

# SKRIPSI

## VISUALISASI DATA HISTORI KIRI PADA GOOGLE MAPS



Jonathan Laksamana Purnomo

NPM:

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2020



# UNDERGRADUATE THESIS

## VISUALIZATION OF KIRI HISTORICAL DATA ON GOOGLE MAPS



Jonathan Laksamana Purnomo

NPM:

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2020



# LEMBAR PENGESAHAN

## VISUALISASI DATA HISTORI KIRI PADA GOOGLE MAPS

Jonathan Laksamana Purnomo

NPM:

Bandung, 3 Desember 2020

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Pascal Alfadian, 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:

### **VISUALISASI DATA HISTORI KIRI PADA GOOGLE MAPS**

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 3 Desember 2020

Meterai Rp. 6000
---------------------

**Jonathan Laksamana Purnomo**  
NPM:





## ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

**Kata-kata kunci:** KIRI, *Navigation System*, *Data History*, *Marker Clustering*, *Heat Map*, Google Javascript API



## ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

**Keywords:** KIRI, *Navigation System, Data History, Marker Clustering, Heat Map, Google Javascript API*



*Dipersembahkan untuk Tuhan YME, keluarga, para dosen, teman-teman yang telah memberi dukungan dalam pembuatan skripsi ini, serta diri sendiri.*



## KATA PENGANTAR

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

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bandung, Desember 2020

Penulis





# DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	1
1.4 Batasan Masalah . . . . .	1
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>3</b>
2.1 KIRI Website . . . . .	3
2.2 JSON . . . . .	4
2.3 CSV . . . . .	4
2.4 Google Maps Javascript API . . . . .	4
2.4.1 <i>Position</i> . . . . .	5
2.4.2 <i>Google Maps Object</i> . . . . .	5
2.4.3 <i>Heat Map</i> . . . . .	5
2.4.4 <i>Marker</i> . . . . .	6
2.5 <i>Marker Clustering</i> . . . . .	7
DAFTAR REFERENSI	9
A KODE PROGRAM	11
B HASIL EKSPERIMEN	13



## DAFTAR GAMBAR

2.1	Tampilan utama website KIRI . . . . .	3
2.2	Contoh Heat Map . . . . .	4
2.3	Contoh Heat Map . . . . .	5
2.4	Contoh Marker . . . . .	6
2.5	Add Marker . . . . .	7
2.6	Contoh Marker Clustering . . . . .	7
B.1	Hasil 1 . . . . .	13
B.2	Hasil 2 . . . . .	13
B.3	Hasil 3 . . . . .	13
B.4	Hasil 4 . . . . .	13



## DAFTAR TABEL



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kemajuan teknologi memudahkan manusia untuk mencari berbagai macam informasi. Salah satu informasi yang dapat diperoleh adalah informasi tentang navigasi transportasi publik. KIRI adalah perangkat lunak yang berguna sebagai navigasi antar kota menggunakan transportasi publik dengan menggunakan perangkat peta digital dan informasi posisi dengan menggunakan satelit GPS.<sup>[1]</sup>

Pada perangkat lunak KIRI seluruh aktivitas yang dilakukan oleh user sudah tercatat. Data yang tercatat disebut juga dengan data histori. Tetapi data histori tersebut belum diolah secara maksimal. Data visualisasi adalah metode yang akan digunakan untuk mengolah data histori sehingga dapat ditemukan pola tertentu.

Metode yang akan digunakan dalam memvisualisasikan data adalah *Heat Map* dan *Marker Clustering*. *Heat Map* adalah teknik visualisasi data yang menunjukkan besarnya suatu fenomena sebagai warna dalam dua dimensi. Sedangkan *Marker Clustering* adalah teknik visualisasi data yang mengelompokkan *marker* atau *pointer* yang jarak *latitude* dan *longitude* nya saling berdekatan antara suatu *marker* dengan *marker* yang lainnya.

### 1.2 Rumusan Masalah

Rumusan masalah dari topik ini adalah sebagai berikut:

- Bagaimana memvisualisasikan data histori KIRI?
- Bagaimana menemukan pola dari data histori KIRI?

### 1.3 Tujuan

Tujuan dari topik ini adalah sebagai berikut:

- Mempelajari *Google Maps Javascript API*.
- Melakukan observasi data.

### 1.4 Batasan Masalah

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris.

Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

## 1.5 Metodologi

Metodologi yang digunakan dalam penelitian ini adalah:

1. Mempelajari *Google Maps Javascript API* khususnya *Heat Map* dan *Marker Clustering*.
2. Analisis masalah perangkat lunak yang akan dibangun.
3. Merancang perangkat lunak yang akan dibangun.
4. Membangun perangkat lunak yang mengimplementasikan *Heat Map* atau *Marker Clustering* dengan memanfaatkan *Google Maps Javascript API*.
5. Menentukan pola dari hasil visualisasi data.
6. Analisis hasil pengujian dan mengambil kesimpulan.

## 1.6 Sistematika Pembahasan

Laporan penelitian tersusun ke dalam enam bab secara sistematis sebagai berikut.

- Bab 1 Pendahuluan  
Berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.
- Bab 2 Dasar Teori  
Berisi metode penentuan pola, *library Google Maps* dan bahasa pemrograman *Javascript*
- Bab 3 Analisis  
Berisi analisis masalah terkait implementasi *Goole Map*, studi kasus teknik penentuan pola yang diimplementasikan, dan gambaran umum perangkat lunak yang meliputi diagram aktivitas dan diagram kelas.
- Bab 4 Perancangan Perangkat Lunak  
Berisi perancangan perangkat lunak yang akan dibangun, meliputi perancangan antarmuka, diagram kelas lengkap dan masukan perangkat lunak.
- Bab 5 Implementasi dan Pengujian  
Berisi implementasi antarmuka perangkat lunak, pengujian fungsional, pengujian eksperimental serta kesimpulan dari pengujian.
- Bab 6 Kesimpulan dan Saran  
Berisi kesimpulan dari awal hingga akhir penelitian dan saran untuk penelitian berikutnya.



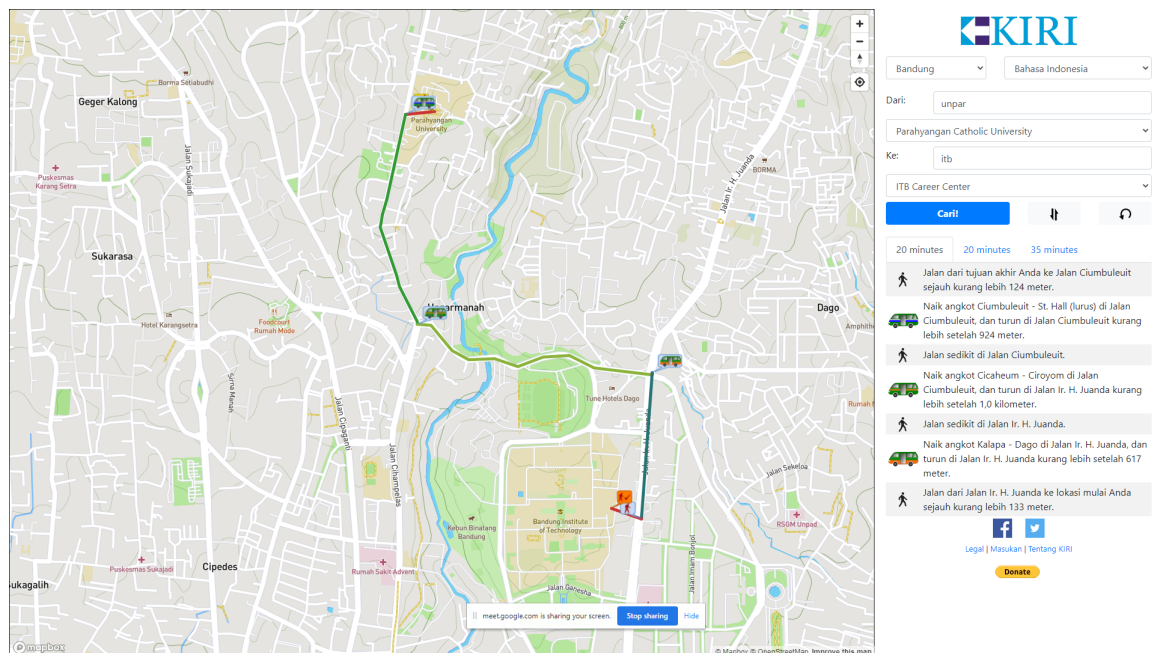
## BAB 2

### LANDASAN TEORI

#### 2.1 KIRI Website

KIRI adalah aplikasi navigasi angkutan umum berbasis web yang melayani Bandung dan kota-kota lain di Indonesia.[1]. Pada awal pembuatannya KIRI dibuat untuk tujuan komersial. Namun karena dinilai kurang sukses project KIRI sekarang menjadi open source project yang dapat di akses. Aplikasi KIRI memiliki beberapa fitur sebagai berikut:

- Pemilihan rute tercepat menggunakan angkutan kota.
- Menampilkan tempat pergantian angkutan kota.
- Memiliki fitur multi bahasa.
- Memiliki fitur pemilihan lokasi.
- Dapat menampilkan beberapa pilihan rute.
- Dapat menampilkan instruksi lengkap mencapai tujuan.



Gambar 2.1: Tampilan utama website KIRI

## 2.2 JSON

JSON (*JavaScript Object Notation*) adalah notasi berbasis teks, format pertukaran data bahasa-independen. berasal dari Standar Bahasa Pemrograman ECMAScript [2]. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh bahasa pemrograman C termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON memiliki enam tipe data, yaitu *string*, angka, *null*, *array* (ditandai dengan tanda kurung siku ([])), *object* (ditandai dengan tanda kurung kurawal ({})), dan *boolean* (*true* dan *false*). Struktur utama JSON terdiri dari pasangan nama atau nilai yang dipisahkan dengan tanda titik dua (:). Contoh struktur JSON mengenai tipe dan jenis atribut dapat dilihat pada Listing 2.1.

Listing 2.1: Contoh Struktur JSON

```
{
  "timestamp": "2014-1-2:0:11",
  "start": "-6.8972513,107.6385574",
  "finish": "-6.91358,107.62718"
}
```

## 2.3 CSV

CSV (*Comma Separated Values*) adalah suatu format data dalam basis data di mana setiap nilai atribut dipisahkan dengan tanda koma (,) dan setiap baris data ditandai dengan baris baru.[3] CSV digunakan untuk bertukar data dan mengonversi data dari sebuah program *spreadsheet* ke program *spreadsheet* lainnya. Contoh CSV dapat dilihat pada Listing 2.2.

Listing 2.2: Contoh CSV

```
logId ,APIKey ,Timestamp (UTC) ,Action ,AdditionalData
113909 ,E5D9904F0A8B4F99 ,2/1/2014 0:07 ,PAGELOAD ,/5.10.83.30/
113910 ,E5D9904F0A8B4F99 ,2/1/2014 0:07 ,PAGELOAD ,/5.10.83.49/
```

## 2.4 Google Maps Javascript API

Google Maps adalah layanan pemetaan web yang dikembangkan oleh Google. Menawarkan citra satelit, foto udara, dan peta jalan yang interaktif, kondisi lalu lintas secara *real time* [4].



Gambar 2.2: Contoh Heat Map

Google Maps dalam service nya telah menyediakan *API (pplication programming interface)* yang dapat di gunakan untuk public. *aplication programming interface* adalah *computer interface* yang mengatur komunikasi antar perangkat lunak [5]. Google Maps telah menyediakan beberapa teknik pemetaan data yaitu :

- *Heat Map*
- *Marker Clustering*

terdapat beberapa parmeter untuk menginisialisasi *Google Maps Javascript API*

- *Google Maps Object*
- *Position*
- *Zoom Level*

### 2.4.1 *Position*

*Google Maps* menggunakan *Latitude* dan *Longitude* sebagai atribut untuk menyatakan sebuah posisi. *Latitude* adalah garis yang horisontal / mendatar. Titik 0 adalah sudut ekuator, tanda + menunjukan arah ke atas menuju kutub utara, sedangkan tanda minus di koordinat *Latitude* menuju ke kutub selatan. *Longitude* adalah garis lintang . Angka dari sudut bundar bumi horisontal. Titik diawali dari 0 ke 180 derajat, dan 0 ke-180 ke arah sebaliknya.

