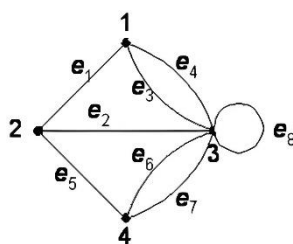


## 2. Graf Tidak Sederhana (*Unsimple Graph*)

Graf tidak sederhana yaitu graf yang mempunyai sisi ganda atau gelang. Terdapat dua jenis graf tidak sederhana, yaitu graf semu (*pseudograph*) dan graf ganda (*multigraph*). Graf ganda yaitu graf yang mempunyai sisi ganda di mana sisi diasosiasikan sebagai pasangan tidak terurut yang sama. Gambar 2.2. adalah contoh graf ganda. Graf semu yaitu graf yang memiliki gelang (*loop*) di mana sisi mampu terhubung ke dirinya sendiri. Gambar 2.3. adalah contoh graf semu.



**Gambar 2.2.** Graf Ganda (Munir, 2012)

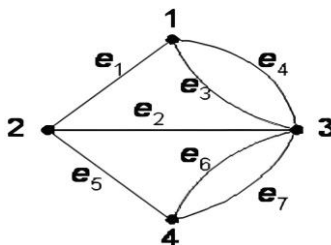


**Gambar 2.3.** Graf Semu (Munir, 2012)

Lalu sisi pada graf dibedakan berdasarkan orientasi arah pada sisi, berikut beberapa jenisnya (Munir, 2012):

### a. Graf Tidak Berarah (*undirected graph*)

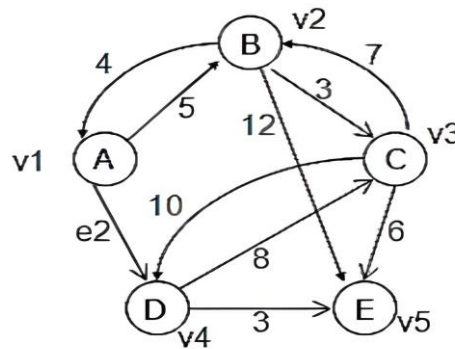
Graf tidak berarah yaitu graf yang orientasi arahnya tidak ada untuk setiap sisinya dan juga tidak diperhatikan urutan pasangan simpulnya. Sehingga,  $(u,v) = (v,u)$  adalah sisi yang sama. Berikut pada gambar 2.4. adalah contoh graf tidak berarah



**Gambar 2.4.** Graf Tak-Berarah (Munir, 2012)

b. Graf Berarah (*directed graph*)

Graf berarah yaitu graf yang orientasi arahnya berbeda untuk setiap sisinya di mana  $(u,v) \neq (v,u)$ . Maka  $u$  sebagai simpul asal (*initial vertex*) dan  $v$  sebagai simpul terminal (*terminal vertex*). Berikut pada Gambar 2.5. adalah contoh graf berarah.

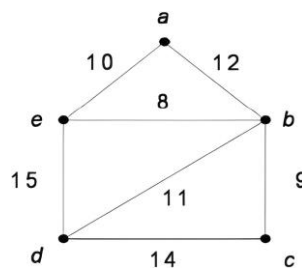


**Gambar 2.5.** Graf Berarah (Munir, 2012)

Sisi pada graf juga dapat membedakan graf menurut bobotnya jadi 2 jenis yaitu:

a. Graf Berbobot (*weighted graph*)

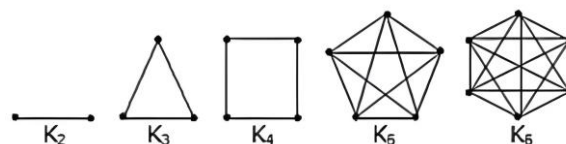
Graf berbobot yaitu graf yang diberi bobot (*cost*) pada setiap sisinya di mana nilainya berbeda-beda sesuai dengan masalah yang digambarkan dengan graf. Contoh graf berbobot ada pada gambar 2.6.



**Gambar 2.6.** Graf Berbobot (Munir, 2012)

b. Graf tidak berbobot (*unweighted graph*)

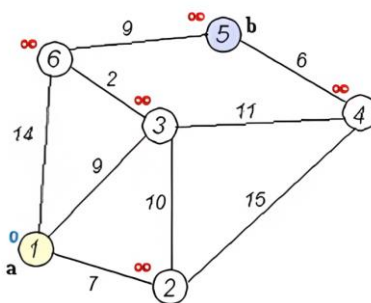
Graf tidak berbobot adalah semua jenis graf yang tidak mempunyai harga pada setiap sisinya. Contoh graf tidak berbobot ada pada gambar 2.7.



**Gambar 2.7.** Graf tidak berbobot (Munir, 2012)

## 2.4 Algoritma Dijkstra

Algoritma *Dijkstra* merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Prinsip algoritma *Dijkstra* adalah pencarian dua lintasan yang paling kecil. Algoritma *Dijkstra* memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek.



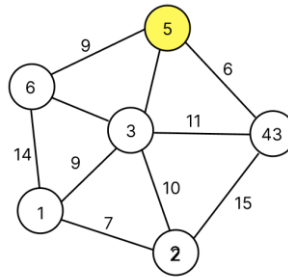
**Gambar 2.8.** menemukan jalur menggunakan algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi *node* awal, lalu beri bobot jarak pada *node* pertama ke *node* terdekat satu per satu, *Dijkstra* akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma *Dijkstra*.

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (belum terisi)
2. Set semua *node* “Belum terjamah” dan set *node* awal sebagai “Node Keberangkatan”
3. Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik 18 keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke *node* C berjarak 2, maka jarak ke C melewati B menjadi  $6+2=8$ . Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah terjamah sebagai “Node terjamah”. *Node* terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “Node belum terjamah” dengan jarak terkecil (dari *node* keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke *step* 3

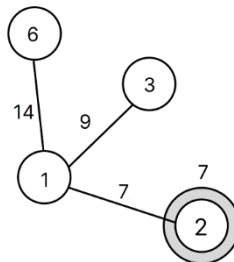
Dibawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dari mulai *node* awal sampai *node* tujuan dengan nilai jarak terkecil.

1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai.



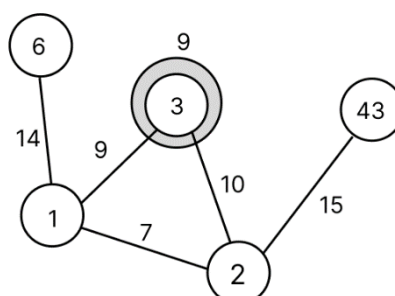
**Gambar 2.9.** menemukan jalur menggunakan algoritma Dijkstra

2. *Dijkstra* melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan hasil yang didapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai =  $0 + 7$



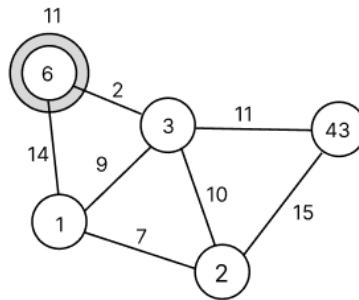
**Gambar 2.10.** menemukan jalur menggunakan algoritma Dijkstra

3. *Node 2* diset menjadi *node* keberangkatan dan ditandai sebagai *node* yang telah terjamah. *Dijkstra* melakukan kalkulasi kembali terhadap *node-node* tetangga yang terhubung langsung dengan *node* yang telah terjamah. Dan kalkulasi *Dijkstra* menunjukkan bahwa *node 3* yang menjadi *node* keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai  $9 (0+9)$ .



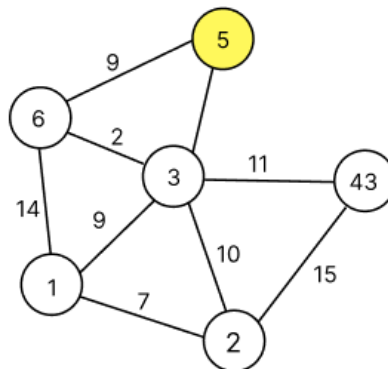
**Gambar 2.11.** menemukan jalur menggunakan algoritma Dijkstra

4. Perhitungan berlanjut dengan *node* 3 ditandai menjadi *node* yang telah terjamah. Dari semua *node* tetangga belum terjamah yang terhubung langsung dengan *node* terjamah, *node* selanjutnya yang ditandai menjadi *node* terjamah adalah *node* 6 karena nilai bobot yang terkecil, nilai 11 ( $9+2$ ).



**Gambar 2.12.** menemukan jalur menggunakan algoritma Dijkstra

5. *Node* 6 menjadi *node* terjamah, *Dijkstra* melakukan kalkulasi kembali, dan menemukan bahwa *node* 5 (*node* tujuan ) telah tercapai lewat *node* 6. Jalur terpendeknya adalah 1-3-6-5, dan nilai bobot yang didapat adalah 20 ( $11+9$ ). Bila *node* tujuan telah tercapai maka kalkulasi *Dijkstra* dinyatakan selesai.



**Gambar 2.13.** menemukan jalur menggunakan algoritma Dijkstra

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 Analisis Sistem

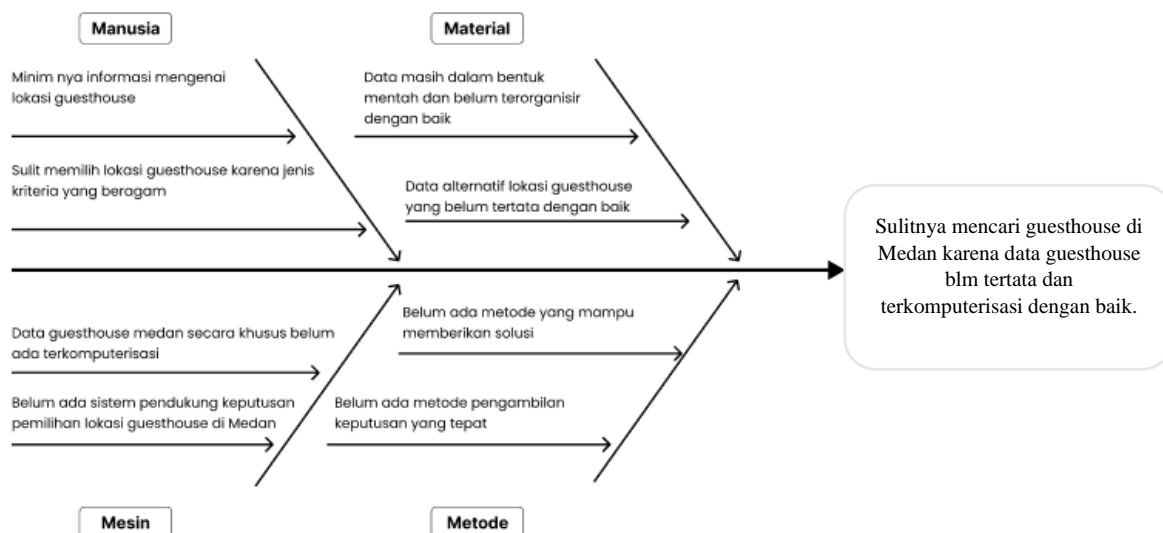
Analisis sistem adalah cara untuk menganalisis informasi dengan tujuan mengidentifikasi masalah dan kebutuhan yang akan muncul dalam produksi sistem, hal ini dilakukan agar pada saat proses perancangan aplikasi tidak berlangsung kesalahan yang signifikan, sehingga sistem yang dirancang dapat berfungsi dengan benar, penggunaan yang tepat dan ketahanan sistem akan lebih baik dipertahankan dan diselesaikan. Analisis Sistem juga dapat mengevaluasi layaknya sebuah sistem untuk digunakan.

##### 3.1.1. Analisis Masalah

Dengan banyaknya *guesthouse* di kota Medan membuat masyarakat bingung dalam memilihnya. Dengan banyaknya pilihan, pengguna akan lebih selektif saat memilih *guesthouse* berdasarkan kriteria dan keinginan seperti lokasi, fasilitas, dan faktor lainnya. Hal tersebut membuat pengguna kesulitan dalam menemukan pilihan *guesthouse*

Sehingga untuk memecahkan masalah tersebut, suatu sistem pendukung keputusan diciptakan untuk memudahkan wisatawan dalam memilih *guesthouse*, di dalam sistem ini akan menerapkan algoritma *Dijkstra* untuk menemukan jarak antar lokasi user dengan alternatif pilihan *guesthouse* dan algoritma *Elimination Et Choix Traduisant La Realite* (ELECTRE) untuk pendukung keputusan multikriteria dalam memilih *guesthouse* di kota Medan. Untuk mengidentifikasi masalah yang ada tersebut, maka digunakan *Ishikawa Diagram* (*fishbone diagram*). Diagram Ishikawa merupakan diagram yang menggambarkan permasalahan dalam format sebab akibat yang digunakan dalam menemukan sumber – sumber permasalahan.

*Ishikawa Diagram* ada pada Gambar 3.1.



**Gambar 3.1** Diagram Ishikawa

Di dalam gambar 3.1 yang menjadi masalah utama yaitu pemilihan lokasi guesthouse di kota Medan, dikarenakan belum ada tercipta sistem khusus yang bertujuan menyelesaikan masalah tersebut. Maka perlu diciptakan sebuah sistem pemilihan lokasi guesthouse untuk memudahkan pengguna dalam menentukan lokasi guesthouse yang mereka inginkan.

### 3.1.2. Analisis Kebutuhan (*Requirement Analysis*)

Analisis kebutuhan merupakan proses menggambarkan fungsionalitas perangkat lunak, menentukan antarmuka perangkat lunak dengan komponen sistem lainnya, dan mengidentifikasi masalah yang harus diatasi oleh perangkat lunak. Analisis kebutuhan perlu dilakukan untuk memahami sepenuhnya masalah perangkat lunak yang dikembangkan dan pengguna yang menggunakannya, dan untuk menentukan apa yang harus dilakukan perangkat lunak untuk memenuhi kebutuhan pengguna. Analisis kebutuhan sistem dibagi menjadi dua bagian yaitu :

#### 3.1.2.1. Kebutuhan Fungsional

Kebutuhan fungsional yaitu aktivitas yang menentukan fungsionalitas atau perilaku sistem. Kebutuhan fungsional ini meliputi *input*, *output*, *processing*, dan *stored data*. Kebutuhan fungsional untuk sistem ini adalah:

1. Sistem dapat menerima inputan dari pengguna sesuai dengan kriteria guesthouse yang pengguna inginkan.
2. Sistem dapat menampilkan lokasi guesthouse yang dipilih sesuai dengan inputan pengguna menggunakan metode ELECTRE.
3. Sistem dapat mengeluarkan hasil dari jarak antara lokasi pengguna dengan lokasi guesthouse yang menjadi alternatif dengan menggunakan metode *Dijkstra*.

### 3.1.2.2. Kebutuhan Non-fungsional

Kebutuhan non-fungsional yaitu deskripsi sistem dalam hal non fungsionalitas seperti fungsi, kondisi batas, dan standard yang harus dipenuhi sistem (kualitas sistem). Kebutuhan non-fungsional untuk sistem ini adalah :

#### 1. Performa

Sistemnya berkualitas baik, yang membantu pengguna dalam memberikan keputusan saat memilih lokasi guesthouse. Sistem ini jugadibangun dengan data yang terintegrasi dengan baik dan tersimpan di *database*, sehingga memudahkan dalam pengambilan dan pengolahan data.

#### 2. Mudah dimengerti dan digunakan

Tampilan pada sistem bersifat *user friendly* sehingga pengguna mudah memahami dalam menggunakannya.

#### 3. Hemat Biaya

Tidak adanya perangkat tambahan yang harus digunakan sehingga tidak mengeluarkan biaya yang besar.

