

SKRIPSI

PENGUKURAN APLIKASI USANG DI HTTPARCHIVE



Vinson Tandra

NPM: 2016730042

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2022**

DAFTAR ISI

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
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	5
2.1 BigQuery[1, 2]	5
2.1.1 Cloud Storage System	5
2.1.2 SQL (Structured Query Language) [3]	5
2.2 HTTP Archive [4]	10
2.3 Pengukuran Aplikasi Usang Pada Beberapa Website Populer Di Indonesia[5]	11
2.3.1 Research Method	11
2.3.2 Hasil Keseluruhan	12
2.4 Node.js [6]	14
2.4.1 Node Package Manager	14
2.4.2 NPM CLI	14
2.5 ReactJS[7]	15
2.5.1 JSX	15
2.5.2 Merender Elemen	16
2.5.3 Components and Props	16
2.5.4 Penanganan Event	17
2.5.5 Render Bersyarat	18
2.5.6 List dan Keys	19
2.5.7 Form	21
2.5.8 Hooks	22
2.5.9 NPX	24
2.6 ChartJS	24
2.7 Express.js	24
3 PERCOBAAN AWAL	25
3.1 Eksplorasi Teknologi	25
3.2 Dataset yang Digunakan pada HTTP Archive	33
3.3 Langkah-Langkah Query Yang Dilakukan	33
3.3.1 Mengumpulkan Daftar Website	34
3.3.2 Mencari Aplikasi Yang Digunakan Website	34
3.3.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai	35

3.3.4	Mencari Data Tentang Versi Aplikasi Yang Masih Didukung	37
3.3.5	Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai Sekarang Dengan Versi Aplikasi Yang Masih Didukung	37
3.4	Hasil Sample Data Dengan Beberapa Aplikasi	38
4	PENGGALIAN DATA	41
4.1	Langkah-Langkah Query Yang Dilakukan Dengan Data Yang Lebih Besar	41
4.1.1	Mengumpulkan List Website	41
4.1.2	Mencari Aplikasi Yang Digunakan Website	41
4.1.3	Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai	41
4.1.4	Mencari Data Tentang Versi Aplikasi Yang Masih Didukung	42
4.1.5	Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai Sekarang Dengan Versi Aplikasi Yang Masih Didukung	42
4.2	Hasil Sample Data	43
4.2.1	Apache dan Nginx	43
4.2.2	PHP dan Python	45
4.2.3	jQuery dan jQuery Migrate	46
5	PEMBANGUNAN PERANGKAT LUNAK	49
5.1	Implementasi Perangkat Lunak	49
5.1.1	BigQuery	49
5.1.2	React.js	53
5.1.3	Node.js	57
6	KESIMPULAN DAN SARAN	61
6.1	Kesimpulan	61
6.2	Saran	61
	DAFTAR REFERENSI	63
	A DATA APLIKASI	65
	B KODE PROGRAM	91

DAFTAR GAMBAR

2.1	<i>Algoritma untuk membandingkan versi yang dipakai dengan versi yang masih didukung</i>	12
3.1	Halaman Awal Google Cloud Project	25
3.2	Memilih <i>Project</i>	26
3.3	Membuat <i>Project</i>	26
3.4	Membuka <i>Console</i>	27
3.5	Data Terlihat Pada Dashboard	27
3.6	Google Cloud Project Page	28
3.7	Create atau Open Project	28
3.8	Membuka BigQuery	29
3.9	Membuat Dataset Baru	29
3.10	Membuat Tabel Baru	30
3.11	Memilih Save Result As BigQuery Table	30
3.12	Export Table	31
3.13	Dashboard Table	31
3.14	Create Table	32
3.15	Pilih Upload	32
3.16	Pilih Lokasi dan Format File	33
3.17	Data Sample Jumlah Aplikasi Dengan Versi yang Dipakai	39
4.1	Aplikasi Nginx	43
4.2	Aplikasi Apache	44
4.3	Aplikasi Apache dari [5]	44
4.4	Aplikasi Apache dari [5]	44
4.5	Aplikasi PHP	45
4.6	Aplikasi Python	45
4.7	Aplikasi PHP dari [5]	46
4.8	Aplikasi jQuery	46
4.9	Aplikasi jQuery Migrate	47
4.10	Aplikasi jQuery dari [5]	47

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 setiap *website* dari *desktop* dan *mobile* pada bulan Januari tahun 2016 sampai sekarang yang dapat dilihat menggunakan teknologi BigQuery. Dataset pada HTTP Archive didapatkan dari Chrome User Experience Report (CrUX). CrUX merupakan public dataset dari user experience data pada jutaan website. Pengujian pada dataset tersebut dilakukan dengan menggunakan Chrome pada desktop dan android (mobile).

HTTP Archive ¹ adalah sebuah *project* 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 project 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.

Pada [5] akan dilakukan penelitian tentang seberapa besar penggunaan aplikasi usang pada website di Indonesia. Data diambil dari website Alexa, dari 1.500 situs teratas menurut peringkat Alexa untuk pengunjung situs di Indonesia dan mengidentifikasi jenisnya aplikasi yang mereka pakai 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*.

¹https://github.com/HTTPArchive/httparchive.org/blob/main/docs/gettingstarted_bigquery.md

Pada skripsi ini, akan dibuat sebuah replikasi dari [5] tetapi dengan data yang lebih besar. Data dapat diambil dari HTTP Archive dengan melakukan *query* pada BigQuery. Pada penelitian ini akan dilakukan perhitungan pada jumlah aplikasi yang sudah diberi versi dan belum diberi versi. Versi aplikasi yang dipakai setiap *website* juga akan dibandingkan dengan versi aplikasi yang masih didukung berdasarkan *official website*-nya. Kemudian hasil tersebut akan ditampilkan dalam bentuk *bar chart*.

1.2 Rumusan Masalah

Berikut ini adalah rumusan masalah dari penelitian ini:

1. Bagaimana cara mendapatkan data dari HTTP Archive?
2. Bagaimana mereplikasi proyek [5] dengan menggunakan data yang lebih besar?
3. Berapa banyak *website* pada HTTP Archive yang menggunakan aplikasi yang masih didukung?

1.3 Tujuan

Berikut ini adalah tujuan dari penelitian ini:

1. Mendapatkan data dari HTTP Archive.
2. Mereplikasi proyek [5] dengan menggunakan data yang lebih besar.
3. Mencari jumlah *website* pada HTTP Archive yang menggunakan aplikasi yang masih didukung.

1.4 Batasan Masalah

Berikut ini adalah batasan masalah dari penelitian ini:

1. Data yang digunakan adalah data pada bulan Agustus tahun 2020.
2. Versi aplikasi berisi simbol merupakan non-konklusif.
3. Pada aplikasi ini hanya akan berfokus pada semantic version yang dimana digit major, minor, dan patch adalah angka.
4. Chart pada aplikasi ini tidak menunjukkan data yang UNVERSIONED dan NON-CONCLUSIVE karena datanya terlalu besar sehingga mengakibatkan data lain tidak terlihat.

1.5 Metodologi

Bagian-bagian pekerjaan skripsi ini adalah sebagai berikut:

1. Mempelajari teori HTTP Archive.
2. Mempelajari teori BigQuery.
3. Mempelajari bagaimana suatu *website* dikatakan usang.
4. Menganalisis beberapa *website* yang dikatakan usang.
5. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

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

- Bab 1 Pendahuluan

Berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.

- Bab 2 Dasar Teori

Berisi teori BigQuery, teori HTTP Archive, teori *library javascript*.

- Bab 3 Percobaan Awal

Berisi eksplorasi teknologi, penjelasan dataset yang digunakan, pengumpulan data secara terbatas.

- Bab 4 Penggalan Data

Berisi pengumpulan data yang besar, penjelasan tentang sample data pada aplikasi tertentu.

- Bab 5 Pembangunan Perangkat Lunak

Berisi perancangan perangkat lunak yang dibangun, masukan dan keluaran dari perangkat lunak, masalah yang dihadapi ketika implementasi.

- Bab 6 Kesimpulan dan Saran

Berisi kesimpulan dari awal hingga akhir penelitian dan saran untuk penelitian berikutnya.

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 di-hosting.

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 klausa ini. Klausa ini memiliki *syntax* seperti:

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

- WHERE

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

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

- GROUP BY

Fungsi dari klausa GROUP BY adalah untuk mengelompokkan baris berdasarkan nilai kolom yang sama. Klausa ini memiliki *syntax* seperti:

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

- ORDER BY

Fungsi dari klausa 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*). Klausa 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 kolom 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 kolom sebuah tabel. Fungsi MIN memiliki contoh *syntax* seperti:

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

- AVG()

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

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

- COUNT()

Fungsi ini bertujuan untuk mengembalikan jumlah baris dari kolom sebuah tabel. Fungsi COUNT memiliki contoh *syntax* seperti:

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

- SUM()

Fungsi ini bertujuan untuk mengembalikan jumlah baris dari kolom sebuah tabel. Fungsi SUM memiliki contoh *syntax* seperti:

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

- STRING_AGG()

Fungsi ini mengambil semua ekspresi dari baris dan menggabungkannya menjadi satu string. Nilai ekspresi secara implisit dikonversi ke tipe string dan kemudian digabungkan. Fungsi STRING_AGG memiliki contoh *syntax* seperti:

```
STRING_AGG ( expression, separator ) [ <order_clause> ]
```

Querying Multiple Tables

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

- LEFT OUTER JOIN

Kata kunci LEFT 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:

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

- RIGHT OUTER JOIN

Kata kunci RIGHT 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:

```
SELECT column_name(s)
FROM table1
RIGHT (OUTER) JOIN table2
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:

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

- INNER JOIN

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

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
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:

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

REPEAT()

Fungsi REPEAT() digunakan untuk mengulang string sebanyak jumlah yang dimasukkan. Fungsi REPEAT() akan mengembalikan string. Contoh dari *syntax* REPEAT() yaitu:

```
REPEAT(string, number)
```

COUNTIF()

Fungsi COUNTIF() mengembalikan jumlah ekspresi yang bernilai true. Karena ekspresi harus berupa boolean, sehingga bentuk COUNTIF(DISTINCT ...) tidak didukung. Contoh *syntax* COUNTIF() yaitu:

```
COUNTIF(
expression
)
[ OVER over_clause ]
```

UNNEST()

Operator UNNEST digunakan untuk mengkonversi ARRAY menjadi sekumpulan baris. UNNEST mengambil ARRAY dan mengembalikan tabel dengan satu baris untuk setiap elemen dalam ARRAY.

SPLIT()

Fungsi SPLIT() digunakan untuk membagi nilai menggunakan argumen pembatas. Untuk string memiliki default pembatas yaitu tanda koma, sedangkan bytes harus ditentukan sendiri pembatasnya. Fungsi SPLIT() akan mengembalikan nilai ARRAY. Contoh *syntax* fungsi SPLIT() yaitu:

```
SPLIT(value[, delimiter])
```

CASE

CASE akan mengevaluasi setiap klausa WHEN yang secara berurutan dan mengembalikan nilai dengan kondisi benar. Jika tidak terdapat klausa WHEN yang memiliki kondisi yang benar, maka nilai dari klausa ELSE yang akan dikembalikan. Contoh *syntax* CASE yaitu:

```
1 CASE
2 WHEN condition THEN result
3 [ ... ]
4 [ ELSE else_result ]
5 END
```

6 2.2 HTTP Archive [4]

7 HTTP Archive adalah sebuah *open-source project* yang melihat bagaimana *website* dibuat. HTTP
8 Archive menyediakan data-data historis untuk melihat bagaimana *website* berkembang. HTTP
9 Archive pertama sekali dimulai pada tahun 2010 oleh Steve Souders dan di-*maintain* oleh Pat
10 Meenan, Rick Viscomi, Paul Calvano, and Barry Pollard. Data url HTTP Archive didapatkan
11 menggunakan CrUX kemudian url dikirimkan ke WebPageTest setiap bulannya. CrUX adalah
12 sebuah dataset yang bersifat publik yang berisi data user experience dari jutaan website. Data
13 ini berasal dari data yang dikumpulkan dari pengguna yang telah memilih untuk mensinkronkan
14 *browsing history*. Data yang dihasilkan tersedia melalui:

- 15 1. PageSpeed Insights
- 16 2. Public Google BigQuery Project
- 17 3. CrUX Dashboard on Data Studio

18 Orang yang menggunakan HTTP Archive adalah anggota komunitas web, para sarjana, dan
19 pemimpin industri:

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

26 Di dalam HTTP Archive terdapat dataset yang dapat diambil menggunakan teknologi BigQuery.
27 Dataset tersebut adalah sebagai berikut:

- 28 1. almanac
29 Tabel ini tidak digunakan dalam pengerjaan skripsi ini.
- 30 2. blink_feature
31 Tabel ini tidak digunakan dalam pengerjaan skripsi ini.
- 32 3. core_web_vitals
33 Tabel ini tidak digunakan dalam pengerjaan skripsi ini.
- 34 4. latest
35 Tabel ini tidak digunakan dalam pengerjaan skripsi ini.
- 36 5. lighthouse
37 Tabel ini tidak digunakan dalam pengerjaan skripsi ini.
- 38 6. pages
39 Tabel ini tidak digunakan dalam pengerjaan skripsi ini.
- 40 7. requests
41 Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

8. response_bodies

Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

9. sample_data

Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

10. sample_data_2020

Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

11. scratchspace

Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

12. summary_pages

Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

13. summary_requests

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 pada desktop memiliki 61.203.638 baris dan pada mobile memiliki 67.452.994 baris.

15. urls

Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

16. wappalyzer

Tabel ini tidak digunakan dalam pengerjaan skripsi ini.

2.3 Pengukuran Aplikasi Usang Pada Beberapa *Website* Populer Di Indonesia[5]

Pada jurnal ini menjelaskan bahwa dalam bidang keamanan komputer, terdapat berbagai jenis metode dalam menyerang kerentanan pada sebuah sistem. Pengelola sistem yang sudah terkena dampak harus memperbarui sistemnya. Penelitian ini mengusulkan metode untuk melakukan pengukuran website tentang seberapa banyak penggunaan aplikasi yang tidak didukung. Pada penelitian ini dibatasi pada mendeteksi versi aplikasi yang digunakan.

2.3.1 *Research Method*

Terdapat empat langkah dalam melakukan penelitian ini, yaitu:

1. Memilih list *website* yang populer

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

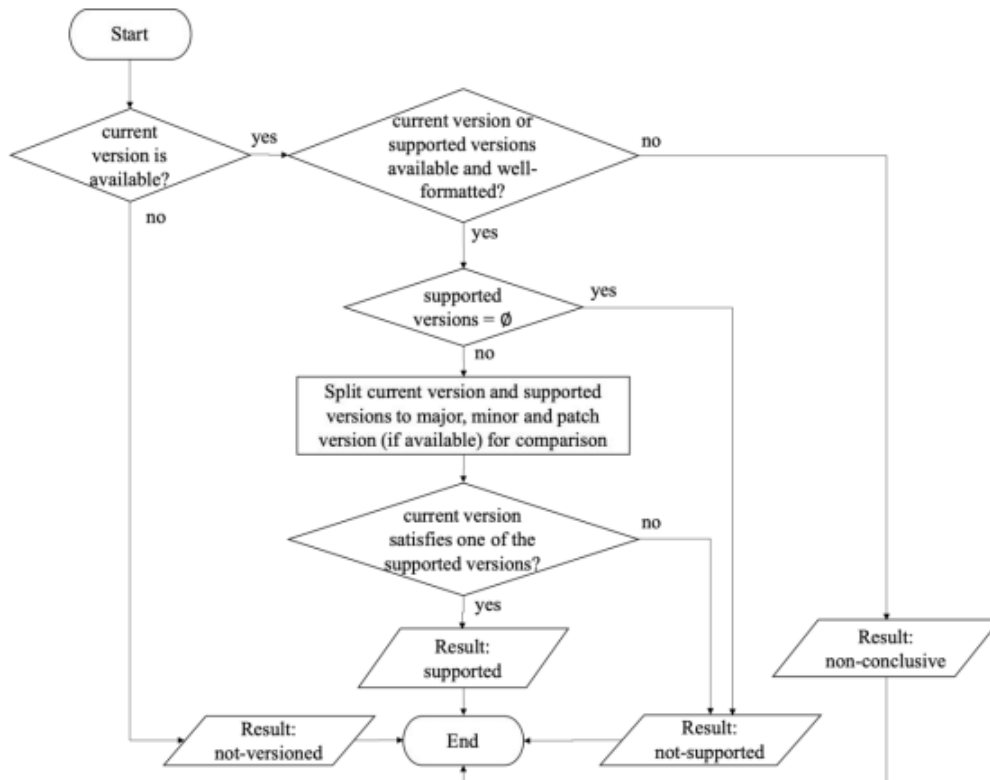
2. Mengidentifikasi aplikasi yang dipakai *website*

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: Algoritma untuk membandingkan versi yang dipakai dengan versi yang masih didukung

2.3.2 Hasil Keseluruhan

Pada jurnal[5], 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: Jumlah keseluruhan aplikasi berdasarkan hasil pengukuran

Tabel 2.2 adalah daftar sepuluh website yang paling populer. Dari daftar tersebut terlihat banyak sekali website yang menggunakan aplikasi yang tidak ada informasi versinya. Tetapi untuk yang ada informasi versinya, terdapat beberapa aplikasi yang sudah tidak didukung. Beberapa aplikasi yang sudah tidak didukung dari sepuluh website tersebut adalah Bootstrap, Font Awesome, jQuery, dan PHP. Pada tabel 2.3 terdapat 1,500 website yang dipisahkan setiap 150 website yang diurutkan berdasarkan rank website tersebut. Untuk setiap baris pada tabel tersebut akan dihitung website yang menggunakan n aplikasi yang sudah tidak didukung.

rank	domain name	not-versioned	non-conclusive	unsupported	supported
1	okezone.com	7	0	1	1
2	google.com	1	0	0	0
3	tribunnews.com	11	2	2	0
4	youtube.com	1	1	0	0
5	grid.id	11	1	2	1
6	detik.com	8	3	0	0
7	kompas.com	10	2	1	0
8	sindonews.com	4	1	1	0
9	tokopedia.com	5	0	0	0
10	liputan6.com	11	1	1	0

Tabel 2.2: Sepuluh Hasil Pengukuran

rank	r=0	r=1	r=2	r=3	r=4
1-150	56	58	26	9	1
151-300	52	55	29	12	2
301-450	59	43	32	10	6
451-600	56	48	22	21	3
601-750	59	58	22	10	1
751-900	68	44	25	8	5
901-1,050	65	42	30	10	3
1,051-1200	56	46	34	10	4
1201-1,350	50	57	31	11	1
1,350-1,500	62	46	29	11	2

Tabel 2.3: Jumlah aplikasi yang tidak didukung berdasarkan rank website

Pada tabel 2.4, terdapat beberapa aplikasi yang banyak digunakan. Beberapa aplikasi tersebut diambil dari 1.500 *website* teratas dan memfilter aplikasi yang versinya tidak dapat diidentifikasi di

1 salah satu dari 1.500 *website* teratas.

numsites	name	supported	unsupported	non-conclusive	not-versioned
1,011	jQuery	260	737	0	14
591	PHP	118	127	0	346
478	Nginx	5	116	0	357
430	Bootstrap	114	228	0	88
400	Font Awesome	70	157	13	160
346	WordPress	118	41	6	181
298	jQuery Migrate	0	0	267	31
237	Apache	79	10	2	146

Tabel 2.4: Aplikasi yang Banyak Digunakan

2 2.4 Node.js [6]

3 *Node.js* adalah sebuah *asynchronous event-driven JavaScript runtime*. *Node.js* memungkinkan
4 untuk menjalankan perintah JavaScript dari sisi server. Contoh *syntax* pada *Node.js* yaitu:

```

5
61  const http = require('http');
72
83  const hostname = '127.0.0.1';
94  const port = 3000;
105
116  const server = http.createServer((req, res) => {
127      res.statusCode = 200;
138      res.setHeader('Content-Type', 'text/plain');
149      res.end('Hello World');
150  });
161
172  server.listen(port, hostname, () => {
183      console.log('Server running at http://${hostname}:${port}/');
194  });

```

21 2.4.1 Node Package Manager

22 *Node Package Manager* (NPM) adalah *software registry* yang digunakan untuk meminjam atau
23 membagikan *software library* [8]. NPM terdiri dari tiga komponen penting, yaitu:

- 24 • NPM *website*.
- 25 • NPM CLI (*Command Line Interface*).
- 26 • NPM *Registry*.

27 Beberapa kegunaan dari menggunakan NPM adalah:

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

31 2.4.2 NPM CLI

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

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

2 2.5 ReactJS[7]

3 ReactJS merupakan *library* yang disediakan JavaScript untuk membuat *interface*. ReactJS dibuat
4 oleh Facebook.

5 2.5.1 JSX

6 JSX adalah sebuah ekstensi Javascript yang dapat mengikutsertakan HTML dalam Javascript. JSX
7 akan menghasilkan elemen React.

8 Menyatukan Ekspresi dalam JSX

9 Berikut ini adalah contoh penggunaan JSX:

```
10  
11 1 const name = 'Budi';  
12 2 const element = <h1>Halo, {name}</h1>;  
13 3  
14 4 ReactDOM.render(  
15 5   element,  
16 6   document.getElementById('root')  
17 7 );
```

19 Pada contoh diatas, variabel name akan dibungkus dengan menggunakan tanda kurung kurawal.
20 Semua ekspresi Javascript valid dalam tanda kurung kurawal di JSX.

21 Mengspesifikasikan Atribut Dengan JSX

22 Penulis dapat menggunakan tanda petik untuk mengspesifikasikan string literal sebagai atribut:

```
23  
24 1 const element = <a href="https://www.reactjs.org"> link </a>;
```

26 Penulis juga dapat menggunakan kurung kurawal untuk mengspesifikasikan ekspresi Javascript di
27 dalam atribut:

```
28  
29 1 const element = <img src={user.avatarUrl}></img>;
```

31 Mengspesifikasikan Elemen Anak dengan JSX

32 Jika tag bersifat kosong atau tidak memiliki elemen anak, penulis dapat menutup tag-nya secara
33 langsung dengan `</>`, seperti pada potongan kode dibawah ini.

```
34  
35 1 const element = <img src={user.avatarUrl} />;
```

37 Didalam tag JSX memungkinkan untuk memiliki elemen anak, yang dapat dilihat pada potongan
38 kode dibawah ini. Elemen anak dalam kode dibawah ini adalah elemen `<h1>` dan `<h2>`.

