

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
1 PENDAHULUAN	1
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
2 LANDASAN TEORI	3
2.1 BigQuery[1, 2]	3
2.1.1 Cloud Storage System	3
2.1.2 SQL (Structured Query Language) [3]	3
2.2 HTTP Archive [4]	6
2.3 Web Almanac [5]	7
2.4 OSEM Framework	9
2.4.1 Obtain Data	9
2.4.2 Scrub Data	9
2.4.3 Explore Data	9
2.4.4 Model Data	9
2.4.5 Interpreting Data	9
2.5 Pengukuran Aplikasi Usang Pada Beberapa Website Populer Di Indonesia[6]	10
2.5.1 Research Method	10
2.5.2 Hasil Keseluruhan	11
3 PERCOBAAN AWAL	13
3.1 Eksplorasi Teknologi	13
3.2 Dataset Pada HTTP Archive	13
3.3 Langkah-Langkah Query Yang Dilakukan	18
3.3.1 Mengumpulkan List Website	18
3.3.2 Mencari Aplikasi Yang Digunakan Website	18
3.3.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai	19
3.3.4 Mencari Data Tentang Versi Aplikasi Yang Masih Didukung	19
3.3.5 Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai Sekarang Dengan Versi Aplikasi Yang Masih Didukung	19
DAFTAR REFERENSI	21
A KODE PROGRAM	23

DAFTAR GAMBAR

2.1	<i>Algorithm to compare current version versus supported versions</i>	11
B.1	Hasil 1	25
B.2	Hasil 2	25
B.3	Hasil 3	25
B.4	Hasil 4	25

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 teknologi BigQuery. HTTP Archive memiliki dataset penggunaan aplikasi suatu website dari desktop dan mobile pada bulan Januari tahun 2016 sampai sekarang. Berdasarkan sumber pada web almanac¹, dapat diambil kesimpulan bahwa website dibuka menggunakan browser di desktop dan mobile. Dataset yang digunakan berada pada label *technologies* merupakan dataset desktop dan mobile dengan nama tabel 2020_08_01. Dataset pada desktop memiliki 61.203.638 baris dan pada mobile memiliki 67.452.994 baris.

HTTP Archive² 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. Didalam HTTP Archive terdapat dataset yang berisi jutaan web setiap bulan dan dapat dianalisis menggunakan teknologi BigQuery. BigQuery [2] adalah salah satu produk dari Google yang berbasis *cloud* dan dapat digunakan untuk menganalisis data tanpa harus memikirkan database. BigQuery dapat menjalankan *query* dalam skala *terabyte* dalam hitungan detik dan *petabyte* dalam hitungan menit.

Berdasarkan [6], 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

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

²https://github.com/HTTPArchive/httparchive.org/blob/main/docs/gettingstarted_bigquery.md

1 terjadi *update*.

2 Pada skripsi ini, akan dibuat sebuah penelitian untuk mengetahui seberapa besar penggunaan
3 aplikasi usang pada seluruh website yang ada di dunia. Data dapat diambil dari HTTP Archive
4 dengan melakukan *query* pada BigQuery. Pada penelitian ini akan dilakukan perhitungan pada
5 jumlah aplikasi yang sudah diberi versi dan belum diberi versi. Versi aplikasi yang dipakai setiap
6 website juga akan dibandingkan dengan versi aplikasi yang masih didukung berdasarkan *official*
7 *website*-nya. Kemudian hasil tersebut akan ditampilkan dalam bentuk bar chart.

8 1.2 Rumusan Masalah

9 Berikut ini adalah rumusan masalah dari penelitian ini:

- 10 1. Bagaimana cara membaca data dari HTTP Archive?
- 11 2. Bagaimana mengimplementasi proyek [6] dengan menggunakan data yang lebih besar?
- 12 3. Berapa banyak website pada web almanac yang menggunakan aplikasi yang masih didukung?

13 1.3 Tujuan

14 Berikut ini adalah tujuan dari penelitian ini:

- 15 1. Membaca data dari HTTP Archive.
- 16 2. Mengimplementasi proyek [6] dengan menggunakan data yang lebih besar.
- 17 3. Mencari jumlah website pada web almanac yang menggunakan aplikasi yang masih didukung.

