

TUGAS AKHIR

PEMVISUALISASI HASIL PENELITIAN AREA HIJAU KELURAHAN



Bosnich Timothy Bonsaleng

NPM: 2017730086

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023**

FINAL PROJECT

**VISUALIZATION OF SUBDISTRICT GREEN AREA
RESEARCH RESULT**



Bosnich Timothy Bonsaleng

NPM: 2017730086

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2023**

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

DAFTAR ISI

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Batasan Masalah	4
1.5 Metodologi	4
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	7
2.1 <i>Command-line Interface</i>	7
2.1.1 SCP(<i>Secure Copy Protocol</i>)	8
2.2 Hadoop Distributed File System	8
2.3 <i>Python</i>	9
2.3.1 <i>Pillow (PIL Fork)</i>	10
2.4 <i>Base64</i>	11
2.5 Framework Laravel	12
3 ANALISIS	17
3.1 Proses Pembentukan Gambar	17
3.1.1 Mengunduh File Text	18
3.1.2 Mengkonversi Baris Menjadi Gambar .png	19
3.1.3 Menggabungkan Gambar	20
3.2 Pembentukan Gambar Hasil Segmentasi	21
3.3 Pengumpulan Data Kelurahan	24
3.4 Analisis Perangkat Lunak	24
4 PERANCANGAN	27
4.1 Perancangan Tabel Data Wilayah	27
4.2 Perancangan Kelas Controller	27
4.3 Perancangan Fungsi	27
4.4 Perancangan Antarmuka	29
5 IMPLEMENTASI DAN PENGUJIAN	31
5.1 Implementasi	31
5.1.1 Lingkungan Perangkat Keras	31
5.1.2 Lingkungan Perangkat Lunak	31
5.1.3 Implementasi Basis Data	31
5.2 Hasil Implementasi	32
5.3 Pengujian	34

5.3.1	Pengujian Fungsional	34
5.3.2	Pengujian Eksperimental	36
6	KESIMPULAN DAN SARAN	41
6.1	Kesimpulan	41
6.2	Saran	41
	DAFTAR REFERENSI	43
	A KODE PROGRAM	45
	B KODE PROGRAM BASIS DATA	51

DAFTAR GAMBAR

1.1	RTH	1
1.2	Kelurahan Ciumbuleuit	3
2.1	<i>Command Prompt</i>	7
2.2	PHP Artisan Laravel	13
3.1	Gambar seluruh tile dari kelurahan Ciumbuleuit	17
3.2	Contoh gambar kelurahan Ciumbuleuit setiap <i>tile</i>	18
3.3	Data Citra Satelit berupa .txt	19
3.4	Gambar seluruh tile dari kelurahan Ciumbuleuit	20
3.5	Proses pengecekan file Bandung.txt	24
3.6	Diagram <i>Use Case User</i>	25
4.1	Rancangan Antarmuka	29
5.1	Antarmuka Halaman Utama	32
5.2	Antarmuka Halaman Utama (<i>Dropdown</i>)	33
5.3	Antarmuka Halaman Utama Menampilkan Data Kelurahan	33
5.4	Antarmuka Halaman Utama Menampilkan Gambar Citra Satelit Segementai	34
5.5	Diagram <i>Pie</i> tanggapan penguji.	37
5.6	Diagram <i>Pie</i> tanggapan penguji terhadap perbandingan kelurahan	37
5.7	Diagram <i>Pie</i> tanggapan penguji terhadap kemudahan pemandangan	38
5.8	Diagram <i>Pie</i> tanggapan pemvisualisasian gamar	38
5.9	Diagram <i>Pie</i> tanggapan informasi yang diterima	38
5.10	Diagram <i>Pie</i> tanggapan eksplorasi area hijau kelurahan	39
5.11	Diagram <i>Pie</i> tanggapan hasil keluaran yang sesuai	39

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Ruang Terbuka Hijau merupakan suatu ruang terbuka di kawasan perkotaan yang didominasi tutupan lahannya oleh unsur hijau (vegetasi) serta memiliki fungsi antara lain sebagai area untuk rekreasi, sosial budaya, estetika, ekologis dan dapat memberikan nilai ekonomis bagi perkembangan suatu wilayah perkotaan (lihat Gambar 1.1). Definisi RTH sendiri dalam pasal 1 UU No.26/2007 tentang Penataan Ruang adalah area memanjang/jalur dan/atau mengelompok, yang penggunaannya lebih bersifat terbuka, tempat tumbuh tanaman, baik yang tumbuh secara alamiah maupun yang sengaja ditanam. Pada pasal 29 disebutkan bahwa ruang terbuka hijau terdiri dari ruang terbuka hijau publik dan ruang terbuka hijau privat, dimana proporsi ruang terbuka hijau kota paling sedikit 30% dari luas wilayah kota, sedangkan proporsi ruang terbuka hijau publik paling sedikit 20% dari luas wilayah kota.



Gambar 1.1: Contoh Ruang Terbuka Hijau¹

Pemanfaatan citra satelit merupakan sebuah cara agar dapat mengetahui luas RTH pada suatu kota. Citra Satelit adalah gambaran dari permukaan bumi yang didapatkan langsung dari satelit. Oleh karena itu, citra satelit dapat digunakan dalam mengidentifikasi RTH yang mana terdapatnya

¹Ilustrasi ruang terbuka hijau: <https://www.rth.bandung.go.id/>

banyak pepohonan pada suatu wilayah. Perhitungan juga dapat dilakukan pada citra satelit, dan hasil dari perhitungan luas RTH pada suatu wilayah diharapkan dapat memberikan dorongan untuk peningkatan dalam penghijauan agar dapat digunakan oleh pemerintah dalam merancang dan meningkatkan penghijauan di berbagai wilayah di Indonesia.

Penelitian yang dilakukan oleh Juan A. Kusjadi yaitu mengimplementasikan program untuk mengumpulkan, menyiapkan, dan menganalisis data citra satelit kelurahan dari beberapa kota/kabupaten di Indonesia menggunakan Hadoop MapReduce. Data kemudian disimpan pada sistem data lake yang telah dibuat pada Hadoop HDFS. Data hasil analisis dan perhitungan luas RTH juga sudah dilakukan evaluasi dengan nilai sesungguhnya[1]. Data hasil penelitian akan digunakan sebagai penunjang dalam pembuatan halaman web.

Pada Skripsi ini, akan dibangun sebuah halaman web yang interaktif yaitu pemvisualisasian dari hasil penelitian area hijau Kota Bandung[1]. Visualisasi adalah rekayasa dalam pembuatan gambar, diagram atau animasi untuk penampilan suatu informasi dalam penjelasan lain visualisasi adalah konversi data ke dalam format visual atau tabel sehingga karakteristik dari data dan relasi diantara item data atau atribut dapat di analisis atau dilaporkan, dan visualisasi data adalah satu dari yang teknik paling baik dan menarik untuk eksplorasi data. Manusia memiliki kemampuan membangun yang baik untuk menganalisis sejumlah besar informasi yang dipresentasi secara visual. Ia dapat mendeteksi pola umum dan trend, pencilaan dan pola yang tidak umum. Oleh karena itu, dengan dikembangkannya halaman web ini memiliki tujuan agar para pengguna dapat mengetahui informasi yang terdapat pada kelurahan. Informasi yang terdapat pada halaman website berupa nama kelurahan, luas wilayah kelurahan, gambar dari kelurahan/kecamatan, dll. Informasi yang terkumpul akan digunakan untuk mengembangkan halaman web.

Halaman web yang akan dikembangkan harusnya dapat diakses melalui komputer atau laptop. Dalam pengembangan halaman web pemvisualisasi area hijau kota Bandung akan dibantu pembuatannya dengan menggunakan *Framework* Laravel. Penggunaan *Framework* Laravel untuk memudahkan pengembang untuk membangun halaman web, sehingga pengguna yang akan mengakses halaman web akan dimudahkan dalam melihat informasi kelurahan dengan cepat.

Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Keunggulan ini didapatkan karena Laravel menggunakan konsep MVC (Model View Controller). Model pada Laravel berguna untuk membantu pengembang berinteraksi dengan database menggunakan *syntax migration* yang merupakan bawaan dari Laravel. Dengan *migration*, pengembangan dapat dengan mudah untuk melakukan modifikasi sebuah database pada sebuah platform secara independen karena implementasi skemas *database* yang direpresentasikan dalam sebuah class. View pada Laravel akan menjadi wadah tampilan website (*front-end*). Dan Controller yang berfungsi untuk merespon setiap request yang ada pada website sehingga setiap fungsi yang ada akan berfungsi sebagaimana mestinya. Dengan berbagai kemudahan dan fitur yang ada pada Laravel inilah yang membuat pengembang ingin menggunakannya dalam membangun halaman web pemvisualisasian area hijau kelurahan kota Bandung.

Dalam proses pengembangan halaman web tentu saja dibutuhkan sebuah data. Data yang akan digunakan dalam pembentukan halaman web berupa gambar dari kelurahan di Kota Bandung. Tidak hanya berupa gambar dari kelurahan tetapi juga berupa luas area wilayah untuk mengetahui besar wilayah kelurahan, mengetahui luas wilayah hijau kelurahan, dan melihat kebutuhan area

- 1 hijau terhadap kelurahan di Kota Bandung. Perhitungan luas wilayah, luas wilayah hijau, dan
- 2 kebutuhan area hijau telah dilakukan perhitungan untuk setiap kelurahan yang ada.



Gambar 1.2: Kelurahan Ciumbuleuit

3 Contoh gambar dari hasil penelitian Juan A. Kusjadi dapat dilihat pada gambar 1.2 yang
4 mana merupakan hasil proses pengambilan gambar sebuah kelurahan Ciumbuleuit yang telah
5 disimpan pada Hadoop HDFS. Proses pengambilan gambar dilakukan dengan menggunakan bahasa
6 pemrograman Python². Berbagai macam *library* yang dapat digunakan pada bahasa pemrograman
7 Python dalam membantu pengembangan laman web, diantaranya menggunakan *Python PIL(Pillow)*
8 yang berguna untuk menggabungkan gambar, dan *library* base 64 yang digunakan dalam melakukan
9 peng-*decode*-an teks yang merupakan sebuah *tile* gambar kelurahan.

10 Hasil dari pemvisualisasi ruang terbuka hijau kelurahan pada kota Bandung akan menjadi sebuah
11 halaman website yang interaktif yang dapat membandingkan kelurahan sesuai dengan masukan oleh
12 pengguna. Dengan dikembangkan halaman website ini maka pengguna dapat memenuhi kebutuhan
13 tempat tinggal bagi masyarakat agar dapat beraktivitas dengan normal dan mendapatkan kadar
14 oksigen yang merata.

15 1.2 Rumusan Masalah

16 Berdasarkan deskripsi dan latar belakang yang sudah dibahas bahwa rumusan masalah yang muncul
17 adalah sebagai berikut:

²Python adalah bahasa pemrograman yang banyak digunakan dalam aplikasi web, pengembangan perangkat lunak, ilmu data, dan *machine learning* (ML). Developer menggunakan Python karena efisien dan mudah dipelajari serta dapat dijalankan di berbagai platform.

- Bagaimana membuat sebuah halaman website interaktif yang dapat membandingkan data dua buah kelurahan Kota Bandung?
- Bagaimana cara pengguna untuk membandingkan atribut-atribut Citra Satelit dari kelurahan Kota Bandung?
- Bagaimana cara mengekstraksi data citra satelit pada HDFS ke *local directory*?

1.3 Tujuan

Tujuan dari skripsi ini adalah:

1. Membuat sebuah halaman website interaktif yang dapat membandingkan dua lokasi kelurahan.
2. Pengguna dapat memilih kelurahan untuk sisi kiri dan kanan, untuk membandingkan atributnya.
3. Data citra satelit yang didapatkan akan digunakan untuk memenuhi kebutuhan pada laman web yang akan dibangun.

1.4 Batasan Masalah

Tidak memiliki batasan masalah

1.5 Metodologi

Metodologi yang akan digunakan dalam pembuatan skripsi adalah:

1. Melakukan survei kepada Fritz H. Hutapea SKom dan Juan A. Kusjadi terkait penelitiannya
2. Melakukan pengumpulan data hasil penelitian
3. Mempelajari ekstraksi data citra satelit yang disimpan pada HDFS
4. Mempelajari bahasa pemrograman php, html, css dan cara menggunakan *framework* laravel.
5. Mempelajari kebutuhan laman web.
6. Melakukan analisis kebutuhan laman web.
7. Melakukan perancangan antar muka laman web.
8. Membangun laman web berdasarkan *framework* Laravel.
9. Melakukan pengujian pada laman web.
10. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Skripsi ini disusun dalam beberapa bab secara sistematis sebagai berikut:

- **Bab 1 Pendahuluan**

Berisikan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.

- **Bab 2 Landasan Teori**

Berisikan tentang dasar-dasar dari teori-teori yang digunakan dalam membangun halaman web seperti *Command-line interface*, *Hadoop Distributed File System*, *Python* beserta *library*-nya, *Base 64*, dan *Framework*.

- **Bab 3 Analisis**

Pada bab ini akan menjelaskan proses pembentukan gambar didalamnya terdapat bagaimana cara pengunduhan teks, pengkonversian baris menjadi gambar, dan menggabungkan gambar. Juga terdapat analisis kebutuhan perangkat lunak.

- **Bab 4 Perancangan**

Berisikan tentang penjelasan mengenai perancangan yang digunakan dalam membangun perangkat lunak.

- **Bab 5 Implementasi dan Pengujian**

Bab ini membahas tentang pengimplementasian rancangan antarmuka agar dapat menampilkan perbandingan visual antara dua kelurahan di kota Bandung. Juga akan membahas tentang pengujian fungsional dan pengujian eksperimental.

- **Bab 6 Kesimpulan dan Saran**

Bab ini berisikan kesimpulan dari hasil pengerjaan skripsi, dan saran untuk pengembangan selanjutnya.

BAB 2

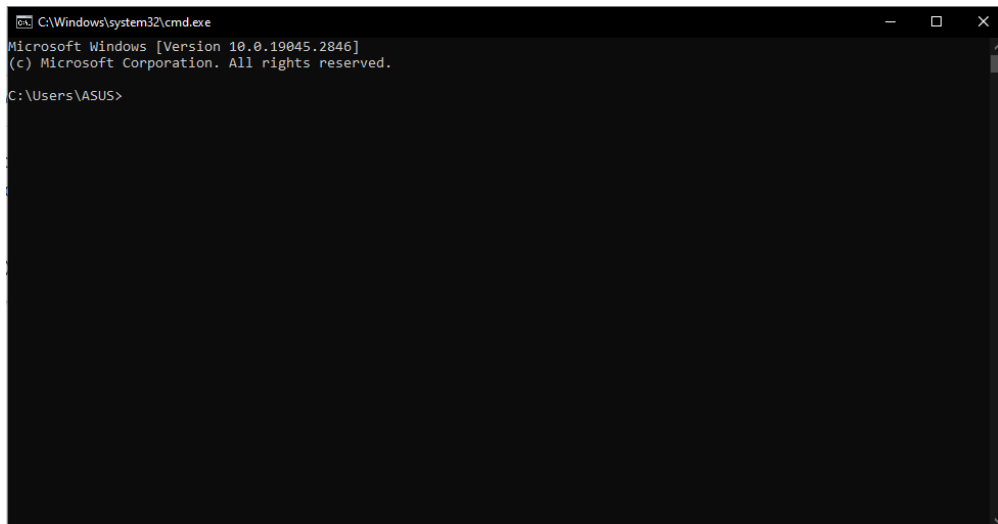
LANDASAN TEORI

Pada bab ini berisikan penjelasan tentang teori-teori yang perlu diketahui sebelum pengembangan halaman web dilakukan.

2.1 *Command-line Interface*

Command-line interface (CLI) merupakan sebuah antarmuka pengguna yang berbasis teks yang digunakan untuk menjalankan program, mengelola berkas-berkas pada komputer, dan dapat berinteraksi dengan komputer¹. *Command-line interface* juga disebut sebagai *command-line user interfaces*, *console user interfaces*, dan *character user interfaces*. *Command-line interface* menerima sebuah perintah yang diinput melalui keyboard perintah yang dipanggil oleh *command prompt* yang dijalankan oleh komputer.

Command-line interface langsung dapat berfungsi ketika sistem komputer dijalankan. *Command-line interface* dapat terbuka di layar kosong dengan *command prompt* lalu perintah-perintah dapat dimasukkan.