### 3.1.3. Analisis Proses

Sistem yang akan dibangun di sini adalah sistem pendukung keputusan pemilihan lokasi guesthouse di kota Medan, di mana pengguna akan melakukan inputan kriteria lokasi guesthouse yang sesuai dengan keinginan mereka dan sistem akan mengeluarkan rekomendasi alternatif lokasi guesthouse yang sesuai. Di dalam sistem ini akan digunakan algoritma *Elimination Et Choix Traduisant la*



*RealitÃ© (ELECTRE)* dalam pemrosesan pemilihan lokasi guesthouse dan akan menggunakan algoritma *Dijkstra* dalam pemrosesan pencarian jarak lokasi user dengan lokasi yang menjadi alternatif. Proses yang akan dikerjakan sistem berupa

1. Menentukan alternatif dan kriteria dari lokasi guesthouse.
2. Memberikan *range* bobot pada setiap kriteria.
3. Pemrosesan inputan dari *user* untuk mengecek jarak lokasi pengguna dengan alternatif lokasi guesthouse dengan algoritma *Dijkstra*. Untuk lokasi pengguna sendiri akan dibatasi sebanyak 15 titik lokasi. Dan untuk lokasi guesthouse ada sebanyak 20 lokasi.
4. Perangkingan alternatif lokasi guesthouse dengan metode *ELECTRE*. Setelah dilakukan perbandingan inputan dari *user*, dengan alternatif lokasi guesthouse maka dari perhitungan akan di tampilan tiga rekomendasi teratas dari perbandingan tersebut.

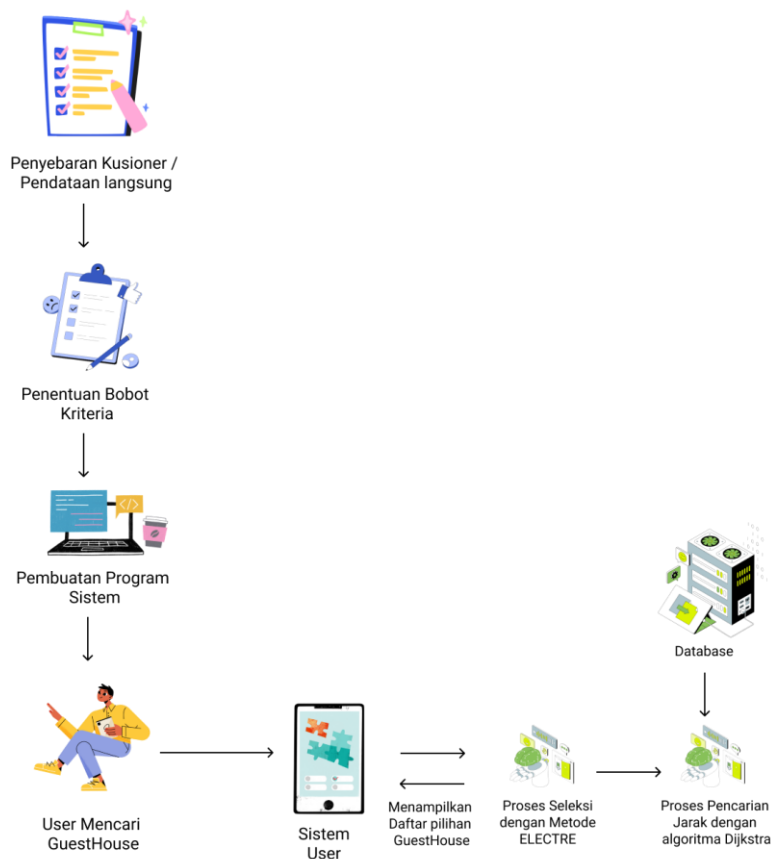
### 3.2 Pemodelan Sistem

Pemodelan sistem perlu dilakukan untuk mendapatkan gambaran yang lebih jelas tentang objek apa saja yang berinteraksi dengan sistem dan apa yang perlu dilakukan sistem agar dapat berfungsi dengan baik.

Penelitian ini menggunakan UML (*Unified Modeling Language*) yang menjadi sebuah bahasa pemodelan untuk merancang dan merancang sistem pendukung keputusan dalam memilih lokasi guesthouse berdasarkan kriteria yang diinginkan oleh pengguna. Model UML yang digunakan yaitu *use case diagram*, *activity diagram*, dan *sequence diagram*.

#### 3.2.1. Diagram Umum Sistem

Diagram umum sistem merupakan perancangan yang menampilkan gambaran proses, alur, dan interaksi antara beberapa komponen di dalam sebuah sistem. Perancangan dalam keseluruhan sistem akan ditampilkan pada diagram umum yang dapat dilihat pada gambar 3.2.



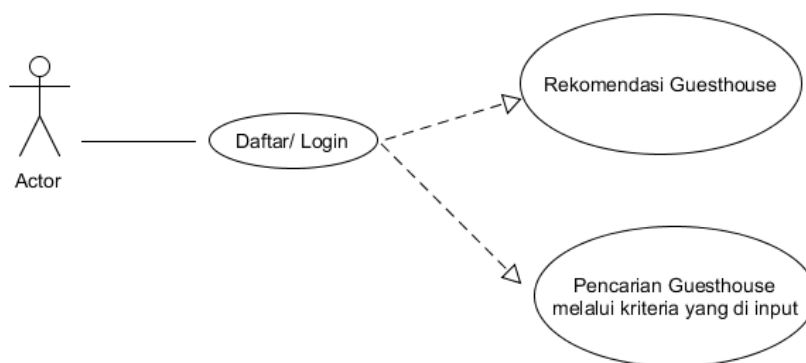
**Gambar 3.2** Diagram Umum Sistem

Berikut Penjelasan alur proses diagram umum sistem pada gambar 3.2:

1. Pengumpulan data referensi dan alternatif dengan menyebarkan kusioner sehingga diperoleh *range* yang cocok untuk bobot kriteria.
2. Membuat program system dengan mengimplementasikan metode ELECTRE dan algoritma Dijkstra
3. Program dibuat sesuai algoritma yang dipakai
4. User menjalankan system dan memasukkan semua data dan lokasi berdasarkan kriteria *guesthouse* yang diinginkan
5. *input* data lokasi pengguna diproses dan sistem menggunakan algoritma Dijkstra untuk menghitung jarak ke setiap lokasi *guesthouse*
6. Sistem menggunakan metode ELECTRE untuk melakukan proses pemilihan *guesthouse*
7. Setelah semua data di proses dan jarak sudah ditentukan, sistem akan menampilkan daftar pilihan *guesthouse*.

### 3.2.2. Use Case Diagram

*Use case diagram* dibangun untuk menampilkan penggambaran fungsionalitas yang diharapkan dari suatu sistem, mengenai "apa" yang dilakukan sistem dan bukan "bagaimana". *Use case* ini merepresentasikan interaksi di antara aktor dan sistem. *Use case diagram* sistem ini ditunjukkan pada Gambar 3.3.



**Gambar 3.3** Diagram Use Case

Pada gambar 3.3. dideskripsikan bahwa pada saat *user* menjalankan sistem untuk mencari lokasi guesthouse, mereka harus melakukan pendaftaran / masuk ke dalam akun untuk mengakses fitur yang ditawarkan. Setelah itu mereka akan memilih kriteria yang mereka inginkan melalui pencarian dengan menggunakan filter atau mereka langsung melihat rekomendasi acak tanpa proses algoritma sistem. Saat pengguna tadi telah memasukkan inputan kriteria, sistem selanjutnya akan melakukan proses mencari jarak dari lokasi *user* ke lokasi guesthouse alternatif memakai algoritma *Dijkstra* lalu setelah itu, akan dilakukan proses pemilihan lokasi guesthouse sesuai kriteria dengan algoritma ELECTRE. Lalu sistem akan menampilkan hasil berbentuk rekomendasi alternatif paling sesuai dengan kriteria guesthouse yang dilakukannya.

Di bawah ini merupakan *narrative use case* pemilihan guesthouse ada pada Tabel 3.1, 3.2, 3.3, dan 3.4.

**Tabel 3.1** Narrative Use Case Daftar / Login Akun

<b>Use case name</b>	Daftar / Login Akun	
<b>Actor</b>	Pengguna	
<b>Description</b>	<i>Use case</i> berisi penjelasan bagaimana <i>user</i> masuk atau mendaftar ke dalam akun nya untuk mengakses fitur-fitur lain.	
<b>Precondition</b>	Pengguna menjalankan sistem	
<b>Typical course of event</b>	Aksi Pengguna	Respon Sistem
	Langkah 1: Mengisi email dan <i>password</i> atau mendaftarkan nama, email, dan <i>password</i> .	Langkah 2 : Sistem menerima inputan data user, dan jika sistem mendeteksi data sudah benar maka akan langsung masuk ke beranda aplikasi
<b>Alternate course</b>	Aksi Pengguna	Respon Sistem
	<i>Inputan</i> nama, email, dan <i>password</i> tidak sesuai	Menampilkan pesan tidak bisa masuk
<b>Post condition</b>	1. Kondisi berhasil Data diterima dan diproses	2. Kondisi tidak berhasil Input tidak sesuai database akan muncul <i>alert</i>

**Tabel 3.2** Narrative Use Case Rekomendasi Lokasi Guesthouse

<b>Use case name</b>	Rekomendasi Lokasi Guesthouse	
<b>Actor</b>	Pengguna	
<b>Description</b>	<i>Use case</i> berisi penjelasan bagaimana <i>user</i> mendapatkan rekomendasi secara acak daftar-daftar lokasi guesthouse di kota medan	
<b>Precondition</b>	Pengguna menjalankan sistem	
<b>Typical course of event</b>	Aksi Pengguna	Respon Sistem
	Langkah 1 : Melakukan <i>scroll</i> serta <i>click card-card</i> informasi lokasi guesthouse	Langkah 2 : Sistem akan langsung menampilkan detail informasi dari lokasi guesthouse
<b>Alternate course</b>	Aksi Pengguna	Respon Sistem
	Tidak melakukan apa-apa di dalam beranda	Tidak akan ada <i>output</i> ataupun hasil dari sistem

<b>Post condition</b>	1. Kondisi berhasil Masuk ke halaman detail lokasi guesthouse dan informasinya didapatkan	2. Kondisi tidak berhasil Tidak menuju halaman detail informasi lokasi guesthouse
-----------------------	---	--

**Tabel 3.3** Narrative Use Pencarian Lokasi guesthouse dari Filter Kriteria

<b>Use case name</b>	Pencarian lokasi guesthouse dari filter kriteria	
<b>Actor</b>	Pengguna	
<b>Description</b>	<i>Use case</i> berisi penjelasan bagaimana <i>user</i> akan memperoleh pilihan lokasi guesthouse setelah melakukan inputan kriteria lokasi guesthouse dan inputan dari lokasi <i>user</i> .	
<b>Precondition</b>	Wisatawan menjalankan sistem	
<b>Typical course of event</b>	Aksi Pengguna	Respon Sistem
	Langkah 1 : Memberi inputan nilai atas kriteria-kriteria dari lokasi guesthouse sesuai keinginan <i>user</i>	Langkah 2 : Setelah inputan diterima oleh sistem, sistem akan mulai untuk menghitung jarak dari lokasi <i>user</i> ke lokasi alternatif guesthouse dengan algoritma <i>Dijkstra</i> lalu dilanjutkan dengan proses pemilihan lokasi guesthouse sesuai kriteria yang diinginkan <i>user</i> dengan menggunakan algoritma <i>ELECTRE</i>
<b>Alternate course</b>	Aksi Pengguna	Respon Sistem
	Filter <i>input</i> kriteria bermasalah atau tidak diisi dengan lengkap	Menampilkan <i>alert</i> jika data tidak lengkap
<b>Post condition</b>	1. Kondisi berhasil Inputan kriteria diterima sistem lalu diproses	2. Kondisi tidak berhasil Inputan kriteria belum lengkap sehingga tidak dapat diproses.

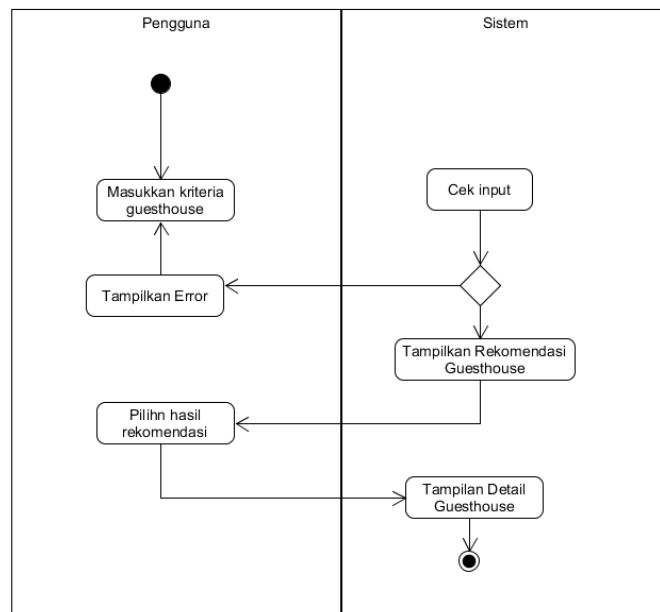
**Tabel 3.4** Narrative Use Informasi Lokasi Guesthouse dan Penginapan Terdekat

<b>Use case name</b>	Informasi Lokasi Guesthouse dan Penginapan Terdekat
<b>Actor</b>	Pengguna
<b>Description</b>	<i>Use case</i> berisi penjelasan bagaimana <i>user</i> akan memperoleh informasi detail tentang lokasi guesthouse

	hasil dari filter kriteria	
<b>Precondition</b>	Pengguna menjalankan sistem	
<b>Typical course of event</b>	Aksi Pengguna	Respon Sistem
	Langkah 1 : Setelah memasukkan kriteria dengan benar maka <i>card</i> lokasi guesthouse pilihan akan muncul dan user akan <i>click card</i> tersebut.	Langkah 2 : Setelah <i>user click card</i> tersebut maka akan muncul halaman detail informasi lokasi guesthouse dan lokasi penginapan terdekat serta <i>maps</i> menuju tempat nya.
<b>Alternate course</b>	Aksi Pengguna	Respon Sistem
	Tidak melakukan apa-apa setelah <i>card</i> hasil pencarian muncul	Tidak akan menampilkan halaman detail selanjutnya
<b>Post condition</b>	1. Kondisi berhasil <i>User</i> masuk ke dalam halaman detail informasi	2. Kondisi tidak berhasil <i>User</i> tidak <i>click</i> maka tidak akan ada respon apapun.

### 3.2.3. Activity Diagram

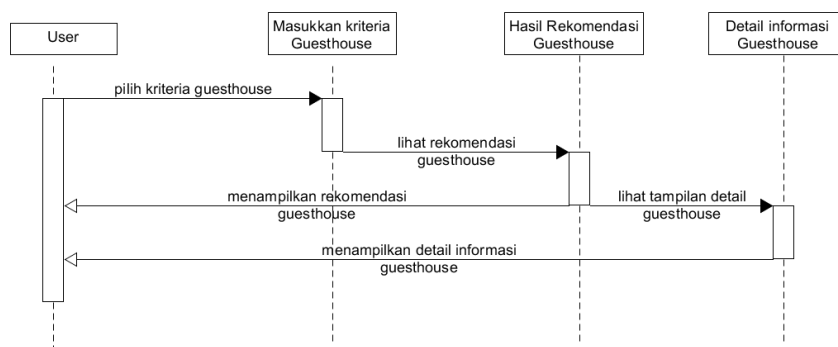
*Activity diagram* merupakan sebuah diagram aktivitas di mana merupakan gambaran *workflow* atau alur kerja dari sebuah sistem, dimulai dari mengawali, lakukan, dan mengakhiri berjalannya proses. *Activity diagram* dalam pemrosesan pemilihan lokasi guesthouse ada pada gambar 3.4 yang mana menjelaskan bahwa saat sistem dimulai wisatawan harus memasukkan nilai dari setiap kriteria sesuai yang mereka inginkan dan memasukkan lokasinya. Selanjutnya sistem akan memproses pencarian jarak antara lokasi *user* ke setiap lokasi alternatif guesthouse dengan algoritma *Dijkstra* lalu dilanjut ke pemrosesan pemilihan lokasi guesthouse sesuai kriteria *user* dengan algoritma *ELECTRE*. Lalu setelah semua diproses maka sistem akan mengeluarkan hasil akhir berupa rekomendasi terbaik alternatif guesthouse yang berisi detail informasinya.



