

LAPORAN TUGAS PROJECT AKHIR



Mata kuliah	TI0263 – Kecerdasan Buatan (GRUP C)
Dosen Pengampu	Gloria Virginia, S.Kom., MAI., Ph.D
Nama Kelompok	Kelompok Scan
Anggota Kelompok	<ol style="list-style-type: none">1. 71220848-Priesma Putra Reditama2. 71220849-Ezra Kristanto Nahumury3. 71220910-Dian Saro Bernad Zebua4. 71220913-Frederik Samra Sarongallo5. 71220914-Glen Daud Crasby Karo Karo6. 71220958-Kevin Glory Prasetyo
Deklarasi	Dengan ini kami menyatakan bahwa tugas ini merupakan hasil karya kelompok kami, tidak ada manipulasi data serta bukan merupakan plagiasi dari karya orang lain.



UNIVERSITAS KRISTEN DUTA WACANA
Fakultas Teknologi Informasi
Program Studi Informatika



Aplikasi : Campus Maps

Aplikasi Campus Maps adalah sebuah aplikasi yang dirancang khusus untuk membantu pengguna terkhususnya mahasiswa baru dalam navigasi di dalam sebuah kampus atau lingkungan universitas. Aplikasi ini menyediakan berbagai fitur untuk membantu pengguna menemukan lokasi tertentu, menentukan rute terpendek antara dua lokasi (Gedung 1 ke Gedung lainnya) .

Metode Inferensi : Heuristic Function & Best First Search

Menggunakan metode heuristic function dan dilanjutkan dengan algoritma best-first search, aplikasi Campus Maps dapat meningkatkan kecepatan pencarian jalur terbaik. dan dengan menggunakan best-first search, pencarian jalur terbaik dilakukan dengan mempertimbangkan nilai heuristik dari setiap simpul secara terus menerus, dan hanya menjelajahi simpul-simpul yang paling menjanjikan terlebih dahulu. Ini dapat mempercepat pencarian jalur terutama dalam lingkungan yang kompleks seperti universitas.

Contoh Kasus :

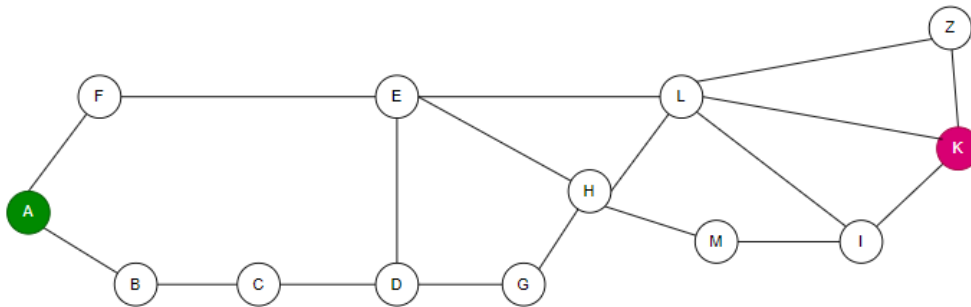


Keterangan :

- | | | | |
|-------------|-----------|-------------|----------------|
| A. AGAPE | E. EUDIO | I. IAMA | Z. GOR SAMAPTA |
| B. BIBLOS | F. FILIA | K. KOINONIA | |
| C. CHARA | G. GNOSIS | L. LOGOS | |
| D. DIDAKTOS | H. HAGIOS | M. MAKARIOS | |

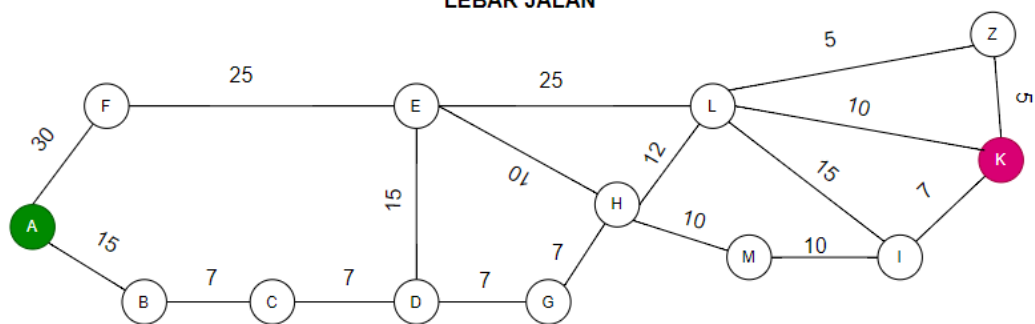
Soal : Buatlah Rute Tercepat Dari Agape Ke Koinonia