Gambar 2.1: *Command Prompt*

Jenis perintah-perintah dari *Command-line interface* akan berisikan :

1. Perintah-perintah dari sistem yang dikodekan sebagai bagian dari antarmuka sistem operasi

¹<https://www.techtarget.com/searchwindowsserver/definition/command-line-interface-CLI>

2. Program yang dapat dijalankan ketika berhasil dipanggil, dan menjalankan aplikasi yang berbasis teks atau grafis.

3. *batch program*² (*batch files* atau *shell script*) yang merupakan berkas teks berisikan urutan perintah-perintah. Ketika perintah berhasil dipanggil, *batch program* akan menjalankan perintahnya yang mungkin berisikan sebuah perintah sistem dan program yang dapat dieksekusi.

Perintah *Command-line interface* yang digunakan antarlain :

2.1.1 SCP (*Secure Copy Protocol*)

Salah satu perintah yang terdapat pada *Command-line interface* yaitu SCP (*secure copy*). SCP memiliki fungsi yang mirip seperti pada perintah *cp* (*copy*) yaitu untuk menyalin berkas^[2]. Perbedaannya yang paling terlihat terletak pada sumber atau tujuan ke *remote host*. Sebagai contoh, jika ingin menyalin sebuah dokumen dari *local directory* (berkas dalam komputer) ke *remote system*, atau dari *working directory* ke *local directory*.

```
C:\Users\Asus>scp ssh i17086@10.100.69.101:Kota_Bandung.txt
```

Kode 2.1: Pemanggilan SCP

Kode program 2.1 merupakan contoh penyalinan berkas Kota_Bandung.txt. Berkas tersebut yang tersimpan didalam *remote host* dan disalin ke *local directory* pengguna.

2.2 Hadoop Distributed File System

HDFS (*Hadoop Distributed File System*) merupakan sistem file terdistribusi yang berada pada penyimpanan server dan memiliki banyak kesamaan pada *base storage system*. Sistem penyimpanan terdistribusi ini dapat menyimpan data dalam jumlah yang sangat besar melalui jaringan komputer dengan redundansi bawaan untuk melindungi data. HDFS dirancang untuk pemrosesan yang cepat dan toleran terhadap kesalahan, sehingga memungkinkan pengguna *hardware* pada penyimpanan tidak terkena biaya yang mahal.

HDFS memungkinkan para pengguna untuk menyimpan data kedalam file yang dibagi menjadi beberapa *block*. Karena *Hadoop* dirancang untuk bekerja dengan jumlah data yang besar, ukuran *block* pada HDFS jauh lebih besar daripada yang digunakan oleh *typical relational databases*. Dengan ukuran awal *block* sebesar 128MB, dan dapat dikonfigurasi ukurannya mencapai 512MB.^[3]

HDFS memiliki 2 jenis *node*, yaitu *namenode* sebagai *node master* dan *datanode* sebagai *node slave*. Kelebihan utama yang ditawarkan HDFS adalah *scalability* dan *availability* yang dicapai dikarenakan memiliki kemampuan replikasi data dan *fault tolerance*. Dengan adanya kemampuan replikasi data/file, ketika ada kegagalan *software* atau *hardware*, HDFS akan melakukan replikasi ulang blok-blok data pada *node* yang mengalami kegagalan.^[4]

Semua perintah HDFS dipanggil menggunakan *script bin/hdfs*. Penjalanan *script "hdfs"* tanpa argumen akan mencetak deskripsi untuk semua perintah.³

```
C:\Users\Asus>hdfs [SHELL_OPTION] COMMAND [GENERIC_OPTIONS] [COMMAND_OPTIONS]
```

Kode 2.2: Perintah HDFS CLI

²file teks yang berisi serangkaian perintah yang dimaksudkan untuk dieksekusi oleh *command interpreter*

³<https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html>

- 1 Kode program 2.2 merupakan pemanggilan perintah pada HDFS. Setiap opsi perintah memiliki
- 2 fungsi untuk menjalankan *script* pada CLI. Penjelasan tiap opsi dijelaskan pada tabel 2.1.

Tabel 2.1: Hadoop memiliki opsi parsing framework yang menjelaskan setiap fungsi kelasnya

COMMAND_OPTION	Deskripsi
SHELL_OPTIONS	kumpulan shell_option yang umum
GENERIC_OPTIONS	kumpulan generic_option yang didukung oleh beberapa perintah
COMMAND COMMAND_OPTIONS	Bermacam perintah dengan opsi

Penggunaan perintah HDFS yang digunakan antara lain :

1. Penggunaan perintah **dfs**

Perintah **dfs** digunakan untuk menjalankan(*run*) perintah *filesystem* yang didukung oleh *Hadoop*. [**COMMAND_OPTIONS**] dapat dilihat pada *File System Guide*. Contoh pemanggilan **dfs** seperti pemanggilan pada Gambar 2.3

```
C:\Users\Asus>hdfs dfs [COMMAND [COMAND_OPTIONS]]
```

Kode 2.3: Perintah HDFS dfs

2. Penggunaan perintah **get**

Perintah **get** digunakan untuk menyalin file HDFS ke *local system*. Gambar 2.4 merupakan contoh yang menunjukkan cara penggunaan perintah **-get** untuk mengunduh file dari HDFS ke *local file system*

```
C:\Users\Asus>hdfs dfs -get /user/if18059/geodata/cropped/arcgis/16/
Jawa_Barat/Kota_Bandung.txt .
```

Kode 2.4: Perintah HDFS dfs -get untuk mengunduh file HDFS Kota_Bandung.txt ke *local system*

3. Penggunaan perintah **-ls**

Perintah **-ls** digunakan untuk menampilkan daftar isi *directory* yang ditentukan oleh *path* yang disediakan oleh pengguna. Gambar merupakan contoh yang menunjukkan cara penggunaan perintah **-ls** untuk melihat isi file HDFS.

```
C:\Users\Asus>hdfs dfs -ls /user/if18059/geodata/cropped/arcgis/16/
Jawa_Barat
```

Kode 2.5: Perintah HDFS dfs -get untuk mengunduh file HDFS Kota_Bandung.txt ke *local system*

2.3 Python

Python adalah bahasa pemrograman yang memulai debutnya pada tahun 1991. *Python* mencakup *object-oriented programming* dan memperkenalkan *syntax* yang membuat banyak *operations* menjadi sangat ringkas dan elegan. Hal yang harus diperhatikan oleh *programmers* baru mengenai *Python* adalah pemakaian spasi(" ") sangat berpengaruh pada arti program yang dikembangkan. Pada proses pengembangan menggunakan bahasa *Python* harus menggunakan *text editor* yang dapat mengenali *syntax*-nya agar memudahkan membuat program sesuai yang diinginkan.[5]

Bahasa pemrograman *Python* Merupakan sebuah bahasa pemograman komputer yang dikembangkan khusus untuk membuat *source code* yang mudah untuk dibaca. *Pyhton* memiliki *library*

- 1 yang lengkap sehingga memudahkan seorang *programmer* untuk membuat sebuah aplikasi sesuai
- 2 dengan keinginan dengan menggunakan *source code* yang terlihat sederhana.

3 2.3.1 *Pillow (PIL Fork)*

4 *Python Imaging Library* merupakan salah satu *library* yang terdapat pada bahasa pemrograman
 5 *Python*. *PIL* dapat menambahkan pemrosesan gambar ke bahasa pemrograman *Python*. *Library* ini
 6 menyediakan *extensive file format*, representasi internal yang efisien, dan memiliki kemampuan yang
 7 baik dalam pemrosesan gambar. Pentingnya *library* yang dirancang untuk dapat mengakses data
 8 yang disimpan dengan cepat dalam berbagai format piksel. Seharusnya memberikan dasar yang
 9 kuat sebagai alat pengolahan gambar⁴.

10 *Python Imaging Library* sangat ideal untuk pengarsipan gambar dan aplikasi pemrosesan *batch*.
 11 Penggunaan *library* untuk membuat thumbnail, mengonversi antara format file, mencetak gambar,
 12 dll. Versi saat ini dapat mengidentifikasi dan membaca sejumlah besar format. Pembantuan
 13 dalam penulisan ini sengaja dibatasi dalam format pertukaran dan representasi yang paling umum
 14 digunakan.

15 *Python Imaging Library* yang rilis saat ini mencakup antarmuka Tk *PhotoImage* dan *BitmapImage*, serta
 16 *Windows DIB interface* yang dapat digunakan dengan *PythonWin* dan berbagai
 17 macam *toolkits* yang berbasis Windows. Banyak *toolkits* GUI (*Graphical User Interface*) lainnya
 18 yang dilengkapi dengan dukungan *PIL*. Untuk *debugging*, ada juga metode *show()* yang menyimpan
 19 gambar ke disk, dan memanggil utilitas tampilan eksternal.

20 Penggunaan kelas *Image*

21 Dalam penggunaan *Python Imaging Library* terdapat kelas yang paling penting yaitu kelas *Image*,
 22 yang didefinisikan dalam modul dengan nama yang sama. Pembuat *instance* dari kelas ini dengan
 23 beberapa cara; Baik dengan memuat gambar dari file, memproses gambar lain, atau membuat
 24 gambar dari awal. Memuat gambar dari file.

25 1. Penggunaan fungsi *Image.open()*

26 Berfungsi untuk membuka dan mengidentifikasi file gambar yang diberikan. Fungsi ini
 27 mengidentifikasi sebuah file, tetapi file tetap terbuka dan data gambar tidak terbaca sampai
 28 data file gambar tersebut diproses. Memiliki parameter **fp** merupakan nama file yang akan
 29 dibuka, **mode** memiliki argumen "r", dan **formats** sebuah daftar format untuk mencoba
 30 memuat file. Parameter ini dapat digunakan untuk membatasi format yang akan diperiksa.
 31 Dalam penggunaan fungsi dapat dilihat pada kode program 2.6.

```
32
33 1 from PIL import Image
34 2
35 3 im = image.open("hopper.ppm")
36
```

Kode 2.6: Pemanggilan fungsi *open()*

37 Jika pemanggilan fungsi berhasil, fungsi yang dipanggil akan mengembalikan sebuah objek
 38 *Image*

⁴<https://pillow.readthedocs.io/en/stable/handbook/overview.html>

2. Penggunaan fungsi *Image.new()*

Berfungsi untuk membuat gambar baru dengan mede dan ukuran yang diberikan. Memiliki parameter **mode** untuk menentukan jenis dan kedalaman piksel dalam gambar seperti mode "L"(8-bit piksel,skala abu-abu), "RGB" (3x8-bit piksel,warna asli), "RBGA" (4x8-bit piksel, warna asli dengan *transparency mask*), dll. Parameter **size** merupakan ukuran dari gambar baru, berisi ukuran horizontal dan vertikal dalam piksel. Parameter **color** memberikan warna apa yang akan digunakan. Biasanya akan langsung diberikan warna hitam.

```
1 from PIL import Image
2
3 im = image.new("hopper.ppm")
```

Kode 2.7: Pemanggilan fungsi *new()*

Jika pemanggilan fungsi pada gambar 2.7 berhasil, maka akan mengembalikan sebuah objek *image*.

3. Penggunaan fungsi *Image.paste()*

Berfungsi untuk menempelkan sebuah objek gambar ke objek gambar lain. Ukuran gambar yang ditempelkan harus sesuai dengan ukuran gambar. Memiliki parameter **im** yang merupakan sebuah objek image atau nilai piksel. Parameter **box** 4-tupel opsional yang memberikan wilayah untuk ditempelkan. Jika 2-tupel digunakan sebagai gantinya, itu diperlakukan sebagai sudut kiri atas. Jika dihilangkan atau tidak ada, objek gambar yang ditempelkan ke sudut kiri atas. Parameter **mask** merupakan sebuah optional *mask* gambar.

```
1 from PIL import Image
2
3 im = image.new("hopper.ppm")
4 im1 = image.open()
5
6 im.paste(im1,(256,256))
```

Kode 2.8: Pemanggilan fungsi *paste()*

Jika fungsi pemanggilan pada gambar 2.8 berhasil, akan mengembalikan sebuah objek *Image* yang memuat gambar im1 yang ditempelkan pada gambar baru im.

4. Penggunaan fungsi *Image.save()*

Berfungsi untuk menyimpan gambar dengan nama file yang diberikan. Jika tidak memiliki format yang ditentukan, maka format yang akan digunakan ditentukan dari ekstensi penamaan file, jika memungkinkan. memiliki beberapa parameter diantaranya adalah **fp** merupakan nama file yang akan digunakan memiliki tipe data string, parameter **format** merupakan format file yang akan digunakan pada file tersebut, dan **params** merupakan parameter tambahan untuk penulisan gambar.

2.4 Base64

Base64 merupakan sebuah algoritma yang digunakan untuk mengubah tipe data *bytes* menjadi tipe data yang dapat dilihat(dan sebaliknya). Skema pengkodean biner ke teks pada Base64 sebagai persyaratan untuk mengirim tipe data *bytes* melalui jaringan komunikasi yang tidak mengizinkan

tipe data biner tetapi hanya tipe data berbasis teks. Data teks yang dihasilkan terdiri dari berbagai karakter yang terdapat pada standar ASCII. Penggunaan kata Base64 berasal dari jumlah karakter ASCII yang digunakan. 64 karakter yang digunakan antara lain adalah 26 karakter a-z *lowercase*, 26 karakter A-Z *uppercase*, ditambah dengan 2 karakter tambahan yaitu karakter tambah "+" dan karakter garis miring "/". Base64 juga sebenarnya memiliki karakter ke 65 yaitu karakter sama dengan "=" yang digunakan sebagai *padding*. Karakter sama dengan ("=") digunakan pada segmen terakhir data biner yang tidak memiliki total 6 *bit*. Keseluruhan karakter yang digunakan pada Base64 disebut juga tabel encoding Base64.

Base64 bekerja dengan cara memotong data biner menjadi segmen-segmen berukuran 6 *bit*. Base64 hanya menggunakan 6 *bit* untuk bisa memenuhi seluruh karakter yang digunakan ($2^6 = 64$). Masing-masing segmen tersebut kemudian dibaca ke dalam tipe desimal lalu dikonversi ke karakter ASCII. Sebagai contoh konversi data yang berisi 3 buah *byte* yaitu 155, 162, dan 233. Tipe data *byte* diubah menjadi data biner dan diambung menjadi satu yaitu 100110111010001011101001. Kemudian data biner dipotong menjadi segmen yang berisi 6 *bit* menjadi 100110, 111010, 001011, 101001. Masing-masing data dikonversi menjadi desimal, 58, 11, 4 yaitu 381. Terakhir data dikonversikan ke karakter ASCII yang berada pada tabel encoding Base64 menjadi m6Lp. Cara yang sama namun terbalik prosesnya digunakan untuk mendeskripsi data dari Base64 kembali ke tipe data *byte*.^[1]

2.5 Framework Laravel

Framework adalah kerangka kerja yang digunakan oleh developer untuk memudahkan pembangunan aplikasi web yang dapat berupa sekumpulan *library* yang berisi fungsi, *tools*, ataupun *class-class*, dan digunakan sebagai kerangka dalam pembangunan aplikasi web. Umumnya didalam *framework* telah menyediakan solusi untuk dapat mengakses *database*, *authentication*, *templating*, *controls*, dan fungsi-fungsi lainnya/ Dalam penggunaan *framework* diharapkan dapat membuat pengembangan aplikasi menjadi rapi dan bersih, memiliki struktur yang optimal, dan *reusable*.

Laravel adalah *framework* aplikasi web dengan sintaks yang ekspresif dan elegan. Laravel adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model-view - controller*. Laravel berada di bawah lisensi MIT *License* dengan menggunakan Github sebagai tempat berbagi code menjalankannya. Laravel berlisensi *open source* yang artinya bebas digunakan tanpa harus melakukan pembayaran. Alamat website resmi dari *framework* Laravel adalah <https://laravel.com>. Kelebihan laravel adalah sebagai berikut:

- Progresif *Framework*

Progresif yang dimaksud adalah *framework* ini dapat bertumbuh bersama developer. Yang artinya dapat diikuti oleh developer baru maupun developer senior dikarenakan terdapat dokumentasi, panduan, dan tutorial video laravel yang dapat membantu membangun perangkat lunak.

- Komunitas *Framework*

Pada laravel terdapat banyak sekali *packages* terbaik dalam ekosistem PHP. selain itu, ribuan pengembang berbakat dari seluruh dunia telah berkontribusi pada *framework* ini.

- Berskala *Framework*

Laravel memberikan dukungan sistem cache yang terdistribusi dengan cepat. Faktanya laravel dapat menangani ratusan juta *request* setiap bulan.

