

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 OSEMN 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
2.6 ReactJS	11
2.6.1 Node Package Manager	12
2.6.2 NPM CLI	12
2.6.3 NPX	12
2.7 JSON	12
3 PERCOBAAN AWAL	13
3.1 Eksplorasi Teknologi	13
3.1.1 BigQuery	13
3.1.2 ReactJS	15
3.2 Dataset Pada HTTP Archive	16
3.3 Langkah-Langkah Query Yang Dilakukan	19
3.3.1 Mengumpulkan List Website	20
3.3.2 Mencari Aplikasi Yang Digunakan Website	20
3.3.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai	20
3.3.4 Mencari Data Tentang Versi Aplikasi Yang Masih Didukung	21

3.3.5	Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai Sekarang Dengan Versi Aplikasi Yang Masih Didukung	21
3.4	Hasil Sample Data Dengan Beberapa Aplikasi	21
4	PENGKALIAN DATA	23
4.1	Langkah-Langkah Query Yang Dilakukan Dengan Data Yang Lebih Besar	23
4.1.1	Mengumpulkan List Website	23
4.1.2	Mencari Aplikasi Yang Digunakan Website	23
4.1.3	Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai	23
4.1.4	Mencari Data Tentang Versi Aplikasi Yang Masih Didukung	24
4.1.5	Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai Sekarang Dengan Versi Aplikasi Yang Masih Didukung	24
4.2	Hasil Sample Data	24
4.2.1	Cherokee	25
4.2.2	Angular Material	25
	DAFTAR REFERENSI	27
	A KODE PROGRAM	29
	B HASIL EKSPERIMEN	31

DAFTAR GAMBAR

2.1	<i>Algorithm to compare current version versus supported versions</i>	11
3.1	Google Cloud Project Page	13
3.2	Create atau Open Project	14
3.3	Membuka BigQuery	14
3.4	Membuat Dataset Baru	15
3.5	Membuat Tabel Baru	15
3.6	Data Sample Jumlah Aplikasi Dengan Versi yang Dipakai	21
4.1	Aplikasi Cherokee	25
4.2	Aplikasi Angular Material	25
B.1	Hasil 1	31
B.2	Hasil 2	31
B.3	Hasil 3	31
B.4	Hasil 4	31

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 mesin komputasi yang menganalisa data. BigQuery dapat digunakan sebagai tempat penyimpanan dan data tersebut dapat dianalisis. Data dalam BigQuery dimasukkan dalam sebuah dataset. Dataset berisikan tabel-tabel yang 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

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

datanya. Dalam mengambil data dari database setidaknya minimal harus menggunakan dua klause ini. Klause ini memiliki syntax seperti:

```
SELECT column1, column2, ...  
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:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition
```

- GROUP BY

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

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition  
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:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition  
GROUP BY column_name, ...  
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:

```
SELECT MAX(column_name)  
FROM table_name  
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

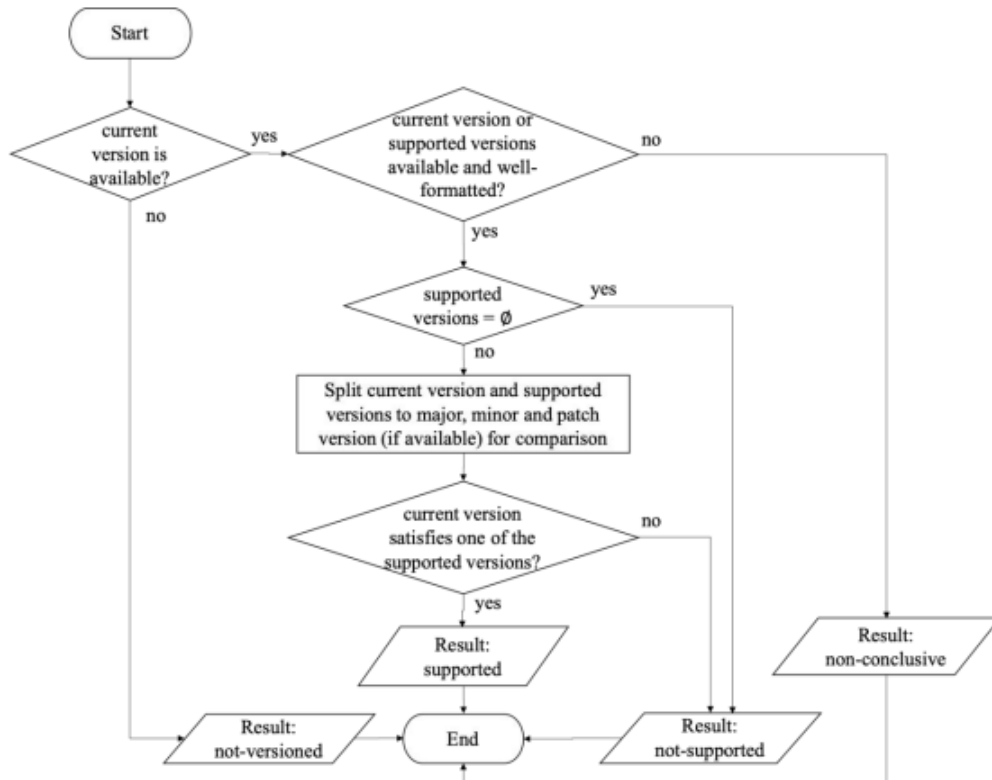
Untuk melihat nomor versi yang masih didukung akan dilakukan pencarian di website resmi dari setiap aplikasi. Terdapat beberapa website yang tidak dapat ditampilkan versinya, sehingga suatu website dapat didefinisikan didukung jika memenuhi kondisi sebagai berikut:

- 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

2.6 ReactJS

ReactJS merupakan *library* yang disediakan JavaScript untuk membuat *interface*. ReactJS dibuat oleh Facebook. Berikut ini contoh sintaks pada ReactJS:

Berikut ini adalah contoh sintaks ReactJS:

