

**ANALISIS KINERJA IMPLEMENTASI ALGORITMA
DIJKSTRA UNTUK Mencari Rute Terdekat dari
BALEEDAH KE PERPUSTAKAAN KAWALUYAAN
DENGAN MENGGUNAKAN PYTHON**

SKRIPSI

Karya Tulis sebagai syarat memperoleh
Gelar Sarjana Komputer dan Fakultas Teknologi Informasi
Universitas Bale Bandung

Oleh :

JAHWAN ANWAR SANI

NPM. 301180041



PROGRAM STRATA 1

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

BANDUNG

2022

LEMBAR PERSETUJUAN PEMBIMBING

**ANALISIS KINERJA IMPLEMENTASI ALGORITMA DIJKSTRA
UNTUK MENCARI RUTE TERDEKAT DARI BALEEDAH KE
PERPUSTAKAAN KAWALUYAAN DENGAN
MENGUNAKAN PYTHON**

Disusun Oleh :

JAHWAN ANWAR SANI
NPM. 301180041

Telah diterima dan disetujui untuk persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2022

Disetujui oleh :

Pembimbing Utama

Pembimbing Pendamping

Yaya Suharya S.Kom. M.T
NIK: 01043170007

Sutiyono W.P, S.T, M.Kom
NIK: 01043180002

LEMBAR PERSETUJUAN PENGUJI

**ANALISIS KINERJA IMPLEMENTASI ALGORITMA DIJKSTRA
UNTUK MENCARI RUTE TERDEKAT DARI BALEEDAH KE
PERPUSTAKAAN KAWALUYAAN DENGAN
MENGUNAKAN PYTHON**

Disusun Oleh :

JAHWAN ANWAR SANI
NPM. 301180041

Telah diterima dan disetujui untuk persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2022

Disetujui oleh :

Penguji 1

Penguji 2

Sukiman S.Tr.Kom, S.Pd, M.Kom
NIK: 04104821001

Denny Rusdianto S.T, M.Kom
NIK: 04104808094

LEMBAR PENGESAHAN LEMBAGA

**ANALISIS KINERJA IMPLEMENTASI ALGORITMA DIJKSTRA
UNTUK MENCARI RUTE TERDEKAT DARI BALEEDAH KE
PERPUSTAKAAN KAWALUYAAN DENGAN
MENGUNAKAN PYTHON**

Disusun Oleh :

JAHWAN ANWAR SANI
NPM. 301180041

Telah diterima dan disetujui untuk persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2022

Mengetahui,
Dekan,

Yudi Herdiana S.T, M.T.
NIK: 04104808008

Mengesahkan,
Ketua Program Studi,

Yusuf Muharam M.Kom
NIK: 04104820003

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini, saya :

Nama : Jahwan Anwar Sani
NIM : 301180041
Jurusan : Teknik Informatika
Fakultas : Teknologi Informasi
Judul : Analisis Kinerja Implementasi Algoritma Dijkstra Untuk Mencari Rute Terdekat Dari Baleendah Ke Perpustakaan Kawalayaan Dengan Menggunakan Python

Dengan ini saya menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepengetahuan saya tidak terdapat karya yang ditulis atau diterbitkan orang lain, kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang lazim.

Bandung, Agustus 2022

Yang menyatakan,

Jahwan Anwar Sani

NPM. 301180041

ABSTRAK

Pada era perkembangan ilmu pengetahuan saat ini banyak orang yang memiliki rasa ingin tahu yang tinggi, pada dasarnya mereka pastinya mencari sumber-sumber yang mendukung untuk pengetahuan mereka, contohnya seperti buku, jurnal, dan artikel. Yang dapat membantu mereka untuk menyelesaikan masalah yang mereka temui seperti tugas sekolah, dan kuliah.

Dengan adanya perpustakaan yang dibuka oleh pemerintah, dengan demikian banyak mahasiswa atau mahasiswi yang datang untuk mencari buku untuk menjadi acuan berbagai tugas contohnya peneliti sendiri yang sering ke Perpustakaan Pawalayaan Bandung untuk mencari referensi skripsi dan jarak yang peneliti tempuh dari kosan peneliti ke Perpustakaan Kawalayaan lumayan jauh maka dari pengalaman peneliti selama ini yang lumayan sering ke Perpustakaan Kawalayaan Bandung dan juga untuk meminimalisir penggunaan bensin motor, maka penulis mencari rute terdekat atau terpendek untuk menuju ke Perpustakaan Kawalayaan Bandung. Untuk menyelesaikan masalah penelitian ini peneliti menggunakan Algoritma Dijkstra yang dimana algoritma ini dapat memecahkan masalah short path problem yang dialami peneliti, dan metode pengumpulan data yang dilakukan peneliti yaitu observasi, studi pustaka, identifikasi masalah, analisis, perancangan, implementasi dan hasil.

Dari hasil analisis yang dilakukan secara manual yaitu rute terpedek yang dapat dilalui yaitu Baleendah, Dayehkolot, Bojongsoang, Batununggal indah IV, Soekarno Hatta, Kawalayaan, ini berdasarkan perhitungan manual. Sedangkan hasil analisis menggunakan python adalah dari Baleendah, Dayehkolot, Bojongsoang, Batununggal indah IV, Soekarno Hatta, Kawalayaan, dengan jarak tempuh 20.2 km. Berdasarkan hasil analisis dengan menggunakan Algoritma Dijkstra maka dapat disimpulkan bahwa Algoritma Dijkstra dapat digunakan untuk pemecahan masalah sort path problem.

Kata Kunci: Algoritma dijkstra, Dijkstra, Python, Rute terpendek

ABSTRACT

In the current era of scientific development, many people have high curiosity, basically they are certainly looking for sources that support their knowledge, for example, such as books, journals, and articles. Which can help them to solve the problems they encounter such as schoolwork, and college.

With the library opened by the government, thus many students or students come to look for books to be a reference for various tasks, for example researchers themselves who often go to the Pawaluyaan Library Bandung to find thesis reviews and the distance that researchers travel from the research house to the Kawaluyaan Library quite far, from the experience of researchers so far who have been to the Bandung Kawaluyaan Library quite often and also to minimize the use of motorbike gasoline, the author is looking for the nearest or shortest route to go to the Bandung Kawaluyaan Library. To solve this research problem, researchers use the Dijkstra Algorithm where this algorithm can solve short path problems experienced by researchers, and data collection methods carried out by researchers, namely observation, literature study, problem identification, analysis, design, implementation and results.

From the results of the analysis carried out manually, namely the fastest route that can be passed, namely Baleendah, Dayehkolot, Bojongsoang, Batununggal indah IV, Soekarno Hatta, Kawaluyaan, this is based on manual calculations. While the results of the analysis using python are from Baleendah, Dayehkolot, Bojongsoang, Batununggal indah IV, Soekarno Hatta, Kawaluyaan, with a distance of 20.2 km. Based on the results of the analysis using the Dijkstra Algorithm, it can be concluded that the Dijkstra Algorithm can be used to solve the problem of sort path problems.

Keywords: *Dijkstra Algorithm, Dijkstra, Python, Shortest route*

KATA PENGANTAR



Puji dan syukur saya panjatkan kepada Allah Swt. atas ridonya saya dapat menyelesaikan penyusunan skripsi ini. Adapun judul skripsi yang saya ajukan adalah “*analisis kinerja implementasi algoritma Dijkstra untuk mencari rute terdekat dari Baleendah ke Perpustakaan Kawalayaan*”. Skripsi ini diajukan untuk memenuhi syarat kelulusan mata kuliah Skripsi di Fakultas Teknologi Informasi Universitas Bale Bandung. Tidak dapat disangkal bahwa butuh usaha yang keras dalam penyelesaian pengerjaan skripsi ini. Namun, karya ini tidak akan selesai tanpa orang-orang tercinta di sekeliling saya yang mendukung dan membantu.

Terima kasih saya sampaikan kepada:

1. Bapak Dr. Ir. H. Ibrahim Danuwikarsa, M.S. Selaku Rektor Universitas Bale Bandung
2. Bapak Yudi Herdiana., S.T., M.T selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
3. Bapak Yusuf Muharam, M.Kom. selaku ketua program Studi Teknik Informasi Universitas Bale Bandung.
4. Bapak Yaya Suharya., S.Kom, M.T selaku Dosen Pembimbing Utama.
5. Bapak Sutiyono W.P, S.T, M.Kom selaku Dosen pembimbing Kedua.
6. Segenap Dosen Fakultas Informasi Universitas Bale Bandung.
7. Kedua Orang Tua yang telah memberikan motivasi, dukungan dan do’a kepada saya.
8. Semua rekan-rekan FTI Angkatan 2018 yang senantiasa membantu dan suport dalam penulisan laporan skripsi.

Baleendah, Agustus 2022

Jahwan Anwar Sani

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Landasan Teori	5
2.2 Dasar Teori	18
BAB III METODOLOGI PENELITIAN	36
3.1 Kerangka Pikir.....	36
3.2 Deskripsi.....	37
BAB IV ANALISIS DAN PEMBAHASAN	39
4.1 Analisis.....	39
4.2 Perancangan.....	39
BAB V HASIL DAN PEMBAHASAN	57
5.1 Hasil.....	57
BAB VI PENUTUP	78
6.1 Kesimpulan.....	78
6.2 Saran	79
DAFTAR PUSTAKA	80
LAMPIRAN.....	82

DAFTAR GAMBAR

Gambar 2.3.1 Jembatan Konigsberg	20
Gambar 2.3.2 Penyajian Jembatan konigsborg	21
Gambar 2.3.3 Terdiri dari 4 ruas dan 5 simpul	22
Gambar 2.3.4 Graf yang sama dan tapi disajikan berbeda.....	22
Gambar 2.3.5 Graf berlebel.....	22
Gambar 2.3.6 Keterhubungan graf.....	23
Gambar 2.4.1 Graf terhubung	24
Gambar 2.6.1 Graf.....	25
Gambar 2.7.1 Planar/map.....	26
Gambar 2.9.1 Rumus Pemecahan Masalah Rute Terdekat	27
Gambar 3.1 Kerangka Pikir.....	36
Gambar 4.1 Google Maps	41
Gambar 4.2 Lokasi asal peneliti.....	41
Gambar 4.3 Lokasi Tujuan Peneliti	42
Gambar 4.4 Jalur yang dilalui berdasarkan google maps	42
Gambar 4.6.1 Peta Titik Yang Akan Dilalui.....	43
Gambar 4.6.2 Graf Berdasarkan Google Maps.....	44
Gambar 4.9.1 Activity diagram pengguna	49
Gambar 4.9.2 Tampilan Program.....	50
Gambar 5.1.1 Tampilan awal	63
Gambar 5.1.2 Jarak dari Baleendah ke Perpustakaan kawaluyaan	64
Gambar 5.1.3 Node A Ke B	64
Gambar 5.1.4 Node B ke C	65
Gambar 5.1.5 Node C ke D	65
Gambar 5.1.6 Node C ke F	66
Gambar 5.1.7 Node D ke E	66
Gambar 5.1.8 Node F ke E.....	67
Gambar 5.1.9 Node F ke H	67
Gambar 5.1.10 Node H ke I	68
Gambar 5.1.11 Node E ke I.....	68

Gambar 5.1.12 Node I ke J.....	69
Gambar 5.1.13 Node G ke H.....	69
Gambar 5.1.14 Node A, B ke C	70
Gambar 5.1.15 Node A, B, C dan D	70
Gambar 5.1.16 Node A, B, C dan F	71
Gambar 5.1.17 Node A, B, C, F dan E	71
Gambar 5.1.18 Node A, B dan G	72
Gambar 5.1.19 Node A, B, G dan H	72
Gambar 5.1.20 Node A, B, G, H dan I.....	73
Gambar 5.1.21 Node B, G, H, I dan J	73
Gambar 5.1.22 Node B, C dan D	74
Gambar 5.1.23 Node B, C, F dan E	74
Gambar 5.1.24 Node C, F, E, I dan J	75
Gambar 5.1.25 Node D, E, I dan J	75
Gambar 5.1.26 Node E, I dan J	76
Gambar 5.1.27 Node F, E, I dan J	76
Gambar 6.1.1 Graf Berbobot dan Berarah	78

DAFTAR TABEL

Tabel 2.1.1 Tabel Penelitian Terdahulu	5
Tabel 2.1.2 Persamaan Dan Perdaan Penelitian.....	14
Tabel 2.1.3 Persamaan Dan Perdaan Penelitian.....	15
Tabel 2.1.4 Persamaan Dan Perdaan Penelitian.....	16
Tabel 2.1.5 Persamaan Dan Perdaan Penelitian.....	17
Tabel 2.1.6 Persamaan Dan Perdaan Penelitian.....	17
Tabel 2.13.1 Kata Kunci Python.....	30
Tabel 2.13.2 Tipe Data Python	31
Tabel 2.13.3 Oprator Aritmatika Python	31
Tabel 2.13.4 Operator Penugasan Python.....	32
Tabel 2.13.5 Oprator Pembandingan Python	33
Tabel 2.13.6 Operator Logika	33
Tabel 2.14.1 Spesifikasi PC/Leptop.....	35
Tabel 4.6.1 Nama setiap koordinat	44
Tabel 4.7.1 Penerapan Algoritma Dijkstra.....	45
Tabel 4.7.2 Panjang Lintasan Yang Akan Ditempuh.....	46
Tabel 4.9.1 Fitur - Fitur Aplikasi	49
Tabel 5.1.1 Spesifikasi Perangkat Keras.....	77
Tabel 5.1.2 Spesifikasi Perangkat Lunak.....	77

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada era perkembangan ilmu pengetahuan saat ini banyak orang yang memiliki rasa ingin tahu yang tinggi, pada dasarnya mereka pasti mencari sumber-sumber yang mendukung untuk pengetahuan mereka, contohnya seperti buku, jurnal, dan artikel. Yang dapat membantu mereka untuk menyelesaikan masalah yang mereka temui contohnya tugas sekolah, kuliah, dan masalah yang ada pada perusahaan mungkin ada ada solusi dapat di temukan dalam jurnal dan artikel tersebut. Akan tetapi saat kita mencari referensi dan artikel di google terkadang tak dapat ditemukan maka dari itu mahasiswa atau mahasiswi baik dari sekolah maupun dari berbagai kampus memutuskan untuk mencari ke perpustakaan sekolah, kampus dan mungkin lebih banyak ke perpustakaan daerah, contohnya seperti perpustakaan yang ada di daerah Bandung lebih tepatnya di jalan Kawalayaan Bandung disana terdapat sebuah perpustakaan daerah Bandung.

Perpustakaan Kawalayaan adalah sebuah gedung yang mana terdapat ruangan-ruangan, yang digunakan untuk menyimpan koleksi bahan pustaka (buku atau monograf, terbitan berseri, brosur, atau pamflet dan bahan non pustaka). Koleksi bahan pustaka pada sebuah perpustakaan digunakan oleh Pemustaka (pengguna dan pembaca) bukan untuk diperjual-belikan, itulah perpustakaan dalam paradigma lama. Sedangkan dalam paradigma baru perpustakaan adalah sesuatu yang hidup, dinamis, segar menawarkan hal-hal yang baru, produk layannya inovatif, dan dikemas sedemikian rupa, sehingga apapun yang ditawarkan oleh perpustakaan akan menjadi atraktif, interaktif, edukatif, dan rekreatif bagi pengunjungnya.

Dengan adanya perpustakaan yang dibuka oleh pemerintah, dengan demikian banyak mahasiswa atau mahasiswi yang datang mencari buku untuk menjadi acuan berbagai tugas contohnya penulis sendiri yang sering ke Perpustakaan Kawalayaan Bandung untuk mencari reprensi skripsi, dan jarak yang penulis tempuh dari kos san penulis ke perpustakaan Kewalayaan lumayan jauh maka dari pengalaman penulis selama ini yang sering ke Perpustakaan Kawalayaan, dan juga

untuk meminimalisir penggunaan bensin motor, maka penulis mencari rute terdekat atau terpendek untuk menuju ke Perpustakaan Kawalayaan.

Berdasarkan permasalahan yang telah diuraikan diatas dapat diselesaikan dengan sebuah algoritma yaitu algoritma Dijkstra. Algoritma Dijkstra ini akan di gunakan dalam memecahkan sebuah permasalahan rute terpendek (shortest path problem) untuk sebuah graf berarah (directed graph).

Sesuai dengan ilustrasi yang diuraikan pada pendahuluan, maka penulis mengangkat topik skripsi ini mengenai salah satu tujuan dalam sustainable development goals (SDG) yaitu untuk mempermudah mahasiswa atau mahasiswa untuk mencari rute perpustakaan terdekat supaya mengefesiensi waktu, bensin motor dan juga jarak yang ditempuh tidak begitu jauh. Sehingga judul yang di ambil *“Analisis Kinerja Implementasi Algoritma Dijkstra Untuk Mencari Rute Terdekat Dari Baleendah Ke Perpustakaan Kawalayaan Dengan Menggunakan Python”*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat diketahui permasalahan yang dapat dikaji lebih lanjut, Adapun permasalahan tersebut adalah sebagai berikut:

- a. Bagaiman cara mengetahui rute terdekat dengan menggunakan algoritma dijkstra ?
- b. Bagaima cara kerja algoritma dijkstara dalam mencari rute terdekat?