Dalam penggunaan laravel memiliki beberapa kekurangan salah satunya yaitu ukuran file yang cukup besar. Di dalam laravel terdapat file yang sifatnya default seperti vendor. File tersebut tidak boleh dihapus sembarangan sehingga ukuran website yang dibuat berukuran cukup besar. Selain itu, dibutuhkan koneksi internet untuk instalasi dan mengunduh *library* laravel, dan PHP minimal versi 5.4 untuk menjalankannya. Berikut adalah dasar-dasar laravel:

1. Artisan

Artisan adalah command line atau perintah yang dijalankan melalui terminal dan disediakan beberapa perintah yang dapat digunakan selama melakukan pengembangan dan pembuatan aplikasi. Salah satu fungsi dari php artisan yaitu "php artisan serve". Php artisan serve berfungsi untuk membuka website yang telah dibuat tanpa menggunakan web server lokal. Gambar 2.2 merupakan contoh salah satu penggunaan artisan dalam laravel.

```
C:\xampp\htdocs\skripsiPemvisualisasi\SkripsiPemvisualisasi\pemvisualisasiRTH>php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

Gambar 2.2: PHP Artisan Laravel

2. Controller

Controller merupakan suatu proses yang bertujuan untuk mengambil data, menambahkan data, menghapus data, atau mengubah data untuk ditampilkan dalam halaman. Cara membuat *controller* adalah dengan menggunakan *command line* dengan memasukkan "php artisan make controller «nama_controller»". File *controller* nantinya akan otomatis terbuat dan sudah masuk ke folder *controller*.

3. Routing

Routing merupakan suatu proses yang dapat memindahkan tampilan halaman ke halaman lain. Dengan menggunakan *routing*, pengguna dapat menentukan halaman yang ingin dikunjungi. Pengaturan *routing* di laravel terletak pada file *web.php*.

4. Blade View

Blade adalah *template engine*. Pada dasarnya *Blade* adalah *view* namun dengan menggunakan *Blade* akan mempermudah untuk mengatur tampilan *website* dan menampilkan data. Cara untuk membuat file *view* menjadi file *Blade* adalah dengan menambahkan ekstensi *.blade.php* pada file *view*. Dan cara untuk memanggil file *Blade* sama dengan cara untuk memanggil file *view* biasa.

Setelah melakukan penginstalan Laravel akan terlihat direktori yang berisi aplikasi Laravel dasar. File dan direktori yang terdapat sebagai berikut:[6]

```
app/
bootstrap/
config/
public/
resources/
routes/
storage/
```

```
1  tests/
2  vendor/
3  .editorconfig
4  .env
5  .env.example
6  .gitattributes
7  .gitignore
8  artisan
9  composer.json
10 composer.lock
11 package.json
12 phpunit.xml
13 readme.md
14 server.php
15 webpack.mix.js
```

Direktori utama (*root directory*) mengandung folder-folder berikut secara *default*:

- *app*
Berisikan sebagian besar aplikasi saat dijalankan. *Model*, *controller*, *commands*, dan kode domain PHP yang dibuat semuanya akan berada di direktori ini.
- *bootstrap*
Berisi file-file yang digunakan oleh *framework* Laravel saat setiap kali dijalankan.
- *config*
Berisikan semua file konfigurasi aplikasi.
- *database*
Berisi file *database migration* dan *seeds*.
- *public*
Direktori yang ditunjuk oleh server ketika menjalankan aplikasi web. Berisi file *index.php*, yang merupakan *entry point* untuk menangani semua *request* yang masuk ke aplikasi. Didalam folder ini dapat menyimpan beberapa aset dari aplikasi seperti gambar, *JavaScript*, dan *CSS*.
- *resource*
Berisi file *view* dari aplikasi yang dibuat. Didalam folder ini juga terdapat file *language* yang digunakan aplikasi.
- *routes*
Berisi file yang digunakan untuk mendefinisikan semua *route* ke aplikasi. Secara *default* ada tiga file *route* yang disediakan oleh Laravel yaitu *api.php*, *console.php*, dan *web.php*.
- *storage*
Berisi *template Blade* yang dikompilasi, file *session*, file *cache*, dan file lainnya yang dihasilkan secara otomatis oleh Laravel.
- *tests*
Berisi semua file *test* yang dibuat untuk aplikasi.
- *vendor*
Berisikan tempat *Composer* menginstal dependensinya. Direktori ini akan diabaikan oleh Git,

karena *Composer* diharapkan dapat berjalan sebagai bagian dari proses implementasi pada server-server jarak jauh.

Direktori utama juga berisi file-file berikut:

- *.editorconfig*

Memberikan instruksi kepada IDE/editor teks tentang standar penulisan kode Laravel (seperti, ukuran indentasi, set karakter, dan apakah harus memotong *whitespace* di ujung baris).

- *.env* dan *.env.example*

Menentukan variabel-variabel lingkungan (variabel yang diharapkan berbeda di setiap lingkungan dan karena itu tidak dimasukkan ke dalam *version control*). *.env.example* adalah template yang setiap lingkungan harusnya menduplikasi untuk membuat file *.env*-nya sendiri, yang akan diabaikan oleh Git.

- *.gitattributes* dan *.gitignore*

Berisikan file-file untuk pengkonfigurasi Git.

- *artisan*

Berisikan file untuk menajalankan perintah-perintah artisan dari *command-line*.

- *composer.json* dan *composer.lock*

Berisikan file konfigurasi untuk *Composer*. *composer.json* dapat diedit oleh pengguna sedangkan *composer.lock* tidak dapat diedit. File-file ini berisi informasi dasar tentang proyek dan juga mendefinisikan dependensi dari PHP.

- *package.json*

File yang berisikan sama seperti *composer.json*, tetapi untuk aset *front-end* dan dependensi dari sistem pembangunan. File ini juga memberi instruksi kepada NPM tentang dependensi berbasis *JavaScript* yang harus diunduh.

- *phpunit.xml*

Berisikan file konfigurasi untuk *PHPUnit* merupakan alat yang digunakan Laravel secara *default* untuk pengujian.

- *readme.md*

Sebuah file *Markdown* yang memberikan pengenalan dasar tentang Laravel. File ini tidak dapat dilihat jika menggunakan instalator dari Laravel.

- *server.php*

Berisikan sebuah server cadangan yang mencoba untuk memungkinkan server yang kurang mampu agar tetap dapat melihat *preview* aplikasi Laravel.

- *webpack.mix.js*

File konfigurasi yang bersifat optional untuk *Mix*. Jika menggunakan Elixir, maka akan melihat *gulpfile.js* sebagai gantinya. File-file ini digunakan untuk memberikan instruksi kepada sistem pembangunan tentang cara mengkompilasi dan memproses aset *front-end* aplikasi Laravel.

BAB 3

ANALISIS

Pada bab ini akan membahas mengenai proses pengumpulan data citra satelit yang berada di Laboratium Fakultas Teknik Informatika dan Sains (FTIS) UNPAR (Universitas Katolik Parahyangan) beserta gambar citra satelit segmentasi, pengambilan data dari hasil penelitian yang telah dilakukan oleh Juan A. Kusnandji, dan analisis kebutuhan perangkat lunak dalam pengembangan halaman web.

3.1 Proses Pembentukan Gambar

Pada Gambar 3.1 merupakan hasil dari penggabungan gambar per *tile*. Langkah-langkah dalam proses pengambilan data berupa gambar citra satelit dari kelurahan di Kota Bandung.



Gambar 3.1: Gambar seluruh tile dari kelurahan Ciumbuleuit

Proses awal dilakukan meliputi data yang diambil dari sistem Hadoop yang disimpan pada Hadoop Laboratoium FTIS UNPAR. Kemudian data yang telah diambil berupa file ".txt" yang setiap baris dari file tersebut merupakan sebuah file gambar berupa *tile* seperti pada gambar(3.2). Kumpulan gambar per *tile* akan digabungkan dengan menggunakan *script*. Penggabungan gambar setiap *tile* akan menghasilkan sebuah gambar dari kelurahan seperti pada gambar 1.2.



Gambar 3.2: Contoh gambar kelurahan Ciumbuleuit setiap *tile*

3.1.1 Mengunduh File Text

Pada penelitian yang telah dilakukan oleh Juan Anthonius Kusjadi menghasilkan data yang telah disimpan pada sistem *data lake* yang telah dibuat pada HDFS.[1] Pada proses pengunduhan data harus terlebih dahulu mendaftarkan akun HDFS pada admin laboratoium FTIS UNPAR agar mendapat akses kedalam sistem penyimpanan HDFS(*Hadoop Distributed File System*). Setelah mendapatkan aksesnya, lalu dapat mengunduh data citra satelit kelurahan per kota dengan perintah seperti pada kode 3.1.

```
hdfs dfs -get /user/if18059/geodata/cropped/arcgis/16/Jawa_Barat/Kota_Bandung.
txt .
```

Kode 3.1: *Command-line* HDFS

Metode yang umum digunakan untuk mengunduh file dari server jarak jauh secara aman adalah dengan menggunakan SCP (*Secure Copy Protocol*). SCP adalah perintah baris perintah yang memungkinkan pengguna untuk mentransfer file antara komputer lokal dan server jarak jauh melalui koneksi yang aman. Dalam pengunduhan file dari server dapat menggunakan perintah 'scp' diikuti oleh alamat sumber file di server dan alamat tujuan pada penyimpanan lokal. Dapat dilihat pada *command-line* 3.2, akan mengunduh file 'Kota_Bandung.txt' dari server HDFS laboratorium FTIS UNPAR ke lokasi yang ditentukan pada penyimpanan lokal. Dengan demikian, File yang telah diunduh dapat dengan mudah digunakan.

```
C:\Users\Asus>scp ssh i17086@10.100.69.101:Kota_Bandung.txt
```

Kode 3.2: *Command-line* SCP

Data Kota_Bandung.txt dapat dilihat pada gambar 3.3. Isi dari berkas Kota_Bandung.txt memiliki 9 kolom yang dipisahkan oleh tanda titik koma (;). Pada kolom pertama diisi dengan nama kelurahan. Pada kolom kedua diisi dengan nama kota. Pada kolom ketiga diisi dengan nama provinsi. Kolom keempat diisi dengan nilai panjang tile untuk kelurahan tersebut. Kolom kelima diisi dengan nilai lebar dari tile untuk kelurahan tersebut. Kolom keenam diisi dengan posisi x koordinat tile untuk kelurahan tersebut. Kolom ketujuh diisi dengan posisi y koordinat tile untuk kelurahan tersebut. Kolom kedelapan diisi dengan ukuran luas per piksel dalam km2 untuk tile pada kelurahan tersebut. Terakhir kolom kesembilan diisi dengan data tile citra satelit dengan format png yang sudah dienkripsi dalam bentuk Base64[1].



Gambar 3.3: Data Citra Satelit berupa .txt

3.1.2 Mengkonversi Baris Menjadi Gambar .png

Dalam penelitian ini telah dikembangkan sebuah *script* yang bertujuan untuk mengekstraksi gambar pertile dari data Kota_Bandung.txt. Script yang telah dikembangkan dapat dilihat kode program 3.3.

```
import base64

file = open("Kota_Bandung.txt", "r+")
kordinat_x = 0
kordinat_y = 0
result = []

for line in file:
    file_line = file.readline().split(";", 8)
    image_data = file_line[8]
    kordinat_x = file_line[5]
    kordinat_y = file_line[6]

    imgdata = base64.b64decode(image_data)
    filename = file_line[0] + str(kordinat_x) + str(kordinat_y) + '.png'
    with open(filename, 'wb') as imgd:
        imgd.write(imgdata)

file.close()
```

Kode 3.3: Script Mengekstraksi gambar per tile

Script ini menggunakan *library* base64 untuk mengelola data gambar yang disimpan dalam format base64. Pertama, *script* membuka file Kota_Bandung.txt dalam mode pembacaan ('r+'). Variabel kordinat_x dan kordinat_y diinisialisasi ke nilai 0. Selama iterasi berlangsung, data dari file dibaca per baris menggunakan perulangan *for line in file*. Setiap baris diproses dengan membaginya menjadi elemen-elemen dengan pemisah titik koma (;). Data gambar yang terdapat di kolom ke-8 di-*decode* dari format base64 menggunakan base64.b64decode dan disimpan dalam

1 variabel `imgdata`. Selanjutnya, nama file gambar ditentukan dengan menggabungkan beberapa
 2 elemen, seperti nama kelurahan, `kordinat_x`, dan `kordinat_y`, dan diberi ekstensi `'.png'`.

3 Gambar yang telah di-decode dan disimpan dalam `'imgdata'` lalu ditulis ke dalam file baru
 4 dengan nama yang telah ditentukan dalam mode *binary* (`'wb'`) menggunakan `open` dan `imgd.write`.
 5 Terakhir, setelah selesai mengolah semua baris, file sumber `Kota_Bandung.txt` ditutup menggunakan
 6 `file.close()`. Dengan *script* code ini, data gambar dalam format base64 di-extract dan disimpan
 7 sebagai file gambar `.png` dengan nama yang koordinat `_x` dan koordinat `_y`. Hasil dari penkonversian
 8 gambar dari baris dapat dilihat pada gambar 3.4.



Gambar 3.4: Gambar seluruh tile dari kelurahan Ciumbuleuit

9 3.1.3 Menggabungkan Gambar

10 *Script* yang dikembangkan dalam penelitian ini memiliki tujuan utama untuk menggabungkan
 11 sejumlah gambar per tile menjadi sebuah gambar utuh yang merepresentasikan kelurahan. Kode
 12 pemrograman dapat dilihat pada kode program 3.4, menggunakan library PIL (*Python Imaging*
 13 *Library*) dengan mengimpor kelas *Image*. Terdapat beberapa variabel kunci yang ditentukan, seperti
 14 nama_kecamatan, jumlah_baris, dan jumlah_kolom, yang digunakan untuk mengidentifikasi nama
 15 kelurahan serta jumlah baris dan kolom yang digunakan untuk mengatur tata letak gambar per tile.

```

16
17 from PIL import Image
18
19 nama_kecamatan = 'nama_kecamatan/kelurahan'
20 jumlah_baris = 2
21 jumlah_kolom = 3
22
23 panjang_tile = 256
24 lebar_tile = 256
25
26 panjang_gambar = jumlah_kolom*panjang_tile
27 lebar_gambar = jumlah_baris*lebar_tile
  
```



```

12
13 canvas = Image.new("RGB", (panjang_gambar, lebar_gambar))
14 for y in range(jumlah_baris):
15     for x in range(jumlah_kolom):
16         img = Image.open(nama_kecamatan + str(y) + str(x) + ".png")
17         canvas.paste(img, (x*panjang_tile, y*lebar_tile))
18     canvas.save(nama_kecamatan + ".png")

```

Kode 3.4: Script Penggabungan Gambar

Selanjutnya, variabel `panjang_tile` dan `lebar_tile` menentukan ukuran tile, yang dalam contoh ini adalah 256 piksel. Variabel `panjang_gambar` dan `lebar_gambar` dihitung berdasarkan jumlah kolom dan baris, sehingga ukuran gambar akhir dapat ditentukan. Proses pembuatan gambar dimulai dengan inisiasi variabel `canvas` menggunakan fungsi *Image.new* dengan mode "RGB" dan ukuran gambar sesuai dengan `panjang_gambar` dan `lebar_gambar`. Selanjutnya, terdapat dua *loop* di mana *loop* pertama digunakan untuk mengatur koordinat y, dan *loop* kedua untuk koordinat x. Di dalam *loop-loop* tersebut, variabel `img` digunakan untuk membuka gambar tile yang sesuai dengan koordinatnya dengan menambahkan format file yang sesuai. Gambar yang diakses melalui variabel `img` kemudian disisipkan ke dalam gambar utuh *canvas* menggunakan fungsi *paste*. Dan terakhir, *canvas.save* untuk menyimpan hasil gambar akhir dengan nama sesuai dengan 'nama_kecamatan'. Hasil akhir gambar-gambar per tile menjadi gambar utuh yang merepresentasikan wilayah atau kecamatan yang diinginkan seperti pada gambar 1.2.

3.2 Pembentukan Gambar Hasil Segmentasi

Pada penelitian yang telah dilakukan oleh Juan Anthonius Kusjadi menghasilkan data hasil segmentasi area hijau yang telah disimpan pada sistem *data lake* yang telah dibuat pada Hadoop HDFS[1]. Pengunduhan data gambar citra satelit hasil segmentasi area hijau kelurahan dapat dilakukan dengan kode perintah seperti pada kode 3.5.

