

PENENTUAN RUTE TERDEKAT MENGUNAKAN KOMPARASI ALGORITMA DIJKSTRA DAN ALGORITMA FLOYD WARSHALL DENGAN FITUR MAPBOX LEAFLET JS DAN CODEIGNITER VERS 4-BETA

PENENTUAN RUTE TERDEKAT MENGUNAKAN KOMPARASI ALGORITMA DIJKSTRA DAN ALGORITMA FLOYD WARSHALL DENGAN FITUR MAPBOX LEAFLET JS DAN CODEIGNITER VERS 4-BETA

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

Syafrial Fachrie Pane

Penyunting:

Faisal Syarifuddin

Desain sampul dan Tata letak:

Faisal Syarifuddin

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.’
Imam Syafi’i*

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 DATA ANALISIS	1
2 ALGORITMA DIJKSTRA	7
3 ALGORITMA FLOYD WARSHALL	13
4 PERBANDINGAN ALGORITMA	21
5 XAMPP (PHP VERSI 7)	23
6 CODEIGNITER VERSI 4 BETA	31
7 LEAFLET JS	39

DAFTAR ISI

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 DATA ANALISIS	1
1.1 Jenis Dan Sumber Data	1
1.1.1 Data Primer	2
1.1.2 Data Sekunder	3
1.2 Pemodelan Graf (Matriks Berbobot)	4
	ix

2	ALGORITMA DIJKSTRA	7
2.1	Algoritma Djikstra	7
2.2	Perhitungan Algoritma Djikstra	9
2.2.1	Analisis Algoritma Djikstra	9
3	ALGORITMA FLOYD WARSHALL	13
3.1	Algoritma Floyd Warshall	13
3.2	Perhitungan Algoritma Floyd Warshall	15
3.2.1	Analisis Algoritma Floyd Warshall	15
4	PERBANDINGAN ALGORITMA	21
4.1	PERBANDINGAN ALGORITMA DIJKSTRA DAN FLOYD WARSHALL	21
5	XAMPP (PHP VERSI 7)	23
5.1	Tutorial Install Xampp	24
5.2	Mengatasi Error Pada Xampp	27
6	CODEIGNITER VERSI 4 BETA	31
6.1	Tutorial Install CodeIgniter 4	31
6.2	Konfigurasi CodeIgniter 4	34
6.3	Konfigurasi Template CodeIgniter 4	35
7	LEAFLET JS	39
7.1	Cara Menampilkan Map Leaflet JS di CodeIgniter	39
7.2	Menampilkan Mapbox Pada index.php	41
7.3	Penentuan Jalur Terdekat Dengan Algoritma Djikstra dan Floyd Warshal	46
Daftar Pustaka		63
Index		65

DAFTAR GAMBAR

1.1	Jalur Lokasi	3
1.2	Graf Berbobot	4
2.1	Algoritma Dijkstra	8
2.2	Matriks Dijkstra	11
3.1	Matriks X0	16
5.1	Control Panel Xampp	27

DAFTAR TABEL

1.1	Data Jarak dan Perkiraan Lokasi	2
1.2	Data Longitude dan Latitude	3
1.3	Matriks Hubungan Graf	5
3.1	Matriks hubungan graf, $K=0$	16
3.2	Matriks hubungan graf, $K=1$	17
3.3	Matriks hubungan graf, $K=2$	18
3.4	Matriks hubungan graf, $K=3$	18
3.5	Hasil Akhir Matriks hubungan graf, K	20
4.1	Perbandingan Algoritma Dijkstra Dan Floyd-Warshall	22

Listings

2.1	Pseudocode Algoritma Dijkstra	11
3.1	Pseudocode Algoritma Floyd Warshall	20
6.1	Konfigurasi Template di CodeIgniter 4	35
7.1	File Leaflet CSS	39
7.2	File JavaScript Leaflet	40
7.3	Elemen div map Leaflet	40
7.4	Menampilkan Mapbox Leaflet JS di CodeIgniter 4	41
7.5	Setting setView Pada Leaflet JS	42
7.6	Call Library setView	43
7.7	Pemanggilan Library setView pada index.php	43
7.8	index.php Full Code	46
7.9	Function OnClick Dijkstra	51
7.10	Button Dijkstra	52
7.11	Function Algoritma Dijkstra	52
7.12	Function OnClick Dijkstra	54
7.13	Button Dijkstra	56
7.14	Function Algoritma Dijkstra	56
7.15	Function Mark Maps	58

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan CodeIgniter 4 dan Leaflet JS sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Agustus, 2019*

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

GLOSSARY

Dijkstra	Sebuah algoritma yang dipakai dalam memecahkan permasalahan jarak terpendek (shortest path problem) untuk sebuah graf berarah (directed graph).
Floyd Warshall	Salah satu varian dari pemrograman dinamis, metode untuk memecahkan masalah pencarian rute terpendek (sama seperti Algoritma Dijkstra).
XAMPP	Merupakan distribusi Apache yang benar-benar gratis dan mudah dipasang yang berisi MariaDB, PHP, dan Perl. Paket open source XAMPP telah diatur agar sangat mudah untuk diinstal dan digunakan.
CodeIgniter	Merupakan kerangka kerja PHP yang kuat dengan tapak yang sangat kecil, dibangun untuk pengembang yang membutuhkan toolkit sederhana dan elegan untuk membuat aplikasi web berfitur lengkap.
Leaflet JS	Merupakan pustaka JavaScript open-source terkemuka untuk peta interaktif ramah-mobile. Dengan berat hanya sekitar 38 KB JS, ia memiliki semua fitur pemetaan yang paling dibutuhkan pengembang.

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. php merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi php dan leaflet js pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

DATA ANALISIS

Sumber data terbagi menjadi dua yaitu data primer dan data sekunder. Data primer adalah data yang diperoleh peneliti secara langsung (dari tangan pertama), sementara data sekunder adalah data yang diperoleh peneliti dari sumber yang sudah ada. Sumber data penelitian yaitu sumber subjek dari tempat mana data bisa didapatkan. Adapun data yang digunakan dalam penelitian ini adalah kuantitatif, Data kuantitatif adalah data yang dapat diinput ke dalam skala pengukuran statistik. Fakta dan fenomena dalam data ini tidak dinyatakan dalam bahasa alami, melainkan dalam numerik.

1.1 Jenis Dan Sumber Data

Pada obyek study penelitian ini menggunakan data-data yang dihasilkan dari observasi langsung dan sebagian data yang di ambil dari google maps. Data yang diperoleh yaitu data nilai dari latitude, data nila longitude, jarak, perkiraan (menit), dan titik lokasi. Tentunya dengan bantuan google maps untuk keterangan selanjutnya. Data tersebut digunakan untuk menghitung dan mencari nilai maksimum untuk mendapatkan rute terdekat menuju daerah wisata yang di tuju oleh wisatawan

baik dari lokal maupun mancanegara. Rute yang di hasilkan dari sistem merupakan hasil proses perbandingan dan pengujian menggunakan komparasi Algoritma Dijkstra dan Floyd Warshall. Data wisata dikumpulkan dan di inputkan kedalam sistem, kemudian sistem dengan algoritma masing-masing akan melakukan eksekusi dan menampilkan hasil yang akurat dari proses masing-masing algoritma.

Pada penelitian ini data yang digunakan sebagai sample untuk analisis adalah rute yang dibuat di area Politeknik Pos Indonesia. Untuk analisis, data sample digunakan untuk mengetahui fungsi dan rumus dari alur algoritma tersebut. Selanjutnya untuk peng-implementasian kedua algoritma diuji dalam bentuk sistem informasi berbasis web. Kemudian data tersebut dikumpulkan dan dikelompokkan menjadi data primer dan data skunder.

1.1.1 Data Primer

Penelitian ini menggunakan data primer yang digunakan untuk mencari rute terdekat dalam pengolahan data menggunakan algoritma Floyd-Warshall. Pengumpulan data bertujuan untuk pengambilan data yang akan dianalisis yang kemudian dikumpulkan dan dipetakan dalam bentuk table agar mudah untuk dikelola, data diperoleh dari survey ke tempat langsung dan beberapa data lainnya di ambil dari google maps. Dari data tersebut dikumpulkan dan kemudian akan diproses pada tahap selanjutnya dalam pemodelan graf (Matriks Berbobot). Adapun data yang diperoleh dapat dilihat pada table 1.1.

Untuk table 1.1 merupakan data lokasi tentang titik-titik lokasi dan jarak serta perkiraan dari jarak satu ke jarak lainnya. Untuk lokasi awal itu merupakan titik pertama kemudian lokasi akhir merupakan titik tujuan dimana titik awal dan akhir saling berkesinambungan sehingga akan membentuk sebuah graf. Untuk jarak diperoleh dari google maps sedangkan perkirann waktu sampai, dari titik awal dan akhir di peroleh dari timer penulis yang menuju dari titik awal ke titik akhir.

Tabel 1.1 Data Jarak dan Perkiraan Lokasi

Node Jarak	Node Akhir	Jarak (m)
0	1	543
0	5	742
1	2	1827
1	4	945
2	3	1119
3	7	647
4	3	1152
5	6	778
6	7	1416

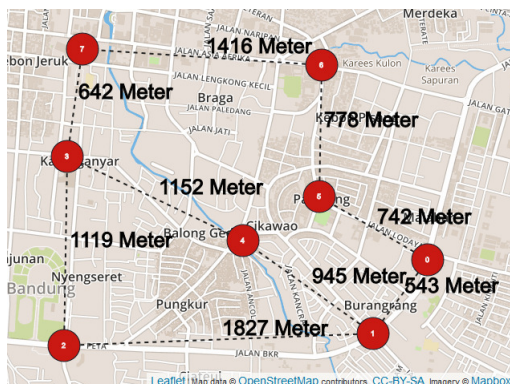
1.1.2 Data Sekunder

Data sekunder didapatkan dari Google Map. Data yang didapatkan dari Google Map yaitu rute-rute yang menghubungkan antara titik awal dengan titik yang akan dituju, serta jarak antar titik tiap jalannya. Data sekunder yang di miliki pada penelitian ini yaitu longitude latitude, graf lokasi, dan jalur lokasi. Data longitude dan latitude di peroleh dengan cara mengakses google maps dan mengklik titik-titik lokasi yang akan dilalui. Untuk data lokasi longitude dan latitude dapat dilihat pada table 1.2:

Tabel 1.2 Data Longitude dan Latitude

Titik Lokasi	Longitude, Latitude
0	-6.932966230283786, 107.62561082839966
1	-6.936885556542032, 107.62269258499146
2	-6.937524574035396, 107.60617017745972
3	-6.927470598347902, 107.60634183883667
4	-6.9319437919845655, 107.61574029922485
5	-6.929600695849448, 107.61981725692749
6	-6.922613940071385, 107.6199460029602
7	-6.921761889605031, 107.6071572303772

Pada tahap pemodelan graf, data yang ada diberikan beberapa perlakuan sehingga membentuk sebuah graf yang dibutuhkan oleh penelitian ini. Dari titik lokasi yang dikumpulkan, kemudian dihubungkan sesuai dengan titik awal dan akhir pada table 1.1 yang kemudian membentuk sebuah graf berbobot. Graf yang akan dihasilkan diperoleh dari proses pada table 1.1 dan table 1.2, serta jalur yang terbentuk dapat dilihat pada gambar 1.1.

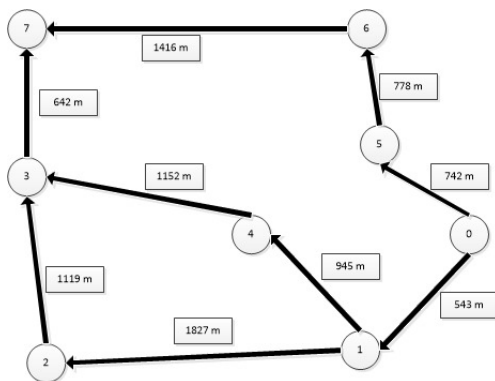


Gambar 1.1 Jalur Lokasi

Untuk jalur gambar 1.1 sudah terdapat bobot nilai yang tercantum yaitu jarak antara titik satu ke titik lainnya, bobot nilai ini akan di kumpulkan dan dibentuk kedalam table matriks agar dapat di proses pada tahap selanjutnya.

1.2 Pemodelan Graf (Matriks Berbobot)

Pemodelan graf adalah graf yang setiap sisinya diberi sebuah bobot. Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah kota atau titik, biaya perjalanan antara dua buah kota, waktu tempuh dari sebuah simpul ke simpul lain, ongkos produksi, dan sebagainya. Pemodelan graf dibentuk atas dasar sample data yang telah di kumpulkan, dimana titik awal dan titik tujuan di hubungkan dengan bobot nilai tertentu kemudian membentuk sebuah jalur dalam bentuk graf. Secara umum sebelum dilakukan iterasi, algoritma sudah mengidentifikasi jarak terdekat dari node terdekatnya. Jika seluruh node berbobot tertentu yang (positif), maka node terdekat berikutnya dari node asal dapat ditemukan selama node berdekatan dengan node awal. Untuk graf yang dihasilkan dapat dilihat pada gambar 1.2:



Gambar 1.2 Graf Berbobot

Graf di atas dapat dilihat bahwa x adalah titik awal dan y adalah titik akhir. Graf di atas merupakan graf satu arah, hubungan graf diperoleh dari gambar 1.2, dengan nilai jarak dan hubungan dari titik awal menuju titik akhir atau titik lainnya. Untuk matriks yang dihasilkan dari graf gambar 1.2, dapat dilihat pada table matriks 1.3:

Pada matriks dibawah merupakan gambaran dari graf berbobot dalam bentuk table matriks hubungan graf. Terdapat nilai yang diperoleh dari graf dan data dari hasil pengumpulan data lokasi dari google maps dan jarak yang diperoleh lalu dituliskan sesuai dengan data dari google maps dengan menghubungkan sumbu x sebagai awal dan sumbu y sebagai tujuan akhir jalur. Table matriks ini bernilai dengan satuan meter (m) dan untuk simbol "--" merupakan jarak yang tidak terjangkau atau tidak ada jalur terhadap titik tersebut.

Tabel 1.3 Matriks Hubungan Graf

	0	1	2	3	4	5	6	7
0	0	543	-	-	-	742	-	-
1	-	0	1827	-	945	-	-	-
2	-	-	0	1119	-	-	-	-
3	-	-	-	0	-	-	-	647
4	-	-	-	1152	0	-	-	-
5	-	-	-	-	-	0	778	-
6	-	-	-	-	-	-	0	1416
7	-	-	-	-	-	-	-	0

BAB 2

ALGORITMA DIJKSTRA

Algoritma Dijkstra (Jalur Terpendek Algoritma) adalah algoritma untuk menemukan jarak terpendek dari suatu vertex ke vertex yang lain pada suatu grafik yang berbobot, dimana jarak antar vertex adalah bobot dari setiap edge pada grafik tersebut [2].

2.1 Algoritma Dijkstra

Algoritma Dijkstra dinamakan sesuai dengan nama penemunya, seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra [3], adalah algoritma yang digunakan untuk mencari lintasan terpendek pada suatu graf berarah. Algoritma Dijkstra (Jalur Terpendek Algoritma) adalah algoritma untuk menemukan jarak terpendek dari suatu vertex ke vertex yang lain pada suatu grafik yang berbobot, dimana jarak antar vertex adalah bobot dari setiap edge pada grafik tersebut [2]. Algoritma dijkstra mencari jarak terpendek untuk setiap titik dari suatu graph yang berbobot. Algoritma dijkstra mencari jarak terpendek dari simpul asal ke simpul terdekatnya, kemudian ke simpul kedua, dan seterusnya [4]. Rumusa dalam algoritma ini adalah sebagai berikut:

$$G = V.E \quad (2.1)$$

Keterangan Rumus 2.1:

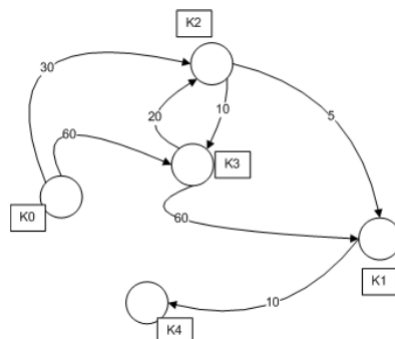
Sebuah grafik (G) didefinisikan oleh satu set simpul (Vertex = V) dan koleksi Edge (E).

