

SKRIPSI

**PEMVISUALISASI HASIL PENELITIAN AREA HIJAU
KELURAHAN**



Bosnich Timothy Bonasleng

NPM: 2017730086

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

«tahun»

UNDERGRADUATE THESIS

«JUDUL BAHASA INGGRIS»



Bosnich Timothy Bonasleng

NPM: 2017730086

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

«tahun»

LEMBAR PENGESAHAN

PEMVISUALISASI HASIL PENELITIAN AREA HIJAU KELURAHAN

Bosnich Timothy Bonasleng

NPM: 2017730086

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing

Pascal Alfadian, Nugroho, M.Comp.

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PEMVISUALISASI HASIL PENELITIAN AREA HIJAU KELURAHAN

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» «tahun»

Meterai Rp. 10000

Bosnich Timothy Bonasleng
NPM: 2017730086

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»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	3
2.1 <i>Command-line Interface</i>	3
2.1.1 SCP(<i>Secure Copy Protocol</i>)	4
2.2 Hadoop Distributed File System	4
2.3 <i>Python</i>	5
2.3.1 <i>Pillow (PIL Fork)</i>	5
2.4 <i>Base64</i>	7
2.5 Framework	8
2.5.1 Framework Laravel	8
DAFTAR REFERENSI	11
A KODE PROGRAM	13
B HASIL EKSPERIMEN	15

DAFTAR GAMBAR

2.1	<i>Command Prompt</i>	3
2.2	Pemanggilan SCP	4
2.3	Perintah HDFS CLI	4
2.4	Perintah HDFS dfs	5
2.5	Perintah HDFS dfs -get untuk mengunduh file HDFS Kota_Bandung.txt ke <i>local system</i>	5
2.6	Perintah HDFS dfs -ls untuk menampilkan isi file yang berada didalam <i>directory Jawa_Barat</i>	5
2.7	Pemanggilan fungsi <i>open()</i>	6
2.8	Pemanggilan fungsi <i>Image.new()</i>	6
2.9	Pemanggilan fungsi <i>Image.paste()</i>	7
2.10	PHP Artisan Laravel	8
B.1	Hasil 1	15
B.2	Hasil 2	15
B.3	Hasil 3	15
B.4	Hasil 4	15

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan tempat tinggal bagi masyarakat kota diperlukannya oksigen untuk hidup dan beraktivitas. Agar dapat memenuhi kebutuhan tempat tinggal bagi masyarakat maka pembangunan dan perkembangan pada suatu wilayah harusnya memiliki fungsi lain sebagai Ruang Terbuka Hijau(RTH). RTH merupakan sebuah area yang memanjang atau jalur dan/atau mengelompok,dan penggunaannya yang lebih bersifat terbuka, tempat tumbuh tanaman, baik yang tumbuh secara alamiah maupun yang sengaja ditanam. Salah satu penyumbang oksigen yang besar bagi kota adalah RTH.Tentu saja seharusnya RTH tersedia dalam jumlah yang cukup, terutama pada kota yang memiliki penduduk yang banyak.

Pemanfaatan citra satelit merupakan salah satu 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 pengidentifikasi RTH yang mana terdapatnya 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 telah dilakukan Dr. Veronica Sri Moertini, Fritz H. Hutapea SKom, dan Juan A. Kusjadi menghasilkan data area hijau dari citra satelit pada Kota Bandung yang dibagi menjadi beberapa kelurahan atau kecamatan Kota Bandung. Hasil dari penelitian terdiri dari area hijau untuk 149 kelurahan di 30 kecamatan kota Bandung dan telah selesai dilakukan perhitungan.

Pada Skripsi ini, akan dibangun sebuah laman web yang bertujuan untuk pemvisualisasian dari hasil penelitian area hijau Kota Bandung .Laman web yang akan dibangun harusnya dapat diakses melalui komputer atau laptop,handphone bersistem android ataupun iOS. Dan dalam pembuatanya akan dibantu dengan menggunakan *Framework* Laravel, sehingga memudahkan pengembang untuk membangun laman web.Dengan adanya laman web pemvisualisasian ini juga akan memudahkan pengguna untuk mengetahui area-area hijau yang ada pada Kota Bandung.

1.2 Rumusan Masalah

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

- Bagaimana membuat sebuah laman web interaktif yang dapat membandingkan data dua buah kelurahan atau kecamatan Kota Bandung?
- Bagaimana cara pengguna untuk membandingkan atribut-atribut Citra Satelit dari kelurahan atau kecamatan Kota Bandung?
- Bagaiaman cara mengekstasi data citra satelit pada HDFS ke *local directory*?

1.3 Tujuan

Tujuan dari skripsi ini adalah:

1. Membuat sebuah laman web interaktif yang dapat membandingkan dua buah kelurahan.
2. Pengguna dapat memilih kecamatan atau 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

Batasan-batasan masalah untuk penelitian ini adalah sebagai berikut:

1. Penelitian dari data yang sudah matang.

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 laman web seperti Ruang terbuka hijau, Citra Satelit, dan *Framework*.