```

26
27 hdfs dfs -get /user/if18059/geodata/result/kmeans-5/arccgis/16/Jawa_Barat/
28 Kota_Bandung.csv .

```

Kode 3.5: Pengembalian Data Hasil Segmentasi

Kode Perintah 3.5 digunakan untuk mengunduh file CSV dari HDFS dan menyimpannya di direktori lokal tempat menjalankan perintah tersebut. Perintah `hdfs dfs -get` merupakan perintah untuk mengunduh file dari HDFS. Perintah `/user/if18059/geodata/result/kmeans-5/arccgis/16/Jawa_Barat/Kota_Bandung.csv` adalah jalur lengkap ke file yang ingin diunduh dari HDFS. File yang dimaksud adalah `Kota_Bandung.csv` yang terletak di direktori `/user/if18059/geodata/result/kmeans-5/arccgis/16/Jawa_Barat/` di HDFS. Dan Perintah titik (".") merupakan tempat menyimpan file yang diunduh ke direktori lokal.

File `Kota_Bandung.csv` berisikan data hasil klasterisasi. Isi dari file `Kota_Bandung.csv` memiliki 9 kolom yang dipisahkan oleh tanda titik koma (";"). Pada kolom pertama diisi dengan nama kelurahan. Pada kolom kedua diisi dengan posisi x *tile* citra satelit. Pada kolom ketiga diisi dengan posisi y *tile* citra satelit. Pada kolom keempat diisi dengan nilai panjang dari *tile* citra satelit pada kelurahan tersebut. Pada kolom kelima diisi dengan nilai lebar dari *tile* citra satelit pada kelurahan

tersebut. Pada kolom keenam diisi dengan data gambar citra satelit yang sudah disegmentasi dengan format .png berdasarkan hasil klasterisasi dan di enkripsi menggunakan Base64. Pada kolom ketujuh diisi dengan data gambar citra satelit asli dengan format .png dan di enkripsi menggunakan Base64. Pada kolom kedelapan diisi dengan nilai luas area hijau dalam km². Terakhir pada kolom kesembilan diisi dengan nilai luas area kelurahan[1].

Proses pengekstrasian file Kota_Bandung.csv berbeda dengan proses pengekstrasian yang ada pada sub-bab 3.1.2 dikarenakan format file yang berbeda maka *script* yang digunakan juga berbeda. Penggunaan script dapat pada file Kota_Bandung.csv dilihat pada kode program 3.6.

```

11 import base64
12 import csv
13
14 def extract_image_pertile(csv_file):
15     with open(csv_file, 'r') as file:
16         csv_reader = csv.reader(file)
17
18         for row in csv_reader:
19             kelurahan = row[0]
20             kordinat_x = row[1]
21             kordinat_y = row[2]
22             segmented_image_data = row[5]
23             img_data = base64.b64decode(segmented_image_data)
24             filename = f"{kelurahan}{kordinat_y}{kordinat_x}.png"
25
26             with open(filename, 'wb') as img_file:
27                 img_file.write(img_data)
28
29 if __name__ == "__main__":
30
31     csv_file = "Kota_Bandung.csv"
32     csv.field_size_limit(1000000)
33
34     extract_image_pertile(csv_file)

```

Kode 3.6: Script Penggabungan Gambar Hasil Klasterisasi

Script Python di atas dirancang untuk memproses data gambar tersegmen dalam format base64 yang terdapat dalam sebuah file Kota_Bandung.csv. *Script* ini bertujuan untuk mengekstrak dan mendekode data gambar tersebut, kemudian menyimpannya sebagai file gambar dengan format .png. Fungsi utama yang terlibat dalam proses ini disebut *extract_image_pertile*, yang akan menerima nama file .csv sebagai parameter. Fungsi tersebut membuka file Kota_Bandung.csv, lalu membaca setiap baris, dan membaca nilai-nilai yang penting seperti kelurahan, koordinat x dan y, serta data gambar tersegmen yang dienkripsi dalam base64. Selanjutnya, data gambar tersebut didekode menggunakan *library* base64 dan disimpan sebagai file gambar .png dengan nama sesuai dengan kelurahan yang terbentuk dari gabungan nilai-nilai kolom tertentu.

Dalam bagian utama *script*, terdapat pemanggilan fungsi *extract_image_pertile* dengan menyertakan nama file Kota_Bandung.csv yang akan diproses. Sebagai tambahan, script ini mengatur batas ukuran untuk file .csv dengan *library* csv yang telah disediakan oleh Python. Fungsi *fie-*

1 *ld_size_limit* untuk menangani batasan ukuran default yang ada. Dengan menjalankan *script* ini,
 2 file gambar .png akan dihasilkan untuk setiap baris. Gambar dari setiap baris tersebut merupakan
 3 gambar pertile dari tiap kelurahan.

4 Setelah mendapatkan gambar per tile langkah selanjut yaitu menggabungkan gambar. Proses
 5 penggabungan gambar segmentasi area hijau sama dengan proses penggabungan gambar kelurahan
 6 sebelum segmentasi. Pada proses tersebut dapat dilihat pada sub-bab 3.1.3.

7 Dalam pencarian data gambar hasil segmentasi ditemukan data berupa file Bandung.txt. Setelah
 8 melakukan pemrosesan pada data file Bandung.txt, file tersebut merupakan data file dari Kabupaten
 9 Bandung. File Bandung.txt juga memiliki beberapa kolom yang menunjukkan nama kelurahan, nama
 10 kabupaten, nama provinsi, panjang tile, lebar tile, koordinat tile(x,y), data poligon, dan *tile* citra
 11 satelit yang di encode menggunakan Base64. Proses pengecekan file tersebut dilakukan dengan
 12 menggunakan *script* python yang dikembangkan.

```

13
14 #cek per satu kelurahan
15 file = open("Bandung.txt", "r+")
16 counter = 0
17 while True:
18     file_line = file.readline().split(";", 8)
19
20     if not file.readline():
21         break
22
23     print(file_line[0], file_line[1], file_line[2], file_line[3], file_line[4], file_line
24           [5], file_line[6], file_line[7])
25     counter += 1
26
27     print(counter)
28
29 file.close()
30

```

Kode 3.7: Script Pengecekan File .txt

31 Setelah *script* 3.7 dijalankan ternyata terdapat data yang hilang. Data yang hilang saat melakukan
 32 pengestraksian gambar adalah beberapa *tile* pada kelurahan tersebut. *Tile-tile* yang hilang
 33 mengakibatkan gambar tidak dapat digabung menjadi sebuah gambar utuh dari kelurahan kabupaten.
 34 Contoh data dari Bandung.txt dapat dilihat pada gambar 3.5. Dari gambar tersebut bahwa nama
 35 kelurahan kabupaten adalah Kopo, dilanjutkan dengan nama kabupaten Bandung, provinsi Jawa
 36 Barat, lebar tile dan panjang tile. Pada kolom koordinat x dan y ada beberapa yang hilang. Data
 37 yang hilang adalah nilai $x = 0$, dan nilai $y = 2$ tidak terdapat pada kolom.

```

Kopo Bandung Jawa Barat 5 5 0 0
Kopo Bandung Jawa Barat 5 5 0 1
Kopo Bandung Jawa Barat 5 5 0 3
Kopo Bandung Jawa Barat 5 5 1 4
Kopo Bandung Jawa Barat 5 5 1 1
Kopo Bandung Jawa Barat 5 5 3 4
Kopo Bandung Jawa Barat 5 5 3 2
Kopo Bandung Jawa Barat 5 5 3 0
Kopo Bandung Jawa Barat 5 5 2 0
Kopo Bandung Jawa Barat 5 5 2 2
Kopo Bandung Jawa Barat 5 5 4 1
Kopo Bandung Jawa Barat 5 5 4 4
Kopo Bandung Jawa Barat 5 5 4 2

```

Gambar 3.5: Proses pengecekan file Bandung.txt

3.3 Pengumpulan Data Kelurahan

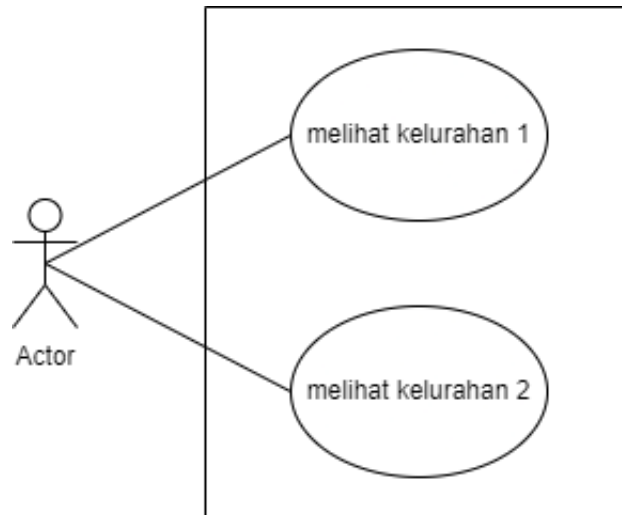
Proses pengumpulan data kelurahan yang didapatkan dari hasil penelitian yang telah dilakukan oleh Juan A. Kusjadi. Data yang ditemukan merupakan hasil perhitungan luas wilayah kelurahan dan wilayah RTH kelurahan di Kota Bandung dengan menggunakan algoritma KMeans dengan $k=5$ dalam pendekatan *pixel based*[1]. Hasil dari penelitian tersebut didapat dari lampiran hasil eksperimen yang berisikan nama kelurahan, luas kelurahan sesungguhnya dalam satuan km², luas kelurahan prediksi dalam satuan km², luas RTH kelurahan prediksi dalam satuan km², dan persentase RTH kelurahan.

Proses penginputan data dari hasil penelitian dan sumber data eksternal dilakukan secara manual dengan memasukkan data satu per satu melalui phpMyAdmin yang merupakan sebuah aplikasi berbasis web yang disediakan oleh XAMPP untuk memudahkan dalam penyimpanan basis data MySQL. Pembuatan tabel pada MySQL dinamai dengan 'data_wilayah'. Tabel tersebut berisikan kolom id sebagai *primary key*, nama_kelurahan, luas_kelurahan, luas_kelurahan_prediksi, luas_rth_kelurahan dan persentase_rth_kelurahan. Data eksternal yang ditambahkan kedalam tabel yaitu link_googlemaps yang berisikan tautan google maps sesuai dengan data kelurahan.

Dengan demikian, proses pengambilan data kelurahan telah dipindahkan kedalam basis data. Data-data tersebut nantinya akan digunakan dalam pengembangan perangkat lunak.

3.4 Analisis Perangkat Lunak

Proses analisis perangkat lunak merupakan kebutuhan yang memerlukan peranan seorang pengguna untuk menjalankan sebuah perangkat lunak yang akan dikembangkan. Sehingga segala proses sistem dijalankan oleh aktor yang terlibat. Dalam sistem ini hanya memiliki aktor sebagai *user*. Seorang pemangku kepentingan atau pembuat keputusan memegang peranan sebagai *user* itu sendiri. Dalam menggambarkan peranan pengguna terhadap interaksinya dengan sistem, maka dapat dilihat pada diagram *use case* yang terdapat pada Gambar 3.6 berikut.

Gambar 3.6: Diagram *Use Case User*

Pada Gambar 3.6, seorang aktor atau *user* pada sistem berperan dalam memegang akses penuh ke dalam sistem. Dalam hal ini *user* dapat masuk ke dalam sistem yang telah dibangun. *User* dapat memilih kelurahan pertama dan kedua yang ingin dilihat. Setiap kelurahan yang dipilih *user* akan menampilkan informasi luas wilayah, luas area hijau, kebutuhan area hijau, gambar citra satelit/gambar luas area hijau, dan juga dapat mengakses ke halaman *Google Maps* yang merujuk ke lokasi kelurahan yang dipilih.

Berdasarkan diagram *use case* pada Gambar 3.6, berikut adalah daftar skenario untuk setiap *use case*:

1. Use Case: Melihat kelurahan 1

Actor: Pengguna

Pre Condition: Pengguna telah dapat mengakses website dan berada pada halaman utama website

Post Condition: Pengguna melihat informasi dari kelurahan dipilih

Steps:

<i>Actor Actions</i>	<i>System Response</i>
Pengguna menekan pada dropdown kelurahan	Dropdown akan menampilkan daftar kelurahan Ditampilkan informasi dari kelurahan
Pengguna dapat memilih salah satu kelurahan	

2. Use Case: Melihat kelurahan 2

Actor: Pengguna

Pre Condition: Pengguna telah dapat mengakses website dan berada pada halaman utama website

Post Condition: Pengguna melihat informasi dari kelurahan dipilih

Steps:

<i>Actor Actions</i>	<i>System Response</i>
Pengguna menekan pada dropdown keluhan	Dropdown akan menampilkan daftar keluhan Ditampilkan informasi dari keluhan
Pengguna dapat memilih salah satu keluhan	

BAB 4

PERANCANGAN

Bagian ini akan menjelaskan mengenai perancangan yang digunakan dalam membangun perangkat lunak.

4.1 Perancangan Tabel Data Wilayah

Pada tabel data wilayah merupakan entitas utama dalam pengembangan perangkat lunak. Perancangan tabel data wilayah dapat dilihat dari tabel 4.1.

Tabel 4.1: Rancangan Tabel Data Wilayah

No	Atribut	Tipe	Ukuran	Primary Key	Foreign Key	Null	Keterangan
1	id	Integer	11	Yes	-	No	AUTO_INCREMENT
2	nama_kelurahan	Varchar	80	-	-	No	-
3	luas_kelurahan	Float	-	-	-	No	-
4	luas_kelurahan_prediksi	Float	-	-	-	No	-
5	luas_rth_kelurahan	Float	-	-	-	No	-
6	persentase_rth_kelurahan	Float	-	-	-	No	-
7	link_googlemapss	Varchar	-	-	-	No	-

4.2 Perancangan Kelas Controller

1. *function home*

- Masukan : -
- Keluaran: *view* homePage
- Tabel yang diakses: data_wilayah
- Deskripsi: Menampilkan halaman utama
- Algoritma: -

4.3 Perancangan Fungsi

Perancangan fungsi pada perangkat lunak berguna untuk pengembangan agar lebih dinamis dan interaktif. Dalam perancangan perangkat lunak akan terdapat *script* yang digunakan untuk pembuatan dropdown dan penggunaan radio dalam menampilkan gambar area hijau kelurahan.

Fungsi *JavaScript* 4.1 menggunakan jQuery untuk mengatur perilaku halaman web berdasarkan pilihan pengguna pada dropdown. Ketika pengguna memilih nilai pada dropdown, elemen-elemen

tertentu akan disembunyikan atau ditampilkan berdasarkan nilai yang dipilih. Fungsi ini memungkinkan untuk menyesuaikan tampilan halaman secara dinamis sesuai dengan pilihan pengguna dan meningkatkan pengalaman pengguna dalam menggunakan website.

```

1 $(document).ready(function() {
2     $(".box").hide();
3     $(".box2").hide();
4
5     $(".dropdown1").change(function() {
16     var optionValue = $(this).val();
17     if (optionValue) {
18         $(".box").not("." + optionValue).hide();
19         "." + optionValue).show();
20     } else {
21         $(".box").hide();
22     }
23 });
24
25     $(".dropdown2").change(function() {
26     var optionValue = $(this).val();
27     if (optionValue) {
28         $(".box2").not("." + optionValue).hide();
29         "." + optionValue).show();
30     } else {
31         $(".box2").hide();
32     }
33 });
34 })

```

Kode 4.1: Fungsi Dropdown

Fungsi *JavaScript 4.2* menggunakan jQuery yang memberikan fungsionalitas interaktif pada halaman web dalam mengontrol tampilan gambar berdasarkan pilihan radio yang dipilih oleh pengguna. Fungsi dapat menggunakan *event handler change* pada input radio. Oleh karena itu, pengguna dapat secara dinamis mengubah gambar yang ditampilkan sesuai dengan pilihan dan meningkatkan pengalaman pengguna dalam menggunakan website.