**Gambar 3.4** Activity Diagram Sistem Pemilihan Lokasi guesthouse

### 3.2.4. Sequence Diagram

*Sequence diagram* merupakan diagram yang memberikan gambaran bagaimana suatu objek berinteraksi berpatokan pada urutan waktu dan juga memberikan gambaran urutan langkah yang dilakukan sebagai respon dari suatu kejadian untuk *output* tertentu. Untuk *sequence diagram* proses pemilihan lokasi guesthouse ada pada gambar 3.5.



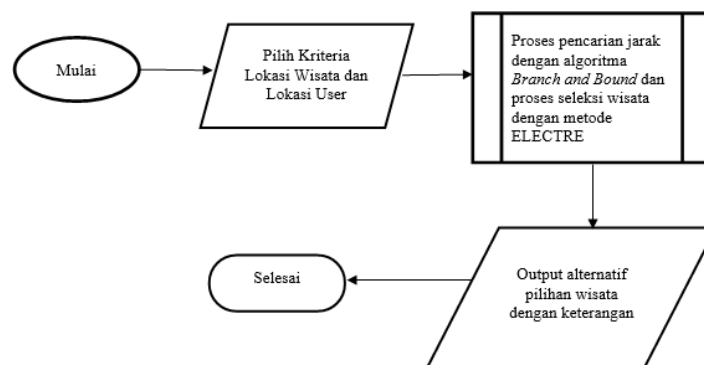
**Gambar 3.5** Sequence Diagram Sistem Pemilihan Lokasi guesthouse

### 3.3 Flowchart

*Flowchart* adalah diagram representasi grafis dari algoritma. *Flowchart* memiliki simbol-simbol yang memiliki arti tersendiri dalam membangun sebuah sistem. Berikut merupakan *flowchart* sistem yang akan dirancang :

### 3.3.1. Flowchart Sistem Pemilihan Lokasi Guesthouse

Gambar 3.6 adalah *flowchart* proses pemilihan guesthouse menggunakan algoritma ELECTRE dan proses pencarian jarak guesthouse menggunakan algoritma *Dijkstra*. Proses ini berjalan ketika pengguna memilih kriteria objek guesthouse yang diinginkan dan masukan lokasi yang diberikan oleh pengguna. Hasil dari proses ini adalah pilihan destinasi guesthouse dengan detail informasinya.

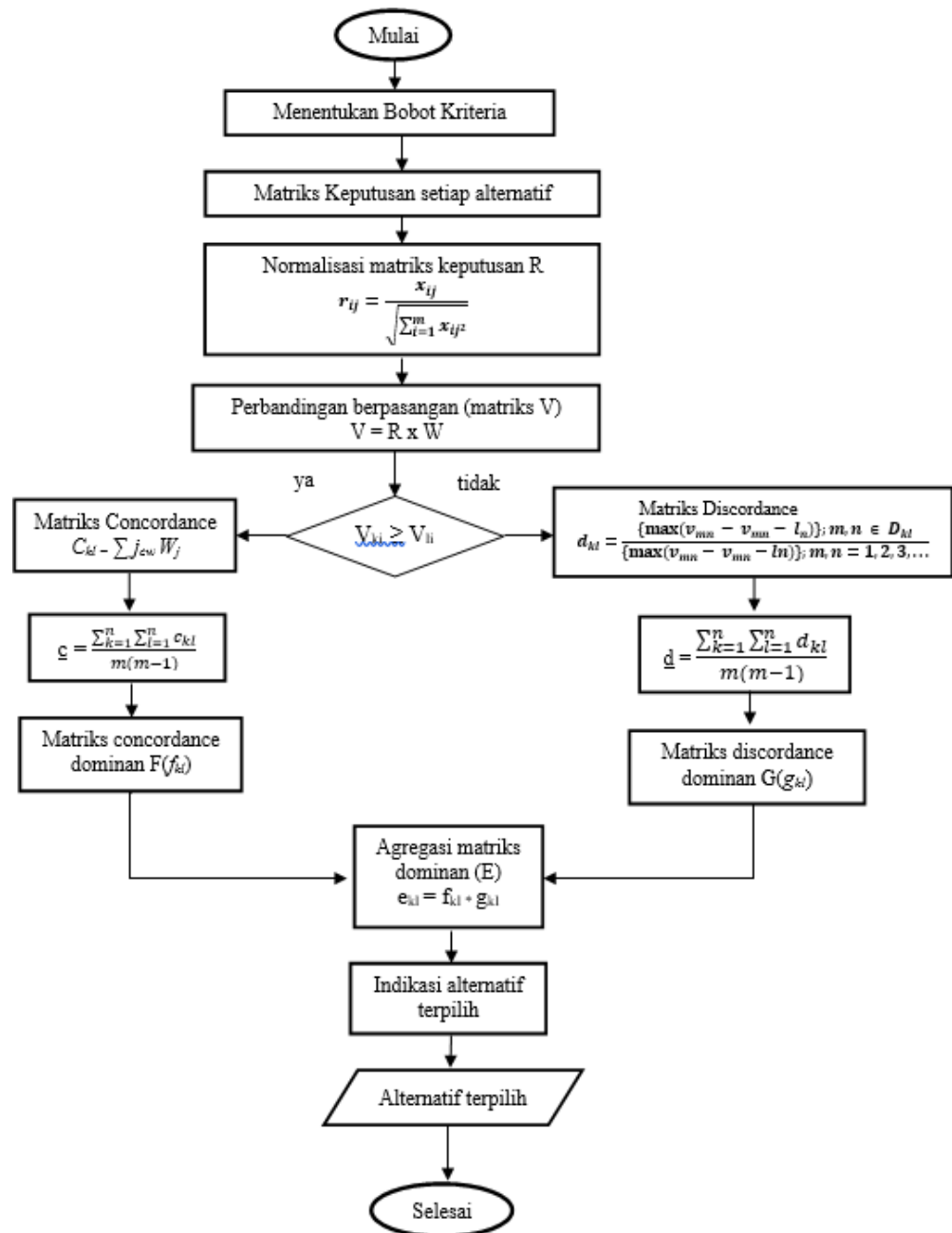


**Gambar 3.6** Flowchart Sistem Pemilihan Lokasi guesthouse

### 3.3.2. Flowchart Metode ELECTRE

*Flowchart* metode ELECTRE dapat dilihat pada Gambar 3.7, metode ini akan menghasilkan hasil keputusan lokasi guesthouse terbaik sesuai kriteria yang diinputkan oleh *user*.



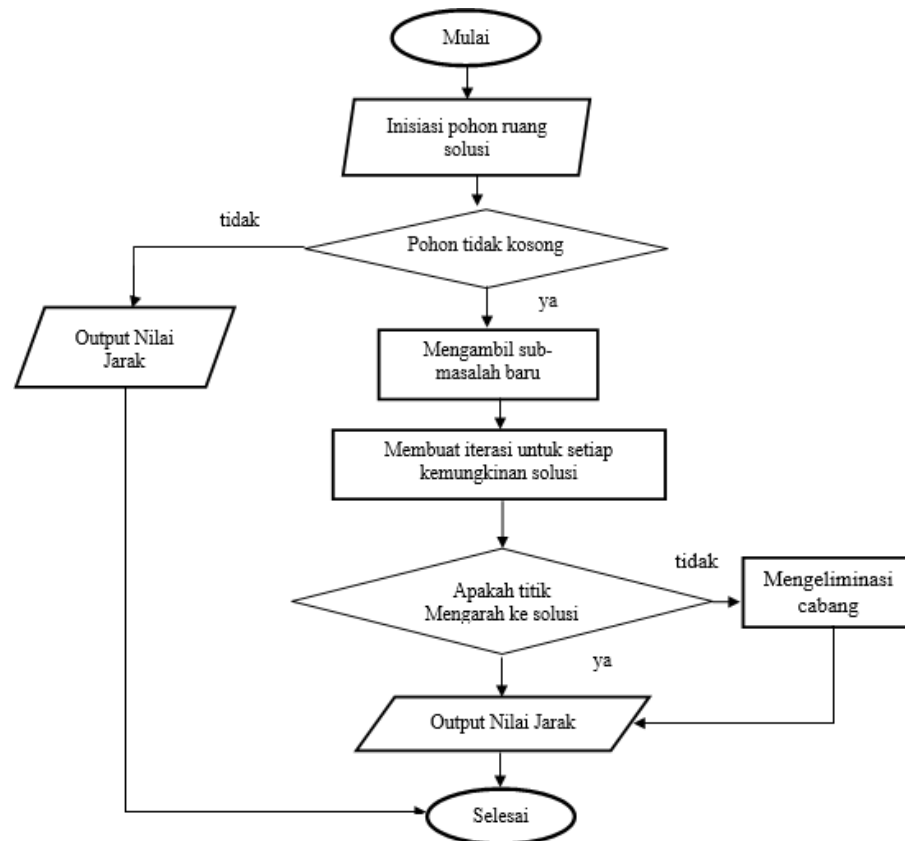


**Gambar 3.7** Flowchart Metode ELECTRE

### 3.3.3. Flowchart Algoritma Dijkstra

Gambar 3.8 menunjukkan bagaimana algoritma *Dijkstra* dimulai dengan pohon ruang solusi, mendapatkan sub-masalah baru, dan menjalankan proses pencarian untuk memeriksa apakah pohon tersebut kosong. Jika pohon kosong, output adalah solusi dan proses selesai. Jika pohon tidak kosong, sub-masalah baru akan terjadi dan iterasi akan dilakukan untuk setiap solusi yang

mungkin. Jika poin mengarah pada solusi, maka solusi terbaik diperoleh dan proses selesai, tetapi jika tidak ada solusi yang diperoleh, cabang dihilangkan dan proses selesai.



**Gambar 3.8** Flowchart Algoritma Dijkstra

### 3.4 Perancangan Antarmuka (*Interface*) Sistem

Perancangan *interface* merupakan penggambaran awal dari tampilan sistem yang akan dibuat. Tujuan dari perancangan ini adalah supaya sistem mempunyai tampilan yang *user-friendly* dan *easy to use*.

#### 3.4.1. Halaman *Splash Screen*

Halaman *Splash Screen* ini akan menjadi tampilan pertama selama beberapa detik yang muncul ketika sistem berjalan. Di dalam ini akan berisi detail judul skripsi, data diri berupa nama dan NIM penulis, logo dari universitas, dan detail

data program studi berupa nama, fakultas, serta universitas. Rancangannya ada pada gambar 3.9.

The diagram illustrates the layout of a splash screen within a rectangular frame. It contains four distinct UI elements, each labeled with a number in its top right corner:

- 1:** A rounded rectangular input field containing the text "Judul Skripsi".
- 2:** A circular placeholder containing the text "Logo Universitas".
- 3:** A rounded rectangular input field containing the text "Nama NIM".
- 4:** A rounded rectangular input field containing the text "Nama Program Studi, fakultas, universitas".

**Gambar 3.9** Rancangan Halaman *Splash Screen*

Keterangan gambar :

1. *Label*, memuat judul skripsi.
2. *Picturebox*, memuat logo Universitas Sumatera Utara.
3. *Label*, memuat nama serta NIM .
4. *Label*, memuat nama prodi, fakultas serta universitas.

### 3.4.2. Halaman Login Akun

Halaman *login akun* merupakan halaman di mana untuk masuk ke dalam sistem, *user* harus menginputkan email dan *password*-nya. Ketika *user* sudah menginputkan barulah dia dapat dialihkan ke page selanjutnya. Jika belum maka dia harus masuk ke halaman daftar. Tampilannya ada pada gambar 3.10

The image shows a wireframe of a login page. It contains the following elements with numbered annotations:

- 1**: A heading "Eksplor Destinasi baru bersama kami!" followed by a subtext "Masuk untuk pantau terus updatean destinasi menarik".
- 2**: An "Email" input field.
- 3**: A "Password" input field.
- 4**: A dark grey "Login" button.
- 5**: A link "Belum punya akun? **Daftar yuk**" at the bottom.

**Gambar 3.10** Rancangan Halaman Login Akun

Keterangan gambar :

1. *Label*, memuat ajakan untuk eksplorasi di dalam aplikasi.
2. *Textbox*, sebagai wadah untuk tempat mengisi email.
3. *Textbox*, sebagai wadah untuk tempat mengisi *password*.
4. *Button*, sebagai aksi untuk ditekan *user* jika semua data sudah benar untuk diproses ke halaman selanjutnya menuju halaman beranda aplikasi.

### 3.4.3. Halaman Daftar Akun

Halaman daftar akun merupakan halaman di mana *user* harus menginputkan data mereka seperti nama, email, dan *password* agar bisa masuk ke dalam sistem, dan dialihkan ke beranda aplikasi. Rancangan tampilannya ada pada gambar 3.11.

The image shows a mobile app wireframe for a registration page. It features a close button (X) at the top left. The main heading is 'Eksplor Destinasi baru bersama kami!' with a callout number 1. Below it is a subtext: 'Buat akun untuk eksplor dan dapatkan updatean terbaru'. There are three input fields: 'Nama' (callout 2), 'Email' (callout 3), and 'Password' (callout 4). A dark grey button labeled 'Buat Akun' is below the fields (callout 5). At the bottom, there is a link 'Udah punya akun? Log in aja!' with a callout number 6.

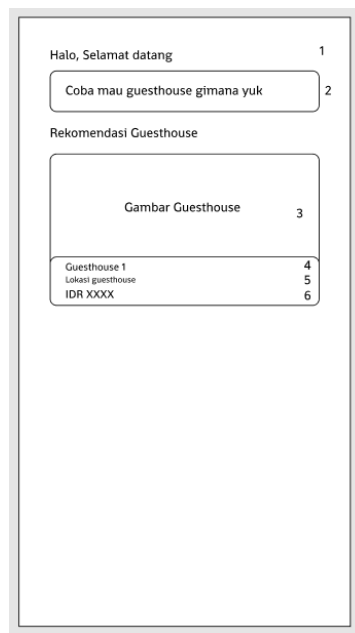
**Gambar 3.11** Rancangan Halaman Daftar Akun

Keterangan gambar :

1. *Label*, memuat ajakan untuk eksplorasi dalam aplikasi.
2. *Textbox*, sebagai wadah untuk mengisi nama..
3. *Textbox*, sebagai wadah untuk mengisi email.
4. *Textbox*, sebagai wadah untuk mengisi *password*.
5. *Button*, sebagai aksi untuk ditekan *user* jika semua data sudah terisi dan sudah benar untuk diproses ke halaman *login*.

#### 3.4.4. Halaman Beranda

Halaman beranda merupakan halaman di mana *user* akan disuguhkan bar untuk melakukan pencarian lokasi guesthouse, *user* juga akan disuguhkan dengan *card-card* pilihan guesthouse secara acak. Dari *page* ini *user* dapat melakukan pencarian dari filter bar yang ada atau langsung *click card-card* yang disediakan. Rancangan tampilannya ada pada gambar 3.12.