Secara umum, sebelum dilakukan iterasi, vertex terdekatnya. Selama seluruh tepi berbobot tertentu yang (positif), maka vertex terdekat berikutnya dari node asal dapat ditemukan selama vertex bertemu dengan vertex Ti. Kumpulan simpul yang lengkap dengan simpul di Ti dapat dianggap sebagai "simpul pinggiran". Vertex inilah yang merupakan kandidat dari algoritma dijkstra untuk memilih vertex berikutnya dari node asal. Algoritma Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi [5]. Sifatnya sederhana dan lempang (straightforward) [6].

Ada beberapa kasus pencarian lintasan terpendek yang diselesaikan menggunakan algoritma Dijkstra, yaitu:

1. Pencarian lintasan terpendek antara dua buah simpul tertentu (a pair shortest path),
2. Pencarian lintasan terpendek antara semua pasangan simpul (all pairs shortest path).
3. Pencarian lintasan terpendek dari simpul tertentu ke semua simpul yang lain (single-source shortest path).
4. Pencarian lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (intermediate shortest path)

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



Gambar 2.1 Algoritma Dijkstra

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

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi).
2. Set semua node Belum terjamah dan set node awal sebagai Node keberangkatan
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan K0 ke K2 memiliki bobot jarak 30 dan dari K2 ke node K1 berjarak 5, maka jarak ke K1 melewati K2 menjadi $30+5=35$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai Node terjamah. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set Node belum terjamah dengan jarak terkecil (dari node keberangkatan) sebagai Node Keberangkatan selanjutnya dan lanjutkan dengan kembali ke step 3.

2.2 Perhitungan Algoritma Dijkstra

2.2.1 Analisis Algoritma Dijkstra

Implementasi algoritma Dijkstra dilakukan untuk memperoleh rute terpendek yang bisa ditempuh dari suatu titik ke titik finish. Hasil yang diperoleh dari implementasi algoritma Dijkstra ini adalah rute terpendek yang terdiri dari node-node yang termasuk ke dalam solusi rute terpendek, dan total bobot minimumnya. Deskripsi matematis untuk grafik dapat diwakili $G = \{V, E\}$, yang berarti sebuah grafik (G) didefinisikan oleh satu set simpul (Vertex = V) dan koleksi Edge (E).

Langkah-langkah algoritma Dijkstra dapat dilakukan dengan langkah - langkah yang sudah di jelaskan pada poin 2.1 bagian 2.1 dan berikut tahap analisis yang dilakukan penulis dalam penerapan algoritma Dijkstra:

1. Pada matriks hubungan graf table ?? sudah terdapat nilai dari setiap jalur yang akan dilewati, kemudian beri penamaan terhadap titik-titik lokasi tersebut dengan memberi label V sehingga seperti berikut :

$$V(G) = \{V0, V1, V2, V3, V4, V5, V6, V7\}.$$
2. Setelah titik-titik terbentuk, langkah selanjutnya adalah menghitung titik-titik yang saling berhubungan dengan cara berikut:

- (a) Dimulai dari menghitung G0 yaitu melihat node terdekat dari node0. Yang terdekat dari node0 adalah node1 dan node5. Node0 ke node1 memiliki jarak 543cm dan dari node0 ke node5 memiliki jarak 742cm. Nilai dari jumlah setiap jarak dimasukkan kedalam tabel V1 dan V5.
- (b) Setelah itu menghitung G1 karena dari G0 nilai yang terkecil yaitu dari V1 dengan nilai 543, maka V1 terlebih dahulu yang dikerjakan atau yang dihitung jarak terhadap node yang terdekat dari V1. Selanjutnya menghitung jarak yang terhubung langsung dengan V1 yaitu V2 dan V4 dengan perhitungan dihasilkan yaitu:
- $$G1 = V1 + (\text{jumlah jarak node1 ke node2}) = 2370$$
- $$G1 = V1 + (\text{jumlah jarak node1 ke node4}) = 1488$$
- (c) Setelah itu menghitung G5 karena dari G1 nilai yang terkecil yaitu dari V5 dengan nilai 742, maka V5 terlebih dahulu yang dikerjakan atau yang dihitung jarak terhadap node yang terdekat dari V5. Selanjutnya menghitung jarak yang terhubung langsung dengan V5 yaitu V6 dengan perhitungan dihasilkan yaitu:
- $$G5 = V5 + (\text{jumlah jarak node5 ke node6}) = 1520$$
- (d) Setelah itu menghitung G4 karena dari G5 nilai yang terkecil yaitu dari V4 dengan nilai 1488, maka V4 terlebih dahulu yang dikerjakan atau yang dihitung jarak terhadap node yang terdekat dari V4. Selanjutnya menghitung jarak yang terhubung langsung dengan V4 yaitu V3 dengan perhitungan dihasilkan yaitu:
- $$G4 = V4 + (\text{jumlah jarak node4 ke node3}) = 2640$$
- (e) Setelah itu menghitung G6 karena dari G4 nilai yang terkecil yaitu dari V6 dengan nilai 1520, maka V6 terlebih dahulu yang dikerjakan atau yang dihitung jarak terhadap node yang terdekat dari V6. Selanjutnya menghitung jarak yang terhubung langsung dengan V6 yaitu V3 dengan perhitungan dihasilkan yaitu:
- $$G6 = V6 + (\text{jumlah jarak node6 ke node7}) = 2936$$
- (f) Setelah itu menghitung G2 karena dari G6 nilai yang terkecil yaitu dari V2 dengan nilai 2370, maka V2 terlebih dahulu yang dikerjakan atau yang dihitung jarak terhadap node yang terdekat dari V2. Selanjutnya menghitung jarak yang terhubung langsung dengan V2 yaitu V3 dengan perhitungan dihasilkan yaitu:
- $$G2 = V2 + (\text{jumlah jarak node2 ke node3}) = 3489$$
- Sebelumnya pada node3 sudah didapatkan jaraknya yaitu dari node 4 menuju node3 dengan nilai 2640, sedangkan nilai pada node2 menuju node3 adalah 3489. Jadi nilai yang tetap diambil adalah nilai terkecil yaitu 2640 nilai dari node4 menuju node3.
- (g) Setelah itu menghitung G3 karena dari G2 nilai yang terkecil yaitu dari V3 dengan nilai 2640, maka V3 terlebih dahulu yang dikerjakan atau yang dihitung jarak terhadap node yang terdekat dari V3. Selanjutnya menghitung

jarak yang terhubung langsung dengan V3 yaitu V7 dengan perhitungan dihasilkan yaitu:

$$G3 = V3 + (\text{jumlah jarak node3 ke node7}) = 3282$$

Sebelumnya pada node7 sudah didapatkan jaraknya yaitu dari node 6 menuju node7 dengan nilai 2936, sedangkan nilai pada node3 menuju node7 adalah 3282. Jadi nilai yang tetap diambil adalah nilai terkecil yaitu 3282 nilai dari node6 menuju node7.

(h) Setelah itu G7 diturunkan saja.

(i) Dihasilkan table penyelesaian graf Dijkstra:

Unvisited (Q)	Visited(S)	Current	Node: Min = (dist[node], prev[node]) iteration							
Initialization{V0, V1,V2,V3,V4,V5,V6,V7}	{-}		V0	V1	V2	V3	V4	V5	V6	V7
{V0, V1,V2,V3,V4,V5,V6,V7}	{V0}	V0	0 ₀	543 ₀	∞	∞	∞	742 ₀	∞	∞
{V1,V2,V3,V4,V5,V6,V7}	{V1}	V1		543 ₀	2370 ₁	∞	1488 ₁	742 ₀	∞	∞
{V2,V3,V4,V5,V6,V7}	{V5}	V2			2370 ₁	∞	1488 ₁	742 ₀	1520 ₁	∞
{V3,V4,V5,V6,V7}	{V4}	V3			2370 ₁	2640 ₁	1488 ₁		1520 ₁	∞
{V4,V5,V6,V7}	{V6}	V4			2370 ₁	2640 ₁			1520 ₁	2936 ₆
{V5,V6,V7}	{V2}	V5			2370 ₁	2640 ₁				2936 ₆
{V6,V7}	{V3}	V6				2640 ₁				2936 ₆
{V7}	{V7}	V7								2936 ₆

Gambar 2.2 Matriks Dijkstra

Untuk membaca tabel perhitungan graf Dijkstra dan mengetahui lintasan mana yang terpendek yaitu:

7 berasal dari node 6

6 berasal dari node 5

5 berasal dari node 0

Dari hasil yang didapatkan maka jalur yang terdekat yang diperoleh dari Algoritma Dijkstra adalah rute dengan titik V 0 → 5 → 6 → 7 dengan jarak tempuh 2932 m. Adapun pseudocode yang akan diimplementasikan kedalam sistem berbasis web dari analisis algoritma Dijkstra adalah sebagai berikut :

```

1 function Dijkstra(Graph, source):
2   for each vertex v in Graph:
3     dist[v] := infinity ;
4     previous[v] := undefined ;
5   end for
6
7   dist[source] := 0 ;
8   Q := the set of all nodes in Graph ;
9   while Q is not empty:
10    u := vertex in Q with smallest distance in dist[] ;
11    remove u from Q ;
12    if dist[u] = infinity:
13      break ;
14    end if
15
16    for each neighbor v of u:

```

```
17         alt := dist[u] + dist_between(u, v) ;
18         if alt < dist[v]:
19             dist[v] := alt ;
20             previous[v] := u ;
21             decrease-key v in Q;
22         end if
23     end for
24
25     end while
26
27     return dist;
```

Listing 2.1 Pseudocode Algoritma Dijkstra

BAB 3

ALGORITMA FLOYD WARSHALL

Algoritma Floyd-Warshall adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu Algoritma Floyd Warshall adalah dengan membandingkan semua lintasan yang mungkin terjadi dalam graf untuk setiap pasang simpul dan melakukan pengujian dari setiap kombinasi simpul yang diperoleh [7].

3.1 Algoritma Floyd Warshall

Algoritma Floyd-Warshall adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu Algoritma Floyd Warshall adalah dengan membandingkan semua lintasan yang mungkin terjadi dalam graf untuk setiap pasang simpul dan melakukan pengujian dari setiap kombinasi simpul yang diperoleh [7]. Algoritma Floyd War-

shall ditemukan oleh Warshall untuk mencari path terpendek merupakan algoritma yang sederhana dan mudah implementasinya. Algoritma Floyd Warshall adalah matriks hubung graf berarah berlabel, dan keluarannya adalah path terpendek dari semua titik kesemua titik. Dalam usaha mencari jalur terpendek, algoritma Warshall memulai iterasi dari titik awalnya kemudian memperpanjang path dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang semini-mum mungkin [8]. mekanisme dari algoritma FloydWarshall ini terdiri dari beberapa langkah yang harus dilakukan, yaitu [9]:

1. Langkah awal yang harus dilakukan untuk menentukan shortest path dengan menggunakan algoritma Floyd-Warshall adalah dengan merepresentasikan suatu graf sebagai suatu matriks berbobot.
2. Langkah kedua adalah melakukan dekomposisi.
3. Langkah ketiga adalah menentukan struktur shortest path. Dalam hal ini, harus dilakukan dua pengamatan terlebih dahulu sebelum melangkah lebih jauh.
4. Dilakukan penentuan shortest path
5. Melakukan iterasi yang dimulai dari iterasi ke 0 sampai dengan.

Hasil akhir dari algoritma Floyd-Warshall adalah matriks untuk iterasi ke-n. Dari matriks ken ini, dapat dilihat shortest path untuk setiap vertex pada suatu graph.

Misalkan terdapat suatu graf G dengan simpul-simpul V yang masing-masing bernomor 1 s.d. N (sebanyak N buah). Misalkan pula terdapat suatu fungsi $\text{shortestPath}(i, j, k)$ yang mengembalikan kemungkinan jalur terpendek dari i ke j dengan hanya memanfaatkan simpul 1 s.d. k sebagai titik perantara. Tujuan akhir penggunaan fungsi ini adalah untuk mencari jalur terpendek dari setiap simpul i ke simpul j dengan perantara simpul 1 s.d. k+1.

Ada dua kemungkinan yang terjadi:

1. Jalur terpendek yang sebenarnya hanya berasal dari simpul-simpul yang berada antara 1 hingga k.
2. Ada sebagian jalur yang berasal dari simpul-simpul i s.d. k+1, dan juga dari k+1 hingga j.

Perlu diketahui bahwa jalur terpendek dari i ke j yang hanya melewati simpul 1 s.d. k telah didefinisikan pada fungsi $\text{shortestPath}(i, j, k)$ dan telah jelas bahwa jika ada solusi dari i s.d. k+1 hingga j, maka panjang dari solusi tadi adalah jumlah (konkatenasi) dari jalur terpendek dari i s.d. k+1 (yang melewati simpul-simpul 1 s.d. k), dan jalur terpendek dari k+1 s.d. j (juga menggunakan simpul-simpul dari 1 s.d. k).

Maka dari itu, rumus untuk fungsi $\text{shortestPath}(i, j, k)$ bisa ditulis sebagai:

$$X[i, j]X[i, k] + X[k, j] \quad (3.1)$$

Keterangan 3.1:

X = Matriks

i = titik awal

j = titik akhir

k = iterasi 1 sampai ke n

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.

3.2 Perhitungan Algoritma Floyd Warshall

3.2.1 Analisis Algoritma Floyd Warshall

Algoritma Floyd Warshall sangat efisien dari sudut pandang penyimpanan data karena dapat diimplementasikan dengan hanya pengubahan sebuah matriks jarak. Adapun mekanisme dari algoritma Floyd-Warshall ini terdiri dari beberapa langkah yang harus dilakukan, yaitu:

1. Langkah awal yang harus dilakukan untuk menentukan shortest path dengan menggunakan algoritma Floyd Warshall adalah dengan merepresentasikan suatu graf sebagai suatu matriks berbobot (dapat dilihat pada bagian 1.2).
2. Langkah kedua adalah melakukan dekomposisi Floyd-Warshall atau membentuk sebuah table matriks dan titik yang akan dihitung (dapat dilihat pada bagian 1.2).

(a) Hasil yang didapatkan dari graf adalah

$$\begin{aligned} K &= 0, 1, 2, 3, 4, 5, 6, 7 \\ i &= 0, 1, 2, 3, 4, 5, 6, 7 \\ j &= 0, 1, 2, 3, 4, 5, 6, 7 \end{aligned}$$

(b) Untuk rumus yang digunakan:

$$Rumus X[i, j] < X[i, k] + X[k, j] \quad (3.2)$$

Penjelasan lebih dapat dilihat pada point 3.1.