```

35 $(document).ready(function() {
36     $('input[name="image-radio"]').change(function() {
37         var selectedRadioId = $(this).attr('id');
38
39         $('.gambar, .gambar-segmentasi').addClass('hidden');
40
41         if (selectedRadioId === 'gambar_kelurahan') {
42             $('.gambar').removeClass('hidden');
43         } else if (selectedRadioId === 'gambar_kelurahan_segmentasi') {
44             $('.gambar-segmentasi').removeClass('hidden');
45         }
46     });
47
48     $('input[name="image-radio2"]').change(function() {
49         var selectedRadioId = $(this).attr('id');

```



```

17
18     $('gambar2, gambar-segmentasi2').addClass('hidden');
19
20     if (selectedRadioId === 'gambar_kelurahan2') {
21         $('gambar2').removeClass('hidden');
22     } else if (selectedRadioId === 'gambar_kelurahan_segmentasi2') {
23         $('gambar-segmentasi2').removeClass('hidden');
24     }
25 });
26
27 });

```

Kode 4.2: Fungsi Radio Button

4.4 Perancangan Antarmuka

Perancangan antarmuka pada perangkat lunak berguna untuk memudahkan pengguna memilih dan melihat hasil dari perbandingan kelurahan kota Bandung. Dalam gambar 4.1, terdapat desain antarmuka yang memungkinkan pengguna untuk memilih kelurahan di Kota Bandung. Pengguna dapat mengklik tautan *Google Maps* sesuai dengan pilihan kelurahan, dan pengguna juga memiliki pilihan untuk melihat citra satelit kelurahan atau memilih citra satelit yang sudah di-segmentasi.

Kelurahan	Batununggal	Ciumbuleuit
Luas Wilayah	183	150
Luas Area Hijau	35.87	90
Kebutuhan Area Hijau	90	90

Citra Satelit

Link ke Google Maps

<https://maps.app.goo.gl/65i7FUeiSXQdjXVg6>
<https://maps.app.goo.gl/V8gEVb4k3KEPtKvK7>

Layer

☒ Citra Satelit
 ☐ Area Hijau

Gambar 4.1: Rancangan Antarmuka

BAB 5

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan membahas tentang hasil dari rancangan antarmuka yang telah dibahas pada Bab 4. Pengimplementasian rancangan antarmuka agar dapat menampilkan perbandingan visual antara dua kelurahan di kota Bandung. Juga akan membahas tentang pengujian fungsional dan pengujian eksperimental.

5.1 Implementasi

Pada bagian ini akan membahas tentang perangkat keras dan perangkat lunak yang digunakan dalam membangaun perangkat lunak berdasarkan dari hasil rancangan antarmuka.

5.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam mengimplementasi rancangan perangkat lunak memiliki spesifikasi sebagai berikut:

1. Laptop : ASUS K55VD
2. *Processor* : Intel(R) Core(TM) i3-3110M CPU @2.40GHz(4CPUs), 2.4GHz
3. RAM : 12GB
4. *Solid State Drive* : 240 GB

5.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam mengimplementasi rancangan perangkat lunak memiliki spesifikasi sebagai berikut:

1. *Operating System* : Windows 10 Home Single Language
2. *Server* : web server Apache XAMPP version 3.3.0
3. *Client* : web browser Brave
4. *PHP Version* : 8.1.12
5. *DBMS* : MySQL
6. Visual Studio Code
7. Laravel

5.1.3 Implementasi Basis Data

Bagian ini terdapat implementasi basis data dalam perancangan perangkat lunak. Perangkat lunak akan memiliki data tabel berupa data_wilayah yang digunakan untuk menyimpan data-data hasil

eksperimen yang telah dilakukan kedalam database. Tabel data_wilayah memiliki kode program seperti pada 5.1

```

1 CREATE TABLE 'data_wilayah' (
2   'id' int(11) NOT NULL,
3   'nama_kelurahan' varchar(80) NOT NULL,
4   'luas_kelurahan' float NOT NULL,
5   'luas_kelurahan_prediksi' float NOT NULL,
6   'luas_rth_kelurahan' float NOT NULL,
7   'persentase_rth_kelurahan' float NOT NULL,
8   'link_googlemaps' varchar(80) NOT NULL
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
10

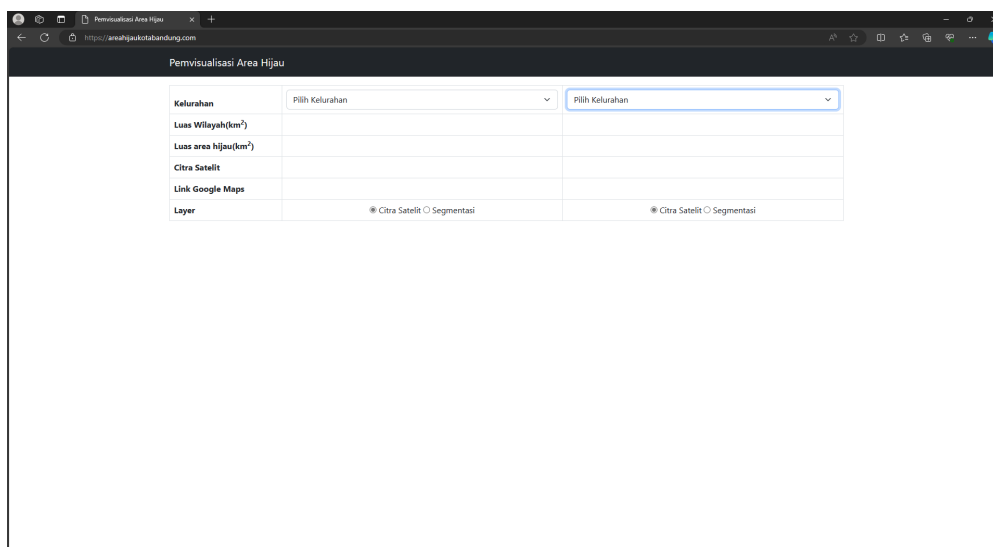
```

Kode 5.1: Implementasi Tabel data_wilayah

Lampiran B merupakan kode program yang digunakan dalam menampilkan seluruh implmentasi basis data pada perangkat lunak.

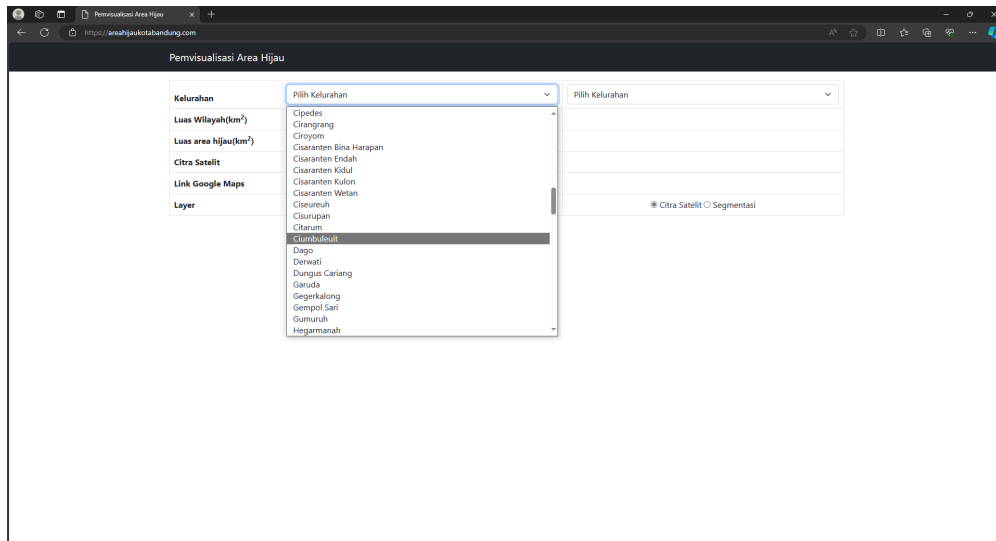
5.2 Hasil Implementasi

Implementasi perangkat lunak 'Pemvisualisasi Kelurahan Kota Bandung' dibangun dengan menggunakan bahasa pemrograman PHP dengan menggunakan *framework* Laravel. Hasil dari implementasi antarmuka saat perangkat lunak dijalankan pertama kali dapat di lihat pada gambar 5.1.

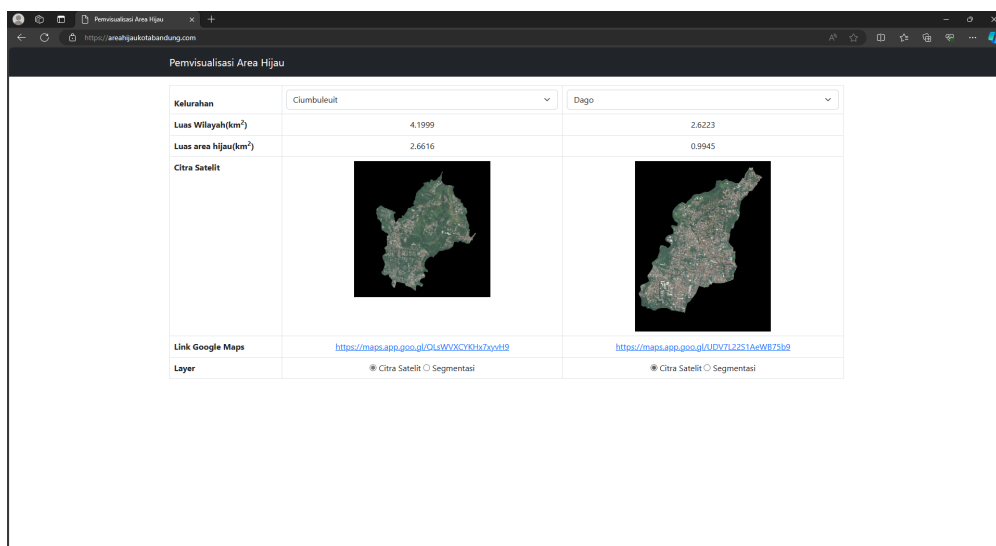


Gambar 5.1: Antarmuka Halaman Utama

Pengguna memilih kelurahan yang ingin dilihat dengan cara menekan *dropdown* yang akan menampilkan pilihan-pilihan kelurahan di kota Bandung. Implementasi antarmuka saat memilih kelurahan dapat dilihat pada gambar 5.2.

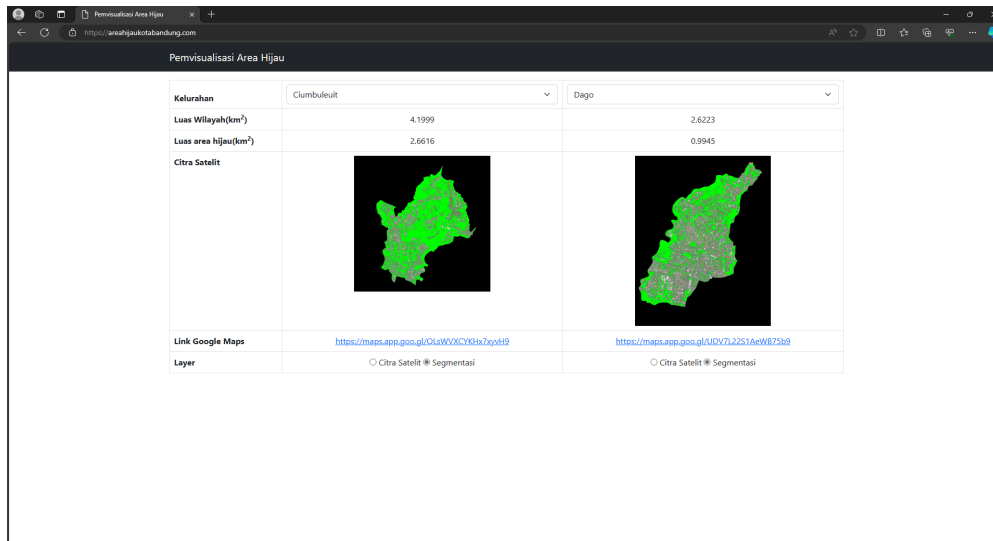
Gambar 5.2: Antarmuka Halaman Utama (*Dropdown*)

- 1 Setelah pengguna memilih kelurahan maka perangkat lunak akan menampilkan data dari setiap
- 2 kelurahan dapat dilihat pada gambar 5.3.



Gambar 5.3: Antarmuka Halaman Utama Menampilkan Data Kelurahan

- 3 Pengguna dapat memilih gambar hasil segmentasi area hijau dengan cara menekan *radio-button*
- 4 segmentasi, maka tampilan rancangan antarmuka akan dapat dilihat seperti pada gambar 5.4



Gambar 5.4: Antarmuka Halaman Utama Menampilkan Gambar Citra Satelit Segementai

5.3 Pengujian

Pada bagian ini akan dibahas mengenai hasil pengujian perangkat lunak. Pengujian perangkat lunak merupakan satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan[7]. Pengujian perangkat lunak memiliki tujuan untuk mengidentifikasi dan mengurangi potensi kesalahan, baik yang bersifat teknis maupun non-teknis. Selain itu, pengujian juga bertujuan untuk memastikan bahwa implementasi yang telah dilakukan berjalan dengan lancar, sehingga dapat diambil kesimpulan akhir mengenai kinerja perangkat lunak secara keseluruhan.

5.3.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan apa yang dibutuhkan. Hasil pengujian fungsional yang dilakukan terhadap dua buah kelurahan yaitu kelurahan Ciumbuleuit dapat dilihat pada tabel 5.1 dan kelurahan Dago dapat dilihat pada tabel 5.2

Tabel 5.1: Pengujian Fungsional Kelurahan Ciumbuleuit

Nama Fungsi	Kelurahan	Hasil yang Diharapkan	Hasil Pengujian
Memilih kelurahan	Ciumbuleuit	Dropdown berisikan nama-nama kelurahan dan dapat dipilih	Dropdown berisikan daftar kelurahan yang terurut berdasarkan abjad dan memilih kelurahan Ciumbuleuit
Menampilkan Luas Wilayah	Ciumbuleuit	Perangkat lunak akan menampilkan informasi berupa luas wilayah kelurahan yang dipilih	Perangkat lunak menampilkan informasi berupa luas wilayah kelurahan Ciumbuleuit dengan ukuran 4.1999 km ²
Menampilkan Luas Area Hijau	Ciumbuleuit	Perangkat lunak akan menampilkan informasi berupa luas area hijau kelurahan yang dipilih	Perangkat lunak menampilkan informasi berupa luas area hijau kelurahan Ciumbuleuit dengan ukuran 2.6616 km ²
Menampilkan Kebutuhan Luas Area Hijau	Ciumbuleuit	Perangkat lunak akan menampilkan informasi berupa kebutuhan luas area hijau kelurahan yang dipilih	Perangkat lunak belum bisa menampilkan informasi berupa kebutuhan luas area hijau kelurahan yang dipilih
Menampilkan Gambar Citra Satelit	Ciumbuleuit	Perangkat lunak akan menampilkan gambar dari kelurahan	Perangkat lunak menampilkan gambar dari kelurahan Ciumbuleuit
Menampilkan Link Google Maps	Ciumbuleuit	Perangkat lunak akan menampilkan link google maps dan jika ditekan akan alihkan ke halaman googlemaps sesuai dengan kelurahan yang dipilih	Perangkat lunak menampilkan link google maps dari kelurahan Ciumbuleuit dan jika ditekan akan dialihkan ke halaman google maps yang menampilkan kelurahan Ciumbuleuit
Menampilkan Gambar Segmentasi Citra Satelit	Ciumbuleuit	Perangkat lunak akan menampilkan gambar segmentasi citra satelit pada saat radio button segmentasi ditekan	Perangkat lunak menampilkan gambar segmentasi citra satelit dari kelurahan Ciumbuleuit pada saat radio button segmentasi ditekan

Tabel 5.2: Pengujian Fungsional Kelurahan Dago

Nama Fungsi	Kelurahan	Hasil yang Diharapkan	Hasil Pengujian
Memilih kelurahan	Dago	Dropdown berisikan nama-nama kelurahan dan dapat dipilih	Dropdown berisikan daftar nama kelurahan yang terurut dan memilih kelurahan Dago
Menampilkan Luas Wilayah	Dago	Perangkat lunak akan menampilkan informasi berupa luas wilayah kelurahan yang dipilih	Perangkat lunak menampilkan informasi berupa luas wilayah kelurahan Dago dengan ukuran 2.6623 km ²
Menampilkan Luas Area Hijau	Dago	Perangkat lunak akan menampilkan informasi berupa luas area hijau kelurahan yang dipilih	Perangkat lunak menampilkan informasi berupa luas area hijau kelurahan Dago dengan ukuran 0.9945 km ²
Menampilkan Kebutuhan Luas Area Hijau	Dago	Perangkat lunak akan menampilkan informasi berupa kebutuhan luas area hijau kelurahan yang dipilih	Perangkat lunak belum bisa menampilkan informasi berupa kebutuhan luas area hijau kelurahan yang dipilih
Menampilkan Gambar Citra Satelit	Dago	Perangkat lunak akan menampilkan gambar dari kelurahan	Perangkat lunak menampilkan gambar dari kelurahan Dago
Menampilkan Link Google Maps	Dago	Perangkat lunak akan menampilkan link google maps dan jika ditekan akan alihkan kehalaman googlemaps sesuai dengan kelurahan yang dipilih	Perangkat lunak menampilkan link google maps kelurahan Dago dan jika ditekan akan dialihkan ke halaman google maps yang menampilkan kelurahan Dago
Menampilkan Gambar Segmentasi Citra Satelit	Dago	Perangkat lunak akan menampilkan gambar segmentasi citra satelit pada saat radio button segmentasi ditekan	Perangkat lunak menampilkan gambar segmentasi citra satelit kelurahan Dago pada saat radio button segmentasi ditekan

5.3.2 Pengujian Eksperimental

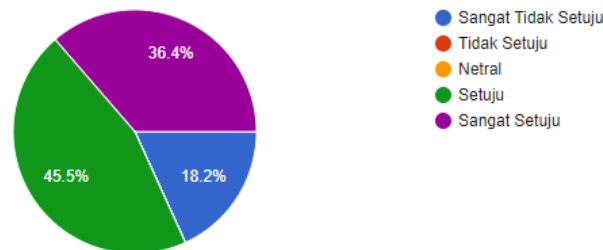
Pada tahap ini pengujian diberikan kesempatan untuk menggunakan semua fitur yang ada pada Perangkat Lunak Pemvisualisasi Hasil Penelitian Area Hijau Kelurahan. Pengujian yang melakukan pengujian ini merupakan masyarakat yang tinggal di daerah Kota Bandung. Tujuan dilakukannya pengujian ini adalah untuk menguji kemudahan dalam menjalankan perangkat lunak. Setelah pengujian menggunakan Perangkat Lunak Pemvisualisasi Hasil Penelitian Area Hijau Kelurahan Kota Bandung, pengujian diminta untuk mengisi kuesioner untuk memberikan tanggapan dan saran terhadap pengembangan perangkat lunak ini. Pernyataan pada kuesioner ini adalah sebagai berikut:

1. Perangkat lunak atau website berjalan dengan baik (tidak ada crash atau error) dan dapat menampilkan informasi tentang area hijau di kelurahan dengan mudah hijau di kelurahan dengan mudah?
2. Perangkat lunak atau website menyediakan perbandingan area hijau antara kelurahan
3. Perangkat lunak atau website memudahkan Saya untuk membandingkan informasi area hijau di beberapa kelurahan
4. Terdapat visualisasi informasi seperti gambar kelurahan yang membantu Saya memahami distribusi area hijau di kelurahan
5. Informasi tentang area hijau tersampaikan dengan baik

6. Perangkat lunak atau website ini dapat memudahkan Saya untuk mengeksplorasi informasi tentang area hijau di kelurahan
 7. Hasil yang dikeluarkan pada perangkat lunak atau website sesuai dengan harapan
 8. Di daerah manakah Anda tinggal dan Bagaimana tanggapan Anda melihat informasi berupa luas area hijau di kelurahan Anda?
 9. Saran yang ingin disampaikan
- Berdasarkan dari hasil eksperimen berikut ini hasil yang didapatkan dari penguji :
1. Gambar 5.5 merupakan tanggapan penguji terhadap *website*

Perangkat lunak atau website berjalan dengan baik (tidak ada crash atau error) dan dapat menampilkan informasi tentang area hijau di kelurahan dengan mudah

11 responses

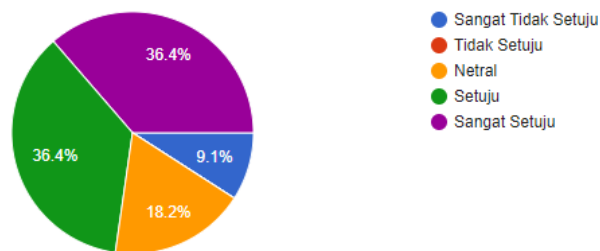


Gambar 5.5: Diagram *Pie* tanggapan penguji.

2. Gambar 5.6 merupakan tanggapan penguji terhadap penyediaan perbandingan kelurahan

Perangkat lunak atau website menyediakan perbandingan area hijau antara kelurahan

11 responses

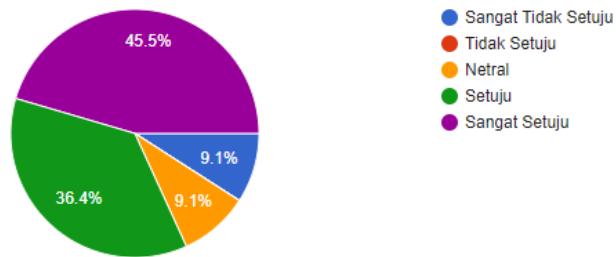


Gambar 5.6: Diagram *Pie* tanggapan penguji terhadap perbandingan kelurahan

3. Gambar 5.7 merupakan tanggapan penguji terhadap kemudahan dalam membandingkan kelurahan

Perangkat lunak atau website memudahkan Saya untuk membandingkan informasi area hijau di beberapa kelurahan

11 responses

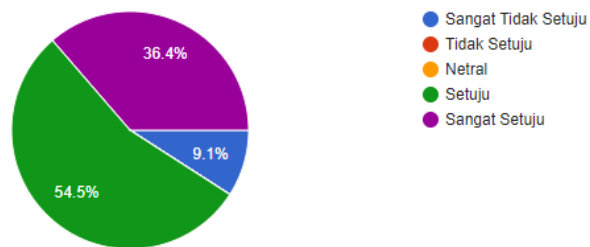


Gambar 5.7: Diagram *Pie* tanggapan penguji terhadap kemudahan pemandangan

4. Gambar 5.8 merupakan tanggapan penguji terhadap pemvisualisasian gambar kelurahan

Terdapat visualisasi informasi seperti gambar kelurahan yang membantu Saya memahami distribusi area hijau di kelurahan

11 responses

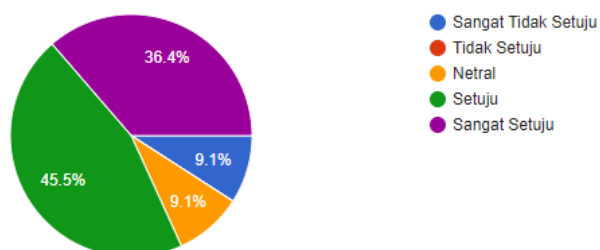


Gambar 5.8: Diagram *Pie* tanggapan pemvisualisasian gambar

5. Gambar 5.9 merupakan tanggapan penguji terhadap informasi yang diterima

Informasi tentang area hijau tersampaikan dengan baik

11 responses

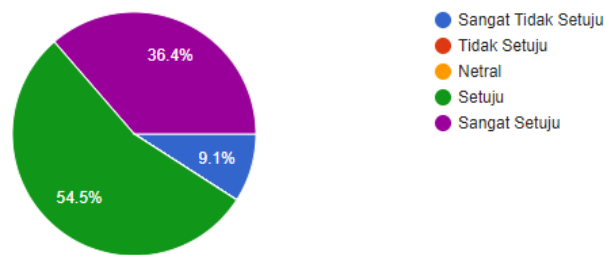


Gambar 5.9: Diagram *Pie* tanggapan informasi yang diterima

6. Gambar 5.10 merupakan tanggapan penguji terhadap pengeksplorasi informasi area hijau

Perangkat lunak atau website ini dapat memudahkan Saya untuk mengeksplorasi informasi tentang area hijau di kelurahan

11 responses

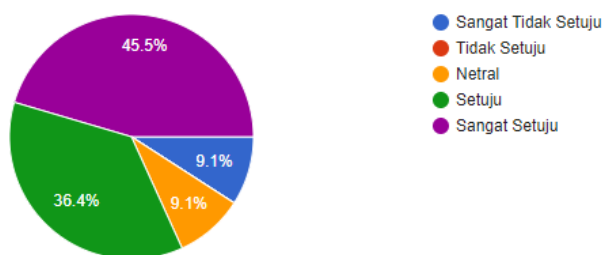


Gambar 5.10: Diagram *Pie* tanggapan eksplorasi area hijau kelurahan

7. Gambar 5.11 merupakan tanggapan penguji terhadap hasil yang diharapkan

Hasil yang dikeluarkan pada perangkat lunak atau website sesuai dengan harapan

11 responses



Gambar 5.11: Diagram *Pie* tanggapan hasil keluaran yang sesuai

8. Berikut pernyataan penguji mengenai daerah penguji tinggal dan tanggapan tentang informasi luas area hijau di daerah penguji :

- Penguji tinggal di daerah Cibiru dan tanggapan penguji yaitu kurang informasi yang didapat
- Penguji tinggal di daerah kelurahan Gegerkalon dan tanggapan penguji yaitu menambah wawasan dan kesadaran terhadap wilayah-wilayah yang memiliki area hijau di Kota Bandung.
- Penguji tinggal di kelurahan Dermawati dan tanggapan penguji yaitu membantu untuk meninjau area sekitar rumah dan mengetahui penduduk area sekitar rumah.
- Penguji tinggal di kelurahan Cihapit dan tanggapan penguji yaitu terdapat luas area hijau di Cihapit cukup banyak.
- Penguji tinggal di Ciumbuleuit dan tanggapan penguji yaitu terdapat informasi luas area hijau pada kelurahan Ciumbuleuit dapat dilihat dengan jelas.
- Penguji tinggal di Dago Pakar dan tanggapan penguji yaitu informasi yang sangat membantu untuk berbagai macam kegiatan ataupun pekerjaan.

9. Berikut merupakan saran dari penguji terhadap perangkat lunak :

- Semoga informasi area hijau ini bisa terus berkembang dan bisa diakses umum.

- 1 • Tersedia visualisasi 360°.
- 2 • Sebaiknya ditambahkan data persentase luas areahijau per kelurahan.
- 3 • Pertahankan untuk digitalisasi informasi dan upgrade secara terus menerus agar bisa
- 4 membantu masyarakat untuk mengumpulkan data.

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

1. Dari hasil penelitian yang telah dilakukan dapat disimpulkan bahwa pembuatan website interaktif telah berhasil dilakukan dalam pengembangan perangkat lunak menggunakan *framework larvel*.
2. Dari hasil penelitian yang telah dilakukan pengguna dapat menekan tombol pilih kelurahan untuk memilih kelurahan yang akan ditampilkan. Atribut-atribut yang ditampilkan berupa perbandingan luas wilayah, luas wilayah RTH, gambar citra satelit, gambar citra satelit segmentasi, dan link googlemaps dari kelurahan yang dipilih oleh pengguna.
3. Dalam penelitian yang dilakukan, pengekstraksian data citra satelit dilakukan dengan cara pengunduhan file dari hasil penelitian sebelumnya yang disimpan ke penyimpanan HDFS pribadi dan dari penyimpanan HDFS pribadi akan dilakukan pengunduhan dari local directory dengan menggunakan *command-line interface*

6.2 Saran

1. Untuk penelitian selanjutnya dapat digunakan data dari luas wilayah dan luas area hijau yang lebih baru.

DAFTAR REFERENSI

- [1] Kusjadi, J. A. (2022) Pengumpulan data citra satelit kelurahan dan perhitungan luas area hijau dengan teknologi big data. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [2] Shotts, W. (2019) *The Linux Command Line*, fifth internet edition edition. A LinuxCommand.org Book.
- [3] Alapati, S. R. (2016) *Expert Hadoop Administration: Managing, Tuning, and Securing Spark, YARN, and HDFS*, 1 edition. Addison-Wesley Professional, US.
- [4] Holmes, A. (2014) *Hadoop in Practice*, 2 edition In Practice. Manning Publications, US.
- [5] John Canning, R. L., Alan Broder (2022) *Data Structures and Algorithms in Python (Developer's Library)*, 1 edition. Addison-Wesley Professional.
- [6] Stauffer, M. (2019) *Laravel: Up and Running: A Framework for Building Modern PHP Apps*, 2 edition. O'Reilly Media.
- [7] dan M. Shalahuddin, R. A. (2011) *Modul pembelajaran rekayasa perangkat lunak (terstruktur dan berorientasi objek)*. MODULA.

LAMPIRAN A

KODE PROGRAM