**Gambar 3.12** Rancangan Halaman Beranda

Keterangan gambar :

1. *Textbox*, memuat ajakan user untuk eksplorasi dalam aplikasi.
2. *FilterBar*, sebagai aksi untuk ditekan jika *user* ingin mencari lokasi guesthouse sesuai inputan keinginannya.
3. *Imageview* sebagai tempat gambar dari alternatif lokasi guesthouse yang akan dimuat dari *database*.
4. *Textbox*, sebagai wadah nama lokasi guesthouse.
5. *Textbox*, sebagai wadah alamat dari lokasi guesthouse.
6. *Textbox*, sebagai wadah menampilkan harga tiket masuk lokasi guesthouse.

### 3.4.5. Halaman *Bottomsheet* Filter Pemilihan Kriteria

Halaman *Bottomsheet* Filter Pemilihan Kriteria ini menjadi tempat untuk *user* memberikan inputannya. Di dalam nya ada tempat untuk menginputkan lokasi dari *user*, lalu akan ada beberapa kriteria yang harus diinputkan *user* yaitu jarak ke lokasi yang diinginkan, harga tiket masuk lokasi guesthouse, harga

penginapan terdekat, dan harga *guide* yang diinginkan. Semua inputan tersebut sesuai keinginan dari *user*. Rancangan untuk tampilannya ada pada gambar 3.13.

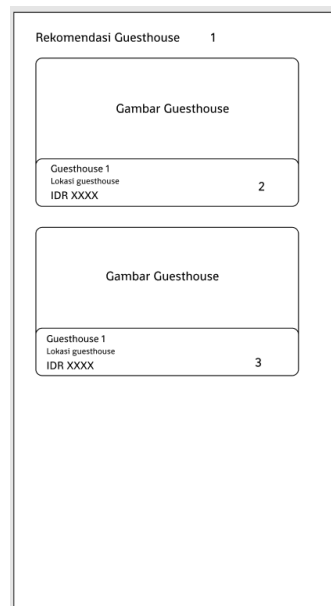
**Gambar 3.13** Rancangan *Bottomsheet* Filter Pemilihan Kriteria

Keterangan Gambar :

1. *Closebutton*, sebagai aksi *user* jika tidak ingin menggunakan pemilihan kriteria.
2. *Radiobutton*, sebagai pilihan lokasi guesthouse untuk *user* yang harus diinputkan.
3. *Slider*, sebagai tempat inputan jarak lokasi yang diinginkan user.
4. *Slider*, sebagai tempat inputan harga kamar penginapan per malam.
5. *Slider*, sebagai tempat inputan total fasilitas.
6. *Slider*, sebagai tempat inputan guesthouse rating.
7. *Button*, sebagai aksi untuk ditekan *user* jika semua data kriteria sudah diinputkan dan sesuai. Jika inputan sudah terisi *user* akan dialihkan ke halaman hasil pencarian lokasi sesuai kriteria.

### 3.4.6. Halaman Hasil Pencarian Sesuai Kriteria

Halaman Hasil Pencarian Sesuai Kriteria ini merupakan halaman yang akan muncul setelah semua inputan terisi dengan benar. Di halaman ini akan memunculkan 3 pilihan *card* alternatif lokasi guesthouse terbaik setelah proses yang dikerjakan oleh sistem. Tampilan untuk halaman ini ada pada gambar 3.14.



**Gambar 3.14** Halaman Hasil Pencarian Sesuai Kriteria

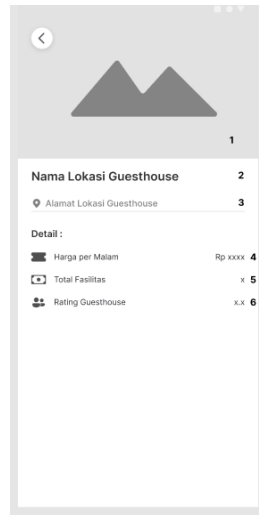
Keterangan gambar :

1. *Back Button*, sebagai aksi jika *user* ingin mengubah inputan kriteria.
2. *Card*, berisi informasi dari lokasi guesthouse pilihan urutan pertama.
3. *Card*, berisi informasi dari lokasi guesthouse pilihan urutan kedua.
4. *Card*, berisi informasi dari lokasi guesthouse pilihan urutan ketiga.



### 3.4.7. Halaman Detail Informasi Lokasi Guesthouse

Halaman detail informasi lokasi guesthouse ini merupakan halaman yang berisi 3 pilihan rekomendasi guesthouse yang telah sesuai dengan kriteria yang *user* inputkan. Dan 3 pilihan rekomendasi guesthouse ini merupakan hasil dari proses yang telah dikerjakan sistem. Untuk rancangan tampilan nya ada pada gambar 3.15.



**Gambar 3.15** Halaman Detail Informasi Lokasi Guesthouse

Keterangan gambar :

1. *Imageview*, berisi gambar dari lokasi guesthouse.
2. *Textbox*, berisi nama dari lokasi guesthouse.
3. *Textbox*, berisi alamat dari lokasi guesthouse.
4. *Textbox*, berisi harga kamar per malam.
5. *Textbox*, berisi total fasilitas guesthouse.
6. *Textbox*, berisi rating guesthouse.

### 3.5 Perancangan Database

Untuk *database management system* yang digunakan pada sistem aplikasi adalah MySQL. Untuk tabel *database* yang akan dibuat ada pada tabel 3.5, 3.6, dan 3.7.

Nama *Database* : db\_skripsi.

**Tabel 3.5** Tabel List Guesthouse

Nama Field	Deskripsi	Tipe Data	Keterangan
id_guesthouse	Urutan data guesthouse	INTEGER	PRIMAR YKEY
Name	Nama dari guesthouse	VARCHA R	
Address	Alamat dari guesthouse	VARCHA R	
price	Harga per malam setiap guesthouse	VARCHA R	
Facility	Fasilitas yang disediakan di setiap guesthouse	VARCHA R	
Image	Gambar dari guesthouse	VARCHA R	
Rating	Rating dari setiap guesthouse	VARCHA R	

**Tabel 3.6** Tabel Data User

Nama Field	Deskripsi	Tipe Data	Keterangan
id	Urutan data user	INTEGER	PRIMARY KEY
name	Inputan nama user	VARCHAR	
email	Inputan email user	VARCHAR	
password	Inputan password user	VARCHAR	

**Tabel 3.7** Tabel Data Jarak

<b>Nama Field</b>	<b>Deskripsi</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id_jarak	Urutan data jarak	INTEGER	PRIMARY KEY
titik_awal	Titik awal tempat	INTEGER	
titik_tujuan	Titik akhir tempat	INTEGER	
bobot_jarak	Nilai dari jarak titik awal ke titik tujuan	INTEGER	

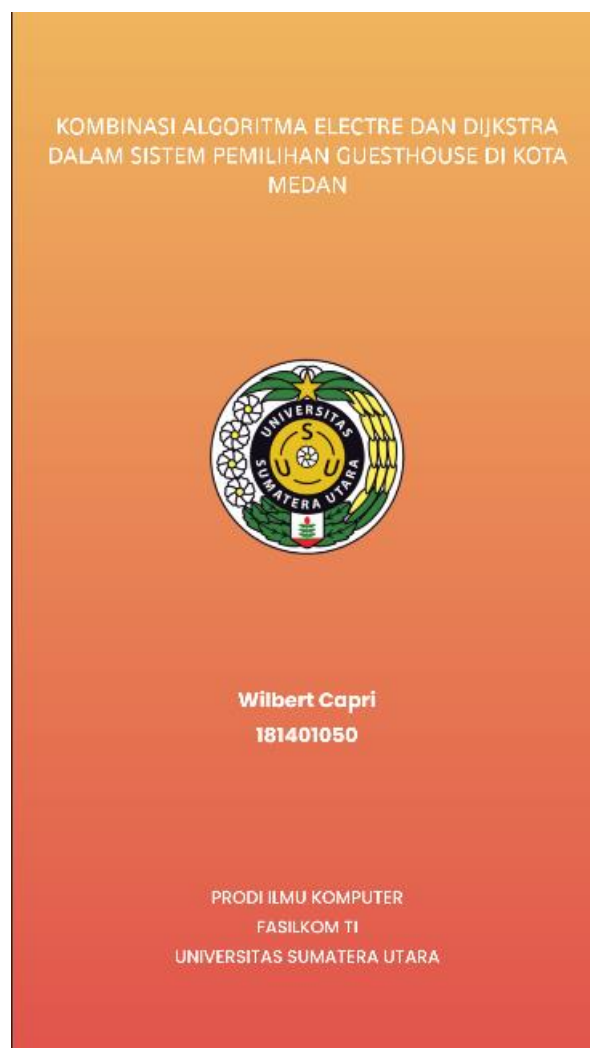
## BAB IV IMPLEMENTASI DAN PENGUJIAN

### 4.1 Implementasi Sistem

Di dalam implementasi sistem ini merupakan tahap dalam perealisasi rancangan desain dari sistem ke dalam bentuk program di mana ini adalah tahap akhir dari perancangan sistem aplikasi tersebut. Pada pengimplementasian di sini akan memakai bahasa pemrograman Java dan memakai MySQL sebagai database untuk tempat menyimpan data lokasi guesthouse, data *user*, serta data jarak. Di dalam sistem ini *user* akan dapat melihat informasi guesthouse langsung dari *card* setiap guesthouse atau mencari dari filter kriteria yang telah disediakan.

#### 4.1.1. Halaman *Splash Screen*

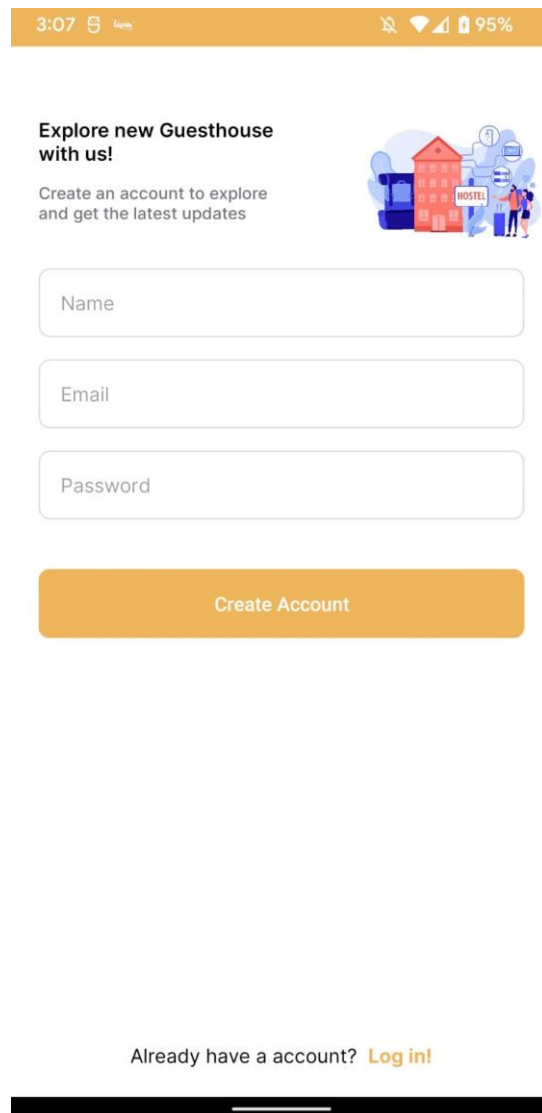
Implementasi halaman *Splash Screen* ada pada gambar 4.1. Ini merupakan tampilan yang muncul saat pertama sistem berjalan. Di dalamnya ada judul sistem dari skripsi, logo universitas, data penulis, prodi, fakultas, serta universitas.



**Gambar 4.1** Halaman *Splash Screen*

#### 4.1.2. Halaman Daftar

Implementasi halaman daftar ada pada gambar 4.2. Di mana di halaman ini *user* harus memasukkan nama, email, dan *password*. Jika sudah benar maka *user* harus menekan *button* buat akun.



3:07 5 lives

Explore new Guesthouse with us!

Create an account to explore and get the latest updates

Name

Email

Password

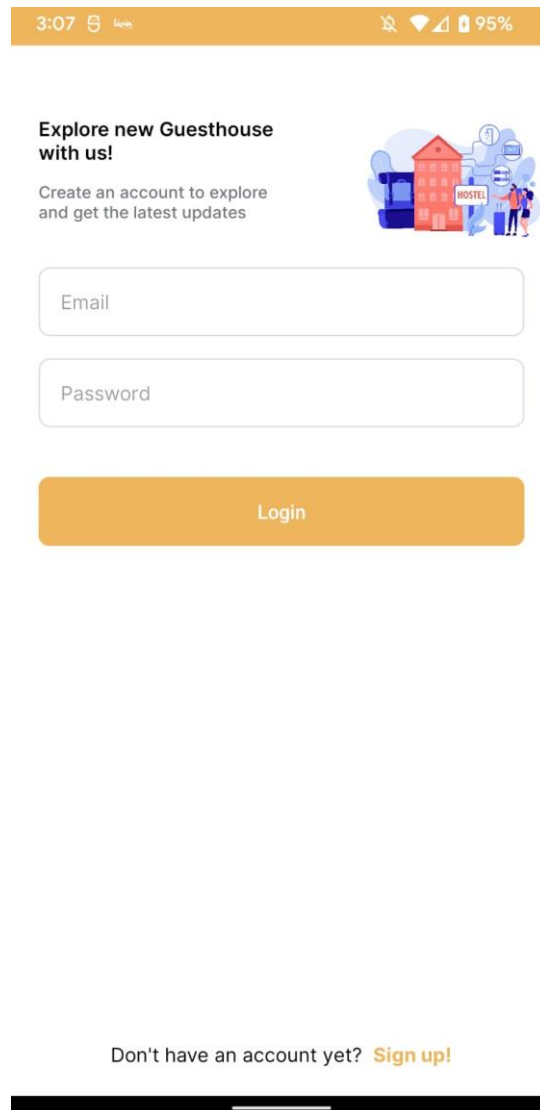
Create Account

Already have a account? [Log in!](#)

**Gambar 4.2** Halaman Daftar

#### 4.1.3. Halaman Login

Implementasi halaman *login* ada pada gambar 4.3. Di dalam halaman ini *user* yang telah melakukan pendaftaran harus memasukkan email dan *password*. Jika sudah benar maka akan langsung dialihkan ke beranda jika menekan *button login*.