- (c) Untuk table matriks hubungan graf, terdapat tujuh table Matriks hubungan graf, dimana setiap table memiliki proses yang sama namun dengan nilai matriks yang berubah-ubah. Adapun table matriks yang dihasilkan diberi penamaan dengan table matriks hubungan graf.
- (d) Langkah selanjutnya, menentukan struktur shortest path kemudian melakukan penentuan shortest path dari i ke j yang memuat vertex k.

i. Matriks hubungan graf, K=0:

Tabel 3.1 Matriks hubungan graf, K=0

X0	V0	V1	V2	V3	V4	V5	V6	V7
V0	0	543	-	-	-	742	-	-
V1	-	0	1827	-	945	-	-	-
V2	-	-	0	1119	-	-	-	-
V3	-	-	-	0	-	-	-	647
V4	-	-	-	1152	0	-	-	-
V5	-	-	-	-	-	0	778	-
V6	-	-	-	-	-	0	28.93	-
V7	-	-	-	-	-	0	-	-

X0 [i,j]	X[i,k]	+	X[k,j]					
1,2	1,0		0,2	4,1	4,0	0,1		
1827	<	-	-	-	<	-	543	
1,3		1,0	0,3	4,2	4,0	0,2		
-	<	-	-	-	<	-	-	
1,4		1,0	0,4	4,3	4,0	0,3		
945	<	-	-	1152	<	-	-	
1,5		1,0	0,5	4,5	<	0,5		
-	<	-	742	-	<	742		
1,6		1,0	0,6	4,6	4,0	0,6		
-	<	-	-	-	<	-	-	
1,7		1,0	0,7	4,7	4,0	0,7		
-	<	-	-	-	<	-	-	
2,1		2,0	0,1	5,1	5,0	0,1		
-	<	-	543	-	<	-	543	
2,3		2,0	0,3	5,2	5,0	0,2		
1119	<	-	-	-	<	-	-	
2,4		2,0	0,4	5,3	5,0	0,3		
-	<	-	-	-	<	-	-	
2,5		2,0	0,5	5,4	5,0	0,4		
-	<	-	742	-	<	-	-	
2,6		2,0	0,6	5,6	5,0	0,6		
-	<	-	-	778	<	-	-	
2,7		2,0	0,7	5,7	5,0	0,7		
-	<	-	-	-	<	-	-	
3,1		3,0	0,1	6,1	6,0	0,1		
-	<	-	543	-	<	-	543	
3,2		3,0	0,2	6,2	6,0	0,2		
-	<	-	-	-	<	-	-	
3,4		3,0	0,4	6,3	6,0	0,3		
-	<	-	-	-	<	-	-	
3,5		3,0	0,5	6,4	6,0	0,4		
-	<	-	742	-	<	-	-	
3,6		3,0	0,6	6,5	6,0	0,5		
-	<	-	-	-	<	-	742	
3,7		3,0	0,7	6,7	6,0	0,7		
647	<	-	-	1416	<	-	-	
			7,1	7,0	0,1			
			-	<	-	543		
			7,2	7,0	0,2			
			-	<	-	-		
			7,3	7,0	0,3			
			-	<	-	-		
			7,4	7,0	0,4			
			-	<	-	-		
			7,5	7,0	0,5			
			-	<	-	742		
			7,6	7,0	0,6			
			-	<	-	-		

Gambar 3.1 Matriks X0

Penyelesaian:

Untuk iterasi terhadap matriks K=0 tidak terdapat jalur dan setiap sel matriks W dicek apakah $X[i, j] < X[i, k] + X[k, j]$ jika ya maka $X[i, j]$ digannti dengan $X[i, k] + X[k, j]$

ii. Matriks hubungan graf, $K=1$:

Tabel 3.2 Matriks hubungan graf, $K=1$

X1	V0	V1	V2	V3	V4	V5	V6	V7
V0	0	543	2370	-	1488	742	-	-
V1	-	0	1827	-	945	-	-	-
V2	-	-	0	1119	-	-	-	-
V3	-	-	-	0	-	-	-	647
V4	-	-	-	1152	0	-	-	-
V5	-	-	-	-	-	0	778	-
V6	-	-	-	-	-	0	28.93	-
V7	-	-	-	-	-	0	-	-

Penyelesaian:

$$X1[i, j] > X[i, k] + X[k, j]$$

$$0, 2 > 0, 1 + 1, 2$$

$$\rightarrow 543 + 1827$$

$$0, 4 > 0, 1 + 1, 4$$

$$\rightarrow 543 + 945$$

Karena $X1[0, 2]$ lebih besar dari jumlah $X[0, 1] + X[1, 2]$, maka nilai $X1[0, 2]$ diubah menjadi nilai total $X[0, 1] + X[1, 2]$ yaitu 2370 m. Dan $X1[0, 4]$ lebih besar dari jumlah $X[0, 1] + X[1, 4]$, maka nilai $X1[0, 4]$ diubah menjadi nilai total $X[0, 1] + X[1, 4]$ yaitu 1488 m. Ini menandakan bahwa pada $X1$, terdapat rute untuk menuju node ke 2 dan 4. jalur yang terbentuk adalah $0 \rightarrow 1 \rightarrow 2$ dan $0 \rightarrow 1 \rightarrow 4$.

iii. Matriks hubungan graf, K=2:

Tabel 3.3 Matriks hubungan graf, K=2

X2	V0	V1	V2	V3	V4	V5	V6	V7
V0	0	543	2370	3489	1488	742	-	-
V1	-	0	1827	2946	945	-	-	-
V2	-	-	0	1119	-	-	-	-
V3	-	-	-	0	-	-	-	647
V4	-	-	-	1152	0	-	-	-
V5	-	-	-	-	-	0	778	-
V6	-	-	-	-	-	0	28.93	-
V7	-	-	-	-	-	0	-	-

Penyelesaian:

$$X2[i, j] > X[i, k] + X[k, j]$$

$$0, 3 > 0, 2 + 2, 3$$

$$- > 2370 + 1119$$

$$1, 3 > 1, 2 + 2, 3$$

$$- > 1827 + 1119$$

Karena $X2[0, 3]$ lebih besar dari jumlah $X[0, 2] + X[2, 3]$, maka nilai $X2[1, 3]$ diubah menjadi nilai total $X[1, 2] + X[2, 3]$. Sehingga jalur yang dihasilkan pada matriks graf K=2 adalah $0 \rightarrow 2 \rightarrow 3$ dan $1 \rightarrow 2 \rightarrow 3$.

iv. Matriks hubungan graf, K=3:

Tabel 3.4 Matriks hubungan graf, K=3

X3	V0	V1	V2	V3	V4	V5	V6	V7
V0	0	543	2370	3489	1488	742	-	4136
V1	-	0	1827	2946	945	-	-	3593
V2	-	-	0	1119	-	-	-	1766
V3	-	-	-	0	-	-	-	647
V4	-	-	-	1152	0	-	-	1799
V5	-	-	-	-	-	0	778	-
V6	-	-	-	-	-	0	28.93	-
V7	-	-	-	-	-	0	-	-

Penyelesaian:

$$X3[i, j] > X[i, k] + X[k, j]$$

$$0, 7 > 0, 3 + 3, 7$$

$$- > 3489 + 647$$

$$1, 7 > 1, 3 + 3, 7$$

$$- > 2946 + 647$$

$$2, 7 > 2, 3 + 3, 7$$

$$- > 1119 + 647$$

$$4, 7 > 4, 3 + 3, 7$$

$$- > 1152 + 647$$

Karena $X3[0, 7]$ lebih besar dari jumlah $X[0, 3] + X[3, 7]$, maka nilai $X2[0, 7]$ diubah menjadi nilai total $X[0, 3] + X[3, 7]$. Sehingga jalur yang dihasilkan pada matriks graf $K=3$ adalah $0 \rightarrow 3 \rightarrow 7$. Dan untuk nilai $x[1,7]$, $x[2,7]$, $x[4,7]$ juga diubah.

v. Matriks hubungan graf, $K=4$:

Penyelesaian:

$$X4[i, j] > X[i, k] + X[k, j]$$

$$0, 7 > 0, 4 + 4, 7$$

$$4136 > 1488 + 1799$$

$$1, 3 > 1, 4 + 4, 3$$

$$2946 > 945 + 1152$$

$$1, 7 > 1, 4 + 4, 7$$

$$3593 > 945 + 1799$$

Karena $X3[0, 7]$ lebih besar dari jumlah $X[0, 4] + X[4, 7]$, maka nilai $X3[0, 7]$ diubah menjadi nilai total $X[0, 4] + X[4, 7]$. Sehingga jalur yang dihasilkan pada matriks graf $K=3$ adalah $0 \rightarrow 4 \rightarrow 7$. Dan untuk nilai $x[1,3]$, $x[1,7]$ juga diubah. Untuk matriks selanjutnya juga di hitung dengan cara yang sama sehingga dihasilkan beberapa titik jalur dan perubahan matriks pada setiap matriks K . Kemudian pada Matriks hubungan graf, $K=5, K=6$ dan $K=7$ tidak terjadi perubahan pada matriksnya.

3. Setelah melakukan perhitungan table matriks graf, langkah selanjutnya kemudian melakukan iterasi yang dimulai dari iterasi ke 0 sampai dengan n . Adapun hasil akhir table matriks hubungan graf K sebagai berikut:

Tabel 3.5 Hasil Akhir Matriks hubungan graf, K

Dari/Ke	0	1	2	3	4	5	6	7
0	0	543	2370	3489	1488	742	-	3287
1	-	0	1827	2097	945	-	-	2744
2	-	-	0	1119	-	-	-	1766
3	-	-	-	0	-	-	-	647
4	-	-	-	1152	0	-	-	1799
5	-	-	-	-	-	0	778	-
6	-	-	-	-	-	-	0	1416
7	-	-	-	-	-	-	-	0

Kemudian dihasilkan jalur dari perhitungan tersebut:

0 → 5 → 6 → 7 Dengan Jarak 2936 m

0 → 1 → 2 → 3 → 7 Dengan Jarak 4136 m

0 → 1 → 4 → 3 → 7 Dengan Jarak 3287 m

4. Hasil akhir dari algoritma ini adalah jalur terpendek yang dihasilkan dari iterasi matriks graf, sehingga dihasilkan jalur dengan node 0 → 5 → 6 → 7 dengan jarak 2936 m. Adapun pseudocode yang akan diimplementasikan kedalam sistem berbasis web dari analisis algoritma Floyd Warshall adalah sebagai berikut:

```

1  function floydwarshall(int[1..n,1..n] graph) {
2      // Inisialisasi
3      var int[1..n,1..n] jarak:= graph
4      var int[1..n,1..n] sebelum
5      for i from 1 to n
6          for j from 1 to n
7              if jarak[i,j] < Tak-hingga
8                  sebelum[i,j]:= i
9      // Perulangan utama pada algoritme
10     for k from 1 to n
11         for i from 1 to n
12             for j from 1 to n
13                 if jarak[i,j] > jarak[i,k] + jarak[k,j]
14                     jarak[i,j] = jarak[i,k] + jarak[k,j]
15                     sebelum[i,j] = sebelum[k,j]
16     return jarak
17 }
```

Listing 3.1 Pseudocode Algoritma Floyd Warshall

BAB 4

PERBANDINGAN ALGORITMA

Pengujian ini dilakukan secara bergantian menggunakan algoritma Dijkstra dan Algoritma Floyd Warshall dengan titik yang sama kemudian dilakukan perbandingan dengan kedua algoritma.

4.1 PERBANDINGAN ALGORITMA DIJKSTRA DAN FLOYD WARSHALL

Menghitung jalur terpendek dengan mencari jarak antara jalur-jalur yang ada. Menentukan rute optimal dengan parameter waktu tempuh tercepat setelah melakukan pengolahan data dengan menggunakan algoritma Dijkstra dan algoritma Floyd-Warshall. Dari perbandingan kedua algoritma tersebut memiliki tingkat akurasi hasil yang sama. Namun kedua algoritma ini memiliki cara dan fungsi berbeda untuk mencari rute terdekat dan hasil yang diperoleh kedua algoritma tersebut memiliki jalur yang sama. Hasil setiap masing-masing algoritma yang telah di analisis maka dihasilkan berikut table perbandingan dengan kedua algoritma:

Tabel 4.1 Perbandingan Algoritma Dijkstra Dan Floyd-Warshall

	Algoritma Dijkstra	Algoritma Floyd Warshall	Akurasi
Rute	0 -> 5 -> 6 -> 7	0 -> 5 -> 6 -> 7	Sama
Jarak	2936 m	2936 m	Sama
Matriks	Matriks Dijkstra	Matriks Graf K	Beda

BAB 5

XAMPP (PHP VERSI 7)

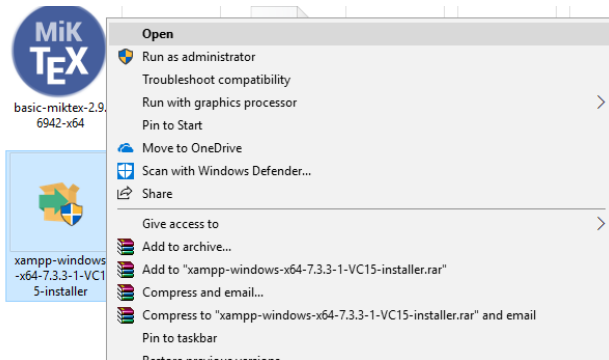
Xampp adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program, Fungsinya adalah sebagai server yang berdiri sendiri (localhost) [10]. Xampp merupakan singkatan dari X (empat sistem operasi apapun), Apache, Mysql, PHP, dan Perl. Xampp adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket Xampp sudah terdapat Apache (web server), Mysql (database), PHP (server side scripting), Perl ,FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya. XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program [11].

Dari definisi tersebut, penulis menyimpulkan bahwa XAMPP adalah sebuah software yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP berperan sebagai server web pada komputer. XAMPP juga dapat disebut sebuah cpanel server virtual, yang dapat membantu melakukan preview sehingga dapat memodifikasi website tanpa harus online atau terakses dengan internet.

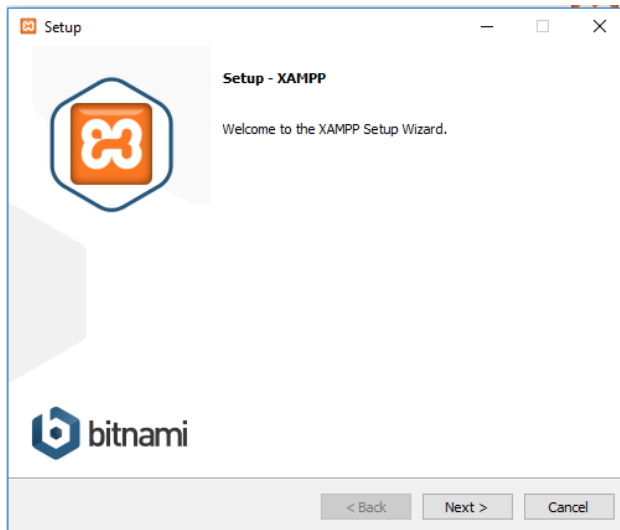
Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Selain itu XAMPP adalah 100% open source, tersedia bebas dan legal [12].

5.1 Tutorial Install Xampp

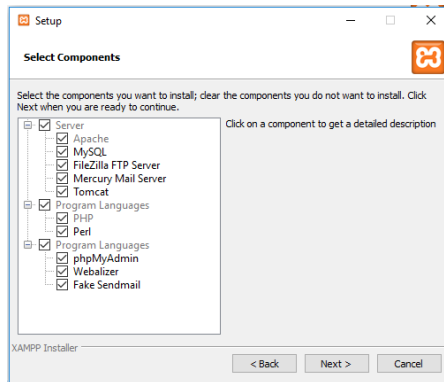
1. Download terlebih dahulu aplikasi Xampp di <https://www.apachefriends.org/index.html>, download sesuai sistem operasi yang anda gunakan, pada tutorial kali ini saya akan melakukan instalasi XAMPP di Windows 10.
2. Setelah download aplikasi, lakukan instalasi XAMPP, dengan cara klik kanan pada file instalasi kemudian pilih Open.