1.3 Batasan Masalah

Batasan masalah terhadap penelitian ini adalah sebagai berikut:

- a. Menganalisis model pemetaan untuk mencari rute terdekat dari Baleendah ke Perpustakaan Kawalayaan.
- b. Membuat program untuk mencari rute terdekat dan cepat dengan menggunakan python.

1.4 Tujuan

Tujuan yang ingin dicapai dengan dibuatnya penelitian akhir ini berjudul *“Analisis Kinerja Implementasi Algoritma Dijkstra Untuk mencari Rute Terdekat Dari Baleendah Ke Perpustakaan Kawalayaan Dengan Menggunakan Python”* adalah sebagai berikut :

1. Membantu dalam mengefesien waktu, jarak tempuh dan mengirit bensin motor.
2. Dan juga mampu mencari rute terbaik dari kecamatan Baleendah ke Perpustakaan Kawalayaan dengan rute jarak terdekat.

1.5 Metodologi Penelitian

Metode penelitian merupakan sebuah cara untuk mengetahui hasil dari sebuah permasalahan spesifik, yang mana permasalahan tersebut disebut juga dengan permasalahan penelitian. Metode penelitian terdiri dari metode pengumpulan data.

Metode pengumpulan data yang dilakukan dalam penelitian adalah :

1. Observasi

Pengumpulan data dengan melakukan pengamatan secara langsung terhadap objek penelitian, dengan mencatat hal-hal penting yang berhubungan dengan judul penelitian, sehingga diperoleh data yang lengkap dan akurat.

2. Studi Pustaka

Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna untuk memperoleh gambaran secara teoritis yang dapat menunjang pada penyusunan laporan penelitian.

1.6 Sistematika Penulisan

Penyusunan laporan penelitian ini menggunakan kerangka pembahasan yang terbentuk dalam susunan bab, dengan uraian sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisikan tentang uraian Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Metode Penelitian, Dan Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan tentang teori-teori atau metode yang digunakan dalam menyelesaikan masalah dan referensi masalah yang terkait dengan penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisi penjelasan tentang metodologi penelitian yang digunakan untuk menganalisis strategi dalam melakukan penelitian.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini memuat hasil analisis dan perancangan strategi dalam melakukan perbandingan algoritma yang digunakan untuk menyelesaikan permasalahan yang di ambil dalam skripsi ini.

BAB V INPLEMENTASI DAN PENGUJIAN

Bab ini berisi tentang bagaimana pengujian sebuah analisis algoritma dan cara mengimplementasikan ke tempat yang telah diteliti.

BAB VI KESIMPULAN DAN SARAN

Bab ini merupakan bab penutupan dari laporan yang isinya berupa kesimpulan dari pembahasan pada bab sebelumnya dan memberikan saran dalam penelitian ini.

BAB II TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori berisi mengenai informasi-informasi penelitian yang berkaitan dengan topik penelitian ini yang pernah dilakukan sebelumnya. Hal ini dimaksudkan untuk melakukan perbandingan mengenai kelebihan dan kekurangan yang sudah ada. Berbagai informasi landasan teori ini didapatkan dari berbagai jurnal-jurnal yang ada.

Tabel 2.1. 1 Tabel Penelitian Terdahulu

No	Judul Riset	Nama Penelitian Dan Tahun	Algoritma Yang Digunakan	Hasil Penelitian
1	Optimization of VSAT IP Installation Routes using the Dijkstra Algorithm	(Suherman & Kurnianda, 2021)	Algoritma Dijkstra	Algoritma Dijkstra terbukti dapat digunakan untuk lihat potensi rute terpendek untuk instalasi VSAT IP di Kecamatan Cimanggis, Kabupaten Beji, dan Bojongsari daerah. Di kecamatan Cimanggis, lokasinya menunjuk memiliki jarak terpendek adalah Telkomsat Cibubur Perkantoran, Desa Harjamukti, Desa Mekarsari, dan Desa Pasir Gunung Selatan, dengan total jarak 12850 m. Di distrik Beji, lokasinya

				menunjukkan bahwa memiliki jarak terpendek adalah Telkomsat Cibubur Perkantoran, Desa Kemiri Muka, Desa Beji Timur, dan Desa Tanah Baru, dengan total jarak 18700 m.
2	Penentuan Rute Terpendek dengan Menggunakan Algoritma Dijkstra pada Jalur Bus Sekolah	(Gautama & Hermanto, 2020)	Algoritma Dijkstra	Peran angkutan umum atau bus sekolah sangat penting dalam mengurangi pelanggaran lalu lintas bagi pengemudi di bawah umur. Transportasi bus sekolah menjadi populer di Bali. Khusus di Kota Denpasar, layanan transportasi Kota Denpasar baru beroperasi pada September 2017. Salah satu optimalisasi yang bisa dilakukan adalah dengan menentukan jarak terpendek dari jalur bus sekolah. Semakin pendek jarak yang

				<p>ditempuh tentunya berdampak pada biaya dan waktu. Biaya yang dikeluarkan dapat diminimalisir dan waktu tempuh lebih efisien. Berdasarkan hasil yang diperoleh, biaya bahan bakar yang dikeluarkan untuk bus sekolah shift pagi sebesar Rp70.132. Hasil tersebut dapat memberikan gambaran bagi Dinas Perhubungan Kota Denpasar mengenai penerapan matematika dalam menentukan rute yang dapat mengoptimalkan biaya bahan bakar.</p>
3	Sistem Pencarian Lokasi Toko Batik Di Wilayah Surabaya Dengan Algoritma Dijkstra	(Arabigh & Saurina, 2020)	Algoritma Dijkstra	Semakin berkembangnya batik saat ini mampu memberikan dampak positif pada kota Surabaya, pada

				<p>dasarnya masih banyak masyarakat maupun para wisatawan yang kesulitan mencari toko batik yang sesuai dengan harapan mereka. Namun karena masih minimnya informasi tentang jarak, harga dan popularitas toko batik menjadikan masyarakat maupun wisatawan masih kebingungan dalam menentukan lokasi toko batik mana yang akan dituju. Berdasarkan permasalahan yang telah dipaparkan maka di penelitian ini menggunakan metode Algoritma Dijkstra dengan pembuatan aplikasi penentuan jalur terpendek pencarian lokasi toko batik di Surabaya dengan metode Algoritma Dijkstra berbasis web. Hasil dari</p>
--	--	--	--	---

				<p>penelitian ini dapat disimpulkan bahwa berdasarkan pengambilan data yang diperoleh, penyusunan rute toko batik di surabaya dengan masing masing titik sudah dapat disusun dalam bentuk graph berbobot, membandingkan bobot masing masing lintasan sehingga diambil nilai perbandingan yang paling terkecil. Sehingga Algoritma Dijkstra secara otomatis akan mencari jalur terpendek dari rute tersebut.</p>
4	Implementasi Algoritma Dijkstra Dalam Penentuan Rute Terdekat Menuju Masjid Di Perumahan Bona Indah Lebak Bulus	(Achmad Fatkharrofiqi & Windu Gata, 2022)	Algoritma dijkstra	<p>Penerapan kecerdasan buatan (Artificial Intelligence) untuk pemecahan masalah (problem solving) dalam bidang ilmu komputer telah mengalami perkembangan yang pesat dari tahun ke tahun seiring perkembangan</p>

				<p>kecerdasan buatan itu sendiri. Permasalahan yang melibatkan pencarian (searching) adalah salah satu contoh penggunaan kecerdasan buatan yang cukup populer untuk memecahkan berbagai macam permasalahan. Dalam aktifitas sehari-hari, penggunaan jalan selalu menjadi aktifitas yang tidak dapat dihindari, sehingga penentuan lintasan terpendek dari satu titik ke titik yang lain menjadi masalah yang sering ditemui. Hal ini pula dirasakan oleh warga yang tinggal dalam sebuah perumahan yang cukup luas. Terkadang untuk dapat mencapai tempat yang dituju sering kali bingung dalam</p>
--	--	--	--	---

				<p>menentukan jalan mana yang harus dilalui untuk mendapatkan jarak terpendek sampai ke tujuan. Perumahan Bona Indah merupakan kawasan perumahan di wilayah Kecamatan Cilandak, Kota Jakarta Selatan, di dalam perumahan terdapat sebuah masjid yang berlokasi di sisi pojok selatan perumahan Bona Indah. Banyaknya jumlah blok menimbulkan permasalahan mengenai rute terpendek yang dapat ditempuh oleh warga untuk menuju masjid.</p> <p>Maka, tujuan penelitian ini adalah untuk menentukan jalur terpendek yang ditempuh warga untuk sampai ke masjid. Metode yang digunakan adalah dengan menerapkan Algoritma</p>
--	--	--	--	---

				Dijkstra yang mampu menghasilkan rute terpendek bagi warga untuk menuju ke masjid.
5	Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta	(Aldy Cantona, Fauziah & Winarsih, 2020)	Algoritma Dijkstra	<p>Persoalan jarak dan awamnya pencarian rute adalah masalah setiap orang ketika mengunjungi tempat-tempat tertentu. Pada penelitian ini, algoritma Dijkstra digunakan untuk mengkalkulasi jarak paling dekat dari satu titik ke museum yang dipilih menjadi tujuan. Pada aplikasi ini dirancang agar beroptimasi mencari rute ke museum dengan bobot jarak paling pendek di kota Jakarta. Dalam proses pencarian mampu diselesaikan menggunakan metode algoritma dijkstra sehingga kita mampu mengetahui tahapan kerja aplikasi dengan solusi perhitungan</p>

				<p>metode algoritma dijkstra. Mencari jalur terpendek dengan metode algoritma dijkstra dalam bentuk graf dengan bobot dan pemetaan area yang saling terhubung melalui jalur yang sudah ditentukan. Aplikasi ini dirancang agar berjalan pada smartphone dengan platform android yang digunakan mayoritas pada saat ini. Berdasarkan pengujian yang dilakukan, penerapan algoritma Dijkstra untuk mendapatkan rute terpendek dinilai efektif sebab dengan total 20 bobot hanya memerlukan 7 bobot, yang mana hal tersebut menyatakan hanya perlu 35% bobot untuk mencapai Museum Nasional.</p>
--	--	--	--	---

1. Jurnal yang ditulis oleh (Suherman & Kurnianda, 2021) dari Fakultas Ilmu Komputer, Universitas Mercu Buana, Jakarta. Yang diberi judul

“Optimization Of VSAT IP Installation Routes Using The Dijkstra Algorithm” pada penelitian ini di angkatlah sebuah permasalahan yang mana permasalahan tersebut sering ditemui, salah satunya yang merupakan keterlambatan atau melebihi periode yang telah ditentukan, sehingga penyedia layanan harus membayar beberapa biaya kompensasi. Ini tidak diragukan lagi masalah serius. Jika boleh untuk melanjutkan, itu memang akan mempengaruhi biaya operasional dan kinerja perusahaan. Oleh karena itu, kita membutuhkan solusi untuk meminimalkan kerugian tersebut. Salah satu solusi yang dapat ditawarkan adalah untuk mengetahui titik lokasi mana yang dimiliki jarak terdekat dengan PT. Telkom Satelit Indonesia (Telkomsat).

Algoritma Dijkstra merupakan salah satu algoritma yang dapat membantu mengoptimalkan rute terpendek dari suatu lokasi. Dengan mengimplementasikan algoritma Dijkstra, diharapkan sehingga dapat membantu dalam pengambilan keputusan terdekat lokasi pasang baru.

Tabel 2.1. 2 Persamaan Dan Perdaan Penelitian

Persamaan Dalam Penelitian	Perbedaan Dengan Penelitian Sebelumnya
Adapun persamaan dalam penelitian ini dengan penelitian sebelumnya yaitu menggunakan algoritma yang sama yang mana algoritma tersebut adalah algoritma Dijkstra.	Dan adapun perbedaan penelitian ini dengan penelitian sebelumnya yaitu di implementasi kan di kecamatan Cimanggis, lokasinya menunjuk memiliki jarak terpendek adalah Telkomsat Cibubur Perkantoran, Desa Harjamukti, Desa Mekarsari, dan Desa Pasir Gunung Selatan, dengan total jarak 12850 m.

- Menurut jurnal yang di tulis oleh (Gautama & Hermanto, 2020) dari Universitas Udayana menjelaskan bahwa Peran angkutan umum atau bus sekolah sangat penting dalam mengurangi pelanggaran lalu lintas bagi pengemudi di bawah umur. Transportasi bus sekolah menjadi populer di Bali. Khusus di Kota Denpasar, layanan transportasi Kota Denpasar baru

beroperasi pada September 2017. Salah satu optimalisasi yang bisa dilakukan adalah dengan menentukan jarak terpendek dari jalur bus sekolah. Semakin pendek jarak yang ditempuh tentunya berdampak pada biaya dan waktu. Biaya yang dikeluarkan dapat diminimalisir dan waktu tempuh lebih efisien. Berdasarkan hasil yang diperoleh, biaya bahan bakar yang dikeluarkan untuk bus sekolah shift pagi sebesar Rp70.132. Hasil tersebut dapat memberikan gambaran bagi Dinas Perhubungan Kota Denpasar mengenai penerapan matematika dalam menentukan rute yang dapat mengoptimalkan biaya bahan bakar.

Tabel 2.1. 3 Persamaan Dan Perdaan Penelitian

Persamaan Dalam Penelitian	Perbedaan Dengan Penelitian Sebelumnya
Persamaan dalam penelitian ini yaitu pemecahan masalah yang digunakan adalah menggunakan algoritma Dijkstra.	Adapun perbedaan penelitian ini yaitu di implementasi kan dalam menentukan rute terpendek bus menuju ke sekolah.

- Adapun Jurnal yang ditulis oleh (Arabigh & Saurina, 2020) dari Universitas Wijaya Kusuma Surabaya, menjelaskan dalam artikelnya yaitu Semakin berkembangnya batik saat ini mampu memberikan dampak positif pada kota Surabaya, pada dasarnya masih banyak masyarakat maupun para wisatawan yang kesulitan mencari toko batik yang sesuai dengan harapan mereka. Namun karena masih minimnya informasi tentang jarak, harga dan popularitas toko batik menjadikan masyarakat maupun wisatawan masih kebingungan dalam menentukan lokasi toko batik mana yang akan dituju. Berdasarkan permasalahan yang telah dipaparkan maka di penelitian ini menggunakan metode Algoritma Dijkstra dengan pembuatan aplikasi penentuan jalur terpendek pencarian lokasi toko batik di Surabaya dengan metode Algoritma Dijkstra berbasis web. Hasil dari penelitian ini dapat disimpulkan bahwa berdasarkan pengambilan data yang diperoleh, penyusunan rute toko batik di surabaya dengan masing-masing titik sudah

dapat disusun dalam bentuk graph berbobot, membandingkan bobot masing-masing lintasan sehingga diambil nilai perbandingan yang paling terkecil. Sehingga Algoritma Dijkstra secara otomatis akan mencari jalur terpendek dari rute tersebut.

Tabel 2.1. 4 Persamaan Dan Perdaan Penelitian

Persamaan Dalam Penelitian	Perbedaan Penelitian Ini Dengan Penelitian Sebelumnya
Persamaan dalam penelitian ini adalah menggunakan algoritma Dijkstra.	Adapun perbedaan dalam penelitian ini yaitu di diimplementasikan dalam bentuk web.

4. Jurnal yang ditulis oleh (Achmad Fatkharrufiqi & Windu Gata, 2022) dari Universitas Mandiri yang menjelaskan dalam jurnalnya yang bahwa Penerapan kecerdasan buatan (Artificial Intelligence) untuk pemecahan masalah (problem solving) dalam bidang ilmu komputer telah mengalami perkembangan yang pesat dari tahun ke tahun seiring perkembangan kecerdasan buatan itu sendiri. Permasalahan yang melibatkan pencarian (searching) adalah salah satu contoh penggunaan kecerdasan buatan yang cukup populer untuk memecahkan berbagai macam permasalahan. Dalam aktifitas sehari-hari, penggunaan jalan selalu menjadi aktifitas yang tidak dapat dihindari, sehingga penentuan lintasan terpendek dari satu titik ke titik yang lain menjadi masalah yang sering ditemui. Hal ini pula dirasakan oleh warga yang tinggal dalam sebuah perumahan yang cukup luas. Terkadang untuk dapat mencapai tempat yang dituju sering kali bingung dalam menentukan jalan mana yang harus dilalui untuk mendapatkan jarak terpendek sampai ke tujuan. Perumahan Bona Indah merupakan kawasan perumahan di wilayah Kecamatan Cilandak, Kota Jakarta Selatan, di dalam perumahan terdapat sebuah masjid yang berlokasi di sisi pojok selatan perumahan Bona Indah. Banyaknya jumlah blok menimbulkan permasalahan mengenai rute terpendek yang dapat ditempuh oleh warga untuk menuju masjid. Maka, tujuan penelitian ini adalah untuk menentukan jalur terpendek yang ditempuh warga untuk sampai ke masjid. Metode yang

digunakan adalah dengan menerapkan Algoritma Dijkstra yang mampu menghasilkan rute terpendek bagi warga untuk menuju ke masjid.