```
39  
40 1 const element = (  
41 2   <div>  
42 3     <h1>Halo!</h1>  
43 4     <h2>Senang melihatmu di sini.</h2>  
44 5   </div>  
45 6 );
```

2.5.2 Merender Elemen

Sebuah elemen menggambarkan hal yang ingin ditampilkan pada layar. Tidak seperti elemen DOM, elemen React merupakan objek biasa dan mudah dibuat. React DOM mengatur pembaruan DOM agar sesuai dengan elemen React.

Me-render Elemen Kedalam DOM

Aplikasi yang dibuat dengan React biasanya memiliki satu node DOM akar. Jika mengintegrasikan React ke dalam aplikasi yang sudah ada, penulis dapat memiliki node DOM akar yang terisolasi sebanyak yang Anda inginkan.

```
<div id="root"></div>
```

Potongan kode diatas disebut sebagai node DOM akar karena semua yang berada didalamnya akan diatur oleh React DOM.

```
const element = <h1>Hello, world</h1>;
ReactDOM.render(element, document.getElementById('root'));
```

Pada kode diatas, elemen dan root dimasukkan kedalam ReactDOM.render() agar elemen tersebut dapat dirender.

Memperbarui Elemen yang Di-render

Elemen React bersifat *immutable* sehingga setelah elemen dibuat, penulis tidak dapat mengubah nilai dari elemen atau atributnya. Satu-satunya cara untuk memperbarui antarmukanya adalah dengan membuat elemen baru atau menggunakan ReactDOM.render()

```
function tick() {
  const element = (
    <div>
      <h1>Hello, world!</h1>
      <h2>It is {new Date().toLocaleTimeString()}.</h2>
    </div>
  );
  ReactDOM.render(element, document.getElementById('root'));
}

setInterval(tick, 1000);
```

Pada kode diatas ReactDOM.render() membuat *callback* setiap detik.

2.5.3 Components and Props

Komponen mempermudah untuk memisahkan antarmuka menjadi bagian tersendiri dan dapat digunakan kembali. Secara konsep, komponen menyerupai fungsi Javascript. Komponen dapat menerima beberapa props (masukan) dan mengembalikan elemen React yang mendeskripsikan apa yang seharusnya tampil dilayar.

Fungsi dan Komponen Kelas

Cara yang paling sederhana untuk mendefinisikan sebuah komponen adalah dengan menuliskan sebuah fungsi Javascript

```
function Welcome(props) {
  return <h1>Halo, {props.name}</h1>;
}
```

- 1 Fungsi diatas adalah contoh komponen React yang sah karena menerima sebuah props tunggal atau
2 argumen objek dengan data dan kembalian sebuah elemen React.

3 Merender Sebuah Komponen

- 4 Didalam React Elemen dapat mewakili komponen yang didefinisikan oleh penulis. Seperti pada
5 kode dibawah.

```
6 1 const element = <Welcome name="Sara" />;
```

- 9 Ketika React melihat sebuah elemen mewakili sebuah komponen yang dibuat oleh penulis, komponen
10 akan mengoper atribut JSX ke dalam komponen ini sebagai objek tunggal. Objek ini disebut sebagai
11 props. Kode dibawah ini akan menghasilkan "Halo Sara" pada halaman.

```
12 13 1 function Welcome(props) {
14 2   return <h1>Halo, {props.name}</h1>;
15 3 }
16 4
17 5 const element = <Welcome name="Sara" />;
18 6 ReactDOM.render(
19 7   element,
20 8   document.getElementById('root')
21 9 );
```

23 Menyusun Komponen

- 24 Komponen dapat merujuk ke komponen lain pada keluarannya. Hal ini memungkinkan kita untuk
25 membuat abstraksi dari komponen yang sama untuk tingkat detail. Seperti membuat sebuah tombol,
26 sebuah form, sebuah tampilan, sebuah dialog. Dalam aplikasi React, semua itu dinyatakan dalam
27 bentuk komponen. Sebagai contoh penulis dapat membuat sebuah komponen App yang mencetak
28 Welcome berkali-kali.

```
29 30 1 function Welcome(props) {
31 2   return <h1>Halo, {props.name}</h1>;
32 3 }
33 4
34 5 function App() {
35 6   return (
36 7     <div>
37 8       <Welcome name="Sara" />
38 9       <Welcome name="Cahal" />
39 0       <Welcome name="Edite" />
40 1     </div>
41 2   );
42 3 }
43 4 }
```

44 2.5.4 Penanganan Event

- 45 Menangani events dengan elemen React sangat mirip seperti menangani sebuah events pada elemen
46 DOM. Ada beberapa perbedaan sintaks:

- 47 • Events pada React biasanya ditulis dalam bentuk camelCase, bukan lowercase.
- 48 • Dengan JSX Anda dapat mengoper function sebagai event handler, bukan sebagai string.

- 49 Berikut ini adalah contoh sintaks pada HTML:

```
50 51 1 <button onclick="activateLasers()">
52 2   Aktivasi Laser
53 3 </button>
```

- 55 Sintaks HTML memiliki sedikit perbedaan dengan sintaks pada React. Berikut ini adalah contoh
56 sintaks pada React:

```

1
2 1 <button onClick={activateLasers}>
3 2 Aktivasi Laser
4 3 </button>

```

6 Perbedaan lainnya adalah penulis tidak dapat mengembalikan nilai false untuk mencegah behavior-
 7 bawaan React. Penulis harus menggunakan preventDefault. Sebagai contoh, pada HTML untuk
 8 mencegah agar link bawaan membuka halaman baru, penulis dapat menulis seperti ini:

```

9
10 1 <a href="#" onClick="console.log('The link was clicked.');" return false">
11 2 Click me
12 3 </a>

```

14 Sedangkan pada React, contoh tersebut dapat ditulis sebagai berikut:

```

15
16 1 function ActionLink() {
17 2   function handleClick(e) {
18 3     e.preventDefault();
19 4     console.log('Tautan diklik.');

```

29 Pada kode diatas e merupakan event tiruan. React mendefinisikan event tiruan ini berdasarkan
 30 W3C spec², jadi Anda tidak perlu khawatir akan kesesuaian antar lintas browser. Event dalam
 31 React tidak bekerja secara sama dengan event native dari __browser.

32 Mengoper Argumen Kedalam Penanganan Event

33 Di dalam perulangan, umumnya Anda ingin mengoper sebuah parameter ekstra kedalam penanganan
 34 event. Sebagai contoh, jika id sama dengan ID baris, maka salah satu dari kedua contoh berikut
 35 dapat dijalankan:

```

36
37 1 <button onClick={() => this.deleteRow(id, e)}>Delete Row</button>
38 2 <button onClick={this.deleteRow.bind(this, id)}>Delete Row</button>

```

40 Dua baris di atas memiliki arti yang sama, masing-masing menggunakan arrow functions dan
 41 Function.prototype.bind. Arrow function adalah sebuah cara alternatif untuk mendefinisikan fungsi
 42 dari fungsi tradisional. Cara ini bersifat terbatas dan tidak dapat digunakan dalam setiap kondisi.
 43 Sedangkan metode bind() adalah sebuah metode yang membuat sebuah fungsi yang ketika dipanggil,
 44 kata kunci this akan berubah menjadi nilai yang diberikan.

45 2.5.5 Render Bersyarat

46 Render bersyarat pada React memiliki fungsi yang sama dengan operator bersyarat pada Javascript.
 47 Pada Javascript operator if atau operator bersyarat digunakan untuk merepresentasikan elemen
 48 pada state tertentu, kemudian React akan memperbarui UI pada state tersebut.

49 Contoh nya seperti dua komponen berikut ini:

```

50
51 1 function UserGreeting(props) {
52 2   return <h1>Welcome back!</h1>;
53 3 }
54 4
55 5 function GuestGreeting(props) {

```

²<https://www.w3.org/TR/DOM-Level-3-Events/>


```

16 |         return <h1>Please sign up.</h1>;
37 |     }

```

4 Komponen diatas akan digunakan untuk melakukan Greetings berdasarkan pada apakah user sudah
5 melakukan login:

```

6 |
7 |     function Greeting(props) {
8 |         const isLoggedIn = props.isLoggedIn;
9 |         if (isLoggedIn) {
10 |             return <UserGreeting />;
11 |         }
12 |         return <GuestGreeting />;
13 |     }
14 |
15 |     ReactDOM.render(
16 |         // Try changing to isLoggedIn={true}:
17 |         <Greeting isLoggedIn={false} />,
18 |         document.getElementById('root')
19 |     );

```

21 2.5.6 List dan Keys

22 Javascript dapat menggunakan fungsi map() untuk mengambil array numbers dan menggandakan
23 angkanya. Map() akan mengembalikan nilai dalam bentuk array yang baru kemudian akan disimpan
24 dalam sebuah variabel doubled. Berikut ini adalah contoh kodenya:

```

25 |
26 |     const numbers = [1, 2, 3, 4, 5];
27 |     const doubled = numbers.map((number) => number * 2);
28 |     console.log(doubled);

```

30 Kode diatas akan mengembalikan nilai [2, 4, 6, 8, 10] ke dalam konsol. Pada React, mengubah
31 array ke dalam list elemen kurang lebih sama.

32 Me-render Banyak Komponen

33 Penulis dapat membangun koleksi dari beberapa elemen dan menyertakannya dalam JSX menggu-
34 nakan tanda kurung kurawal.

```

35 |
36 |     const numbers = [1, 2, 3, 4, 5];
37 |     const listItems = numbers.map((number) =>
38 |         <li>{number}</li>
39 |     );

```

41 Pada kode diatas, akan dilakukan perulangan pada numbers yang berisi array dengan menggunakan
42 fungsi map(). Hasil yang akan dikeluarkan adalah sebuah elemen untuk setiap item yang
43 kemudian akan dimasukkan kedalam variabel listItem.

```

44 |
45 |     ReactDOM.render(
46 |         <ul>{listItems}</ul>,
47 |         document.getElementById('root')
48 |     );

```

50 Kemudian dengan menggunakan kode diatas, array listItem tersebut dapat dimasukkan kedalam
51 elemen dan akan me-render-nya kedalam DOM.

52 Daftar Komponen Dasar

53 Penulis dapat me-refaktor contoh sebelumnya ke dalam sebuah komponen yang menerima array
54 numbers dan mengeluarkan sebuah list elemen yang tidak berurutan.

```

1
21 function NumberList(props) {
32   const numbers = props.numbers;
43   const listItems = numbers.map((number) =>
54     <li>{number}</li>
65   );
76   return (
87     <ul>{listItems}</ul>
98   );
109 }
110
121 const numbers = [1, 2, 3, 4, 5];
132 ReactDOM.render(
143   <NumberList numbers={numbers} />,
154   document.getElementById('root')
165 );
17

```

18 Ketika penulis menjalankan kode ini, penulis akan mendapatkan peringatan bahwa key harus
 19 disediakan untuk item di dalam list. Sebuah "key" adalah atribut string spesial yang perlu
 20 disertakan dalam pembuatan list elemen. Penulis harus sertakan key ke dalam list item pada
 21 `numbers.map()` dan memperbaiki masalah key yang hilang.

```

22
231 function NumberList(props) {
242   const numbers = props.numbers;
253   const listItems = numbers.map((number) =>
264     <li key={number.toString()}>
275       {number}
286     </li>
297   );
308   return (
319     <ul>{listItems}</ul>
320   );
331 }
342
353 const numbers = [1, 2, 3, 4, 5];
364 ReactDOM.render(
375   <NumberList numbers={numbers} />,
386   document.getElementById('root')
397 );
40

```

41 Key

42 Key membantu React untuk mengidentifikasi item mana yang telah diubah, ditambahkan, atau
 43 dihilangkan. Key harus diberikan di dalam elemen yang terdapat di dalam sebuah array untuk
 44 memberikan elemen tersebut identitas yang stabil:

```

45
461 const numbers = [1, 2, 3, 4, 5];
472 const listItems = numbers.map((number) =>
483   <li key={number.toString()}>
494     {number}
505   </li>
516 );
52

```

53 Cara terbaik untuk menentukan key yang akan digunakan adalah menggunakan string unik untuk
 54 mengidentifikasikan item. Biasanya menggunakan ID dari data sebagai key:

```

55
561 const todoItems = todos.map((todo) =>
572   <li key={todo.id}>
583     {todo.text}
594   </li>
605 );
61

```

62 Ketika tidak memiliki ID yang stabil untuk me-render item, penulis dapat menggunakan indeks
 63 dari item sebagai key untuk pilihan terakhir:

```

64
651 const todoItems = todos.map((todo, index) =>
662   <li key={index}>
673     {todo.text}
684   </li>
695 );
70

```

2.5.7 Form

Elemen form HTML bekerja sedikit berbeda dari elemen DOM lainnya di React, karena elemen form secara natural menyimpan beberapa state internal. Sebagai contoh, form ini pada HTML biasa menerima nama tunggal:

```
<form>
<label>
  Name:
  <input type="text" name="name" />
</label>
  <input type="submit" value="Submit" />
</form>
```

Form ini memiliki perilaku dasar dari form HTML biasa yakni menuju ke laman baru ketika user mengirim form tersebut. Jika Anda menginginkan perilaku seperti ini di React, ini sebenarnya dapat bekerja. Namun di banyak kasus, akan lebih mudah untuk memiliki sebuah fungsi JavaScript yang menangani sebuah submisi dari sebuah form dan memiliki akses terhadap data yang dimasukkan pengguna ke dalam form. Cara standar untuk mencapai hal ini adalah dengan teknik yang disebut "controlled component".

Controlled Component

Elemen form pada HTML seperti `<input>`, `<select>`, dan `<textarea>` biasanya menyimpan state mereka sendiri dan memperbaruinya berdasarkan masukan dari pengguna. Pada React, state yang dapat berubah akan disimpan dalam sebuah prop atau komponen dan hanya dapat diubah dengan menggunakan `setState()`. komponen React yang me-render sebuah form juga mengontrol hal yang terjadi dalam form tersebut pada masukan pengguna selanjutnya. Sebuah elemen masukan form yang nilainya dikontrol oleh React melalui cara seperti ini disebut sebagai "controlled component".

```
class NameForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = {value: ''};

    this.handleChange = this.handleChange.bind(this);
    this.handleSubmit = this.handleSubmit.bind(this);
  }

  handleChange(event) {
    this.setState({value: event.target.value});
  }

  handleSubmit(event) {
    alert('A name was submitted: ' + this.state.value);
    event.preventDefault();
  }

  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <label>
          Name:
          <input type="text" value={this.state.value} onChange={this.handleChange} />
        </label>
        <input type="submit" value="Submit" />
      </form>
    );
  }
}
```

Dengan sebuah controlled component, nilai input akan selalu didorong oleh state di React. Meskipun menulis kode yang lebih banyak, tetapi juga bisa mengoper nilai ini ke elemen antarmuka lain, atau me-reset nilai tersebut dari event handler lain.

1 Mengendalikan Nilai Kosong

- 2 Menentukan nilai prop pada controlled component mencegah pengguna mengubah masukan. Jika
- 3 sudah menetapkan nilai value tetapi masukan masih dapat diubah, mungkin nilai value yang
- 4 ditetapkan bersifat undefined atau null.

5 2.5.8 Hooks

- 6 Hooks merupakan sebuah fungsi yang memungkinkan penulis untuk mengaitkan *state* dan fitur-fitur
- 7 *lifecycle* react dari *function component*.

8 useState

- 9 useState adalah sebuah hook yang memungkinkan penulis untuk menambahkan *react state* ke
- 10 *function components*. Cara untuk menggunakan useState adalah dengan meng-*import* useState
- 11 hook dari react dengan menggunakan sintak:

```
12 import React, { useState } from 'react';
13
14 function Example() {
15   // ...
16 }
```

- 17 Berikut ini adalah beberapa informasi mengenai useState:

- 18 • Mendeklarasi Sebuah State Variabel

- 19 Di dalam sebuah kelas, dalam menginisialisasikan variabel dalam state biasanya menggunakan
- 20 `this.state`.

```
21 1 class Example extends React.Component {
22 2   constructor(props) {
23 3     super(props);
24 4     this.state = {
25 5       count: 0
26 6     };
27 7   }
28 }
```

- 30 Di dalam sebuah component function tidak terdapat `this`, sehingga tidak dapat menetapkan
- 31 variabel atau membaca `this.state`. Melainkan menggunakan useState hook didalam komponen.

```
32 1 import React, { useState } from 'react';
33 2
34 3 function Example() {
35 4   // Declare a new state variable, which we'll call "count"
36 5   const [count, setCount] = useState(0);
37 }
```

- 39 • Membaca State

- 40 Ketika ingin menampilkan count saat ini dalam kelas, dapat menggunakan kode `this.state.count`
- 41 seperti:

```
42 <p>You clicked {this.state.count} times</p>
```

- 43 Sedangkan didalam *function*, dapat menggunakan count secara langsung:

```
44 <p>You clicked {count} times</p>
```

- Meng-*update* State

Ketika ingin meng-update count, dapat menggunakan kode `this.setState()` seperti:

```
<button onClick={() => this.setState({ count: this.state.count + 1 })}>
Click me
</button>
```

Sedangkan dalam sebuah fungsi, terdapat variabel `setCount` dan `count` sehingga tidak membutuhkan `this`:

```
<button onClick={() => setCount(count + 1)}>
Click me
</button>
```

UseEffect

Dalam react terdapat istilah seperti side-effects. Disebut side-effects karena dapat mempengaruhi komponen lain dan tidak dapat dilakukan pada saat proses render. `UseEffect` menambahkan kemampuan untuk melakukan side-effects dari sebuah function component. Hook ini memiliki fungsi yang sama dengan `componentDidMount`, `componentDidUpdate`, dan `componentWillUnmount` pada kelas react, tetapi sudah disatukan menjadi satu API.

```
import React, { useState, useEffect } from 'react';

function Example() {
  const [count, setCount] = useState(0);

  // Sama seperti componentDidMount dan componentDidUpdate:
  useEffect(() => {
    // Memperbarui judul dokumen menggunakan API browser
    document.title = `Anda klik sebanyak ${count} kali`;
  });

  return (
    <div>
      <p>Anda klik sebanyak {count} kali</p>
      <button onClick={() => setCount(count + 1)}>
        Klik saya
      </button>
    </div>
  );
}
```

Pada kode diatas, React diperintah untuk menjalankan fungsi efek setelah membersihkan perubahan dari DOM. Efek dideklarasikan didalam komponen untuk mendapatkan akses pada props dan state dari komponen tersebut.

Rules of Hooks

Hooks adalah fungsi JavaScript, tetapi terdapat dua aturan yang perlu diikuti keteika ingin menggunakannya:

1. Hanya Panggil Hooks di Tingkat Atas

Jangan memanggil hooks dari dalam loops, condition, atau nested functions. Dengan mengikuti aturan ini, menyebabkan react ddaapt menyimpan state dari hooks dengan benar diantara banyak panggilan `useState` dan `useEffect`.

2. Hanya Panggil Hooks dari Fungsi React

Jangan memanggil hooks dari fungsi-fungsi JavaScript biasa. Dengan mengikuti aturan ini, dapat dengan yakin bahwa semua logika stateful didalam sebuah komponen terlihat jelas dari kodenya.

2.5.9 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. Pembuatan *project react* dapat dilakukan dengan menggunakan sintaks:

```
npx create-react-app my-app
cd my-app
npm start
```

2.6 ChartJS

ChartJS adalah sebuah open-source library JavaScript yang digunakan untuk visualisasi data. Tipe-tipe chart yang didukung oleh ChartJS adalah bar, line, area, pie, bubble, radar, polar, mix, dan scatter. Pada skripsi ini tipe chart yang digunakan adalah bar chart. Untuk menginstall library ChartJS dapat dilakukan dengan menggunakan sintaks:

```
npm i react-chartjs-2 chart.js
```

2.7 Express.js

Express.js adalah sebuah framework sebuah aplikasi web untuk Node.js. Untuk menginstall Express.js dapat dilakukan dengan melakukan sintaks:

```
npm install express --save
```

Ketika menjalankan sintaks diatas, maka secara otomatis menambahkan library express.js yang akan disimpan pada folder package.json.

Routing

Routing adalah sebuah proses dari perangkat lunak dalam merespon beberapa endpoint. Dalam mendefinisikan routing menggunakan express.js terdapat struktur perintah sebagai berikut:

```
app.METHOD(PATH, HANDLER)
```

Contoh perintah dalam membuat route pada Express.js adalah:

```
app.get('/', (req, res) => {
  res.send('hello world')
})
```

BAB 3

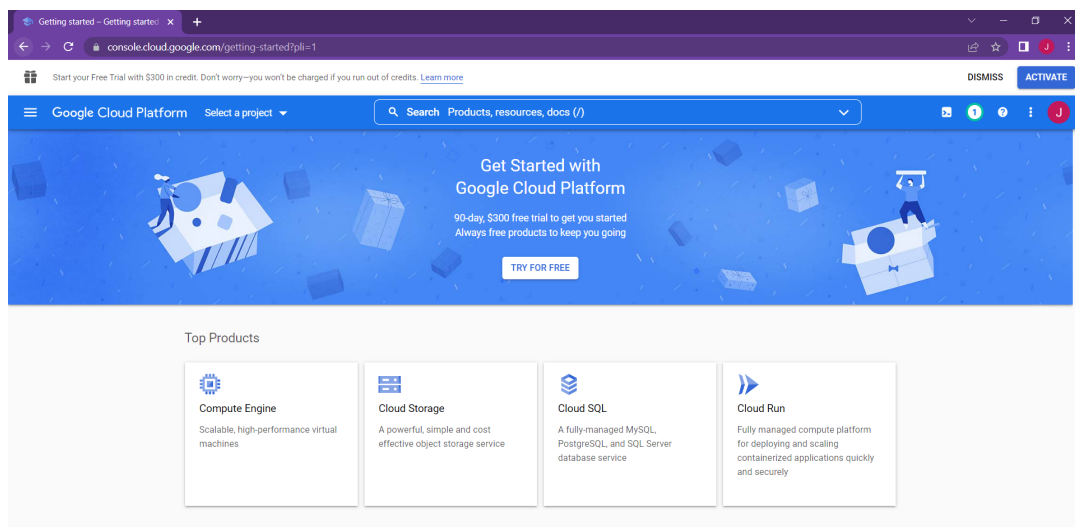
PERCOBAAN AWAL

Pada bab ini akan dijelaskan analisis masalah penelitian ini. Analisis meliputi Eksplorasi Teknologi, Dataset Pada HTTP Archive, Langkah-Langkah Query Yang Dilakukan, dan Hasil Sample Data Dengan Beberapa Aplikasi.

3.1 Eksplorasi Teknologi

Dalam pengerjaan skripsi ini akan menggunakan teknologi bernama BigQuery. Dataset pada HTTP Archive didapatkan dengan menggunakan teknologi BigQuery. Berikut ini adalah langkah untuk mendapatkan dataset tersebut:

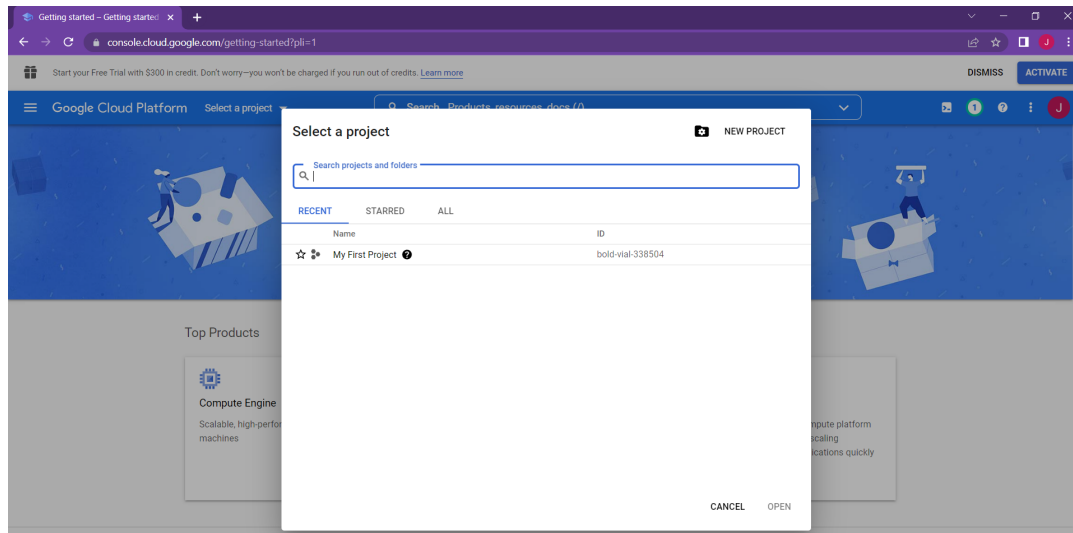
1. Membuka Google Cloud Project Page¹ dan masuk dengan menggunakan Google account.



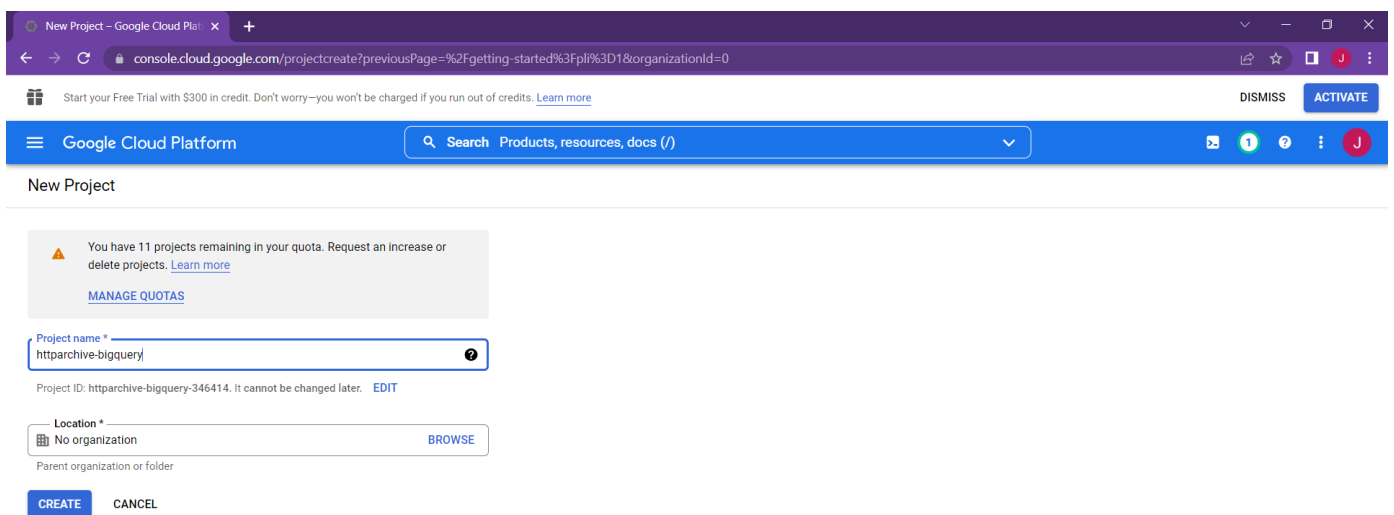
Gambar 3.1: Halaman Awal Google Cloud Project

2. Memilih project kemudian *"New Project"*

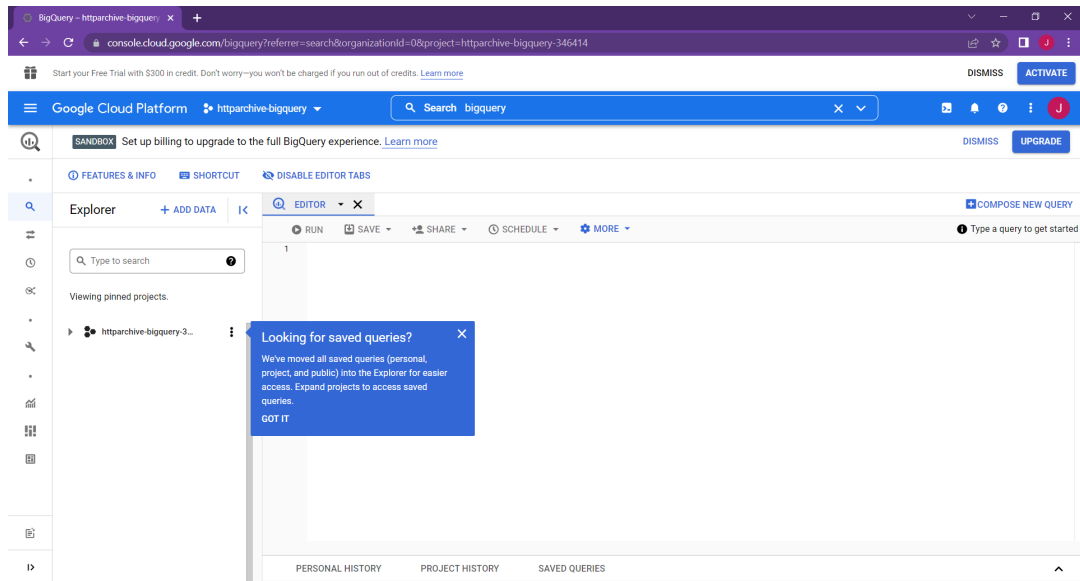
¹<https://console.cloud.google.com/getting-started>

Gambar 3.2: Memilih *Project*

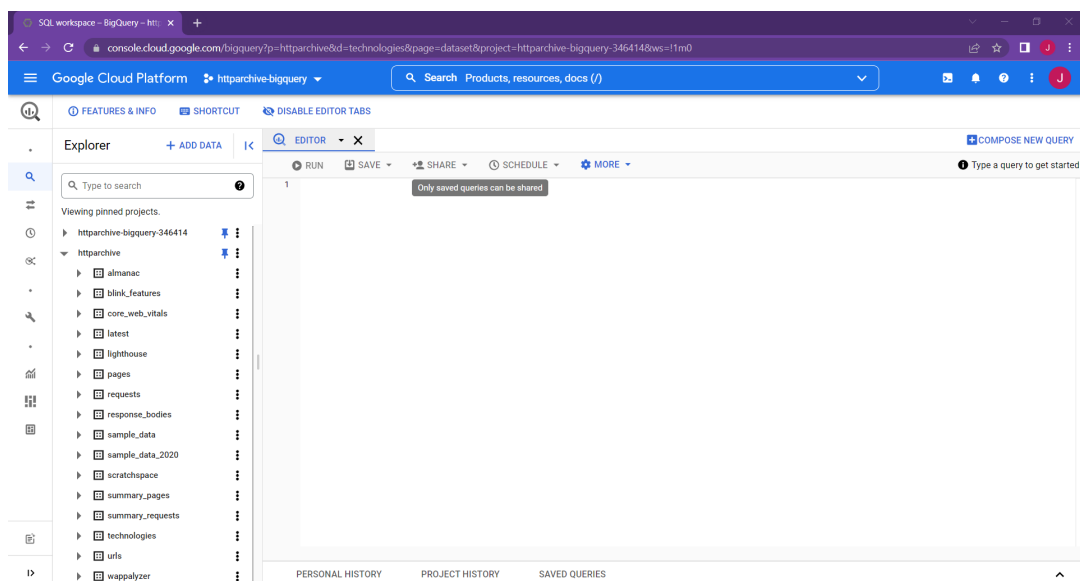
- 1 3. Masukkan nama *project* kemudian tekan tombol *create*

Gambar 3.3: Membuat *Project*

- 2 4. Buka BigQuery *console*

Gambar 3.4: Membuka *Console*

- 1 5. Untuk menambahkan tabel HTTP Archive pada *project* didapatkan dari link ²
- 2 6. Data HTTP Archive dapat dilihat pada dashboard BigQuery.



Gambar 3.5: Data Terlihat Pada Dashboard

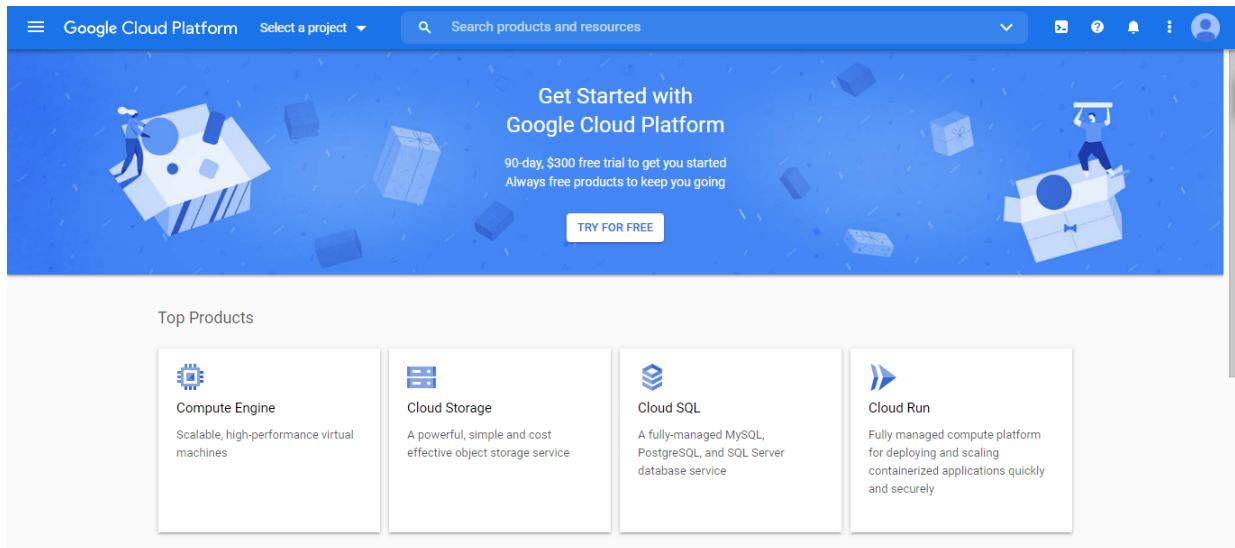
- 3 Di dalam BigQuery, terdapat salah satu fitur yang akan digunakan yaitu membuat dataset baru.
- 4 Dataset bisa saja diambil dari public dataset maupun membuat sendiri dataset tersebut. Dataset
- 5 berisi tabel-tabel yang akan dianalisis. Tabel-tabel tersebut dapat dibuat secara manual maupun
- 6 di-*upload*.

- 7 Berikut ini langkah-langkah dalam pembuatan dataset dan tabel:

- 8 1. Membuka Google Cloud Project Page³. Halaman yang akan ditampilkan dapat dilihat pada
- 9 gambar 3.6

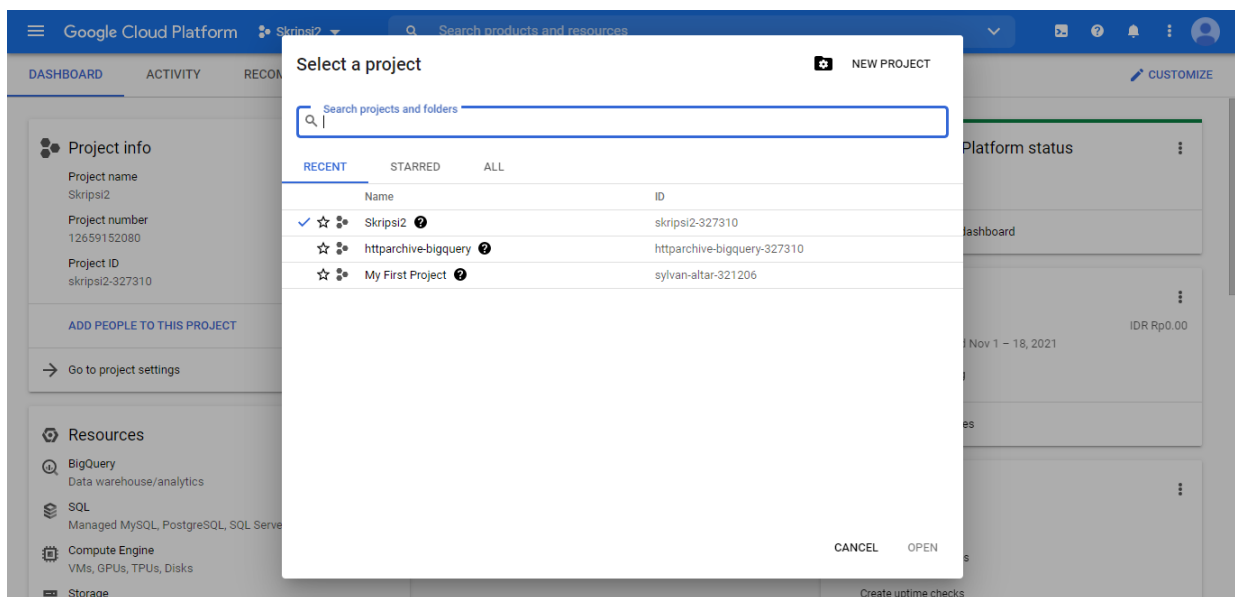
²<https://console.cloud.google.com/bigquery?p=httparchive>

³<https://console.cloud.google.com/getting-started>



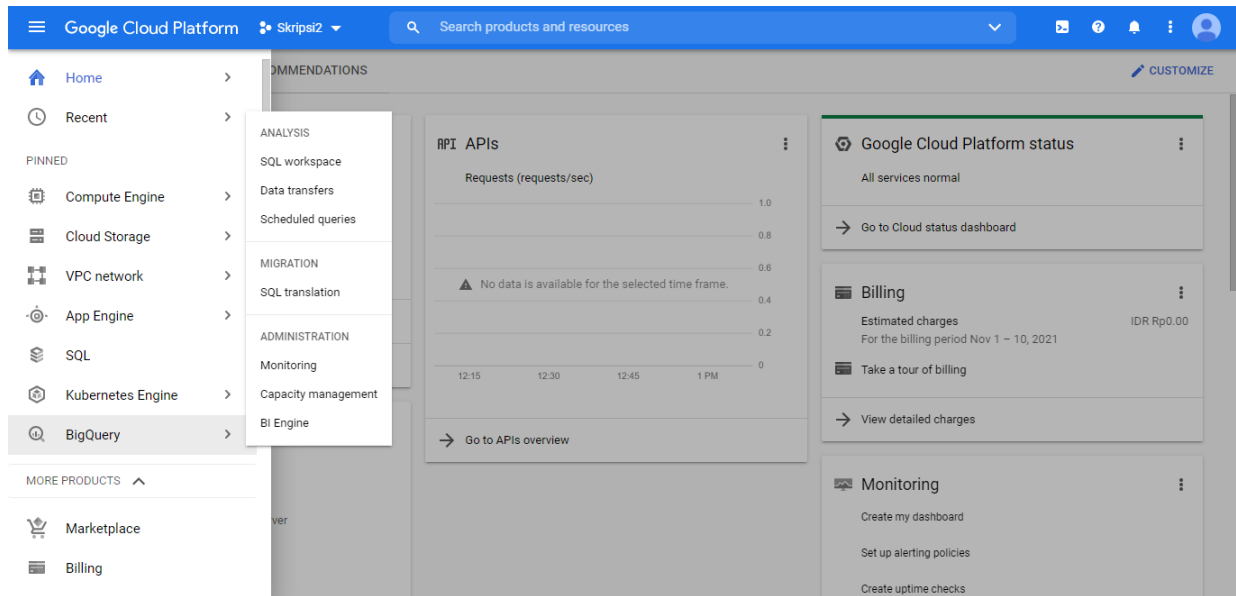
Gambar 3.6: Google Cloud Project Page

- 1 2. Membuat atau memilih *project* yang akan dikerjakan. Halaman yang akan ditampilkan dapat
- 2 dilihat pada gambar 3.7



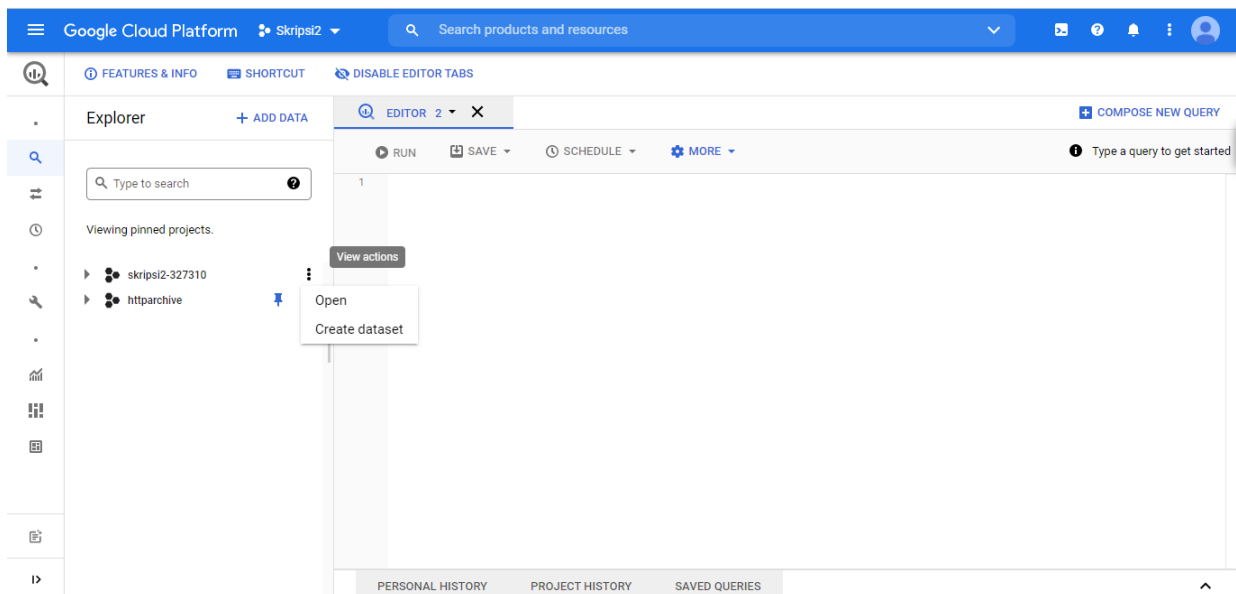
Gambar 3.7: Create atau Open Project

- 3 3. Membuka *console* kemudian memilih BigQuery. Halaman yang akan ditampilkan dapat dilihat
- 4 pada gambar 3.8



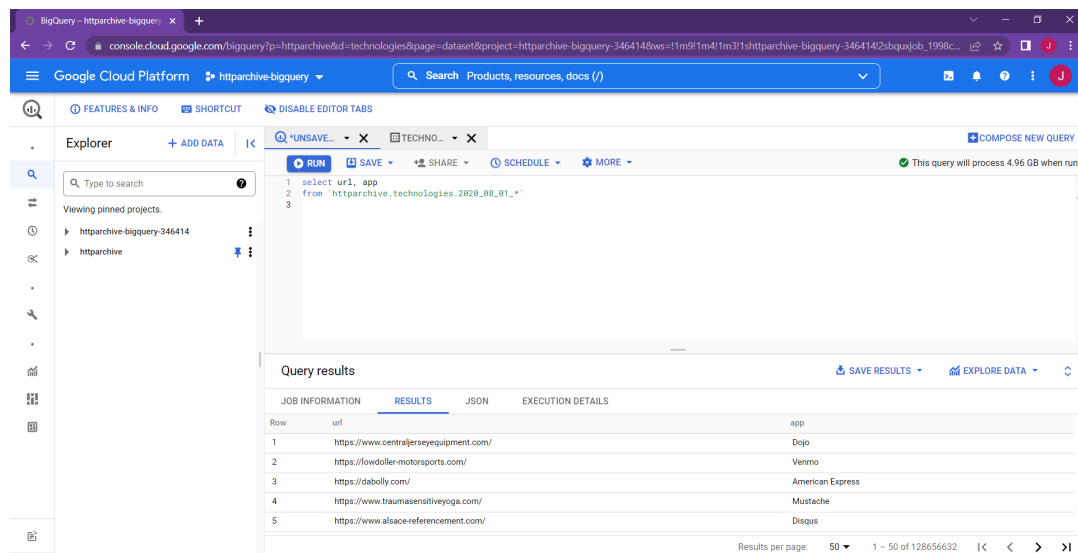
Gambar 3.8: Membuka BigQuery

- 1 4. Pada tab explorer terdapat project kemudian pengguna harus menekan tombol titik tiga dan
- 2 pilih *create* dataset. Halaman yang akan ditampilkan dapat dilihat pada gambar 3.9



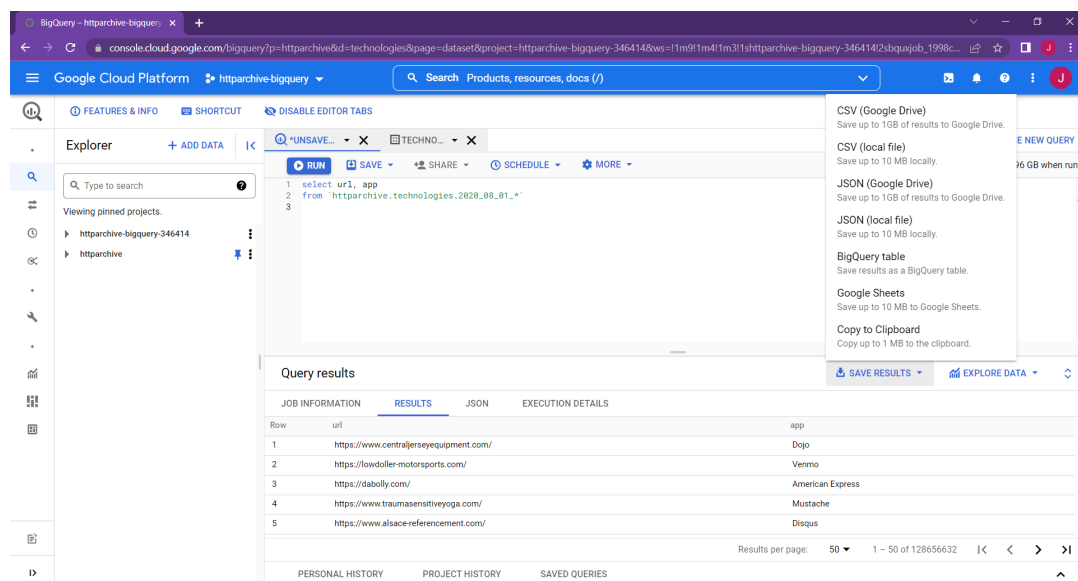
Gambar 3.9: Membuat Dataset Baru

- 3 Skripsi ini akan membuat tabel baru agar tidak melakukan query yang sama berulang. Tabel
- 4 dapat dibuat dengan cara:
- 5 1. Membuat query yang akan disimpan dalam tabel



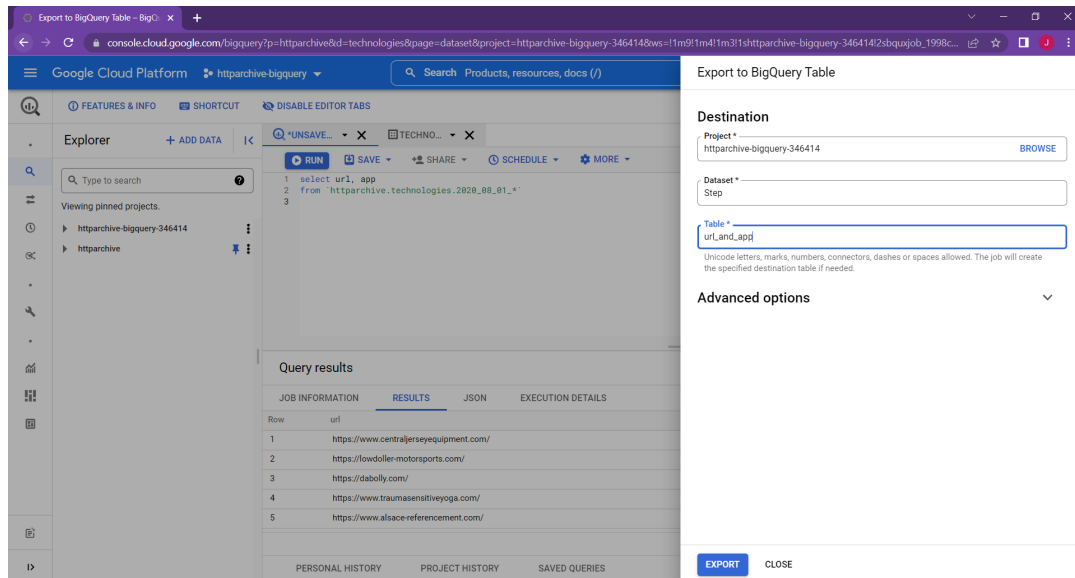
Gambar 3.10: Membuat Tabel Baru

- 1
2. Memilih save result as BigQuery Table



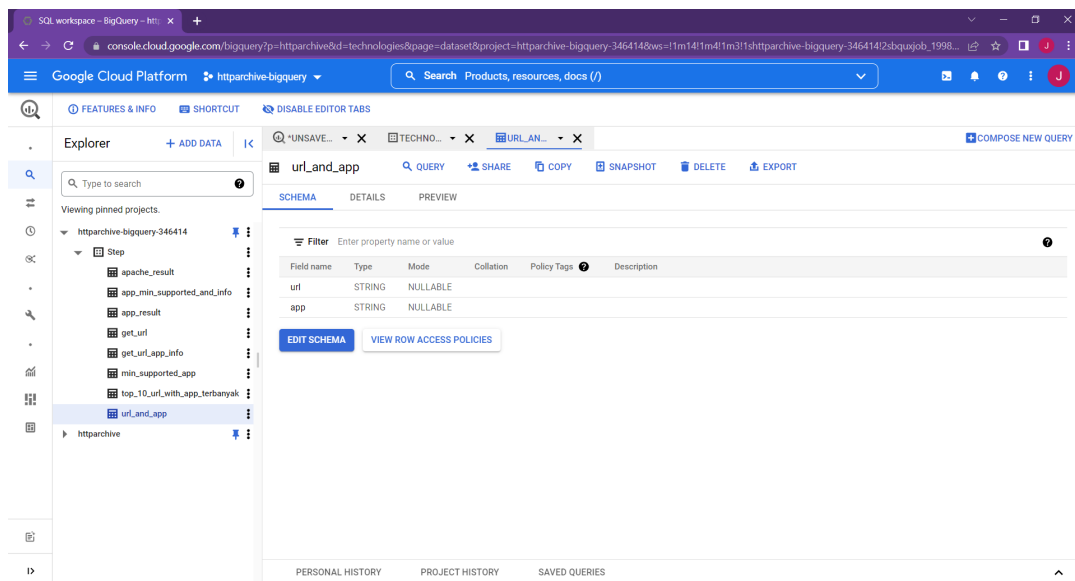
Gambar 3.11: Memilih Save Result As BigQuery Table

- 2
3. Memilih lokasi atau dataset dan nama tabel untuk disimpan kemudian export



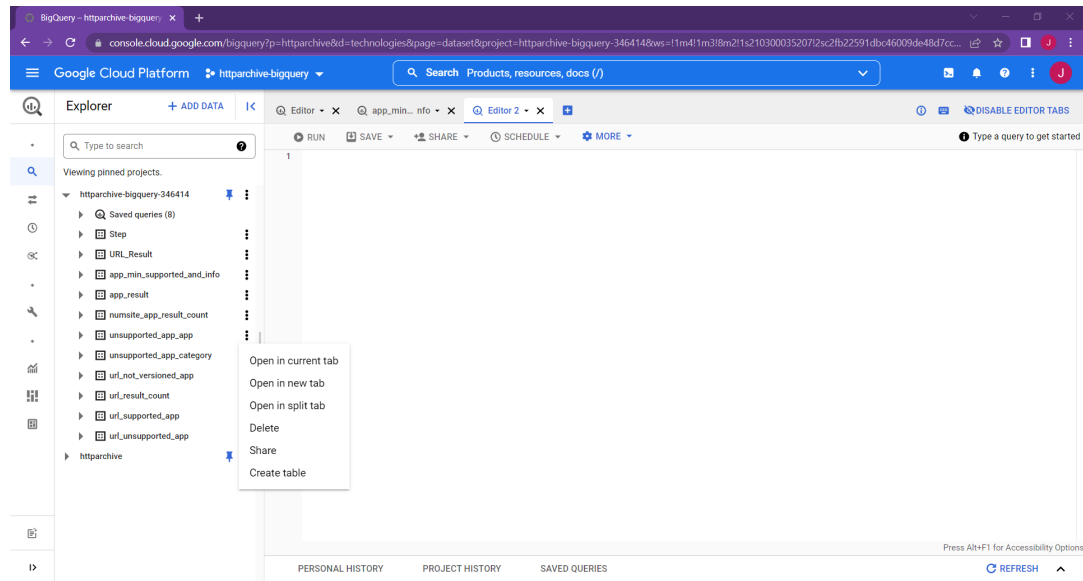
Gambar 3.12: Export Table

- 1 4. Lokasi tabel dapat dilihat pada dashboard



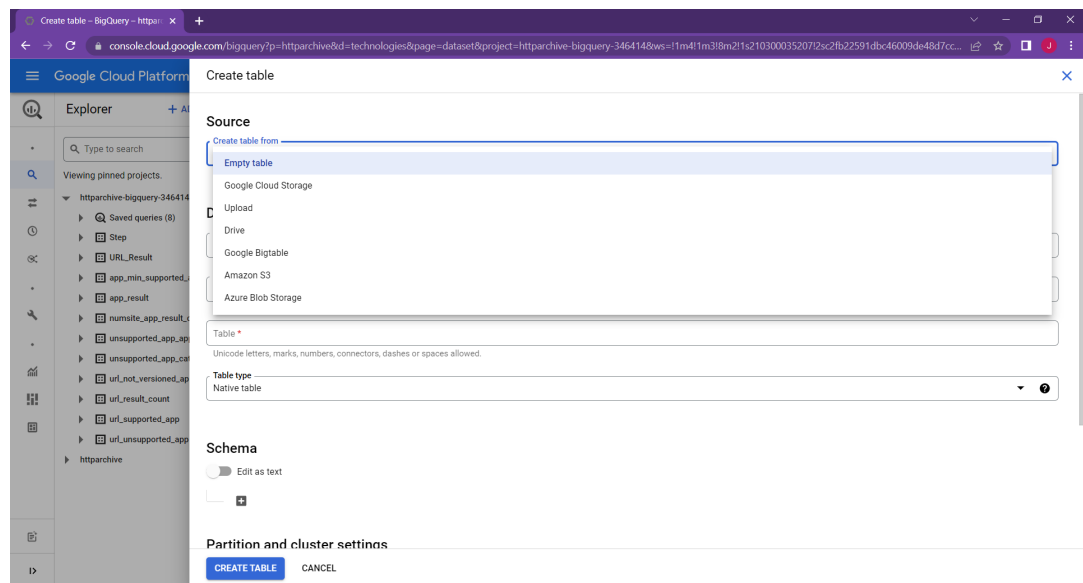
Gambar 3.13: Dashboard Table

- 2 Selain itu, table dapat dibuat dengan mengupload table dari CSV atau JSON. Tabel dapat
 3 dibuat dengan cara:
 4 1. Pilih titik tiga pada salah satu dataset dan pilih *create table* yang dapat dilihat pada gambar??.



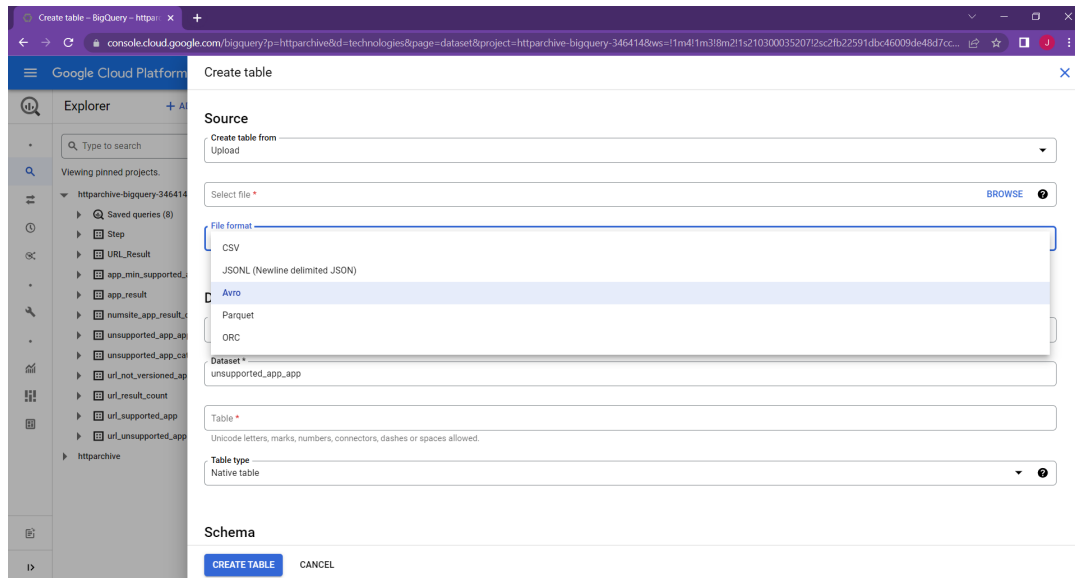
Gambar 3.14: Create Table

- 1 2. Kemudian pilih upload pada field *create table from* yang dapat dilihat pada gambar 3.15



Gambar 3.15: Pilih Upload

- 2 3. Pilih lokasi file, pilih format, dan nama tabel file yang dapat dilihat pada gambar 3.16



Gambar 3.16: Pilih Lokasi dan Format File

3.2 Dataset yang Digunakan pada HTTP Archive

Dataset pada HTTP Archive yang digunakan pada skripsi ini adalah sebagai berikut:

1. technologies Pada tabel technologies terdapat beberapa kolom seperti url, category, app, dan info. Url adalah alamat dari sebuah website. Contoh dari dataset dapat dilihat pada tabel 3.1

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.palaciodelslampadas.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 3.1: Technologies Desktop Data Sample

3.3 Langkah-Langkah Query Yang Dilakukan

Pada section ini akan dijelaskan tentang langkah-langkah *query* yang dilakukan dalam memperoleh data. Data yang diambil adalah data percobaan sebanyak 10 data. Data yang diambil merupakan dataset dari tabel technologies 2020_08_01:

3.3.1 Mengumpulkan Daftar Website

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