AGAPE KE KOINONIA



Perhitungan Berdasarkan Lebar Jalan :

PERHITUNGAN BERDASARKAN LEBAR JALAN



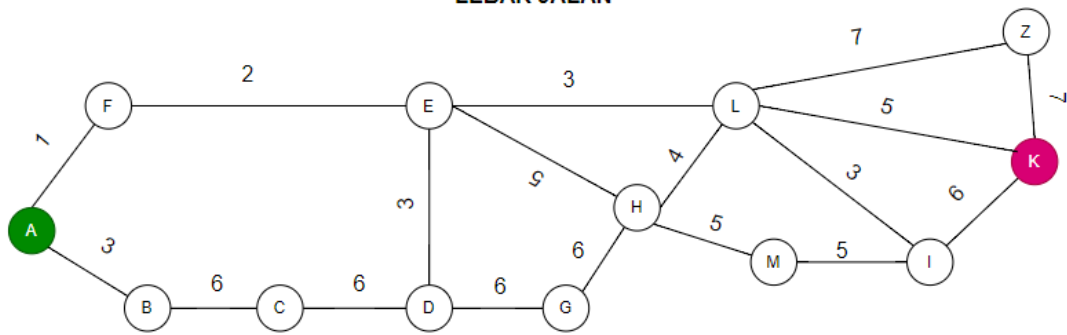
NOTE : SEMAKIN BESAR RUTE NYA (LEBAR JALAN)
MAKA ITU YANG AKAN DIJADIKAN JALUR TERCEPAT
KARENA DAPAT MENGURANGI KEPADATAN

Perubahan : Bobot (Lebar Terbesar akan diubah menjadi nilai Terkecil) agar kita bisa mendapatkan nilai terkecil yang bisa disimpulkan sebagai jalur tercepat.

Lebar Jalan	Nilai Perubahan
30	1
25	2
15	3
12	4
10	5
7	6
5	7

Semakin Besar Nilai Perubahannya maka dia akan semakin padat.

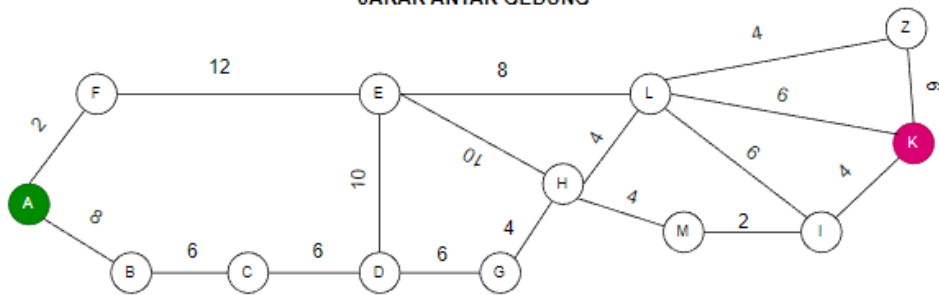
PERHITUNGAN BERDASARKAN LEBAR JALAN



NOTE : HASIL PERUBAHAN BOBOT

Perhitungan Berdasarkan Jarak Antar Gedung :

PERHITUNGAN BERDASARKAN JARAK ANTAR GEDUNG

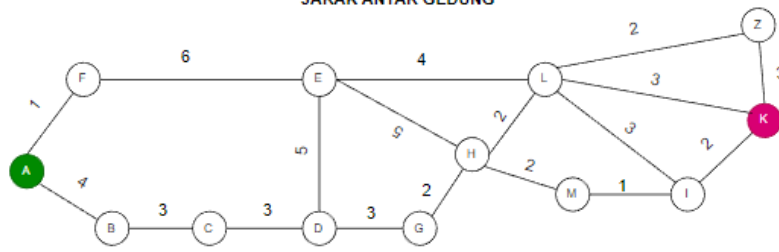


NOTE : SEMAKIN KECIL JARAK MAKA SEMAKIN
DEKAT DENGAN GEDUNG YANG LAIN

Perubahan : Jarak terkecil akan menjadi nilai perubahan yang paling kecil (jarak terbaik atau terdekat dengan Gedung yang lainnya).