```
1
2 <?php
3
4 namespace App\Http\Controllers;
5
6 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
7 use Illuminate\Foundation\Validation\ValidatesRequests;
8 use Illuminate\Routing\Controller as BaseController;
9 use Illuminate\Support\Facades\DB;
10
11
12 class Controller extends BaseController
13 {
14     use AuthorizesRequests, ValidatesRequests;
15
16     public function home(){
17
18         $data_wilayah = DB::table('data_wilayah')->get();
19         // dd($data_wilayah->nama_kelurahan);
20         return view('homePage',['data_wilayah' => $data_wilayah]);
21
22     }
23 }
24 }
```

Kode A.1: Controller.php

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Pemvisualisasi Area Hijau</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.
  css" rel="stylesheet" integrity="sha384-
  rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin
  ="anonymous">
8 <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
9 <style>
10 .hidden{
11     display:none;
12 }
13 </style>
14
15 <script>
16 $(document).ready(function() {
17     $(".box").hide();
18     $(".box2").hide();
19 }
```

```

20     $(".dropdown1").change(function() {
21         var optionValue = $(this).val();
22         if (optionValue) {
23             $(".box").not("." + optionValue).hide();
24             $(". " + optionValue).show();
25         } else {
26             $(".box").hide();
27         }
28     });
29
30     $(".dropdown2").change(function() {
31         var optionValue = $(this).val();
32         if (optionValue) {
33             $(".box2").not("." + optionValue).hide();
34             $(". " + optionValue).show();
35         } else {
36             $(".box2").hide();
37         }
38     });
39 });
40 </script>
41
42 <script>
43 $(document).ready(function() {
44     $('input[name="image-radio"]').change(function() {
45         var selectedRadioId = $(this).attr('id');
46
47         $('.gambar, .gambar-segmentasi').addClass('hidden');
48
49         if (selectedRadioId === 'gambar_kelurahan') {
50             $('.gambar').removeClass('hidden');
51         } else if (selectedRadioId === 'gambar_kelurahan_segmentasi') {
52             $('.gambar-segmentasi').removeClass('hidden');
53         }
54     });
55
56
57     $('input[name="image-radio2"]').change(function() {
58         var selectedRadioId = $(this).attr('id');
59
60         $('.gambar2, .gambar-segmentasi2').addClass('hidden');
61
62         if (selectedRadioId === 'gambar_kelurahan2') {
63             $('.gambar2').removeClass('hidden');
64         } else if (selectedRadioId === 'gambar_kelurahan_segmentasi2') {
65             $('.gambar-segmentasi2').removeClass('hidden');
66         }
67     });
68 });
69 </script>
70
71 </head>
72 <body>
73
74 @include('partials.navbar')
75
76 <div class="container mt-3">
77 @yield('container')
78 </div>
79
80
81
82

```

```

83 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlhm/jp3" crossorigin="anonymous"></script>
84 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonzOPPNlHhBs68uyIAVpII0ZZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>
85
86
87
88 </body>
89 </html>

```

Kode A.2: main.blade.php

```

1 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
2 <div class="container">
3 <a class="navbar-brand" href="#">Pemvisualisasi</a>
4 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
5 <span class="navbar-toggler-icon"></span>
6 </button>
7 <div class="collapse navbar-collapse" id="navbarNav">
8 <ul class="navbar-nav">
9 {{-- <li class="nav-item">
10 <a class="nav-link {{ ($title == "Home" ? 'active' : '') }}" href="/">Home</a>
11 </li> --}}
12 </ul>
13 </div>
14 </div>
15 </nav>

```

Kode A.3: navbar.blade.php

```

1 @extends('layout.main')
2
3 @section('container')
4
5
6 <div class="table-responsive">
7 <table class="table table-bordered table-fixed">
8 <thead>
9 <tr>
10 <th scope="col" class="col-2">Kelurahan</th>
11 <th scope="col" class="col-5">
12 <div class="dropdown-center">
13 <select class="dropdown1" name="kelurahan" value="Pilih Kelurahan">
14 <option value="" selected>Pilih Kelurahan</option>
15 @foreach($data_wilayah as $item)
16 <option value="{{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }}">
17 >{{ $item->nama_kelurahan }}</option>
18 </select>
19 </div>
20 </th>
21 <th scope="col" class="col-5">
22 <div class="dropdown-center">
23 <select class="dropdown2" name="kelurahan2">
24 <option value="" selected>Pilih Kelurahan</option>
25 @foreach($data_wilayah as $item)
26 <option value="{{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }}">
27 >{{ $item->nama_kelurahan }}</option>