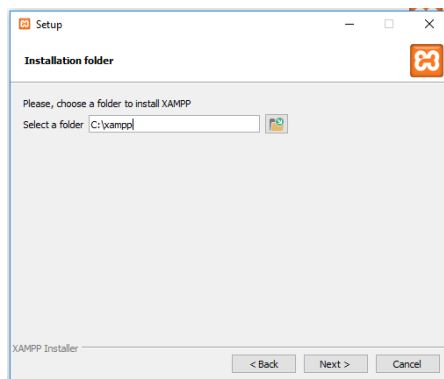
3. Jika pada saat melakukan instalasi muncul peringatan yang bertujuan untuk memastikan apakah Anda akan menginstal aplikasi ini, Silakan klik Ok/Yes untuk melanjutkan instalasi.



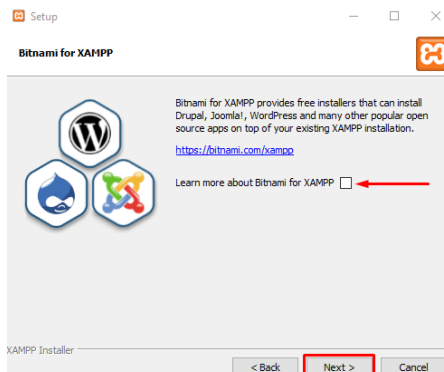
4. Klik next untuk melanjutkan, kemudian akan tampil pilihan aplikasi apa yang akan Anda install dan tidak ingin Anda install.



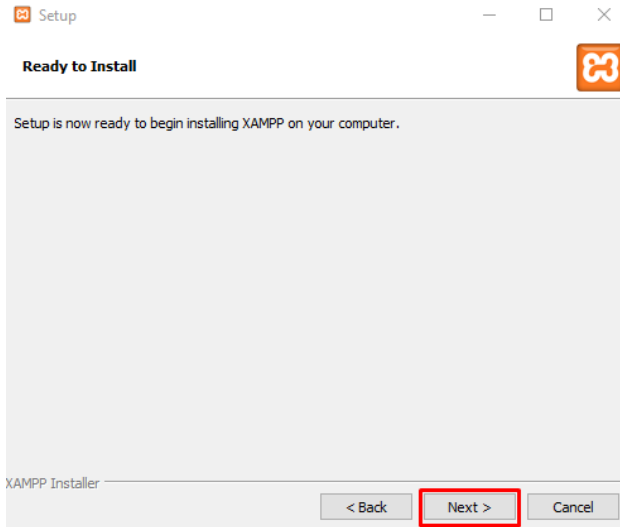
5. Tahap selanjutnya adalah memilih folder dimana lokasi instalasi xampp akan disimpan.



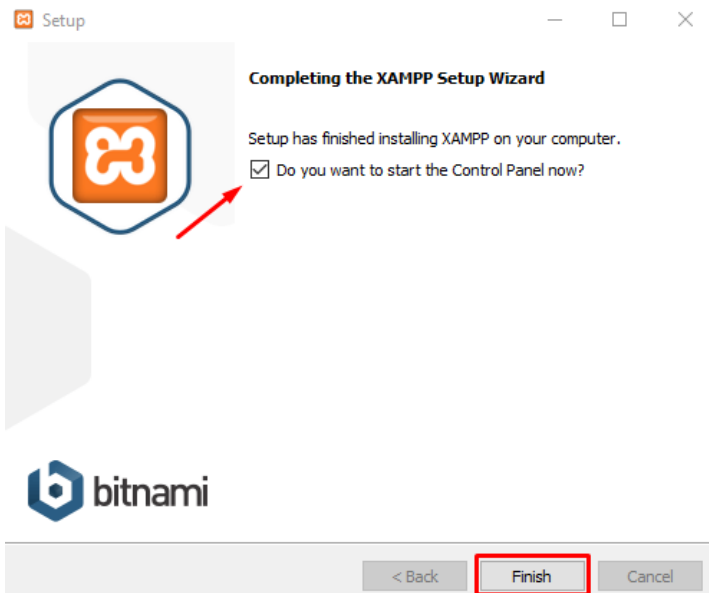
6. Silakan hilangkan centang pada Learn more about Bitnami for XAMPP, kemudian klik Next.



7. Klik next untuk malnjutkan ke proses instalasi xampp.

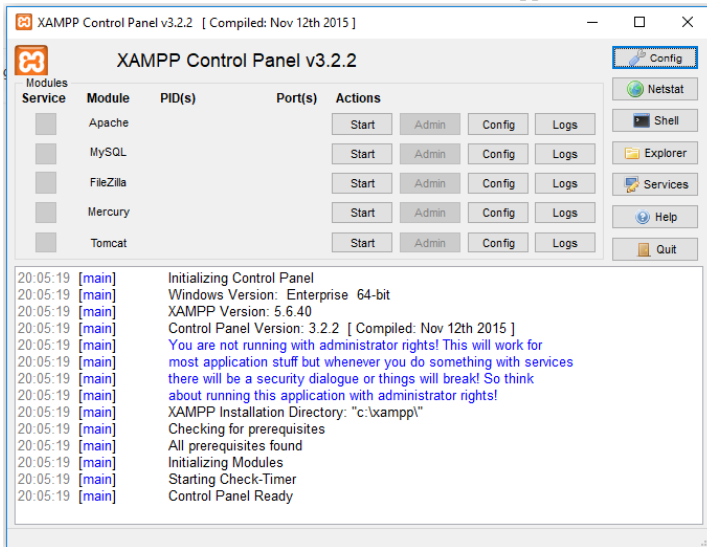


8. Apabila aplikasi sudah terinstal maka akan tampil pertanyaan mengenai apakah Anda ingin langsung menjalankan control panel. Pastikan pilihan tersebut sudah tercentang, kemudian klik tombol Finish.



9. Control panel akan muncul otomatis, tapi jika Anda tidak mencentang pilihan di halaman sebelumnya, maka Anda perlu membuka langsung control panel melalui start menu atau folder XAMPP di komputer Anda.
10. Apabila control panel sudah muncul dan terlihat seperti gambar 5.1, maka proses instalasi Xampp berhasil.

Gambar 5.1 Control Panel Xampp



5.2 Mengatasi Error Pada Xampp

Hal yang menjadi penyebab utama kenapa tampil error pada XAMPP biasanya disebabkan aplikasi lain pada komputer Anda menggunakan port 80 atau 443, dimana port tersebut digunakan oleh Apache dan MySQL. Berikut cara mengatasi error pada xampp:

1. Klik Start, kemudian ketikkan services.msc pilih Services yang ada di bagian Best match.
2. Scrol ke bawah, pada bagian World Wide Web Publishing Service klik kanan dan pilih Stop.
3. Silakan close XAMPP, kemudian buka kembali dan jalankan Apache dan MySQL pada XAMPP.

Jika langkah yang Anda lakukan tidak berhasil mengatasi masalah yang dihadapi atau tidak menemukan World Wide Web Publishing, silakan lakukan langkah di bawah ini:

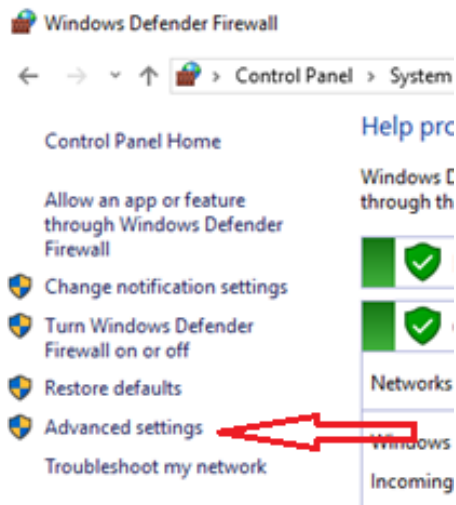
1. Buka control panel melalui tombol start yang ada pada pojok kiri bawah
2. Kemudian pilih system and security



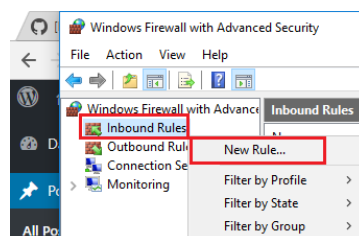
3. Pilih windows defender firewall



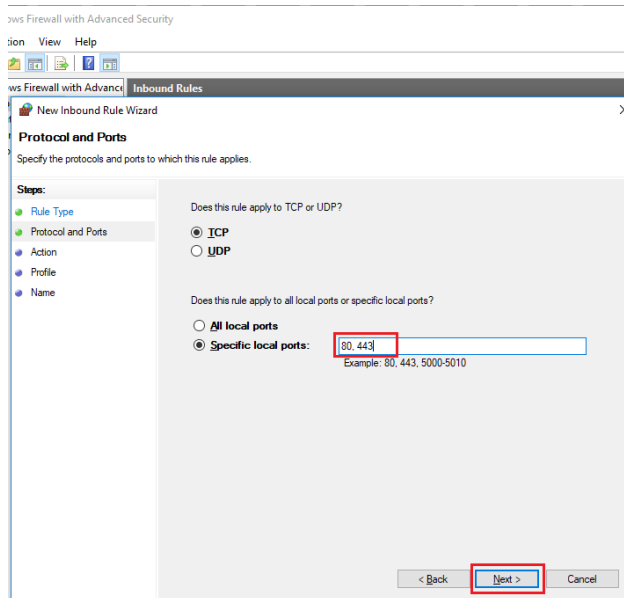
4. Pilih advanced settings



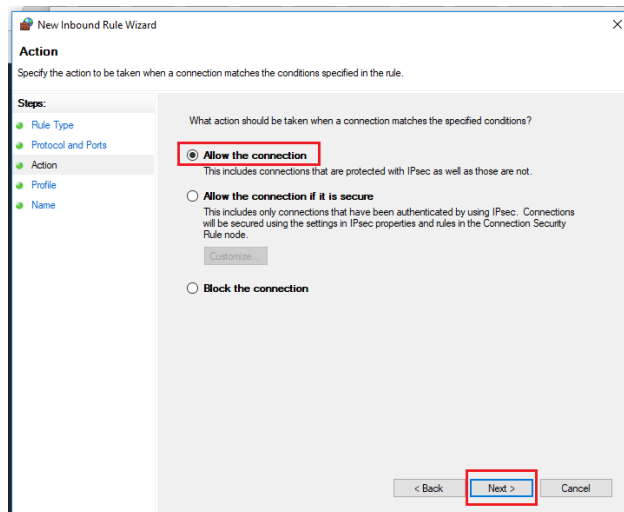
5. Klik Inbound dan klik kanan kemudian pilih New Rule, dapat dilihat seperti pada gambar dibawah



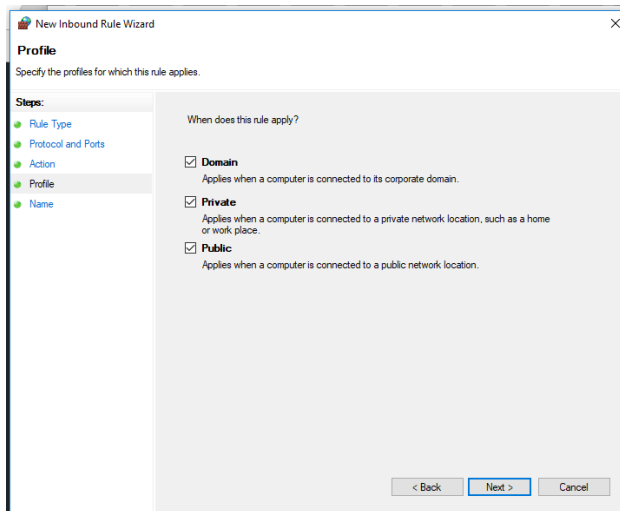
6. Pilih Port dan tekan tombol Next, kemudian pada kolom Specific Ports isi dengan 80, 443 kemudian klik Next.



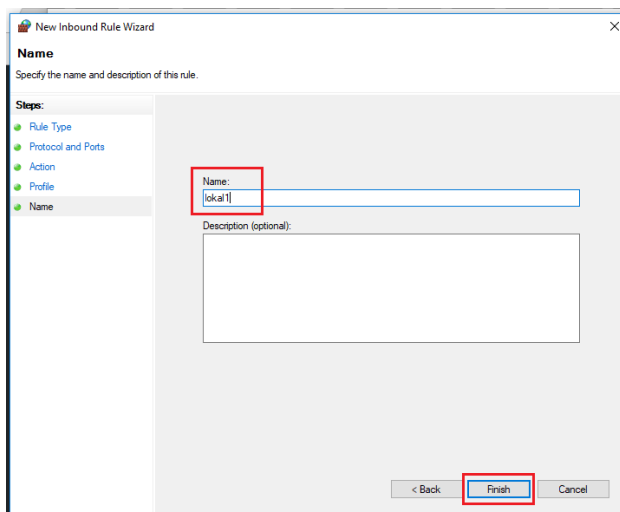
7. Centang Allow the Connection kemudian klik Next



8. Pastikan semua pilihan dicentang seperti pada gambar dibawah, kemudian klik Next



9. Masukkan lokal1 pada kolom name, kemudian klik Finish



10. Ulangi kembali langkah 1 sampai 6, untuk langkah 6 isi dengan lokal2, kemudian klik Finish

11. Restart komputer Anda

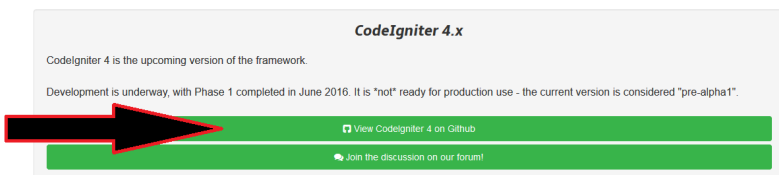
BAB 6

CODEIGNITER VERSI 4 BETA

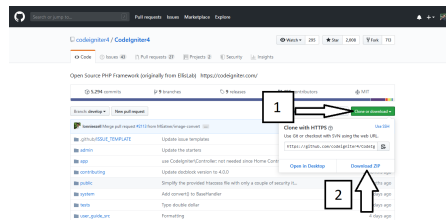
Codeigniter adalah sebuah framework untuk web yang dibuat dalam format PHP. Format yang dibuat ini selanjutnya dapat digunakan untuk membuat sistem aplikasi web yang kompleks. Codeigniter dapat mempercepat proses pembuatan web, karena semua class dan modul yang dibutuhkan sudah ada dan programmer hanya tinggal menggunakannya kembali pada aplikasi web yang akan dibuat [13].

6.1 Tutorial Install CodeIgniter 4

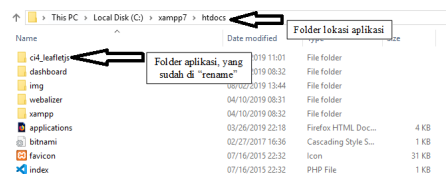
1. Kunjungi Link Resmi CodeIgniter di "<https://www.codeigniter.com/download>", kemudian pilih menu "View CodeIgniter 4 on Github".



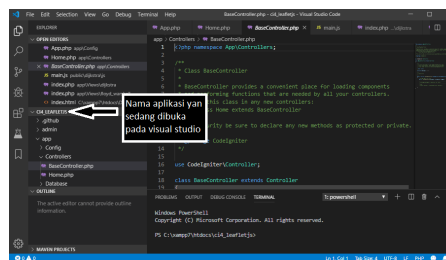
2. Anda akan dibawa ke web github, dimana terdapat repository resmi untuk pengembangan Framework CodeIgniter 4, klik Clone or download kemudian Download ZIP untuk melakukan download Framework CodeIgniter 4.