18 1.4 Batasan Masalah

19 Penelitian ini dibuat dengan batasan - batasan berikut:

20 1.5 Metodologi

21 Bagian-bagian pekerjaan skripsi ini adalah sebagai berikut:

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

27 1.6 Sistematika Pembahasan

BAB 2

LANDASAN TEORI

2.1 BigQuery[1, 2]

Google memiliki salah satu produk yaitu BigQuery yang berbasis *cloud* dan dapat digunakan untuk menganalisis data tanpa harus memikirkan database. BigQuery memaksimalkan fleksibilitas dengan memisahkan memisahkan mesin komputasi yang menganalisa data. BigQuery dapat digunakan sebagai tempat penyimpanan dan data tersebut dapat dianalisis. Google meluncurkan BigQuery secara publik pada tahun 2012. 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.

2.1.2 *SQL (Structured Query Language) [3]*

SQL adalah bahasa pemrograman menghasilkan, memanipulasi, dan mengambil informasi dari database relasional. BigQuery mendukung dua jenis gaya SQL yaitu *Standard SQL* dan *Legacy SQL*¹. Mengambil informasi dari database relasional harus menggunakan *query*. *Query* merupakan *syntax* atau perintah yang digunakan untuk mengambil dan menghasilkan data dari database.

Query Clauses

Terdapat beberapa komponen atau klausa dari *query* yang digunakan mengambil dan menghasilkan data dari database, seperti:

- SELECT dan FROM

Fungsi dari klausa SELECT adalah untuk menentukan kolom dari suatu tabel yang ditampilkan dalam *query result*. Fungsi dari klausa FROM adalah Mengidentifikasi tabel yang ingin diambil datanya. Dalam mengambil data dari database setidaknya minimal harus menggunakan dua klausa ini. Klausa ini memiliki syntax seperti:

¹<https://cloud.google.com/bigquery/docs/reference/standard-sql/enabling-standard-sql>

```
1  SELECT coloumn1, coloumn2, ...
2  FROM table_name
```

- WHERE

Fungsi dari klause WHERE adalah untuk membatasi jumlah baris dalam *query result* berdasarkan kondisi tertentu. Klause WHERE digunakan jika terdapat beberapa kondisi yang ingin dicari dari database tersebut. Klause ini memiliki syntax seperti:

```
7  SELECT coloumn1, coloumn2, ...
8  FROM table_name
9  WHERE condition
```

- GROUP BY

Fungsi dari kaluse GROUP BY adalah untuk mengelompokkan baris berdasarkan nilai kolom yang sama. Klause ini memiliki syntax seperti:

```
13 SELECT coloumn1, coloumn2, ...
14 FROM table_name
15 WHERE condition
16 GROUP BY column_name, ...
```

- ORDER BY

Fungsi dari kaluse ORDER BY adalah untuk mengurutkan *query result* berdasarkan satu atau lebih kolom. Pada saat menggunakan ORDER BY, akan ditambahkan dua fungsi yaitu ASC (*Ascending*) dan DESC (*Descending*). Klause ini memiliki syntax seperti:

```
21 SELECT coloumn1, coloumn2, ...
22 FROM table_name
23 WHERE condition
24 GROUP BY column_name, ...
25 ORDER BY column_name, ... ASC|DESC
```

Query Aggregation

Didalam *query* juga terdapat beberapa fungsi agregat untuk melakukan operasi tertentu yaitu:

- MAX()

Fungsi ini bertujuan untuk mengembalikan nilai maksimal dari atribut sebuah tabel. Fungsi MAX memiliki contoh *syntax* seperti:

```
31 SELECT MAX(column_name)
32 FROM table_name
33 WHERE condition;
```

- MIN()

Fungsi ini bertujuan untuk mengembalikan nilai minimum dari atribut sebuah tabel. Fungsi MIN memiliki contoh *syntax* seperti:

```
1  SELECT MIN(column_name)
2  FROM table_name
3  WHERE condition;
```

- AVG()

Fungsi ini bertujuan untuk mengembalikan nilai rata-rata dari atribut sebuah tabel. Fungsi AVG memiliki contoh *syntax* seperti:

```
7  SELECT AVG(column_name)
8  FROM table_name
9  WHERE condition;
```