Tabel 2.1. 5 Persamaan Dan Perdaan Penelitian

Persamaan Dalam Penelitian	Perbedaan Dalam Penelitian
Persamaan dalam penelitian ini yaitu menggunakan algoritma yang sama yaitu algoritma Dijkstra.	Adapun perbedaan dalam penelitian ini adalah diimplementasikan dalam android.

5. Berdasarkan jurnal yang ditulis oleh (Aldy Cantona, Fauziah dan Winarsih) dari Universitas Nasional dan dari Fakultas Teknologi Komunikasi dan Informatika yang menjelaskan bahwa Persoalan jarak dan awamnya pencarian rute adalah masalah setiap orang ketika mengunjungi tempat-tempat tertentu. Pada penelitian ini, algoritma Dijkstra digunakan untuk mengkalkulasi jarak paling dekat dari satu titik ke museum yang dipilih menjadi tujuan. Dalam proses pencarian mampu diselesaikan menggunakan metode algoritma dijkstra sehingga kita mampu mengetahui tahapan kerja aplikasi dengan solusi perhitungan metode algoritma dijkstra. Mencari jalur terpendek dengan metode algoritma dijkstra dalam bentuk graf dengan bobot dan pemetaan area yang saling terhubung melalui jalur yang sudah ditentukan. Aplikasi ini dirancang agar berjalan pada smartphone dengan platform android yang digunakan mayoritas pada saat ini. Berdasarkan pengujian yang dilakukan, penerapan algoritma Dijkstra untuk mendapatkan rute terpendek dinilai efektif sebab dengan total 20 bobot hanya memerlukan 7 bobot, yang mana hal tersebut menyatakan hanya perlu 35% bobot untuk mencapai Museum Nasional.

Tabel 2.1. 6 Persamaan Dan Perdaan Penelitian

Persamaan Dalam Penelitian	Perbedaan Dalam Penelitian
Persamaan dalam penelitian ini adalah menggunakan algoritma yang sama yaitu algoritma Dijkstra.	Dan Adapun perbedaan dalam penelitian ini adalah diimplementasikan dalam pemrograman android.

2.2 Dasar Teori

2.2.1 Pengertian Algoritma

Pengertian algoritma adalah urutan aksi-aksi yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah dalam rentang waktu tertentu. Setiap aksi harus dapat dikerjakan dan mempunyai efek tertentu. Algoritma merupakan logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan. Dalam ilmu matematika dan komputer, pengertian algoritma merupakan prosedur dari beberapa langkah demi langkah untuk penghitungan. Algoritma dipakai untuk penghitungan, penalaran otomatis, dan pemrosesan data.

Pengertian algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu software. Dalam dunia sehari-hari, mungkin tanpa kita sadari algoritma telah masuk dalam kehidupan kita (Maulana, 2017).

Algoritma dapat dituliskan dengan banyak cara, mulai dari menggunakan bahasa alami yang digunakan sehari-hari, simbol grafik bagan alir (flowchart), sampai menggunakan bahasa pemrograman seperti bahasa C atau C++. Penulisan algoritma tidak terikat pada suatu aturan tertentu, tetapi harus jelas maksudnya dari suatu penulisan algoritma untuk tiap langkah algoritma nya.

2.2.2 Struktur Dasar Algoritma

Adapun struktur algoritma dasar pada algoritma adalah sebagai berikut:

1. Sekuensial (Runtunan)

Pada struktur sekuensial ini langkah-langkah yang dilakukan dalam algoritma diproses secara berurutan. Dimulai dari langkah pertama, kedua, dan seterusnya. Pada dasarnya suatu program memang menjalankan suatu proses dari yang dasar seperti struktur ini.

2. Struktur Seleksi

Struktur seleksi menyatakan pemilihan langkah yang didasarkan oleh suatu kondisi atau pengambilan suatu keputusan. Struktur ini ditandai selalu

dengan bentuk flowchart decision (flowchart yang berbentuk belah ketupat).

3. Struktur perulangan

Struktur ini memberikan suatu perintah atau tindakan yang dilakukan beberapa kali. Misalnya jika teman mau menuliskan kata “belajar c” sebanyak sepuluh kali. Akan lebih efisien jika teman menggunakan struktur ini dari pada sekedar menuliskannya berturut-turut sebanyak sepuluh kali.

2.2.3 Analisis

Menurut kamus besar Bahasa Indonesia (KBBI), analisis yaitu:

1. penyelidikan terhadap suatu peristiwa (karangan, perbuatan, dan sebagainya) untuk mengetahui keadaan yang sebenarnya (sebab-musabab, duduk perkaranya, dan sebagainya).
2. penguraian suatu pokok atas berbagai bagiannya dan penelaahan bagian itu sendiri serta hubungan antarbagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan.
3. penyelidikan kimia dengan menguraikan sesuatu untuk mengetahui zat bagiannya dan sebagainya.
4. penjabaran sesudah dikaji sebaik-baiknya.
5. pemecahan persoalan yang dimulai dengan dugaan akan kebenarannya.

2.2.4 Kinerja

Dalam kamus besar bahasa Indonesian (KBBI), kinerja yaitu:

1. Suatu yang dicapai.
2. Prestasi yang diperhatikan.
3. Kemampuan kerja (tentang peralatan).

Menurut (Chusnah & Purwanti, 2019) menjelaskan di dalam artikel nya bahwa Kinerja (prestasi kerja) adalah hasil kerja secara kualitas dan kuantitas yang dicapai oleh seseorang dalam melaksanakan tugasnya sesuai dengan tanggung jawab yang diberikan kepadanya.

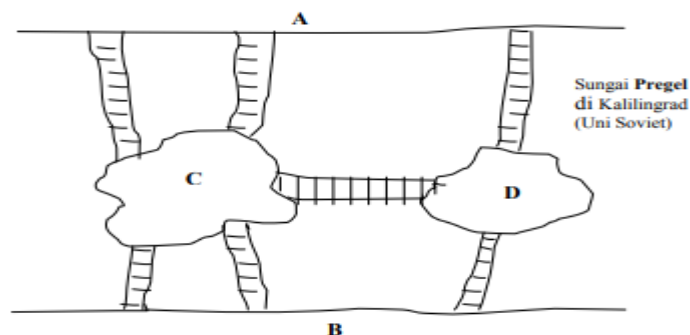
2.2.5 Implementasi

Implementasi adalah “sesuatu hal yang bermuara pada aksi, aktifitas, tindakan, serta adanya mekanisme dari suatu sistem. Implementasi tidak hanya sekedar aktifitas monoton belaka, tetapi merupakan suatu kegiatan yang terencana secara baik untuk mencapai tujuan tertentu” (Usman, 2002).

2.3 Dasar Graf

2.3.1 Kelahiran Teori Graf

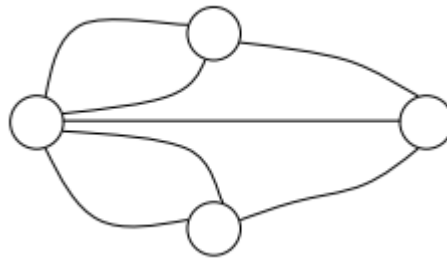
Teori Graf mulai dikenal pada saat seorang matematikawan bangsa Swiss, Bernama Leonhard Euler, berhasil mengungkapkan Misteri Jembatan Konigsberg pada tahun 1736. Di Kota Konigsberg (sekarang bernama Kaliningrad, di Uni Soviet) mengalir sebuah sungai bernama sungai Pregel. Di tengah sungai tersebut terdapat dua buah pulau. Dari kedua pulau tersebut terdapat jembatan yang menghubungkan ke tepian sungai dan diantara kedua pulau. Jumlah jembatan tersebut adalah 7 buah seperti gambar berikut:



Gambar 2.3.1 Jembatan Konigsberg

Konon kabarnya, penduduk kota Konigsberg sering berjalan-jalan ke tempat tersebut pada hari-hari libur. Kemudian muncul suatu keinginan untuk dapat menikmati daerah tersebut dengan melalui ketujuh jembatan tepat satu kali, yakni bermula dari satu tempat (A, B, C atau D) dan kembali ke tempat semula. Mereka berusaha untuk memperoleh rute yang sesuai dengan keinginan tersebut, dengan selalu mencoba menjalaninya. Setelah mencoba berkali-kali dan karena sudah cukup lama tidak diperoleh rutennya, akhirnya penduduk tersebut mengirim surat kepada Euler. Euler dapat memecahkan masalah tersebut, yakni bahwa perjalanan / rute yang diinginkan (yakni berawal dari suatu tempat, melalui ketujuh jembatan tepat satu kali, dan kembali ke tempat semula) tidak mungkin dicapai.

Secara singkat, dalam tulisanya, Euler menyajikan keadaan jembatan Königsberg disebut seperti gambar berikut:



Gambar 2.3. 2 Penyajian Jembatan Königsberg

Dalam masalah di atas, daratan (tepi A dan B, serta pulau C dan D) disajikan sebagai titik dan jembatan disajikan sebagai ruas garis. Euler mengemukakan teoremanya yang mengatakan bahwa perjalanan yang diinginkan di atas (yang kemudian dikenal sebagai perjalanan Euler) akan ada apabila graf terhubung dan banyaknya garis yang datang pada setiap titik (derajat simpul) adalah genap.

a. Problema & Model Graf

Secara umum, langkah-langkah yang perlu dilalui dalam penyelesaian suatu masalah dengan bantuan komputer adalah sebagai berikut:

Problema → Model Yang Tepat → Algoritma → Program Komputer

b. Graf Secara Formal

Sebuah Graf G mengandung 2 himpunan:

(1). Himp. V , yang elemennya disebut simpul

→ Vertex / point / titik / node

(2). Himp. E , yang merupakan pasangan tak terurut dari simpul-simpul, disebut ruas → Edge / rusuk / sisi Sehingga sebuah graf dinotasikan sebagai $G(V, E)$

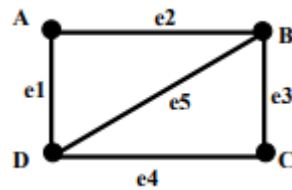
Contoh:

$G(V, E)$

$V = \{A, B, C, D\}$

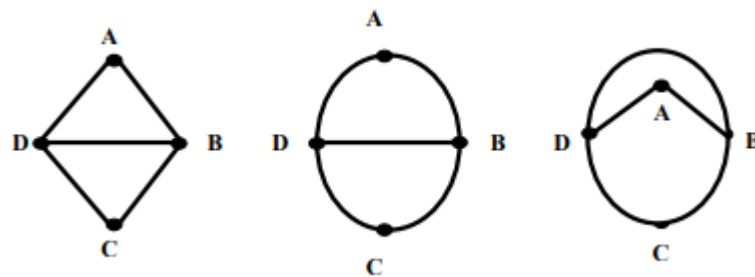
$E = \{(A, B), (B, C), (C, D), (D, A), (B, D)\}$

Secara Geometri:



Gambar 2.3. 3 Terdiri dari 4 ruas dan 5 simpul

Tidak ada ketentuan khusus dalam penyajian graf secara geometri, seperti dimana dan bagaimana menyajikan simpul dan ruas. Berikut contoh penyajian Graf yang sama, tetapi disajikan berbeda.

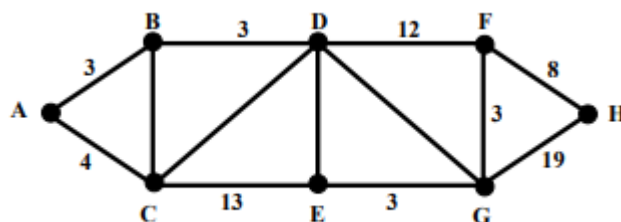


Gambar 2.3. 4 Graf yang sama dan tapi disajikan berbeda

c. Graf Berlebel

Graf berlebel atau berbobot adalah graf yang setiap ruasnya mempunyai nilai/bobot berupa bilangan non negatif.

Contoh:



Gambar 2.3. 5 Graf berlebel

d. Operasi pada Graf

Berdasarkan definisi graf (yang terdiri dari 2 himpunan) dan operasi pada himpunan, maka pada graf juga dapat dilakukan operasi-operasi. Bila diketahui 2 buah graf: $G_1(V_1, E_1)$ dan $G_2(V_2, E_2)$, maka:

1. Gabungan $G_1 \cup G_2$ adalah graf dengan himpunan V nya = $V_1 \cup V_2$ dan himpunan E nya = $E_1 \cup E_2$
2. Irisan $G_1 \cap G_2$ adalah graf dengan himpunan V nya = $V_1 \cap V_2$ dan himpunan E nya = $E_1 \cap E_2$
3. Selisih $G_1 - G_2$ adalah graf dengan himpunan V nya = V_1 dan himpunan E nya = $E_1 - E_2$

Sedangkan Selisih $G_2 - G_1$ adalah graf dengan himpunan V nya = V_2 dan himpunan E nya = $E_2 - E_1$

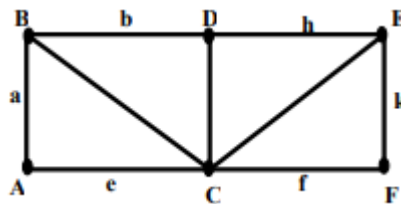
4. Penjumlahan Ring $G_1 \oplus G_2$ adalah graf yang dihasilkan dari $(G_1 \cup G_2) - (G_1 \cap G_2)$ atau $(G_1 - G_2) \cup (G_2 - G_1)$

2.3.2 Keterhubungan Graf

Dalam keterhubungan sebuah graf, akan dikenal beberapa istilah-istilah berikut:

1. Walk: barisan simpul dan ruas
2. Trail: Walk dengan ruas yang berbeda
3. Path / Jalur: Walk dengan simpul yang berbeda
4. Cycle / Sirkuit: Trail tertutup dengan derajat setiap simpul = 2

Contoh:



Gambar 2.3. 6 Keterhubungan graf

Penjelasan:

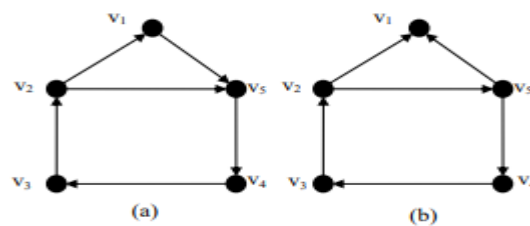
1. A, B, C, D, E, F, C, A, B, D, C \rightarrow Walk
2. A, B, C, D, E, F, C, A \rightarrow Trail
3. A, B, C, A \rightarrow Cycle
4. A, B, D, C, B, D, E \rightarrow Walk
5. A, B, C, D, E, C, F \rightarrow Trail
6. A, B, D, C, E, D \rightarrow Trail

7. A, B, D, E, F, C, A \rightarrow Cycle
8. C, E, F \rightarrow Path
9. B, D, C, B \rightarrow Cycle
10. C, A, B, C, D, E, C, F, E \rightarrow Trail
11. A, B, C, E, F, C, A \rightarrow Trail

2.4 Graf Terhubung (*Connected Graph*)

Graf G disebut graf terhubung jika untuk setiap pasang simpul u dan v di dalam himpunan V terdapat lintasan dari u ke v . Jika tidak, maka graf G disebut graf tak terhubung (*disconnected graph*), (Nugraha, 2011).

Keterhubungan dua buah simpul adalah penting di dalam graf. Jika dua buah simpul terhubung maka pasti simpul yang pertama dapat dicapai dari simpul yang kedua. Pada graf berarah, graf G dikatakan terhubung jika graf tak berarahnya terhubung (graf tak-berarah dari G diperoleh dengan menghilangkan arahnya). Keterhubungan dua buah simpul pada graf berarah dibedakan menjadi terhubung kuat dan terhubung lemah. Dua simpul, u dan v pada graf berarah G disebut terhubung kuat (*strongly connected*) jika terdapat lintasan berarah dari u ke v dan juga sebaliknya lintasan berarah dari v ke u . Jika u dan v tidak terhubung kuat tetapi tetap terhubung pada graf tak berarahnya, maka u dan v dikatakan terhubung lemah (*weakly connected*).

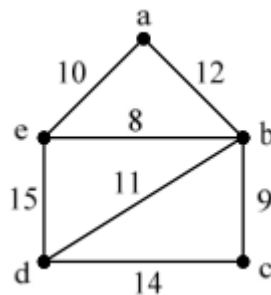


Gambar 2.4. 1 Graf terhubung

Graf pada Gambar 2.4.1 (a) merupakan graf terhubung kuat, karena untuk sembarang sepasang simpul di dalam graf tersebut terdapat lintasan. Sedangkan graf pada Gambar 2.4.1 (b) merupakan graf terhubung lemah, karena tidak semua pasangan simpul mempunyai lintasan arah.