```

class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

```

```
17     );  
28   }  
39 }  
40  
51 ReactDOM.render(  
62   <HelloMessage name="World" />,  
73   document.getElementById('hello-example')  
84 );
```

2.6.1 Node Package Manager

Node Package Manager (NPM) adalah *software registry* yang digunakan untuk meminjam atau membagikan *software library* [7]. NPM terdiri dari tiga komponen penting, yaitu:

- NPM website.
- NPM CLI (*Command Line Interface*).
- NPM *Registry*.

Beberapa kegunaan dari menggunakan NPM adalah:

- Membagikan kode kepada pengguna NPM lainnya dimanapun.
- Men-download *software library*
- Menjalankan package tanpa harus meng-install npx

2.6.2 NPM CLI

NPM merupakan *package manager* untuk *Node JavaScript*. NPM menempatkan modul sehingga dapat ditemukan oleh *node*. Selain itu NPM juga dapat mengelola *dependency conflicts*. NPM digunakan untuk menginstall dan mengembangkan *node program*.. Dalam penulisan NPM dapat dilakukan didalam CLI (*Command Line Interface*). NPM memiliki tiga komponen penting dalam penulisan perintah CLI, komponen tersebut seperti:

```
npm <command> [args]
```

2.6.3 NPX

NPX merupakan *execute NPM package binaries*.. NPX digunakan untuk menjalankan *command* yang dimiliki NPM. NPX mengeksekusi file *binary* dari *package Node.js*, baik yang sudah terinstal maupun yang belum.

2.7 JSON

JSON (JavaScript Object Notation) merupakan format penulisan data yang mudah untuk dibaca manusia maupun mesin. JSON adalah format teks yang bersifat *language independent* tetapi menggunakan konvensi yang akrab bagi programmer C, Java, JavaScript, Perl, Python, dan banyak lainnya. Properti ini menjadikan JSON sebagai bahasa pertukaran data yang ideal. Terdapat dua struktur dalam JSON:

- Kumpulan pasangan nilai yang akan dibuat menjadi sebuah objek, *hash table*, dan lainnya.
- Daftar nilai yang diurutkan, seperti array, vektor, dan lainnya.

BAB 3

PERCOBAAN AWAL

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

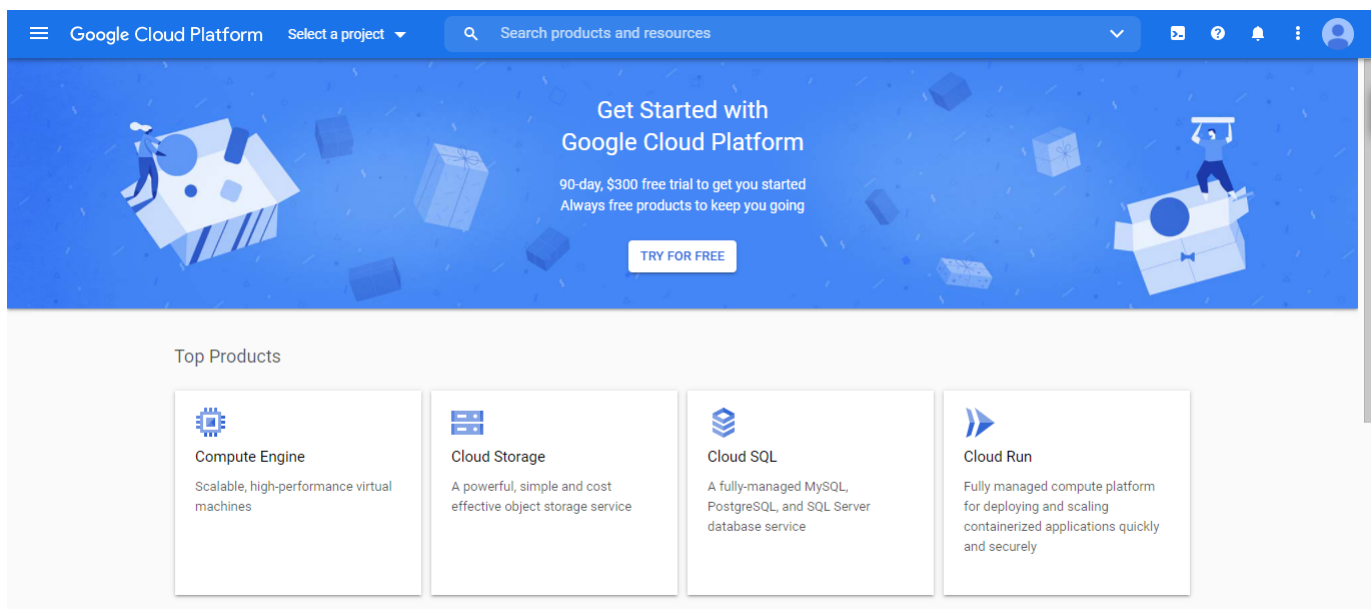
3.1 Eksplorasi Teknologi

3.1.1 BigQuery

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