- COUNT() Fungsi ini bertujuan untuk mengembalikan jumlah baris dari atribut sebuah tabel. Fungsi COUNT memiliki contoh *syntax* seperti:

```
12 SELECT COUNT(column_name)
13 FROM table_name
14 WHERE condition;
```

- SUM() Fungsi ini bertujuan untuk mengembalikan jumlah baris dari atribut sebuah tabel. Fungsi SUM memiliki contoh *syntax* seperti:

```
17 SELECT SUM(column_name)
18 FROM table_name
19 WHERE condition;
```

20 Querying Multiple Tables

21 Karena database relasional di-*design* dibentuk dengan mengamanatkan bahwa setiap entitas dibuat
22 kedalam tabel yang terpisah, sehingga dibutuhkan mekanisme untuk menghubungkan beberapa
23 tabel dalam *query* yang sama. Mekanisme ini disebut dengan join. Terdapat beberapa jenis join
24 sebagai berikut:

- LEFT OUTER JOIN

Kata kunci kiri menunjukkan bahwa tabel di sisi kiri klausa from bertanggung jawab untuk
menentukan jumlah baris dalam kumpulan hasil, sedangkan tabel di sisi kanan digunakan
untuk memberikan nilai kolom setiap kali ditemukan kecocokan. LEFT OUTER JOIN
memiliki *syntax* seperti:

```
30 SELECT column_name(s)
31 FROM table1
32 LEFT (OUTER) JOIN table2
33 ON table1.column_name = table2.column_name;
```

- RIGHT OUTER JOIN

Kata kunci kiri menunjukkan bahwa tabel di sisi kanan klausa from bertanggung jawab untuk
menentukan jumlah baris dalam kumpulan hasil, sedangkan tabel di sisi kiri digunakan untuk
memberikan nilai kolom setiap kali ditemukan kecocokan. RIGHT OUTER JOIN memiliki
syntax seperti:

```
1  SELECT column_name(s)
2  FROM table1
3  RIGHT (OUTER) JOIN table2
4  ON table1.column_name = table2.column_name;
```

- FULL OUTER JOIN

Full outer join merupakan gabungan dari LEFT OUTER JOIN dan RIGHT OUTER JOIN. FULL OUTER JOIN memiliki *syntax* seperti:

```
8  SELECT column_name(s)
9  FROM table1
10 FULL OUTER JOIN table2
11 ON table1.column_name = table2.column_name
12 WHERE condition;
```

- INNER JOIN

Inner join menghubungkan dua atau lebih tabel dengan hubungan antara dua kolom. INNER JOIN memiliki *syntax* seperti:

```
16 SELECT column_name(s)
17 FROM table1
18 INNER JOIN table2
19 ON table1.column_name = table2.column_name;
```

Subquery

Subquery merupakan query yang terkandung dalam *query* lain. Sebuah *subquery* selalu diapit dalam tanda kurung, dan biasanya dieksekusi terlebih dahulu sebelum *query* yang memuatnya. Tabel yang dikembalikan oleh *subquery* menentukan bagaimana tabel tersebut dapat digunakan dan operator mana yang dapat digunakan oleh *query* yang memuatnya untuk berinteraksi dengan tabel yang dikembalikan oleh *subquery*. Ketika query yang memuat telah selesai dieksekusi, tabel yang dikembalikan oleh *subquery* akan dibuang, membuat *subquery* bertindak seperti tabel sementara dengan cakupan pernyataan. Salah satu *syntax* pada *subquery* adalah sebagai berikut:

```
28 SELECT column_name(s)
29 FROM (subquery)
```

2.2 HTTP Archive [4]

HTTP Archive adalah sebuah *open-source project* yang melihat bagaimana website dibuat. HTTP Archive menyediakan data-data historis untuk melihat bagaimana website berkembang. HTTP Archive pertama sekali dimulai pada tahun 2010 oleh Steve Souders dan di-maintain oleh Pat Meenan, Rick Viscomi, Paul Calvano, and Barry Pollard. HTTP Archive memiliki keterbatasan seperti HTTP Archive hanya melihat halaman utama. Misalnya sebagian besar *website* terdiri dari banyak halaman web terpisah. Karena batasan ini sehingga ada kemungkinan bahwa suatu halaman yang dianalisis tidak mewakili sebuah situs website. Orang yang menggunakan HTTP Archive adalah anggota komunitas web, para sarjana, dan pemimpin industri:

- Komunitas web menggunakan data ini untuk mempelajari lebih lanjut tentang keadaan web. Biasanya dapat dilihat pada blog, presentasi, atau media sosial.
- Para sarjana mengutip data ini untuk mendukung penelitian dalam publikasi besar seperti ACM dan IEEE.
- Para pemimpin industri menggunakan data ini untuk mengkalibrasi alat mereka untuk secara akurat mewakili bagaimana web dibuat.

2.3 Web Almanac [5]

Web Almanac adalah sebuah proyek yang dikelola oleh HTTP Archive. Misi web almanac adalah menggabungkan statistik mentah dan tren HTTP Archive dengan keahlian komunitas web. Semua metrik yang disediakan oleh web almanac dapat direproduksi secara publik menggunakan dataset di BigQuery. Kueri dapat ditelusuri dengan menggunakan semua bab di repositori GitHub web almanac yang dapat dilihat pada ²:

1. Accessibility

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

2. Caching

Caching adalah teknik yang memungkinkan penggunaan kembali konten yang diunduh sebelumnya. Caching melibatkan sesuatu seperti server atau web browser untuk menyimpan konten dan menandainya agar dapat digunakan kembali.

3. Capabilities

Capabilities memberikan *overview* tentang berbagai API web modern. Hal ini penting untuk menjaga web tetap relevan sebagai platform.

4. CMS

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

5. Compression

Menggunakan HTTP Compression membuat pemuatan situs lebih cepat dan menjamin pengalaman penggunaan yang lebih baik. Penggunaan compression yang efektif dapat mengurangi berat halaman dan meningkatkan kinerja web.

6. CSS

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

7. Ecommerce

Ecommerce platform adalah perangkat lunak atau layanan yang memungkinkan untuk membuat dan mengoperasikan sebuah toko online.

8. Fonts

²<https://github.com/HTTPArchive/almanac.httparchive.org/tree/main/sql/2020>

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

Saat ini, mobile-web sudah menjadi cara utama banyak orang untuk mengakses website. Dalam mobile-web akan terlihat tren saat ini pada mobile-web.

15. Page-weight

Page-weight adalah salah satu metrik sederhana yang tersedia. Memuat sebuah halaman akan memberikan gambaran tentang ukuran dari *resource* yang diambil atau di-*request*.

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.³

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.

³<https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492>

2.5 Pengukuran Aplikasi Usang Pada Beberapa Website Populer Di Indonesia^[6]

Pada bagian ini akan dijelaskan tentang research method dan hasil keseluruhan dari ^[6].

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

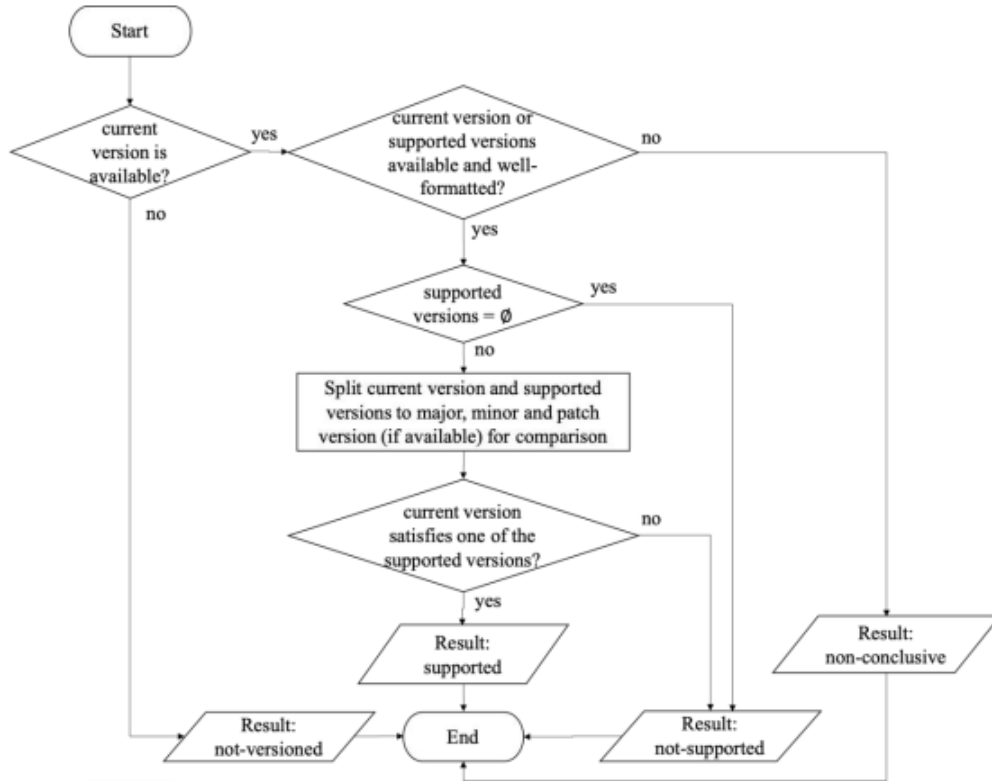
Untuk melihat nomor versi yang masih didukung akan dilakukan pencarian di website resmi dari setiap aplikasi. Terdapat beberapa website yang tidak dapat ditampilkan versinya, sehingga suatu website dapat didefinisikan didukung jika memenuhi kondisi sebagai berikut:

- 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 dapat dilihat pada gambar 2.1

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[6], 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 tabel 2.1

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

Tabel 2.1: Overall application count for measurement result

BAB 3

PERCOBAAN AWAL

Pada bab ini akan dijelaskan analisis masalah penelitian ini. Analisis meliputi Dataset Pada HTTP Archive, Langkah-Langkah Query Yang Dilakukan.

3.1 Eksplorasi Teknologi

Dalam pengerjaan skripsi ini akan menggunakan teknologi bernama BigQuery. Di dalam BigQuery, terdapat salah satu fitur yang akan digunakan yaitu membuat dataset baru. Dataset yang dibuat berisi tabel-tabel yang akan dianalisis. Tabel-tabel tersebut dapat dibuat secara manual maupun di-*upload*.

Berikut ini langkah-langkah dalam pembuatan dataset dan tabel:

1. Membuka Google Cloud Project Page.
2. Membuat atau memilih project yang akan dikerjakan.
3. Membuka console kemudian memilih BigQuery.
4. Pada tab explorer terdapat project kemudian pengguna harus menekan tombol titik tiga dan pilih create dataset.
5. Buka dataset, kemudian pilih menu *create table*.

3.2 Dataset Pada HTTP Archive

Di dalam HTTP Archive terdapat dataset yang dapat diambil menggunakan teknologi BigQuery, dataset tersebut adalah sebagai berikut:

1. almanac
Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.
2. blink_features
Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.
3. core_web_vitals
Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.
4. latest
Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.
5. lighthouse

Dataset pada lighthouse berisi tabel-tabel dari bulan Juni tahun 2017 sampai dengan sekarang yang terdiri dari website pada mobile. Dataset bulan Agustus tahun 2020 baris pada mobile memiliki 6.290.147 baris [3.1](#) yang dapat dianalisis. Masing-masing terdiri dari URL dan report.

- ¹ *URL (Uniform Resource Locator)* merupakan nama-nama domain dan *report*. Tetapi tabel ini
² tidak digunakan dalam pengerjaan skripsi ini.