```
SELECT url
FROM 'httparchive.technologies.2020_08_01_*'
GROUP BY url
LIMIT 10
```

Pada *query* diatas akan dilakukan pemilihan pada kolom url dengan menggunakan perintah SELECT dari project httparchive dataset *technologies* tabel 2020_08_01_* dengan menggunakan perintah FROM. Mengelompokkan pada kolom url yang dilakukan dengan menggunakan perintah GROUP BY sehingga tidak ada nama url yang sama. Kolom akan dibatasi sebanyak 10 baris dengan menggunakan perintah LIMIT. Sepuluh contoh hasil keluaran dari *query* diatas dapat dilihat pada 3.2:

Row	url
1	https://www.theinsider.life/
2	http://www.mtctutorials.com/
3	https://noticias24horases.com.br/
4	https://www.tonyburke.com.au/
5	http://www.bakedbyjoanna.com/
6	https://stuftburgerbar.com/
7	https://www.skagitpowersports.com/
8	http://www.arazatimaderas.com/
9	https://oasisexc.com/
10	https://www.captainslanding.com/

Tabel 3.2: Hasil Pengumpulan Daftar Website

3.3.2 Mencari Aplikasi Yang Digunakan Website

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

```
SELECT DISTINCT url, app
FROM 'httparchive.technologies.2020_08_01_*'
ORDER BY url asc
LIMIT 10
```

Pada *query* diatas akan dilakukan pemilihan pada kolom url dan app dengan menggunakan perintah SELECT dari project httparchive dataset *technologies* tabel 2020_08_01_* dengan menggunakan perintah FROM. Kolom akan diurutkan berdasarkan url secara *ascending*. Kolom akan dibatasi sebanyak 10 baris dengan menggunakan perintah LIMIT. Sepuluh contoh hasil keluaran dari *query* diatas dapat dilihat pada tabel 3.3:

Row	url	app
1	http://0-1.ru/	Liveinternet
2	http://0-1.ru/	Yandex.Metrika
3	http://0-1.ru/	IIS
4	http://0-1.ru/	Microsoft ASP.NET
5	http://0-1.ru/	YouTube
6	http://0-1.ru/	Windows Server
7	http://0-10-10.cocolog-nifty.com/	Nginx
8	http://0-10-10.cocolog-nifty.com/	Twitter
9	http://0-10-10.cocolog-nifty.com/	jQuery
10	http://0-10-10.cocolog-nifty.com/	Osano

Tabel 3.3: Contoh Aplikasi Yang Digunakan Website

3.3.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai

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

```

4 SELECT tableName.app, num.num_sites , versioned.versioned_count , unversioned.unversioned_count
5 FROM
6 (SELECT DISTINCT app
7 FROM 'httparchive.technologies.2020_08_01_*' ) tableName
8
9 LEFT JOIN
10
11 (SELECT tabel1.app, count(app) AS versioned_count
12 FROM 'httparchive.technologies.2020_08_01_*' AS tabel1
13 WHERE tabel1.app!="" AND tabel1.info != ""
14 GROUP BY tabel1.app) AS versioned
15
16 ON(versioned.app = tableName.app)
17
18 LEFT JOIN
19
20 (SELECT tabel2.app, count(app) AS unversioned_count
21 FROM 'httparchive.technologies.2020_08_01_*' AS tabel2
22 WHERE tabel2.app!="" AND tabel2.info = ""
23 GROUP BY tabel2.app) AS unversioned
24
25 ON (unversioned.app = tableName.app)
26
27 LEFT JOIN
28
29 (SELECT app, count(url) AS num_sites
30 FROM 'httparchive.technologies.2020_08_01_*'
31 GROUP BY app) AS num
32
33 ON (tableName.app = num.app)
34 LIMIT 10
35

```

Pada *query* diatas akan dibuat beberapa tabel baru yang bersifat sementara. Pada baris ke tiga dan empat, query akan mengembalikan tabel yang berisi semua app yang ada pada tabel menggunakan perintah SELECT dan menggunakan DISTINCT agar app yang ditampilkan hanya keluar satu kali. Data diambil dari project httparchive dataset *technologies* tabel 2020_08_01_* dengan menggunakan perintah FROM. Kemudian pada baris ke delapan sampai sebalas, query akan mengembalikan tabel yang berisi app dan jumlah app yang memiliki info tidak kosong atau memiliki informasi versi. Pada baris tujuh belas sampai dua puluh, query akan mengembalikan tabel yang berisi app dan jumlah ap yang tidak memiliki informasi versi. Pada baris dua puluh enam sampai dengan baris dua puluh delapan, query akan mengembalikan tabel app, jumlah url yang menggunakan app tersebut. Kemudian semua tabel tersebut akan digabungkan dengan perintah

- 1 LEFT JOIN. Kemudian dengan menggunakan perintah SELECT, akan dipanggil beberapa variabel
 2 dari setiap kolom dari setiap tabel. Kolom yang diambil berupa: app, jumlah situs yang dipakai
 3 aplikasi (num_sites), jumlah aplikasi yang memiliki versi (versioned_count), dan jumlah aplikasi
 4 yang tidak memiliki versi (unversioned_count). Kolom akan dibatasi sebanyak 10 baris dengan
 5 menggunakan perintah LIMIT. Sepuluh contoh hasil keluaran dari *query* diatas dapat dilihat pada
 6 tabel 3.4:

Row	app	num_sites	versioned_count	unversioned_count
1	jQuery	10.003.030	9.979.001	24.029
2	Apache	4.067.380	1.118.200	2.949.180
3	PHP	5.977.790	2.522.620	3.455.170
4	MySQL	4.047.343	null	4.047.343
5	Microsoft SharePoint	14.419	11.402	3.017
6	YouTube	1.028.360	null	1.028.360
7	Microsoft ASP.NET	865.276	407.366	457.910
8	Google Code Prettify	32.171	null	32.171
9	Typekit	253.890	253.203	687
10	Slick	759.805	66.249	693.556

Tabel 3.4: Hasil Pengelompokan Aplikasi Beserta Jumlah *Versioned* Dan *Unversioned*

- 7 Pada [5], jumlah data yang digunakan lebih sedikit sehingga jumlah keseluruhan data juga akan
 8 berbeda. Terdapat beberapa aplikasi yang sama sehingga dapat dibandingkan datanya. Tabel pada
 9 [5] dapat dilihat pada tabel 3.5:

Name	num-sites	avg-confidence	num-unversioned	num-versioned	website	num-supported-version
jQuery	1.011	99.70	14	997	https://jquery.com	>=3
Bootstrap	340	99.30	88	342	https://getbootstrap.com	>=4
jQuery Migrate	298	99.66	31	267	https://github.com/jquery/jquery-migrate	?
PHP	591	99.83	348	245	https://www.php.net	>=7.2
Font Awesome	400	99.50	160	240	https://fontawesome.com	>=5
jQuery UI	176	99.43	7	169	https://jqueryui.com	?
WordPress	346	100.00	181	165	https://wordpress.org	>=5.4.2
Underscore.js	124	24.19	2	122	https://underscorejs.org	?
Lodash	125	59.20	3	122	https://lodash.com	?

Tabel 3.5: Tabel Sepuluh Data Aplikasi Pada [5]

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 tabel A.

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

Setelah mendapatkan data versi minimal dari setiap aplikasi, data tersebut akan dibandingkan dengan versi aplikasi yang dipakai *url*. *Supported* adalah versi aplikasi dari yang dipakai *url* masih mendukung atau diatas atau sama dengan versi yang didukung didokumen. *unsupported* adalah versi aplikasi dari yang dipakai *url* sudah tidak mendukung atau dibawah versi yang didukung didokumen. *not_versioned* adalah versi aplikasi dari *url* tidak ditampilkan. *non_conclusive* adalah versi aplikasi tidak dapat ditentukan.