```

```

27 @endforeach
28 </select>
29 </div>
30 </th>
31 </tr>
32 </thead>
33 <tbody>
34 <tr>
35 <th scope="row">Luas Wilayah(km<sup>2</sup></th>
36 <td style="text-align: center;" class="mx-auto">
37 @foreach($data_wilayah as $item)
38 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) } box"
39 >{{ $item->luas_kelurahan }}</div>
40 @endforeach
41 </td>
42 <td style="text-align: center;" class="mx-auto">
43 @foreach($data_wilayah as $item)
44 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }2 box2"
45 >{{ $item->luas_kelurahan }}</div>
46 @endforeach
47 </td>
48 </tr>
49
50 <tr>
51 <th scope="row" class="mx-auto">Luas area hijau(km<sup>2</sup></th>
52 <td style="text-align: center;">
53 @foreach($data_wilayah as $item)
54 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) } box"
55 >{{ $item->luas_rth_kelurahan }}</div>
56 @endforeach
57 </td>
58 <td style="text-align: center;">
59 @foreach($data_wilayah as $item)
60 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }2 box2"
61 >{{ $item->luas_rth_kelurahan }}</div>
62 @endforeach
63 </td>
64 </tr>
65 <tr>
66 <th scope="row" class="mx-auto">Kebutuhan Luas Area Hijau</th>
67 <td style="text-align: center;">
68 @foreach($data_wilayah as $item)
69 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) } box"
70 ></div>
71 @endforeach
72 </td>
73 </tr>
74 <tr>
75 <th scope="row" class="mx-auto">Citra Satelit</th>
76 <td style="text-align: center;" class="mx-auto">
77 @foreach($data_wilayah as $item)
78 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }2 box2"
79 ></div>
80 @endforeach
81 </td>
82 </tr>
83 <tr>
84 <th scope="row" class="mx-auto">Citra Satelit</th>
85 <td style="text-align: center;" class="mx-auto">
86 @foreach($data_wilayah as $item)
87 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) } box"
88 text-center">

```

```

83 
84
85 
86 </div>
87 @endforeach
88 </td>
89
90 <td style="text-align: center;" class="mx-auto">
91 @foreach($data_wilayah as $item)
92 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }}" box2
    text-center">
93 
94
95 
96 </div>
97 @endforeach
98 </td>
99 </tr>
100 <tr>
101 <th scope="row" class="mx-auto">Link Google Maps</th>
102 <td style="text-align: center;">
103 @foreach($data_wilayah as $item)
104 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }}" box">
105 <a href="{ $item->link_googlemaps }}" target="_blank">{ $item->link_googlemaps
    }}</a>
106 </div>
107 @endforeach
108 </td>
109
110 <td style="text-align: center;">
111 @foreach($data_wilayah as $item)
112 <div class="{ str_replace(' ', '_', strtolower($item->nama_kelurahan)) }}" box2
    ">
113 <a href="{ $item->link_googlemaps }}" target="_blank">{ $item->link_googlemaps
    }}</a>
114 </div>
115 @endforeach
116 </td>
117 </tr>
118 <tr>
119 <th scope="row" class="mx-auto">Layer</th>
120 <td style="text-align: center;">
121 <div>
122 <input type="radio" name="image-radio" id="gambar_kelurahan" checked> Citra
    Satelit
123 <input type="radio" name="image-radio" id="gambar_kelurahan_segmentasi" >
    Segmentasi
124 </div>
125 </td>
126
127 <td style="text-align: center;" class="mx-auto">
128 <div>
129 <input type="radio" name="image-radio2" id="gambar_kelurahan2" checked> Citra
    Satelit
130 <input type="radio" name="image-radio2" id="gambar_kelurahan_segmentasi2" >
    Segmentasi
131 </div>

```

```
132 </td>
133 </tr>
134 </tbody>
135 </table>
136 </div>
137
138
139
140 @endsection
```

Kode A.4: homePage.php

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\Controller;
5 use App\Http\Controllers\KelurahanController;
6 use Illuminate\Support\Facades\Http;
7 use Illuminate\Support\Facades\DB;
8
9
10 /*
11 |-----
12 | Web Routes
13 |-----
14 |
15 | Here is where you can register web routes for your application. These
16 | routes are loaded by the RouteServiceProvider and all of them will
17 | be assigned to the "web" middleware group. Make something great!
18 |
19 */
20
21
22 Route::get('/', [Controller::class, 'home'])->name('home');
```

Kode A.5: web.php

LAMPIRAN B

KODE PROGRAM BASIS DATA

```

1  -- phpMyAdmin SQL Dump
2  -- version 5.2.0
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1
6  -- Generation Time: Dec 14, 2023 at 01:28 PM
7  -- Server version: 10.4.27-MariaDB
8  -- PHP Version: 8.1.12
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Database: 'skripsi'
22 --
23
24 -- -----
25
26 --
27 -- Table structure for table 'data_wilayah'
28 --
29
30 CREATE TABLE 'data_wilayah' (
31   'id' int(11) NOT NULL,
32   'nama_kelurahan' varchar(80) NOT NULL,
33   'luas_kelurahan' float NOT NULL,
34   'luas_kelurahan_prediksi' float NOT NULL,
35   'luas_rth_kelurahan' float NOT NULL,
36   'persentase_rth_kelurahan' float NOT NULL,
37   'link_googlemaps' varchar(80) NOT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
39
40 --
41 -- Dumping data for table 'data_wilayah'
42 --
43
44 INSERT INTO 'data_wilayah' ('id', 'nama_kelurahan', 'luas_kelurahan', '
    luas_kelurahan_prediksi', 'luas_rth_kelurahan', 'persentase_rth_kelurahan', '
    link_googlemaps') VALUES
45 (1, 'Ancol', 0.7861, 0.7946, 0.1451, 18.2567, 'https://maps.app.goo.gl/
    ffhX93pQDe7Z555C8'),

```

```

46 (2, 'Antapani Kidul', 1.4316, 1.4479, 0.2755, 19.0282, 'https://maps.app.goo.gl/
AHjE4SYpQnScMVQU8'),
47 (3, 'Antapani Kulon', 0.5371, 0.5463, 0.0964, 17.6518, 'https://maps.app.goo.gl/
dgyLHb9UW3eanDKc9'),
48 (4, 'Antapani Tengah', 1.2021, 1.2174, 0.1699, 13.9533, 'https://maps.app.goo.gl/
w1AWseFJwTJGLC4h8'),
49 (5, 'Antapani Wetan', 1.0472, 1.0614, 0.2305, 21.7133, 'https://maps.app.goo.gl/
jSdLS6mjnB2Zqbpn9'),
50 (6, 'Arjuna', 0.7975, 0.8074, 0.132, 16.3517, 'https://maps.app.goo.gl/
wUnvgRXquSo1U7Wx6'),
51 (7, 'Babakan', 1.3048, 1.322, 0.0896, 6.7758, 'https://maps.app.goo.gl/
Kd3P8EB7uEv7mAgMA'),
52 (8, 'Babakan Asih', 0.3623, 0.3661, 0.0157, 4.2915, 'https://maps.app.goo.gl/6
h4SMoMGvYxbGRPY7'),
53 (9, 'Babakan Ciamis', 0.7873, 0.7969, 0.2621, 32.8914, 'https://maps.app.goo.gl/8
kGTUXeFkP5XWN2o7'),
54 (10, 'Babakan Ciparay', 1.5609, 1.5809, 0.1089, 6.8885, 'https://maps.app.goo.gl/
HoFGh8zK25Z9Tcwe6'),
55 (11, 'Babakan Penghulu', 1.7286, 1.7497, 0.3214, 18.3674, 'https://maps.app.goo.gl/
/8db2yWXAaQsFz39Z6'),
56 (12, 'Babakan Sari', 1.1354, 1.1509, 0.1235, 10.7327, 'https://maps.app.goo.gl/
FFpnsnz8kQewNm78'),
57 (13, 'Babakan Surabaya', 0.8119, 0.824, 0.1365, 16.5696, 'https://maps.app.goo.gl/
FqC1jMMuSasjAiP99'),
58 (14, 'Babakan Tarogong', 0.4291, 0.435, 0.0184, 4.2351, 'https://maps.app.goo.gl/
MSXqfsAvaU1YUjw6A'),
59 (15, 'Balong Gede', 0.5437, 0.5501, 0.115, 20.9107, 'https://maps.app.goo.gl/
uEr21QN3jNZpugqVA'),
60 (16, 'Batununggal', 1.8327, 1.8558, 0.3936, 21.2068, 'https://maps.app.goo.gl/65
i7FUeiSXQdjXVg6'),
61 (17, 'Binong', 0.5793, 0.5861, 0.088, 15.0153, 'https://maps.app.goo.gl/
DzrZ59ywCSk2bc4F9'),
62 (18, 'Braga', 0.6399, 0.6486, 0.1372, 21.1467, 'https://maps.app.goo.gl/
KxAYAfKzAqyPGBCT6'),
63 (19, 'Burangrang', 1.1061, 0.4724, 0.0963, 20.3903, 'https://maps.app.goo.gl/2
y8DaPrpiW1nDNda7'),
64 (20, 'Campaka', 0.7546, 1.0615, 0.1094, 10.3019, 'https://maps.app.goo.gl/
kcYY86uhb1d7yBju5'),
65 (21, 'Caringin', 0.6515, 0.6603, 0.0496, 7.5184, 'https://maps.app.goo.gl/
NSBHfYetF7ppWane9'),
66 (22, 'Ciateul', 0.6474, 0.6474, 0.1966, 30.37, 'https://maps.app.goo.gl/
tptqjStmGEa71R6S9'),
67 (23, 'Cibadak', 0.4833, 0.4898, 0.0475, 9.6897, 'https://maps.app.goo.gl/
NucEpR6RN6uqqMBj9'),
68 (24, 'Cibaduyut', 0.6403, 0.709, 0.0556, 7.8379, 'https://maps.app.goo.gl/
hBB6YqDSn4NTrjGc7'),
69 (25, 'Cibaduyut Kidul', 0.3065, 0.3091, 0.0204, 6.6133, 'https://maps.app.goo.gl/
rbYduGYgPcqARdvSA'),
70 (26, 'Cibaduyut Wetan', 0.7276, 0.7384, 0.0897, 12.1503, 'https://maps.app.goo.gl/
uuDth3kbEdt8yxP38'),
71 (27, 'Cibangkong', 0.4961, 0.5007, 0.0614, 12.2527, 'https://maps.app.goo.gl/
kr3gc2nQhY5NYCr7A'),
72 (28, 'Cibuntu', 0.742, 0.7522, 0.0439, 5.8383, 'https://maps.app.goo.gl/
j2yFgRP4ncwMT8NQ8'),
73 (29, 'Cicadas', 0.429, 0.4342, 0.0481, 11.0758, 'https://maps.app.goo.gl/
JFPdRD4HHpZ4nAYGJ9'),
74 (30, 'Cicaheum', 0.9194, 0.9305, 0.157, 16.8693, 'https://maps.app.goo.gl/
waQh9FSqifmvW6p98'),
75 (31, 'Cigadung', 2.4247, 2.4521, 0.8755, 35.7047, 'https://maps.app.goo.gl/
GdPtJ2wf11CtApuM9'),
76 (32, 'Cigending', 0.8431, 0.8542, 0.1939, 22.6963, 'https://maps.app.goo.gl/
EvajE7KtdrcbE73x5'),

```



```

77 (33, 'Cigereleng', 0.7768, 0.7855, 0.1434, 18.2539, 'https://maps.app.goo.gl/
    bdBjbuET4gyyavDe7'),
78 (34, 'Cigondewah Kaler', 1.2983, 1.3145, 0.0682, 5.191, 'https://maps.app.goo.gl/2
    LdAac25kHP1YQ2XA'),
79 (35, 'Cigondewah Kidul', 0.5921, 0.5996, 0.0465, 7.7555, 'https://maps.app.goo.gl/
    BqsUrXaTHFRNt3EV9'),
80 (36, 'Cigondewah Rahayu', 0.5926, 0.5997, 0.0276, 4.6088, 'https://maps.app.goo.gl
    /aZV6Pki7cntDAUTu8'),
81 (37, 'Cihapit', 1.0957, 1.1083, 0.4055, 36.5894, 'https://maps.app.goo.gl/
    y1qa9VwhycxdVNdr7'),
82 (38, 'Cihaurgeulis', 0.6471, 0.6556, 0.1312, 20.0095, 'https://maps.app.goo.gl/5
    HPJXXUNJFVMcJPNA'),
83 (39, 'Cijagra', 1.1061, 1.1182, 0.2701, 24.1562, 'https://maps.app.goo.gl/
    G4afP1ZWzJmWicZr7'),
84 (40, 'Cijaura', 1.6188, 1.6396, 0.3409, 20.7898, 'https://maps.app.goo.gl/
    UkckRaMTFRcn257U8'),
85 (41, 'Cijerah', 0.9431, 0.957, 0.0627, 6.5465, 'https://maps.app.goo.gl/
    aWJQMJS5EhEnERAct9'),
86 (42, 'Cikawao', 0.3601, 0.3649, 0.06, 16.4349, 'https://maps.app.goo.gl/
    zrj4KiYer5oDmGga7'),
87 (43, 'Cikutra', 0.7806, 0.7895, 0.1017, 12.8829, 'https://maps.app.goo.gl/555
    ar5vRc6GbDVHJ6'),
88 (44, 'Cimenerang', 1.8934, 0.7895, 0.1017, 12.8829, 'https://maps.app.goo.gl/4
    PHnUPuVve9FB7Bx7'),
89 (45, 'Cipadung', 1.3355, 1.3513, 0.3503, 25.9199, 'https://maps.app.goo.gl/
    eEEbc2YaTvMcPjor6'),
90 (46, 'Cipadung Kidul', 1.521, 1.541, 0.3155, 20.4742, 'https://maps.app.goo.gl/
    VcRoYzxBLyYprjob7'),
91 (47, 'Cipadung Kulon', 1.0872, 1.099, 0.2342, 21.3074, 'https://maps.app.goo.gl/
    aNALNuTv74uScbgN6'),
92 (48, 'Cipadung Wetan', 0.9104, 0.9226, 0.2634, 28.5501, 'https://maps.app.goo.gl/8
    Yfmk5dCD8T64ro37'),
93 (49, 'Cipaganti', 0.6825, 0.6962, 0.1936, 27.8143, 'https://maps.app.goo.gl/
    WKb56TEC5zTvkvY89'),
94 (50, 'Cipamokolan', 1.9779, 2.0196, 0.439, 21.7352, 'https://maps.app.goo.gl/8
    c1Vg984xTX4MBBAA'),
95 (51, 'Cipedes', 0.9541, 0.9639, 0.1783, 18.496, 'https://maps.app.goo.gl/
    tEFrVBCqPvudDHDk7'),
96 (52, 'Cirangrang', 0.8205, 0.8438, 0.0591, 6.9979, 'https://maps.app.goo.gl/
    DTuBxWqJjnJRMzYv5'),
97 (53, 'Ciroyom', 0.62, 0.628, 0.0575, 9.1494, 'https://maps.app.goo.gl/
    Yapymy8HTmee3Rag7'),
98 (54, 'Cisaranten Bina Harapan', 1.0533, 1.0663, 0.3247, 30.449, 'https://maps.app.
    goo.gl/2wujMrsTP46pbtAe9'),
99 (55, 'Cisaranten Endah', 1.6293, 1.6516, 0.3364, 20.3671, 'https://maps.app.goo.gl
    /CceA8DYi8daJKN9E9'),
100 (56, 'Cisaranten Kidul', 4.1126, 1.6516, 0.3364, 20.3671, 'https://maps.app.goo.gl
    /TV2CgKNtLEtAePxZ9'),
101 (57, 'Cisaranten Kulon', 2.1651, 2.1921, 0.3804, 17.3528, 'https://maps.app.goo.gl
    /jpxH3fopF5XEVQtT6'),
102 (58, 'Cisaranten Wetan', 0.6072, 0.6149, 0.1281, 20.833, 'https://maps.app.goo.gl/
    niXmKh7KyEq7gvFj7'),
103 (59, 'Ciseureuh', 0.6914, 0.7, 0.1039, 14.8459, 'https://maps.app.goo.gl/
    crTqPwn87TyRAATR7'),
104 (60, 'Cisurupan', 2.3029, 2.458, 1.4618, 59.4709, 'https://maps.app.goo.gl/
    mR6cvYk4nhKR4iiCA'),
105 (61, 'Citarum', 1.2892, 1.3056, 0.544, 41.6668, 'https://maps.app.goo.gl/
    tr4WBjShXRxidHQDA'),
106 (62, 'Ciumbuleuit', 4.1999, 4.2497, 2.6616, 62.6299, 'https://maps.app.goo.gl/
    QLsWVXCyKHx7xyvH9'),
107 (63, 'Dago', 2.6223, 2.6551, 0.9945, 37.4579, 'https://maps.app.goo.gl/
    UDV7L22S1AeWB75b9'),

```