Row	url	report
1	https://votesearch.utah.gov/	<pre>{ "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36", "environment": { "networkUserAgent": "Mozilla/5.0 (Linux; Android 7.0; Moto G (4)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4143.7 Mobile Safari/537.36", "Chrome-Lighthouse": true, "hostUserAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36", "benchmarkIndex": 506, "lighthouseVersion": "6.1.1", "fetchTime": "2020-08-06T10:36:03.335Z", "requestedUrl": "https://votesearch.utah.gov/", "finalUrl": "https://vote.utah.gov/", "runWarnings": ["The page may not be loading as expected because your test URL (https://votesearch.utah.gov/) was redirected to https://vote.utah.gov/. Try testing the second URL directly."], "audits": { "is-on-https": { "id": "is-on-https", "title": "Does not use HTTPS", "description": "All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](https://developers.google.com/..." } } } }</pre>
2	https://otricolore.ru/	<pre>{ "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.125 Safari/537.36", "environment": { "networkUserAgent": "Mozilla/5.0 (Linux; Android 7.0; Moto G (4)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4143.7 Mobile Safari/537.36", "Chrome-Lighthouse": true, "hostUserAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.125 Safari/537.36", "benchmarkIndex": 456, "lighthouseVersion": "6.2.0", "fetchTime": "2020-08-11T09:30:51.743Z", "requestedUrl": "https://otricolore.ru/", "finalUrl": "https://otricolore.ru/", "runWarnings": [], "audits": { "is-on-https": { "id": "is-on-https", "title": "Uses HTTPS", "description": "All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](https://developers.google.com/web/fundamentals/security/prevent-mixed-content/what-is-mixed-content), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents int..." } } } }</pre>

Tabel 3.1: Lighthouse Data Example

6. pages

Dataset pada pages 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. Masing-masing terdiri dari URL dan payload. *URL (Uniform Resource Locator)* merupakan nama-nama domain dan *payload*. Tetapi tabel ini tidak digunakan dalam pengerjaan skripsi ini.

7. requests

Dataset pada request 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 579.752.745 baris yang dapat dianalisis. Masing-masing terdiri dari URL dan payload. *URL (Uniform Resource Locator)* merupakan nama-nama domain dan *payload*. Tetapi tabel ini tidak digunakan dalam pengerjaan skripsi ini.

8. response_bodies

Dataset pada response_bodies 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 215.621.667 baris dan pada mobile memiliki 270.249.686 baris yang dapat dianalisis. Masing-masing terdiri dari page, URL, body, truncated, dan requestId. Tetapi tabel ini tidak digunakan dalam pengerjaan skripsi ini.

9. sample_data

Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.

10. sample_data_2020

Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.

11. scratchspace

Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.

12. summary_pages

Dataset pada summary_pages berisi tabel-tabel dari bulan November tahun 2010 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 6.347.919 baris yang dapat dianalisis. Masing-masing terdiri dari pageid, createDate, archive, label, crawlid, wptid, wptrun, url, urlShort, urlhash, cdn, startedDateTime, TTFB, renderStart, onContentLoaded, onLoad, fullyLoad, visualComplete, PageSpeed, SpeedIndex, rank, reqTotal, reqHTML, reqJS, reqCSS, reqImg, reqGif, reqJpg, reqPng, reqFont, reqFlash, reqJson, reqOther, bytesTotal, bytesHTML, bytesJS, bytesCSS, bytesImg, bytesGif, bytesJpg, bytesPng, bytesFont, bytesFlash, bytesJson, bytesOther, bytesHtmlDoc, numDomains, maxDomainReqs, numRedirects, numErrors, numGlibs, numHttps, numCompressed, numDomElements, maxageNull, maxage0, maxage1, maxage30, maxage365, maxageMore, gzipTotal, gzipSavings, _connections, _adult_site, avg_dom_depth, document_height, document_width, localStorage_size, sessionStorage_size, num_iframes, num_scripts, doctype, meta_viewport, reqAudio, reqVideo, reqText, reqXml, reqWebp, reqSvg, bytesAudio, bytesVideo, bytesText, bytesXml, bytesWebp, bytesSvg, num_scripts_async, num_scripts_sync, usertiming. Tetapi

tabel ini tidak digunakan dalam pengerjaan skripsi ini.