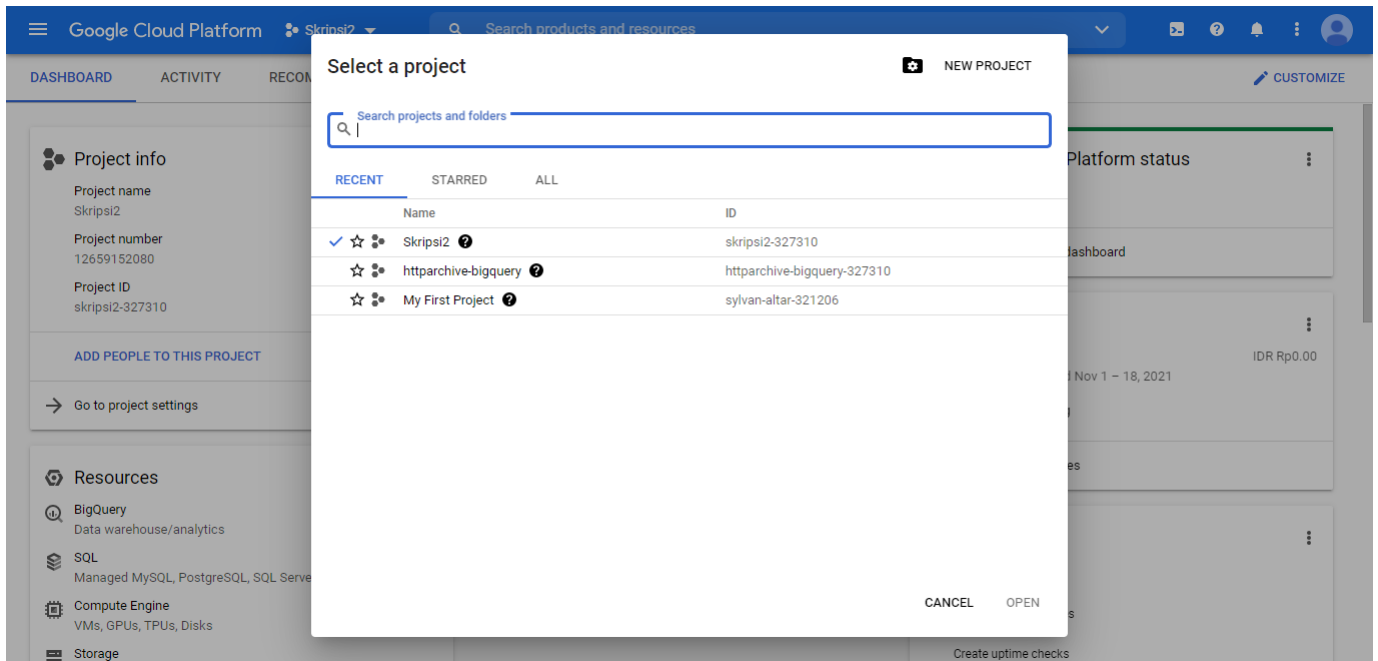
Berikut ini langkah-langkah dalam pembuatan dataset dan tabel:

1. Membuka [Google Cloud Project Page](#).



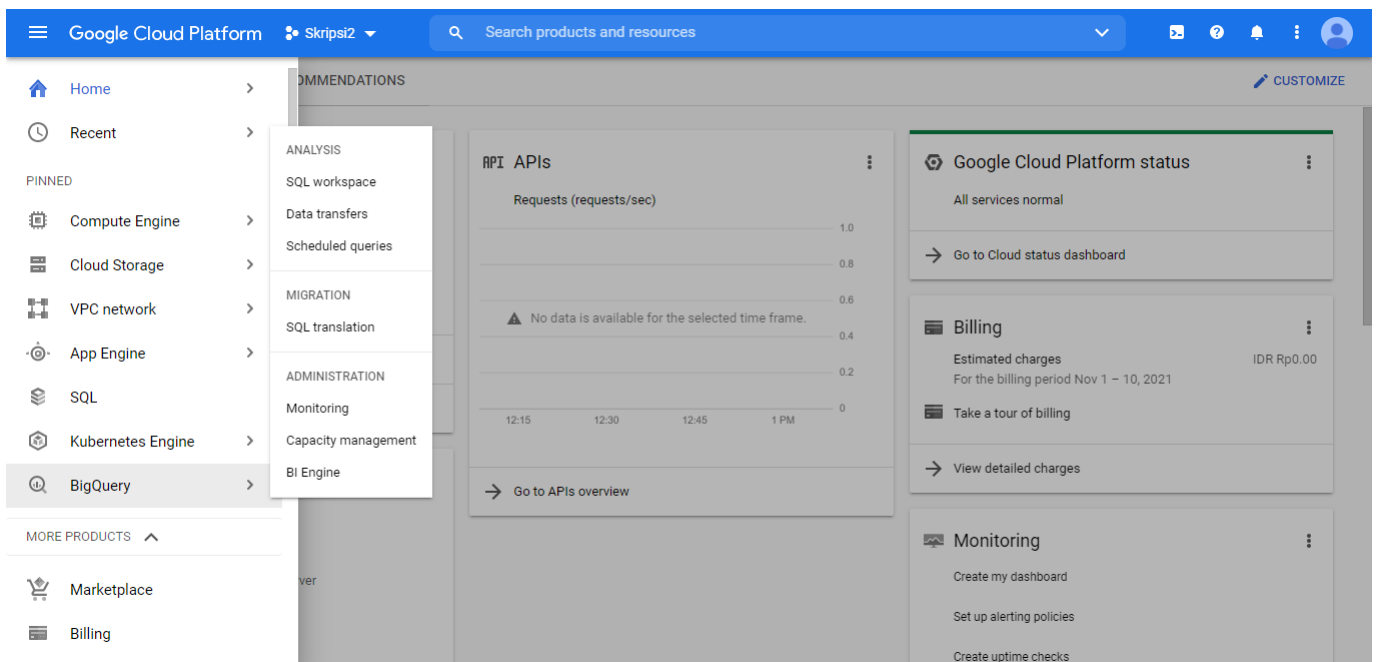
Gambar 3.1: Google Cloud Project Page

2. Membuat atau memilih *project* yang akan dikerjakan.



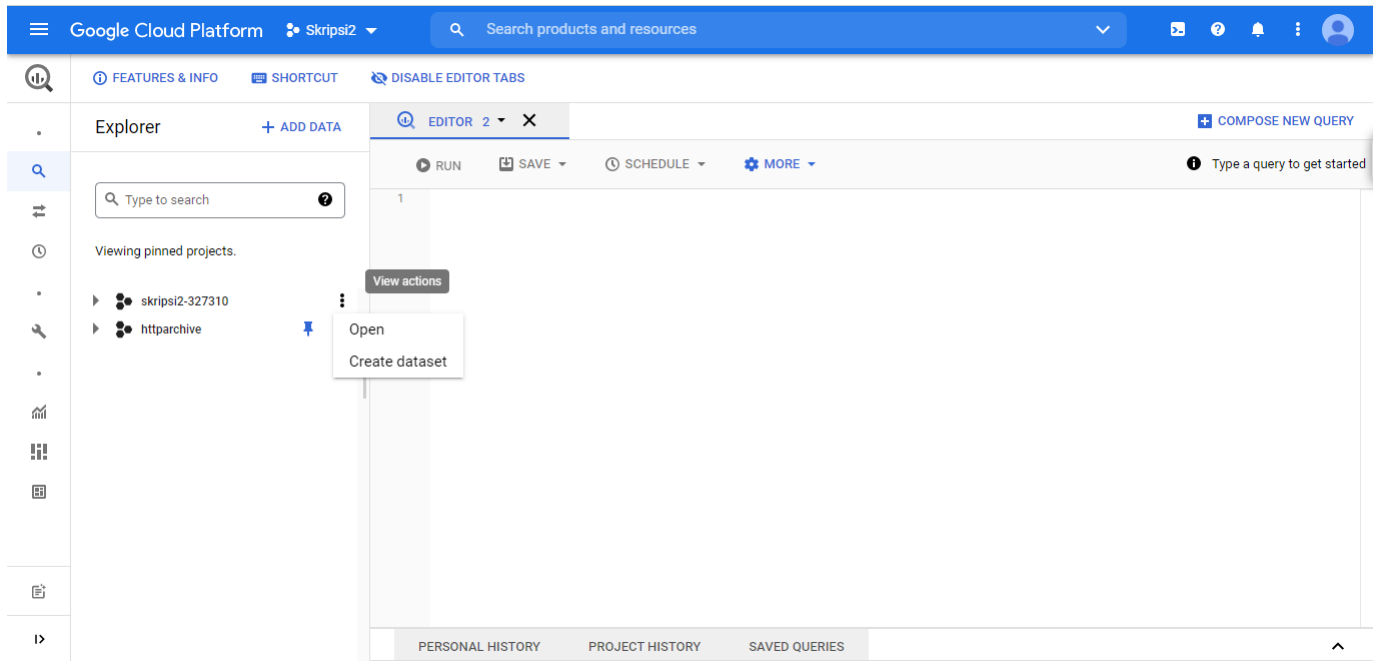
Gambar 3.2: Create atau Open Project

- 1 3. Membuka *console* kemudian memilih BigQuery.



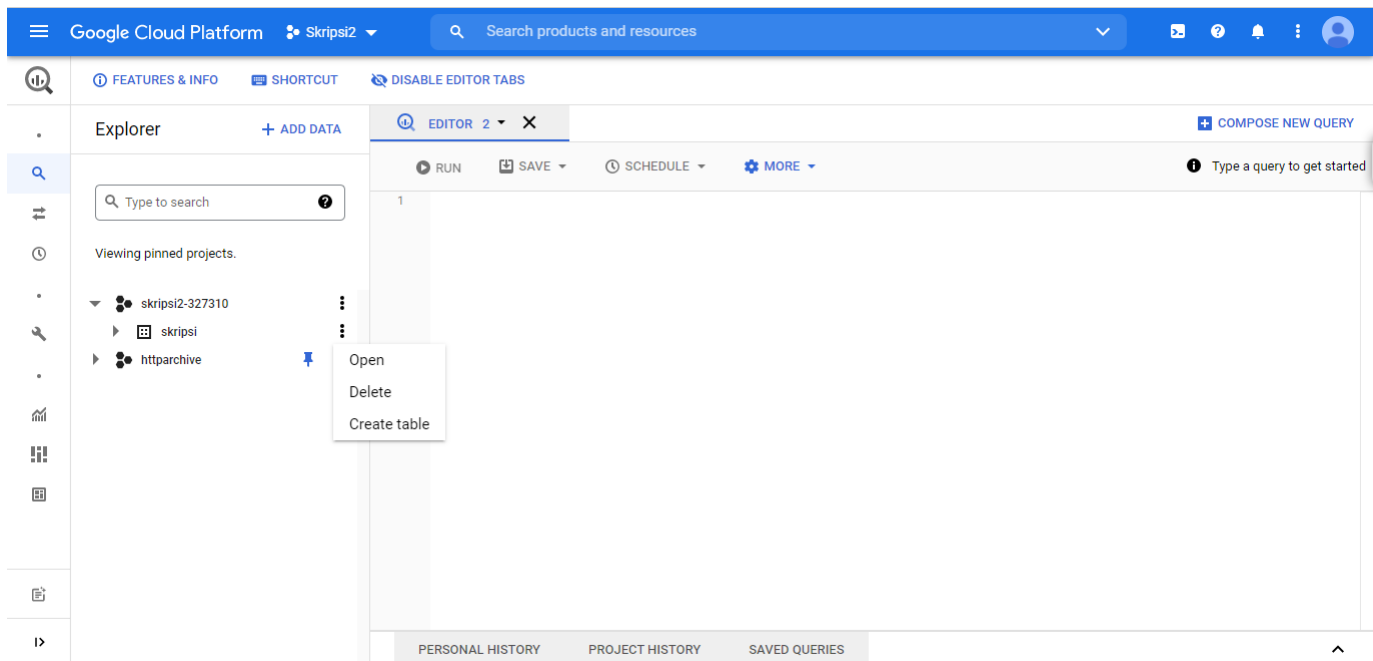
Gambar 3.3: Membuka BigQuery

- 2 4. Pada tab explorer terdapat project kemudian pengguna harus menekan tombol titik tiga dan
- 3 pilih *create dataset*.



Gambar 3.4: Membuat Dataset Baru

- 1 5. Buka dataset, kemudian pilih menu *create table*.



Gambar 3.5: Membuat Tabel Baru

2 3.1.2 ReactJS

- 3 Selain penggunaan BigQuery, skripsi ini juga menggunakan ReactJS dalam pembuatan charts
- 4 maupun melakukan perbandingan versi dari aplikasi. Data yang dipakai untuk membuat chart
- 5 merupakan data JSON yang sudah di-*download* melalui BigQuery. Pembuatan project react dapat
- 6 dilakukan dengan menggunakan sintaks:

```

1 npx create-react-app my-app
2 cd my-app
3 npm start

```

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

url	https://votesearch.utah.gov/
report	{ "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", "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...)" } } }

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. Contoh data dapat dilihat pada tabel 3.2. 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.

url	https://tutorinmobiliario.cl/
payload	{ "startedDateTime": "2020-08-14T17:45:37.606+00:00", "title": "Run 1, First View for https://tutorinmobiliario.cl/ ", "id": "page_1_0_1", "pageTimings": { "onLoad": 27048, "onContentLoad": -1, "startRender": 6500 }, "_cpu.BlinkGC.LazySweepInIdle": 10, "_testStartOffset": 0, "_start_epoch": 0, "_cpu.ParseAuthorStyleSheet": 95, "_bytesOutDoc": 208779, "_cpu.V8.GC_MC_CLEAR_STRING_TABLE": 1, "_cpu.V8.GC_SCAVENGER_SCAVENGE_UPDATE_REFS": 0, "_cpu.V8.GC_MC_MARK_EMBEDDER_TRACING_CLOSURE": 0, "_cpu.V8.GC_MC_MARK_FINISH_INCREMENTAL": 0, "_firstPaint": 6445.524999995541, "_cpu.BlinkGC.AtomicPauseMarkEpilogue": 0, "_cpu.V8.GC_MC_INCREMENTAL_EMBEDDER_PROLOGUE": 7, "_cpu.V8.GC_SCAVENGER_COMPLETE_SWEEP_ARRAY_BUFFERS": 0, "_cpu.V8.GC_MC_EVACUATE_REBALANCE": 0, "_optimization_checked": 1, "_cpu.V8.GC_MC_MARK_ROOTS": 0, "_cpu.BlinkGC.IncrementalMarkingStartMarking": 4, "_responses_404": 0, "_URL": " https://tutorinmobiliario.cl/ ", "_cpu.V8.GC_SCAVENGER_SCAVENGE_ROOTS": 3, "_loadEventStart": 27048, "_cpu.EvaluateScript": 452, "_score_gzip": 100, "_cpu.V8.GC_MC_MARK_WEAK_CLOSURE_EPHEMERON..." }

Tabel 3.2: Pages Data Example

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 response_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 3.3 dan pada mobile memiliki 67.452.994 baris 3.4 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 3.3: 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://pakhmule.com/	Web servers	Apache	
9	https://edilsonalves.com.br/	JavaScript libraries	jQuery	1.12.4
10	https://mobiliertasie.com/	Ecommerce	Google Analytics Enhanced eCommerce	

Tabel 3.4: Technologies Mobile Data Sample

1 15. urls

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

3 16. wappalyzer

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

5 3.3 Langkah-Langkah Query Yang Dilakukan

6 Pada section ini akan dijelaskan tentang langkah-langkah query yang dilakukan dalam memperoleh
7 data dan analisis yang dilakukan. Data yang diambil adalah data percobaan sebanyak 100 data.

8 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_*'
ORDER BY url asc
LIMIT 100
```

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_*'
ORDER BY url asc
LIMIT 100
```

3.3.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai

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

```
SELECT tableName.app, num.num_sites , versioned.versioned_count , unversioned.unversioned_count
FROM
(SELECT DISTINCT app
FROM 'httparchive.technologies.2020_08_01_*' ) tableName
LEFT JOIN
(SELECT tabel1.app, count(app) AS versioned_count
FROM 'httparchive.technologies.2020_08_01_*' AS tabel1
WHERE tabel1.app!="" AND tabel1.info != ""
GROUP BY tabel1.app) AS versioned
ON(versioned.app = tableName.app)
LEFT JOIN
(SELECT tabel2.app, count(app) AS unversioned_count
FROM 'httparchive.technologies.2020_08_01_*' AS tabel2
WHERE tabel2.app!="" AND tabel2.info = ""
GROUP BY tabel2.app) AS unversioned
```