3:07 95%

**Explore new Guesthouse with us!**

Create an account to explore and get the latest updates

Email

Password

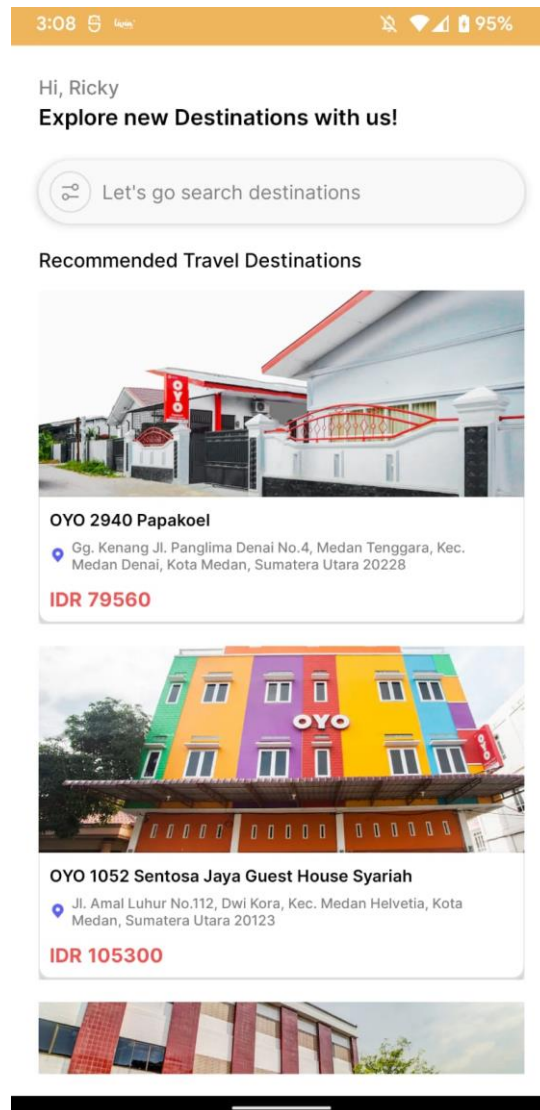
Login

Don't have an account yet? [Sign up!](#)

**Gambar 4.3** Halaman Login

#### 4.1.4. Halaman Beranda

Implementasi halaman beranda ada pada gambar 4.4. Di dalam halaman ini *user* akan disuguhkan *card-card* yang berisi informasi lokasi guesthouse dan disuguhkan secara *random* dari *database*. Sedangkan untuk *user* yang ingin mencari sesuai kriteria yang diinginkan harus meng-klik bagian filter bar.



**Gambar 4.4** Halaman Beranda

#### 4.1.5. Halaman *Bottomsheet* Filter Pemilihan Kriteria

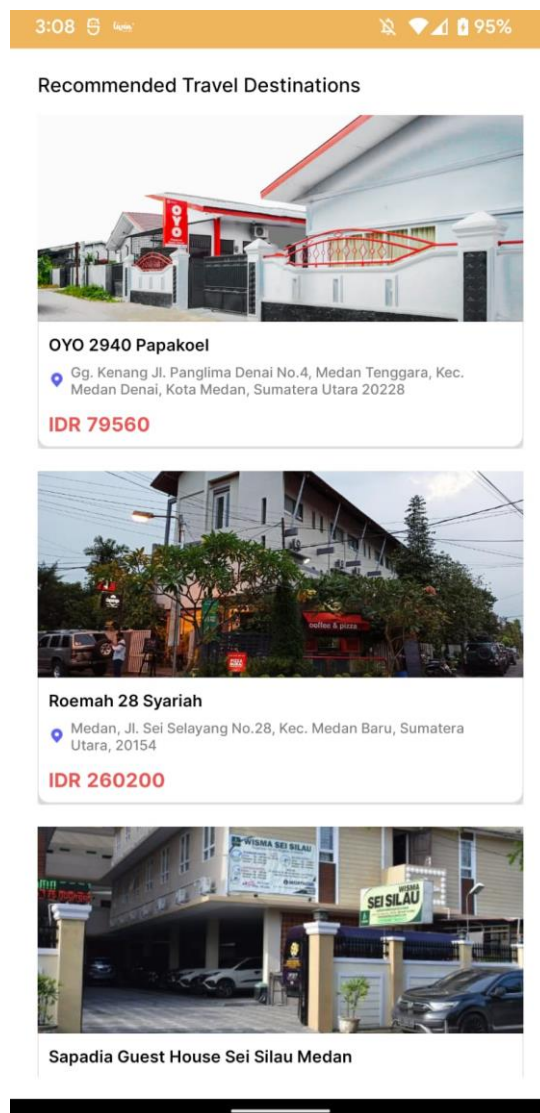
Implementasi halaman *bottomsheet* filter pemilihan kriteria ada pada gambar 4.5. Di dalam halaman ini *user* akan menginputkan sesuai lokasi dan kriteria dari guesthouse yang diinginkannya. Ada beberapa hal yang harus diisi *user* yaitu lokasi *user*, jarak lokasi, harga per malam guesthouse, total fasilitas, dan rating.

**Gambar 4.5** Halaman *Bottomsheet* Pemilihan Kriteria



#### 4.1.6. Halaman Hasil Pencarian Sesuai Kriteria

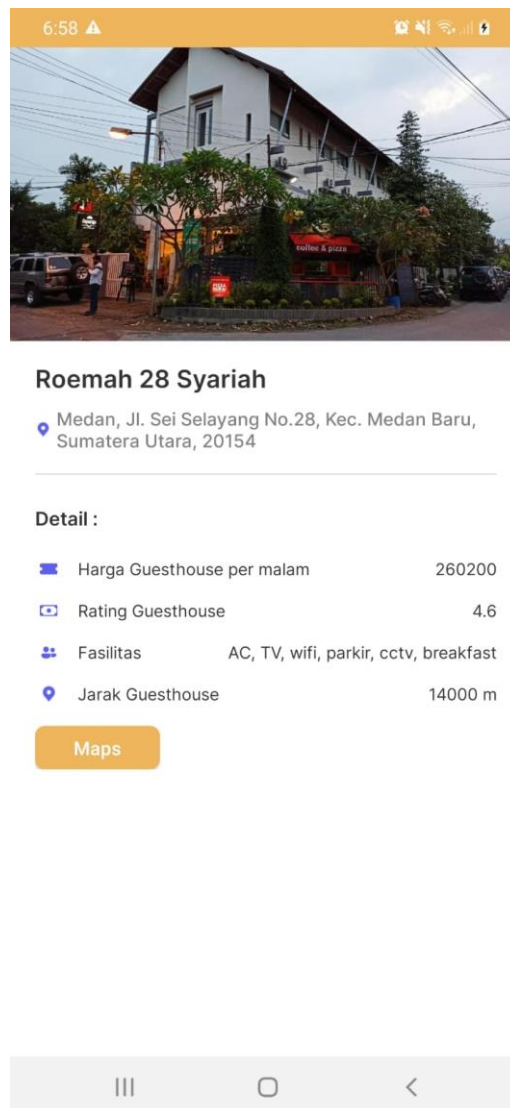
Implementasi halaman hasil pencarian sesuai kriteria ini ada pada gambar 4.6. Di dalam halaman ini *user* disuguhkan hasil dari pencarian lokasi guesthouse sesuai kriteria yang dia inginkan. Di sini akan ditampilkan 5 *card* hasil dari proses sistem yang menggunakan algoritma *Dijkstra* dan metode ELECTRE untuk mencari hasil yang paling sesuai dengan kriteria yang diinginkan *user*.



**Gambar 4.6** Halaman Hasil Pencarian Sesuai Kriteria

#### 4.1.7. Halaman Detail Informasi Lokasi Guesthouse

Implementasi halaman detail informasi lokasi guesthouse ini ada pada gambar 4.7. Di dalam halaman ini *user* disuguhkan informasi detail dari lokasi guesthouse yang dipilih setelah diklik. Di dalam halaman ini ada gambar dari lokasi guesthouse, nama lokasi guesthouse, alamat lokasi guesthouse, harga per malam, harga rating guesthouse, total fasilitas, dan jarak dari user ke guesthouse tersebut.



**Gambar 4.7** Halaman Detail Informasi Lokasi Guesthouse

## 4.2 Pengujian

Pengujian dari sistem aplikasi ini adalah tahap selanjutnya setelah implementasi. Pengujian ini dilakukan agar dapat melihat dan melakukan pembuktian sistem yang dibentuk mampu berjalan dengan baik dan memberi rekomendasi sesuai inputan *user*. Pengujian ini dilakukan dengan batasan 20 guesthouse dan 10 titik lokasi *user* di kota Medan dengan algoritma Dijkstra dan metode ELECTRE.

### 4.2.1. Pengujian Sistem Rekomendasi Alternatif Guesthouse

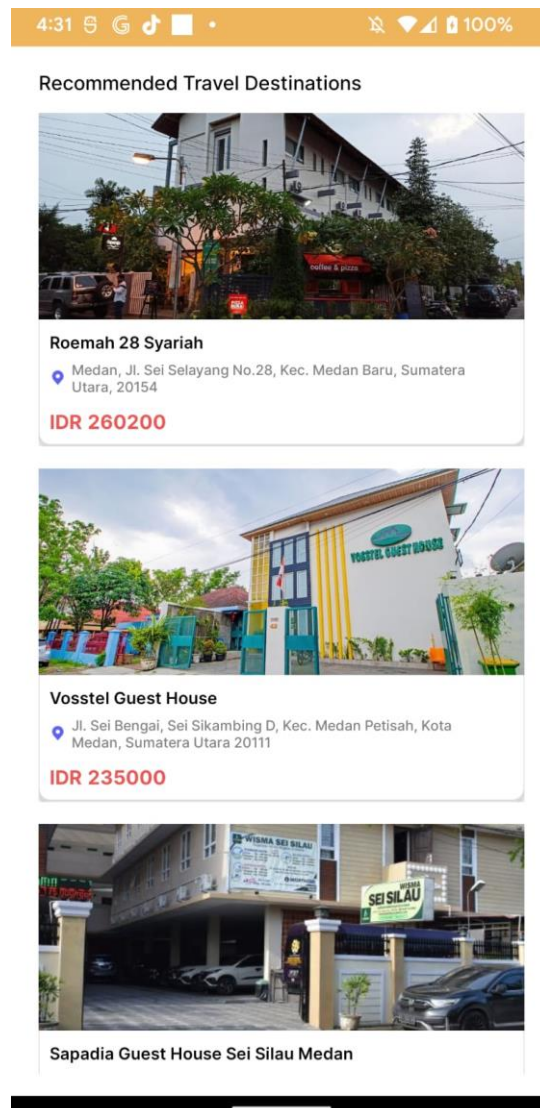
Dengan dukungan sistem pendukung keputusan multikriteria yang menggunakan metode ELECTRE, akan sangat berguna untuk memberikan hasil rekomendasi terbaik untuk guesthouse. Dan dengan algoritma *Dijkstra* yang diterapkan akan dapat menghitung pencarian simpul berdasarkan *cost* dari setiap node. Ada 2 kali pengujian pada sistem dengan inputan kriteria yang berbeda.

#### Pengujian 1

Untuk hasil dari pengujian sistem pertama kali ada pada gambar 4.8 dan 4.9

The image shows a mobile application interface for guesthouse recommendations. It is divided into two main panels. The left panel, titled 'Where is your location?', lists 10 locations: Merdeka Walk, Medan Fair (selected), Universitas Sumatera Utara, Aksara, Ramayana Pringgane, Simalingkar, Pajak Melati, Pinang Baris Terminal, Kantor Polisi Belawan, and J-City. Below the list is a slider for 'Distance to the location you want' ranging from 0 km to 35 km, with a marker at 8 km. The right panel shows filters for 'Guesthouse Price per night' (IDR 0 - 260,000) with a marker at IDR 211,000, 'Total Facilities' (1 - 5) with a marker at 3, and 'Guesthouse Rating' (2.5 - 5) with a marker at 4.25. A 'Let's search!' button is located at the bottom right of the right panel.

**Gambar 4.8** Pengujian 1 Berupa Input Kriteria dan Lokasi *User*



**Gambar 4.9** Hasil Rekomendasi Guesthouse

#### 4.2.2. Pengujian Manual Algoritma Dijkstra

Berikut data lokasi awal user dan data jarak dari setiap lokasi awal user dan lokasi *user* ada pada tabel 4.1 dan 4.2.

**Tabel 4.1** Data Lokasi *User*

No	Nama Lokasi
1	Stasiun Kereta Api Medan
2	Pool Damri Medan Fair
3	USU
4	Aksara
5	Ramayana Pringgau
6	Simalingkar
7	Pajak Melati
8	Terminal Pinang Baris
9	Pelabuhan Belawan
10	J-City
11	Terminal Ampas
12	Bandara Kualanamu
13	Halte Transdeli SM Raja
14	PT ALS Bus
15	Halte Transdeli Jamin Ginting

**Tabel 4.2** Data Jarak

<b>No</b>	<b>Titik Awal</b>	<b>Titik Tujuan</b>	<b>Bobot Jarak (m)</b>
1	OYO 92429 Vania Guest House	Pajak melati	3300
2	OYO 92429 Vania Guest House	Johje Guest House Medan RedPartner	2800
3	Pajak melati	Super OYO 2574 Z Suites Hotel	6400
4	Pajak melati	Johje Guest House Medan RedPartner	5500
5	Johje Guest House Medan RedPartner	Super OYO 2574 Z Suites Hotel	7600
6	Johje Guest House Medan RedPartner	RedDoorz @ Joyce Guest House Medan	4500
7	Johje Guest House Medan RedPartner	Simalingkar	6300
8	Super OYO 2574 Z Suites Hotel	OYO 92342 Hotel Batik Traveller	2500
9	Super OYO 2574 Z Suites Hotel	RedDoorz @ Joyce Guest House Medan	4800
10	RedDoorz @ Joyce Guest House Medan	Sapadia Guest House Sei Silau Medan	4500
11	RedDoorz @ Joyce Guest House Medan	J-City	5200
12	RedDoorz @ Joyce Guest House Medan	Simalingkar	8050
13	Simalingkar	J-City	5600
14	J-City	Roemah 54	10400
15	J-City	Tobali Guest House	9800
16	OYO 92342 Hotel Batik Traveller	OYO 1585 Ring Road Residence	2700
17	OYO 92342 Hotel Batik Traveller	Sapadia Guest House Sei Silau Medan	1100
18	Sapadia Guest House Sei Silau Medan	Roemah 54	2500
19	Roemah 54	OYO 1585 Ring Road Residence	3400
20	Roemah 54	Roemah 28 Syariah	2400
21	Roemah 54	USU	3400
22	Tobali Guest House	USU	5100

23	Tobali Guest House	Adam Malik Guesthouse near Regale ICC Medan RedPartner	5300
24	USU	Merdeka walk	5800
25	USU	Ramayan pringgan	4600
26	USU	Adam Malik Guesthouse near Regale ICC Medan RedPartner	5200
27	Adam Malik Guesthouse near Regale ICC Medan RedPartner	Merdeka walk	2800
28	Adam Malik Guesthouse near Regale ICC Medan RedPartner	OYO 2147 Mono Guest House	2800
29	Adam Malik Guesthouse near Regale ICC Medan RedPartner	OYO 2940 Papakoel Guest House	10600
30	Ramayan pringgan	Roemah 28 Syariah	3400
31	Ramayan pringgan	Vosstel Guest House	1100
32	Roemah 28 Syariah	Medan fair	3700
33	Vosstel Guest House	Medan fair	1100
34	Medan fair	OYO 1052 Sentosa Jaya Guest House Syariah	4300
35	Medan fair	SUPER OYO 90874 Millenium Inn 2	4100
36	OYO 1052 Sentosa Jaya Guest House Syariah	OYO 91706 Hotel Serena Anggrek	2600
37	SUPER OYO 90874 Millenium Inn 2	OYO 91706 Hotel Serena Anggrek	4200
38	OYO 1585 Ring Road Residence	OYO 1052 Sentosa Jaya Guest House Syariah	3800
39	OYO 1585 Ring Road Residence	Terminal pinang baris	4400
40	OYO 1052 Sentosa Jaya Guest House Syariah	Terminal pinang baris	4600
41	OYO 91706 Hotel Serena Anggrek	Terminal pinang baris	3000
42	OYO 1585 Ring Road Residence	Terminal pinang baris	4400
43	Vosstel Guest House	Merdeka walk	3600
44	Vosstel Guest House	Super OYO 1457 Tmj Guest House	5000
45	Vosstel Guest House	OYO 91224 Brayan Guest House	6800
46	Vosstel Guest House	OYO 2229 Bunga Raya Residence	6500

47	Kantor polisi belawan	OYO 2229 Bunga Raya Residence	21800
48	OYO 2229 Bunga Raya Residence	OYO 91224 Brayan Guest House	1200
49	OYO 91224 Brayan Guest House	Super OYO 457 tmj Guestouse	5100
50	Super OYO 1457 Tmj Guest House	Merdeka walk	2300
51	Merdeka walk	OYO 2147 Mono Guest House	3400
52	Super OYO 1457 Tmj Guest House	OYO 2147 Mono Guest House	1800
53	OYO 2147 Mono Guest House	OYO 2940 Papakoel Guest House	7700
54	Aksara	Kantor polisi belawan	20400
55	Aksara	OYO 2229 Bunga Raya Residence	2400
56	Aksara	OYO 91224 Brayan Guest House	2000
57	Aksara	Super OYO 1457 Tmj Guest House	4500
58	Terminal Amplas	OYO 92429 Vania Guest House	8600
59	Terminal Amplas	Bandara Kualanamu	29300
60	Bandara Kualanamu	OYO 2940 Papakoel Guest House	31800
61	OYO 2940 Papakoel Guest House	Halte Transdeli SM Raja	4300
62	Halte Transdeli SM Raja	Sapadia Guest House Sei Silau Medan	1800
63	Sapadia Guest House Sei Silau Medan	PT ALS Bus	7400
64	Halte Transdeli Jamin Ginting	J-City	9600

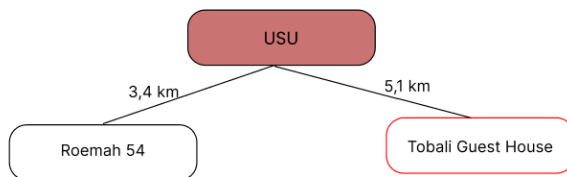
Lokasi *user* adalah di **USU**, dan perlu untuk menghitung jarak dari USU ke setiap guesthouse dengan algoritma Dijkstra dimana untuk data jarak dari graph dapat dilihat pada tabel 4.2 diatas. Berikut proses perhitungan jarak memakai algoritma Dijkstra .