BAB 2

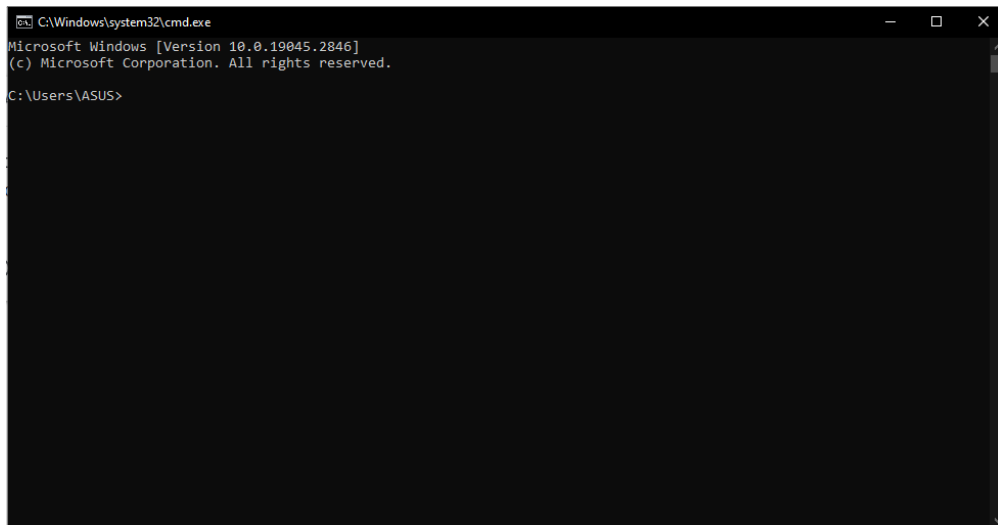
LANDASAN TEORI

Pejelasan tentang teori-teori yang perlu diketahui sebelum pembuatan laman web.

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 komputer. *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
2. Program yang dapat dijalankan ketika berhasil dipanggil, dan menjalankan aplikasi yang berbasis teks atau grafis.
3. *batch program* (*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 antaralain :

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. Perbedaannya yang paling terlihat terletak pada sumber atau tujuan ke *remote host*. Sebagai contoh, jika ingin menyalin sebuah dokumen dari *home directory*(berkas dalam komputer) ke *remote system*, atau dari *working directory* ke sistem lokal.

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

Gambar 2.2: Pemanggilan SCP

Gambar 2.2 merupakan contoh penyalinan berkas Kota_Bandung.txt. Berkas tersebut yang tersimpan didalam *remote host* dan disalin ke *local system* 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.

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.

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

```
C:\Users\ASUS>hdfs [SHELL_OPTIONS] COMMAND [GENERIC_OPTIONS] [COMMAND_OPTIONS]_
```

Gambar 2.3: Perintah HDFS CLI

Gambar 2.3 merupakan pemanggilan perintah pada HDFS. Setiap opsi perintah memiliki fungsi untuk menjalankan *script* pada CLI. Penjelasan tiap opsi dijelaskan pada tabel 2.1.

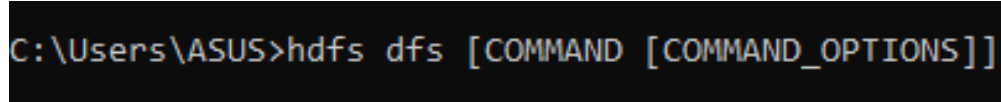
Tabel 2.1: *Hadoop* memiliki opsi *parsing framework* yang menjelaskan setiap fungsi kelasnya

COMMAND_OPTION	Deskripsi
SHELL_OPTIONS	kumpulan shell_option yang umum
GENERIC_OPTIONS	kumpulan generic_option yang didukung oleh beberapa perintah
COMMAND COMMAND_OPTIONS	Berbagai 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 2.4

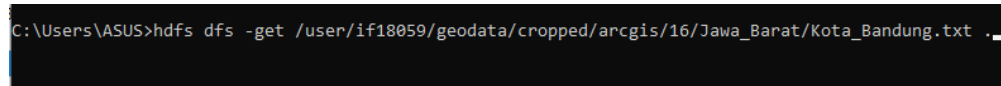


```
C:\Users\ASUS>hdfs dfs [COMMAND [COMMAND_OPTIONS]]
```

Gambar 2.4: Perintah HDFS dfs

2. Penggunaan perintah **get**

Perintah **get** digunakan untuk menyalin file HDFS ke *local system*. Gambar 2.5 merupakan contoh yang menunjukkan cara penggunaan perintah **-get** untuk mengunduh file dari HDFS ke *local file system*

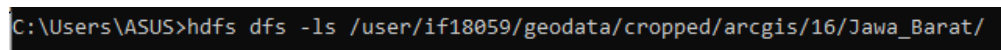


```
C:\Users\ASUS>hdfs dfs -get /user/lf18059/geodata/cropped/arcgis/16/Jawa_Barat/Kota_Bandung.txt -
```