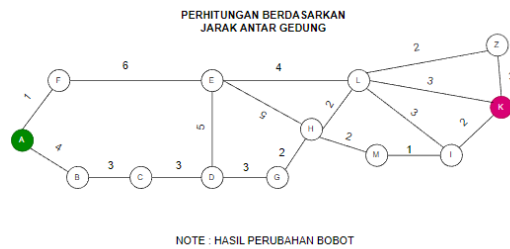
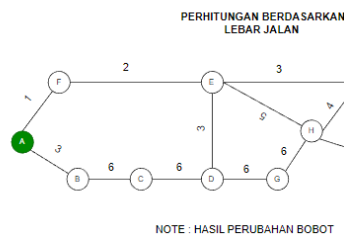
Jarak Jalan	Nilai Perubahan
2	1
4	2
6	3
8	4
10	5
12	6

PERHITUNGAN BERDASARKAN
JARAK ANTAR GEDUNG



NOTE : HASIL PERUBAHAN BOBOT

MAKA HASIL DARI KEDUA HEURISTIC FUNCTION NYA IALAH :



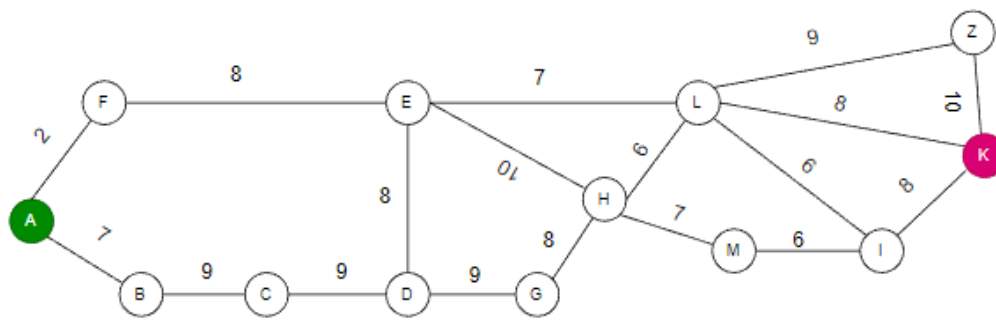
KNOWLADGE BASE :

Basis Pengetahuan dan Database untuk Mencari Rute Tercepat di Universitas Kristen Duta Wacana

Kelompok kami menggunakan basis pengetahuan tentang lebar jalan antar gedung di Universitas Kristen Duta Wacana, serta jarak antar gedung satu ke gedung lain. Data pengetahuan ini kemudian kami representasikan ke dalam bentuk database. Database ini akan dijalankan untuk mencari rute tercepat menggunakan metode BEST FIRST SEARCH. Dengan menggunakan basis pengetahuan ini, kami akan dapat mencari rute tercepat di dalam Universitas Kristen Duta Wacana. Dengan demikian, penggunaan BEST FIRST SEARCH akan memungkinkan kami untuk menemukan rute tercepat antar gedung-gedung di universitas tersebut.