```

CREATE TEMP FUNCTION normaizedSemanticVersion(semanticVersion STRING)
AS ((
  SELECT STRING_AGG(
    IF(isDigit, REPEAT('0', 100 - LENGTH(chars)) || chars, chars) ORDER BY grp
  )
  FROM (
    SELECT grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,
    FROM (
      SELECT OFFSET, char, isDigit,
      COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
    FROM UNNEST(SPLIT(semanticVersion, '')) AS char WITH OFFSET,
    UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit
    )
  ) GROUP BY grp, isDigit
  ));
CREATE TEMP FUNCTION compareSemanticVersions(
  normSemanticVersion1 STRING,
  normSemanticVersion2 STRING)
AS ((
  SELECT CASE
    WHEN info < min_supported THEN 'UNSUPPORTED'
    ELSE 'SUPPORTED'
  END
  FROM UNNEST([STRUCT(
    normaizedSemanticVersion(normSemanticVersion1) AS info,
    normaizedSemanticVersion(normSemanticVersion2) AS min_supported
  )])
  ));
WITH test AS (
  SELECT url, category, app, if (array_length(split(info, ".") > 2, split(info, ".")[offset(0)] || "." || split(info, ".")[offset(1)], info) as info, min_supported
  FROM 'httparchive-bigquery-346414.app_min_supported_and_info.app_min_supported_and_info'
  where info != ""
  )
  SELECT url, category, app, info, min_supported, if(info = '', "NOT VERSIONED",if(min_supported = '?','NON CONCLUSIVE',
    compareSemanticVersions(info, min_supported)) ) as result
  FROM test
  ORDER BY url

```

Pada awalnya akan dibuat sebuah fungsi dengan nama `normaizedSemanticVersion(semanticVersion STRING)`, dan `compareSemanticVersions(normSemanticVersion1 STRING, normSemanticVersion2 STRING)`. Kemudian pada baris 29 sampai dengan 33, akan dibuat sebuah tabel sementara untuk membuat group versi aplikasi yang dipisahkan berdasarkan titik.

Setelah melakukan version compare tersebut, data diteliti dan berikut ini adalah hasil sepuluh data yang dapat dilihat pada tabel 3.6. Data diambil berdasarkan banyak aplikasi yang dipakai oleh *url* tertentu.

url	supported	unsupported	not_versioned	non_conclusive
authservice.pegipegi.com	0	9	224	2
serviceauth.pegipegi.com	0	13	220	2
mcatsselfprep.com	0	14	52	8
perpetua.it	0	14	50	12
sulava.com	0	10	59	10
theraceclub.com	2	12	48	16
jobs.discover.com	4	8	58	8
dickssportinggoods.jobs	4	8	56	8
careers.symphonytalent.com	4	8	56	8
jobs.cedarfair.com	4	8	52	12

Tabel 3.6: Hasil Perbandingan Aplikasi Berdasarkan url

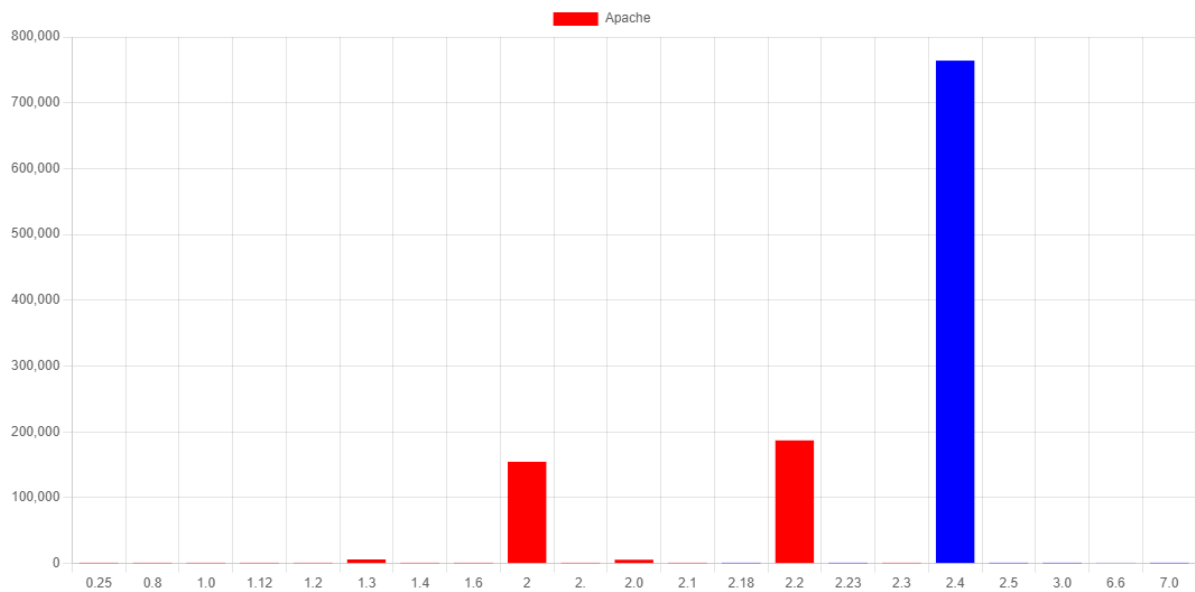
- 1 Data juga dibandingkan berdasarkan aplikasi tertentu. Data yang dihasilkan adalah num_sites
2 atau jumlah url yang menggunakan aplikasi tertentu, app, supported atau aplikasi yang masih
3 didukung, unsupported atau aplikasi yang sudah tidak didukung, not_versioned atau aplikasi yang
4 tidak diberi informasi versi, dan non_conclusive atau versi aplikasi tidak dapat ditentukan. Hasil
5 dari data dapat dilihat pada tabel 3.7.

num_sites	app	supported	unsupported	not_versioned	non_conclusive
10.003.030	jQuery	1.604.830	8.374.171	24.029	0
8.190.668	Google Analytics	0	0	8.190.668	0
7.494.642	WordPress	350	4.891.016	2.603.276	0
7.230.612	Nginx	652	1.789.692	5.440.268	0
5.977.790	PHP	167.095	2.355.525	3.455.170	0
5.481.111	Google Font API	0	0	5.481.111	0
4.529.823	Google Tag Manager	0	0	4.529.823	0
4.067.380	Apache	764.690	353.510	2.949.180	0
4.047.343	MySQL	0	0	4.047.343	0

Tabel 3.7: Hasil Perbandingan Aplikasi

7 3.4 Hasil Sample Data Dengan Beberapa Aplikasi

- 8 Diambil satu data sample dengan aplikasi dan informasi versinya. Data sample tersebut merupakan
9 data Apache. Data dapat dilihat pada gambar 3.17.



Gambar 3.17: Data Sample Jumlah Aplikasi Dengan Versi yang Dipakai

- 1 Pada data 3.17 terdapat bagian bawah yang menunjukkan informasi versi dari aplikasi dan
- 2 bagian kiri merupakan jumlah url yang menggunakan aplikasi. Chart yang berwarna merah adalah
- 3 chart yang menunjukkan versi aplikasi tersebut sudah tidak didukung. Chart yang berwarna biru
- 4 menunjukkan versi aplikasi tersebut masih didukung.

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. query yang dilakukan sama dengan bab sebelumnya 3.3 tetapi tidak menggunakan limit.

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. 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.

```
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
```

```

17 (SELECT tabel1.app, count(app) AS versioned_count
18 FROM 'httparchive.technologies.2020_08_01_*' AS tabel1
19 WHERE tabel1.app!="" AND tabel1.info != ""
20 GROUP BY tabel1.app) AS versioned
21
22 ON(versioned.app = tabelName.app)
23
24 LEFT JOIN
25
26 (SELECT tabel2.app, count(app) AS unversioned_count
27 FROM 'httparchive.technologies.2020_08_01_*' AS tabel2
28 WHERE tabel2.app!="" AND tabel2.info = ""
29 GROUP BY tabel2.app) AS unversioned
30
31 ON (unversioned.app = tabelName.app)
32
33 LEFT JOIN
34
35 (SELECT app, count(url) AS num_sites
36 FROM 'httparchive.technologies.2020_08_01_*'
37 GROUP BY app) AS num
38
39 ON (tabelName.app = num.app)
40

```

4.1.4 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 A.

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

Setelah mendapatkan data versi minimal dari setiap aplikasi, data tersebut akan dibandingkan dengan versi aplikasi yang dipakai *url_supported* adalah versi aplikasi dari yang dipakai url masih mendukung atau diatas atau sama dengan versi yang didukung didokumen. *unsupported* adalah versi aplikasi dari yang dipakai url sudah tidak mendukung atau dibawah versi yang didukung didokumen. *not_versioned* adalah versi aplikasi dari url tidak ditampilkan. *non_conclusive* adalah versi aplikasi tidak dapat ditentukan. Data diambil berdasarkan banyak aplikasi yang dipakai oleh url tertentu. Data yang sudah dibandingkan juga digunakan untuk mencari jumlah website yang jumlah semua aplikasinya yang masih didukung. Terdapat 4.820 jumlah aplikasi yang digunakan website. Berikut adalah *query* yang digunakan untuk mencari datanya:

```

41 SELECT url1.url, url1.jumlah1, url2.jumlah2
42 FROM
43 (
44 SELECT url, count(app) AS jumlah1
45 FROM 'httparchive-bigquery-346414.app_result.app_result'
46 WHERE result = "SUPPORTED"
47 GROUP BY url
48 ORDER BY url ASC
49 ) AS url1
50
51 JOIN
52 (
53 SELECT url, count(app) AS jumlah2
54 FROM 'httparchive-bigquery-346414.app_result.app_result'
55 GROUP BY url
56 ORDER BY url ASC
57 ) AS url2
58 ON url1.url = url2.url
59 WHERE url1.jumlah1 = url2.jumlah2
60

```

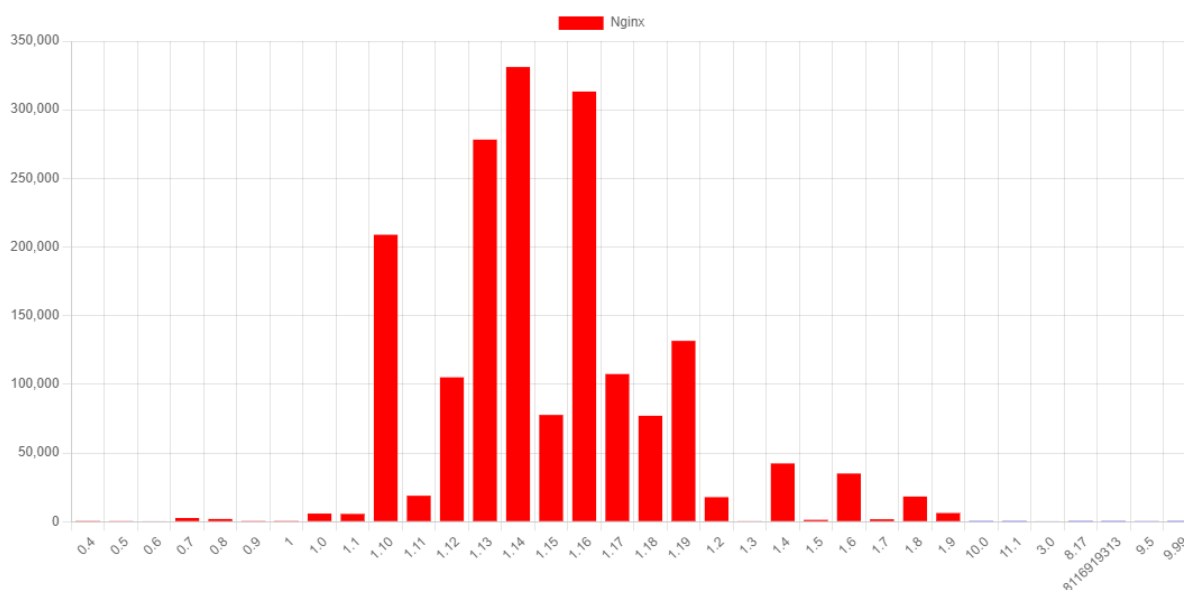

1 *Project* `httparchive-bigquery-346414` dengan nama dataset `app_result` dan tabel `app_result` adalah
2 sebuah tabel pembantu. Tabel ini berasal dari hasil `version compare` pada 3.3.5. *Project* `httparchive-`
3 `bigquery-346414` ini dibuat berdasarkan data dari *project* `httparchive`, dataset `technologies`, dan
4 tabel `2020_08_01_*` yang kemudian dibuat tabel baru agar *query* tidak dipanggil beberapa kali.
5 Pada *query* diatas awalnya dibuat sebuah tabel yang bersifat sementara. Tabel diambil dari *project*
6 `httparchive-bigquery-346414` dengan nama dataset `app_result` dan tabel `app_result`. Pada tabel ini
7 akan dicari url dan data dengan informasi versi dari aplikasi yang masih didukung url tersebut, tabel
8 diberi nama `url1`. Kemudian tabel akan digabungkan dengan tabel lain yang bersifat sementara.
9 Pada tabel ini dicari semua url dan jumlah aplikasi yang dipakai oleh url tersebut, tabel diberi
10 nama `url2`. Hasil akhir dari *query* ini berupa url yang dan jumlah dari tabel `url1` dan tabel `url2`.

11 4.2 Hasil Sample Data

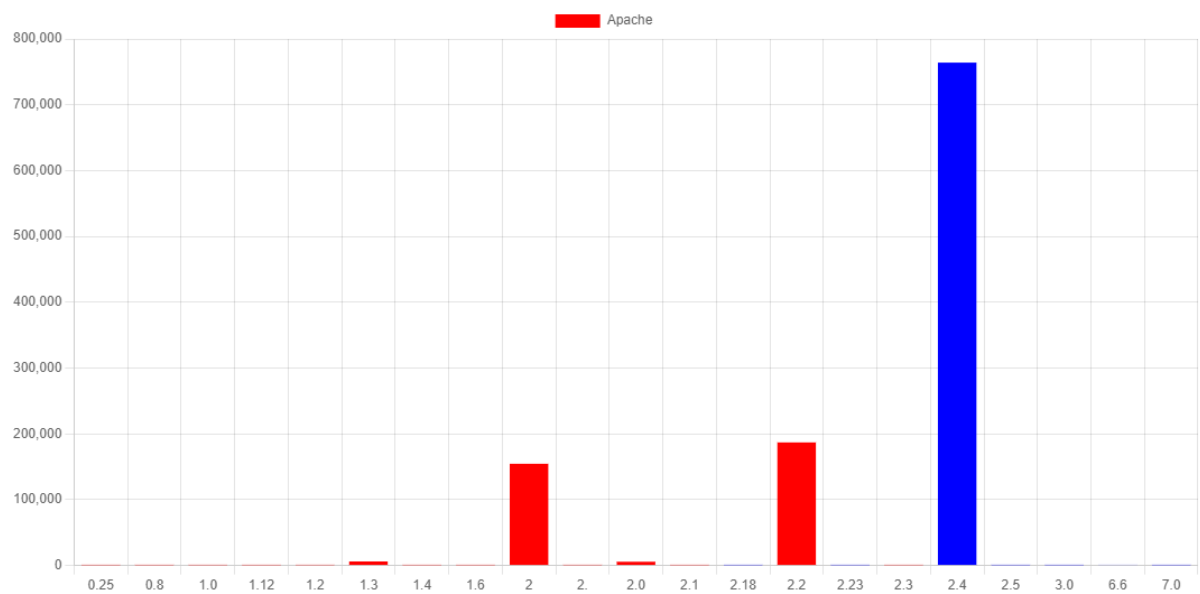
12 Data yang ditampilkan adalah data beberapa aplikasi yang sudah dipisahkan berdasarkan aplikasi
13 dan nomor versi dari aplikasi yang dipakai serta jumlahnya dalam bentuk *chart*.

14 4.2.1 Apache dan Nginx

15 Apache dan Nginx merupakan dua web servers yang paling banyak digunakan. Pada dua web
16 server ini, aplikasi Apache memiliki lebih banyak jumlah yang supported daripada aplikasi Nginx.
17 Pada aplikasi Nginx terdapat 5.440.268 aplikasi yang *unversioned*. Versi pada aplikasi Nginx yang
18 paling banyak digunakan adalah versi 1.14 dengan jumlah 267.102. Pada aplikasi Apache terdapat
19 2.949.180 aplikasi yang *unversioned*. Versi pada aplikasi Apache yang paling banyak digunakan
20 adalah versi 2.4 dengan jumlah 154.533. Berikut ini adalah chart yang dapat dilihat pada gambar
21 4.1 dan 4.2:

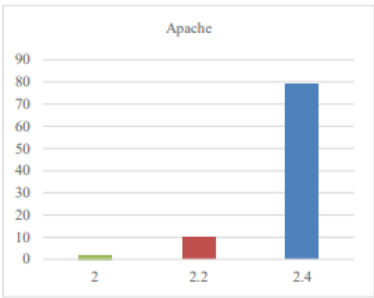


Gambar 4.1: Aplikasi Nginx

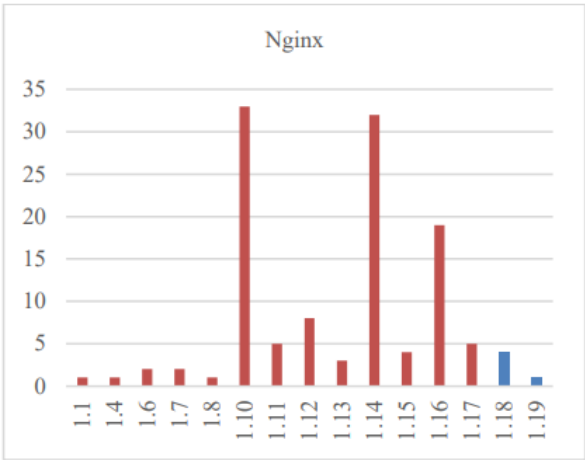


Gambar 4.2: Aplikasi Apache

- 1
- Berdasarkan penelitian dengan aplikasi yang sama, didapatkan hasil dalam bentuk chart. Chart
- 2
- yang dibandingkan dapat dilihat pada gambar 4.3 dan gambar 4.4.



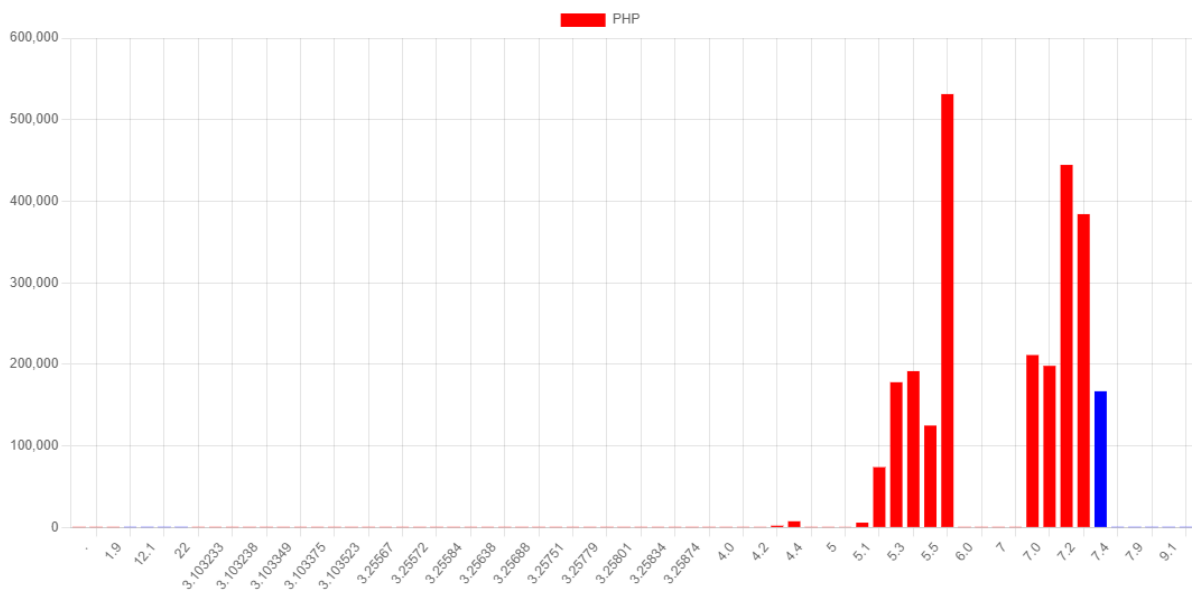
Gambar 4.3: Aplikasi Apache dari [5]



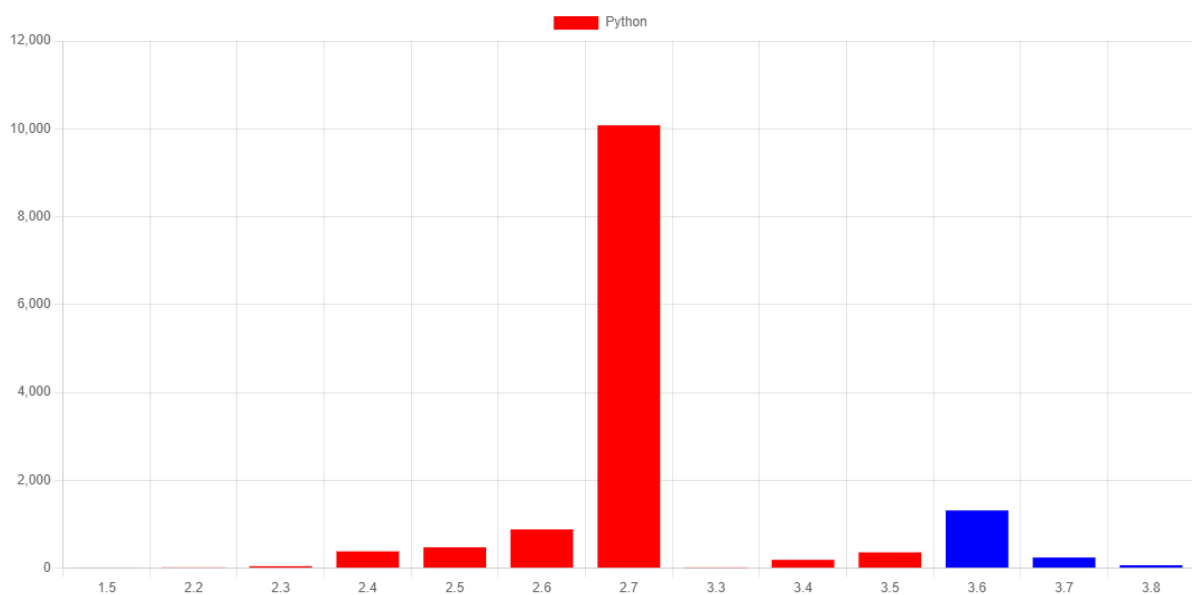
Gambar 4.4: Aplikasi Apache dari [5]

1 4.2.2 PHP dan Python

2 PHP merupakan bahasa pemrograman yang digunakan dalam pembuatan website. PHP menjadi
 3 bahasa pemrograman yang paling banyak digunakan. Pada aplikasi PHP terdapat 3.455.170 aplikasi
 4 yang *unversioned*. Versi pada aplikasi PHP yang paling banyak digunakan adalah versi 5.6 dengan
 5 jumlah 358.750. Python merupakan bahasa pemrograman tingkat tinggi dan berorientasi objek.
 6 Python adalah bahasa pemrograman tingkat tinggi karena perintah atau kode program yang
 7 digunakan sudah mirip dengan bahasa manusia. Pada aplikasi Python terdapat 360.531 aplikasi
 8 yang *unversioned*. Versi pada aplikasi Python yang paling banyak digunakan adalah versi 2.7 dengan
 9 jumlah 7.481. Berikut ini adalah chart yang dapat dilihat pada gambar 4.5 dan 4.6:

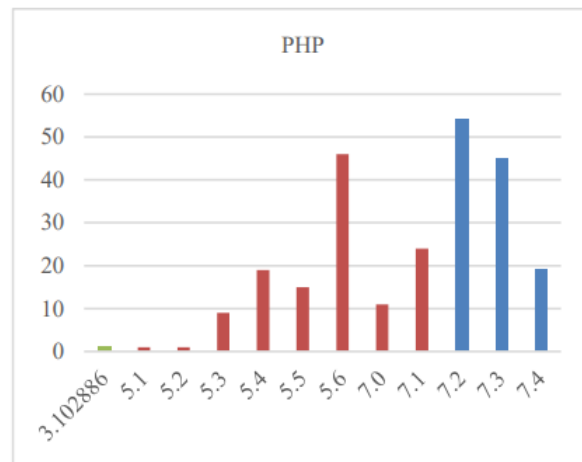


Gambar 4.5: Aplikasi PHP



Gambar 4.6: Aplikasi Python

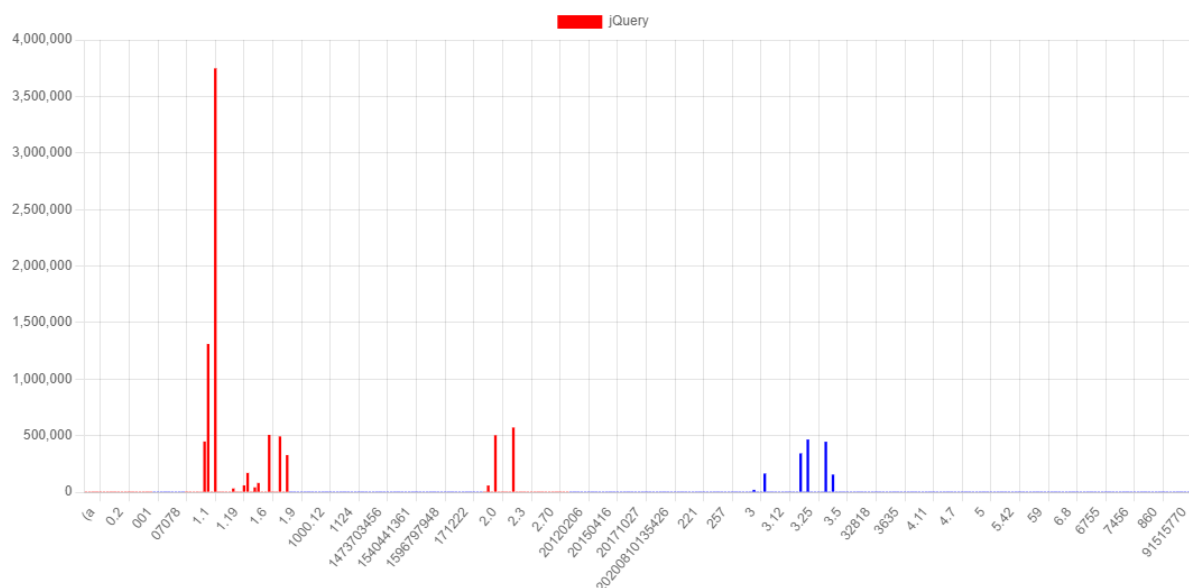
- 1 Berdasarkan penelitian dengan aplikasi yang sama, didapatkan hasil dalam bentuk chart. Chart
 2 yang dibandingkan dapat dilihat pada gambar 4.7.



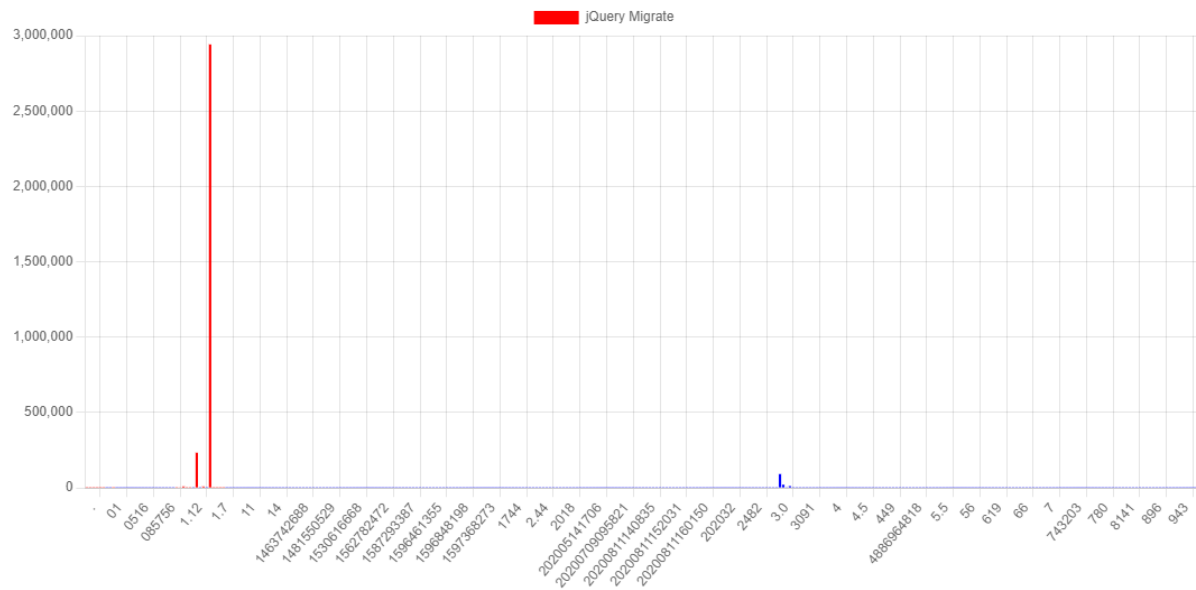
Gambar 4.7: Aplikasi PHP dari [5]

3 4.2.3 jQuery dan jQuery Migrate

- 4 jQuery dan jQuery Migrate merupakan *javascript libraries* yang paling banyak digunakan. jQuery
 5 berfungsi untuk membantu mengatur interaksi antara javascript dan html pada sisi *client*. Pada
 6 aplikasi jQuery terdapat 24.029 aplikasi yang *unversioned*. Versi pada aplikasi jQuery yang paling
 7 banyak digunakan adalah versi 1.12 dengan jumlah 3.603.522. jQuery Migrate berfungsi untuk
 8 membantu memulihkan API yang telah dihapus dan menunjukkan peringatan pada *browser concole*.
 9 Pada aplikasi jQuery Migrate terdapat 268.962 aplikasi yang *unversioned*. Versi pada aplikasi
 10 jQuery yang paling banyak digunakan adalah versi 1.4 dengan jumlah 2.935.408. Hasil chart dapat
 11 dilihat pada gambar 4.8 dan 4.9

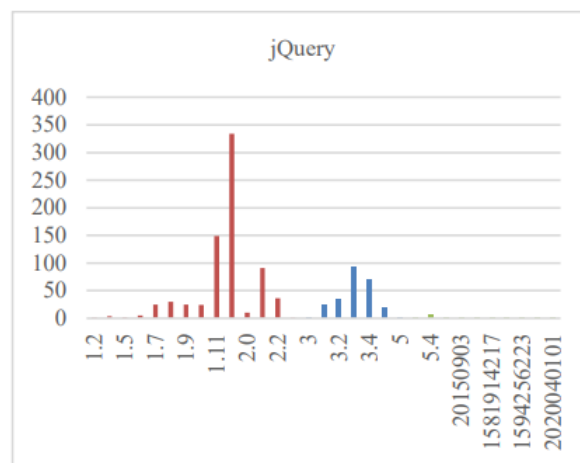


Gambar 4.8: Aplikasi jQuery



Gambar 4.9: Aplikasi jQuery Migrate

- 1 Berdasarkan penelitian dengan aplikasi yang sama, didapatkan hasil dalam bentuk chart. Chart
- 2 yang dibandingkan dapat dilihat pada gambar 4.10.



Gambar 4.10: Aplikasi jQuery dari [5]

BAB 5

PEMBANGUNAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan tentang implementasi perangkat lunak.

5.1 Implementasi Perangkat Lunak

Perangkat lunak dibuat sesuai dengan data pada Bab 3 dan 4. Dalam skripsi ini terdapat 3 bagian yaitu:

- BigQuery

Bagian ini adalah representasi dari data. Pada BigQuery akan dilakukan beberapa query untuk mendapatkan data yang diinginkan. Kemudian hasil-hasil dari query tersebut disimpan dalam sebuah dataset.

- React.js

Bagian ini merupakan bagian tampilan atau web-client. Pada bagian ini bertujuan untuk menampilkan hasil data atau visualisasi data.

- Node.js

Bagian ini adalah sebuah penghubung antara data dari BigQuery dan web-client.

5.1.1 BigQuery

Bagian ini adalah representasi dari data. Pada BigQuery akan dilakukan beberapa query untuk mendapatkan data yang diinginkan. Kemudian hasil-hasil dari query tersebut disimpan dalam sebuah dataset.

Minimal Supported Data

Minimal supported data didapatkan dengan mencari sendiri informasi versi dari setiap aplikasi berdasarkan dokumen resminya. Kemudian data-data tersebut akan dimasukkan kedalam excel dan didownload sebagai csv. Data-data csv tersebut akan di-upload menggunakan fitur yang ada pada BigQuery dan akan terbentuk tabel yang berisi csv tersebut. Tabel dari excel tersebut dapat dilihat pada lampiran [A](#)

Menyatukan Tabel Informasi Versi dan Minimal Supported