13. summary_requests

Dataset pada `summary_requests` berisi tabel-tabel dari bulan November tahun 2010 sampai dengan sekarang yang terdiri dari website pada desktop. Dataset bulan Agustus tahun 2020 baris pada desktop memiliki 215.621.667 baris dan pada mobile memiliki 1.234.599 baris yang dapat dianalisis. Masing-masing terdiri dari `requestid`, `pageid`, `startedDateTime`, `time`, `method`, `url`, `urlShort`, `redirectUrl`, `firstReq`, `firstHtml`, `reqHttpVersion`, `reqHeaderSize`, `reqBodySize`, `reqCookieLen`, `reqOtherHeader`, `status`, `respHttpVersion`, `respHeaderSize`, `respBodySize`, `respSize`, `respCookieLen`, `expAge`, `mimeType`, `respOtherHeader`, `req_accept`, `req_accept_charset`, `req_accept_encoding`, `req_accept_language`, `req_connection`, `req_host`, `req_if_modified_since`, `req_if_none_match`, `req_referer`, `req_user_agent`, `resp_accept_ranges`, `resp_age`, `resp_cache_control`, `resp_connection`, `resp_content_encoding`, `resp_content_language`, `resp_content_length`, `resp_content_location`, `resp_content_type`, `resp_date`, `resp_etag`, `resp_expires`, `resp_keep_alive`, `resp_last_modified`, `resp_location`, `resp_pragma`, `resp_server`, `resp_transfer_encoding`, `resp_vary`, `resp_via`, `resp_x_powered_by`. Tetapi tabel ini tidak digunakan dalam pengerjaan skripsi ini.

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 dapat dilihat pada gambar 3.2 dan pada mobile memiliki 67.452.994 baris 3.3 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.

Row	url	category	app	info
1	https://www.3-king.com/	Analytics	Google Analytics	
2	https://www.fleabites.net/	Miscellaneous	Twitter Emoji (Twemoji)	
3	http://www.elcarnicero.cl/	Widgets	OWL Carousel	
4	https://thankyou.ws/	Analytics	Google Analytics	
5	https://rogerwaters.com/	Reverse proxies	Nginx	
6	http://www.palaciodelaslampadas.com.br/	JavaScript libraries	jQuery	2.1.1
7	https://copenhagencamping.dk/	CMS	WordPress	
8	https://eachat.ma/	Ecommerce	WooCommerce	4.3.0
9	https://advokat-bondarchuk.ru/	Blogs	WordPress	
10	https://passport.rsl.ru/	JavaScript libraries	jQuery	1.7.1

Tabel 3.2: Technologies Desktop Data Sample

Row	url	category	app	info
1	http://www.carobd.fr/	UI frameworks	Bootstrap	4.1.3
2	http://www.minikabebe.com/	Font scripts	Font Awesome	
3	https://sibirskisamojedcom.wordpress.com/	Blogs	WordPress	
4	https://www.peauideale.com/	Analytics	Google Analytics	
5	https://www.bestcours.com/	JavaScript libraries	jQuery	1.11.1
6	https://www.chirurgo-stefanoenrico.it/	UI frameworks	Bootstrap	
7	https://retrocores.com/	JavaScript libraries	jQuery	1.12.4
8	https://pakmule.com/	Web servers	Apache	
9	https://edilsonalves.com.br/	JavaScript libraries	jQuery	1.12.4
10	https://mobilierdasie.com/	Ecommerce	Google Analytics Enhanced eCom- merce	

Tabel 3.3: Technologies Mobile Data Sample

15. urls

Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.

16. wappalyzer

Pada tabel ini tidak terdapat keterangan dan tidak berhubungan dengan skripsi ini.

3.3 Langkah-Langkah Query Yang Dilakukan

Pada section ini akan dijelaskan tentang langkah-langkah query yang dilakukan dalam memperoleh data dan analisis yang dilakukan. Data yang diambil merupakan dataset dari tabel technologies 2020_08_01:

3.3.1 Mengumpulkan List Website

Langkah pertama yang dilakukan yaitu mengumpulkan website. Website yang dicari tidak berdasarkan berdasarkan *rank* karena tidak tersedia pada dataset tersebut. Berikut adalah *query* yang digunakan untuk mengumpulkan list website.

```
SELECT url
FROM 'httparchive.technologies.2020_08_01_01'
ORDER BY url asc
```

3.3.2 Mencari Aplikasi Yang Digunakan Website

Setiap website akan dicari aplikasi apa saja yang digunakan dalam pembangunan website tersebut dan versi dari aplikasi yang dipakainya. Berikut adalah query yang digunakan.

```
SELECT url, app, info
FROM 'httparchive.technologies.2020_08_01_01'
ORDER BY url asc
```