Proses mencari jarak dari **USU** menuju **Johje Guest House**:



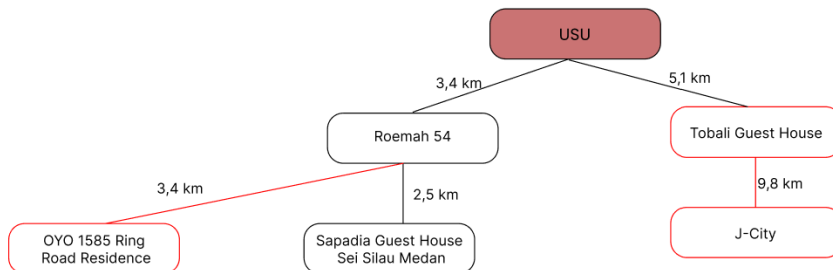
**Langkah 1 :**

Titik terpilih : USU

**Langkah 2 :**

USU – Roemah 54 = 3,4 km

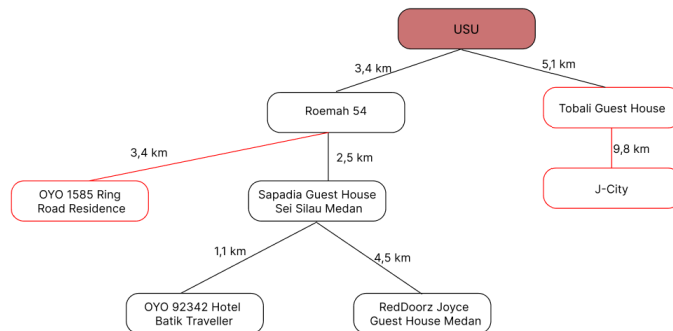
USU – Tobali Guesthouse = 5,1 km

**Langkah 3 :**

USU – Roemah 54- OYO 1585 Ring Road Residence = 3,4 + 3,4 = 6,8 km

USU – Roemah 54 – Sapadia Guesthouse = 3,4 + 2,5 = 5,9 km

USU – Tobali Guesthouse – J-City = 5,1 + 9,8 = 14,9 km (Jarak terlalu besar)

**Langkah 4:**

$$\text{USU} - \text{Roemah 54} - \text{OYO 1585 Ring Road Residence} = 3,4 + 3,4 = 6,8 \text{ km}$$

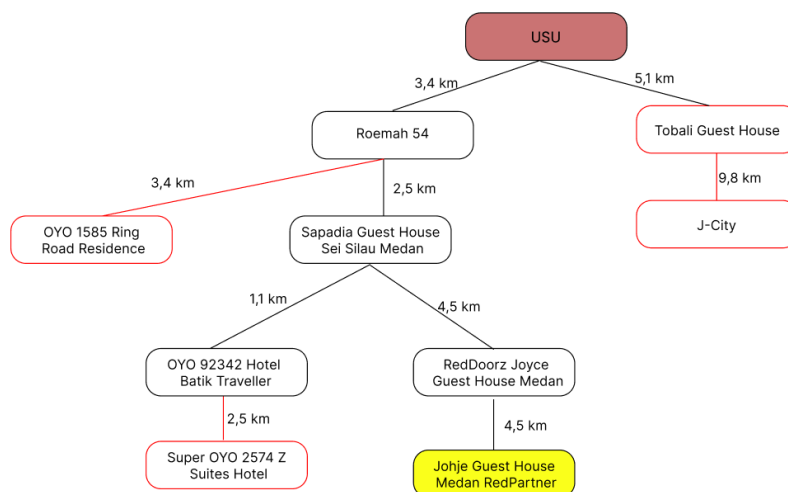
$$\text{USU} - \text{Roemah 54} - \text{Sapadia Guesthouse} - \text{Hotel Batik Traveller}$$

$$= 3,4 + 2,5 + 1,1 = 7 \text{ km}$$

$$\text{USU} - \text{Roemah 54} - \text{Sapadia Guesthouse} - \text{RedDoorz Joyce Guest House}$$

$$= 3,4 + 2,5 + 4,5 = 10,4 \text{ km}$$

$$\text{USU} - \text{Tobali Guesthouse} - \text{J-City} - \text{Simalingkar} = 5,1 + 9,8 + 5,2 = 20,1 \text{ km}$$

**Langkah 5:**

$$\text{USU} - \text{Roemah 54} - \text{OYO 1585 Ring Road Residence} = 3,4 + 3,4 = 6,8 \text{ km}$$

$$\text{USU} - \text{Roemah 54} - \text{Sapadia Guesthouse} - \text{Hotel Batik Traveller} - \text{Super}$$

$$\text{OYO 2574 Z Suites}$$

$$= 3,4 + 2,5 + 1,1 + 2,5 = 9,5 \text{ km}$$

USU – Roemah 54 – Sapadia Guesthouse - RedDoorz Joyce Guest House –  
**Johje Guesthouse**

$$= 3,4 + 2,5 + 4,5 + 4,5 = 14,9 \text{ km (rute terpendek)}$$

Semua node yang memungkinkan telah ditelusuri, dan sisanya telah dieliminasi, sehingga pencarian sudah ditemukan dengan total cost 14,9 km atau 14900 meter. Jarak dari USU ke Johje guesthouse ditempuh dengan 14,9 km. Hal berikut dilakukan ke setiap lokasi guesthouse yang lainnya. Untuk data range jarak ada pada tabel 4.3 dan tabel 4.4

**Tabel 4.3** Data Bobot Jarak

Range Jarak (K1)	Bobot
0 – 7000	1
7100 – 14000	2
14100 – 21000	3
21100 – 28000	4
28100 - 35000	5

Tabel diatas menyatakan bobot dari jarak lokasi user ke destinasi guesthouse yang diinginkan.

**Tabel 4.4** Data Jarak dari USU ke Seluruh Lokasi Guesthouse

No	Lokasi Awal	Tujuan	Jarak (m)	Bobot
1		OYO 2940 Papakoel Guest House	15700	3
2		OYO 1052 Sentosa Jaya Guest House Syariah	10600	2
3		OYO 91224 Brayan Guest House	9400	2
4		OYO 92429 Vania Guest House	9600	2
5		OYO 2147 Mono Guest House	7700	2
6		OYO 2229 Bunga Raya Residence	8600	2
7		Johje Guest House Medan RedPartner	9200	2

8	USU	Super OYO 2574 Z Suites Hotel	5400	1
9		RedDoorz @ Joyce Guest House Medan	5700	1
10		Adam Malik Guesthouse near Regale ICC Medan RedPartner	5500	1
11		Roemah 28 Syariah	1900	1
12		Super OYO 1457 Tmj Guest House	7100	2
13		SUPER OYO 90874 Millenium Inn 2	6900	1
14		OYO 1585 Ring Road Residence	5900	1
15		Roemah 54	3500	1
16		OYO 91706 Hotel Serena Anggrek	7200	2
17		Vosstel Guest House	3200	1
18		Sapadia Guest House Sei Silau Medan	2700	1
19		Tobali Guest House	5100	1
20		OYO 92342 Hotel Batik Traveller	3300	1

Semua nilai jarak terdekat dari lokasi inputan user yaitu USU ke setiap lokasi guesthouse sudah ditemukan, selanjutnya interval range dari setiap jarak tersebut akan dimasukkan selanjutnya ke dalam perhitungan metode ELECTRE.

#### 4.2.3. Pengujian Manual menggunakan metode ELECTRE

Setelah selesai melakukan penghitungan jarak maka, akan dilanjutkan dengan metode ELECTRE yang diterapkan pada sistem. Metode ini digunakan untuk menentukan rekomendasi guesthouse berdasarkan kriteria yang telah diinputkan *user*.

Pengujian Manual di sini dilakukan sesuai hasil dari pengujian sistem ke 1 untuk membandingkan hasil. Data bobot dari setiap kriteria dapat dilihat pada tabel di bawah ini

**Tabel 4.5** Data Bobot Harga guesthouse per malam

<b>Range Harga Guesthouse(K2)</b>	<b>Bobot</b>
0 – 20000	1
20001 - 40000	2
40001 – 60000	3
60001 – 80000	4
80001 - 100000	5

Tabel diatas menyatakan bobot dari harga per malam pada setiap guesthouse.

**Tabel 4.6** Data Bobot Rating Guesthouse Terdekat

<b>Range Rating (K3)</b>	<b>Bobot</b>
0 – 3.0	1
3.1 – 3.5	2
3.6 – 4.0	3
4.1 – 4.5	4
>4.5	5

Tabel diatas menyatakan bobot dari rating pada setiap guesthouse.

**Tabel 4.7** Data Bobot Total Fasilitas

<b>Total Fasilitas (K4)</b>	<b>Bobot</b>
AC, TV	1
AC, TV, Wifi	2
AC, TV, Wifi, Parkir	3
AC, TV, Wifi, Parkir, CCTV	4
AC, TV, Wifi, Parkir, CCTV, Breakfast	5

Tabel diatas menyatakan bobot dari fasilitas yang disediakan pada setiap guesthouse.

Inputan yang dimasukkan oleh *user* di pengujian pertama :

- Lokasi *user* : USU
- Jarak lokasi Guesrhousse yang diinginkan (K1) : 8km
- Harga guesthouse per malam(K2) : 210000
- Rating guesthouse(K3) : 4.25
- Total fasilitas (K4) : 3

Sebelum memasuki tahap metode ELECTRE, inputan *user* dibuat menjadi bobot pengambilan keputusan, tabel bobot inputan *user* ada pada tabel 4.8:

**Tabel 4.8** Bobot Pengambilan Keputusan

Kriteria	Bobot
K1	3
K2	2
K3	4
K4	3

Inputan yang dimasukkan oleh *user* di pengujian pertama:

- Lokasi *user* : USU
- Jarak lokasi Guesrhouse yang diinginkan (K1) : 8km (2)
- Harga guesthouse per malam(K2) : 210000 (2)
- Rating guesthouse(K3) : 4.25 (4)
- Total fasilitas (K4) : 3 (3)

Sebelum memasuki tahap metode ELECTRE, inputan *user* dibuat menjadi bobot pengambilan keputusan, tabel bobot inputan *user* ada pada tabel 4.8 :

**Tabel 4.8** Bobot Pengambilan Keputusan

Kriteria	Bobot
K1	3
K2	2
K3	4
K4	3

**Tabel 4.9** Data Bobot Setiap Alternatif dihitung dari lokasi user di USU

Alternatif	Kriteria			
	K1	K2	K3	K4
OYO 2940 Papakoel Guest House	3	1	5	4
OYO 1052 Sentosa Jaya Guest House Syariah	2	2	4	4
OYO 91224 Brayan Guest House	2	2	2	3
OYO 92429 Vania Guest House	2	1	2	4
OYO 2147 Mono Guest House	2	1	1	4
OYO 2229 Bunga Raya Residence	2	1	4	1
Johje Guest House Medan RedPartner	2	2	4	4
Super OYO 2574 Z Suites Hotel	1	3	4	1
RedDoorz @ Joyce Guest House Medan	1	2	5	2
Adam Malik Guesthouse near Regale ICC Medan RedPartner	1	3	3	2
Roemah 28 Syariah	1	5	5	5
Super OYO 1457 Tmj Guest House	2	2	4	4
SUPER OYO 90874 Millenium Inn 2	1	3	4	1
OYO 1585 Ring Road Residence	1	2	3	3
Roemah 54	1	4	4	3
OYO 91706 Hotel Serena Anggrek	2	1	3	3
Vosstel Guest House	1	4	5	4
Sapadia Guest House Sei Silau Medan	1	3	5	5
Tobali Guest House	1	5	3	4
OYO 92342 Hotel Batik Traveller	1	1	5	3

Setelah bobot inputan dari *user* dibuat maka akan dilanjutkan dengan proses dari metode ELECTRE :



### 1. Normalisasi Matriks Keputusan

Matriks di normalisasi menggunakan persamaan:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Berdasarkan Tabel 4.9 maka perhitungan matriks dapat dikalkulasikan dengan perhitungan sebagai berikut:

$$X_{ij} = \sqrt{x_{11}^2 + x_{12}^2 + \dots + x_{1n}^2}$$

Dimana i merupakan kolom dan j merupakan baris maka diperoleh hasil :

$$r_{11} = 0.057692$$

Matriks ternormalisasi R :

**Tabel 4.10** Matriks Ternormalisasi R

0.057692	0.006757	0.016287	0.017094
0.038462	0.013514	0.013029	0.017094
0.038462	0.013514	0.006515	0.012821
0.038462	0.006757	0.006515	0.017094
0.038462	0.006757	0.003257	0.017094
0.038462	0.006757	0.013029	0.004274
0.038462	0.013514	0.013029	0.017094
0.019231	0.02027	0.013029	0.004274
0.019231	0.013514	0.016287	0.008547
0.019231	0.02027	0.009772	0.008547
0.019231	0.033784	0.016287	0.021368
0.038462	0.013514	0.013029	0.017094
0.019231	0.02027	0.013029	0.004274
0.019231	0.013514	0.009772	0.012821
0.019231	0.027027	0.013029	0.012821
0.038462	0.006757	0.009772	0.012821
0.019231	0.027027	0.016287	0.017094
0.019231	0.02027	0.016287	0.021368
0.019231	0.033784	0.009772	0.017094
0.019231	0.006757	0.016287	0.012821

2. Pembobotan pada matriks yang telah dinormalisasi

$$W = [2,3,4,3]$$

Berdasarkan Tabel 4.9 maka perhitungan matriks dapat dikalkulasikan dengan perhitungan sebagai berikut:

$$V = R * W$$

Dimana W merupakan matriks bobot, R merupakan matriks yang telah dinormalisasi, dan V merupakan matriks dari pembobotan matriks R.