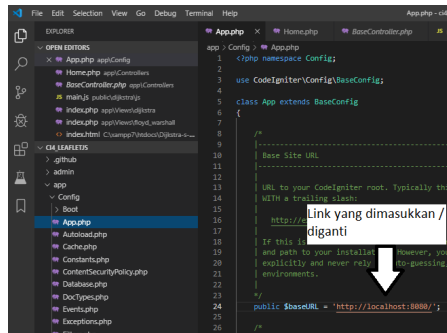
3. Tunggu proses download sampai selesai, kemudian buka lokasi file yang didownload tadi dan copy file CodeIgniter4 ke dalam folder Xampp yang sudah di install sebelumnya di C : \xampp7\htdocs.
4. Extract file di folder tersebut, kemudian rename foldernya dengan nama aplikasi yang akan anda bangun, disini saya melakukan penamaan foldernya menjadi ci4_leafletjs.



5. Buka aplikasi ci4_leafletjs dengan editor kesayangan anda, disini saya menggunakan visual studio code. Cara instal dan download bisa mengunjungi link berikut: "<https://code.visualstudio.com/download>".

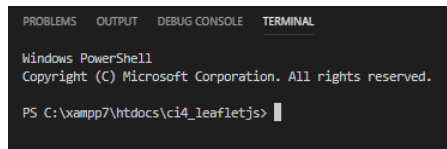


6. Buka file App.php pada folder "app/Config/App.php", kemudian pada bagian "public \$baseUrl = ''" diganti menjadi link aplikasi, sehingga seperti ini "public \$baseUrl = 'http://localhost:8080/' ;".

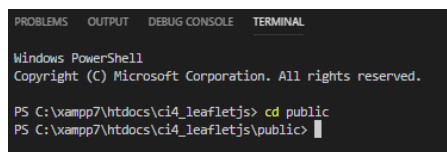


7. CodeIgniter 4 sudah siap digunakan, jalankan aplikasi tersebut dengan cara berikut:

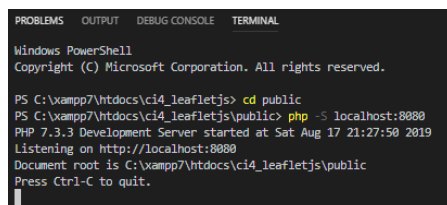
- (a) Klik menu Terminal yang ada pada menu bar atas Visual Studio Code, kemudian klik New Terminal atau bisa dengan cara menekan "Ctrl + Shift + ``" pada keyboard anda. Setelah itu akan muncul terminal dibagian bawah pada Visual Studio Code seperti pada gambar berikut:



- (b) Masuk ke dalam folder public dengan cara mengetikkan "cd public" pada terminal.



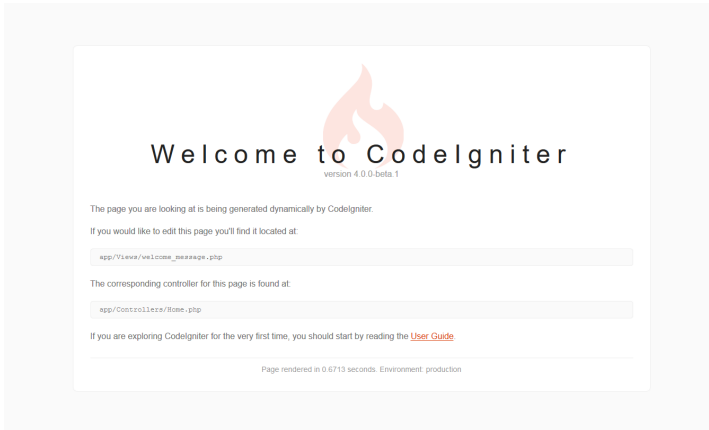
- (c) Ketikkan "php -S localhost:8080" pada terminal sehingga terlihat seperti pada gambar berikut:



- (d) Bisa juga menggunakan cara kedua yaitu dengan mengetikkan "php spark serve" pada terminal sehingga terlihat seperti pada gambar berikut:

```
PS C:\xampp7\htdocs\ci4-tracking> php spark serve
CodeIgniter CLI Tool - Version 4.0.0-beta.1 - Server-Time: 2019-07-06 05:22:27am
CodeIgniter development server started on http://localhost:8080
Press Control-C to stop.
```

- (e) Jalankan aplikasi `ci4_leaflet.js` anda pada browser kesayangan anda, dengan cara mengetikkan "localhost:8080".



- (f) Kenapa "localhost:8080"? karena CodeIgniter 4 dilengkapi dengan server pengembangan lokal, memanfaatkan server web bawaan PHP dengan perutean CodeIgniter. Untuk info lebih jelasnya dapat dilihat pada link "<https://codeigniter4.github.io/userguide/installation/running.html>".

6.2 Konfigurasi CodeIgniter 4

Di dalam folder config pada CodeIgniter terdapat berbagai macam file konfigurasi yang dapat kita atur sendiri nantinya. File tersebut dapat ditemukan pada folder `C:/xampp7/htdocs/ci4_leaflet.js/app/Config/`. Untuk codeigniter 4 default konfigurasi bisa dilakukan pada 3 file yaitu, file `App.php`, `Database.php` dan `Routes.php`. Berikut cara konfigurasinya:

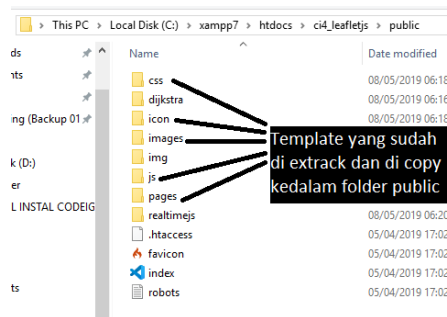
1. **App.php**, digunakan untuk membuat pengaturan dasar untuk web app codeigniter anda, seperti `base_url`, index page, cookie, proxy dan lain lain. Konfigurasi pada file ini dapat dilakukan sama seperti pada konfigurasi sebelumnya di section 2.1
2. **Database.php**, digunakan untuk mengatur koneksi web app kita ke database. Pada database.php konfigurasi yang dilakukan untuk mengkoneksikan database yaitu MySQL dengan aplikasi web berbasis framework CodeIgniter.

3. **Routes.php**, digunakan untuk mengatur default controller dan override 404.

6.3 Konfigurasi Template CodeIgniter 4

Ada berbagai macam konfigurasi template terhadap CodeIgniter, baik secara install maupun dengan cara konfigurasi sendiri. Pada tutorial kali ini saya ingin menerapkan bootstrap dan template di CodeIgniter dengan cara cepat. Untuk yang ingin menggunakan cara instan, bisa dengan cara mengunjungi website *w3layout.com* dan website yang menyediakan assets template dan bootstrap siap pakai. Sedikit berbeda dengan codeigniter 3 yang dimana harus dibuatkan folder assets terlebih dahulu, untuk codeigniter 4 semua assets akan ditampung dalam satu folder yaitu folder public, dapat di temukan pada folder `C:\xampp7\htdocs\ci4_leafletjs\public`, berikut cara konfigurasi template pada codeigniter 4:

1. Siapkan template atau bootstrap yang sudah didownload
2. Ekstrak file tersebut jika dalam bentuk .rar atau .zip
3. Copy file hasil ekstrak tadi ke dalam folder public.



4. Setelah menkopi file kedalam folder public, langkah selanjutnya adalah memanggil config tersebut. Jangan lupa untuk membuat header dan footer ketika membuat website guna untuk mempermudah apabila terjadi perubahan terhadap beberapa menu.
5. Copy isi dari index.html yang ada dalam template kemudian buat file di dalam folder `C:/xampp7/htdocs/ci4_leafletjs/app/views/agoritma/index.php` dengan format .php dan pastekan dalam file tersebut.
6. Pertama lakukan konfigurasi terhadap index.php dengan cara memanggil link dan script yang sudah di copy di dalam folder public, berikut contoh pemanggilanya:

```
1 <!DOCTYPE html>
2 <html lang="en">
```

```

3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-
7         scale=1.0, user-scalable=0, minimal-ui">
8     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
9     <meta name="description" content="CodedThemes">
10    <meta name="keywords" content="Admin , Responsive, Landing ,
11        Bootstrap , App, Template, Mobile, iOS, Android, apple ,
12        creative app">
13    <meta name="author" content="CodedThemes">
14
15    <title>Dijkstra's Algorithm and Floyd-Warshall Algorithm </
16        title>
17
18    <link rel="icon" type="image/png" href="/images/favicon-32x32
19        .png" sizes="32x32" />
20    <link rel="icon" type="image/png" href="/images/favicon-16x16
21        .png" sizes="16x16" />
22    <link rel="icon" href="/images/favicon.ico" type="image/x-
23        icon">
24    <link rel="stylesheet" type="text/css" href="/css/bootstrap /
25        css / bootstrap.min.css">
26    <link rel="stylesheet" type="text/css" href="/css/datatables .
27        css">
28    <link rel="stylesheet" type="text/css" href="/css/buttons .
29        dataTables.min.css">
30    <link rel="stylesheet" type="text/css" href="/css/select2.css
31        ">
32    <link rel="stylesheet" type="text/css" href="/css/style.css">
33    <link rel="stylesheet" type="text/css" href="/css/style-bulog
34        .css">
35    <link rel="stylesheet" type="text/css" href="/css/style-bulog
36        -print.css" media="print">
37    <link rel="stylesheet" type="text/css" href="/css/jquery .
38        mCustomScrollbar.css">
39    <script type="text/javascript" src="/js/jquery/jquery.min.js"
40        ></script>
41 </head>
42 <body>
43     ....
44     <div class="pcoded-content">
45         <div class="pcoded-inner-content">
46             <div class="main-body">
47                 <div class="page-wrapper">
48                     <div class="page-header card">
49                         <div class="row align-items-start">
50                             <div class="col-lg-8">
51                                 <div class="page-header-title">
52                                     &copy; Faisal Syarifuddin <?php echo date('Y
53                                     ') ?>
54                                 <br>CodeIgniter Version <?php echo CodeIgniter\
55                                     CodeIgniter::CI_VERSION ?>
56                             </div>
57                         </div>
58                     </div>
59                 </div>
60             </div>
61         </div>
62     </div>

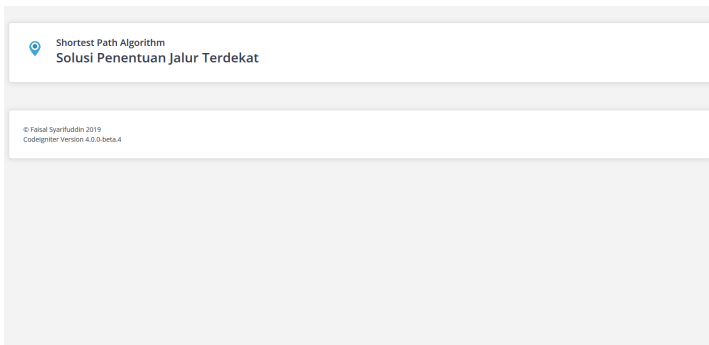
```

```
42         </div>
43     </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 <script type="text/javascript" src="/js/script.js"></script>
49 </body>
50 </html>
```

Listing 6.1 Konfigurasi Template di CodeIgniter 4

Lakukan pemanggilan terhadap semua code yang berbaw href dan src, seperti pada codingan diatas. Setelah itu simpan.

7. Pemanggilan template sudah selesai, jalankan aplikasi seperti pada cara yang sudah diterapkan sebelumnya di subbab 6.1, maka akan muncul tampilan seperti gambar berikut:



BAB 7

LEAFLET JS

Leaflet adalah pustaka JavaScript open-source terkemuka untuk peta interaktif ramah-mobile. Dengan berat hanya sekitar 38 KB JS, ia memiliki semua fitur pemetaan yang paling dibutuhkan pengembang. Leaflet dirancang dengan kesederhanaan, kinerja, dan kegunaan dalam pikiran. Ini bekerja secara efisien di semua platform desktop dan seluler utama, dapat diperluas dengan banyak plugin, memiliki API yang indah, mudah digunakan dan didokumentasikan dengan baik dan kode sumber yang mudah dibaca dan menyenangkan untuk berkontribusi [14].

7.1 Cara Menampilkan Map Leaflet JS di CodeIgniter

1. Buka salah satu file php yang ingin Anda berikan fitur map, file dapat ditemukan ada pada folder `C:/xampp7/htdocs/ci4_leafletjs/app/Views/`.
2. Pada bagian `<head>` sertakan file Leaflet CSS.

```
1 <head>
2   <meta charset="utf-8">
3   <meta name="viewport" content="width=device-width, initial-
    scale=1.0, user-scalable=0, minimal-ui">
```

```

4   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
5   <meta name="description" content="CodedThemes">
6   <meta name="keywords" content=" Admin , Responsive , Landing ,
    Bootstrap , App, Template , Mobile , iOS, Android, apple ,
    creative app">
7   <meta name="author" content="CodedThemes">
8
9   <link rel="stylesheet" href="https://unpkg.com/leaflet@1.5.1/
    dist/leaflet.css" integrity="sha512-xwE/
    Az9zrjBIphAcBb3F6JVqxf46+
    CDLwLMHloNu6KEQCAWi6HcDUbeOfBIptF7tcCzusKFjFw2yuvEpDL9wQ=="
    crossorigin=""/>
10 </head>
11 .....

```

Listing 7.1 File Leaflet CSS

3. Kemudian sertakan file JavaScript Leaflet setelah CSS Leaflet, pada bagian footer atau sebelum penutup `</body>`.

```

1   .....
2   <script src="https://unpkg.com/leaflet@1.5.1/dist/leaflet.js"
    integrity="sha512-GffPMF3RvMeYyc1LWMHtK8EbPv0iNZ8/oTtHPx9/
    cc2ILxQ+u905qlwduLQdKykBkgOaB57QTMg7ztg8Jm2Og==" crossorigin=
    ""></script>
3
4   </body>
5   </html>

```

Listing 7.2 File JavaScript Leaflet

4. Map sudah siap untuk ditampilkan, selanjutnya adalah meletakkan elemen `div` dengan id tertentu di mana Anda ingin peta Anda berada.

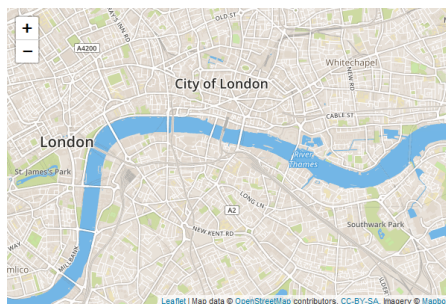
```

1   .....
2   <div id="mapid"></div>
3   .....

```

Listing 7.3 Elemen `div` map Leaflet

5. Jalankan aplikasi kemudian akan tampil map seperti berikut:



7.2 Menampilkan Mapbox Pada index.php

Pada bab sebelumnya sudah dilakukan konfigurasi beberapa template untuk mepercantik tampilan index.php. Kemudian selanjutnya adalah bagaimana cara memasang mapbox pada index.php? Berikut caranya:

1. Tambahkan config pada bagian head dan footer index.php dapat dilihat seperti pada codingan dibawah ini:

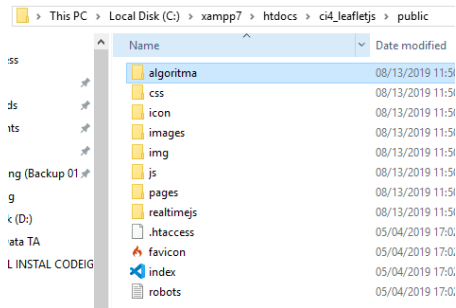
```

1  ....
2  <head>
3      <link href="/ algoritma / style / justified -nav .css" rel="
        stylesheet">
4      <link href="https :// fonts . googleapis . com / css ? family = Open + Sans
        :400 ,600" rel="stylesheet">
5      <link rel="stylesheet" href="https :// unpkg . com / leaflet @ 1 . 3 . 1 /
        dist / leaflet . css" integrity="sha512-
        Rksm5RenBEKSKFjgI3a4Ivrjkw4EVPIJ3+
        OiI65vTjIdo9briAacEuKOiQ5OFh7cOI1bkDwLqdLw3Zg0cRJAAQ=="
        crossorigin="" />
6  </head>
7  ....
8      <script src="https :// code . jquery . com / jquery -3.2.1.min . js"></
        script>
9      <script src="https :// cdnjs . cloudflare . com / ajax / libs / popper . js
        / 1.11.0/umd/popper.min . js" integrity="sha384-b/U6ypiBEHpOf
        /4+1nzFpr53nxSS+GLCKfwBdFNTxtclqqenISfwAzpKaMNFNmj4"
        crossorigin="anonymous"></script>
10     <script src="https :// maxcdn . bootstrapcdn . com / bootstrap / 4.0.0 -
        beta / js / bootstrap . min . js" integrity="sha384-
        h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3SEJAg45jXnGIfYzk4Si90RDIq
        Nm1" crossorigin="anonymous"></script>
11     <script src="https :// cdn . jsdelivr . net / npm / d3@3.3.0 / d3 . min . js"
        ></script>
12
13     <script src="https :// cdn . jsdelivr . net / npm / leaflet @ 0 . 7 . 7 / dist /
        leaflet . js"></script>
14
15     <script src="/ algoritma / js / main . js"></script>
16     <script src="/ algoritma / js / ie10-viewport-bug-workaround . js"></
        script>
17     ....

