

**SKRIPSI**

**PENGUKURAN APLIKASI USANG DI HTTPARCHIVE**



**Vinson Tandra**

**NPM: 2016730042**

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

**«tahun»**



# DAFTAR ISI

DAFTAR ISI	iii
DAFTAR GAMBAR	v
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>3</b>
2.1 BigQuery[?] . . . . .	3
2.1.1 <i>Cloud Storage System</i> . . . . .	3
2.2 HTTP Archive . . . . .	3
2.3 Web Almanac . . . . .	5
2.4 OSEMN Framework . . . . .	7
2.4.1 Obtain Data . . . . .	7
2.4.2 Scrub Data . . . . .	7
2.4.3 Explore Data . . . . .	7
2.4.4 Model Data . . . . .	7
2.4.5 Interpreting Data . . . . .	7
2.5 Pengukuran Aplikasi Usang Pada Beberapa Website Terkenal Di Indonesia[?] . . .	8
2.5.1 <i>Research Method</i> . . . . .	8
2.5.2 <i>Hasil Keseluruhan</i> . . . . .	9
DAFTAR REFERENSI	11
A KODE PROGRAM	13
B HASIL EKSPERIMEN	15



## DAFTAR GAMBAR

2.1	<i>Algorithm to compare current version versus supported versions</i>	9
2.2	<i>Overall application count for measurement result</i>	9
B.1	Hasil 1	15
B.2	Hasil 2	15
B.3	Hasil 3	15
B.4	Hasil 4	15



# BAB 1

## PENDAHULUAN

Pada bab ini dijelaskan latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.

### 1.1 Latar Belakang

Di masa teknologi saat ini, banyak perusahaan yang menggunakan website sebagai tempat untuk mencari informasi. Terdapat banyak website yang menggunakan aplikasi yang sudah usang. HTTP Archive memiliki dataset penggunaan aplikasi suatu website yang dapat dilihat atau dianalisis menggunakan Google Cloud Platform. HTTP Archive memiliki dataset penggunaan aplikasi suatu website dari desktop dan mobile pada bulan Januari tahun 2016 sampai sekarang. Berdasarkan sumber pada website almanac<sup>1</sup>, dapat diambil kesimpulan bahwa website dibuka menggunakan browser di desktop dan mobile. Dataset yang digunakan berada pada label *technologies* merupakan dataset desktop dan mobile pada bulan Agustus tahun 2020. Dataset pada desktop memiliki 61.203.638 baris dan pada mobile memiliki 67.452.994 baris.

Berdasarkan jurnal ilmiah [?], dari 1.500 situs teratas menurut peringkat Alexa untuk pengunjung situs di Indonesia dan mengidentifikasi jenisnya aplikasi yang mereka gunakan beserta nomor versinya, lebih dari setengah atau 63% aplikasi yang digunakan berhasil dibandingkan dengan skrip yang telah dibuat dan hasilnya aplikasi tidak lagi didukung oleh pengelolanya.

Beberapa aplikasi sudah menyediakan fitur untuk meng-*update* ke versi yang paling baru tanpa harus menginstal ulang. Dalam kebanyakan kasus, versi aplikasi yang semakin baru sudah memperbaiki banyak kerentanan yang sudah diketahui. Beberapa aplikasi usang tidak memiliki pemberitahuan untuk meng-*update* sehingga pengguna tidak mengetahui jika terdapat *update*. Aplikasi yang baik biasanya memberikan update otomatis dan memberikan pesan yang efektif jika terjadi *update*.