**Tabel 4.11** Pembobotan pada matriks yang telah dinormalisasi

0.115385	0.013514	0.065147	0.051282
0.076923	0.027027	0.052117	0.051282
0.076923	0.027027	0.026059	0.038462
0.076923	0.013514	0.026059	0.051282
0.076923	0.013514	0.013029	0.051282
0.076923	0.013514	0.052117	0.012821
0.076923	0.027027	0.052117	0.051282
0.038462	0.040541	0.052117	0.012821
0.038462	0.027027	0.065147	0.025641
0.038462	0.040541	0.039088	0.025641
0.038462	0.067568	0.065147	0.064103
0.076923	0.027027	0.052117	0.051282
0.038462	0.040541	0.052117	0.012821
0.038462	0.027027	0.039088	0.038462
0.038462	0.054054	0.052117	0.038462
0.076923	0.013514	0.039088	0.038462
0.038462	0.054054	0.065147	0.051282
0.038462	0.040541	0.065147	0.064103
0.038462	0.067568	0.039088	0.051282
0.038462	0.013514	0.065147	0.038462

3. Menentukan himpunan *concordance* dan *discordance* index.

a. Concordance

$$C_{kl} = \{ j, y_{kj} \geq y_{lj} \}, \text{ untuk } j = 1, 2, 3, \dots, n$$

Baris 1:

**Tabel 4.12.1** Menentukan himpunan *concordance* baris 1

C1	C2	C3	C4
T	F	T	T
T	F	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	F	T	T
T	F	T	T
T	F	T	T
T	F	T	T
T	F	T	F
T	F	T	T
T	F	T	T
T	F	T	T
T	F	T	T
T	F	T	T
T	T	T	T
T	F	T	T
T	F	T	F
T	F	T	T
T	T	T	T
T	T	T	T

Baris 2:

**Tabel 4.12.2** Menentukan himpunan *concordance* baris 2

C1	C2	C3	C4
F	T	F	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	F	T	T
T	T	F	T
T	F	T	T
T	F	F	F
T	T	T	T
T	F	T	T
T	T	T	T
T	F	T	T
T	T	T	T
T	F	F	T
T	F	F	F
T	F	T	T
T	T	F	T

Baris 3:

**Tabel 4.12.3** Menentukan himpunan *concordance* baris 3

C1	C2	C3	C4
F	T	F	F
T	T	F	F
T	T	T	T
T	T	T	F
T	T	T	F
T	T	F	T
T	T	F	F
T	F	F	T
T	T	F	T
T	F	F	T
T	F	F	F
T	T	F	F
T	F	F	T
T	T	F	T
T	F	F	T
T	F	F	F
T	F	F	F
T	F	F	F
T	T	F	T

Baris 4:

**Tabel 4.12.4** Menentukan himpunan *concordance* baris 4

C1	C2	C3	C4
F	T	F	T
T	F	F	T
T	F	T	T
T	T	T	T
T	T	T	T
T	T	F	T
T	F	F	T
T	F	F	T
T	F	F	T
T	F	F	T
T	F	F	F
T	F	F	T
T	F	F	T
T	F	F	T
T	F	F	T
T	F	F	T
T	T	F	T
T	F	F	T
T	F	F	F
T	F	F	T
T	T	F	T

Baris 5:

**Tabel 4.12.5** Menentukan himpunan *concordance* baris 5

C1	C2	C3	C4
F	T	F	T
T	F	F	T
T	F	F	T
T	T	F	T
T	T	T	T
T	T	F	T
T	F	F	T
T	F	F	T
T	F	F	T
T	F	F	T
T	F	F	F
T	F	F	T
T	F	F	T
T	F	F	T
T	F	F	T
T	T	F	T
T	F	F	T
T	F	F	F
T	F	F	T
T	T	F	T

Baris 6:

**Tabel 4.12.6** Menentukan himpunan *concordance* baris 6

C1	C2	C3	C4
F	T	F	F
T	F	T	F
T	F	T	F
T	T	T	F
T	T	T	F
T	T	T	T
T	F	T	F
T	F	T	T
T	F	F	F
T	F	T	F
T	F	F	F
T	F	T	F
T	F	T	T
T	F	T	F
T	F	T	F
T	T	T	F
T	F	F	F
T	F	F	F
T	F	T	F
T	T	F	F

Baris 7:

**Tabel 4.12.7** Menentukan himpunan *concordance* baris 7

C1	C2	C3	C4
F	T	F	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	F	T	T
T	T	F	T
T	F	T	T
T	F	F	F
T	T	T	T
T	F	T	T
T	T	T	T
T	F	T	T
T	T	T	T
T	F	F	T
T	F	F	F
T	F	T	T
T	T	F	T

Baris 8:

**Tabel 4.12.8** Menentukan himpunan *concordance* baris 8

C1	C2	C3	C4
F	T	F	F
F	T	T	F
F	T	T	F
F	T	T	F
F	T	T	F
F	T	T	T
F	T	T	F
T	T	T	T
T	T	F	F
T	T	T	F
T	F	F	F
F	T	T	F
T	T	T	T
T	T	T	F
T	F	T	F
F	T	T	F
T	F	F	F
T	T	F	F
T	F	T	F
T	T	F	F

Baris 9:

**Tabel 4.12.9** Menentukan himpunan *concordance* baris 9

C1	C2	C3	C4
F	T	T	F
F	T	T	F
F	T	T	F
F	T	T	F
F	T	T	F
F	T	T	T
F	T	T	F
T	F	T	T
T	T	T	T
T	F	T	T
T	F	T	F
F	T	T	F
T	F	T	T
T	T	T	F
T	F	T	F
F	T	T	F
T	F	T	F
T	F	T	F
T	F	T	F
T	F	T	F
T	T	T	F

Baris 10:

**Tabel 4.12.10** Menentukan himpunan *concordance* baris 10

C1	C2	C3	C4
F	T	F	F
F	T	F	F
F	T	T	F
F	T	T	F
F	T	T	F
F	T	F	T
F	T	F	F
T	T	F	T
T	T	F	T
T	T	T	T
T	F	F	F
F	T	F	F
T	T	F	T
T	T	T	F
T	F	F	F
F	T	T	F
T	F	F	F
T	T	F	F
T	F	T	F
T	T	F	F

Baris 11:

**Tabel 4.12.11** Menentukan himpunan *concordance* baris 11

C1	C2	C3	C4
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
F	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
F	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T



Baris 12:

**Tabel 4.12.12** Menentukan himpunan *concordance* baris 12

C1	C2	C3	C4
F	T	F	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	F	T	T
T	T	F	T
T	F	T	T
T	F	F	F
T	T	T	T
T	F	T	T
T	T	T	T
T	F	T	T
T	T	T	T
T	F	F	T
T	F	F	F
T	F	T	T
T	T	F	T

Baris 13:

**Tabel 4.12.13** Menentukan himpunan *concordance* baris 13

C1	C2	C3	C4
F	T	F	F
F	T	T	F
F	T	T	F
F	T	T	F
F	T	T	F
F	T	T	T
F	T	T	F
T	T	T	T
T	T	F	F
T	T	T	F
T	F	F	F
F	T	T	F
T	T	T	T
T	T	T	F
T	F	T	F
F	T	T	F
T	F	F	F
T	T	F	F
T	F	T	F
T	T	F	F

Baris 14:

**Tabel 4.12.14** Menentukan himpunan *concordance* baris 14

C1	C2	C3	C4
F	T	F	F
F	T	F	F
F	T	T	T
F	T	T	F
F	T	T	F
F	T	F	T
F	T	F	F
T	F	F	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	F	F
T	F	F	T
T	T	T	T
T	F	F	T
F	T	T	T
T	F	F	F
T	F	F	F
T	F	T	F
T	T	F	T

Baris 15:

**Tabel 4.12.15** Menentukan himpunan *concordance* baris 15

C1	C2	C3	C4
F	T	F	F
F	T	T	F
F	T	T	T
F	T	T	F
F	T	T	F
F	T	T	T
F	T	T	F
T	T	T	T
T	T	F	T
T	T	T	T
T	F	F	F
F	T	T	F
T	T	T	T
T	T	T	T
T	T	T	T
F	T	T	T
T	T	F	F
T	T	F	F
T	F	T	F
T	T	F	T

Baris 16:

**Tabel 4.12.16** Menentukan himpunan *concordance* baris 16

C1	C2	C3	C4
F	T	F	F
T	F	F	F
T	F	T	T
T	T	T	F
T	T	T	F
T	T	F	T
T	F	F	F
T	F	F	T
T	F	F	T
T	F	T	T
T	F	F	F
T	F	F	F
T	F	F	T
T	F	T	T
T	F	F	T
T	T	T	T
T	F	F	F
T	F	F	F
T	F	T	F
T	T	F	T

Baris 17:

**Tabel 4.12.17** Menentukan himpunan *concordance* baris 17

C1	C2	C3	C4
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
T	T	T	T
T	T	T	T
T	F	T	F
F	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
F	T	T	T
T	T	T	T
T	T	T	F
T	F	T	T
T	T	T	T

Baris 18:

**Tabel 4.12.18** Menentukan himpunan *concordance* baris 18

C1	C2	C3	C4
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	T	T
T	T	T	T
T	T	T	T
T	T	T	T
T	F	T	T
F	T	T	T
T	T	T	T
T	T	T	T
T	F	T	T
F	T	T	T
T	F	T	T
T	T	T	T
T	F	T	T
T	T	T	T

Baris 19:

**Tabel 4.12.19** Menentukan himpunan *concordance* baris 19

C1	C2	C3	C4
F	T	F	T
F	T	F	T
F	T	T	T
F	T	T	T
F	T	T	T
F	T	F	T
F	T	F	T
T	T	F	T
T	T	F	T
T	T	T	T
T	T	F	F
F	T	F	T
T	T	F	T
T	T	T	T
T	T	F	T
F	T	T	T
T	T	F	T
T	T	F	F
T	T	T	T
T	T	F	T

Baris 20:

**Tabel 4.12.20** Menentukan himpunan *concordance* baris 20

C1	C2	C3	C4
F	T	T	F
F	F	T	F
F	F	T	T
F	T	T	F
F	T	T	F
F	T	T	T
F	F	T	F
T	F	T	T
T	F	T	T
T	F	T	T
T	F	T	F
F	F	T	F
T	F	T	T
T	F	T	T
T	F	T	T
F	T	T	T
T	F	T	F
T	F	T	F
T	F	T	F
T	T	T	T

b. Discordance

Baris 1:

**Tabel 4.13.1** Menentukan himpunan *discordance* baris 1

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	T	F	F



Baris 4:

**Tabel 4.13.4** Menentukan himpunan *discordance* baris 4

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 5:

**Tabel 4.13.5** Menentukan himpunan *discordance* baris 5

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F





Baris 8:

**Tabel 4.13.8** Menentukan himpunan *discordance* baris 8

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 9:

**Tabel 4.13.9** Menentukan himpunan *discordance* baris 9

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 10:

**Tabel 4.13.10** Menentukan himpunan *discordance* baris 10

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 11:

**Tabel 4.13.11** Menentukan himpunan *discordance* baris 11

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 12:

**Tabel 4.13.12** Menentukan himpunan *discordance* baris 12

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 13:

**Tabel 4.13.13** Menentukan himpunan *discordance* baris 13

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 14:

**Tabel 4.13.14** Menentukan himpunan *discordance* baris 14

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 15:

**Tabel 4.13.15** Menentukan himpunan *discordance* baris 15

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 16:

**Tabel 4.13.16** Menentukan himpunan *discordance* baris 16

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

Baris 17:

**Tabel 4.13.17** Menentukan himpunan *discordance* baris 17

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F



Baris 20:

**Tabel 4.13.20** Menentukan himpunan *discordance* baris 20

C1	C2	C3	C4
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F
F	F	F	F

#### 4. Menentukan matriks *concordance* dan *discordance*

##### a. Matriks *Concordance*

Matriks *Concordance* diperoleh dari hasil penjumlahan bobot-bobot yang termasuk dalam bagian himpunan *Concordance*. Dengan contoh hasil:

	–	13	10	10	13	13	13	9	10	6	6	6	13	13	13	9	10	6	6	9
$F_9$	–	10	10	13	13	13	13	9	10	6	6	6	13	13	13	9	10	6	5	6 <sup>1</sup>
$I_6$	10	–	13	13	13	13	13	9	10	9	9	13	13	13	13	9	13	9	9	5 <sup>1</sup>
$I_6$	10	7	–	13	13	13	13	7	10	9	9	10	13	13	13	9	10	9	6	6
$I_6$	10	7	10	–	13	13	13	7	10	6	6	10	13	13	13	9	10	6	6	5 <sup>1</sup>
$I_6$																				5 <sup>1</sup>
$I_6$																				5 <sup>1</sup>
$I_6$	6	3	5	8	13	–	3	6	6	6	6	10	13	13	8	9	10	6	6	5 <sup>1</sup>
$I_6$	10	10	10	13	13	13	–	10	10	10	10	13	13	13	13	1	1	6	6	5 <sup>1</sup>
$I_8$	13	10	10	13	13	13	9	–	6	6	6	10	13	13	13	9	10	6	6	5 <sup>1</sup>
$I_6$	10	3	8	8	13	13	3	6	–	13	10	13	13	8	13	10	9	6	6	5 <sup>1</sup>
$I_6$	10	3	8	8	13	13	3	6	13	–	10	13	13	8	13	10	9	6	6	5 <sup>1</sup>
$I_6$	10	3	8	13	13	13	9	6	6	6	–	13	9	13	13	13	6	6	6	9 <sup>1</sup>
$I_6$	13	9	9	8	3	3	13	13	13	9	9	–	6	9	6	10	10	6	6	5 <sup>1</sup>
$I_6$	13	10	10	13	13	13	9	10	10	10	13	13	–	8	9	10	6	6	6	5 <sup>1</sup>
$I_6$	9	10	10	13	13	10	10	7	10	9	9	13	13	–	13	9	13	9	6	1
$I_6$	13	7	10	13	13	13	9	10	9	9	10	13	13	8	–	10	6	6	6	5 <sup>1</sup>
$I_6$	13	7	5	13	3	13	9	6	9	6	6	6	6	13	9	–	6	6	6	1
$I_6$	9	10	5	8	3	13	7	10	9	6	6	13	13	13	6	13	–	9	6	1
$I_6$	10	10	13	8	13	10	3	10	6	9	10	13	13	16	6	10	9	–	6	1
$I_6$	10	3	13	13	13	13	3	10	6	6	10	13	13	13	9	10	9	9	9	–



### b. Discordance

Matriks *Discordance* didapatkan dengan membagi selisih maksimum nilai kriteria yang termasuk dalam himpunan *Discordance* dan membaginya dengan maksimum selisih nilai seluruh kriteria yang ada. Dengan contoh hasil:

D =

-	0	0	1,01	0,34	0	0,34	0	0,66	0,26	0,34	0,34	0,67	0	0	0,33	0,33	0,67	1	1,01
F	-	0,67	0,24	0,34	0,34	0,67	0	0	0,33	0,33	0,67	1	1,01	0,67	1	1,01	0,67	0,97	1
I <sub>0</sub>	0,67	-	0,52	0,33	0,33	0	0	0,33	0,33	0	1	1	0,67	0,97	0,53	0,67	NAN	NAN	0,27I
I <sub>0</sub>	NAN	NAN	-	0,52	0,33	0,67	0	1	1	0,26	0,34	0,34	0,67	0	0	0,33	0,33	0,67	0
I	0	1	0,25	-	0,54	0,33	0,33	0	0,34	0,33	0,67	1	1	1	0	0	0,67	0,67	0
I	0	1	0,67	0,33	-	0,67	1	1	0	0,26	NAN	NAN	1	1	0,67	0,33	0,33	1	1
I <sub>0</sub>	1	1	1	0	0,34	-	1	1	0	0,26	0,33	0,33	0,67	0,53	1	1	0,53	1	1
I	1	0	1	0,67	NAN	0,33	-	1	0,33	0,3	0,34	0,67	0,33	0,24	0,34	0,34	1,01	1	1
I <sub>1</sub>	1	0	1	1	1	0,33	0	-	0,34	0,26	0,34	0,67	1,01	0,67	0,34	0,67	0,67	1,01	0
I <sub>1</sub>	0	0	1	1	0	0	0,25	0,67	-	0,34	0,67	0,34	0,67	0,53	1	1	0	0	0
I <sub>1</sub>	1	1	1	1	0	0,33	0,34	1	1	-	0	1	1	0,67	0,33	0	1	1	1,01I
I	1	1	1	0	0	1	1	1	0,33	0,34	-	1	1	1	0,67	0,67	1,01	0,26	1
I <sub>1</sub>	1	0,26	-	0,52	0,33	0,67	0	1	1	0,26	0,34	-	0,67	0	0	0,33	0,33	0,67	1
I <sub>1</sub>	0	1	0,25	-	0,54	0,33	0,33	0	0,34	0,33	0,67	1	-	1	0	0	0,67	0,67	1
I <sub>1</sub>	0,26	1	0	1	-	0,67	1	1	0	0,26	NAN	1	1	-	0,67	0,33	0,33	1	1
I	1	NAN	1	0	1	-	1	1	0	0,26	0,33	1	NAN	NAN	-	1	0,53	1	0,67I
I <sub>0</sub>	1	0	1	0,67	NAN	0,33	-	1	0	0,33	0,3	0,34	NAN	NAN	NAN	-	0,34	1,01	0,67I
I	1	0	1	1	1	0,33	0	-	0,34	0,26	1	1	1,01	0,67	0,34	0,67	-	1,01	1
I	0	0	1	1	0	0	0,25	0,67	-	0,34	0,67	0,34	0,67	0,53	1	1	0	-	1
I <sub>1</sub>	1	1	1	0	0	1	1	1	0,67	1	1	0	NAN	NAN	NAN	1,01	0,53	1	0

### 5. Menentukan matriks dominan *concordance* dan *discordance*

Menentukan matriks dominan *concordance* menggunakan nilai threshold untuk indeks concordance, yaitu dengan membandingkan setiap nilai elemen matriks concordance dengan nilai threshold.

Menghitung Nilai threshold  $\underline{c}$ :

$$\underline{c} = \frac{\sum_{k=1}^n \sum_{l=1}^n C_{kl}}{m(m-1)}$$

$$\underline{c} = \frac{3318}{20(20-1)}$$

$$\underline{c} = 8,37$$

Berdasarkan nilai threshold  $\underline{c}$ , nilai setiap elemen matriks F sebagai matriks dominan *concordance* ditentukan sebagai berikut:

$$f_{kl} = 1, \text{ jika } c_{kl} \geq \underline{c} \text{ dan } f_{kl} = 0, \text{ jika } c_{kl} < \underline{c}$$

```

-1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1
0 -1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1
0 0 -1 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1
1 0 1 -1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 -1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 1 -1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0
1 1 1 1 1 1 -1 1 1 1 0 1 1 1 1 1 0 0 1 1
0 1 1 0 1 1 0 -1 0 1 0 1 1 1 0 1 0 0 0 0
0 1 1 0 1 1 0 1 -1 1 0 1 1 1 0 1 0 0 0 1
0 0 0 0 0 0 0 0 0 -1 0 0 0 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 -1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 0 -1 1 1 1 1 0 0 1 1
0 1 1 0 1 1 0 1 0 1 0 1 -1 1 0 1 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0 0 -1 0 1 0 0 0 1
0 0 1 0 0 1 0 1 0 1 0 0 1 1 -1 1 0 0 0 1
0 0 1 0 1 1 0 0 0 1 0 0 0 1 0 -1 0 0 0 1
1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 -1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1
0 0 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 0 -1 1
0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1 0 0 0 -1

```

Menghitung nilai threshold  $\underline{d}$ :

Dominasi matriks *discordance* didefinisikan dengan menggunakan threshold  $\underline{d}$ .

$$\underline{d} = \frac{\sum_{k=1}^n \sum_{l=1}^n d_{kl}}{m(m-1)}$$

$$\underline{d} = \frac{231,4}{20(20-1)}$$

$$\underline{d} = 0,608$$

Berdasarkan nilai threshold  $\underline{d}$ , nilai setiap elemen matriks G sebagai matriks dominan concordance ditentukan sebagai berikut:

$$g_{kl} = 1, \text{ jika } D_{kl} \geq \underline{d} \text{ dan } g_{kl} = 0, \text{ jika } D_{kl} < \underline{d}$$

## 6. Menentukan Agregat dominan matriks

Matriks agregat dominan diperoleh dengan mengkalikan matriks dominan F dan matriks dominan G yang didapatkan di tahap sebelumnya. Setelah kedua matriks dikalikan maka akan diperoleh hasil:

```

-1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1
0 -1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1
0 0 -1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1
1 0 1 -1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 -1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 1 -1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0
1 1 1 1 1 1 -1 1 1 1 0 1 1 1 1 1 0 0 1 1
0 1 1 0 1 1 0 -1 0 1 0 1 1 1 0 1 0 0 0 0
0 1 1 0 1 1 0 1 -1 1 0 1 1 1 0 1 0 0 0 1
0 0 0 0 0 0 0 0 0 -1 0 0 0 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 -1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 0 -1 1 1 1 1 0 0 1 1
0 1 1 0 1 1 0 1 0 1 0 1 -1 1 0 1 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0 0 -1 0 1 0 0 0 1
0 0 1 0 0 1 0 1 0 1 0 0 1 1 -1 1 0 0 0 1
0 0 1 0 1 1 0 0 0 1 0 0 0 1 0 -1 0 0 0 1
1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 -1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1
0 0 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 0 -1 1
0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1 0 0 0 -1

```

## 7. Eliminasi alternatif yang less favourable

Dari jumlah  $E_{kl} = 1$  yang terbanyak adalah  $A_3$ , selanjutnya yang mendominasi adalah  $A_8$ ,  $A_{19}$ . Sehingga alternatif terpilih yaitu Roemah 28 Syariah, diikuti oleh

Vosstel Guest House dan Sapadia Guest house sei silau medan. Dan hasil dari perhitungan manual dan sistem memiliki *output* yang sama.

#### 4.3 User Acceptance Test(UAT)

UAT (User Acceptance Test) Aplikasi pemilihan Guesthouse di Kota Medan dengan cara menyebarkan kuesioner kepada 20 responden. Responden tersebut terdiri dari 12 mahasiswa alumni dan 8 karyawan swasta. UAT (User Acceptance Test) dilakukan dalam berbagai aspek, yaitu aspek tampilan, aspek kemudahan, dan aspek apresiasi keseluruhan mengenai sistem yang dibangun. Berikut merupakan hasil dari UAT (User Acceptance Test).

Berikut merupakan daftar pertanyaan yang terdapat pada kuesioner UAT.

**Tabel 5.1** Pertanyaan UAT

No	Pertanyaan	A	B	C	D	E
1	Menurut Anda, apakah tampilan front-end Aplikasi pemilihan guesthouse di kota medan menarik?					
2	Menurut Anda, bagaimana perpaduan warna dari tampilan Aplikasi pemilihan guesthouse di kota medan?					
3	Menurut Anda, apakah tata letak (layout) Aplikasi pemilihan guesthouse di kota medan cukup bagus?					
4	Menurut Anda, apakah aksesibilitas Aplikasi pemilihan guesthouse di kota medan ini cukup mudah?					
5	Menurut Anda, apakah fitur-fitur dalam Aplikasi pemilihan guesthouse di kota medan ini mudah dipahami?					
6	Menurut Anda, apakah Aplikasi pemilihan guesthouse di kota medan membantu anda mencari guesthouse yang tepat?					
7	Menurut Anda, apakah Aplikasi pemilihan guesthouse di kota medan memberikan informasi yang baik mengenai guesthouse yang diinginkan?					

8	Menurut Anda, apakah perlu adanya Aplikasi pemilihan guesthouse di kota medan?					
9	Menurut Anda, apakah Anda akan menyarankan agar Aplikasi pemilihan guesthouse di kota medan ke orang lain?					

Berikut ini merupakan hasil UAT dari para responden untuk berbagai aspek dari aplikasi yang mencakup aspek tampilan, aspek kemudahan dan aspek apresiasi keseluruhan mengenai sistem yang dibangun.

**Tabel 5. 2 Hasil UAT**

No	Pertanyaan	Jawaban					Persentase				
		A	B	C	D	E	A	B	C	D	E
1	Pertanyaan 1	18	2	-	-	-	90%	10%	-	-	-
2	Pertanyaan 2	18	1	1	-	-	90%	5%	5%	-	-
3	Pertanyaan 3	16	4	-	-	-	80%	20%	-	-	-
4	Pertanyaan 4	17	2	-	-	-	89.5%	10.5%	-	-	-
5	Pertanyaan 5	17	3	-	-	-	85%	15%	-	-	-
6	Pertanyaan 6	14	5	1	-	-	70%	25%	5%	-	-
7	Pertanyaan 7	19	1	-	-	-	95%	5%	-	-	-
8	Pertanyaan 8	15	4	1	-	-	75%	20%	5%	-	-
9	Pertanyaan 9	12	7	1	-	-	60%	35%	5%	-	-

**Tabel 5. 3** Pembobotan Hasil UAT

No	Pertanyaan	Nilai					Jml	Nilai Rata-rata	Persentase nilai
		Ax5	Bx4	Cx3	Dx2	Ex1			
1	Pertanyaan 1	18	2	-	-	-	98	4.9	98%
2	Pertanyaan 2	18	1	1	-	-	97	4.85	97%
3	Pertanyaan 3	16	4	-	-	-	96	4.8	96%
4	Pertanyaan 4	17	2	-	-	-	93	4.65	93%
5	Pertanyaan 5	17	3	-	-	-	97	4.85	97%
6	Pertanyaan 6	14	5	1	-	-	93	4.65	93%
7	Pertanyaan 7	19	1	-	-	-	99	4.95	99%
8	Pertanyaan 8	15	4	1	-	-	94	4.7	94%
9	Pertanyaan 9	12	7	1	-	-	91	4.55	91%

Penjelasan:

Dari table diatas dapat dilihat bahwa nilai dari 20 responden untuk pertanyaan pertama adalah 98. Nilai rata-ratanya adalah  $98/20=4.9$ . persentase nilainya adalah  $4.9/5 \times 100=98\%$  artinya aplikasi ini memiliki tampilan yang sangat menarik.

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil implelementasi sistem pemilihan lokasi guesthouse di kota Medandengan menggunakan kombinasi algoritma *Dijkstra* dan metode ELECTRE, maka dapat diambil kesimpulan yaitu :

1. Sistem pemilihan lokasi guesthouse dapat memberikan rekomendasi guesthouse yang sesuai dengan kriteria keinginan *user*.
2. Sistem pemilihan lokasi guesthouse ini dapat mencari jarak dari lokasi *user* ke setiap alternatif guesthouse.
3. Berdasarkan User Acceptance Test, sistem pemilihan lokasi guesthouse memiliki ketepatan pemilihan guesthouse sesuai dengan yang diinginkan user sebesar 95%.

#### 5.2 Saran

Saran dari penulis yang mampu diberikan agar dapat mengembangkan penelitian ini yaitu:

1. Data dari kriteria adalah statis dan alternatif guesthouse tidak dapat ditambah. Untuk selanjutnya, sistem boleh dibuat lebih dinamis sehingga mampu melakukan penambahan data.
2. Ke depannya diharapkan algoritma ini dapat dikembangkan untuk studi kasus lainnya.

### DAFTAR PUSTAKA

- Adisaputra Sinaga, N., & Purba, A. T. (2019). Decision Support System for Purchase of Used Cars using Electre Method. In *Journal of Computer Networks* (Vol. 1, Issue 2). Aiding System Using Fuzzy Electre III Method For Vendor Selection, Expert Systems with Applications 3 , 10837–10847. 2009
- Burstein, F., & Holsapple, C. (2009). *Handbook On Decision Support Systems I. basic themes*. Springer.
- Desmile, J., Orisa, M., & Santi Wahyuni, F. (2021). Aplikasi Pencarian Bengkel Sepeda Motor Menggunakan Location Based Service Pada Wilayah Kota Tebing Tinggi Berbasis Android. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 5, Issue 1).
- Gading Sadewo, M., Perdana Windarto, A., & Retno Andani, S. (2017). *Pemanfaatan Algoritma Clushtering Dalam Mengelompokkan Jumlah Desa/Kelurahan Yang Memiliki Sarana Kesehatan Menurut Provinsi Dengan K-MEANS*. 124–131. <http://ejurnal.stmik-budidarma.ac.id/index.php/komik>
- Gholam, A.M., Saremi, H.Q., dan Ramezani, M. Design A New Mixed Expert Decision
- Iqbal, M., Zhang, K., Iqbal, S., & Tariq, I. (2018). A Fast and Reliable Dijkstra Algorithm for Online Shortest Path. *International Journal of Computer Science and Engineering*, 5(12), 24–27. <https://doi.org/10.14445/23488387/IJCSE-V5I12P106>
- Iqbal, M., Zhang, K., Iqbal, S., & Tariq, I. (2018). A Fast and Reliable Dijkstra Algorithm for Online Shortest Path. *International Journal of Computer Science and Engineering*, 5(12), 24–27. <https://doi.org/10.14445/23488387/IJCSE-V5I12P106>
- Junanda, B., Huda, Y., & Kurniadi, D. (2018). Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra pada Sistem Informasi Geografis Pemetaan Stasius Pengisian Bahan Bakar Umum. *Teknik Elektronika & Informatika*, 4(1), 1–8.
- Kusumadewi, Sri and Purnomo, Hari., (2010), Aplikasi Logika Fuzzy untuk Pendukung Keputusan, Yogyakarta : Graha Ilmu.
- Munir. (2012). Multimedia Konsep dan Aplikasi Dalam Pendidikan. Bandung: Alfabeta
- Parapat, M. N., Kusbianto, D., & Rahmad, C. (2017). Rancang Bangun Aplikasi Pencarian Rute Terpendek Jasa Kiriman Barang Berbasis Mobile Dengan Metode Algoritma Dijkstra. *Jurnal Informatika Polinema*, 3(3), 15–19. <https://doi.org/10.33795/jip.v3i3.28>
- Praba, A., Pinem, R., & Pungkasanti, P. T. (2018). Application Elimination Et Choix Traduisant La REalité (ELECTRE) On Decission Support System. *TRANSFORMATIKA*, 15(2), 106–113.
- Siahaan, R. (2017). Perbandingan Kompleksitas Waktu Teoretis Dan Real Time Algoritma Strand Sort, Sieve Sort, Gnome Sort. *Skripsi. Universitas Sumatera Utara*.
- Simaremare, M., & Putera Utama Siahaan, A. (2016). Decision Support System in Selecting The Appropriate Laptop Using Simple Additive Weighting. In *Jenderal Abdul Haris Nasution No. 73F* (Vol. 4).
- Yanie, A., Hasibuan, A., Ishak, I., Marsono, M., Lubis, S., Nurmalini, N., Mesran, M., Nasution, S. D., Rahim, R., Nurdianto, H., & Ahmar, A. S. (2018). Web Based Application for Decision Support System with ELECTRE Method. *Journal of Physics: Conference Series*, 1028(1). <https://doi.org/10.1088/1742-6596/1028/1/012054>