```
select jumlah.url, appl.category, appl.app, appl.info, sup.min_supported
from(
select url, count(app) as jumlah
from 'httparchive.technologies.2020_08_01_*'
group by url
```

```

16  order by jumlah desc
27  ) as jumlah
38
49  join
50
61  (select url, category, app, info
72  from 'httparchive.technologies.2020_08_01_.*'
83  ) as appl
94  on jumlah.url = appl.url
105
116
127  join
138
149  (select distinct app, min_supported
150  from 'httparchive-bigquery-346414.Step.min_supported_app') as sup
161  on appl.app = sup.app
172
183  order by jumlah.jumlah desc

```

20 Pada bagian ini terdapat beberapa query yang disatukan sehingga membentuk suatu tabel yang
 21 menyatukan informasi versi yang dipakai aplikasi dengan minimal supported.

- 22 • Mencari url dan jumlah app yang dipakai url tersebut.

```

23  1      select url, count(app) as jumlah
24  2      from 'httparchive.technologies.2020_08_01_.*'
25  3      group by url
26  4      order by jumlah desc
27
28

```

- 29 • Mencari url, kategori, app, informasi versi.

```

30  1      select url, category, app, info
31  2      from 'httparchive.technologies.2020_08_01_.*'
32
33

```

- 34 • Mencari app dan minimal supported

```

35  1      select distinct app, min_supported
36  2      from 'httparchive-bigquery-346414.Step.min_supported_app'
37
38

```

39 Version Compare

```

40  1  CREATE TEMP FUNCTION normaizedSemanticVersion(semanticVersion STRING)
41  2  AS ((
42  3  SELECT STRING_AGG(
43  4  IF(isDigit, REPEAT('0', 100 - LENGTH(chars)) || chars, chars) ORDER BY grp
44  5  )
45  6  FROM (
46  7  SELECT grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,
47  8  FROM (
48  9  SELECT OFFSET, char, isDigit,
49  10 COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
50  11 FROM UNNEST(SPLIT(semanticVersion, '')) AS char WITH OFFSET,
51  12 UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit
52  13 )
53  14 GROUP BY grp, isDigit
54  15 )));
55  16 CREATE TEMP FUNCTION compareSemanticVersions(
56  17 normSemanticVersion1 STRING,
57  18 normSemanticVersion2 STRING)
58  19 AS ((
59  20 SELECT CASE
60  21 WHEN info < min_supported THEN 'UNSUPPORTED'
61  22 ELSE 'SUPPORTED'
62  23 END
63  24 FROM UNNEST([STRUCT(
64  25 normaizedSemanticVersion(normSemanticVersion1) AS info,
65  26 normaizedSemanticVersion(normSemanticVersion2) AS min_supported
66  27 )])
67  28 );
68  29 WITH test AS (
69  30 SELECT url, category, app, if (array_length(split(info, ".") > 2, split(info, ".")[offset(0)] || "." || split(info, ".")[
70  31 offset(1)], info) as info, min_supported
71  32 FROM 'httparchive-bigquery-346414.app_min_supported_and_info.app_min_supported_and_info'
72  33 where info != ""
73  34 )

```



```

134 SELECT url, category, app, info, min_supported, if(info = '', "NOT VERSIONED",if(min_supported = '?','NON CONCLUSIVE',
135   compareSemanticVersions(info, min_supported)) ) as result
136 FROM test
137 ORDER BY url

```

Pada query diatas adalah query yang dilakukan untuk melakukan version compare. Berikut ini adalah beberapa step atau query yang dilakukan untuk membuat version compare tersebut:

1. Normalisasi pada Versi

Pada bagian ini query digunakan untuk menormalisasikan digit dari setiap informasi versi sehingga dapat di bandingkan. Berikut ini adalah query yang digunakan:

- Mengecek Digit Dari Versi

```

1 SELECT OFFSET, char, isDigit,
2   COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
3 FROM UNNEST(SPLIT('3.14', '')) AS char WITH OFFSET,
4   UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit

```

Pada query ini akan mengembalikan offset atau index yang dimulai dari nol, char sebagai input atau parameter berupa informasi versi, isDigit adalah setiap angka dari setiap input, dan jumlah boolean yang bernilai false jika input berupa angka.

Berikut ini adalah hasil dari query diatas:

OFFSET	char	isDigit	grp
0	3	true	0
1	.	false	1
2	1	true	1
3	4	true	1

- Memotong Setiap Char Dari Version

```

1 SELECT grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,
2   FROM (
3     SELECT OFFSET, char, isDigit,
4       COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
5     FROM UNNEST(SPLIT('3.14', '')) AS char WITH OFFSET,
6       UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit
7   )
8   GROUP BY grp, isDigit

```

Pada query ini akan mengembalikan jumlah boolean yang bernilai false, boolean yang menandakan apakah input merupakan digit atau bukan, dan input yang dibagi-bagi dalam bentuk string.

Berikut ini adalah hasil dari query diatas:

grp	isDigit	chars
0	true	3
1	false	.
1	true	14

- Normalisasi Informasi Versi

```

1 CREATE TEMP FUNCTION normaizedSemanticVersion(semanticVersion STRING)
2 AS ((
3   SELECT STRING_AGG(
4     IF(isDigit, REPEAT('0', 100 - LENGTH(chars)) || chars, chars) ORDER BY grp
5   )
6   FROM (
7     SELECT grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,

```



```

1      1      SELECT url, category, app, info, min_supported, if(info = '', "NOT VERSIONED",if(min_supported = '?','NON
2              CONCLUSIVE',compareSemanticVersions(info, min_supported)) ) as result
3      2      FROM test
4      3      ORDER BY url

```

Pada query ini akan mengembalikan url, kategori dari aplikasi, aplikasi yang dipakai, informasi versi, minimal supported, dan hasil atau result dari query yang menunjukkan jika aplikasi tersebut SUPPORTED atau UNSUPPORTED.

Ketika semua query disatukan, berikut adalah 10 contoh hasilnya:

url	category	app	info	min sup	result
http://0-1.ru/	Analytics	Yandex.Metrika		null	NOT VERSIONED
http://0-1.ru/	Web Frameworks	Microsoft ASP.NET		3.1.20	NOT VERSIONED
http://0-1.ru/	Video Players	YouTube		null	NOT VERSIONED
http://0-1.ru/	Web servers	IIS	6.0	8	UNSUPPORTED
http://0-1.ru/	Operating systems	Windows Server		null	NOT VERSIONED
http://0-1.ru/	Analytics	Liveinternet		null	NOT VERSIONED
http://0-10-10.cocolog-nifty.com/	Tag managers	Google Tag Manager		null	NOT VERSIONED
http://0-10-10.cocolog-nifty.com/	Reverse proxies	Nginx	1.15	1.20	UNSUPPORTED
http://0-10-10.cocolog-nifty.com/	Web servers	Nginx	1.15	1.20	UNSUPPORTED
http://0-10-10.cocolog-nifty.com/	JavaScript libraries	jQuery	1.11	3	UNSUPPORTED

5.1.2 React.js

Bagian ini merupakan bagian tampilan atau web-client. Pada bagian ini bertujuan untuk menampilkan hasil data atau visualisasi data. Terdapat beberapa file dalam perangkat lunak ini, yaitu:

- Components

Didalam folder components terdapat beberapa fungsi yaitu:

- AppUrl.js

Pada fungsi ini digunakan untuk membuat tabel yang menampilkan url dengan jumlah dari setiap result (*supported, unsupported, not versioned, dan non conclusive*). Berikut

ini adalah kode yang digunakan:

```

1      import React, { useEffect, useState } from "react";
2
3      const MAX = 10;
4
5      export default function AppUrl() {
6          const [step, setStep] = useState(0);
7          const [data, setData] = useState([]);
8          const [isLoading, setIsLoading] = useState(true);
9          function getUrlData() {
10              setIsLoading(true);
11              fetch('http://localhost:3000/get/app/url?limit=${MAX}&offset=${step}')
12                  .then((res) => res.json())
13                  .then((data) => {
14                      setData(data);
15                      setIsLoading(false);
16                  });
17          }
18          useEffect(() => {
19              getUrlData();
20          }, [step]);
21          return (
22              <div className="container">
23                  <h2 className="title">App Url</h2>
24                  {data[0] && (
25                      <>
26                          <table className="table">
27                              <thead>
28                                  <tr>
29                                      <th>No</th>
30                                      <th>Object.keys(data[0]).map((val) => (
31                                          <th>{val}</th>
32                                      ))}
33                                  </tr>
34                              </thead>
35                              <tbody>
36                                  {data.map((val, index) => (
37                                      <tr key={index}>
38                                          <td>{index + 1 + MAX * step}</td>
39                                          <td>{Object.values(val).map((val) => (
40                                              <td>{val}</td>
41                                          ))}
42                                      </tr>
43                                  ))}
44                              </tbody>
45                          </table>
46                          <div className="action">
47                              <button
48                                  onClick={() => setStep((prev) => prev - 1)}
49                                  disabled={step <= 0 || isLoading}
50                              >
51                                  Prev
52                              </button>
53                              <button
54                                  onClick={() => setStep((prev) => prev + 1)}
55                                  disabled={isLoading}
56                              >
57                                  Next
58                              </button>
59                          </div>
60                      </>
61                  )}
62              </div>
63          );
64      }

```

— List.js

Pada fungsi ini digunakan untuk membuat tabel yang berisi list dari app. Berikut adalah kode yang digunakan:

```

1      import React, { useEffect, useState } from "react";
2      import { CategoryScale } from "chart.js";
3      import { Bar } from "react-chartjs-2";
4      import Chart from "chart.js/auto";
5
6      const MAX = 10;
7
8      export default function List() {
9          const [step, setStep] = useState(0);

```

```

1      10      const [data, setData] = useState([]);
2      11      const [isLoading, setIsLoading] = useState(true);
3      12      const [selectedData, setSelectedDate] = useState();
4      13
5      14      function color(arr) {
6      15          let temp = [];
7      16          for (let i = 0; i < arr.length; i++) {
8      17              const dataType = arr[i].result;
9      18              switch (dataType) {
10     19                  case "SUPPORTED":
11     20                      temp.push("blue");
12     21                      break;
13     22                  case "UNSUPPORTED":
14     23                      temp.push("red");
15     24                      break;
16     25                  default:
17     26                      temp.push("green");
18     27              }
19     28          }
20     29          return temp;
21     30      }
22     31      useEffect(() => {
23     32          Chart.register(CategoryScale);
24     33      }, []);
25     34
26     35      function getListData(page) {
27     36          setIsLoading(true);
28     37          fetch('http://localhost:3000/get/app/type?limit=${MAX}&offset=${page}')
29     38              .then((res) => res.json())
30     39              .then((data) => {
31     40                  setData(data);
32     41                  setIsLoading(false);
33     42              });
34     43      }
35     44
36     45      function getData(name) {
37     46          fetch('http://localhost:3000/get/app/name/${name}')
38     47              .then((res) => res.json())
39     48              .then((data) => {
40     49                  setSelectedDate({
41     50                      name,
42     51                      data: {
43     52                          labels: data.map((val) => String(val.info)),
44     53                          datasets: [
45     54                              {
46     55                                  label: name,
47     56                                  data: data.map((val) => String(val.jumlah)),
48     57                                  backgroundColor: color(data),
49     58                              },
50     59                          ],
51     60                      },
52     61                  });
53     62              });
54     63      }
55     64
56     65      useEffect(() => {
57     66          getListData(step);
58     67      }, [step]);
59     68      return (
60     69          <div className="container">
61     70              <h2 className="title">List APP</h2>
62     71              {data[0] && (
63     72                  <>
64     73                      <table className="table table-hover">
65     74                          <thead>
66     75                              <tr>
67     76                                  <th className="no">No</th>
68     77                                  <th>Name</th>
69     78                                  <th className="action-head"></th>
70     79                              </tr>
71     80                          </thead>
72     81                          <tbody>
73     82                              {data.map((val, index) => (
74     83                                  <tr key={index} onClick={() => getData(val.app)}>
75     84                                      <th className={index + 1 + MAX * step}</th>
76     85                                      <td>{val.app}</td>
77     86                                      <td>View</td>
78     87                                  </tr>
79     88                              ))}
80     89                          </tbody>
81     90                          </table>
82     91                      <div className="action">

```

```

1      92      <button
2      93      onClick={() => setStep((prev) => prev - 1)}
3      94      disabled={step <= 0 || isLoading}
4      95      >
5      96      Prev
6      97      </button>
7      98      <button
8      99      onClick={() => setStep((prev) => prev + 1)}
9      100     disabled={isLoading}
10     101     >
11     102     Next
12     103     </button>
13     104     </div>
14     105     </>
15     106     )}
16     107
17     108     {selectedData && (
18     109     <div className="selected">
19     110     <h3 className="title">{selectedData.name}</h3>
20     111     <Bar data={selectedData.data} />
21     112     </div>
22     113     )}
23     114 </div>
24     115 );
25     116 }

```

— Popular.js

Pada fungsi ini digunakan untuk membuat tabel yang berisi app yang populer berdasarkan jumlah url yang menggunakan app tersebut. Berikut ini adalah kode yang digunakan:

```

1      import React, { useEffect, useState } from "react";
2
3      const MAX = 10;
4
5      export default function Popular() {
6      const [step, setStep] = useState(0);
7      const [data, setData] = useState([]);
8      const [isLoading, setIsLoading] = useState(true);
9      function getPopularData() {
10     setIsLoading(true);
11     fetch('http://localhost:3000/get/app/popular?limit=${MAX}&offset=${step}')
12     .then((res) => res.json())
13     .then((data) => {
14         setData(data);
15         setIsLoading(false);
16     });
17     }
18     useEffect(() => {
19         getPopularData();
20     }, [step]);
21     return (
22     <div className="container">
23     <h2 className="title">Popular</h2>
24     {data[0] && (
25     <
26     <table className="table">
27     <thead>
28     <tr>
29     <th>No</th>
30     {Object.keys(data[0]).map((val) => (
31     <th>{val}</th>
32     ))}
33     </tr>
34     </thead>
35     <tbody>
36     {data.map((val, index) => (
37     <tr key={index}>
38     <td>{index + 1 + MAX * step}</td>
39     {Object.values(val).map((val) => (
40     <td>{val}</td>
41     ))}
42     </tr>
43     ))}
44     </tbody>
45     </table>
46     <div className="action">
47     <button
48     onClick={() => setStep((prev) => prev - 1)}
49     disabled={step <= 0 || isLoading}
50     >

```

```

1      51          Prev
2      52          </button>
3      53          <button
4      54              onClick={() => setStep((prev) => prev + 1)}
5      55              disabled={isLoading}
6      56          >
7      57              Next
8      58          </button>
9      59      </div>
10     60  </>
11     61  })
12     62  </div>
13     63  });
14     64  }

```

– Navbar.js

Pada fungsi ini digunakan untuk membuat header yang merujuk ke tabel pada AppUrl.js dan tabel pada Popular.js. Berikut adalah kode yang digunakan:

```

19     1  import React from "react";
20     2  import { Link } from "react-router-dom";
21
22     3
23     4  export default function Navbar() {
24     5      return (
25     6          <ul className="menu">
26     7              <li className="item">
27     8                  <a href="/">List</a>
28     9              </li>
29     10             <li className="item">
30     11                 <a href="/app-url">App Url</a>
31     12             </li>
32     13             <li className="item">
33     14                 <a href="/popular">Popular</a>
34     15             </li>
35     16         </ul>
36     17     );
37     18  }

```

• App.js

Dalam fungsi ini akan memanggil components yang dibuat untuk ditampilkan. Berikut ini adalah kode pada App.js:

```

42     1  import * as React from "react";
43     2  import Navbar from "./components/Navbar";
44     3  import { BrowserRouter, Routes, Route } from "react-router-dom";
45     4  import List from "./components/List";
46     5  import Popular from "./components/Popular";
47     6  import AppUrl from "./components/AppUrl";
48
49     7
50     8  function App() {
51     9      return (
52     10          <main className="app">
53     11              <Navbar />
54     12              <BrowserRouter>
55     13                  <Routes>
56     14                      <Route path="/" element={<List />} />
57     15                      <Route path="/app-url" element={<AppUrl />} />
58     16                      <Route path="/popular" element={<Popular />} />
59     17                  </Routes>
60     18              </BrowserRouter>
61     19          </main>
62     20      );
63     21  }
64     22
65     23  export default App;

```

5.1.3 Node.js

Bagian ini adalah sebuah penghubung antara data dari BigQuery dan web-client. Terdapat tiga bagian utama dalam perangkat lunak yang dibuat yaitu:

1. Features

Bagian ini merupakan sebuah folder yang berisi media untuk berkomunikasi dengan BigQuery. Dalam Features terdapat kelas GetApplication.js. Kelas GetApplication.js memiliki beberapa function untuk mendapatkan data dari BigQuery Berikut ini adalah contoh kode dalam kelas GetApplication.js:

```

1  const {BigQuery} = require('@google-cloud/bigquery');
2  const options = {
3    keyFilename: 'gsm-bigquery-credentials.json',
4    projectId: 'httparchive-bigquery-346414',
5  };
6  const bigquery = new BigQuery(options)
7
8
9  async function getApplications(app = "Apache") {
10    const getAppSql = 'select app, info, count(app) as jumlah, result from httparchive-bigquery-346414.app_result.
11      app_result where app = "${app}" and (result != "NON CONCLUSIVE" and result != "NOT VERSIONED")
12    group by app, info, result order by info ASC'
13    const options = {
14      query: getAppSql,
15      location: 'US',
16    };
17    const [job] = await bigquery.createQueryJob(options);
18    const [rows] = await job.getQueryResults();
19    return rows.filter(item => !item.info.includes("\\"));
20  }
21
22  async function getApplicationsType(limit = 5 , offset = 1) {
23    const getAppSql = 'select app from httparchive-bigquery-346414.Step.app_result where info != '' group by app
24      limit ${limit} offset ${offset}'
25    const options = {
26      query: getAppSql,
27      location: 'US',
28    };
29    const [job] = await bigquery.createQueryJob(options);
30    const [rows] = await job.getQueryResults();
31    return rows;
32  }
33
34  async function getApplicationsUrl(limit = 10, offset = 1) {
35    const getAppSql = 'select * from \'httparchive-bigquery-346414.URL_Result.url_result\' limit ${limit} offset $
36      {offset}'
37    const options = {
38      query: getAppSql,
39      location: 'US',
40    };
41    const [job] = await bigquery.createQueryJob(options);
42    const [rows] = await job.getQueryResults();
43    return rows;
44  }
45
46  async function getPopularTech(limit = 10, offset = 1) {
47    const getAppSql = 'select * from \'httparchive-bigquery-346414.numsite_app_result_count.
48      numsite_app_result_count\' limit ${limit} offset ${offset}'
49    const options = {
50      query: getAppSql,
51      location: 'US',
52    };
53    const [job] = await bigquery.createQueryJob(options);
54    const [rows] = await job.getQueryResults();
55    return rows;
56  }
57
58  module.exports = {getApplications, getApplicationsType, getApplicationsUrl, getPopularTech}

```

Berikut ini adalah penjelasan setiap fungsi:

- function getApplications(app = "Apache")

Pada fungsi ini akan mengembalikan app, info, jumlah app, result. Pada fungsi terdapat parameter untuk menentukan app yang ingin ditampilkan. Data tidak akan menampilkan result yang UNVERSIONED dan NON CONCLUSIVE.

- function getApplicationType(limit = 5, offset = 1)

Pada fungsi ini akan mengembalikan semua app. Pada fungsi ini terdapat parameter limit untuk membatasi data dan offset sebagai index data.

- function getApplicationsUrl(limit = 10, offset = 1)

Pada fungsi ini akan mengembalikan semua isi tabel. Pada fungsi ini terdapat parameter limit untuk membatasi data dan offset sebagai index data.

- function getPopularTech(limit = 10, offset = 1)

Pada fungsi ini akan mengembalikan semua isi tabel. Pada fungsi ini terdapat parameter limit untuk membatasi data dan offset sebagai index data.

2. Router Bagian ini berfungsi sebagai penghubung antara backend logic dengan web-client.

Berikut ini adalah contoh kodenya:

```
1  const express = require('express')
2  const {GetPopularTech} = require("../Controllers/GetPopularTech");
3  const {GetAppRecap} = require("../Controllers/GetRecap");
4  const {GetAppType} = require("../Controllers/GetType");
5  const {GetAppByName} = require("../Controllers/Applications");
6  const cors = require('cors')
7  const app = express()
8  const port = 3000
9  app.use(cors())
10 app.get('/get/app/name/:name', GetAppByName())
11 app.get('/get/app/type', GetAppType())
12 app.get('/get/app/url', GetAppRecap())
13 app.get('/get/app/popular', GetPopularTech())
14 app.listen(port, () => {
15   console.log('Listening at http://localhost:${port}')
16 })
```

3. Controllers Bagian ini berfungsi untuk mengimplementasikan *feature* berdasarkan *use case* yang di berikan. Berikut ini adalah kode yang digunakan:

- GetApplications.js

```
1  const {getApplications} = require("../Features/GetApplications");
2  const GetAppByName = () => {
3    return (req, res) => {
4      const {name} = req.params
5      getApplications(name).then((rows) => {
6        res.send(rows);
7      }).catch((e) => {
8        res.send(e.message)
9      });
10   };
11 }
12 module.exports = {GetAppByName}
```

- GetPopularTech.js

```
1  const {getPopularTech} = require("../Features/GetApplications");
2  const GetPopularTech = () => {
3    return (req, res) => {
4      const {limit, offset} = req.query
5      getPopularTech(limit, offset).then((rows) => {
6        res.send(rows);
7      }).catch((e) => {
8        res.send(e.message)
9      });
10   };
11 }
12 module.exports = {GetPopularTech}
```

- GetRecap.js

```
1  const {getApplicationsUrl} = require("../Features/GetApplications");
2  const GetAppRecap = () => {
3    return (req, res) => {
4      const {limit, offset} = req.query
5      getApplicationsUrl(limit, offset).then((rows) => {
6        res.send(rows);
7      }).catch((e) => {
```

```
1      res.send(e.message)
2    });
3  };
4  }
5  module.exports = {GetAppRecap}
```

- GetType.js