HTTP Archive<sup>2</sup> adalah sebuah proyek yang bersifat *open source* untuk melihat bagaimana website dibuat. Di dalam HTTP Archive terdapat data-data historis yang disediakan untuk menunjukkan bagaimana website terus berkembang dan proyek ini sering digunakan untuk penelitian. HTTP Archive memiliki jutaan web setiap bulan dan menyediakan terabyte data metadata untuk dianalisis menggunakan BigQuery. BigQuery<sup>3</sup> adalah gudang data perusahaan yang dikelola sepenuhnya yang membantu mengelola dan menganalisis data dengan fitur bawaan. BigQuery

---

<sup>1</sup><https://almanac.httparchive.org/en/2020/mobile-web>

<sup>2</sup>[https://github.com/HTTPArchive/httparchive.org/blob/main/docs/gettingstarted\\_bigquery.md](https://github.com/HTTPArchive/httparchive.org/blob/main/docs/gettingstarted_bigquery.md)

<sup>3</sup><https://cloud.google.com/bigquery/docs>

1 membantu dalam membuat *query* dalam skala *terabyte* dalam hitungan detik dan *petabyte* dalam  
2 hitungan menit.

3 Pada skripsi ini, akan dibuat sebuah penelitian untuk mengetahui seberapa besar penggunaan  
4 aplikasi usang pada seluruh website yang ada di dunia. Data dapat diambil dari HTTP Archive  
5 dengan melakukan query pada google BigQuery . Pada penelitian ini akan dilakukan perhitungan  
6 pada jumlah aplikasi yang sudah diberi versi dan belum diberi versi. Jika belum diberikan versi  
7 maka aplikasi dinyatakan usang. Kemudian hasil tersebut akan ditampilkan dalam bentuk bar  
8 chart.

## 9 1.2 Rumusan Masalah

10 Berikut ini adalah rumusan masalah dari penelitian ini:

- 11 1. Berapakah jumlah data di HTTP Archive?
- 12 2. Bagaimana cara membaca data dari HTTP Archive?
- 13 3. Bagaimana mengimplementasi proyek yang sudah ada dengan menggunakan data yang lebih  
14 besar?

## 15 1.3 Tujuan

16 Berikut ini adalah tujuan dari penelitian ini:

- 17 1. Mengetahui jumlah data di HTTP Archive.
- 18 2. Membaca data dari HTTP Archive.
- 19 3. Mengimplementasi proyek yang sudah ada dengan menggunakan data yang lebih besar.

## 20 1.4 Batasan Masalah

21 Penelitian ini dibuat dengan batasan - batasan berikut:

## 22 1.5 Metodologi

23 Bagian-bagian pekerjaan skripsi ini adalah sebagai berikut:

- 24 1. Mempelajari teori HTTP Archive.
- 25 2. Mempelajari teori BigQuery.
- 26 3. Mempelajari bagaimana suatu website dikatakan usang.
- 27 4. Menganalisis beberapa website yang dikatakan usang.
- 28 5. Menulis dokumen skripsi.

## 29 1.6 Sistematika Pembahasan



## BAB 2

### LANDASAN TEORI

#### 2.1 BigQuery[?]

BigQuery<sup>1</sup> adalah gudang data perusahaan yang terkelola sepenuhnya yang membantu mengelola dan menganalisis data dengan fitur bawaan seperti machine learning, analisis geospasial, dan kecerdasan bisnis. BigQuery memaksimalkan fleksibilitas dengan memisahkan mesin komputasi yang menganalisa data. BigQuery dapat digunakan sebagai tempat penyimpanan dan data tersebut dapat dianalisis. Data dapat dibaca dengan menggunakan *query* gabungan. Google meluncurkan bigquery secara publik pada tahun 2012 yang memungkinkan pengguna diluar Google untuk memanfaatkan *power* dan *performance* dari Dremel. Saat ini BigQuery sudah berkembang menjadi penyedia penyimpanan terstruktur berbasis cloud yang dikelola dan dihosting.

##### 2.1.1 Cloud Storage System

Selain sebagai tempat untuk menjalankan *query* dari data, saat ini BigQuery juga merupakan tempat penyimpanan data terstruktur di cloud. Data akan direplikasi ke beberapa lokasi yang berbeda secara geografis untuk meningkatkan ketersediaan dan ketahanan. Jika pusat data di Google pada suatu lokasi ditutup, data tetap dapat diakses tanpa terjadi gangguan. Data juga akan direplikasi dalam sebuah kluster agar tidak terjadi kehilangan data jika terjadi kegagalan perangkat keras.