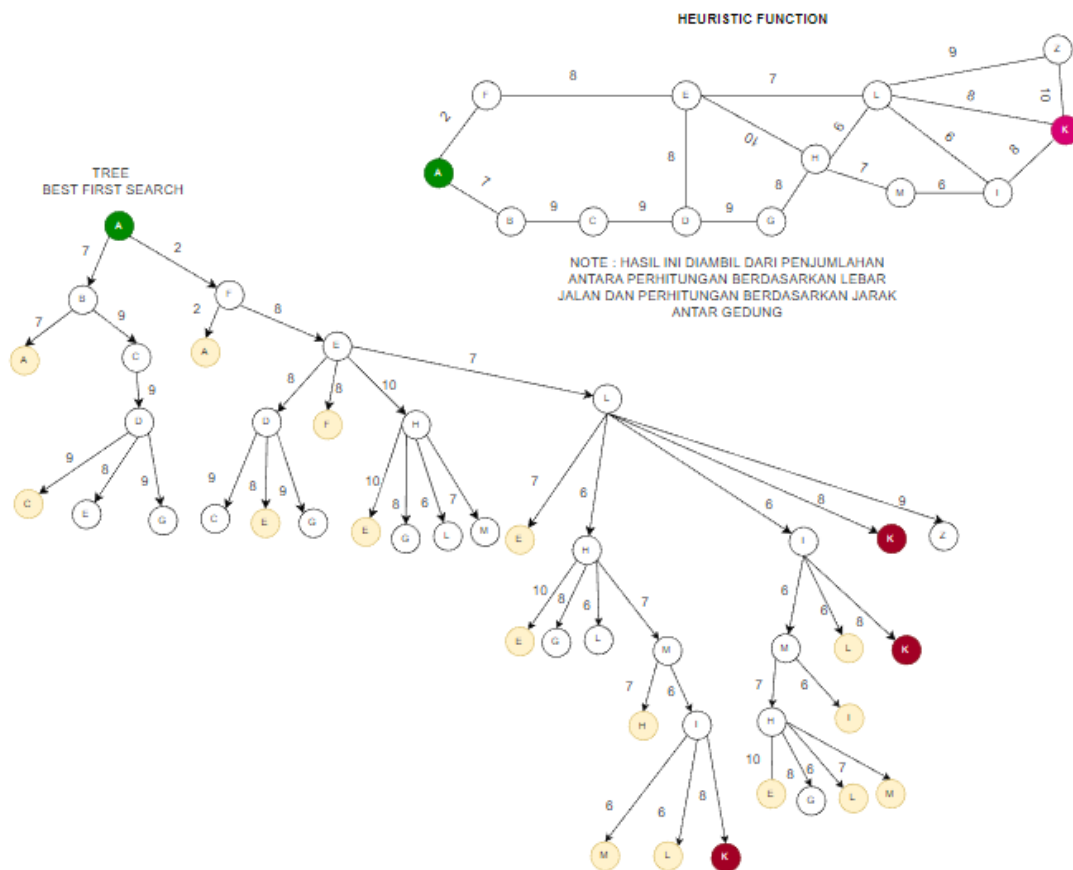
Nama Gedung	Lebar Jalan	Jarak Jalan	Nilai Perubahan
Agape ke Filia	1	1	2
Filia ke Euodia	2	6	8
Agape Ke biblos	3	4	7
Biblos ke Chara	6	3	9
Chara ke Didaktos	6	3	9
Euodia ke Logos	3	4	7
Euodia Ke Didaktos	3	5	8
Euodia Ke Hagios	5	5	10
Didaktos ke Gnosis	6	3	9
Gnosis Ke Hagios	6	2	8
Logos Ke Hagios	4	2	6
Hagios Ke Makarios	5	2	7
Makarios Ke lama	5	1	6
Logos Ke lama	3	3	6
lama Ke Koinonia	6	2	8
Logos Ke Koinonia	5	3	8
Logos Ke Ze	7	2	9
Ze Ke Koinonia	7	3	10

HEURISTIC FUNCTION



NOTE : HASIL INI DIAMBIL DARI PENJUMLAHAN
ANTARA PERHITUNGAN BERDASARKAN LEBAR
JALAN DAN PERHITUNGAN BERDASARKAN JARAK
ANTAR GEDUNG

BEST FIRST SEARCH :



NOTE : HASIL INI DIAMBIL DARI PENJUMLAHAN
ANTARA PERHITUNGAN BERDASARKAN LEBAR
JALAN DAN PERHITUNGAN BERDASARKAN JARAK
ANTAR GEDUNG