2.5 Graf Berbobot (*weighted Graph*)

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga. Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah tiang listrik, kapasitas, biaya perjalanan antara dua buah kota, waktu tempuh pesan (message) dari sebuah simpul komunikasi ke simpul komunikasi lain, ongkos produksi, dan sebagainya. Untuk lebih jelasnya, graf berbobot dapat digambarkan pada gambar 2.2.



Gambar 2.5 1 Graf Berbobot

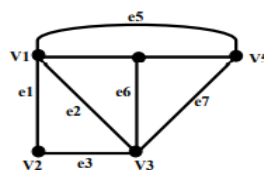
Misalkan terdapat graf yang sangat kompleks, dimana ada beberapa alternatif untuk melakukan kunjungan-kunjungan (visiting) dari satu simpul ke simpul - simpul yang lainnya, tentu dapat segera dicari tahu alternatif yang terbaik untuk melakukannya. Dalam hal ini, alternatif yang cukup baik adalah dengan mencari jarak yang terdekat antar simpul itu.

2.6 Matriks dan Graf

Untuk menyelesaikan suatu permasalahan model graf dengan bantuan komputer, maka graf tersebut disajikan dalam bentuk matriks. Matriks-matriks yang dapat menyajikan model graf tersebut antara lain:

- Matriks ruas
- Matriks adjacency
- Matriks incidence

Sebagai contoh, untuk graf seperti dibawah ini:



Gambar 2.6. 1 Graf

Maka graf juga dapat di selesaikan dengan matriks sebagai berikut:

- Matriks ruas

$$\begin{matrix} & n \times 2 \\ \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 2 & 3 \\ 3 & 4 \\ 3 & 5 \\ 4 & 5 \end{bmatrix} \end{matrix}$$

Atau

$$\begin{matrix} & 2 \times n \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 3 & 3 & 4 \\ 2 & 3 & 4 & 5 & 3 & 4 & 5 & 5 \end{bmatrix} \end{matrix}$$

- Matriks Adjacency:

$$\begin{matrix} & V1 & V2 & V3 & V4 & V5 \\ \begin{matrix} V1 \\ V2 \\ V3 \\ V4 \\ V5 \end{matrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

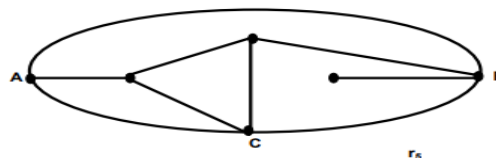
- Matriks Incidence:

$$\begin{matrix} & e1 & e2 & e3 & e4 & e5 & e6 & e7 & e8 \\ \begin{matrix} V1 \\ V2 \\ V3 \\ V4 \\ V5 \end{matrix} \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

2.7 Graf Planar

Sebuah graf dikatakan graf planar bila graf tersebut dapat disajikan (secara geometri) tanpa adanya ruas yang berpotongan. Sebuah graf yang disajikan tanpa adanya ruas yang berpotongan disebut dengan penyajian planar/map/peta.

Pada penyajian planar/map, dikenal istilah region. Derajat dari suatu region adalah panjang walk batas region tersebut.



Gambar 2.7. 1 Planar/map

$$d(r1) = 3$$

$$d(r2) = 3$$

$$d(r3) = 5$$

$$d(r4) = 4$$

$$d(r5) = 3$$

$$\sum = 18 = 2 \times \text{SIZE}$$

Region dengan batasnya gelung, maka $d(r) = 1$

Region dengan batasnya ruas sejajar, maka $d(r) = 2$

Formula Euler untuk graf planar

Untuk Graf Planar berlaku Formula Euler berikut:

$V - E + R = 2$ Dimana p = jumlah simpul dan q = jumlah ruas

2.8 Shortest Path Problem

Jalur pendek adalah suatu jaringan pengarah perjalanan dimana seseorang pengarah jalan ingin menentukan jalur terpendek antara dua seperti jalur A ke B, berdasarkan beberapa jalur alternatif yang tersedia, dimana titik tujuan hanya satu (Hasibuan, 2016).

2.9 Shortest Rute

Graf yang digunakan untuk menentukan lintasan terpendek adalah grafik berbobot. Jika pada lintasan M dan V_0 adalah titik asal atau titik keberangkatan, dan V_n adalah tujuan titik, notasi jalur terpendek adalah P_t , di mana P_t adalah jalur dari V_0 ke V_n (Suherman & Kurnianda, 2021).

$$l_t(V_0, V_n) = \min_{i=1, \dots, m} l_i(V_0, V_n)$$

Gambar 2.9. 1 Rumus Pemecahan Masalah Rute Terdekat

2.10 Algoritma Dijkstra

Algoritma ini dinamakan sesuai dengan nama penemunya, yaitu seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra. Algoritma Dijkstra termasuk ke dalam pembahasan teori graf pada matematika diskrit yang berhubungan dengan graf berbobot dan lintasan terpendek (shortest

path). Algoritma ini digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Cara kerja algoritma Dijkstra memakai strategi greedy, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih (Rinaldi Munir, 2005).

Contoh:

$$V(G) = \{v_1, v_2, v_n\}$$

L = Himpunan titik-titik $\in V(G)$ yang sudah terpilih dalam jalur path terpendek.

$D(j)$ = Jumlah bobot path terkecil dari v_1 ke v_j .

$w(i, j)$ = Bobot garis dari titik v_i ke v_j .

$w^*(1, j)$ = Jumlah bobot path terkecil dari v_1 ke v_j .

Secara formal, algoritma Dijkstra untuk mencari path terpendek adalah sebagai berikut:

1. $L = \{ \}$;
 $V = \{v_2, v_3, \dots, v_n\}$.
2. Untuk $i = 2, \dots, n$, lakukan $D(i) = w(1, i)$
3. Selama $v_n \notin L$ lakukan:
 - a. Pilih titik $v_k \in V - L$ dengan $D(k)$ terkecil.
 $L = L \cup \{v_k\}$
 - b. Untuk setiap $v_j \in V - L$ lakukan:
 Jika $D(j) > D(k) + W(k, j)$ maka ganti $D(j)$ dengan $D(k) + W(k, j)$
4. Untuk setiap $v_j \in V$, $w^*(1, j) = D(j)$

Menurut algoritma di atas, path terpendek dari titik v_1 ke v_n adalah melalui titik-titik dalam L secara berurutan, dan jumlah bobot path terkecilnya adalah $D(n)$.

2.11 Flowchart

Flowchart pada dasarnya merupakan sebuah bentuk gambar ataupun diagram yang memiliki aliran satu atau dua arah yang berlaku secara sekuensial atau berkesinambungan. Fungsi utama dari flowchart ini adalah untuk menggambarkan

sebuah desain program dan untuk mempresentasikan sebuah program atau sistem yang akan dibuat, berdasarkan pola berpikir pengguna (berdasarkan algoritma).

2.12 Google Maps

Google Maps merupakan sebuah layanan yang disediakan oleh pihak Google untuk memberikan kemudahan kepada pengembang aplikasi dalam melakukan proses pemetaan dalam sebuah aplikasi yang dibuat. Sedangkan API (Application Programming Interface) merupakan sebuah dokumentasi yang disediakan dan terdiri dari interface, fungsi, class, struktur dan beberapa fitur lainnya untuk memberikan kemudahan kepada penggunanya untuk membangun sebuah aplikasi tertentu atau perangkat tertentu. Maka dapat disimpulkan bahwa Google Maps API merupakan kumpulan dari API yang memungkinkan penggunanya untuk mendapatkan data dari Google Map yang dapat disesuaikan dengan kebutuhan dari pengembang bersangkutan (Fernando, 2020).

2.13 Kebutuhan Perangkat Lunak

a. Python

Menurut (Wahyudin, 2019) di dalam bukunya edisi pertamanya menjelaskan python sendiri mulai dikembangkan pada tahun 1989 oleh Guido van Rossum, seorang engineer di Google Inc. Pada saat itu, nama Python sendiri bukan berasal dari nama spesies ular, namun berasal dari sebuah grup komedi di Inggris yang bernama Monty Python.

Versi stable perdana Python, yaitu Python 1.0, dirilis pada tahun 1991. Saat ini juga sedang menikmati puncak popularitasnya, menggeser bahasa pemrograman lain yang sudah lebih dulu mendapat tempat di hati para software engineer. Kelebihan Python antara lain adalah sebagai berikut:

1. Ekspresif. Python termasuk bahasa pemrograman yang cukup ekspresif, syntax dan penulisan block programmnya termasuk yang to the point, sehingga cukup mudah digunakan dan dipelajari.

2. Portable. Python juga cukup mudah dalam hal instalasi, tidak perlu dijelaskan lebih lanjut, anda bisa merasakan kemudahannya saat instalasi Anaconda Data Platform.
3. Object Oriented. Python juga sudah mendukung paradigma pemrograman berbasis objek.
4. Mendukung Functional Programming. Lambda Expression, yang merupakan implementasi Functional Programming juga sudah didukung secara penuh pada pemrograman Python.
5. GUI Based. Python juga sudah mendukung pengembangan software dengan user interface berbasis desktop (dengan interface QT) maupun web.

Dari penjelasan diatas, tentunya tools ini dapat penulis gunakan untuk melakukan penelitian ini, yang mana tools ini nantinya digunakan sebagai alat untuk menulis program atau pseudocode.

Python terkenal karena kesederhanaanya dan ketahananya. Selain itu, penggunaan python gratis dan mudah. Berikut beberapa kata kunci yang digunakan pada *python* menurut (Bhasin, 2019).

Tabel 2.13. 1 Kata Kunci Python

<i>and</i>	<i>del</i>	<i>from</i>	<i>not</i>
<i>while</i>	<i>as</i>	<i>elif</i>	<i>global</i>
<i>else</i>	<i>If</i>	<i>pass</i>	<i>yield</i>
<i>break</i>	<i>except</i>	<i>import</i>	<i>class</i>
<i>raise</i>	<i>continue</i>	<i>finally</i>	<i>return</i>
<i>def</i>	<i>for</i>	<i>try</i>	

Python menyediakan penggunaan kamus yang membuat pemetaan mudah dan efisien Bernama tipe data. Tipe data pada python terdiri atas 2 jenis yakni tipe data untuk angka dan kalimat. Berikut merupakan tipe data angka yang terdapat pada python menurut (Bhasin, 2019).

Tabel 2.13. 2 Tipe Data Python

Tipe Data	Penjelasan	Contoh
<i>Integer</i>	Menyatakan bilangan bulat	49
<i>Floating Point</i>	Menyatakan bilangan pecahan atau decimal	2,0
<i>Complex</i>	Menyatakan bilangan real dan imajiner	3+4i
<i>String</i>	Membuat karakter berupa kata atau kalimat dan tidak dapat diubah	'Hello'
<i>List</i>	Kumpulan objek yang nilainya dapat diubah	[1,"abc",2.4]
<i>Tuple</i>	Membuat element yang dapat diperlakukan sebagai data individu atau kelompok dan terdiri dari unsur yang berbeda	(1, "abc", 2.4)

Pada *python* juga terdapat oprator yang membantu dalam perhitungan matematika. Berikut oprator aritmarika menurut (Amos, Bader dan Jablonski, 2020).

Tabel 2.13. 3 Oprator Aritmatika Python

Operator	Keterangan	Contoh
+	Penjumlahan	2 + 3 outputnya 5
-	Pengurangan	3 – 2 outputnya 1
*	Perkalian	2 * 2 outputnya 4
/	Pembagian	9 / 3 outputnya 3
//	Pembagian bulat	3 // 2 outputnya 2
**	Pangkat	2 ** 2 outputnya 4
%	Modulo	5 % 3 outputnya 2

Python memiliki oprator penugasan yang berguna untuk menempatkan data ke dalam variable. Berikut oprator penugasan pada *Python* yang disampaikan oleh (Mueller, 2018).

Tabel 2.13. 4 Operator Penugasan Python

Operator	Keterangan	Contoh
=	Menetapkan nilai di operan kanan menjadi operan kiri.	$x = 3$ outputnya 3
+=	Nilai pada operan kiri dijumlahkan dengan nilai operan kanan	$x += 2$ outputnya 5 karena $x = 3$ maka $3 + 2 = 5$
-=	Nilai pada operan kiri dikurangi dengan nilai operan kanan	$x -= 2$ outputnya 1 karena $x = 3$ maka $3 - 2 = 1$
*=	Nilai pada operan kiri dikalikan dengan nilai pada operan kanan	$x *= 2$ outputnya 6 karena $x = 3$ maka $3 * 2 = 6$
/=	Nilai pada operan kiri dibagi dengan nilai pada operan kanan	$x /= 2$ outputnya 1,5 karena $x = 3$ maka $3 / 2 = 1,5$
%=	Nilai pada operan kiri dibagi dengan nilai pada operan kanan kemudian sisa pembagian merupakan output yang dihasilkan	$x \% = 2$ outputnya 1 karena $x = 3$ maka $3 \% 2 = 1$
**=	Nilai pada operan kiri dipangkatkan dengan nilai pada operan kanan	$x ** = 2$ outputnya 9 karena $x = 3$ maka $3 ** 2 = 9$
//=	Nilai pada operan kiri dibagi dengan nilai pada operan kanan dan output yang dihasilkan berupa hasil bagi yang bulat	$x //= 2$ outputnya 1 karena $x = 3$ maka $3 // 2 = 1$

Pada *Python* terdapat operator perbandingan yang berguna untuk membandingkan operan kiri dengan operan kanan. Berikut oprator perbandingan pada *Python* menurut (Mueller, 2018).

Tabel 2.13. 5 Oprator Pembandingan Python

Operator	Keterangan
==	Sama dengan
!=	Tidak sama dengan
>	Lebih dari
<	Kurang dari
>=	Lebih dari atau sama dengan
<=	Kurang dari atau sama dengan

Python memiliki oprator logika yang berguna untuk menentukan nilai kebenaran dari suatu nilai. Berikut operator logika pada python yang disampaikan oleh (Mueller, 2018).

Tabel 2.13. 6 Operator Logika

Operator	Keterangan	Contoh
<i>and</i>	Menentukan kedua operator benar	Benar and Benar adalah Benar Benar and Salah adalah Salah Salah and Benar adalah Salah Salah and Salah adalah Salah
<i>or</i>	Menentukan nilai Ketika salah satu dari dua operan benar	Benar or Benar adalah Benar Benar or Salah adalah Benar Salah or Benar adalah Benar Salah or Salah adalah Salah
<i>not</i>	Menegasikan nilai kebenaran dari suatu operan	Not Benar adalah Salah Not Salah adalah Benar

Terdapat 3 pernyataan dalam pengambilan keputusan pada Python antara lain *if*, *if-else*, dan *if-elif-else*. Pernyataan *if* dapat berjalan jika kondisi yang diberikan memenuhi. Berikut format pernyataan *if*.

```
if <kondisi>
    <pernyataan>
```

Pengambilan keputusan *if-else* pernyataan pada *if* akan berjalan jika kondisi

bernilai benar namun jika bernilai salah maka yang akan berjalan adalah pernyataan pada *else*. Format pernyataan *if-else* dapat dituliskan sebagai berikut.

if <kondisi>:

<pernyataan 1>

else:

<pernyataan 2>

Prinsip pada pengambilan keputusan *if-elif-else* sama dengan *if-else*.

Perbedaannya adalah kondisi *else* lebih banyak sehingga kondisir tersebut akan dimasukkan pada *elif*. Berikut format pernyataan *if-elif-else*.

if <kondisi 1>:

<pernyataan 1>

elif <kondisi 2>:

<pernyataan 2>

elif <kondisi 3>:

<pernyataan 3>

else:

<pernyataan 4>

Pada Python terdapat 3 perulangan yakni *while*, *for*, dan *nested*. Perulangan *while* akan terus berjalan selama kondisi bernilai benar. Berikut format perulangan *while*.

while <kondisi>:

<pernyataan>

Perulangan *for* akan melakukan pengulangan beberapa kondisi dalam suatu objek. Format perulangan *for* dapat dituliskan sebagai berikut:

for i in <objek>:

<perulangan>

Perulangan berbentuk *nested loops* akan memuat perulangan di dalam perulangan.