```

1
2 ON (unversioned.app = tabelName.app)
3
4 LEFT JOIN
5
6 (SELECT app, count(url) AS num_sites
7 FROM 'httparchive.technologies.2020_08_01_*'
8 GROUP BY app) AS num
9
10 ON (tabelName.app = num.app)
11 LIMIT 100

```

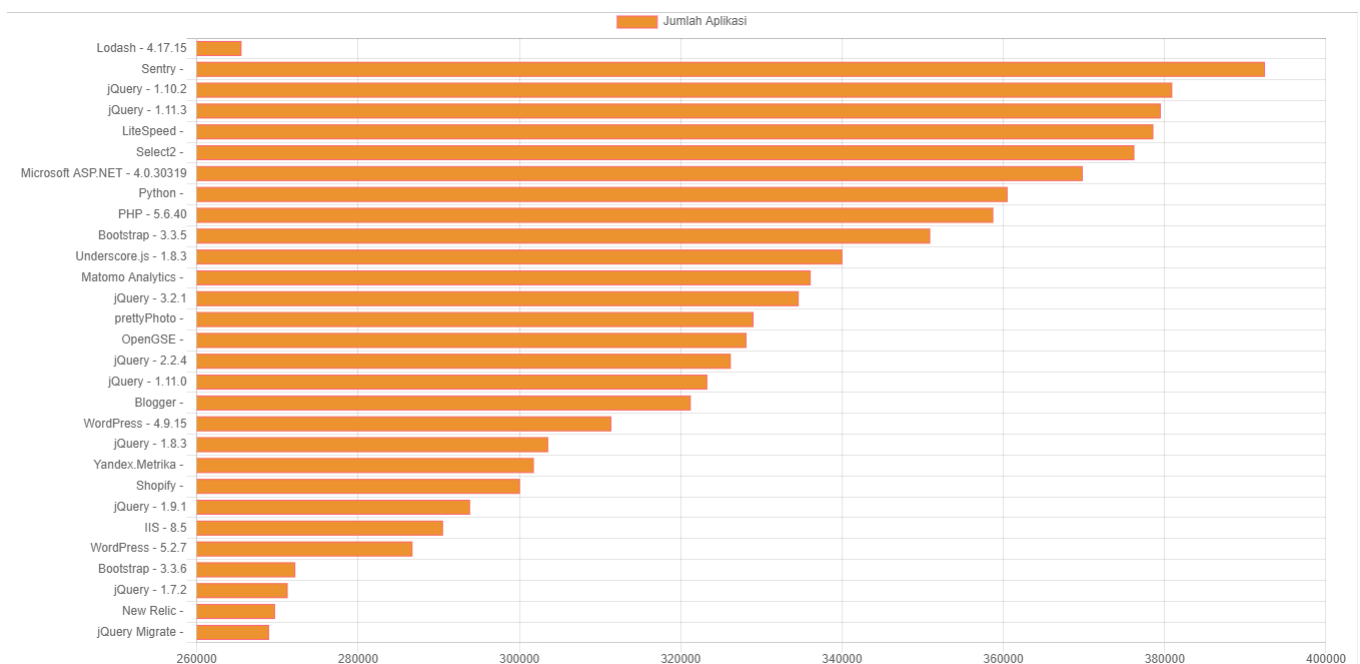
3.3.4 Mencari Data Tentang Versi Aplikasi Yang Masih Didukung

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 gambar ??.

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

3.4 Hasil Sample Data Dengan Beberapa Aplikasi

Diambil 29 data sample dengan aplikasi dan nomor versinya. Pada gambar 3.6 aplikasi Lodash dengan versi 4.7.15 memiliki jumlah 265.552 kali dipakai oleh website.



Gambar 3.6: Data Sample Jumlah Aplikasi Dengan Versi yang Dipakai

BAB 4

PENGGALIAN DATA

Pada bab ini akan dijelaskan analisis masalah penelitian ini. Analisis meliputi Langkah-Langkah Query Yang Dilakukan Dengan Data Yang Lebih Besar.

4.1 Langkah-Langkah Query Yang Dilakukan Dengan Data Yang Lebih Besar

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

4.1.1 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_*'
ORDER BY url asc
```

4.1.2 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_*'
ORDER BY url asc
```

4.1.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai

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