STEP	QUEUE	SORTED QUEUE
1	A(0)	A(0)
2	AB (7) , AF (2)	AF (2) , AB (7)
3	AFA (2) , AFE (8) , AB (7)	AB(7) , AFE (8)
4	ABA(7) , ABC (9) , AFE (8)	AFE(8) , ABC (9)
5	AFED (8) , AFEF(8) , AFEH(10) , AFEL (7) , ABC(9)	AFEL(7) , AFED (8) , ABC(9), AFEH(10)
6	AFELE(7) , AFELH(6) , AFELI(6) , AFELK(8) , AFELZ(9) , , AFED (8) , ABC(9), AFEH(10)	AFELI (6) , AFELH(6), AFED (8) , AFELK(8) , ABC(9) , AFELZ(9), AFEH(10)
7	AFELIM (6) , AFELIL (6) , AFELIK (8), AFELH(6), AFED (8) , AFELK(8) , ABC(9) , AFELZ(9), AFEH(10)	AFELH(6) , AFELIM (6), AFELIK (8), AFED (8) , AFELK(8) , ABC(9) , AFELZ(9), AFEH(10)
8	AFELHE (10) , AFELHG(8) , AFELHL(6) , AFELHM (7), AFELIM (6), AFELIK (8), AFED (8) , AFELK(8) , ABC(9) , AFELZ(9), AFEH(10)	AFELIM (6) , AFELHM (7), AFELIK (8), AFED (8) , AFELK(8) , AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)
9	AFELIMH(7) , AFELIMI(6) , AFELHM (7), AFELIK (8), AFED (8) , AFELK(8) , AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)	AFELHM (7) , AFELIMH(7), AFELIK (8), AFED (8) , AFELK(8) , AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)
10	AFELHMH(7) , AFELHMI (6) , AFELIMH(7), AFELIK (8), AFED (8) , AFELK(8) , AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)	AFELHMI (6) , AFELIMH(7), AFELIK (8), AFED (8) , AFELK(8) , AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)
11	AFELHMH(7) , AFELHMIM(6) , AFELHMIK(8) , AFELIMH(7), AFELIK (8), AFED (8) , AFELK(8), AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)	AFELIMH(7) , AFELIK (8), AFED (8) , AFELK(8), AFELHMIK(8), AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)
12	AFELIMHE(10) , AFELIMHG(8) , AFELIMHL(6) , AFELIMHM(7) , AFELIK (8), AFED(8), AFELK(8), AFELHMIK(8), AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)	AFELK(8) , AFELIK (8), AFED (8) AFELHMIK(8), AFELIMHG(8), AFELHG(8), ABC(9) , AFELZ(9), AFEH(10)