#### 2.2 HTTP Archive

HTTP Archive adalah sebuah *open-source project* yang melihat bagaimana website dibuat. HTTP Archive menyediakan data-data historis untuk melihat bagaimana website berkembang<sup>2</sup>. HTTP Archive pertama kali dimulai pada tahun 2010 oleh Steve Souders dan di-*maintain* oleh Pat Meenan, Rick Viscomi, Paul Calvano, and Barry Pollard. Pada HTTP Archive terdapat batasan, yaitu HTTP Archive memeriksa setiap URL tetapi tidak sampai melihat halaman lainnya. Misalnya Sebagian besar *website* terdiri dari banyak halaman web terpisah. Karena batasan ini sehingga ada kemungkinan bahwa dokumen HTML sebenarnya yang dianalisis tidak mewakili situs web. Di dalam HTTP Archive terdapat dataset yang dapat diambil menggunakan teknologi bigquery, dataset tersebut adalah sebagai berikut:

---

<sup>1</sup><https://cloud.google.com/bigquery/docs/introduction>

<sup>2</sup><https://httparchive.org/faq>

1. almanac

2. blink features

3. core web vitals

4. latest

5. lighthouse

6. pages

Dataset pada technologies berisi tabel-tabel dari bulan Januari tahun 2016 sampai dengan sekarang yang terdiri dari website pada desktop dan mobile. Dataset bulan Agustus tahun 2020 baris pada desktop memiliki 5.593.642 baris dan pada mobile memiliki 6.347.640 baris yang dapat dianalisis. Masing-masing terdiri dari URL dan payload. *URL (Uniform Resource Locator)* merupakan nama-nama domain dan *payload*

7. requests

Dataset pada technologies berisi tabel-tabel dari bulan Januari tahun 2016 sampai dengan sekarang yang terdiri dari website pada desktop dan mobile. Dataset bulan Agustus tahun 2020 baris pada desktop memiliki 535.841.778 baris dan pada mobile memiliki 579752745 baris yang dapat dianalisis. Masing-masing terdiri dari URL dan payload. *URL (Uniform Resource Locator)* merupakan nama-nama domain dan *payload*.

8. response bodies

9. sample data

10. sample data 2020

11. scratchspace

12. summary pages

13. summary requests

14. technologies

Dataset pada technologies berisi tabel-tabel dari bulan Januari tahun 2016 sampai dengan sekarang yang terdiri dari website pada desktop dan mobile. Dataset bulan Agustus tahun 2020 baris pada desktop memiliki 61.203.638 baris dan pada mobile memiliki 67.452.994 baris yang dapat dianalisis. Masing-masing terdiri dari 4 kolom yaitu *URL*, *category*, *app*, *info*. Pada kolom *URL (Uniform Resource Locator)* merupakan nama-nama domain, *category* merupakan jenis aplikasi yang digunakan pada website tersebut, *app* merupakan aplikasi yang digunakan website tersebut, *info* merupakan informasi tambahan dari aplikasi.

1 15. urls

3 16. wappalyzer

## 5 2.3 Web Almanac

6 Web Almanac <sup>3</sup> adalah sebuah projek yang dikelola oleh HTTP Archive. Misi web almanac adalah  
7 menggabungkan statistik mentah dan tren HTTP Archive dengan keahlian komunitas web. Semua  
8 metrik yang disediakan oleh web almanac dapat direproduksi secara publik menggunakan dataset  
9 di BigQuery. Kueri dapat ditelusuri dengan menggunakan semua bab di repositori GitHub web  
10 almanac yang dapat dilihat pada <sup>4</sup>:

### 11 1. Accessibility

12 Aksesibilitas web adalah tentang pencapaian fitur dan informasi serta memberikan akses  
13 lengkap ke semua aspek antarmuka bagi orang yang tidak memiliki akses. Sebuah produk  
14 digital atau situs web tidak lengkap jika tidak dapat digunakan oleh semua orang.