Bahasa pemrograman Python dapat digunakan juga dalam menyelesaikan permasalahan matematika karena memuat operator aritmatika. Sehingga bahasa pemrograman Python juga dapat berguna

dalam menentukan penyelesaian algoritma Dijkstra untuk menentukan rute terpendek.

b. Flowchart Maker & Online Diagram Software

Flowchart maker adalah sebuah situs yang menyediakan perancangan UML dan sebagainya dalam konteks perancangan. Flowchart maker ini bisa digunakan secara gratis dan dapat digunakan dalam berbagai hal perancangan seperti Process Diagram, Org Chart, UML, ER, network diagram dan masih banyak juga fitur – fitur lainnya. Adapun link yang dapat dikunjungi sebagai berikut: <https://app.diagrams.net/>

2.14 Kebutuhan Perangkat Keras (*Hardware*)

Berdasarkan kebutuhan perangkat keras yang akan digunakan dalam penelitian ini yaitu:

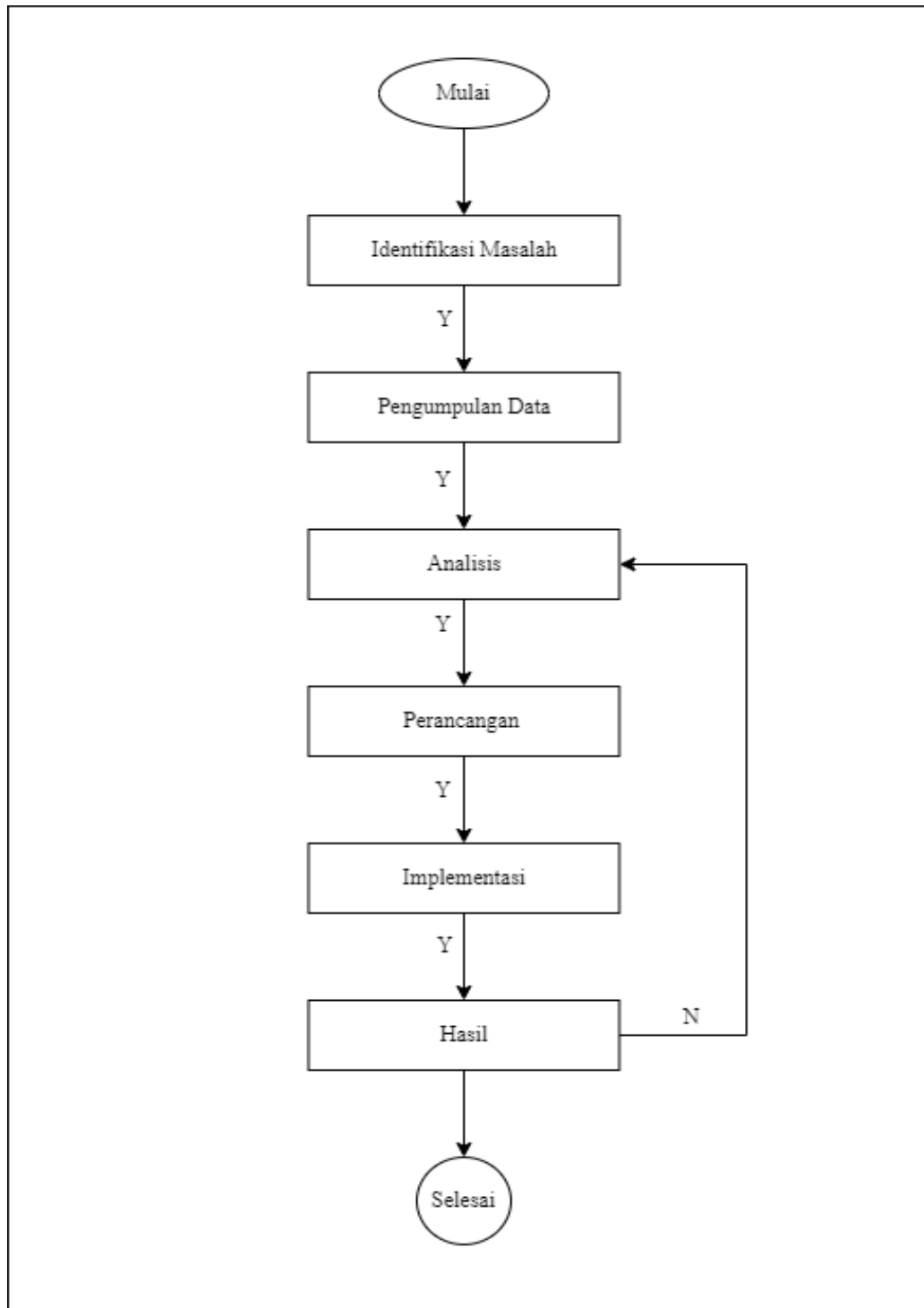
Tabel 2.14. 1 Spesifikasi PC/Leptop

Operating Sistem	Windows 10 Pro
Processor	Intel(R) Core (TM) i3-6006U CPU @ 2.00GHz 1.99 GHz
RAM	8,00 GB (7,88 GB usable)
Hardisk	500 GB
System type	64-bit
Ukuran PC/Laptop	14 inc

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Pikir



Gambar 3. 1 Kerangka Pikir

3.2 Deskripsi

3.2.1 Identifikasi Masalah

Identifikasi masalah adalah tahap permulaan dari penugasaan masalah di mana objek dalam suatu jalinan tertentu bisa kita kenali sebagai suatu masalah. Berdasarkan pemaparan di bab satu, penulis sudah memaparkan masalah yang akan penulis teliti dan adapun identifikasi masalah ini penulis lakukan berdasarkan fakta yang peneliti alami sendiri.

3.2.2 Pengumpulan Data

Dalam penelitian ini penulis melakukan pengumpulan data dengan pengamatan secara langsung terhadap objek penelitian, dengan mencatat hal-hal penting yang berhubungan dengan penelitian yang dilakukan penulis, sehingga penulis dapat memperoleh data yang lengkap dan akurat.

3.2.3 Analisis

Analisis adalah penguraian suatu pokok atas berbagai bagiannya dan penelaahan bagian itu sendiri serta hubungan antar bagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan dari analisis data yang penulis lakukan.

3.2.4 Perancangan Atau Pengkodean

Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya.

Berdasarkan definisi diatas dapat disimpulkan bahwa perancangan adalah tahapan setelah analisis yang tujuannya untuk menghasilkan rancangan yang memenuhi kebutuhan yang ditentukan selama tahap analisis dan tahapan ini juga untuk membuat tools yang nantinya digunakan sebagai tampilan desktop tampilan ini nanti untuk menampilkan hasil dari program.

3.2.5 Implementasi

Implementasi adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci. Implementasi biasanya dilakukan setelah perencanaan sudah dianggap sempurna.

3.2.6 Hasil

Hasil adalah suatu yang dapat dikatakan bawah penelitian ini sudah memenuhi syarat untuk menyelesaikan penelitian ini.

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Analisis

4.1.1 Analisis Sistem

Sistem ini akan melakukan pencarian rute terpendek antar kos san penulis yang terletak di Baleendah ke Perpustakaan Kawaluyaan Bandung dengan menggunakan algoritma Dijkstra.

Analisis sistem memiliki tiga tahapan dalam mendeskripsikan pengembangan sistem, yaitu:

a. Analisis masalah

Analisis masalah digunakan untuk mempelajari, mengidentifikasi, dan menganalisis suatu masalah serta batasan-batasan masalahnya.

b. Analisis kebutuhan

Analisis kebutuhan digunakan untuk mendeskripsikan fungsi-fungsi yang diberikan dan dapat dikerjakan oleh sistem, baik kebutuhan fungsional maupun non-fungsional.

c. Analisis proses

Analisis proses digunakan untuk menggambarkan pola tingkah laku yang dimiliki sistem.

4.1.2 Analisis Masalah

Luasnya sebuah wilayah dapat menyebabkan seseorang mengalami kesulitan untuk menentukan rute terpendek dari satu lokasi ke lokasi lainnya. Setiap orang yang ingin menuju ke lokasi Perpustakaan Kawaluyaan, cenderung memilih rute terpendek. Posisi awal pengunjung ditentukan oleh lokasi Perpustakaan Kawaluyaan yang berbeda. Permasalahan pada penelitian ini adalah pencarian rute terpendek menggunakan algoritma Dijkstra terhadap graf yang untuk menentukan jarak terpendek antara lokasi Perpustakaan Kawaluyaan Bandung.

4.1.3 Analisis Proses

Pada analisis proses ini menggunakan algoritma Dijkstra dalam proses pencarian rute terpendek antara kos dan lokasi penulis ke Perpustakaan Kawalayaan Bandung. Pengguna dapat melakukan pencarian dengan memasukkan lokasi Perpustakaan Kawalayaan. Lalu, sistem akan menampilkan hasil berupa total jarak, rute, dan waktu yang dibutuhkan untuk mengeksekusi program berdasarkan algoritma yang digunakan.

4.1.4 Pengumpulan Data

1. Langkah – Langkah Pengumpulan Data

a. Google Maps

Google Maps adalah sebuah aplikasi yang dimana aplikasi ini dapat menampilkan maps di seluruh dunia. Maka dari itu peneliti menggunakan google maps untuk mendapatkan data yang diinginkan.

b. Lokasi (*Asal*)

Lokasi asal adalah lokasi peneliti dimana lokasi peneliti dijadikan sebagai lokasi titik awal yang akan dimulainya perjalanan ke Perpustakaan Kawalayaan tersebut.

c. Lokasi (*Tujuan*)

Lokasi tujuan adalah lokasi yang akan peneliti tuju yang dimana lokasi tersebut terletak di jalan kawalayaan lebih tepatnya di jln.kawalayaan indah raya yaitu lokasi Perpustakaan Kawalayaan berada.

d. Rute yang akan dilalui

Adapun jalur rute yang akan di ambil datanya yaitu dari google maps.

2. Google Maps

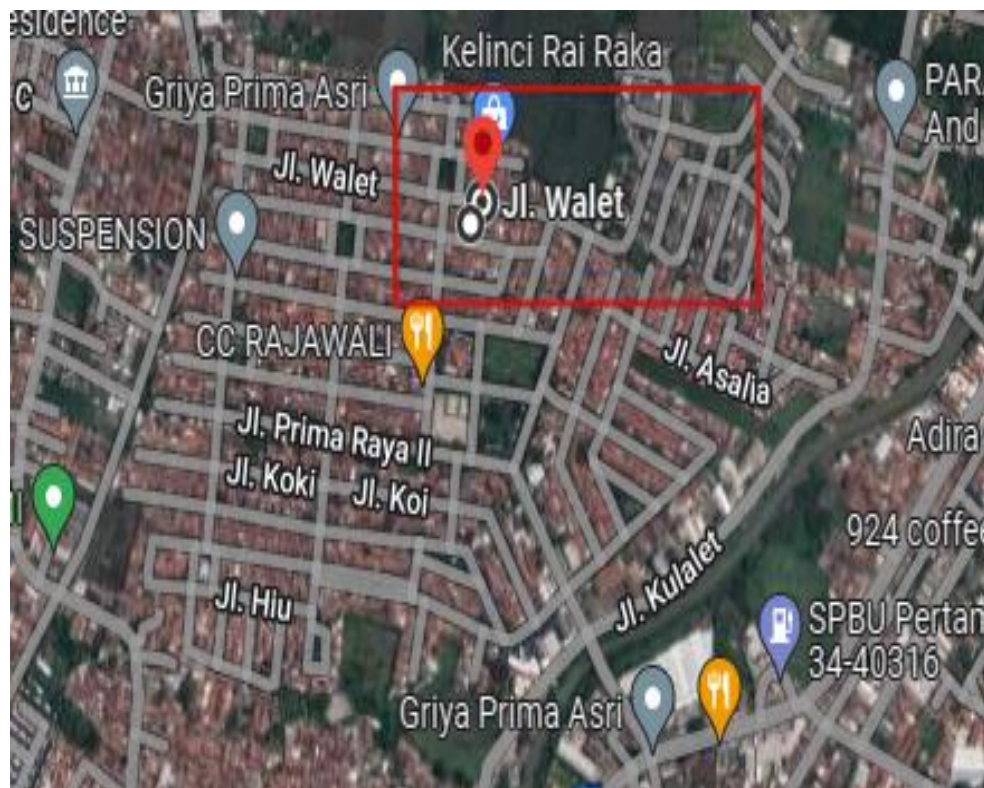
Google Maps adalah layanan pemetaan web yang dikembangkan oleh Google. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan rute untuk bepergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum.



Gambar 4. 1 Google Maps

3. Lokasi Asal

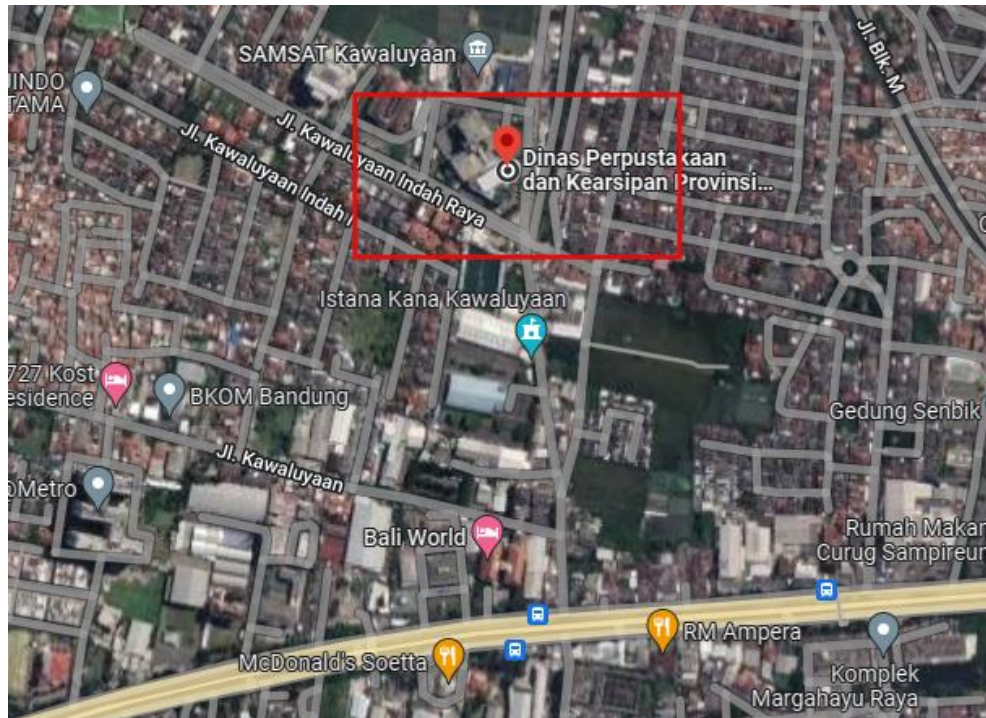
Lokasi asal peneliti berada di Baleendah lebih tepatnya adapun lokasi sebagai berikut:



Gambar 4. 2 Lokasi asal peneliti

4. Lokasi Tujuan

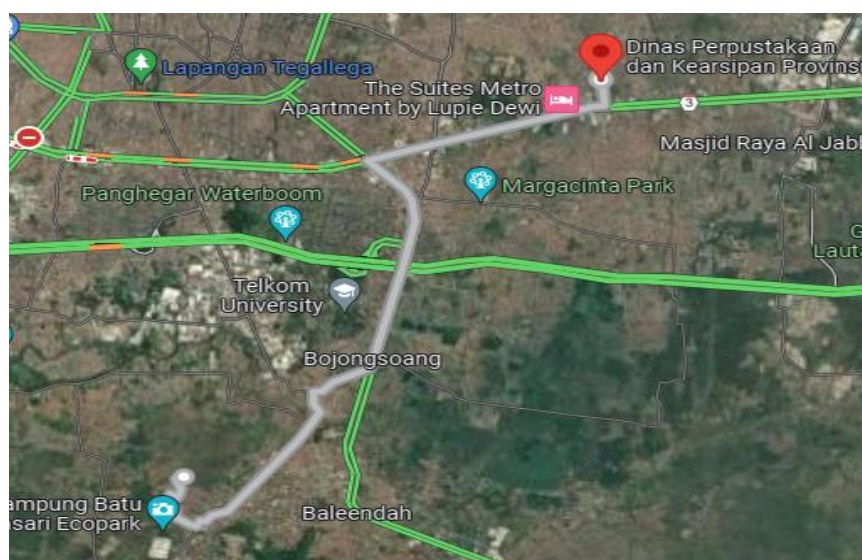
Lokasi tujuan peneliti berada di dinas perpustakaan (jalan Kawalayaan Raya) gambar lokasinya sebagai berikut:



Gambar 4. 3 Lokasi Tujuan Peneliti

5. Rute

Berdasarkan data jalur yang di tampilkan dari google maps yaitu sebagai berikut:



Gambar 4. 4 Jalur yang dilalui berdasarkan google maps

4.1.5 Analisis Data

Analisis data dilakukan dengan merepresentasikan rute yang menghubungkan jalur-jalur yang menghubungkan rute yang akan dilalui dari lokasi awal ke lokasi tujuan.