JAWABAN NYA : A – F – E – L – K

A = 0

L = 7

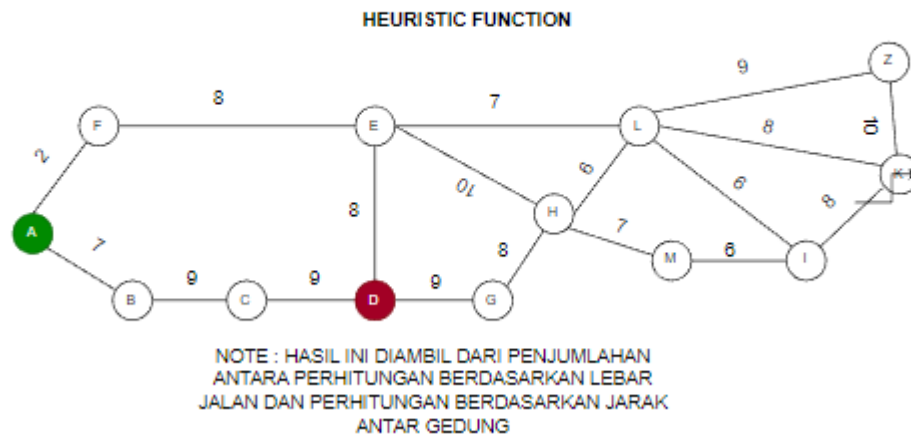
A – F – E – L – K = 25

F = 2

K = 8

E = 8

Soal : Buatlah Rute Tercepat Dari Agape Ke Didaktos



STEP	QUEUE	SORTED QUEUE
1	A(0)	A(0)
2	AB(7) , AF(2)	AF(2) , AB(7)
3	AFE(8) , ABA(2) , AB(7)	AB(7) , AFE(8)
4	ABC(9) , ABA(7) , AFE(8)	AFE(8) , ABC(9)
5	AFED(8) , AFEL(7) , AFEH(10) , AFEE(8) , ABC(9)	AFEL(7) , AFED(8) , ABC(9) , AFEH(10)
6	AFELE(7) , AFELZ(9) , AFELI(6) , AFELK(8) , AFELH(6) , AFED(8) , ABC(9) , AFEH(10)	AFELH(6) , AFELI(6) , AFELK(8) , AFED(8) , ABC(9) , AFELZ(9) , AFEH(10)
7	AFELHE(10) , AFELHG(8) , AFELHM(7) , AFELHL(6) , AFELI(6) , AFELK(8) , AFED(8) , ABC(9) , AFELZ(9) , AFEH(10)	AFELI(6) , AFELHM(7) , AFELK(8) , AFED(8) , AFELHG(8) , ABC(9) , AFELZ(9) , AFEH(10)
8	AFELIL(6) , AFELIM(6) , AFELIK(8) , AFELHM(7) , AFELK(8) , AFED(8) , AFELHG(8) , ABC(9) , AFELZ(9) , AFEH(10)	AFELIM(6) , AFELHM(7) , AFELK(8) , AFED(8) , AFELHG(8) , AFELIK(8) , ABC(9) , AFELZ(9) , AFEH(10)
9	AFELIMI(6) , AFELIMH(7) , AFELHM(7) , AFELK(8) , AFED(8) , AFELHG(8) , AFELIK(8) , ABC(9) , AFELZ(9) , AFEH(10)	AFELIMH(7) , AFELHM(7) , AFELK(8) , AFED(8) , AFELHG(8) , AFELIK(8) , ABC(9) , AFELZ(9) , AFEH(10)
10	AFELIMHM(7) , AFELIMHL(6) , AFELIMHG(8) , AFELIMHE(10) , AFELHM(7) , AFELK(8) , AFED(8) , AFELHG(8) , AFELIK(8) , ABC(9) , AFELZ(9) , AFEH(10)	AFELHM(7) , AFELK(8) , AFED(8) , AFELHG(8) , AFELIK(8) , AFELIMHG(8) , ABC(9) , AFELZ(9) , AFEH(10)

11.	AFELHMH(7) , AFELHMI(6) , AFELK(8), AFED(8), AFELHG(8) , AFELIK(8), AFELIMHG(8), ABC(9), AFELZ(9) , AFEH(10)	AFELHMI(6) , AFELK(8), AFED(8), AFELHG(8) , AFELIK(8), AFELIMHG(8), ABC(9), AFELZ(9) , AFEH(10)
12	AFELHMM(6) , AFELHMIK(8), AFELHML(6) , AFELK(8), AFED(8), AFELHG(8) , AFELIK(8), AFELIMHG(8), ABC(9), AFELZ(9) , AFEH(10)	AFED(8) , AFELK(8), AFELHG(8) , AFELIK(8), AFELIMHG(8), AFELHMIK(8), ABC(9), AFELZ(9) , AFEH(10)

JAWABAN = A – F – E – D

A = 0

F = 2 A – F – E – D = 18

E = 8

D = 8

APLIKASI : (Menggunakan PHP)