```

108 (64, 'Derwati', 1.7371, 1.7594, 0.7727, 43.9183, 'https://maps.app.goo.gl/
    rASGKNLYx4SW5i1k9'),
109 (65, 'Dungus Cariang', 0.6464, 0.6544, 0.0595, 9.0906, 'https://maps.app.goo.gl/
    oMCyHaBi3ygoekAw5'),
110 (66, 'Garuda', 0.4587, 0.4629, 0.0562, 12.1339, 'https://maps.app.goo.gl/
    MZ7FDSt35XYAD6Sw9'),
111 (67, 'Gegerkalong', 1.8574, 1.8855, 0.4505, 23.89, 'https://maps.app.goo.gl/
    rASGKNLYx4SW5i1k9'),
112 (68, 'Gempol Sari', 1.1758, 1.1937, 0.0346, 2.8959, 'https://maps.app.goo.gl/
    TBAPtKcpgKUnvfJH7'),
113 (69, 'Gumuruh', 0.8895, 0.9016, 0.1654, 18.3447, 'https://maps.app.goo.gl/3
    GErJhH8sZj3BoaK8'),
114 (70, 'Hegarmanah', 1.8852, 1.9099, 0.6872, 35.9809, 'https://maps.app.goo.gl/
    FcP8hWvWiso3QmLA7'),
115 (71, 'Husein Sastranegara', 2.7406, 2.7739, 0.339, 12.2227, 'https://maps.app.goo.
    gl/tgQLPZg3RVpMKHHC9'),
116 (72, 'Isola', 1.9591, 1.9848, 0.7991, 40.2597, 'https://maps.app.goo.gl/
    G92AgCthPJ5oUkFQ8'),
117 (73, 'Jamika', 0.5067, 0.512, 0.0351, 6.8488, 'https://maps.app.goo.gl/
    PLgd2M3Y3MQD35TF7'),
118 (74, 'Jati Sari', 1.1633, 1.1767, 0.3154, 26.808, 'https://maps.app.goo.gl/
    XgsHuNcx2jMTaQwX7'),
119 (75, 'Jatihandap', 1.6682, 1.6894, 0.6005, 35.5462, 'https://maps.app.goo.gl/
    TrLhtCTnnXcMu8qp9'),
120 (76, 'Kacapiring', 0.8738, 0.8853, 0.2299, 25.9655, 'https://maps.app.goo.gl/
    uNCLYYfd5zbfMJ318'),
121 (77, 'Karang Anyar', 0.4003, 0.4065, 0.0671, 16.4981, 'https://maps.app.goo.gl/9
    kHYEhe1pp1ehmDH8'),
122 (78, 'Karang Pamulang', 1.4673, 1.4903, 0.5478, 36.7593, 'https://maps.app.goo.gl/
    oSckgZeeoo6n9Rfy7'),
123 (79, 'Karasak', 0.4664, 0.4723, 0.0563, 11.9246, 'https://maps.app.goo.gl/5
    Ygiinyqa5CZEmvM6'),
124 (80, 'Kebon Gedang', 0.2225, 0.2248, 0.0181, 8.0436, 'https://maps.app.goo.gl/
    T7Pp2V7doJ4uaTRy8'),
125 (81, 'Kebon Jayanti', 0.279, 0.2826, 0.0214, 7.5719, 'https://maps.app.goo.gl/
    KCw53NEcF9yq2qxGA'),
126 (82, 'Kebon Jeruk', 0.8082, 0.818, 0.1139, 13.922, 'https://maps.app.goo.gl/
    nn4zm4juMAaCpMen7'),
127 (83, 'Kebon Kangkung', 0.5595, 0.5649, 0.0889, 15.7357, 'https://maps.app.goo.gl/
    TFQZojC3aJFHotxr7'),
128 (84, 'Kebon Lega', 1.3419, 1.3586, 0.1668, 12.28, 'https://maps.app.goo.gl/
    oNBmV2wdeuiK2jAL9'),
129 (85, 'Kebon Pisang', 0.6063, 0.6136, 0.1323, 21.567, 'https://maps.app.goo.gl/
    nvx8rJGck5E8mMjUA'),
130 (86, 'Kebon Waru', 0.8925, 0.9032, 0.1299, 14.3799, 'https://maps.app.goo.gl/
    n9EkDgcQz7Xy5E599'),
131 (87, 'Kopo', 0.8735, 0.8856, 0.0567, 6.4046, 'https://maps.app.goo.gl/
    fP8WmgbTW8GKpzW6A'),
132 (88, 'Kujangsari', 1.1114, 1.1245, 0.2814, 25.0224, 'https://maps.app.goo.gl/
    y5QFj8kbNrtQqFUM6'),
133 (89, 'Lebak Gede', 0.9722, 0.9848, 0.3494, 35.4798, 'https://maps.app.goo.gl/
    rueZbwVUPT4JTsoY6'),
134 (90, 'Lebak Siliwangi', 1.0992, 1.1144, 0.6223, 55.838, 'https://maps.app.goo.gl/
    AGdn5PXbSb1Lk8Ar6'),
135 (91, 'Ledeng', 1.5876, 1.6067, 0.7724, 48.0719, 'https://maps.app.goo.gl/
    NVPkfJx5te4TDS3g7'),
136 (92, 'Lingkar Selatan', 1.1061, 1.2858, 0.3527, 27.4276, 'https://maps.app.goo.gl/
    Ldx7VMExy5m8PRC76'),
137 (93, 'Malabar', 0.683, 0.69, 0.1717, 24.8889, 'https://maps.app.goo.gl/4
    dcW5rCncJnPyqGt6'),
138 (94, 'Maleber', 0.6561, 0.6651, 0.0401, 6.0272, 'https://maps.app.goo.gl/
    RD5tsGj4FiSAqc476'),

```

139 (95, 'Maleer', 0.4458, 0.451, 0.0613, 13.5982, 'https://maps.app.goo.gl/8
fuCfzLL7Xf55jps8'),
 140 (96, 'Manjahlega', 1.4596, 1.4781, 0.3132, 21.1893, 'https://maps.app.goo.gl/9
M3WgBbEajwTRHE77'),
 141 (97, 'Margahayu Utara', 1.299, 1.3141, 0.0737, 5.6066, 'https://maps.app.goo.gl/
gDNfitdRvotmu9Ny8'),
 142 (98, 'Margasari', 2.4336, 2.461, 0.4702, 19.1077, 'https://maps.app.goo.gl/
VcNjN6o8Ncx4fJip9'),
 143 (99, 'Margasuka', 1.1792, 1.1948, 0.103, 8.6238, 'https://maps.app.goo.gl/
rXKZ9JDbzcN1Z8ES8'),
 144 (100, 'Mekar Mulya', 1.6593, 1.5717, 0.295, 18.7676, 'https://maps.app.goo.gl/
a4pmEcGgnjexHDZt8'),
 145 (101, 'Mekarmulya', 1.5535, 1.5717, 0.295, 18.7676, 'https://maps.app.goo.gl/
P3ULyM6TnSUPQGgx8'),
 146 (102, 'Mekar Wangi', 1.0922, 1.1073, 0.1676, 15.1351, 'https://maps.app.goo.gl/
UrNdRcNn4ve8eoZv5'),
 147 (103, 'Mengger', 1.4452, 1.4624, 0.368, 25.1623, 'https://maps.app.goo.gl/
nKKUL5p4AdZnuVQv7'),
 148 (104, 'Merdeka', 1.4063, 1.4224, 0.4845, 34.0628, 'https://maps.app.goo.gl/
xv9qW9MC5f33AyJ99'),
 149 (105, 'Neglasari', 0.7776, 0.7854, 0.2371, 30.1868, 'https://maps.app.goo.gl/
ajAKY8pcU2f4ucV28'),
 150 (106, 'Nyengseret', 0.3376, 0.3412, 0.0447, 13.1134, 'https://maps.app.goo.gl/
cmk9TwWK2hentPm28'),
 151 (107, 'Padasuka', 0.5451, 0.5524, 0.0616, 11.151, 'https://maps.app.goo.gl/
QnuA4GMmRkU5JWPU6'),
 152 (108, 'Pajajaran', 1.0109, 1.0232, 0.1616, 15.7934, 'https://maps.app.goo.gl/
R1EHF8WPSkUHyF4GA'),
 153 (109, 'Pakemitan', 1.5335, 1.5531, 0.5774, 37.1761, 'https://maps.app.goo.gl/
S4GUxVJbgipzCMpUA'),
 154 (110, 'Palasari', 1.8968, 1.9191, 0.878, 45.752, 'https://maps.app.goo.gl/
zVpTcy83MCGLd5dQ9'),
 155 (111, 'Paledang', 0.3613, 0.366, 0.0567, 15.5021, 'https://maps.app.goo.gl/1
A96F9AVSokovCi79'),
 156 (112, 'Pamoyanan', 0.6309, 0.6394, 0.1411, 22.0666, 'https://maps.app.goo.gl/
utzFMX8E266HcJnDA'),
 157 (113, 'Panjuran', 0.2927, 0.296, 0.0617, 20.8296, 'https://maps.app.goo.gl/
SC4hHfjXcjRvo9A1A'),
 158 (114, 'Pasanggrahan', 1.4596, 1.5036, 0.5897, 39.2226, 'https://maps.app.goo.gl/
Tb668krqVD8peFDN8'),
 159 (115, 'Pasir Biru', 1.1597, 1.306, 0.4499, 34.444, 'https://maps.app.goo.gl/
UwqrULAiicLVq5na9'),
 160 (116, 'Pasir Endah', 0.9365, 0.9475, 0.208, 21.9488, 'https://maps.app.goo.gl/
yVMBUAFg4qHdSS4D9'),
 161 (117, 'Pasir Impun', 0.8104, 0.8218, 0.3172, 38.6006, 'https://maps.app.goo.gl/
M7KSGmPBrj5CzZ4p7'),
 162 (118, 'Pasir Wangi', 1.2171, 0.8218, 0.3172, 38.6006, 'https://maps.app.goo.gl/
hqAW9sbrsgsfpgJA'),
 163 (119, 'Pasirjati', 1.4968, 1.5157, 0.6172, 40.7196, 'https://maps.app.goo.gl/
Wt53Mo3UNu2LMXXe6'),
 164 (120, 'Pasirkaliki', 1.0574, 1.0697, 0.2749, 25.7021, 'https://maps.app.goo.gl/
g5GB76QtsVK5ixy17'),
 165 (121, 'Pasirlayung', 0.9943, 1.0066, 0.1949, 19.3621, 'https://maps.app.goo.gl/
M5hJRoN1VoMhFy1U9'),
 166 (122, 'Pasirluyu', 1.0232, 1.0411, 0.1873, 17.9924, 'https://maps.app.goo.gl/
Eacw2cZoj2zxeBkY8'),
 167 (123, 'Pasteur', 1.1916, 1.2018, 0.3139, 26.1238, 'https://maps.app.goo.gl/
kgzytm5aasfCtwWR6'),
 168 (124, 'Pelindung Hewan', 0.6599, 0.6683, 0.0969, 14.5069, 'https://maps.app.goo.gl/
/Cv54XuMncrwHag6K7'),
 169 (125, 'Pungkur', 0.3958, 0.3995, 0.0642, 16.0721, 'https://maps.app.goo.gl/
ZodNjNtAnNLaoKKt5'),

```

170 (126, 'Rancabolang', 4.1126, 2.6521, 0.6876, 25.9279, 'https://maps.app.goo.gl/
    PyTX783DuaQPZK889'),
171 (127, 'Rancanumpang', 1.1537, 1.1682, 0.1881, 16.0974, 'https://maps.app.goo.gl/6
    V21ipTfifhKjYfN6'),
172 (128, 'Sadang Serang', 0.7181, 0.7313, 0.0996, 13.6224, 'https://maps.app.goo.gl/
    SNQEjwkY5NSecZNj7'),
173 (129, 'Samoja', 0.3569, 0.3622, 0.0549, 15.1444, 'https://maps.app.goo.gl/
    e3ppcNqNs7tF3Jm99'),
174 (130, 'Sarijadi', 1.2538, 1.2694, 0.2365, 18.6342, 'https://maps.app.goo.gl/6
    heDqMgwvQuZVFqN6'),
175 (131, 'Sekejati', 2.1434, 2.1702, 0.6031, 27.7881, 'https://maps.app.goo.gl/
    ffhX93pQDe7Z555C8'),
176 (132, 'Sekeloa', 2.6223, 1.1052, 0.2092, 18.9303, 'https://maps.app.goo.gl/9
    sCcArvefKXVnQ7X9'),
177 (133, 'Sindang Jaya', 0.7563, 0.7801, 0.1865, 23.9088, 'https://maps.app.goo.gl/
    USPZJRWzHpuKgeKr8'),
178 (134, 'Situsaaur', 0.7695, 0.7797, 0.0979, 12.5509, 'https://maps.app.goo.gl/
    bPthZDi3nNfdV9Z56'),
179 (135, 'Suka Asih', 0.8917, 0.9032, 0.1055, 11.6832, 'https://maps.app.goo.gl/
    GtqBeVBHLWVzCfJ16'),
180 (136, 'Sukabungah', 0.4992, 0.5048, 0.0508, 10.0644, 'https://maps.app.goo.gl/5
    vDdnFxZq437GuCu6'),
181 (137, 'Sukagalih', 1.6038, 1.6237, 0.3809, 23.459, 'https://maps.app.goo.gl/
    cqDyLG4BgArKQ2Cg6'),
182 (138, 'Sukahaji', 0.8528, 0.8619, 0.0597, 6.925, 'https://maps.app.goo.gl/
    pU1yeB95cTvTNW1Q8'),
183 (139, 'Sukaluyu', 0.7276, 0.7366, 0.1178, 15.9982, 'https://maps.app.goo.gl/
    tHCJUrazYr2Qme4j8'),
184 (140, 'Sukamaju', 0.5102, 0.5163, 0.1137, 22.015, 'https://maps.app.goo.gl/
    qgfGVGfjqxT7638TA'),
185 (141, 'Sukamiskin', 2.4874, 2.5166, 0.6071, 24.1243, 'https://maps.app.goo.gl/2
    W7yS8ascHuTVgR36'),
186 (142, 'Sukamulya', 0.3495, 0.3536, 0.0544, 15.3778, 'https://maps.app.goo.gl/
    D6gNxLaGtakUgx2V7'),
187 (143, 'Sukapada', 0.8342, 0.8449, 0.1504, 17.7957, 'https://maps.app.goo.gl/
    Be7MaAatPv9gmbeB9'),
188 (144, 'Sukapura', 1.9864, 2.0145, 0.7958, 39.5054, 'https://maps.app.goo.gl/
    FJhQhynP36VvoLSY9'),
189 (145, 'Sukaraja', 1.3007, 1.3155, 0.1625, 12.3509, 'https://maps.app.goo.gl/
    dPyNZetyQEQfXsvU7'),
190 (146, 'Sukarasa', 1.2633, 1.2784, 0.3325, 26.0126, 'https://maps.app.goo.gl/
    J18NKNRwaHhF2dfr9'),
191 (147, 'Sukawarna', 0.8878, 0.8994, 0.166, 18.4564, 'https://maps.app.goo.gl/
    Qjbbon5BKyoYtRXs8'),
192 (148, 'Taman Sari', 1.0117, 1.024, 0.2475, 24.1667, 'https://maps.app.goo.gl/
    AarjDTDU5jzfzPGHA'),
193 (149, 'Turangga', 1.1061, 1.5753, 0.3384, 21.4826, 'https://maps.app.goo.gl/
    ohvUbNS7PWH25zib8'),
194 (150, 'Warung Muncang', 0.714, 0.7234, 0.0658, 9.0988, 'https://maps.app.goo.gl/
    ER8d6omFH9LxWkg96'),
195 (151, 'Wates', 0.5994, 0.6077, 0.0844, 13.8814, 'https://maps.app.goo.gl/
    hU6A9dxv6oshGKAN9');
196
197 --
198 -- Indexes for dumped tables
199 --
200
201 --
202 -- Indexes for table 'data_wilayah'
203 --
204 ALTER TABLE 'data_wilayah'
205 ADD PRIMARY KEY ('id');
206

```

```
207 --
208 -- AUTO_INCREMENT for dumped tables
209 --
210
211 --
212 -- AUTO_INCREMENT for table 'data_wilayah'
213 --
214 ALTER TABLE 'data_wilayah'
215 MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=152;
216 COMMIT;
217
218 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
219 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
220 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```