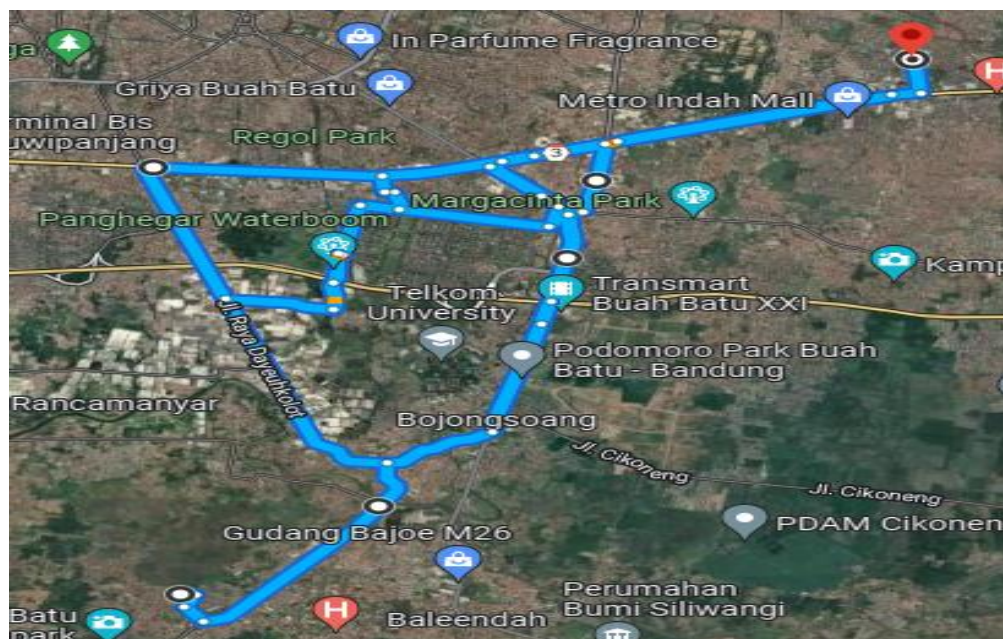
Berikut langkah-langkah untuk merepresentasikan rute yang menghubungkan lokasi awal ke lokasi tujuan dalam bentuk graf berarah dan berbobot:

1. Mengambil peta yang menunjukkan jalur-jalur yang mengarahkan ke Perpustakaan Kawalayaan.
2. Menentukan cakupan lokasi yang memuat (titik asal) Baleendah dan (titik tujuan) Perpustakaan Kawalayaan Bandung.
3. Mengumpulkan data koordinat dan alamat atau nama lokasi-lokasi jalur yang mengarahkan lokasi awal ke lokasi tujuan dan sertakan panjang koordinat (dalam satuan meter atau kilometer).
4. Membuat graf berarah dan berbobot dengan menghubungkan titik-titik menggunakan garis yang ada.

4.2 Perancangan

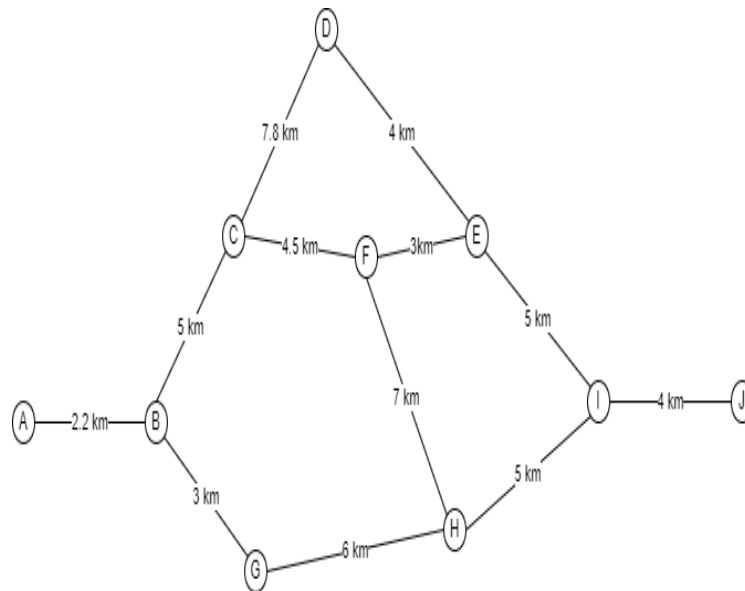
4.2.1 Titik Rute Yang Akan Dilalui

Adapun titik rute atau jalur yang akan di lalui adalah seperti gambar berikut:



Gambar 4.6. 1 Peta Titik Yang Akan Dilalui

Berdasarkan gambar diatas, maka dapat dibuat graf berarah dimana masing-masing titik koordinat akan terhubung satu dengan yang lainnya. Dan adapun jarak antara masing-masing koordinat didapatkan dari google maps.



Gambar 4.6. 2 Graf Berdasarkan Google Maps

Adapun keterangan dari setiap titik akan dijelaskan dalam tabel dibawah:

Tabel 4.6. 1 Nama setiap koordinat

Titik	Nama – Nama Setiap Koordinat
A	Baleendah (asal)
B	Dayeuh kolot
G	Bojongsoang
C	Bandung Kidul
D	Moh.toha
F	Batununggal indah II
E	Suka ati
H	Batu Nunggal Indah IV
I	Soekarno Hatta
J	Kawalayaan Indah I (tujuan)

4.2.2 Mencari Jalur Atau Rute Dengan Algoritma Dijkstra

Bagian ini akan menerapkan Algoritma Dijkstra untuk menentukan rute terpendek dari Baleendah ke Perpustakaan Kawalayaan Bandung. Pencarian rute terpendek akan dilakukan berdasarkan jalur koordinat-koordinat yang akan dilalui. Maka dilakukan perhitungan dengan algoritma dijkstra secara manual terlebih dahulu, dan cara kerja algoritma dijkstra sebagai berikut:

Tabel 4.7. 1 Penerapan Algoritma Dijkstra

V	A	B	C	D	E	F	G	H	I	J
A	0 _A	4.7 _A	∞	∞	∞	∞	∞	∞	∞	∞
B		4.7 _A	12.7 _B				9.7 _B	∞	∞	∞
G		4.7 _A					9.7 _G	24.7 _G	∞	∞
C		4.7 _A		8.7 _C		14. _C	∞	∞	∞	∞
D		4.7 _A		8.7 _D	9.7 _D	∞	∞	∞	∞	∞
E		4.7 _A				7.7 _E			12.7 _E	∞
F		4.7 _A				7.7 _F		19.7 _F	∞	∞
H		4.7 _A						19.7 _H	8.7 _H	∞
I		4.7 _A								24.7 _I
J		4.7 _A								24.7 _J

Berdasarkan hasil perhitungan algoritma Dijkstra dengan menggunakan tabel diatas maka didapatkan hasil yaitu: (A, B, G, H, I, J)

Deskripsi

Node A = Baleendah (Kota Asal)

Node B = Dayeuh Kolot

Node G = Bojongsoang

Node H = Batununggal Indah IV

Node I = Soekarno Hatta

Node J = Kawalayaan

Berdasarkan tabel diatas kita dapat membua sebuah tabel lintasan yang dimana tabel ini akan menerangkan jalaur mana saja yang akan kita lalui Adapun tabelnya sebagai berikut:

Tabel 4.7. 2 Panjang Lintasan Yang Akan Ditempuh

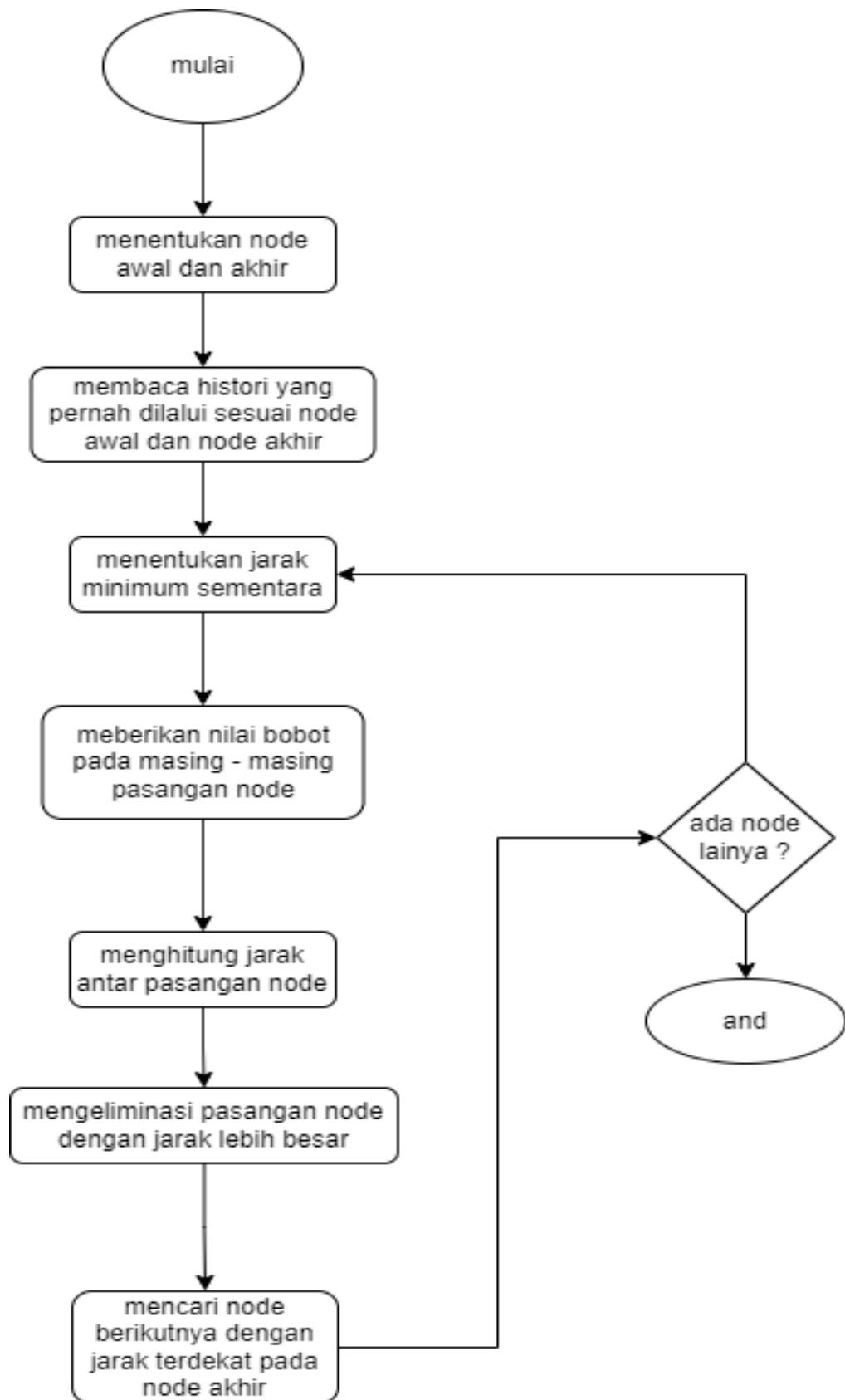
Kota Asal	Tujuan	Rute Yang Dilalui	Panjang Lintasan (km)
Baleendah Kosan	Perpustakaan Kawalayaan Bandung	Baleendah → Dayehkolot → Bojongsoang → Batununggal Indah IV → Soekarno Hatta → Kawalayaan	20.2 km
Total Panjang Lintasan			20.2 km

4.2.3 Alur Algoritma Dijkstra Dalam Mencari Lintasan Terpendek

Dari analisis yang sudah dikerjakan secara manual sebelumnya masih memerlukan waktu yang lama dan kurang efesien untuk melakukan pencarian ruter terpendek yang dimana rute tersebut berlokasi kosan Baleendah ke lokasi Perpustakaan Kawalayaan dan untuk menanggulangi hal tersebut peneliti ingin membuat sebuah alur program, dimana alur ini nantinya akan digunakan untuk merancang sebuah program yang betujuan untuk mencari rute terdekat.

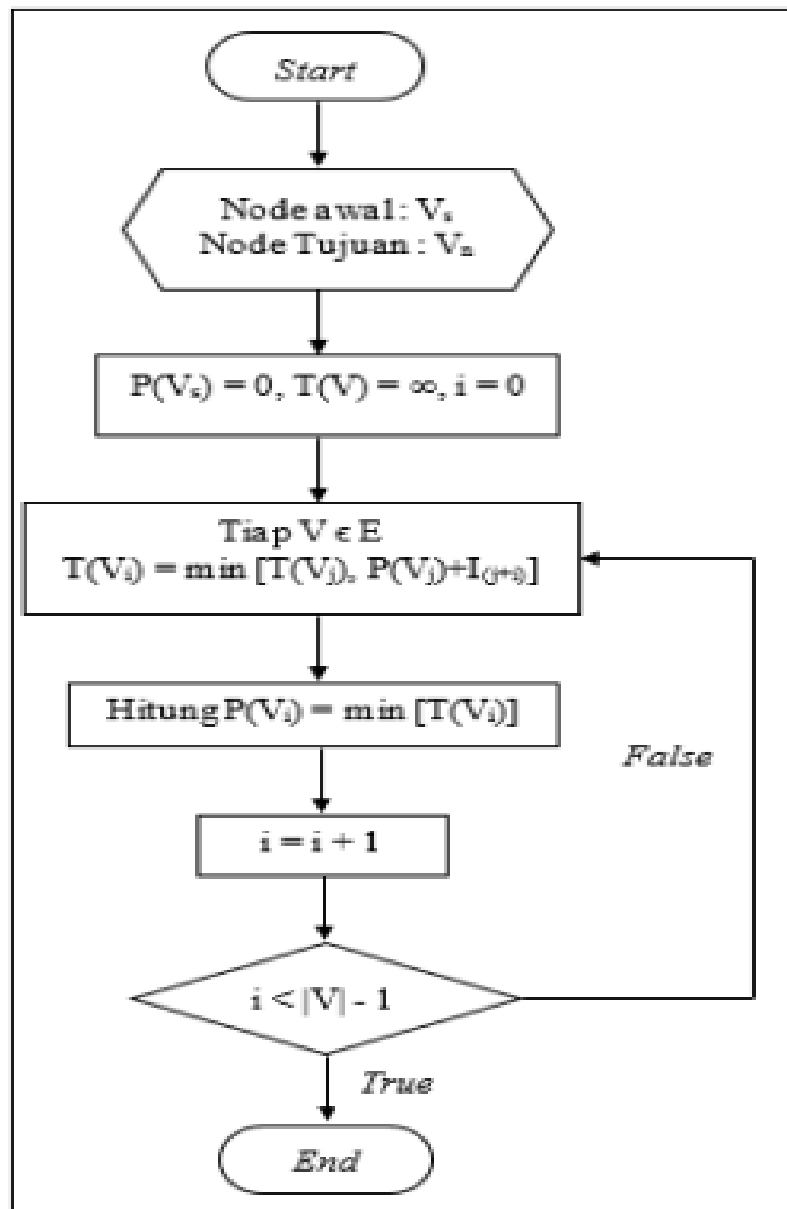
a. Alur perancangan program

Alur pernacangan ini digunakan untuk merancang aplikasi yang nantinya akan ditampilkan di desktop.



Gambar 4.8. 1 Alur Program Algoritma Dijkstra

b. Alur Program Algoritma Dijkstra



Gambar 4.8. 2 Gambar Alur Dijkstra

4.2.4 Mencari Rute Terpendek Menggunakan Bahasa Pemrograman Python

Proses pencarian rute terpendek dengan algoritma Dijkstra tidak hanya dapat dilakukan dengan menghitung secara manual melainkan juga bisa dicari menggunakan bahasa pemrograman Python. Pemanfaatan bahasa pemrograman Python dapat mempersingkat waktu dalam mencari rute terpendek. Pada Python yang dibuat untuk menentukan rute terpendek juga memanfaatkan GUI (Graphical User Interface) supaya program dapat berjalan lebih interaktif serta dapat

memudahkan pengguna dalam menggunakannya. Sebelum ke programnya berikut adalah fitur-fitur yang digunakan dalam aplikasi:

4.2.5 Fitur-Fitur Yang Ada Pada Aplikasi

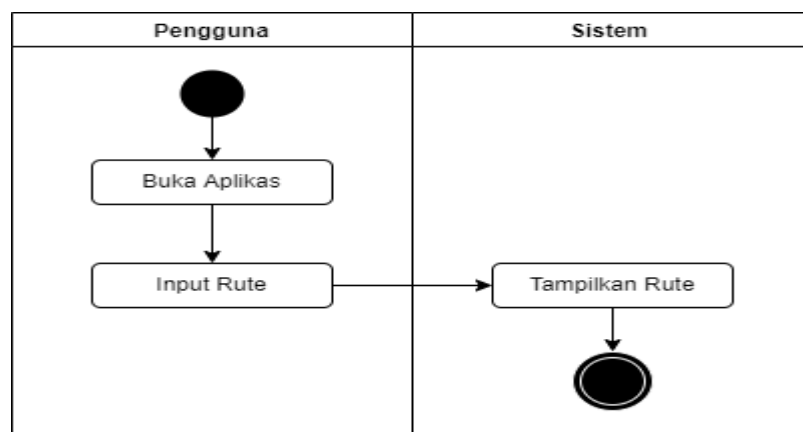
Fitur yang dapat digunakan dalam aplikasi ini yang akan digunakan adalah sebagai berikut:

- a. Input data
- b. Boton Mulai
- c. Hapus

Tabel 4.9. 1 Fitur - Fitur Aplikasi

Fitur – Fitur	Keterangan
Input	Fitur input adalah fitur yang digunakan untuk memasukan data yang kita inginkan
Boton Mulai	Fitur ini digunakan untuk memulai pencarian data.
Boton Hapus	Fitur ini digunakan untuk menghapus data pencarian.