Code :

```
<?php

// Koneksi ke database (sesuaikan dengan informasi koneksi database Anda)
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "projectai";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $start = isset($_POST['start']) ? $_POST['start'] : '';
    $end = isset($_POST['end']) ? $_POST['end'] : '';

    if (!empty($start) && !empty($end)) {
        // Implementasi Best First Search
        $paths = findShortestPaths($start, $end, $conn);
        if ($paths) {
            echo "Rute Jalur Dari $start Ke $end:<br>";
            foreach ($paths as $path) {
                if ($path['cost'] > 0) { // Hanya tampilkan jalur dengan total
biaya lebih dari 0
                    echo "- " . $start . " -> " . implode(" -> ",
$path['path']) . " dengan Total Nilai " . $path['cost'] . "<br>";
                }
            }
        } else {
            echo "There is no path from $start to $end.";
        }
    } else {
        echo "Please select start and end locations.";
    }
}

// Tutup koneksi database setelah selesai digunakan
$conn->close();

// Fungsi untuk mencari jalur terpendek menggunakan Best First Search
function findShortestPaths($start, $end, $conn) {
    $queue = new SplPriorityQueue();
    $queue->insert([$start, [], 0], 0); // Menyimpan path dan total cost
```

```

$visited = [];

$shortestPaths = []; // Array untuk menyimpan semua jalur terpendek

while (!$queue->isEmpty()) {
    [$current, $path, $cost] = $queue->extract();
    if ($current == $end) {
        $shortestPaths[] = ['path' => $path, 'cost' => $cost]; //
        Menyimpan jalur terpendek
        continue; // Lanjut ke iterasi berikutnya untuk mencari jalur
        lainnya
    }

    if (!isset($visited[$current])) {
        $visited[$current] = true;

        // Ambil tetangga-tetangga dari current node dari database
        $sql = "SELECT node2, weight FROM connections_bidirectional WHERE
        node1='$current'";
        $result = $conn->query($sql);
        if ($result->num_rows > 0) {
            while ($row = $result->fetch_assoc()) {
                $neighbor = $row["node2"];
                $neighborCost = $row["weight"];
                if (!isset($visited[$neighbor])) {
                    $queue->insert([$neighbor, array_merge($path,
                    [$neighbor]), $cost + $neighborCost], -$neighborCost); // Prioritas negatif
                    agar yang terkecil keluar terlebih dahulu
                }
            }
        }
    }
}

return $shortestPaths; // Mengembalikan array dari jalur-jalur terpendek
}

?>

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Personal Portfolio Using HTML & CSS</title>
        <link rel="stylesheet" href="style.css" />
    </head>
    <body>

```

```

<header>
  <div class="header-left-side">
    <h1>
      Campus <br />
      Maps
    </h1>
    <h2>KELOMPOK 6 AI GRUP C</h2>
    <p>
      Aplikasi Campus Maps adalah sebuah aplikasi yang dirancang khusus
      untuk membantu pengguna
      terkhusus nya mahasiswa baru dalam navigasi di dalam sebuah kampus
      atau lingkungan universitas.
      Aplikasi ini menyediakan berbagai fitur untuk membantu pengguna
      menemukan lokasi tertentu, menentukan rute terpendek antara dua lokasi (
      Gedung 1 ke Gedung lainnya ) , dan mendapatkan informasi tambahan mengenai
      fasilitas-fasilitas di kampus
    </p>
    <form action="UI.php" method="post">
      <label for="start">Starting Location:</label>
      <select id="start" name="start">
        <!-- Opsi pilihan start location -->
        <option value="Agape">Agape</option>
        <option value="Biblos">Biblos</option>
        <option value="Chara">Chara</option>
        <option value="Didaktos">Didaktos</option>
        <option value="Euodia">Euodia</option>
        <option value="Filia">Filia</option>
        <option value="Gnosis">Gnosis</option>
        <option value="Hagios">Hagios</option>
        <option value="Iama">Iama</option>
        <option value="Koinonia">Koinonia</option>
        <option value="Logos">Logos</option>
        <option value="Makarios">Makarios</option>
        <option value="Ze">Ze</option>
      </select><br><br>

      <label for="end">End Location:</label>
      <select id="end" name="end">
        <!-- Opsi pilihan end location -->
        <option value="Agape">Agape</option>
        <option value="Biblos">Biblos</option>
        <option value="Chara">Chara</option>
        <option value="Didaktos">Didaktos</option>
        <option value="Euodia">Euodia</option>
        <option value="Filia">Filia</option>
        <option value="Gnosis">Gnosis</option>
        <option value="Hagios">Hagios</option>
        <option value="Iama">Iama</option>

```

```

        <option value="Koinonia">Koinonia</option>
        <option value="Logos">Logos</option>
        <option value="Makarios">Makarios</option>
        <option value="Ze">Ze</option>
    </select><br><br>

    <input type="submit" value="CARI">
</form>
</div>
<div class="header-right-side">
    
</div>
</header>
</body>
</html>

```

KNOWLADE BASE (DATABASE) :

Server: 127.0.0.1 » Database: projectai » Table: connections_b

Browse Structure SQL Search Insert

Showing rows 0 - 24 (36 total, Query took 0.0003 seconds.)