### 15 2. Caching

16 Caching adalah teknik yang memungkinkan penggunaan kembali konten yang diunduh se-  
17 belumlahnya. Caching melibatkan sesuatu seperti server atau web browser untuk menyimpan  
18 konten dan menandainya agar dapat digunakan kembali.

### 19 3. Capabilities

### 21 4. CMS

22 Istilah CMS mengacu pada sistem yang memungkinkan individu dan organisasi untuk membuat,  
23 mengelola, dan mempublikasikan konten. CMS pada konten web adalah sistem yang bertujuan  
24 untuk membuat, mengelola, dan menerbitkan konten untuk dikonsumsi dan dialami melalui  
25 internet.

### 26 5. Compression

27 Menggunakan HTTP Compression membuat pemuatan situs lebih cepat dan menjamin penga-  
28 laman penggunaan yang lebih baik. Penggunaan compression yang efektif dapat mengurangi  
29 berat halaman dan meningkatkan kinerja web.

### 30 6. CSS

31 CSS adalah bahasa yang digunakan untuk membuat tampilan dan format pada web dan media  
32 lainnya.

### 33 7. Ecommerce

34 Ecommerce platform adalah perangkat lunak atau layanan yang memungkinkan untuk mem-  
35 buat dan mengoperasikan sebuah toko online.

### 36 8. Fonts

37 Teks adalah bagian penting dalam sebuah situs web dan tipografi adalah seni menyajikan teks  
38 tersebut dengan cara yang menarik dan efektif secara visual. Dalam pembuatan tipografi  
39 yang baik dibutuhkan pemilihan font yang sesuai. Dalam hal ini akan ditunjukkan bagaimana

---

<sup>3</sup><https://almanac.httparchive.org/en/2020/methodology>

<sup>4</sup><https://github.com/HTTPArchive/almanac.httparchive.org/tree/main/sql/2020>

font web digunakan dan bagaimana font tersebut dioptimalkan.

## 9. HTTP

HTTP adalah protokol lapisan aplikasi yang dirancang untuk mentransfer informasi antara perangkat jaringan dan berjalan di atas lapisan lain dari tumpukan protokol jaringan. Dalam web almanac akan mengulas bagaimana status penerapan HTTP/2 atau HTTP versi dua pada saat ini.

## 10. Jamstack

Jamstack adalah konsep arsitektur yang relatif baru yang dirancang untuk membuat web lebih cepat, lebih aman, dan lebih mudah untuk diskalakan. Dalam web almanac akan memperkirakan dan menganalisis pertumbuhan situs Jamstack, kinerja kerangka kerja Jamstack populer, serta analisis pengalaman pengguna nyata menggunakan metrik Core Web Vitals.

## 11. Javascript

JavaScript adalah bahasa pemrograman yang digunakan untuk menentukan perilaku.

## 12. Markup

HTML adalah dasar dari sebuah website yang akan ditampilkan ke-*user*. Dalam web almanac mengacu pada kumpulan halaman *mobile*.

## 13. Media

Pada web almanac, media digunakan untuk menganalisa bagaimana menggunakan gambar dan video di web.

## 14. mobile-web

## 15. Page-weight

## 16. Performance

Dalam web almanac, akan melihat data kinerja di dunia nyata yang disediakan oleh Laporan Pengalaman Pengguna Chrome (CrUX) melalui lensa perkembangan baru tersebut serta menganalisis beberapa metrik relevan lainnya.

## 17. Privacy

Web almanac memberikan gambaran umum tentang keadaan privasi saat ini di web. Hal ini bertujuan untuk meningkatkan akuntabilitas pemroses data dan transparansi mereka terhadap pengguna. Dalam hal ini, kami membahas prevalensi pelacakan online dengan berbagai teknik dan tingkat adopsi spanduk persetujuan cookie dan kebijakan privasi oleh situs web.

## 18. PWA

Dalam web almanac, kita akan melihat setiap komponen yang membuat PWA seperti apa adanya, dari perspektif berbasis data.