```

Listing 7.4 Menampilkan Mapbox Leaflet JS di CodeIgniter 4

2. Kemudian pada folder C : \xampp7\htdocs\ci4_leafletjs\public, buat satu folder dengan nama algoritma dimana dalam folder ini akan di masukkan beberapa js tentang algoritma dijkstra dan floyd warshall dan js tentang package leaflet.



3. Buka folder `algorithma` kemudian buat folder dengan nama `js` dan buat file di dalam folder `js` dengan nama `main.js`. Dalam file ini nantinya tersimpan beberapa konfigurasi seperti konfigurasi untuk `leaflet js` dan `algorithma` yang akan digunakan dalam penentuan rute terdekat.

4. Buka file `main.js` kemudian isikan kodingan berikut:

```

1  var graphmap, svg, maps, g;
2
3  var mapdata = {
4      allnodes: [],
5      paths: [],
6      distances: [],
7      getui: {
8          htmlSelectStartingNode: "#from-starting",
9          htmlSelectEndNode: "#to-end"
10     },
11     getstate: {
12         selectedNode: null,
13         fromNode: null,
14         toNode: null
15     }
16 };
17
18 maps = L.map('svg-map').setView([-6.8731953,107.5737873,17], 15);
19 //Menampilkan lokasi maps politeknik pos indonesia dengan max
    zoom 15 saat pertama kali load aplikasi
20 mapLink = '<a href="http://openstreetmap.org">OpenStreetMap</a>';
21 L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png
    ?access_token={accessToken}', {
22     attribution: 'Map data &copy; <a href="http://openstreetmap.
    org">OpenStreetMap</a> contributors, <a href="http://
    creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>,
    Imagery <a href="http://mapbox.com">Mapbox</a>',
23     maxZoom: 18,
24     id: 'mapbox.streets',
25     accessToken: 'pk.
    eyJ1IjoiaWlhZHVkZGluaGFyaXMiLCJhIjoieY2p1d2E3MzM4MGFiZDRkcGYyM
    WF3emtlYiJ9.KTemDE4IAujR0X-lttotg '
26 }).addTo(maps);
27 maps._initPathRoot()
28 svg = d3.select("#svg-map").select("svg")

```

```

29 .attr("class", "svgmap")
30 .on("contextmenu", function () { d3.event.preventDefault();
    })

```

Listing 7.5 Setting setView Pada Leaflet JS

5. Simpan, kemudian panggil library js dalam index.php

```

1 <script src="/algoritma/js/main.js"></script>

```

Listing 7.6 Call Library setView

6. Berikut codingan full untuk tampilan index.php

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-
7         scale=1.0, user-scalable=0, minimal-ui">
8     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
9     <meta name="description" content="CodedThemes">
10    <meta name="keywords" content=" Admin , Responsive , Landing ,
11        Bootstrap , App , Template , Mobile , iOS , Android , apple ,
12        creative app">
13    <meta name="author" content="CodedThemes">
14    <title >Shortest Path Algorithm </title >
15
16    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
17        bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-
18        -/Y6pD6FV/Vv2HJnA6t+
19        vslU6fwYXjCFTcEpHbNJ0lyAFsXTsjBbfaDjzALEQsN6M" crossorigin="
20        anonymous">
21    <link href="/algoritma/style/justified-nav.css" rel="stylesheet
22        ">
23
24    <link rel="icon" type="image/png" href="/images/favicon-32x32
25        .png" sizes="32x32" />
26    <link rel="icon" type="image/png" href="/images/favicon-16x16
27        .png" sizes="16x16" />
28    <link rel="icon" href="/images/favicon.ico" type="image/x-
29        icon">
30    <link href="https://fonts.googleapis.com/css?family=Open+Sans
31        :400,600" rel="stylesheet">
32    <link rel="stylesheet" type="text/css" href="/css/bootstrap /
33        css/bootstrap.min.css">
34    <link rel="stylesheet" type="text/css" href="/icon/themify-
35        icons/themify-icons.css">
36    <link rel="stylesheet" type="text/css" href="/icon/icofont /
37        css/icofont.css">
38    <link rel="stylesheet" type="text/css" href="/css/datatables .
39        css">
40    <link rel="stylesheet" type="text/css" href="/css/buttons .
41        dataTables.min.css">
42    <link rel="stylesheet" type="text/css" href="/css/select2.css
43        ">

```

```

26 <link rel="stylesheet" type="text/css" href="/css/style.css">
27 <link rel="stylesheet" type="text/css" href="/css/style-bulog
    .css">
28 <link rel="stylesheet" type="text/css" href="/css/style-bulog
    -print.css" media="print">
29 <link rel="stylesheet" type="text/css" href="/css/jquery.
    mCustomScrollbar.css">
30 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/
    ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
31
32 <script type="text/javascript" src="/js/jquery/jquery.min.js"
    ></script>
33 <link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.1/
    dist/leaflet.css" integrity="sha512-
    Rksm5RenBEKSKFjgI3a4Ivrjkw4EVPIJ3+
    Oil65vTjIdo9briAacEuKOiQ5OFh7cOI1bkDwLqDLw3Zg0cRJAAQ=="
    crossorigin=""/>
34 </head>
35 <body>
36 <div class="theme-loader">
37 <div class="ball-scale">
38 
39 </div>
40 </div>
41
42 <div class="loading-wrap">
43 <div class="loading-box">
44 
45 <span>Mohon tunggu...</span>
46 <div class="progress" style="height: 10px;">
47 <div id="progress-bar" class="progress-bar" role="
    progressbar"></div>
48 </div>
49 </div>
50 </div>
51
52 <div class="pcoded-content">
53 <div class="pcoded-inner-content">
54 <div class="main-body">
55 <div class="page-wrapper">
56 <div class="page-header card">
57 <div class="row align-items-start">
58 <div class="col-lg-12">
59 <div class="page-header-title">
60 <div class="icon-logo">
61 
62 </div>
63 <div class="d-inline">
64 <h4 class="m-b-15 block">Shortest Path
    Algorithm</h4>
65 <h3 class="m-t-5"><strong>Solusi Penentuan
    Jalur Terdekat</strong></h3>
66 </div>
67 </div>
68 </div>
69 </div>

```

```

70         </div>
71     <br />
72 </div>
73 </div>
74 </div>
75 </div>
76
77 <div class="row">
78     <div class="col-lg-12">
79         <div id="svg-map" style="width: 1110px; height: 800px"
80             class="card">
81         </div>
82     </div>
83
84     <div class="pcoded-content">
85         <div class="pcoded-inner-content">
86             <div class="main-body">
87                 <div class="page-wrapper">
88                     <div class="page-header card">
89                         <div class="row align-items-start">
90                             <div class="col-lg-8">
91                                 <div class="page-header-title">
92                                     &copy; Faisal Syarifuddin <?php echo date('Y
93                                     <br>CodeIgniter Version <?= CodeIgniter\
94                                     CodeIgniter::CLVERSION ?>
95                                 </div>
96                             </div>
97                         </div>
98                     </div>
99                 </div>
100             </div>
101         </div>
102     </div>
103
104 <script src="https://code.jquery.com/jquery-3.2.1.min.js"></
105     script>
106 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js
107     /1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf
108     /4+1nzFpr53nxSS+GLCKfwBFNTxtclqenISfwAzpKaMNFNmj4"
109     crossorigin="anonymous"></script>
110 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
111     beta/js/bootstrap.min.js" integrity="sha384-
112     h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGL03SEJAg45jXnGIfYzk4Si90RDIq
    Nml" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/d3@3.3.0/d3.min.js"
    ></script>
    <script src="https://cdn.jsdelivr.net/npm/leaflet@0.7.7/dist/
    leaflet.js"></script>
    <script src="/algoritma/js/main.js"></script>
    <script src="/algoritma/js/ie10-viewport-bug-workaround.js"></
    script>
    <script type="text/javascript" src="/js/script.js"></script>

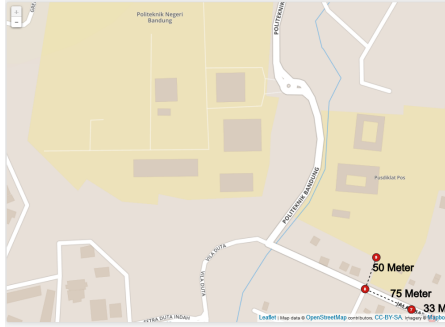
```

113

114 </html>

Listing 7.7 Pemanggilan Library setView pada index.php

7. Simpan file `index.php` tersebut kemudian jalankan aplikasi anda, maka pada browser akan muncul map box seperti pada gambar berikut.



7.3 Penentuan Jalur Terdekat Dengan Algoritma Dijkstra dan Floyd Warshal

Melanjutnkan Kodingan sebelumnya, pada tahap ini anda akan membuat UI, dan beberapa function yang untuk algoritma dijsktra dan floyd warshal. Berikut tahapan membuat sistem penentuan jalur terdekat menggunakan algoritma dijsktra dan floyd warshall:

1. Membuat index.php pada folder algoritma (Sebelumnya sudah di buat pada point 7.2).
2. Melengkapi library yang ada pada index.php pada point 7.2, seperti pada kodin-gan dibawah:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-
7       scale=1.0, user-scalable=0, minimal-ui">
8     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
9     <meta name="description" content="CodedThemes">
10    <meta name="keywords" content="Admin , Responsive, Landing,
11      Bootstrap, App, Template, Mobile, iOS, Android, apple,
12      creative app">
13    <meta name="author" content="CodedThemes">
14
15 </head>
16
17 <body>
18
19 <div class="page-wrapper">
20
21 <div class="main">
22
23 <div class="row">
24
25 <div class="col">
26
27 <div class="card">
28
29 <div class="card-body">
30
31 <div class="row">
32
33 <div class="col">
34
35 <div class="row">
36
37 <div class="col">
38
39 <div class="row">
40
41 <div class="col">
42
43 <div class="row">
44
45 <div class="col">
46
47 <div class="row">
48
49 <div class="col">
50
51 <div class="row">
52
53 <div class="col">
54
55 <div class="row">
56
57 <div class="col">
58
59 <div class="row">
60
61 <div class="col">
62
63 <div class="row">
64
65 <div class="col">
66
67 <div class="row">
68
69 <div class="col">
70
71 <div class="row">
72
73 <div class="col">
74
75 <div class="row">
76
77 <div class="col">
78
79 <div class="row">
80
81 <div class="col">
82
83 <div class="row">
84
85 <div class="col">
86
87 <div class="row">
88
89 <div class="col">
90
91 <div class="row">
92
93 <div class="col">
94
95 <div class="row">
96
97 <div class="col">
98
99 <div class="row">
100
101 <div class="col">
102
103 <div class="row">
104
105 <div class="col">
106
107 <div class="row">
108
109 <div class="col">
110
111 <div class="row">
112
113 <div class="col">
114
115 <div class="row">
116
117 <div class="col">
118
119 <div class="row">
120
121 <div class="col">
122
123 <div class="row">
124
125 <div class="col">
126
127 <div class="row">
128
129 <div class="col">
130
131 <div class="row">
132
133 <div class="col">
134
135 <div class="row">
136
137 <div class="col">
138
139 <div class="row">
140
141 <div class="col">
142
143 <div class="row">
144
145 <div class="col">
146
147 <div class="row">
148
149 <div class="col">
150
151 <div class="row">
152
153 <div class="col">
154
155 <div class="row">
156
157 <div class="col">
158
159 <div class="row">
160
161 <div class="col">
162
163 <div class="row">
164
165 <div class="col">
166
167 <div class="row">
168
169 <div class="col">
170
171 <div class="row">
172
173 <div class="col">
174
175 <div class="row">
176
177 <div class="col">
178
179 <div class="row">
180
181 <div class="col">
182
183 <div class="row">
184
185 <div class="col">
186
187 <div class="row">
188
189 <div class="col">
190
191 <div class="row">
192
193 <div class="col">
194
195 <div class="row">
196
197 <div class="col">
198
199 <div class="row">
200
201 <div class="col">
202
203 <div class="row">
204
205 <div class="col">
206
207 <div class="row">
208
209 <div class="col">
210
211 <div class="row">
212
213 <div class="col">
214
215 <div class="row">
216
217 <div class="col">
218
219 <div class="row">
220
221 <div class="col">
222
223 <div class="row">
224
225 <div class="col">
226
227 <div class="row">
228
229 <div class="col">
230
231 <div class="row">
232
233 <div class="col">
234
235 <div class="row">
236
237 <div class="col">
238
239 <div class="row">
240
241 <div class="col">
242
243 <div class="row">
244
245 <div class="col">
246
247 <div class="row">
248
249 <div class="col">
250
251 <div class="row">
252
253 <div class="col">
254
255 <div class="row">
256
257 <div class="col">
258
259 <div class="row">
260
261 <div class="col">
262
263 <div class="row">
264
265 <div class="col">
266
267 <div class="row">
268
269 <div class="col">
270
271 <div class="row">
272
273 <div class="col">
274
275 <div class="row">
276
277 <div class="col">
278
279 <div class="row">
280
281 <div class="col">
282
283 <div class="row">
284
285 <div class="col">
286
287 <div class="row">
288
289 <div class="col">
290
291 <div class="row">
292
293 <div class="col">
294
295 <div class="row">
296
297 <div class="col">
298
299 <div class="row">
300
301 <div class="col">
302
303 <div class="row">
304
305 <div class="col">
306
307 <div class="row">
308
309 <div class="col">
310
311 <div class="row">
312
313 <div class="col">
314
315 <div class="row">
316
317 <div class="col">
318
319 <div class="row">
320
321 <div class="col">
322
323 <div class="row">
324
325 <div class="col">
326
327 <div class="row">
328
329 <div class="col">
330
331 <div class="row">
332
333 <div class="col">
334
335 <div class="row">
336
337 <div class="col">
338
339 <div class="row">
340
341 <div class="col">
342
343 <div class="row">
344
345 <div class="col">
346
347 <div class="row">
348
349 <div class="col">
350
351 <div class="row">
352
353 <div class="col">
354
355 <div class="row">
356
357 <div class="col">
358
359 <div class="row">
360
361 <div class="col">
362
363 <div class="row">
364
365 <div class="col">
366
367 <div class="row">
368
369 <div class="col">
370
371 <div class="row">
372
373 <div class="col">
374
375 <div class="row">
376
377 <div class="col">
378
379 <div class="row">
380
381 <div class="col">
382
383 <div class="row">
384
385 <div class="col">
386
387 <div class="row">
388
389 <div class="col">
390
391 <div class="row">
392
393 <div class="col">
394
395 <div class="row">
396
397 <div class="col">
398
399 <div class="row">
400
401 <div class="col">
402
403 <div class="row">
404
405 <div class="col">
406
407 <div class="row">
408
409 <div class="col">
410
411 <div class="row">
412
413 <div class="col">
414
415 <div class="row">
416
417 <div class="col">
418
419 <div class="row">
420
421 <div class="col">
422
423 <div class="row">
424
425 <div class="col">
426
427 <div class="row">
428
429 <div class="col">
430
431 <div class="row">
432
433 <div class="col">
434
435 <div class="row">
436
437 <div class="col">
438
439 <div class="row">
440
441 <div class="col">
442
443 <div class="row">
444
445 <div class="col">
446
447 <div class="row">
448
449 <div class="col">
450
451 <div class="row">
452
453 <div class="col">
454
455 <div class="row">
456
457 <div class="col
```

```

13 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
    bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-
    -/Y6pD6FV/Vv2HJnA6t+
    vsIU6fwYXjCFtcEpHbNJ0lyAFsXTsJBbfaDjzALeQsN6M" crossorigin="
    anonymous">
14 <link href="/algoritma/style/justified-nav.css" rel="stylesheet
    ">
15
16 <link rel="icon" type="image/png" href="/images/favicon-32x32
    .png" sizes="32x32" />
17 <link rel="icon" type="image/png" href="/images/favicon-16x16
    .png" sizes="16x16" />
18 <link rel="icon" href="/images/favicon.ico" type="image/x-
    icon">
19 <link href="https://fonts.googleapis.com/css?family=Open+Sans
    :400,600" rel="stylesheet">
20 <link rel="stylesheet" type="text/css" href="/css/bootstrap/
    css/bootstrap.min.css">
21 <link rel="stylesheet" type="text/css" href="/icon/themify-
    icons/themify-icons.css">
22 <link rel="stylesheet" type="text/css" href="/icon/icofont/
    css/icofont.css">
23 <link rel="stylesheet" type="text/css" href="/css/datatables.
    css">
24 <link rel="stylesheet" type="text/css" href="/css/buttons.
    dataTables.min.css">
25 <link rel="stylesheet" type="text/css" href="/css/select2.css
    ">
26 <link rel="stylesheet" type="text/css" href="/css/style.css">
27 <link rel="stylesheet" type="text/css" href="/css/style-bulog
    .css">
28 <link rel="stylesheet" type="text/css" href="/css/style-bulog
    -print.css" media="print">
29 <link rel="stylesheet" type="text/css" href="/css/jquery.
    mCustomScrollbar.css">
30 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/
    ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
31
32 <script type="text/javascript" src="/js/jquery/jquery.min.js"
    ></script>
33 <link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.1/
    dist/leaflet.css" integrity="sha512-
    RkSm5RenBEKSKFjgI3a4Ivrjkw4EVPIJ3+
    OiI65vTjIdo9briAacEuKOiQ5OFh7cOI1bkDwLqDLw3Zg0cRJAAQ=="
    crossorigin="">
34
35 </head>
36 <body>
37 <div class="theme-loader">
38 <div class="ball-scale">
39 
40 </div>
41 </div>
42
43 <div class="loading-wrap">
44 <div class="loading-box">