SELECT * FROM `connections_bidirectional`

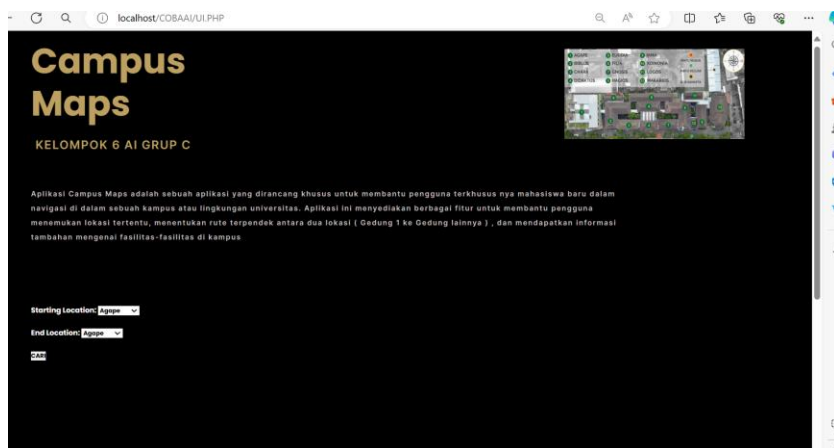
☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> ☐ Show all Number of rows: 25

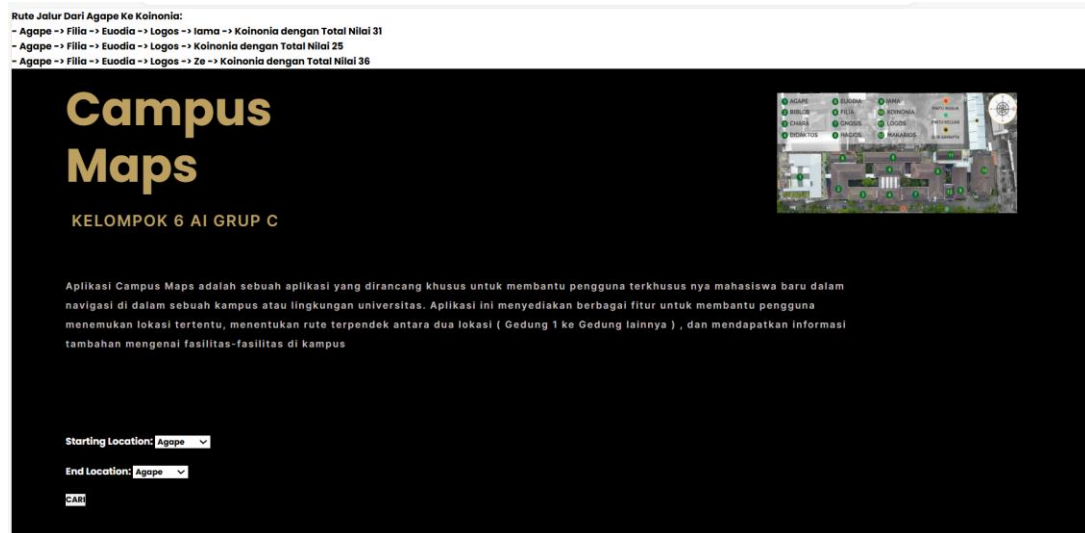
Extra options

node1	node2	weight
Agape	Filia	2
Filia	Agape	2
Filia	Euodia	8
Euodia	Filia	8
Agape	Biblos	7
Biblos	Agape	7
Biblos	Chara	9
Chara	Biblos	9
Chara	Didaktos	9
Didaktos	Chara	9
Euodia	Logos	7
Logos	Euodia	7
Euodia	Didaktos	8
Didaktos	Euodia	8
Euodia	Hagios	10
Hagios	Euodia	10
Didaktos	Gnosis	9
Gnosis	Didaktos	9
Gnosis	Hagios	8
Hagios	Gnosis	8
Logos	Hagios	6
Hagios	Logos	6
Hagios	Makarios	7
Makarios	Hagios	7
Makarios	Iama	6

UI :



Pembuktian : AGAPE TO KOINONIA



Pembuktian : AGAPE TO DIDAKTOS