```

1 SELECT tabelName.app, num.num_sites , versioned.versioned_count , unversioned.unversioned_coun
2 FROM
3 (SELECT DISTINCT app
4 FROM 'httparchive.technologies.2020_08_01_*' ) tabelName
5
6 LEFT JOIN
7
8 (SELECT tabel1.app, count(app) AS versioned_count
9 FROM 'httparchive.technologies.2020_08_01_*' AS tabel1
10 WHERE tabel1.app!="" AND tabel1.info != ""
11 GROUP BY tabel1.app) AS versioned
12
13 ON(versioned.app = tabelName.app)
14
15 LEFT JOIN
16
17 (SELECT tabel2.app, count(app) AS unversioned_count
18 FROM 'httparchive.technologies.2020_08_01_*' AS tabel2
19 WHERE tabel2.app!="" AND tabel2.info = ""
20 GROUP BY tabel2.app) AS unversioned
21
22 ON (unversioned.app = tabelName.app)
23
24 LEFT JOIN
25
26 (SELECT app, count(url) AS num_sites
27 FROM 'httparchive.technologies.2020_08_01_*'
28 GROUP BY app) AS num
29
30 ON (tabelName.app = num.app)

```

31 4.1.4 Mencari Data Tentang Versi Aplikasi Yang Masih Didukung

32 Sebelum menentukan suatu aplikasi usang atau tidak, kita harus mencari versi dari setiap aplikasi
 33 secara manual. Versi setiap aplikasi dapat dilihat di-*official documentation* dari setiap aplikasi.
 34 Hasil pencarian dari aplikasi yang masih didukung dapat dilihat pada gambar ??.

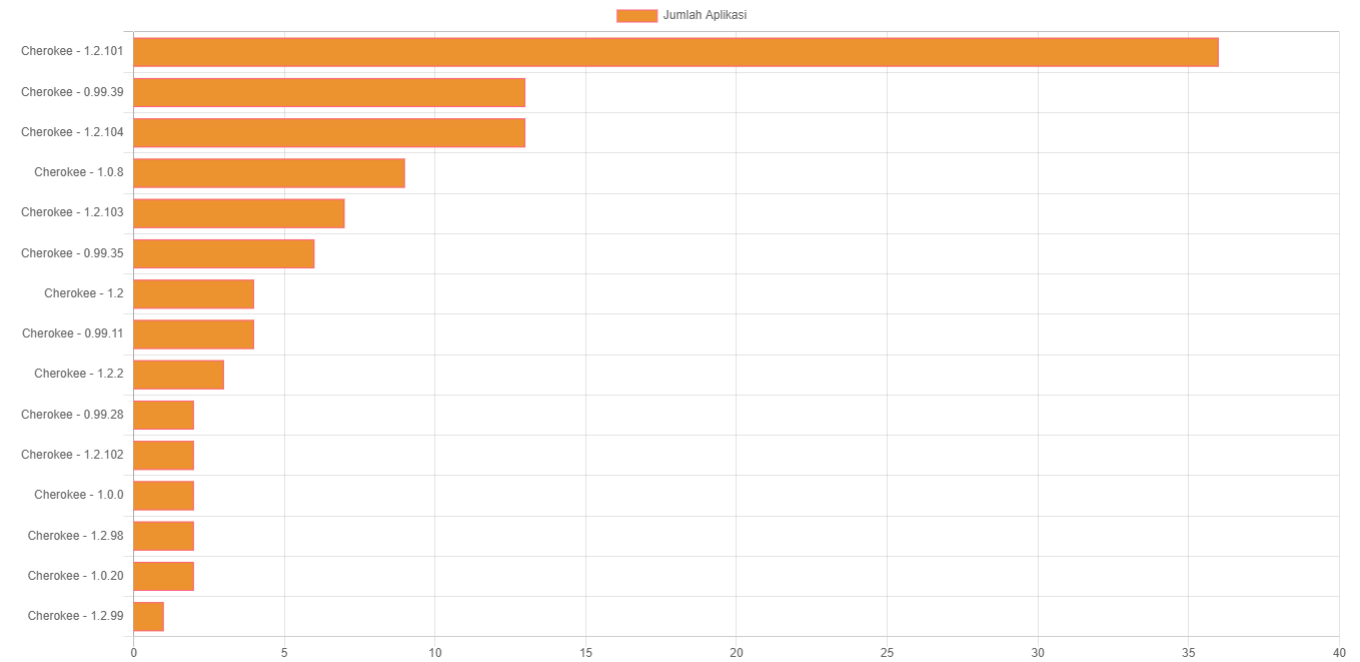
35 4.1.5 Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai 36 Sekarang Dengan Versi Aplikasi Yang Masih Didukung

37 4.2 Hasil Sample Data

38 Data yang ditampilkan adalah data beberapa aplikasi yang sudah dipisahkan berdasarkan aplikasi
 39 dan nomor versi dari aplikasi yang dipakai serta jumlahnya dalam bentuk *chart*.

1 4.2.1 Cherokee

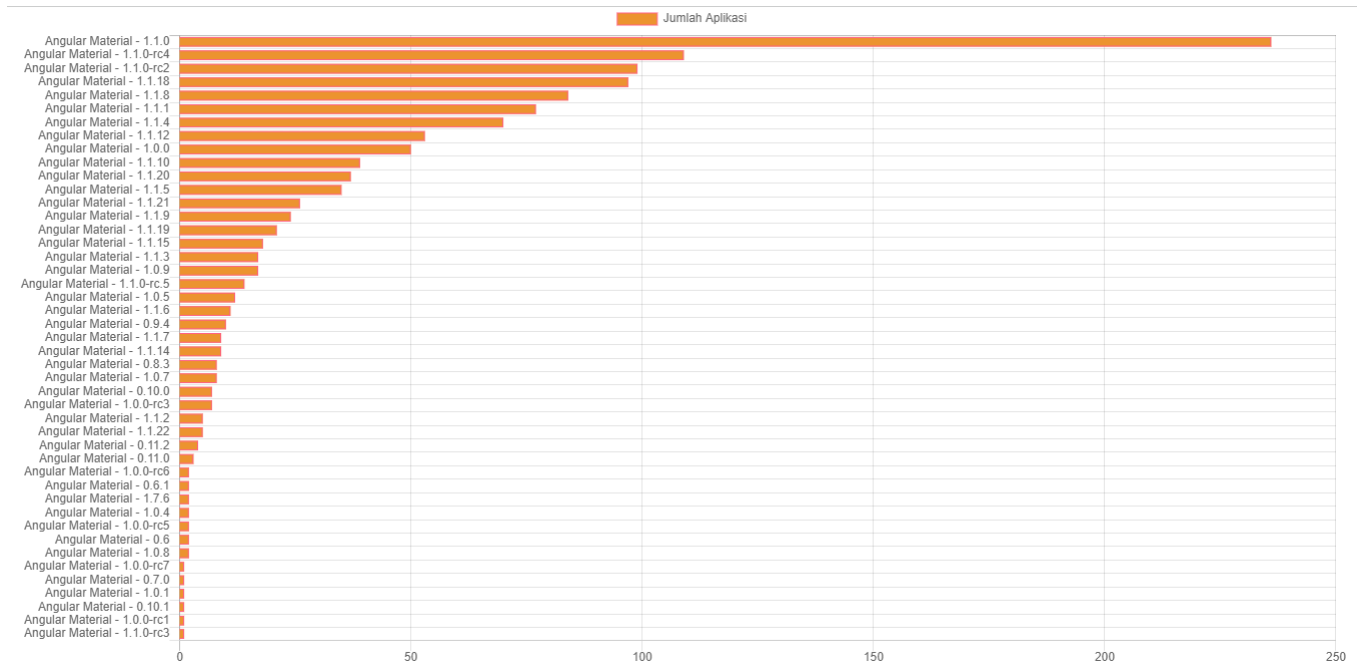
2 Berikut ini adalah chart yang dapat dilihat:



Gambar 4.1: Aplikasi Cherokee

3 4.2.2 Angular Material

4 Berikut ini adalah chart yang dapat dilihat:



Gambar 4.2: Aplikasi Angular Material

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.
- [7] Dahl, R. npmjs. <https://docs.npmjs.com/about-npm>.

LAMPIRAN A
KODE PROGRAM

Tabel A.1: Generated by Spread-Latex

app	num_sites	unversioned_count	versioned_count	website
jQuery	10003030	24,029	9,979,001	https://jquery.com
WordPress	7494642	2,603,276	4,891,366	https://wordpress.org
jQuery Migrate	3597289	268,962	3,328,327	https://github.com/jquery/jquery-migrate
Font Awesome	3827039	1,040,702	2,786,337	https://fontawesome.com/
jQuery UI	2575775	33,127	2,542,648	http://jqueryui.com
jQuery UI	2575775	33,127	2,542,648	http://jqueryui.com
Bootstrap	3391648	868,637	2,523,011	https://getbootstrap.com
Bootstrap	3391648	868,637	2,523,011	https://getbootstrap.com
PHP	5977790	3,455,170	2,522,620	http://php.net
Nginx	7230612	5,440,268	1,790,344	http://nginx.org/en
Modernizr	1900060	231,481	1,668,579	https://modernizr.com
Yoast SEO	1366287	49	1,366,238	http://yoast.com
Apache	4067380	2,949,180	1,118,200	http://apache.org
FancyBox	827596	72,541	755,055	http://fancyapps.com/fancybox
IIS	962466	218,661	743,805	http://www.iis.net
WooCommerce	616072	75,796	540,276	https://woocommerce.com
Moment.js	555640	37,730	517,910	https://momentjs.com
Lodash	517688	10,781	506,907	http://www.lodash.com
Underscore.js	540820	101,362	439,458	http://underscorejs.org
MediaElement.js	420046	33	420,013	http://www.mediaelementjs.com