## 19. Resource-hints

## 20. Security

Dalam web almanac, akan dilakukan menganalisis penerapan berbagai fitur keamanan secara mendalam dan dalam skala besar, kami mengumpulkan wawasan tentang berbagai cara pemilik situs web menerapkan mekanisme keamanan ini, didorong oleh insentif untuk melindungi penggunaannya.

## 21. SEO

Dalam web almanac, untuk mengidentifikasi dan menilai elemen dan konfigurasi utama yang berperan dalam pengoptimalan pencarian organik situs web.

#### 22. Third-parties

Web almanac meninjau prevalensi konten pihak ketiga dan bagaimana hal ini telah berubah sejak 2019.

## 2.4 OSEMN Framework

OSEMN merupakan data science framework yang memberikan langkah-langkah pengerjaan proyek.<sup>5</sup>

### 2.4.1 Obtain Data

Obtain data berarti mengumpulkan data dari berbagai sumber. Langkah ini adalah langkah pertama. Mengumpulkan data sangat penting karena dalam melakukan sebuah proyek harus memiliki data. Data dapat didapat dengan meng-query dari database.

### 2.4.2 Scrub Data

Pada proses scrubbing data, data yang dikumpulkan tersebut akan dibersihkan atau difilter. Jika menggunakan data yang tidak difilter maka akan mempengaruhi keakuratan hasil akhir. Scrubbing data bisa saja merupakan ekstraksi data dan bertukar nilai.

### 2.4.3 Explore Data

Pada explore data, akan dilakukan pengecekan terhadap tipe dari data. Kemudian data-data tersebut akan dikumpulkan dan dibandingkan sehingga mendapat kesimpulan dari data yang ingin dicari.

### 2.4.4 Model Data

Model data adalah pembuatan hasil akhir dari data yang diselidiki. Tujuan dari model data adalah mengelompokan data untuk memahami logika di balik cluster tersebut.

### 2.4.5 Interpreting Data

Interpreting data mengacu pada penyajian data, penyampaian hasil agar dapat menunjukkan kesimpulan. Hasil-hasil yang ditunjukkan dapat berupa grafik-garfik agar dapat dijelaskan secara jelas dan aplikatif.

---

<sup>5</sup><https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492>

## 2.5 Pengukuran Aplikasi Usang Pada Beberapa Website Terkenal Di Indonesia[?]

### 2.5.1 *Research Method*

#### 1. Memilih list website yang populer

Memilih website paling populer dilakukan dengan mengambil daftar dari website teratas dari Alexa dengan negara tertentu.

#### 2. Mengidentifikasi aplikasi yang dipakai website

Untuk setiap website akan dilakukan pengidentifikasian nomor versi yang dipakai. Hal ini dibantu dengan menggunakan *third party* yaitu Wappalyzer.