```

```

45     
46     <span>Mohon tunggu...</span>
47     <div class="progress" style="height: 10px;">
48         <div id="progress-bar" class="progress-bar" role="
progressbar"></div>
49     </div>
50 </div>
51 </div>
52
53 <div class="pcoded-content">
54     <div class="pcoded-inner-content">
55         <div class="main-body">
56             <div class="page-wrapper">
57                 <div class="page-header card">
58                     <div class="row align-items-start">
59                         <div class="col-lg-12">
60                             <div class="page-header-title">
61                                 <div class="icon-logo">
62                                     
63                                 </div>
64                                 <div class="d-inline">
65                                     <h4 class="m-b-15 block">Shortest Path
Algorithm </h4>
66                                     <h3 class="m-t-5"><strong>Solusi Penentuan
Jalur Terdekat </strong></h3>
67                                     </div>
68                                 </div>
69                             </div>
70                         </div>
71                     </div>
72                     <br />
73                 </div>
74             </div>
75         </div>
76     </div>
77
78 <div class="container">
79     <div class="row">
80         <div class="col-lg-12">
81             <label class="mr-sm-2" for="from">Cara penggunaan:
Tentukan titik-titik sesuai di maps dengan cara klik kiri ,
kemudian sambungkan titik satu dengan titik lainnya dengan
cara klik kanan titik awal dan klik kanan titik tujuan , pilih
route dari mana kemana di pilihan titik awal dan titik
tujuan kemudian klik start route untuk melihat hasil
algoritma sesuai tombol yg diklik.</label>
82
83             <hr>
84             <div class="form-row align-items-center">
85                 <div class="col-auto">
86                     <label class="mr-sm-2" for="from">Titik Awal : </
label>
87                     <select id="from-starting" class="custom-select mb
-2 mr-sm-2 mb-sm-0"></select>
88                 </div>
89

```

```

90         <div class="col-auto">
91             <label class="mr-sm-2" for="to">Titik Tujuan : </
label>
92             <select id="to-end" class="custom-select mb-2 mr-sm
-2 mb-sm-0"></select>
93         </div>
94
95         <div class="col-auto">
96             <button type="button" id="getshortestroute" class="
btn btn-primary" title="find shortest path between nodes
using dijkstra"><i class="fa fa-map-o" aria-hidden="true">
Start Dijkstra </i></button>
97         </div>
98
99         <div class="col-auto">
100             <button type="button" id="floyd" class="btn btn-
primary" title="find shortest path between nodes using floyd"
><i class="fa fa-map-o" aria-hidden="true"> Start Floyd </i></
button>
101         </div>
102
103         <div class="col-auto">
104             <button type="button" id="clearmap" class="btn btn-
danger"><i class="fa fa-trash-o" aria-hidden="true"> Hapus
Map</i></button>
105         </div>
106
107         <div class="col-12">
108             <br><br>
109             <h6>Jalur terpendek: <span id="jtp"></span></h6>
110         </div>
111     </div>
112
113 </div>
114 </div>
115
116 <div class="row">
117     <div class="col-lg-12">
118         <div id="svg-map" style="width: 1110px; height: 800px"
class="card">
119         </div>
120     </div>
121 </div>
122
123 <hr>
124 <form style="margin-top: 5px;">
125 <div class="form-row align-items-center">
126     <div class="col-auto">
127         <button type="button" class="btn btn-primary" id="data-
export"> Export Data </button>
128     </div>
129 </div>
130 </form>
131 <hr>
132
133 <div class="col-auto" style="margin-top: 5px;">

```



```

134 <label class="mr-sm-2" for="inlineFormCustomSelect">Route
    Wisata Bandung</label>
135 <select class="custom-select mb-2 mr-sm-2 mb-sm-0" id="
    setexample">
136 <option value="0"></option>
137 <option value="1">Titik Wisata Bandung</option>
138 <option value="2">Data Titik Lokasi Wisata Bandung</
    option>
139 <option value="3">Sample Data Politeknik Pos Indonesia</
    option>
140 <option value="4">Dusun Bambu Lembang</option>
141 <option value="5">Farmhouse Lembang</option>
142 <option value="6">Gunung Tangkuban Perhau</option>
143 <option value="7">Kebun Teh Sukawana</option>
144 <option value="9">Tebing Kraton</option>
145 <option value="10">Alun-Alun Bandung</option>
146 </select>
147 </div>
148 <hr>
149
150 <div class="pcoded-content">
151 <div class="pcoded-inner-content">
152 <div class="main-body">
153 <div class="page-wrapper">
154 <div class="page-header card">
155 <div class="row align-items-start">
156 <div class="col-lg-8">
157 <div class="page-header-title">
158 &copy; Faisal Syarifuddin <?php echo date('Y
    ') ?>
159 <br>CodeIgniter Version <?= CodeIgniter\
    CodeIgniter::CLVERSION ?>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168
169 </div>
170
171 <script src="https://code.jquery.com/jquery-3.2.1.min.js"></
    script>
172 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js
    /1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf
    /4+1nzFpr53nxSS+GLCKfwBdFNTxtclqqenISfwAzpKaMNFNmj4"
    crossorigin="anonymous"></script>
173 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
    beta/js/bootstrap.min.js" integrity="sha384-
    h0ABiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1
    " crossorigin="anonymous"></script>
174 <script src="https://cdn.jsdelivr.net/npm/d3@3.3.0/d3.min.js"
    ></script>

```

```

175 <script src="https://cdn.jsdelivr.net/npm/leaflet@0.7.7/dist/
    leaflet.js"></script>
176
177 <script src="/algoritma/js/main.js"></script>
178 <script src="/algoritma/js/ie10-viewport-bug-workaround.js"></
    script>
179 <script type="text/javascript" src="/js/script.js"></script>
180
181 </body>
182 </html>

```

Listing 7.8 index.php Full Code

3. Selanjutnya membuat function algoritma Dijkstra dan Floyd Warshall

(a) Function Algoritma Dijsktra

- i. Buka file main.js yang sudah dibuat sebelumnya pada point 7.2 atau bisa ditemukan pada folder ci4_leafletjs\ kemudian dalam folder public\algoritma\js\main.js.
- ii. Buat function klik dijsktra pada main.js

```

1 $( '#getshortestroute' ).on( 'click', function () {
2     alert ( ' Dijkstra ' );
3     d3.selectAll( "line" ).classed( { "shortest": false } )
4     ;
5     calculateDistancesbetweennodes ( ) ;
6     if ( !$( mapdata . getui . htmlSelectStartingNode ) . val ( )
7     || !$( mapdata . getui . htmlSelectEndNode ) . val ( ) )
8     return ;
9     var sourceNode = $( mapdata . getui .
10    htmlSelectStartingNode ) . val ( ) ;
11    var targetNode = $( mapdata . getui . htmlSelectEndNode )
12    . val ( ) ;
13    var results = dijkstra ( sourceNode , targetNode ) ;
14    var distTotal = 0 ;
15    var dist = null ;
16    var stepnode = ' ' ;
17    if ( results . path ) {
18        results . path . forEach ( function ( step ) {
19
20            dist = mapdata . distances [ step . source ] [ step .
21            target ]
22            stepLine = d3 . select (
23                "line . from" + step . source + "to" + step
24                . target + " , "
25                + "line . from" + step . target + "to" +
26                step . source
27            ) ;
28            stepLine . classed ( { "shortest": true } ) ;
29            stepnode += step . source + " -> " + step . target
30            + " -> " ;
31            distTotal = distTotal + dist ;
32        } ) ;
33        var estimasiMenit = distTotal / 60 ; //
34        perkiraan kecepatan 60 km / jam

```

```

25         // var estimasiDetik = estimasiMenit * 3600
        / 60;
26         // var estimasiDetik = estimasiMenit * 60 /
        3600;
27         var estimasiDetik = (estimasiMenit - Math.
floor(estimasiMenit)) * 60;
28         var persentasilocaccuracy = distTotal /
400;
29         if(persentasilocaccuracy <= 800){
30             persentasiloc = 'High';
31         }else if(persentasilocaccuracy >= 5000)
{
32             persentasiloc = 'Low';
33         }else{
34             persentasiloc = 'Medium';
35         }
36         $('#jtp').html('( Dijkstra ) '+Math.round(
distTotal)+' m <br><br>Melewati node : start ->'+
stepnode+'end<br><br>Perkiraan waktu perjalanan: '+
Math.round(estimasiMenit)+' menit, '+Math.round(
estimasiDetik)+' detik <br><br>Persentasi Akurasi: '+
persentasiloc+' ('+Math.round(persentasilocaccuracy
)+' %)');
37     }
38
39 });
40 $('#clearmap').on('click', function () {
41     clearGraph();
42 });

```

Listing 7.9 Function OnClick Dijkstra

- iii. Function ini akan di eksekusi ketika button pada index.php dengan id=getshortestroute di klik (button dijkstra).

```

1 <button type="button" id="getshortestroute" class="btn
btn-primary" title="find shortest path between
nodes using dijkstra">

```

Listing 7.10 Button Dijkstra

- iv. Selanjutnya membuat function dengan rumus dijkstra yang dihasilkan dari code listing pseudocode 2.2.1. Function ini akan digunakan untuk mengeksekusi beberapa titik yang digunakan untuk menentukan jalur terdekat.

```

1 function dijkstra(start, end) {
2     var nodeCount = mapdata.distances.length,
        infinity = 99999, // infinity
3     shortestPath = new Array(nodeCount),
4     nodeChecked = new Array(nodeCount),
5     pred = new Array(nodeCount);
6
7
8     for (var i = 0; i < nodeCount; i++) {
9         shortestPath[i] = infinity;
10        pred[i] = null;
11        nodeChecked[i] = false;

```

```

12     }
13
14     shortestPath[start] = 0;
15     for (var i = 0; i < nodeCount; i++) {
16         var minDist = infinity;
17         var closestNode = null;
18         for (var j = 0; j < nodeCount; j++) {
19             if (!nodeChecked[j]) {
20                 if (shortestPath[j] <= minDist) {
21                     minDist = shortestPath[j];
22                     closestNode = j;
23                 }
24             }
25         }
26
27         nodeChecked[closestNode] = true;
28         for (var k = 0; k < nodeCount; k++) {
29             if (!nodeChecked[k]) {
30                 var nextDistance = distanceBetween(
31                     closestNode, k, mapdata.distances);
32                 if ((parseInt(shortestPath[closestNode]) +
33                     parseInt(nextDistance)) < parseInt(
34                     shortestPath[k])) {
35                     soFar = parseInt(shortestPath[
36                         closestNode]);
37                     extra = parseInt(nextDistance);
38                     shortestPath[k] = soFar + extra;
39                     pred[k] = closestNode;
40                 }
41             }
42         }
43     }
44
45     if (shortestPath[end] < infinity) {
46         var newPath = [];
47         var step = {
48             target: parseInt(end)
49         };
50         var v = parseInt(end);
51
52         while (v >= 0) {
53             v = pred[v];
54             if (v !== null && v >= 0) {
55                 step.source = v;
56                 newPath.unshift(step);
57                 step = {
58                     target: v
59                 };
60             }
61         }
62
63         totalDistance = shortestPath[end];
64         return {
65             msg: 'Status: OK',
66             path: newPath,
67             source: start,

```

```

64         target: end,
65         distance: totalDistance
66     };
67 } else {
68     return {
69         mesg: 'Sorry No path found',
70         path: null,
71         source: start,
72         target: end,
73         distance: 0
74     };
75 }
76
77 function distanceBetween(fromNode, toNode,
78     distances) {
79     dist = distances[fromNode][toNode];
80     if (dist === 'x') dist = infinity;
81     return dist;
82 };

```

Listing 7.11 Function Algoritma Dijkstra

Function ini akan di eksekusi ketika button dijsktra diklik, kemudian akan memanggil function dijsktra pada variable "var results = dijsktra(sourceNode, targetNode);" di function onclick button dijsktra.