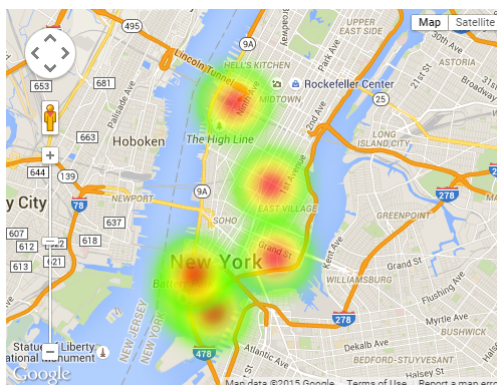
### 2.4.2 *Google Maps Object*

*Maps Object* adalah Kelas *JavaScript* yang mewakili peta adalah kelas Peta. Objek kelas ini mendefinisikan satu peta di halaman. (Anda dapat membuat lebih dari satu *instance* kelas ini - setiap objek akan menentukan peta terpisah pada halaman.) Kami membuat *instance* baru dari kelas ini menggunakan *operator* baru *JavaScript*.

```
map = new google.maps.Map(document.getElementById("map"), { ... });
```

### 2.4.3 *Heat Map*

*Heat Map* adalah sebuah teknik visualisasi data dimana data digambarkan sebagai warna. Saat Lapisan *heat map* diaktifkan, hamparan berwarna akan muncul di atas map. Secara *default*, area dengan intensitas lebih tinggi akan diwarnai merah, dan area dengan intensitas lebih rendah akan tampak hijau.



Gambar 2.3: Contoh Heat Map

Beberapa Jenis *Heat Map*:

- *Click-tracking map*
- *Scroll map*
- *Hover/Mouse tracking map*
- *Eye tracking map*

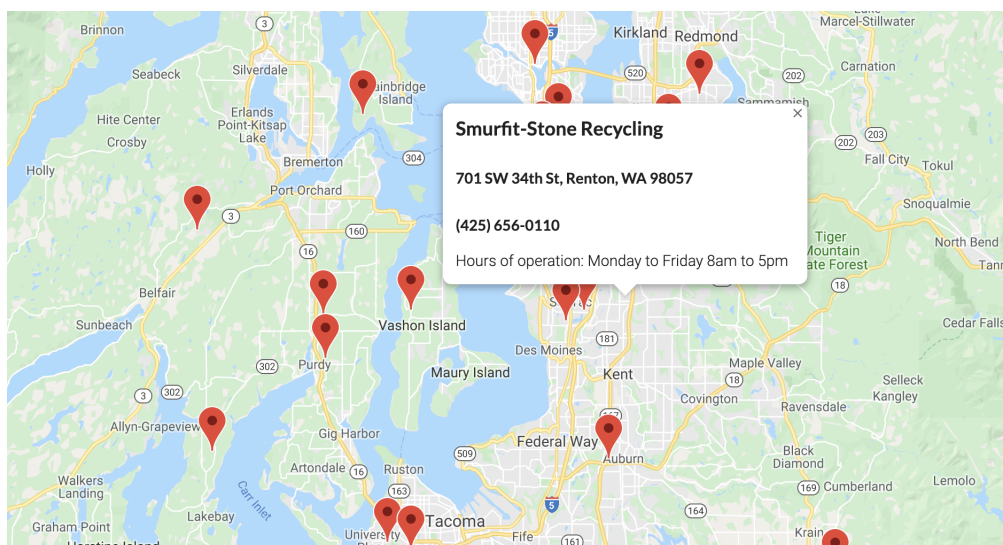
Untuk menampilkan *heat map* menggunakan *Google Javascript API* dapat menggunakan syntax.

```
var heatmap = new google.maps.visualization.HeatmapLayer({
  data: heatmapData
});
heatmap.setMap(map);
```

#### 2.4.4 *Marker*

*Marker* adalah suatu tanda yang di pasang di dalam sebuah maps atau peta sebagai petanda lokasi atau tempat. Marker dapat menampilkan gambar khusus, dalam hal ini biasanya disebut sebagai *icon*.<sup>[6]</sup> Untuk Menambahkan Marker. Untuk menambahkan *Marker* pada *Google Map Javascript Api* membutuhkan beberapa parameter

- *Position* menggunakan *latlng* untuk mengidentifikasi letak suatu tempat
- *Map* menggunakan object map dari google javascript api



Gambar 2.4: Contoh Marker

Untuk menambahkan *Marker* dapat menggunakan *syntax* seperti:

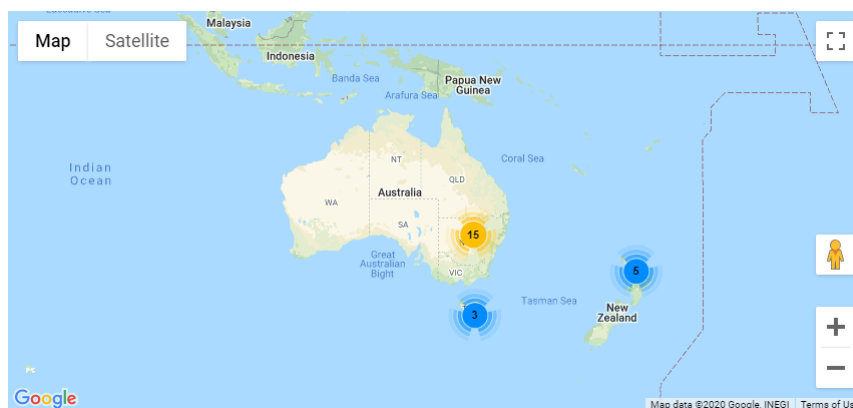
```
var heatmap = new google.maps.visualization.HeatmapLayer({
  data: heatmapData
});
heatmap.setMap(map);
```



Gambar 2.5: Add Marker

## 2.5 Marker Clustering

*Marker Clustering* adalah teknik visualisasi data dimana data akan di representasikan sebagai tanda / *Mark* pada suatu space 2 dimensi, semakin banyak data yang terdapat pada suatu tempat maka akan semakin banyak quantitas penanda / *Mark* yang di berikan. [7]



Gambar 2.6: Contoh Marker Clustering

Untuk menggunakan *Marker Clustering* pada *Google Maps Javascript API* dapat menggunakan *Object Marker Clusterer* yang telah di sediakan oleh *Google Maps*

```
new MarkerClusterer(map, markers, {
  imagePath:
    "https://developers.google.com/maps/documentation/javascript/examples/markerclusterer/m",
});
```

*Marker Clustering* akan menggunakan *grid-based clustering technique* yang membagi peta menjadi kotak dengan ukuran tertentu (ukuran berubah di setiap tingkat zoom), dan mengelompokkan penanda ke dalam setiap kisi persegi. Ini membuat *cluster* di *marker* tertentu, dan menambahkan marker yang berada dalam batas-batasnya ke *cluster.Marker Clustering* merupakan sebuah teknik yang menggunakan *Marker* sehingga untuk dapat memunculkan *Marker Clustering* diperlukan object *Marker* berikut ini contoh pengimlementasian *Marker Clustering*

```
function initMap() {
  const map = new google.maps.Map(document.getElementById("map"), {
    zoom: 3,
```

```
    center: { lat: -28.024, lng: 140.887 },
  });
const labels = "ABCDEFGHJKLMNOPQRSTUVWXYZ";
const markers = locations.map((location, i) => {
  return new google.maps.Marker({
    position: location,
    label: labels[i % labels.length],
  });
});
new MarkerClusterer(map, markers, {
  imagePath: "",
});
}
const locations = [
  { lat: -31.56391, lng: 147.154312 },
  { lat: -33.718234, lng: 150.363181 },
  { lat: -33.727111, lng: 150.371124 },
];
```

## DAFTAR REFERENSI

- [1] Nugroho, P. dan Natali, V. (2017) Open sourcing proprietary application case study: Kiri website. *International Journal of New Media Technology*, **4**, 82–86.
- [2] Bray, T. dan Google (2014) The javascript object notation (json) data interchange format. Technical Report 8259.
- [3] Shafranovich, Y. (2005) Common format and mime type for comma-separated values (csv) files. Technical Report 7111.
- [4] Mehta, H., Kanani, P., dan Lande, P. (2019) Google maps. *International Journal of Computer Applications*, **178**, 41–46.
- [5] Libby, A. (2020) Working with the API.
- [6] Developers, G. Marker. <https://developers.google.com/maps/documentation/javascript/markers>.
- [7] Developers, G. Marker clustering. <https://developers.google.com/maps/documentation/javascript/marker-clustering>.





# LAMPIRAN A

## KODE PROGRAM

Listing A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4
5 #include<stdio.h>
6
7 void myFunction( int input, float* output ) {
8     switch ( array[i] ) {
9         case 1: // This is silly code
10             if ( a >= 0 || b <= 3 && c != x )
11                 *output += 0.005 + 20050;
12             char = 'g';
13             b = 2^n + ~right_size - leftSize * MAX_SIZE;
14             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
15             strcpy(a,"hello_$@?");
16         }
17         count = ~mask | 0x00FF00AA;
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

```

Listing 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