#### 3. Mengelompokkan berdasarkan nama aplikasi dan ambil versi yang didukung

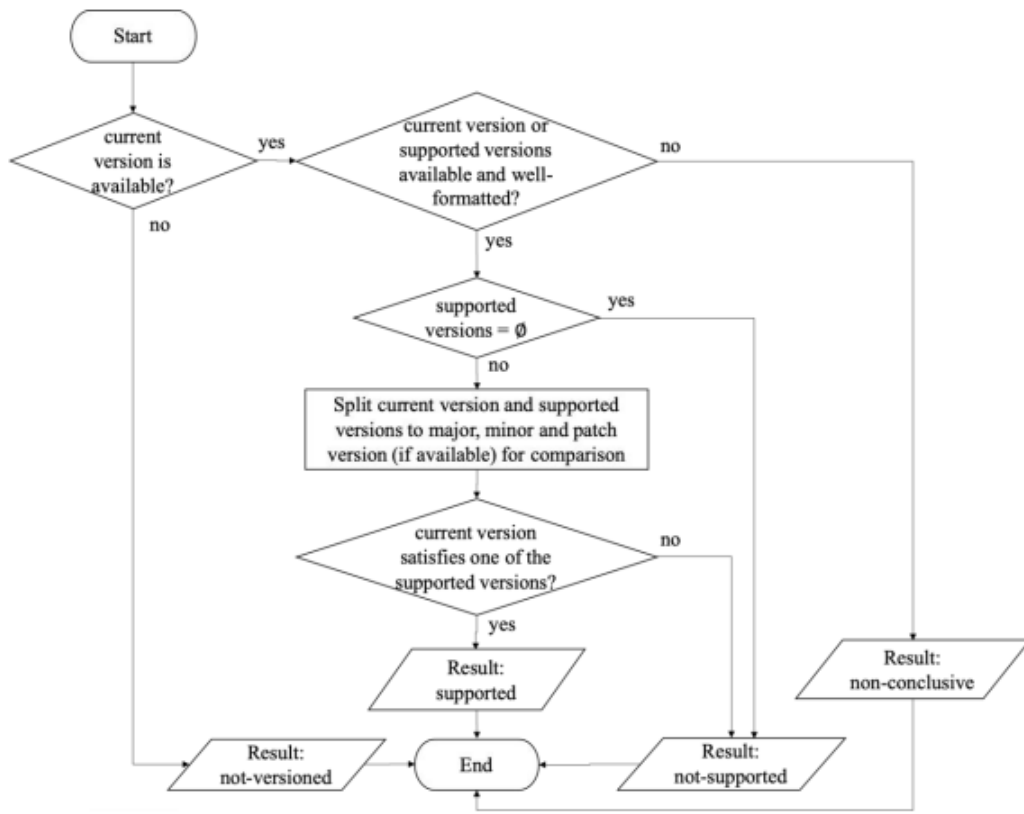
Untuk melihat nomor versi yang masih didukung akan dilakukan pencarian di website resmi dari setiap aplikasi. Terdapat beberapa website yang tidak dapat ditampilkan versinya, sehingga suatu website dapat didefinisikan didukung jika memenuhi kondisi sebagai berikut:

- Versi aplikasi yang didukung dapat dilihat secara eksplisit di dalam website.
- Dokumen untuk versi aplikasi tersebut masih tersedia.
- Aplikasi secara langsung memberikan pernyataan untuk versi yang masih didukung.

#### 4. Membandingkan versi yang dipakai aplikasi saat ini dengan versi aplikasi yang didukung

Buka kembali setiap aplikasi kemudian menggunakan Wappalyzer untuk membandingkan versi aplikasi yang dipakai dengan versi aplikasi yang masih didukung. Klasifikasikan setiap aplikasi di setiap situs web menjadi salah satu dari berikut ini:

- *Not-versioned* berarti aplikasi yang terdeteksi oleh Wappalyzer tidak memiliki informasi versi sehingga tidak dapat dibandingkan.
- Non-konklusif dapat berarti salah satu dari dua:
  - Dapat mengambil nomor versi yang digunakan dalam aplikasi, tetapi kami tidak dapat menentukan apakah versi tersebut masih didukung atau tidak oleh pengelola.
  - Versi yang didukung untuk aplikasi tertentu tidak diketahui.
- Tidak didukung berarti dapat disimpulkan bahwa aplikasi yang digunakan menggunakan nomor versi yang tidak didukung oleh pengelola.
- Didukung berarti dapat disimpulkan bahwa aplikasi yang digunakan menggunakan nomor versi masih didukung oleh pengelola.

Gambar 2.1: *Algorithm to compare current version versus supported versions*

### 2.5.2 Hasil Keseluruhan

Pada paper[?], dari 1.500 URL yang dideteksi oleh Wappalyzer, hanya 1.439 URL yang berhasil diidentifikasi. Dari 1.500 URL tersebut ditemukan total 12.762 aplikasi yang dapat dilihat pada Gambar 2.2

Result	Application count	Percentage
Not-versioned	8,980	70.37
Non-conclusive	1,409	11.04
Unsupported	1,508	11.82
Supported	865	6.78
Total	12,762	100.00

Gambar 2.2: *Overall application count for measurement result*





## 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