(b) Function Algoritma Floyd Warshall

i. Masih di file main.js, buat functionin klik floyd warshal pada main.js

```

1  $('#floyd').on('click', function () {
2      alert('Floyd Warshal');
3      d3.selectAll("line").classed({ "shortest": false });
4      ;
5      calculateDistancesbetweennodes();
6      if (!$ (mapdata.getui.htmlSelectStartingNode).val()
7      || !$ (mapdata.getui.htmlSelectEndNode).val())
8      return;
9      var sourceNode = $(mapdata.getui.htmlSelectStartingNode).val();
10     var targetNode = $(mapdata.getui.htmlSelectEndNode).val();
11     var results = floyd(sourceNode, targetNode);
12     var distTotal = 0;
13     var dist = null;
14     var stepnode = '';
15
16     if (results.path) {
17         results.path.forEach(function (step) {
18             dist = mapdata.distances[step.source][step.target]
19             stepLine = d3.select(
20                 "line.from" + step.source + "to" + step
21                 .target + ","

```

```

19         + "line.from" + step.target + "to" +
step.source
20     );
21     stepLine.classed({ "shortest": true });
22     stepnode += step.source+'->'+step.target
+'->';
23     distTotal = distTotal + dist;
24     });
25     var estimasiMenit = distTotal / 60; //
perkiraan kecepatan 60 km/jam
26     var estimasiDetik = (estimasiMenit - Math.
floor(estimasiMenit)) * 60;
27     var persentasilocaccuracy = distTotal /
400;
28     if(persentasilocaccuracy <= 800){
29         persentasiloc = 'High';
30     }else if(persentasilocaccuracy >= 5000)
{
31         persentasiloc = 'Low';
32     }else{
33         persentasiloc = 'Medium';
34     }
35     $('#jtp ').html('(Floyd Warshall) '+Math.
round(distTotal)+' m <br><br>Melewati node : start
->'+stepnode+'end<br><br>Perkiraan waktu perjalanan
: '+Math.round(estimasiMenit)+' menit, '+Math.round(
estimasiDetik)+' detik<br><br>Persentasi Akurasi:
'+persentasiloc+' ('+Math.round(
persentasilocaccuracy)+' %)');
36 }else{
37     results.path.forEach(function (step) {
38
39         dist = mapdata.distances[step.source][step.
target]
40         stepLine = d3.select(
41             "line.from" + step.source + "to" + step
.target + ", "
42             + "line.from" + step.target + "to" +
step.source
43         );
44         stepLine.classed({ "shortest": true });
45         stepnode += step.source+'->'+step.target
+'->';
46         distTotal = distTotal + dist;
47         });
48         var estimasiMenit = distTotal / 60; //
perkiraan kecepatan 60 km/jam
49         var estimasiDetik = (estimasiMenit - Math.
floor(estimasiMenit)) * 60;
50         var persentasilocaccuracy = distTotal /
400;
51         if(persentasilocaccuracy <= 800){
52             persentasiloc = 'High';
53         }else if(persentasilocaccuracy >= 5000)
{
54             persentasiloc = 'Low';

```

```

55         } else {
56             persentasiloc = 'Medium';
57         }
58         $('#jtp').html(Math.round(distTotal)+' m <
br>Melewati node : start->'+stepnode+'end<br>
br>Perkiraan waktu perjalanan: '+Math.round(
estimasiMenit)+' menit, '+Math.round(estimasiDetik)
+'detik<br>Persentasi Akurasi: '+persentasiloc
+' ('+Math.round(persentasilocaccuarcy)+' %)');
59     }
60
61 });
62 $('#clearmap').on('click', function () {
63     clearGraph();
64 });

```

Listing 7.12 Function OnClick Dijkstra

- ii. Function ini akan di eksekusi ketika button pada index.php dengan id=floyd di klik (button dijsktra)

```

1 <button type="button" id="floyd" class="btn btn-primary
" title="find shortest path between nodes using
floyd"><i class="fa fa-map-o" aria-hidden="true">
Start Floyd </i></button>

```

Listing 7.13 Button Dijkstra

- iii. Selanjutnya membuat function dengan rumus floyd warshal yang dihasilkan dari code listing psoudocode 4. Function ini akan digunakan untuk mengeksekusi beberapa titik yang digunakan untuk menentukan jalur terdekat.

```

1 function floyd(start,end){
2     console.log('start: '+start);
3     console.log('end: '+end);
4     var nodeCount = mapdata.distances.length,
5         infinity = 99999, // infinity
6         shortestPath = new Array(nodeCount),
7         nodeChecked = new Array(nodeCount),
8         pred = new Array(nodeCount);
9     console.log('nodecount: '+nodeCount);
10
11     for (var i = 0; i < nodeCount; i++) {
12         shortestPath[i] = infinity;
13         pred[i] = null;
14         nodeChecked[i] = false;
15     }
16     shortestPath[start] = 0;
17
18     for (var i = 0; i < nodeCount; i++) {
19         var minDist = infinity;
20         var closestNode = null;
21         for (var j = 0; j < nodeCount; j++) {
22             if (!nodeChecked[j]) {
23                 if (shortestPath[j] <= minDist) {
24                     minDist = shortestPath[j];

```

```

25         closestNode = j;
26     }
27 }
28 }
29 nodeChecked[closestNode] = true;
30
31 if(start > end){
32     for (var k = closestNode; k > 0; k--) {
33         if (!nodeChecked[k]) {
34             var nextDistance = distanceBetween(
35                 closestNode, k, mapdata.distances);
36             if ((parseInt(shortestPath[
37                 closestNode]) + parseInt(nextDistance)) < parseInt(
38                 shortestPath[k])) {
39                 soFar = parseInt(shortestPath[
40                     closestNode]);
41                 extra = parseInt(nextDistance);
42                 shortestPath[k] = soFar + extra
43 ;
44                 pred[k] = closestNode;
45             }
46         }
47     }
48 } else {
49     for (var k = closestNode; k < nodeCount; k
50 ++){
51         if (!nodeChecked[k]) {
52             var nextDistance = distanceBetween(
53                 closestNode, k, mapdata.distances);
54             if ((parseInt(shortestPath[
55                 closestNode]) + parseInt(nextDistance)) < parseInt(
56                 shortestPath[k])) {
57                 soFar = parseInt(shortestPath[
58                     closestNode]);
59                 extra = parseInt(nextDistance);
60                 shortestPath[k] = soFar + extra
61 ;
62                 pred[k] = closestNode;
63             }
64         }
65     }
66 }
67 console.log('shortestPath[end]: ' + shortestPath[end
68 ]);
69 if (shortestPath[end] < infinity) {
70     var newPath = [];
71     var step = {
72         target: parseInt(end)
73     };
74
75     var v = parseInt(end);
76     while (v >= 0) {
77         v = pred[v];
78         if (v !== null && v >= 0) {
79             step.source = v;

```



```

69         newPath.unshift(step);
70         step = {
71             target: v
72         };
73     }
74 }
75 totalDistance = shortestPath[end];
76
77 return {
78     msg: 'Status: OK',
79     path: newPath,
80     source: start,
81     target: end,
82     distance: totalDistance
83 };
84 } else {
85     return {
86         msg: 'Sorry No path found',
87         path: null,
88         source: start,
89         target: end,
90         distance: 0
91     };
92 }
93
94 function distanceBetween(fromNode, toNode,
95     distances) {
96     dist = distances[fromNode][toNode];
97     if (dist === 'x') dist = infinity;
98     return dist;
99 }

```

Listing 7.14 Function Algoritma Dijkstra

Function ini akan di eksekusi ketika button floyd warshall diklik, kemudian akan memanggil function floyd pada variable "var results = floyd(sourceNode, targetNode);" di function onclick button dijkstra.

4. Membuat function penanda saat klik maps, ketika melakukan pemilihan lokasi maka mark yang akan muncul pada maps adalah mark dengan bentuk circle merah, berikut kodingan function:

```

1 function redrawNodes() {
2     svg.selectAll("g.nodes").data([]).exit().remove();
3     var elements = svg.selectAll("g.nodes").data(mapdata.allnodes
4     , function (d, i) { return d.name; });
5     var nodesEnter = elements.enter().append("g")
6     .attr("class", "nodes");
7
7     elements.attr("transform", function (d, i) {
8         return "translate(" +
9             maps.latLngToLayerPoint(new L.LatLng(d.x, d.y)).x + "
, " +

```

```

10         maps.latLngToLayerPoint(new L.LatLng(d.x, d.y)).y + "
11     );
12     });
13     nodesEnter.append("circle")
14         .attr("nodeId", function (d, i) { return i; })
15         .attr("r", '20')
16         .attr("class", "node")
17         .style("cursor", "pointer")
18         .on('click', nodeClick)
19         .on("mouseenter", function () { maps.dragging.disable();
20     })
21         .on("mouseout", function () { maps.dragging.enable(); })
22         .on('contextmenu', function (d, i) { startEndPath(i); })
23         .call(dragManager)
24     nodesEnter
25         .append("text")
26         .attr("nodeLabelId", function (d, i) { return i; })
27         .attr("dx", "-3")
28         .attr("dy", "3")
29         .attr("class", "label")
30         .on('contextmenu', function (d, i) { startEndPath(i); })
31         .call(dragManager)
32         .text(function (d, i) { return d.name });
33     elements.exit().remove();
34 };

```

Listing 7.15 Function Mark Maps

5. Membuat item line, penghubung antara titik satu dengan titik lainnya, fungsi ini akan menghubungkan node marks maps yang kemudian di eksekusi pada masing algoritma.

```

1  function redrawLines() {
2      svg.selectAll("g.line").data([]).exit().remove();
3      var elements = svg
4          .selectAll("g.line")
5          .data(mapdata.paths, function (d) { return d.id });
6      var newElements = elements.enter();
7
8      var group = newElements
9          .append("g")
10         .attr("class", "line");
11
12     var line = group.append("line")
13         .attr("class", function (d) {
14             return "from" + mapdata.allnodes[d.from].name + "to"
15             + mapdata.allnodes[d.to].name
16         })
17         .attr("x1", function (d) { return maps.latLngToLayerPoint
18             (new L.LatLng(mapdata.allnodes[d.from].x, mapdata.allnodes[d.
19             from].y)).x; })
20         .attr("y1", function (d) { return maps.latLngToLayerPoint
21             (new L.LatLng(mapdata.allnodes[d.from].x, mapdata.allnodes[d.
22             from].y)).y; })

```

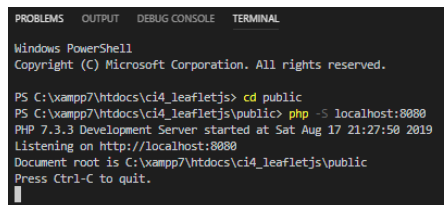
```

18     .attr("x2", function (d) { return maps.latLngToLayerPoint
    (new L.LatLng(mapdata.allnodes[d.to].x, mapdata.allnodes[d.to
    ].y)).x; })
19     .attr("y2", function (d) { return maps.latLngToLayerPoint
    (new L.LatLng(mapdata.allnodes[d.to].x, mapdata.allnodes[d.to
    ].y)).y; });
20
21     var text = group.append("text")
22     .attr("x", function (d) { return parseInt((maps.
    latLngToLayerPoint(new L.LatLng(mapdata.allnodes[d.from].x,
    mapdata.allnodes[d.from].y)).x + maps.latLngToLayerPoint(new
    L.LatLng(mapdata.allnodes[d.to].x, mapdata.allnodes[d.to].y))
    .x) / 2) + 5; })
23     .attr("y", function (d) { return parseInt((maps.
    latLngToLayerPoint(new L.LatLng(mapdata.allnodes[d.from].x,
    mapdata.allnodes[d.from].y)).y + maps.latLngToLayerPoint(new
    L.LatLng(mapdata.allnodes[d.to].x, mapdata.allnodes[d.to].y))
    .y) / 2) - 5; })
24     .attr("class", "line-label");
25
26     elements.selectAll("text")
27     .text(function (d) {
28         return Math.round(mapdata.distances[d.from][d.to]) +
    " Meter";
29     });
30     elements.exit().remove();
31 };

```

Listing 7.16 Function Lines Maps

6. Save semua function yang dibuat dan jalankan aplikasi dengan cara buka terminal pada visual studio code kemudian masuk ke folder public dan masukkan perintah php -S localhost:8080.



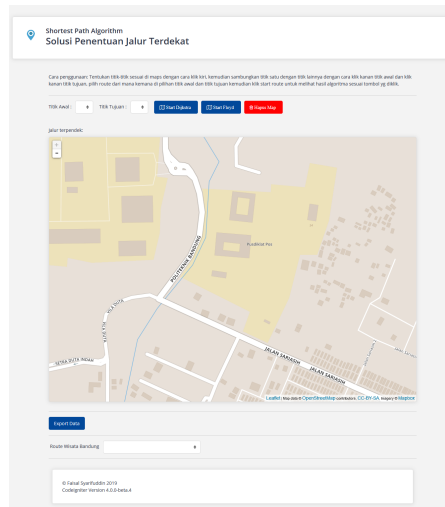
```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\xampp7\htdocs\ci4_leafletjs> cd public
PS C:\xampp7\htdocs\ci4_leafletjs\public> php -S localhost:8080
PHP 7.3.3 Development Server started at Sat Aug 17 21:27:50 2019
Listening on http://localhost:8080
Document root is C:\xampp7\htdocs\ci4_leafletjs\public
Press Ctrl-C to quit.

```

7. Jalankan aplikasi `ci4_leaflet.js` anda pada browser kesayangan anda, dengan cara mengetikkan "localhost:8080", maka akan muncul tampilan seperti berikut:



DAFTAR PUSTAKA

1. R. Awangga, “Sampeu: Servicing web map tile service over web map service to increase computation performance,” in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.
2. S. A. Salman and A. Hussin, “The minimum cost for the vascular network using linear programming based its path graph,” *Iraqi Journal of Science*, vol. 60, no. 4, pp. 859–867, 2019.
3. D. J. Williams, “The crusader and the dictator: An exploration of ideology and neurodivergence in contemporary technology practice,” 2019.
4. S. Ghanbartehrani and J. D. Porter, “An efficient algorithm for solving the median problem on real road networks,” *Engineering Optimization*, pp. 1–14, 2019.
5. M. Hellwig, D. Entner, T. Prante, A.-C. Zăvoianu, M. Schwarz, and K. Fink, “Optimization of ascent assembly design based on a combinatorial problem representation,” in *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*. Springer, 2019, pp. 291–306.
6. H. T. Hidayat, “Sistem penunjang keputusan pencarian jarak terpendek menuju rumah sakit dan puskesmas dengan metode dijkstra,” *TECHSI-Jurnal Teknik Informatika*, vol. 8, no. 1, pp. 95–124, 2019.
7. R. Novianti and R. A. Krisdiawan, “Implementasi algoritma floyd warshall pada aplikasi pengaduan masyarakat berbasis android,” *NUANSA INFORMATIKA*, vol. 13, no. 1, 2019.

8. A. R. Hasibuan, “Penerapan algoritma floyd warshall untuk menentukan jalur terpendek dalam pengiriman barang,” *JURIKOM (Jurnal Riset Komputer)*, vol. 3, no. 6, 2016.
9. Y. Darnita, R. Toyib, and R. Rinaldi, “Implementasi algoritma floyd warshall untuk menentukan letak dan lokasi perusahaan travel/rental mobil di kota bengkulu,” *Pseudocode*, vol. 4, no. 2, pp. 144–155, 2017.
10. H. F. Siregar and N. Sari, “Rancang bangun aplikasi simpan pinjam uang mahasiswa fakultas teknik universitas asahan berbasis web,” *Jurnal Teknologi Informasi*, vol. 2, no. 1, pp. 53–59, 2018.
11. M. I. Sugiarto, A. Linarta, and A. Sofiyan, “Aplikasi layanan informasi absen dan nilai berbasis sms gayeway menggunakan php pada smk taruna persada dumai,” *INFORMATIKA*, vol. 9, no. 2, pp. 60–69, 2019.
12. I. K. Siregar and F. Taufik, “Perancangan aplikasi sms alert berbasis web,” *JIMP-Jurnal Informatika Merdeka Pasuruan*, vol. 2, no. 2, 2017.
13. D. Prabowo, “Website e-commerce menggunakan model view controller (mvc) dengan framework codeigniter studi kasus: Toko miniatur,” *Data Manajemen dan Teknologi Informasi (DASI)*, vol. 16, no. 1, p. 23, 2015.
14. V. Agafonkin, “Overview leaflet ’<https://leafletjs.com/index.html>’,” 2017.

Index

disruptif, xxix
modern, xxix