Gambar 2.5: 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/lf18059/geodata/cropped/arcgis/16/Jawa_Barat/
```

Gambar 2.6: Perintah HDFS dfs -ls untuk menampilkan isi file yang berada didalam *directory* Jawa_Barat

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. Bahasa pemrograman *Python* Merupakan sebuah bahasa pemrograman komputer yang dikembangkan khusus untuk membuat *source code* yang mudah untuk dibaca. *Python* memiliki *library* yang lengkap sehingga memudahkan seorang *programmer* untuk membuat sebuah aplikasi sesuai dengan keinginan dengan menggunakan *source code* yang terlihat sederhana.

2.3.1 Pillow (PIL Fork)

Python Imaging Library merupakan salah satu *library* yang terdapat pada bahasa pemrograman *Python*. *PIL* dapat menambahkan pemrosesan gambar ke bahasa pemrograman *Python*. *Library* ini menyediakan *extensive file format*, representasi internal yang efisien, dan memiliki kemampuan yang baik dalam pemrosesan gambar. Pentingnya *library* yang dirancang untuk dapat mengakses data

yang disimpan dengan cepat dalam berbagai format piksel. Seharusnya memberikan dasar yang kuat sebagai alat pengolahan gambar

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

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

Penggunaan kelas *Image*

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

1. Penggunaan fungsi *Image.open()*

Berfungsi untuk membuka dan mengidentifikasi file gambar yang diberikan. Fungsi ini mengidentifikasi sebuah file, tetapi file tetap terbuka dan data gambar tidak terbaca sampai data file gambar tersebut diproses. Memiliki parameter **fp** merupakan nama file yang akan dibuka, **mode** memiliki argumen "r", dan **formats** sebuah daftar format untuk mencoba memuat file. Parameter ini dapat digunakan untuk membatasi format yang akan diperiksa. Dalam penggunaan fungsi dapat dilihat pada gambar 2.7.

```
>>> from PIL import Image
>>> im = Image.open("hopper.ppm")
```

Gambar 2.7: Pemanggilan fungsi *open()*

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

2. Penggunaan fungsi *Image.new()*

Berfungsi untuk membuat gambar baru dengan mode 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), "RGBA" (4x8-bit piksel, warna asli dengan *transparacy 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.

```
from PIL import Image

im = Image.new("RGB", (256,256))
```

Gambar 2.8: Pemanggilan fungsi *Image.new()*

Jika pemanggilan fungsi pada gambar 2.8 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.

```
from PIL import Image

im = Image.new("RGB", (256,256))
im1 = Image.open()

im.paste(im1, (256,256))
```

Gambar 2.9: Pemanggilan fungsi *Image.paste()*

Jika fungsi pemanggilan pada gambar 2.9 berhasil, akan mengembalikan sebuah objek *Image* yang memuat gambar im1 yang ditempelkan pada gambar baru im.

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 enkoding 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 (26 = 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 enkoding Base64 menjadi m6Lp. Cara yang sama namun terbalik prosesnya digunakan untuk mendeskripsi data dari Base64 kembali ke tipe data *byte*.

2.5 Framework

Framework adalah kumpulan kode yang menggunakan *library* dan *tools* secara terstruktur sehingga dapat memudahkan developer untuk membangun dan mengembangkan sebuah perangkat lunak. Pada perangkat lunak yang akan dibangun akan menggunakan beberapa *framework* yang akan membantu proses pengerjaan. Berikut adalah *framework* yang digunakan:

2.5.1 Framework Laravel

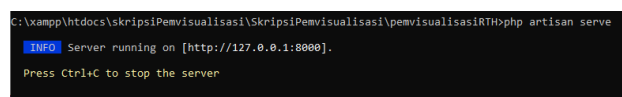
Laravel adalah *framework* aplikasi web dengan sintaks yang ekspresif dan elegan. Laravel adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model-view – controller*. Laravel berada di bawah lisensi MIT *License* dengan menggunakan Github sebagai tempat berbagi code menjalankannya. Kerangka kerja web menyediakan struktur dan titik awal untuk membuat aplikasi Anda, memungkinkan Anda untuk fokus membuat sesuatu yang luar biasa sementara kami memikirkan detailnya. 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 penggunaannya 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 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 2.10 merupakan contoh salah satu penggunaan artisan dalam laravel.



Gambar 2.10: PHP Artisan Laravel

2. 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*.

3. 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*.

4. *Blade View*

Blade adalah *template engine* bawaan dari laravel. Blade memiliki kode yang lebih mudah untuk menghasilkan laravel. Cara membuat file blade dilakukan secara manual dengan membuat nama_file.php.blade di dalam folder views

DAFTAR REFERENSI

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Kode A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35 }
36 }
```


LAMPIRAN B

HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4