3.3.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai

Pengelompokan aplikasi dapat dilakukan dengan menggunakan query. Berikut adalah query yang digunakan.

```
SELECT versioned.app, num.num_sites , unversioned.unversioned_count , versioned.versioned_count
FROM
(SELECT app, count(*) AS unversioned_count
FROM 'httparchive.technologies.2020_08_01_*'
WHERE info = ""
GROUP BY app) AS unversioned
FULL OUTER JOIN
(SELECT app, count(*) AS versioned_count
FROM 'httparchive.technologies.2020_08_01_*'
WHERE info != ""
GROUP BY app) AS versioned
ON unversioned.app = versioned.app
JOIN
(SELECT app, count(url) AS num_sites
FROM 'httparchive.technologies.2020_08_01_*'
GROUP BY app) as num
ON versioned.app = num.app
WHERE versioned_count != 0 AND versioned.app is not null
```

3.3.4 Mencari Data Tentang Versi Aplikasi Yang Masih Didukung

Sebelum menentukan suatu aplikasi usang atau tidak, kita harus mencari versi dari setiap aplikasi secara manual. Versi setiap aplikasi dapat dilihat di *official documentation* dari setiap aplikasi. Hasil pencarian dari aplikasi yang masih didukung dapat dilihat pada link: <https://docs.google.com/spreadsheets/d/1P627vSIS7-f4E4U3zHONbwetftRlmfKQj98zk5RMAE4/edit#gid=1378116505>.

3.3.5 Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai Sekarang Dengan Versi Aplikasi Yang Masih Didukung

DAFTAR REFERENSI

- [1] Tigani, J. dan Naidu, S. (2014) *Google BigQuery Analytics*, 1 edition. Wiley.
- [2] Developer, G. Bigquery. <https://cloud.google.com/bigquery/docs/introduction>.
- [3] Beaulieu, A. (2005) *Learning SQL*, 1st ed edition. O'Reilly Media.
- [4] Souders, S. Http archive. <https://httparchive.org/faq>.
- [5] Pyltsyn, A. Web almanac. <https://almanac.httparchive.org/en/2020/methodology>.
- [6] Nugroho, P. A. dan Steven, H. (2013) Measuring unsupported applications in indonesia popular websites. *JITEKI*, **66**, 595–614.

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
```

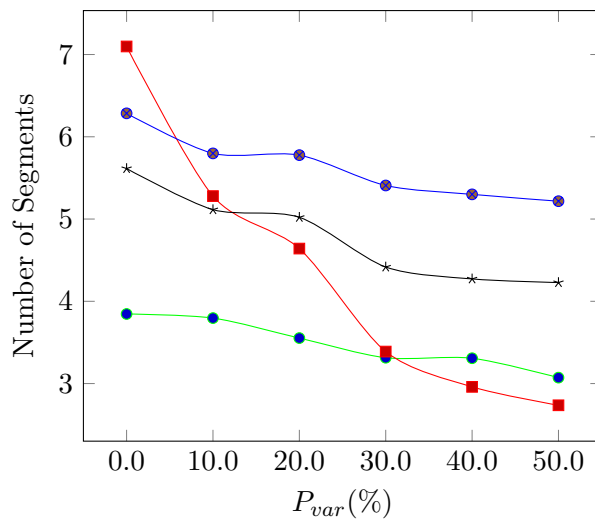
Kode A.2: MyCode.java

```
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

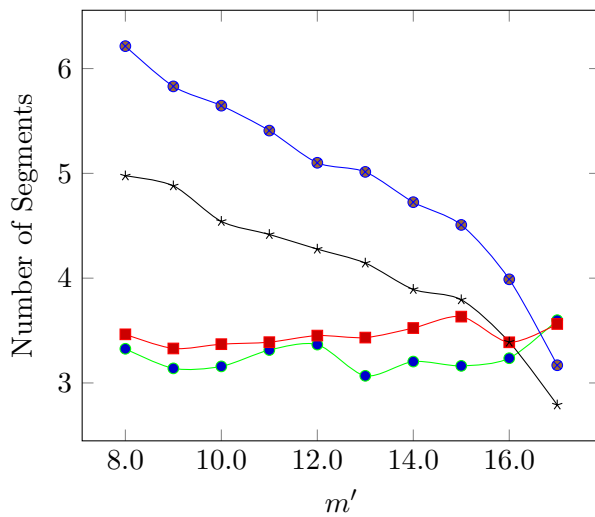
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4