4.2.6 Activity Diagram



Gambar 4.9. 1 Activity diagram pengguna

4.2.7 Desain Aplikasi

Pada tahap perancangan tampilan antarmuka aplikasi penjadwalan praktikum dibuat untuk memudahkan dalam tahap pengembangan yang dilakukan dengan memanfaatkan tools atau software pada aplikasi Balsamiq Mockups. Perancangan

desain tampilan antarmuka ditujukan agar tampilan website yang akan dibuat sudah terancang dengan baik. Ketika akan membuat tampilan aplikasi cukup mengikuti desain yang sudah dibuat dengan aplikasi Balsamiq mockups. Berikut adalah desain antarmuka untuk aplikasi penjadwalan praktikum:

The diagram shows a rectangular window layout. At the top is a wide box labeled 'Title'. Below it are two rows of input fields: the first row is labeled 'Masukan Titik Awal :' and the second row is labeled 'Masukan Titik Tujuan :'. Under the input fields are two buttons, 'Mulai' and 'Hapus', positioned side-by-side. Below the buttons is a wide box labeled 'Hasil'. At the bottom of the window is a section labeled 'Keterangan' which contains three empty rectangular boxes arranged horizontally.

Gambar 4.9. 2 Tampilan Program

4.2.8 Program Pencarian Rute Menggunakan Python

Pada langkah awal, menuliskan module yang digunakan dalam penyusunan program. Module yang digunakan pada program ini yaitu tkinter yang berguna untuk membangun GUI pada program yang dibuat. Tkinter.font digunakan karena dalam membuat program terdapat kalimat yang perlu ditebalkan.

```
from tkinter import *
from tkinter.font import BOLD
```

Variabel yang digunakan untuk mengatur jendela GUI adalah root. root.title berfungsi untuk menuliskan judul pada jendela GUI. Supaya ukuran yang telah ditetapkan untuk jendela GUI tidak berubah maka diberi perintah root.resizable (width=False,height=False). Ukuran jendela GUI diatur dengan variabel root.geometry sebesar 1000 × 600. Supaya tampilan jendela GUI menarik, diberikan warna sebagai latar pada jendela GUI dengan variabel root.configure dan warna yang dipilih adalah skyblue.

```

root=Tk()
root.title("Menentukan Rute Terpendek Antara Baleendah Ke
Perpustakaan Kawalayaan")
root.resizable(width=False,height=False)
root.geometry("1000x600")
root.configure(background="blue")

```

Tampilan awal yang dibuat untuk program ini berisi judul program. Kemudian di bawah judul terdapat tulisan perintah untuk memasukkan titik awal dan akhir serta tempat untuk menuliskan titik awal dan akhir yang ditentukan oleh pengguna. Pada tampilan awal juga terdapat daftar titik awal dan titik akhir. Daftar titik ini dapat membantu pengguna dalam memasukkan titik pada kotak yang tersedia sebelumnya. Bagian ini hanya perlu mengatur posisi tulisan dan kotak yang digunakan untuk memasukkan data. Selain itu jenis huruf, ukuran huruf, dan warna latar tulisan juga perlu diatur pada bagian ini.

```

judul =Label(root,bg='silver',text='Mencari Rute Dari
Baleendah Ke Perpustakaan Kawalayaan',font=('Times New
Roman',20,BOLD),fg='black')
judul.place(relx=0.5,relly=0.03,relwidth=0.8,relheight=0.1,
anchor='n')

label1 =Label(root,bg='skyblue',text='Masukkan titik
awal:',font=('Times New Roman',15),fg='black')
label1.place(relx=0.35,relly=0.15,relwidth=0.20,relheight=0
.06,anchor='n')

Input1 = Entry(root)
Input1.place(relx=0.6,relly=0.15,relwidth=0.1,
relheight=0.05,anchor='n')

label2 =Label(root,bg='skyblue',text='Masukkan titik
akhir:',font=('Times New Roman',15),fg='black')
label2.place(relx=0.35,relly=0.22,relwidth=0.20,relheight=0
.06,anchor='n')

Input2 = Entry(root)
Input2.place(relx=0.6,relly=0.22,relwidth=0.1,relheight=0.0
5,anchor='n')

label5=Label(root,bg='silver',text='Daftar Titik Awal Dan
Titik Akhir Yang Dapat Diisikan',font=('Times New
Roman',15,BOLD),fg='black')
label5.place(relx=0.25,relly=0.63,relwidth=0.5,relheight=0.
05,anchor='n')

label9=Label(root,bg='white',text='Baleendah =
A',font=('Times New Roman',12),fg='black')

```

```

label9.place(relx=0.15, rely=0.75, relwidth=0.3, relheight=0.05, anchor='n')

label10=Label(root, bg='white', text='Dayehkolot B', font=('Times New Roman', 12), fg='black')
label10.place(relx=0.15, rely=0.8, relwidth=0.3, relheight=0.05, anchor='n')

label11=Label(root, bg='white', text='Bojongsoang G', font=('Times New Roman', 12), fg='black')
label11.place(relx=0.15, rely=0.85, relwidth=0.3, relheight=0.05, anchor='n')

label12=Label(root, bg='white', text='Bandung Kidul C', font=('Times New Roman', 12), fg='black')
label12.place(relx=0.15, rely=0.9, relwidth=0.3, relheight=0.05, anchor='n')

label13=Label(root, bg='white', text='Moh. toha D', font=('Times New Roman', 12), fg='black')
label13.place(relx=0.5, rely=0.75, relwidth=0.3, relheight=0.05, anchor='n')

label14=Label(root, bg='white', text='Suka Ati E', font=('Times New Roman', 12), fg='black')
label14.place(relx=0.5, rely=0.8, relwidth=0.3, relheight=0.05, anchor='n')

label15=Label(root, bg='white', text='Batu Nunggal Indah II I', font=('Times New Roman', 12), fg='black')
label15.place(relx=0.5, rely=0.85, relwidth=0.3, relheight=0.05, anchor='n')

label16=Label(root, bg='white', text='Suka Ati E', font=('Times New Roman', 12), fg='black')
label16.place(relx=0.85, rely=0.75, relwidth=0.3, relheight=0.05, anchor='n')

label8=Label(root, bg='white', text='Batu Nunggal Indah IV H', font=('Times New Roman', 12), fg='black')
label8.place(relx=0.85, rely=0.8, relwidth=0.3, relheight=0.05, anchor='n')

label8=Label(root, bg='white', text='Soekarno Hatta I', font=('Times New Roman', 12), fg='black')
label8.place(relx=0.85, rely=0.8, relwidth=0.3, relheight=0.05, anchor='n')

label8=Label(root, bg='white', text='Kawalayaan Indah I J', font=('Times New Roman', 12), fg='black')
label8.place(relx=0.85, rely=0.8, relwidth=0.3, relheight=0.05, anchor='n')

```


Pada bagian selanjutnya memuat kode yang berguna dalam menentukan alur perjalanan dan jarak rute terpendek. Kode tersebut terdiri atas 1 bagian yang diatur pada *def mulai()*. Secara garis besar ketika proses perhitungan tersebut sama, perbedaannya terdapat pada variabel lokasi yang berisikan jarak antar titik. Titik awal dan titik akhir akan dimasukkan oleh pengguna saat program berjalan, maka supaya program dapat mengerjakan rute terpendek dari titik awal menuju titik akhir tersebut diperlukan variable *Input1.get()* dan *Input2.get()*. Variabel tersebut berfungsi untuk menyimpan data yang dimasukkan oleh pengguna. Besar jarak yang ditempuh dari titik awal menuju titik akhir dihasilkan pada variabel *jarak[berhenti]*. Hasil tersebut kemudian dicetak dalam bentuk sebuah kalimat menggunakan variabel *label3*. Sedangkan untuk rute wajib dilalui termuat pada variabel *alur[:-1]*. Hasil tersebut juga dicetak dalam bentuk kalimat menggunakan variabel *label4*. Program diharapkan dapat melakukan perhitungan beberapa kali tanpa harus menutup jendela *GUI*. Sehingga data yang sudah ditampilkan perlu dihapus supaya tidak mengganggu tampilan selanjutnya. Proses penghapusan data sebelumnya diatur pada variabel *def reset()* dan akan terhapus ketika menekan tombol Hapus.

```
def mulai():
    infinity = float("infinity")
    mulai = str(Input1.get())
    berhenti = str(Input2.get())
    lokasi = {
        'A':{
            'B':2.2
        },
        'B':{
            'G':3,
            'C':5
        },
        'G':{
            'H':6
        },
        'C':{
            'D':7.8,
            'F':4.5
        },
        'D':{
            'E':4
        },
    },
```

```

        'E':{
            'F':3,
            'I':5
        },
        'F':{
            'E':3,
            'H':7
        },
        'E':{
            'I':5
        },
        'H':{
            'I':5
        },
        'I':{
            'J':4
        },
        'J':{}
    }
    jarak = {}
    titik = {}
    for node in lokasi:
        jarak[node] = infinity
    titik[node] = {}
    jarak[mulai] = 0

    def rute_terpendek(jarak, not_checked):
        lowest_dist = infinity

        cheapest_node = ""
        for node in jarak:
            if node in not_checked and jarak[node]
<=lowest_dist:
                lowest_dist = jarak[node]
                cheapest_node = node
        return cheapest_node

    not_checked = [node for node in jarak]
    node = rute_terpendek(jarak, not_checked)

    while not_checked:
        dist = jarak[node]
        child_dist = lokasi[node]
        for c in child_dist:
            if jarak[c] > dist + child_dist[c]:
                jarak[c] = dist + child_dist[c]
                titik[c] = node

        not_checked.pop(not_checked.index(node))
        node = rute_terpendek(jarak, not_checked)

```

```

        label3=Label(root,text="Jarak yang ditempuh sejauh "+str(jarak[berhenti])+" km",font=('Times New Roman',15),bg='peachpuff')

label3.place(relx=0.5,rely=0.5,relwidth=1,relheight=0.05,anchor='n')

        if jarak[berhenti] < infinity:
            alur=[berhenti]
            i=0
            while mulai not in alur:
                alur.append(titik[alur[i]])
                i+=1

            label4=Label(root,text="Alurnya adalah " + str(alur[:-1]),font=('Times New Roman',15),bg='peachpuff')

label4.place(relx=0.5,rely=0.55,relwidth=1,relheight=0.05,anchor='n')
        else:
            label4=Label(root,text="Alurnya tidak ditemukan " + str(alur[:-1]),font=('Times New Roman',15),bg='peachpuff')
            label4.place(relx=0.5,rely=0.55,relwidth=1,relheight=0.05,anchor='n')

def reset():
    label3.destroy()
    label4.destroy()
    Input1.delete(0,END)
    Input2.delete(0,END)
    Hapus_Button = Button(root, text="Hapus", font=('Times New Roman', 15), command=reset)
    Hapus_Button.place(relx=0.6, rely=0.4, relwidth=0.2, relheight=.07)

```

Program diharapkan dapat mencari rute terpendek dari Baleendah ke Kawaluyaan dengan rute dan jarak yang berbeda. Maka dari itu terdapat tombol Mulai yang akan menampilkan besar jarak serta rute yang dilalui. Tombol tersebut diatur dengan variabel Enter_button, Hapus_botton. tombol tersebut diberi perintah untuk menampilkan hasil dengan perintah berupa lokasi yang telah didefinisikan sebelumnya.

```

enter_button=Button(root,text='Mulai',font=('Times New Roman',15),command=mulai)

```

```
enter_button.place(relx=0.4, rely=0.4, relwidth=0.2, relheight=0.07, anchor='n')
```

Pada bagian akhir terdapat variabel `root.mainloop()`. Variabel tersebut berfungsi untuk menampilkan seluruh program yang telah disusun pada jendela GUI.

```
root.mainloop()
```

BAB V

HASIL DAN IMPLEMENTASI

5.1 Hasil

5.1.1 Listening Program

Listening program yaitu menampilkan coding yang digunakan dalam mencari rute terdekat, terutama coding solusi untuk mengatasi masalah yang diteliti. Berikut program yang digunakan dalam mencari rute terdekat:

```
from tkinter import *
from tkinter.font import BOLD

root=Tk()
root.title("Menentukan Rute Terpendek Antara Baleendah Ke
Perpustakaan Kawalayaan")
root.resizable(width=False,height=False)
root.geometry("1000x600")
root.configure(background="blue")

judul =Label(root,bg='silver',text='Mencari Rute Dari
Baleendah Ke Perpustakaan Kawalayaan',font=('Times New
Roman',20,BOLD),fg='black')
judul.place(relx=0.5,rely=0.03,relwidth=0.8,relheight=0.1,
anchor='n')

label1 =Label(root,bg='skyblue',text='Masukkan titik
awal:',font=('Times New Roman',15),fg='black')
label1.place(relx=0.35,rely=0.15,relwidth=0.20,relheight=0
.06,anchor='n')

Input1 = Entry(root)
Input1.place(relx=0.6,rely=0.15,relwidth=0.1,
relheight=0.05,anchor='n')
```

```

label2 =Label(root,bg='skyblue',text='Masukkan titik
akhir:',font=('Times New Roman',15),fg='black')
label2.place(relx=0.35,relx=0.22,relwidth=0.20,relheight=0
.06,anchor='n')

Input2 = Entry(root)
Input2.place(relx=0.6,relx=0.22,relwidth=0.1,relheight=0.0
5,anchor='n')

label5=Label(root,bg='silver',text='Daftar Titik Awal Dan
Titik Akhir Yang Dapat Diisikan',font=('Times New
Roman',15,BOLD),fg='black')
label5.place(relx=0.25,relx=0.63,relwidth=0.5,relheight=0.
05,anchor='n')

label9=Label(root,bg='white',text='Baleendah =
A',font=('Times New Roman',12),fg='black')
label9.place(relx=0.15,relx=0.75,relwidth=0.3,relheight=0.
05,anchor='n')

label10=Label(root,bg='white',text='Dayehkolot =
B',font=('Times New Roman',12),fg='black')
label10.place(relx=0.15,relx=0.8,relwidth=0.3,relheight=0.
05,anchor='n')

label11=Label(root,bg='white',text='Bojongsoang =
G',font=('Times New Roman',12),fg='black')
label11.place(relx=0.15,relx=0.85,relwidth=0.3,relheight=0
.05,anchor='n')

label12=Label(root,bg='white',text='Bandung Kidul =
C',font=('Times New Roman',12),fg='black')
label12.place(relx=0.15,relx=0.9,relwidth=0.3,relheight=0.
05,anchor='n')

```

```

label13=Label(root,bg='white',text='Moh.toha =
D',font=('Times New Roman',12),fg='black')
label13.place(relx=0.5,relx=0.75,relwidth=0.3,relheight=0.
05,anchor='n')

label14=Label(root,bg='white',text='Suka Ati =
E',font=('Times New Roman',12),fg='black')
label14.place(relx=0.5,relx=0.8,relwidth=0.3,relheight=0.0
5,anchor='n')

label15=Label(root,bg='white',text='Batu Nunggal Indah II
= I',font=('Times New Roman',12),fg='black')
label15.place(relx=0.5,relx=0.85,relwidth=0.3,relheight=0.
05,anchor='n')

label16=Label(root,bg='white',text='Suka Ati =
E',font=('Times New Roman',12),fg='black')
label16.place(relx=0.85,relx=0.75,relwidth=0.3,relheight=0
.05,anchor='n')

label8=Label(root,bg='white',text='Batu Nunggal Indah IV =
H',font=('Times New Roman',12),fg='black')
label8.place(relx=0.85,relx=0.8,relwidth=0.3,relheight=0.0
5,anchor='n')

label8=Label(root,bg='white',text='Soekarno Hatta =
I',font=('Times New Roman',12),fg='black')
label8.place(relx=0.85,relx=0.8,relwidth=0.3,relheight=0.0
5,anchor='n')

label8=Label(root,bg='white',text='Kawalayaan Indah I =
J',font=('Times New Roman',12),fg='black')
label8.place(relx=0.85,relx=0.8,relwidth=0.3,relheight=0.0
5,anchor='n')

```

```
def mulai():  
    infinity = float("infinity")  
    mulai = str(Input1.get())  
    berhenti = str(Input2.get())  
    lokasi = {  
        'A':{  
            'B':2.2  
        },  
        'B':{  
            'G':3,  
            'C':5  
        },  
        'G':{  
            'H':6  
        },  
        'C':{  
            'D':7.8,  
            'F':4.5  
        },  
        'D':{  
            'E':4  
        },  
        'E':{  
            'F':3,  
            'I':5  
        },  
        'F':{  
            'E':3,  
            'H':7  
        },  
        'E':{  
            'I':5
```



```

        },
        'H':{
            'I':5
        },
        'I':{
            'J':4
        },
        'J':{}
    }
    jarak = {}
    titik = {}
    for node in lokasi:
        jarak[node] = infinity
    titik[node] = {}
    jarak[mulai] = 0

    def rute_terpendek(jarak, not_checked):
        lowest_dist = infinity

        cheapest_node = ""
        for node in jarak:
            if node in not_checked and jarak[node]
<=lowest_dist:
                lowest_dist = jarak[node]
                cheapest_node = node
        return cheapest_node

    not_checked = [node for node in jarak]
    node = rute_terpendek(jarak, not_checked)

    while not_checked:
        dist = jarak[node]
        child_dist = lokasi[node]
        for c in child_dist:
            if jarak[c] > dist + child_dist[c]:

```

```

        jarak[c] = dist + child_dist[c]
        titik[c] = node

    not_checked.pop(not_checked.index(node))
    node = rute_terpendek(jarak, not_checked)

    label3=Label(root,text="Jarak yang ditempuh sejauh
"+str(jarak[berhenti])+" km",font=('Times New
Roman',15),bg='peachpuff')

label3.place(relx=0.5,rely=0.5,relwidth=1,relheight=0.05,a
nchor='n')

    if jarak[berhenti] < infinity:
        alur=[berhenti]
        i=0
        while mulai not in alur:
            alur.append(titik[alur[i]])
            i+=1

        label4=Label(root,text="Alurnya adalah " +
str(alur[::-1]),font=('Times New Roman',15),
bg='peachpuff')

label4.place(relx=0.5,rely=0.55,relwidth=1,relheight=0.05,
anchor='n')

    else:
        label4=Label(root,text="Alurnya tidak ditemukan "
+ str(alur[::-1]),font=('Times New
Roman',15),bg='peachpuff')

        label4.place(relx=0.5,rely=0.55,relwidth=
1,relheight=0.05,anchor='n')

def reset():
    label3.destroy()

```

```

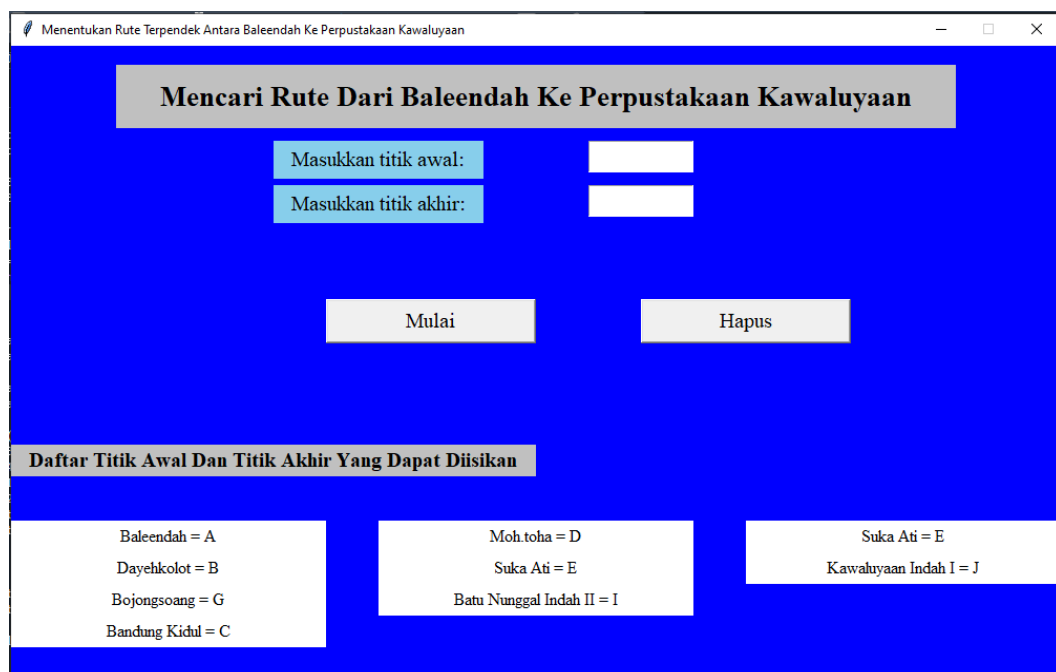
label4.destroy()
Input1.delete(0,END)
Input2.delete(0,END)
Hapus_Button = Button(root, text="Hapus", font=('Times New
Roman', 15), command=reset)
Hapus_Button.place(relx=0.6, rely=0.4, relwidth=0.2,
relheight=.07)

enter_button=Button(root,text='Mulai',font=('Times New
Roman',15),command=mulai)
enter_button.place(relx=0.4, rely=0.4, relwidth=0.2, relheight=
0.07, anchor='n')

root.mainloop()

```

Setelah program di run akan menampilkan jendela GUI beserta komponen yang telah dibuat sebelumnya. Berikut tampilan dari program GUI yang telah disusun dan contoh penggunaannya.



Gambar 5.1. 1 Tampilan awal

merupakan tampilan awal setelah program di run. Pada tampilan awal tersebut berisi judul. Setelah judul terdapat kotak untuk memasukkan data berupa

titik awal dan titik akhir. Terdapat juga 2 tombol yaitu mulai, dan tombol hapus yang dapat menampilkan hasil berupa besar jarak serta rute yang dilalui. Daftar titik awal dan titik akhir juga ditampilkan supaya membantu pengguna dalam proses mengisi titik awal dan titik akhir dan adapun tombol hapus yang berfungsi untuk menghapus inputan.

Gambar 5.1. 2 Jarak dari Baleendah ke Perpustakaan Kawalayaan

Dari hasil pencarian diatas maka hasil rute atau jalur yang didapatkan yang didapatkan adalah {A(Baleendah), B(Dayeuh kolot), G(Bojong Soang), H(Buah Batu), I(Soekarno Hatta)}.

Adapun pencarian jalur dari setiap node yang diinginkan adalah sebagai berikut:

Gambar 5.1. 3 Node A Ke B

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 5 km
Alurnya adalah ['B', 'C']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 4 Node B ke C

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 7.8 km
Alurnya adalah ['C', 'D']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 5 Node C ke D

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 4.5 km
Alurnya adalah ['C', 'F']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 6 Node C ke F

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 4 km
Alurnya adalah ['D', 'E']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 7 Node D ke E

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 3 km
Alurnya adalah ['F', 'E']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawahayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 8 Node F ke E

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 7 km
Alurnya adalah ['F', 'H']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawahayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 9 Node F ke H

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 5 km
Alurnya adalah ['H', 'I']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 10 Node H ke I

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 5 km
Alurnya adalah ['E', 'I']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 11 Node E ke I

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 4 km
Alurnya adalah ['I', 'J']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 12 Node I ke J

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 6 km
Alurnya adalah ['G', 'H']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 13 Node G ke H

Mencari node dengan melewati 1 node dari titik semula ke titik node tujuan adapun jarak tempuhnya sebagai berikut:

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 7.2 km
Alurnya adalah ['A', 'B', 'C']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawahayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 14 Node A, B ke C

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 15.0 km
Alurnya adalah ['A', 'B', 'C', 'D']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawahayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 15 Node A, B, C dan D

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 11.7 km
Alurnya adalah ['A', 'B', 'C', 'F']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 16 Node A, B, C dan F

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 14.7 km
Alurnya adalah ['A', 'B', 'C', 'F', 'E']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 17 Node A, B, C, F dan E

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawaluyaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawaluyaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 5.2 km
Alurnya adalah ['A', 'B', 'G']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawaluyaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 18 Node A, B dan G

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawaluyaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawaluyaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 11.2 km
Alurnya adalah ['A', 'B', 'G', 'H']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawaluyaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 19 Node A, B, G dan H

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 16.2 km
Alurnya adalah ['A', 'B', 'G', 'H', 'I']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 20 Node A, B, G, H dan I

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 18 km
Alurnya adalah ['B', 'G', 'H', 'I', 'J']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 21 Node B, G, H, I dan J

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 12.8 km
Alurnya adalah ['B', 'C', 'D']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 22 Node B, C dan D

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 12.5 km
Alurnya adalah ['B', 'C', 'F', 'E']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 23 Node B,C,F dan E

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 16.5 km
Alurnya adalah ['C', 'F', 'E', 'I', 'J']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 24 Node C, F, E, I dan J

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 13 km
Alurnya adalah ['D', 'E', 'I', 'J']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 25 Node D,E,I dan J

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 9 km
Alurnya adalah ['E', 'T', 'J']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 26 Node E, I dan J

Menentukan Rute Terpendek Antara Baleendah Ke Perpustakaan Kawalayaan

Mencari Rute Dari Baleendah Ke Perpustakaan Kawalayaan

Masukkan titik awal:

Masukkan titik akhir:

Mulai Hapus

Jarak yang ditempuh sejauh 12 km
Alurnya adalah ['F', 'E', 'T', 'J']

Daftar Titik Awal Dan Titik Akhir Yang Dapat Diisikan

Baleendah = A	Moh.toha = D	Suka Ati = E
Dayehkolot = B	Suka Ati = E	Kawalayaan Indah I = J
Bojongsoang = G	Batu Nunggal Indah II = I	
Bandung Kidul = C		

Gambar 5.1. 27 Node F,E,I dan J

5.1.2 Implementasi Sistem

Implementasi system merupakan tahap penerapan system yang akan dilakukan jika system telah disetujui termasuk program yang sesuai berdasarkan tahap pengkodean yang sudah dibuat pada tahap sebelumnya. Adapun waktu dan tempat untuk implentasi program yaitu:

Tempat : Baleendah ke Perpustakaan Kawalayaan Bandung

Alamat : Jln.wiranata Kusuma,Baleendah

Waktu : Bulan juli

5.1.3 Spesifikasi Sistem

Spesifikasi system merupakan spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk menjalankan program:

a. Spesifikasi Perangkat Keras

Tabel 5.1. 1 Spesifikasi Perangkat Keras

Prosesor	Intel(R) Core (TM) i3-6006U CPU @ 2.00GHz 1.99 GHz
RAM	8,00 GB (7,88 GB usable)
SSD	500 GB

b. Spesifikasi Perangkat Lunak

Tabel 5.1. 2 Spesifikasi Perangkat Lunak

Tools	Python versi 3.10
Tkinter	Versi 2.10
Microsoft word	Versi 356

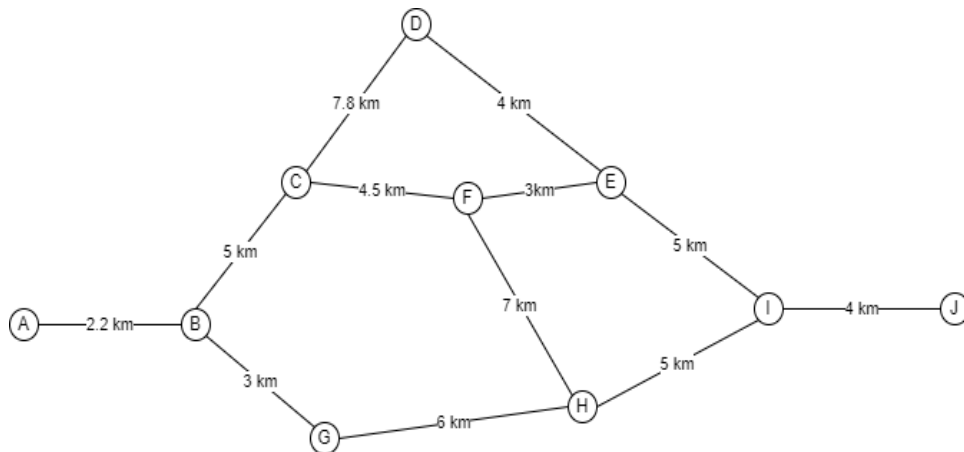
BAB VI PENUTUP

Sebagai penutup dari penulisan skripsi ini terdapat kesimpulan dan saran mengenai penelitian ini. Kesimpulan dan saran berdasarkan pada hasil penelitian yang telah diperoleh. Berikut kesimpulan dan saran tersebut.

6.1 Kesimpulan

Berdasarkan rumusan masalah dan serangkaian penelitian yang telah dilakukan didapatkan analisis dan hasil pembahasan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Berdasarkan representasi graf untuk rute perjalanan dari kosan Baleendah menuju perpustakaan kawalyaan Bandung terdiri atas satu graf sesuai dengan jalur. Representasi graf dilakukan dengan cara merepresentasikan titik awal (asal) dan titik tujuan menjadi titik-titik, garis jalan arah pada sisi graf seperti pada gambar berikut.



Gambar 6.1. 1 Graf Berbobot dan Berarah

2. Berdasarkan hasil penerapan algoritma Dijkstra dalam menentukan rute terpendek dari Baleendah ke Perpustakaan Kawalayaan Bandung didapatkan rute dan jarak terpendek sebagai berikut:
 - Baleendah, Dayehkolot, Bojongsoang, Batununggal Indah IV, Soekarno Hatta, Kawalayaan.
3. Bahasa pemrograman Python digunakan untuk mencari rute terpendek dari

Baleendah ke Perpustakaan Kawalayaan Bandung. Pada Python pengguna dapat mencari rute terpendek.

6.2 Saran

Berikut beberapa saran yang dapat digunakan untuk mengembangkan penelitian ini dalam menentukan rute terpendek:

1. Penelitian ini tidak menggunakan data base sebagai media penyimpanan lokasi atau maps, maka perlu untuk mengembang penelitian ini supaya lebih mudah lagi digunakan.
2. Memperluas cakupan wilayah untuk pengembangan penelitian berikutnya.
3. Penelitian ini mengabaikan pengalihan arus, kepadatan lalu lintas, dan aturan khusus lalu lintas lainnya. Pada penelitian selanjutnya diharapkan dapat menambahkan hal tersebut.

DAFTAR PUSTAKA

- Akhirina, T. Y., & Afrizal, T. (2020). Pendekatan Matriks Ketetangaan Berbobot Untuk Solusi Minimum Spanning Tree (Mst). *String (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 4(3), 280–287.
- Arabigh, M. H., & Saurina, N. (2020). Sistem Pencarian Lokasi Toko Batik Di Wilayah Surabaya Dengan Algoritma Dijkstra. *Melek It Information Technology Journal*, 6(2). <https://doi.org/10.30742/Melek-It.V6i2.314>
- Chusnah, C., & Purwanti, H. (2019). Pengaruh Disiplin Kerja Dan Motivasi Kerja Terhadap Kinerja Karyawan Pada Pt. Sgi. *Kinerja*, 2(01), 122–132.
- Elgamal, D. (N.D.). 23. Jurnal “*Analisis Kinerja Implementasi Algoritma Digital Signature Rsa (Rivest-Shamir-Adleman) Dan Elgamal Pada Kriptografi.*” 27.
- Fernando, Y., Mustaqov, M. A., & Megawaty, D. A. (2020). Penerapan Algoritma A-Star Pada Aplikasi Pencarian Lokasi Fotografi Di Bandar Lampung Berbasis Android. *Jurnal Teknoinfo*, 14(1), 27. <https://doi.org/10.33365/Jti.V14i1.509>
- Gautama, I. P. W., & Hermanto, K. (2020). Penentuan Rute Terpendek Dengan Menggunakan Algoritma Dijkstra Pada Jalur Bus Sekolah. *Jurnal Matematika*, 10(2), 116. <https://doi.org/10.24843/Jmat.2020.V10.I02.P128>
- Hasibuan, A. R. (2016). *Penerapan Algoritma Floyd Warshall Untuk Menentukan Jalur Terpendek Dalam Pengiriman Barang*. 3(6), 5.

- Maulana, G. G. (2017). Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web. *Jurnal Teknik Mesin*, 6(2), 8.
<https://doi.org/10.22441/jtm.v6i2.1183>
- Nugraha, D. W. (2011). *Aplikasi Algoritma Prim Untuk Menentukan Minimum Spanning Tree Suatu Graf Berbobot Dengan Menggunakan Pemrograman Berorientasi Objek*. 10.
- Pritandhari, M., & Wibawa, F. A. (2021). Pelatihan Desain Grafis Coreldraw Meningkatkan Kreativitas Karang Taruna Pemuda Mandiri Kelurahan Rejomulyo. *Sinar Sang Surya: Jurnal Pusat Pengabdian Kepada Masyarakat*, 5(1), 33–41.
- Suherman, R., & Kurnianda, N. R. (2021). *Optimization Of Vsat Ip Installation Routes Using The Dijkstra Algorithm*. 8(1), 5.
- Wahyudin, I., Tosida, E. T., & Andria, F. (N.D.). *Data Science Dan Big Data*.

LAMPIRAN

Lampiran A-1. Daftar Riwayat Hidup

Daftar Riwayat Hidup

Nama : Jahwan Awar Sani
 Tempat, Tgl Lahir : Tualang Baru, 29 April 2000
 Jenis Kelamin : Laki-laki
 Status : Lajang
 Alamat : Mbak Sako, Bukit Tusam, Aceh Tenggara
 Aceh, Indonesia 24671
 Telepon : 0822 1041 5765
 Email : jahwananwar.s@mail.com



PENDIDIKAN

2006 – 2012 SDN Tenebak bintang aceh tenggara
 2012 – 2015 SMP Negeri 2 bambel
 2015 – 2018 SMK Negeri 4 kutacane
 2018 – 2022 Universitas Bale Bandung
 (Sarjana Teknik Informatika)

SERTIFIKAT

- ✓ Python course
- ✓ Data Manipulation in Python A Pandas Crash Course
- ✓ The Complete Python Hacking Course

KEMAMPUAN

- ✓ Memahami jaringan teknologi informasi
- ✓ Menguasai pengoperasian komputer
- ✓ Software engineer
- ✓ Data analisis
- ✓ Web developer
- ✓ Menguasai bahasa Inggris secara lisan maupun tulisan