MediaElement.js	420046	33	420,013	http://www.mediaelementjs.com
Microsoft ASP.NET	865276	457,910	407,366	https://www.asp.net
Revslider	502266	97,735	404,531	https://revolution.themepunch.com/
RequireJS	334538	22,537	312,001	http://requirejs.org
OpenSSL	279452	46	279,406	http://openssl.org
Typekit	253890	687	253,203	http://typekit.com
Hammer.js	302093	52,360	249,733	https://hammerjs.github.io
Google PageSpeed	225920	18	225,902	http://developers.google.com/speed/pagespeed/mod
Handlebars	249598	32,220	217,378	http://handlebarsjs.com
YUI	201973	486	201,487	http://yuilibrary.com
Drupal	256421	64,326	192,095	https://drupal.org
MooTools	190400	2,356	188,044	https://mootools.net
ZURB Foundation	273337	90,398	182,939	http://foundation.zurb.com
Backbone.js	178970	2,540	176,430	http://backbonejs.org
All in One SEO Pack	170116	35	170,081	https://wordpress.org/plugins/all-in-one-seo-pack/
Elementor	275007	110,287	164,720	https://elementor.com
AngularJS	170870	8,261	162,609	https://angularjs.org
Mustache	160656	1,263	159,393	https://mustache.github.io
Prototype	151881	8,485	143,396	http://www.prototypejs.org
Vue.js	249087	127,606	121,481	https://vuejs.org
React	501584	402,192	99,392	https://reactjs.org
React	501584	402,192	99,392	https://reactjs.org
jQuery Mobile	105150	6,933	98,217	https://jquerymobile.com
Varnish	307088	215,982	91,106	http://www.varnish-cache.org
VideoJS	139619	58,869	80,750	http://videojs.com
Leaflet	79227	1,093	78,134	http://leafletjs.com
NextGEN Gallery	78889	799	78,090	https://www.imagely.com/wordpress-gallery-plugin
Dojo	83516	8,379	75,137	https://dojotoolkit.org
Fingerprintjs	86871	16,625	70,246	https://valve.github.io/fingerprintjs2/
Gravity Forms	109959	42,565	67,394	http://gravityforms.com
Slick	759805	693,556	66,249	https://kenwheeler.github.io/slick
Firestore	64037	851	63,186	https://firebase.com
Stripe	63812	6,400	57,412	http://stripe.com
Magento	118689	66,799	51,890	https://magento.com
WP-Statistics	50635		50,635	https://wp-statistics.com
OpenResty	258293	208,454	49,839	http://openresty.org
Angular	50196	1,230	48,966	https://angular.io
W3 Total Cache	220179	172,179	48,000	http://www.w3-edge.com/wordpress-plugins/w3-total-cache/
Knockout.js	44595	5	44,590	http://knockoutjs.com
Raphael	37091	147	36,944	https://dmitrybaranovskiy.github.io/raphael/
Raphael	37091	147	36,944	https://dmitrybaranovskiy.github.io/raphael/
Apache Tomcat	46424	11,498	34,926	http://tomcat.apache.org
mod_ssl	33928	22	33,906	http://modssl.org
script.aculo.us	77344	45,988	31,356	https://script.aculo.us
Flywheel	31288		31,288	https://getflywheel.com/
SoundManager	31075	2,117	28,958	http://www.schillmania.com/projects/soundmanager2
Sentry	421294	392,442	28,852	https://sentry.io/
TinyMCE	29228	747	28,481	http://tinymce.com
Highcharts	27438	279	27,159	https://www.highcharts.com
Snap.svg	26460	75	26,385	http://snapsvg.io/
Gatsby	27922	1,552	26,370	https://www.gatsbyjs.org/
Gatsby	27922	1,552	26,370	https://www.gatsbyjs.org/
D3	27147	1,184	25,963	http://d3js.org
MediaWiki	25549	246	25,303	https://www.mediawiki.org
MediaWiki	25549	246	25,303	https://www.mediawiki.org
CKEditor	47426	22,382	25,044	http://ckeditor.com
XRegExp	24799	53	24,746	http://xregexp.com

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