```
7
8
9  1      const {getApplicationsType} = require("../Features/GetApplications");
10  2      const GetAppType = () => {
11  3          return (req, res) => {
12  4              const {limit, offset} = req.query
13  5              getApplicationsType(limit, offset).then((rows) => {
14  6                  res.send(rows);
15  7              }).catch((e) => {
16  8                  res.send(e.message)
17  9              });
18  10          };
19  11      }
20  12      module.exports = {GetAppType}
```

BAB 6

KESIMPULAN DAN SARAN

Pada bab ini akan membahas kesimpulan berdasarkan implementasi, serta saran-saran untuk pengembangan berikutnya.

6.1 Kesimpulan

Berdasarkan data yang dikumpulkan, diperoleh kesimpulan-kesimpulan sebagai berikut:

1. Data pada HTTP Archive didapatkan dari public dataset.
2. Data berhasil direplikasi dengan mengikuti step atau langkah dari jurnal Pascal[5].
3. Setelah diolah, terdapat 4.820 *website* yang menggunakan aplikasi yang masih didukung.

6.2 Saran

Penulis memiliki beberapa saran untuk pengembangan aplikasi selanjutnya:

1. Data yang ada pada HTTP Archive masih kotor sehingga data harus dibersihkan dahulu agar data yang dihasilkan lebih valid.
2. Jika menggunakan teknologi BigQuery, dalam melakukan query harus dengan hati-hati karena setiap melakukan query akan dilakukan charge atau pembayaran.

DAFTAR REFERENSI

- [1] Tigani, J. dan Naidu, S. (2014) *Google BigQuery Analytics*, 1 edition. Wiley.
- [2] Google Developer 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] Nugroho, P. A. dan Steven, H. (2013) Measuring unsupported applications in indonesia popular websites. *JITEKI*, **66**, 595–614.
- [6] Dahl, R. nodejs. <https://nodejs.org/en/about/>.
- [7] Walke, J. Reactjs. <https://reactjs.org/docs/getting-started.html>.
- [8] Dahl, R. npmjs. <https://docs.npmjs.com/about-npm>.

LAMPIRAN A

DATA APLIKASI

app	num sites	unversioned	versioned	website	min supported	min supported version reference
jQuery	10,003,030	24,029	9,979,001	https://jquery.com	3	https://jquery.com/
WordPress	7,494,642	2,603,276	4,891,366	https://wordpress.org	5.8	https://github.com/twbs/release
jQuery Migrate	3,597,289	268,962	3,328,327	https://github.com/jquery/jquery-migrate	1.12	https://github.com/jquery/jquery-migrate
Font Awesome	3,827,039	1,040,702	2,786,337	https://fontawesome.com/	5	https://fontawesome.com/6#is-version-5-still-being-supported
jQuery UI	2,575,775	33,127	2,542,648	http://jqueryui.com	1.13.0	https://jqueryui.com/
jQuery UI	2,575,775	33,127	2,542,648	http://jqueryui.com	1.13.0	https://jqueryui.com/
Bootstrap	3,391,648	868,637	2,523,011	https://getbootstrap.com	4	https://github.com/twbs/release
Bootstrap	3,391,648	868,637	2,523,011	https://getbootstrap.com	4	https://github.com/twbs/release
PHP	5,977,790	3,455,170	2,522,620	http://php.net	7.4	https://www.php.net/supported-versions.php
Nginx	7,230,612	5,440,268	1,790,344	http://nginx.org/en	1.20	https://nginx.org/en/download.html
Modernizr	1,900,060	231,481	1,668,579	https://modernizr.com	?	https://github.com/Modernizr/Modernizr/releases

Yoast SEO	1,366,287	49	1,366,238	http://yoast.com	?	https://yoast.com/wordpress/plugins/seo/change-log-wordpress-seo/
Apache	4,067,380	2,949,180	1,118,200	http://apache.org	2.4	https://httpd.apache.org
FancyBox	827,596	72,541	755,055	http://fancyapps.com/fancybox	?	https://github.com/fancyapps/fancybox/releases
IIS	962,466	218,661	743,805	http://www.iis.net	8	https://support.microsoft.com/en-us/lifecycle/search?alpha=Windows%20Server%202012
WooCommerce	616,072	75,796	540,276	https://woocommerce.com	?	https://developer.woocommerce.com/releases/
Moment.js	555,640	37,730	517,910	https://momentjs.com	?	https://github.com/moment/moment/blob/develop/CHANGELOG.md
Lodash	517,688	10,781	506,907	http://www.lodash.com	?	https://github.com/lodash/lodash/releases
Underscore.js	540,820	101,362	439,458	http://underscorejs.org	?	http://underscorejs.org/
MediaElement.js	420,046	33	420,013	http://www.mediaelementjs.com	?	https://github.com/mediaelement/releases
MediaElement.js	420,046	33	420,013	http://www.mediaelementjs.com	?	https://github.com/mediaelement/releases
Microsoft ASP.NET	865,276	457,910	407,366	https://www.asp.net	3.1.20	https://dotnet.microsoft.com/platform/support/policy/dotnet-core
Revsilder	502,266	97,735	404,531	https://revolution.themepunch.com/	?	https://www.sliderrevolution.com/documentation/changelog/
RequireJS	334,538	22,537	312,001	http://requirejs.org	?	https://github.com/requirejs/requirejs/tags
OpenSSL	279,452	46	279,406	http://openssl.org	1.1.1	https://www.openssl.org/policies/releasestrat.html

Typekit	253,890	687	253,203	http://typekit.com	?	https://www.drupal.org/project/typekit/releases
Hammer.js	302,093	52,360	249,733	https://hammerjs.github.io	?	https://hammerjs.github.io/changelog/
Google PageSpeed	225,920	18	225,902	http://developers.google.com/speed/pagespeed/mod	?	https://www.modpagespeed.com/doc/release_notes
Handlebars	249,598	32,220	217,378	http://handlebarsjs.com	?	https://github.com/handlebars-lang/handlebars.js/blob/master/release-notes.md
YUI	201,973	486	201,487	http://yuilibrary.com	?	https://github.com/yui/yui3/releases
Drupal	256,421	64,326	192,095	https://drupal.org	7	https://www.drupal.org/psa-2019-02-25
MooTools	190,400	2,356	188,044	https://mootools.net	?	https://mootools.net/blog/category/releases/page/1
ZURB Foundation	273,337	90,398	182,939	http://foundation.zurb.com	?	https://get.foundation/sites/docs/#
Backbone.js	178,970	2,540	176,430	http://backbonejs.org	?	https://backbonejs.org/
All in One SEO Pack	170,116	35	170,081	https://wordpress.org/plugins/all-in-one-seo-pack/	?	https://aioseo.com/changelog/
Elementor	275,007	110,287	164,720	https://elementor.com	?	https://elementor.com/pro/changelog/
AngularJS	170,870	8,261	162,609	https://angularjs.org	1.7	https://blog.angular.io/stable-angularjs-and-long-term-support-7e077635ee9c
Mustache	160,656	1,263	159,393	https://mustache.github.io	?	https://openbase.com/js/mustache/versions
Prototype	151,881	8,485	143,396	http://www.prototypejs.org	?	http://prototypejs.org/download/
Vue.js	249,087	127,606	121,481	https://vuejs.org	2	https://forum.vuejs.org/t/vue-1-x-end-of-life-support/58143

React	501,584	402,192	99,392	https://reactjs.org	?	https://reactjs.org/versions/
React	501,584	402,192	99,392	https://reactjs.org	?	https://reactjs.org/versions/
jQuery Mobile	105,150	6,933	98,217	https://jquerymobile.com	?	https://jquerymobile.com/changelog/
Varnish	307,088	215,982	91,106	http://www.varnish-cache.org	6.0.8	https://varnish-cache.org/releases/
VideoJS	139,619	58,869	80,750	http://videojs.com	?	https://github.com/videojs/Video.js/releases
Leaflet	79,227	1,093	78,134	http://leafletjs.com	?	https://github.com/Leaflet/Leaflet/releases
NextGEN Gallery	78,889	799	78,090	https://www.imagely.com/wordpress-gallery-plugin	2	https://www.imagely.com/docs/shortcodes/
Dojo	83,516	8,379	75,137	https://dojotoolkit.org	?	https://dojotoolkit.org/reference-guide/1.9/releasesnotes/index.html
Fingerprintjs	86,871	16,625	70,246	https://valve.github.io/fingerprintjs2/	?	https://github.com/fingerprintjs/fingerprintjs/releases
Gravity Forms	109,959	42,565	67,394	http://gravityforms.com	?	https://docs.gravityforms.com/gravityforms-change-log/
Slick	759,805	693,556	66,249	https://kenwheeler.github.io/slick	?	https://github.com/slick/slick/releases
Firebase	64,037	851	63,186	https://firebase.com	?	https://firebase.google.com/support/releases
Stripe	63,812	6,400	57,412	http://stripe.com	?	https://support.fivetrans.com/hc/en-us/articles/360061749154-Stripe-Release-Notes
Magento	118,689	66,799	51,890	https://magento.com	?	https://devdocs.magento.com/recommendations/release-notes.html

WP-Statistics	50,635		50,635	https://wp-statistics.com	?	https://github.com/wp-statistics/wp-statistics/releases
OpenResty	258,293	208,454	49,839	http://openresty.org	?	https://openresty.org/en/changes.html
Angular	50,196	1,230	48,966	https://angular.io	10	https://angular.io/guide/releases#support-policy-and-schedule
W3 Total Cache	220,179	172,179	48,000	http://www.w3-edge.com/wordpress-plugins/w3-total-cache	?	https://github.com/szepeviktork/w3-total-cache-fixed/releases
Knockout.js	44,595	5	44,590	http://knockoutjs.com	?	https://github.com/knockout/releases
Raphael	37,091	147	36,944	https://dmitrybaranovskiy.github.io/raphael/	?	https://github.com/DmitryBaranovskiy/raphael/releases
Raphael	37,091	147	36,944	https://dmitrybaranovskiy.github.io/raphael/	?	https://github.com/DmitryBaranovskiy/raphael/releases
Apache Tomcat	46,424	11,498	34,926	http://tomcat.apache.org	8.5	http://tomcat.apache.org/whichversion.html
mod_ssl	33,928	22	33,906	http://modssl.org	?	http://www.modssl.org/
script.aculo.us	77,344	45,988	31,356	https://script.aculo.us	?	https://github.com/madrobby/scriptaculous/blob/v1.9.0/CHANGELOG
Flywheel	31,288		31,288	https://getflywheel.com/	?	https://docs.flywheel.io/hc/en-us/sections/360002865234-Release-Notes
SoundManager	31,075	2,117	28,958	http://www.schillmania.com/projects/soundmanager2	?	https://github.com/nicklockwood/SoundManager

Sentry	421,294	392,442	28,852	https://sentry.io/	?	https://docs.datafabric.hpe.com/62/EcosystemRN/SentryRN.html
TinyMCE	29,228	747	28,481	http://tinymce.com	5.3	https://www.tiny.cloud/docs/general-configuration-guide/system-requirements/
Highcharts	27,438	279	27,159	https://www.highcharts.com	?	https://www.highcharts.com/blog/changelog/
Snap.svg	26,460	75	26,385	http://snapsvg.io/	?	https://github.com/adobe-webplatform/Snap.svg/blob/master/history.md
Gatsby	27,922	1,552	26,370	https://www.gatsbyjs.org/	3	https://github.com/gatsbyjs/gatsby#contributing-to-gatsby-v1
Gatsby	27,922	1,552	26,370	https://www.gatsbyjs.org/	3	https://github.com/gatsbyjs/gatsby#contributing-to-gatsby-v1
D3	27,147	1,184	25,963	http://d3js.org	?	https://github.com/d3/d3/releases
MediaWiki	25,549	246	25,303	https://www.mediawiki.org	?	https://www.mediawiki.org/wiki/Release_notes
MediaWiki	25,549	246	25,303	https://www.mediawiki.org	?	https://www.mediawiki.org/wiki/Release_notes
CKEditor	47,426	22,382	25,044	http://ckeditor.com	?	https://ckeditor.com/ckeditor-4/download/releases/
XRegExp	24,799	53	24,746	http://xregexp.com	?	https://github.com/slevithan/xregexp/releases
Joomla	244,033	220,989	23,044	https://www.joomla.org	3.10	https://docs.joomla.org/Joomla!_CMS_versions
Phusion Passenger	35,462	12,634	22,828	https://phusionpassenger.com	?	https://github.com/phusion/passenger/releases

Kendo UI	22,212	778	21,434	https://www.telerik.com/kendo-ui	?	https://www.telerik.com/support/whats-new/kendo-ui/release-history
Duda	22,241	1,627	20,614	https://www.duda.co/website-builder	?	
Ember.js	20,458	32	20,426	http://emberjs.com	3.20	https://emberjs.com/releases/lts
FrontPage	21,073	815	20,258	http://office.microsoft.com/frontpage	?	https://microsoft.fandom.com/wiki/Microsoft_FrontPage
Marionette.js	20,745	1,452	19,293	https://marionettejs.com/	?	https://github.com/marionettejs/backbone.marionette/releases
GoDaddy Website Builder	#REF!		18,616	https://id.godaddy.com/websites/website-builder	?	https://www.godaddy.com/garage/hubupdates/
MathJax	18,738	336	18,402	https://www.mathjax.org/	?	https://github.com/mathjax/MathJax/releases?after=2.7.8#:~:text=dpvc%20released%20this%20on%20Sep%204%2C%202019%20After,techniques%2C%20including%20Typescript%2C%20ES6%20modules%2C%20Promises%2C%20and%20more.
Shopware	16,687	114	16,573	https://www.shopware.com/en/	?	https://www.emizentech.com/blog/shopware-6-4-3-1-released.html#:~:text=August%2025%2C%202021%202%20Mins%20Read%20In%20August%2C,new%20features.%20It%20requires%20Shopware%206.3.0.0%20or%20newer.
CodeMirror	16,677	404	16,273	https://codemirror.net/	?	https://github.com/felixhuse/codemirror-addon

DNN	20,994	4,928	16,066	http://dnnssoftware.com	8.5	https://www.dnnsoftware.com/docs/developers/product-versions.html
mod_dav	14,398	5	14,393	http://webdav.org/mod_dav	?	http://webdav.org/mod_dav/
Python	374,597	360,531	14,066	http://python.org	3.6	https://www.python.org/downloads/
mod_perl	13,220	5	13,215	http://perl.apache.org	?	https://perl.apache.org/download/
Algolia	20,096	7,084	13,012	http://www.algolia.com	?	https://github.com/algolia/algoliasearch-client-javascript/releases
Perl	17,085	5,113	11,972	http://perl.org	5.32	http://www.cpan.org/src/
Microsoft SharePoint	14,419	3,017	11,402	https://www.microsoft.com/id-id/microsoft-365/sharepoint/collaboration?ms.officeurl=sharepoint&rtc=1	?	https://docs.microsoft.com/en-us/officeupdates/sharepoint-updates
Outbrain	11,349	52	11,297	https://www.outbrain.com	?	https://support.fivetran.com/hc/en-us/articles/1500003153781-Outbrain-Release-Notes
mod_jk	11,308	137	11,171	http://tomcat.apache.org/tomcat-3.3-doc/mod_jk-howto.html	?	https://archive.apache.org/dist/tomcat/tomcat-connectors/
Debian	189,721	178,767	10,954	https://debian.org	?	https://www.debian.org/releases/
EasyEngine	10,054		10,054	https://easyengine.io	?	https://github.com/EasyEngine/easyengine/releases

vBulletin	10,238	1,050	9,188	https://www.vbulletin.com/	?	https://enxf.net/resources/vbulletin-v5-5-6-connect=vbulletin-v5=connect-enxf-nulled.1178/updates
Sitefinity	9,121	2	9,119	https://www.progress.com/sitefinity-cms	?	https://www.progress.com/sitefinity-cms/release-notes#:~:text=Sitefinity%20CMS%2013.0.7300%20%280official,Release%29%20May%2013%2C%202020.
three.js	9,540	459	9,081	https://threejs.org/	?	https://github.com/mrdoob/three.js/releases
Concrete5	10,842	2,100	8,742	https://www.concretecms.com/	?	https://www.concretecms.com/about/blog/core-releases
Liferay	14,281	5,690	8,591	https://www.liferay.com/	?	https://github.com/liferay/liferay-portal/releases
JavaServer Pages	8,459	18	8,441	https://www.oracle.com/java/technologies/jspt.html	?	https://jcp.org/aboutJava/communityprocess/maintenance/jsr245/245-MR2_1.html
Meteor	8,888	460	8,428	https://www.meteor.com/	?	https://docs.meteor.com/changelog.html
Java Servlet	10,267	1,963	8,304	https://www.oracle.com/java/technologies/java-servlet-tec.html	?	
PDF.js	11,055	2,935	8,120	https://mozilla.github.io/pdf.js/	?	https://github.com/mozilla/pdf.js/releases
mod_wsgi	7,840	2	7,838	https://code.google.com/p/modwsgi	?	https://github.com/GrahamDumpleton/mod_wsgi/releases
MyWebsite	7,789		7,789	https://www.ionos.com	?	
ExtJS	7,690	374	7,316	https://www.sencha.com	?	http://api.cenboomh.com/extjs/release-notes.html

SPiP	8,340	1,161	7,179	https://www.spip.net/en_rubrique25.html	https://www.spip.net/en_article6499.html	4
Hugo	6,793	16	6,777	http://gohugo.io	https://github.com/gohugoio/hugo/releases	?
Braintree	7,280	561	6,719	https://www.braintreepayments.com/	https://www.drupal.org/project/commerce_braintree/releases	?
Volusion	6,395	4	6,391	https://www.volusion.com/	https://www.volusion.com/v1-release-notes/index.html	?
Open Journal Systems	6,297	7	6,290	https://pkp.sfu.ca/ojs/		?
Dynamicweb	6,165		6,165	https://www.dynamicweb.dk/	https://doc.dynamicweb.com/downloads/releases/release-notes/dw-9-10-release-notes	?
Chart.js	233,052	227,016	6,036	https://www.chartjs.org	https://github.com/chartjs/Chart.js/releases	?
Apache Traffic Server	13,233	7,294	5,939	http://trafficserver.apache.org/	https://docs.trafficserver.apache.org/en/latest/release-notes/roadmap.en.html	7.1
Microsoft HT-TPAPI	5,803		5,803	https://www.microsoft.com/id-id/		?
Supersized	9,853	4,055	5,798	http://buildinternet.com/project/supersized	https://github.com/buildinternet/supersized	?
mod_python	5,253		5,253	http://www.modpython.org	http://modpython.org/	?
Infusionsoft	4,814	3	4,811	https://keap.com/	https://help.infusionsoft.com/help/release-notes	?
gunicorn	4,593	11	4,582	https://gunicorn.org/	https://github.com/benoitcgunicorn/releases	?
Mobirise	4,272		4,272	https://mobirise.com/	https://mobirise.com/history.html	?

WebSite X5	4,261		4,261	https://www.websitex5.com/en/	?	https://www.websitex5.com/changelog/v2019.1/en.php?ed=&_ga=2.8172977.285220766.1636086202-530522610.1634454942
Discourse	3,997	14	3,983	https://discourse.org	?	https://meta.discourse.org/c/releases/30
Next.js	55,570	51,606	3,964	https://nextjs.org	?	https://github.com/vercel/next.js/releases
Socket.io	37,349	33,459	3,890	https://socket.io	?	https://github.com/socketio/socket.io/releases
Apollo	3,751	2	3,749	https://www.apollographql.com	?	https://github.com/ApolloAuto/apollo/releases
Nette Framework	18,202	14,464	3,738	https://nette.org/	?	https://files.nette.org/releases/
OpenLayers	6,527	2,912	3,615	https://openlayers.org/	?	https://github.com/openlayers/openlayers/releases/
Jekyll	4,873	1,511	3,362	http://jekyllrb.com/	?	https://jekyllrb.com/news/releases/
WP Rocket	214,557	211,230	3,327	https://wp-rocket.me/	?	https://wp-rocket.me/changelog/
Microsoft Word	25,390	22,079	3,311	https://www.microsoft.com/id-id/microsoft-365/word?legRedir=true&CorrelationId=bfde8a81-96e3-403d-ab78-3c3e7b0c9e3d&rtc=1	?	https://docs.microsoft.com/en-us/officeupdates/current-channel
AMP Plugin	3,102		3,102	https://amp-wp.org/	?	https://amp-wp.org/category/amp-plugin/
NVD3	3,189	124	3,065	https://nvd3.org/	?	https://github.com/novus/nvd3/releases
lighttpd	3,590	654	2,936	http://www.lighttpd.net/	1.4.54	https://repology.org/project/lighttpd/versions

Plyr	27,608	24,711	2,897	https://plyr.io/	?	https://github.com/sampotts/plyr/blob/master/CHANGELOG.md
Ghost	2,927	51	2,876	https://ghost.org/	?	https://github.com/TryGhost/Ghost/releases
SMF	2,812	2	2,810	https://simplemachines.org/	?	https://github.com/SimpleMachines/SMF2.1/releases
Discuz! X	4,657	1,986	2,671	https://www.discuz.net/	?	https://gitee.com/Discuz/DiscuzX/releases
Google Maps	579,317	576,804	2,513	http://maps.google.com	?	https://developers.google.com/maps/documentation/javascript/releases
Welcart	2,693	206	2,487	https://www.welcart.com/	?	
Polymer	2,393	23	2,370	http://polymer-project.org	?	https://github.com/Polymer/polymer/releases
Atlassian Jira	2,103	57	2,046	https://www.atlassian.com/software/jira	?	https://marketplace.atlassian.com/apps/1213607/jira-software/version-history
Mura CMS	1,982		1,982	https://www.murasoftware.com/	?	https://docs.murasoftware.com/v10/release-notes/
JBoss Application Server	2,160	230	1,930	https://www.jboss.org/jbossas.html	?	https://jbossas.jboss.org/downloads/
jQuery-pjax	22,688	20,786	1,902	https://github.com/defunkt/jquery-pjax	?	https://github.com/defunkt/jquery-pjax/tags
Resin	1,875		1,875	https://caucho.com/	?	https://caucho.com/products/resin/download
Jetty	1,955	117	1,838	http://www.eclipse.org/jetty/	9.4	https://www.eclipse.org/jetty/download.php
amCharts	6,021	4,320	1,701	http://amcharts.com	4.0	https://www.amcharts.com/versions/
Hogan.js	16,334	14,691	1,643	https://twitter.github.io/hogan.js/	?	https://support.lumary.com/hc/en-us/articles/360023301851-Hogan-release-notes

mod_fastcgi	4,049	2,408	1,641	http://www.fastcgi.com/mod_fastcgi/docs/mod_fastcgi.html	?	http://freshmeat.sourceforge.net/projects/mod_fastcgi/releases
Oracle Application Server	1,616	24	1,592	https://www.oracle.com/middleware/technologies/internet-application-server.html	?	https://support.oracle.com/knowledge/Middleware/397022_1.html
Material Design Lite	36,380	34,867	1,513	https://getmdl.io/	?	https://github.com/google/material-design-lite/releases
JBoss Web	1,505		1,505	https://jbossweb.jboss.org/	?	https://jbossas.jboss.org/downloads/
KineticJS	5,084	3,614	1,470	https://github.com/ericdrowell/KineticJS/	?	https://github.com/ericdrowell/KineticJS/releases
DreamWeaver	159,256	157,813	1,443	https://www.adobe.com/products/dreamweaver.html	?	https://helpx.adobe.com/dreamweaver/dreamweaver-releasenotes.html
Miva	1,490	100	1,390	https://www.miva.com/	?	https://www.miva.com/template-changes
Highstock	1,334		1,334	https://www.highcharts.com/blog/products/stock/	?	https://www.highcharts.com/blog/changelog/
JavaServer Faces	1,324	4	1,320	https://javaee.github.io/javaxserverfaces-spec/	?	https://www.javatpoint.com/what-is-jsf
Flask	1,246		1,246	https://flask.palletsprojects.com/en/2.0.x/	?	https://github.com/pallets/flask/releases
Angular Material	10,281	9,040	1,241	https://material.angularjs.org/latest/	?	https://github.com/angular/material/blob/master/CHANGELOG.md
Zend	1,389	171	1,218	https://www.zend.com/	?	https://www.zend.com/release-notes/zend-server
Sensors Data	3,146	1,932	1,214	https://www.sensorsdata.cn/auto	?	

Hexo	1,378	166	1,212	https://hexo.io/	?	https://github.com/hexojs/hexo/releases
Outlook Web App	3,205	2,091	1,114	https://support.microsoft.com/en-us/outlook?ui=en-us&rs=en-us&ad=us	?	https://docs.microsoft.com/en-us/officeupdates/current-channel
hCaptcha	1,122	21	1,101	https://www.hcaptcha.com/	?	https://github.com/hcaptcha/hcaptcha-wordpress-plugin/releases
Adobe GoLive	1,913	823	1,090	https://www.adobe.com/products/golive	?	
Alpine.js	1,695	613	1,082	https://github.com/alpinejs/alpine	?	https://github.com/alpinejs/alpine/releases
Umbraco	1,368	365	1,003	https://umbraco.com/	?	https://umbraco.com/products/knowledge-center/versioning-and-release-cadence/
Neos Flow	977	6	971	https://flow.neos.io/	?	https://github.com/neos/flow-development-collection/releases
Bluefish	926	17	909	https://sourceforge.net/projects/bluefish/	?	https://bluefish.openoffice.nl/index.html
EPrints	849	13	836	https://www.eprints.org/uk/	?	https://wiki.eprints.org/wiki/Manual
Neos CMS	873	39	834	https://www.neos.io/	4.3	https://www.neos.io/features/release-process.html
CodeIgniter	818		818	http://codeigniter.com	?	https://www.codeigniter.com/download#:~:text=CodeIgniter%20has%20two%20supported%20versions%3A%20CodeIgniter%204%20%28current%29,is%20ongoing%2C%20and%20the%20current%20version%20is%20v4.1.4.

Ionic	2,203	1,432	771	https://ionicframework.com/	?	https://ionicframework.com/docs/reference/release-notes
HHVM	669	10	659	https://hhvm.com/	4.102	https://docs.hhvm.com/hhvm/installation/release-schedule
RockRMS	657		657	https://www.rockrms.com/	?	https://www.rockrms.com/releasesnotes?version
ADPLAN	1,003	383	620	https://adplan.gafcodes.com/	?	
GlassFish	1,676	1,065	611	https://javaee.github.io/glassfish/	?	https://glassfish.org/download
iWeb	565	5	560	https://www.apple.com/mac/	?	
Contenido	559		559	https://www.contenido.org/en	?	
Gridsome	558		558	https://gridsome.org/	?	https://gridsome.org/blog/
Koha	580	26	554	https://koha-community.org/	?	https://git.koha-community.org/Koha-community/koha-release-notes
Sulu	535		535	https://sulu.io/	?	https://sulu.io/know-how/blog
Plotly	521		521	https://plotly.com/javascript/	?	https://github.com/plotly/plotly.py/releases
Webdev	561	49	512	https://windev.com/webdev/index.html	?	https://web-develop.ca/index.php?board=48.0
Adyen	509		509	https://www.adyen.com/	?	https://docs.adyen.com/online-payments/release-notes
Oracle Web Cache	519	10	509	https://www.oracle.com/index.html	?	https://opensolution.org/download/quick-cms-and-quick-cms-ext=other.html
Chamilo	474		474	https://chamilo.org/en/	?	https://www.radiotallercepra.org/campus/documentation/changelog.html
Vaadin	592	128	464	https://vaadin.com/	?	https://vaadin.com/releases/

MkDocs	438		438	https://www.mkdocs.org/	?	https://www.mkdocs.org/about/release-notes/
GrowingIO	434		434	https://www.growingio.com/	?	https://docs.growingio.com/op/developer-manual/sdkintegrated/mp/gtouchsdk-releasenotes
Chevereto	402	3	399	https://chevereto.com/	?	https://releases.chevereto.com/3.X/3.20/3.20.12.html#links
imperia CMS	450	64	386	https://www.pirobase-imperia.com/de/produkte/produktuebersicht/imperia-cms	?	http://imperiamcms.com/release-notes/
Contensis	330		330	https://www.contensis.com/	?	https://github.com/zengenti/contensis-react-base
Sarka-SPIP	329	1	328	http://www.sarka-spip.net/	4.0	https://www.spip.net/en_article6499.html
Phaser	329	2	327	https://phaser.io/	?	https://github.com/photonstorm/phaser/releases
GitBook	359	32	327	https://www.gitbook.com/	?	
govCMS	323		323	https://www.govcms.gov.au/	?	https://github.com/govCMS/govCMS8/releases
GeneXus	356	38	318	https://www.genexus.com/en/	?	https://www.genexus.com/en/developers/downloadcenter
Sphinx	3,010	2,701	309	https://www.sphinx-doc.org/en/master/	?	https://github.com/sphinx/releases
A-Frame	302	12	290	https://aframe.io/	?	https://github.com/aframevr/aframe/releases
Transifex	289		289	https://www.transifex.com/	?	https://github.com/transifex/transifex-client/releases
Ckan	301	13	288	https://ckan.org/	?	https://github.com/KSP-CKAN/CKAN/releases

TornadoServer	286		286	https://www.tornadoweb.org/en/stable/	?	https://www.tornadoweb.org/en/stable/releases.html
FlexCMP	293	10	283	https://www.flexcmp.com/dxp	?	
Catberry.js	280	4	276	https://catberry.github.io/	?	https://github.com/catberry/catberry/releases
Oracle Commerce Cloud	255	2	253	https://cloud.oracle.com/commerce-cloud	?	https://docs.adyen.com/plugins/oracle-commerce-cloud/release-notes
Akka HTTP	226		226	https://akka.io/	?	https://doc.akka.io/docs/akka-http/current/release-notes/index.html
X-Cart	1,065	842	223	https://www.x-cart.com/	?	https://devs.x-cart.com/changelog/
Typecho	227	7	220	http://typecho.org/	?	https://github.com/typecho/typecho/releases
OpenUI5	219		219	https://openui5.org/	?	https://openui5.org/releases/
phpwind	344	142	202	https://www.phpwind.net/	?	
Highlight.js	35,637	35,437	200	https://highlightjs.org/	?	https://github.com/highlightjs/highlight.js/releases
CherryPy	194	2	192	https://www.cherrypy.org/	?	https://github.com/cherrypy/cherrypy/tags
CMSimple	254	62	192	https://www.cmsimple.org/en/	?	http://freshmeat.sourceforge.net/projects/cmsimple/releases
Kooboo CMS	247	56	191	https://www.kooboo.com/	?	https://github.com/Tsingbo-Kooboo/KoobooMvc5/releases
mod_auth_pam	197	20	177	http://pam.sourceforge.net/mod_auth_pam/	?	http://pam.sourceforge.net/mod_auth_pam/download.html
Coppermine	174		174	https://coppermine-gallery.net/	?	https://github.com/coppermine-gallery/cpg1.6.x/releases

Serendipity	162	8	154	https://docs.s9y.org/	?	https://github.com/s9y/Serendipity/releases
Elm	150		150	https://elm-lang.org/	?	https://github.com/elm/compiler/releases
MochiKit	171	32	139	https://mochi.github.io/mochikit/	?	https://mochi.github.io/mochikit/doc/html/MochiKit/index.html
FreeBSD	14,647	14,517	130	https://www.freebsd.org/	12.2	https://www.freebsd.org/releases/
Quick.Cart	459	335	124	https://opensolution.org/home.html	?	https://opensolution.org/download/quick-cart-and-quick-cart-ext-other.html
Quick.CMS	443	321	122	https://opensolution.org/home.html	?	
thttpd	121	5	116	https://acme.com/software/thttpd/	?	https://www.acme.com/software/thttpd/
Dancer	114		114	https://perldancer.org/	?	
Danneo CMS	108		108	https://danneo.com/	?	
Cherokee	132	26	106	http://cherokee-project.com/	?	http://cherokee-project.com/downloads.html
MoinMoin	154	50	104	https://moinmo.in/	?	https://github.com/moinwiki/moin-1.9/blob/1.9.11/docs/CHANGES#L13
Microsoft Publisher	97	3	94	https://www.microsoft.com/id-id/microsoft-365/publisher?legRedirect=true&CorrelationId=1f1413ff-e9cc-4b09-b58f=3b46df35f0f7&rtc=1	?	https://en.wikipedia.org/wiki/Microsoft_Publisher#Release_history
Kentico CMS	13,969	13,877	92	https://www.kentico.com/index.html	?	https://xperience.io/product/roadmap/version-history

Clipboard.js	89,517	89,426	91	https://clipboardjs.com/	?	https://github.com/zenorocha/clipboard.js/releases
JavaScript In- favis Toolkit	283	197	86	https://philob.github.io/jit/	?	
WebGUI	84		84	http://www.webgui.org/	?	https://blog.webgui.org/
VuePress	84		84	https://vuepress.vuejs.org/	?	https://github.com/vuejs/vuepress/releases
GoJS	76		76	https://gojs.net/latest/index.html	?	https://github.com/NorthwoodsSoftware/GoJS/releases?after=v2.1.27
Wink	21,296	21,222	74	http://ww25.winktoolkit.org/?subid=20211017-1942-127b-942c-c06d160ea89a	?	https://github.com/chamilo/chamilo-lms/releases
Trac	83	16	67	https://trac.edgewall.org/	?	https://trac.edgewall.org/wiki/TracDownload
IBM HTTP Server	793	728	65	https://www.ibm.com/cloud/websphere-application-server	?	https://www.ibm.com/docs/en/ibm-http-server
AOLserver	63		63	https://www.aol.com/	?	http://aolserver.sourceforge.net/category/releases/
Lift	61		61	https://liftweb.net/	?	https://liftweb.net/
Reveal.js	1,123	1,064	59	https://revealjs.com/	?	https://github.com/hakimel/reveal.js/releases
jComponent	56		56	https://componentator.com/	?	https://docs.totaljs.com/components/
Microsoft Excel	76	20	56	https://www.microsoft.com/id-id/microsoft-365/excel?legRedirect=true&CorrelationId=c9040e26-c5d7-41a0-8a12-98f3b77df4a6&rtc=1	?	https://docs.microsoft.com/en-us/officeupdates/current-channel

ReDoc	53		53	https://github.com/Redocly/redoc	?	https://github.com/Redocly/redoc
Shopfa	52		52	https://shopfa.com/	?	
Aegea	50		50	https://blogengine.ru/	?	
Hiawatha	49		49	https://www.hiawatha-webserver.org/about	?	https://www.hiawatha-webserver.org/about
Backdrop	716	667	49	https://backdropcms.org/	?	https://github.com/backdrop/backdrop/releases#:~:text=The%20Backdrop%20community%20is%20proud%20to%20release%20version,Migrate%20to%201.x-1.0.17%20along%20with%20this%20core%20update.
decimal.js	2,793	2,744	49	https://mikemcl.github.io/decimal.js/	?	https://www.npmjs.com/package/decimal.js/v/3.0.0
Adobe RoboHelp	63	15	48	https://www.adobe.com/products/robohelp.html	?	https://helpx.adobe.com/robohelp/using/rh_system_requirements.html#RoboHelp
Warp	46		46	https://www.stackage.org/package/warp	?	https://www.stackage.org/package/warp
AsciiDoc	51	6	45	http://www.methods.co.nz/asciidoc	?	https://github.com/asciidoctor/asciidoctor/releases
Koken	43		43	https://www.koken.me/	?	https://www.koken.me/
Atlassian Bitbucket	46	3	43	https://www.atlassian.com/software/bitbucket	?	https://developer.atlassian.com/server/bitbucket/reference/api-changelog/
MODX	8,319	8,277	42	https://modx.com/	?	https://modx.com/blog/category/release/
PHP-Fusion	408	369	39	https://www.php-fusion.co.uk/home.php	?	https://www.php-fusion.co.uk/infusions/downloads/downloads.php?type=recent

OroCommerce	58	22	36	https://oroinc.com/	?	https://github.com/oroinc/orocommerce-application/releases
Doxygen	49	14	35	https://www.doxygen.nl/index.html	?	https://www.doxygen.nl/manual/changelog.html
TwistedWeb	236	202	34	https://twistedmatrix.com/trac/wiki/TwistedWeb	?	https://pypi.org/project/Twisted/#history
Invenio	33		33	https://invenio-software.org/	?	https://invenio-software.org/blog/
DokuWiki	1,361	1,328	33	https://www.dokuwiki.org/dokuwiki	?	https://www.dokuwiki.org/devel:releases#releases
Roadiz CMS	50	18	32	https://www.roadiz.io/	?	https://github.com/roadiz/releases
Semantic-ui	9,749	9,717	32	https://semantic-ui.com/	?	https://github.com/Semantic-UI/releases
Coaster CMS	30		30	https://www.coastercms.org/	?	https://www.coastercms.org/blog/
cgkit	30		30	https://git.zx2c4.com/cgkit	?	https://git.zx2c4.com/cgkit
TiddlyWiki	36	8	28	https://tiddlywiki.com/	?	https://github.com/Jermolene/TiddlyWiki5/releases
otrs	72	45	27	https://otrs.com/	?	https://otrs.com/overview-release-notes=security-advisories/release-notes/
Open Classifieds	27	2	25	https://open-classifieds.com/	?	
Gallery	112	87	25	http://galleryproject.org/	?	http://galleryproject.org/
MobX	22,987	22,963	24	https://mobx.js.org/README.html	?	https://github.com/mobxjs/mobx/releases

Bokeh	78	57	21	https://bokeh.org/	?	https://docs.bokeh.org/en/latest/docs/releases.html#release-2-4-0
AppDynamics	2,645	2,624	21	https://www.appdynamics.com/	?	https://docs.appdynamics.com/21.5/en/product-and-release-announcements/release-notes
BrowserCMS	20		20	http://browsercms.org/	?	
CoinHive	338	318	20	https://www.troyhunt.com/i-now-own-the-coinhive-domain-heres-how-im-fighting-cryptojacking-and-doing-good-things-with-content-security-policies/	?	
Halo	18		18	https://halo.run/	?	https://halo.run/blog.html
Atlassian Confluence	1,317	1,299	18	https://www.atlassian.com/	?	https://confluence.atlassian.com/doc/confluence-release-notes-327.html
Boa	17		17	https://github.com/boa-dev/boa	?	https://github.com/boa-dev/boa/blob/main/CHANGELOG.md
Immutable.js	14,166	14,149	17	https://immutable-js.com/	?	https://github.com/immutable-js/releases
Fusion Ads	22	8	14	http://fusionads.net/	?	https://wpfusion.com/documentation/faq/changelog/
InfernoJS	35	21	14	https://www.infernojs.org/	?	https://github.com/infernojs/inferno/releases
Tessitura	13		13	https://www.tessituranetwork.com/	?	
Yaws	53	40	13	http://yaws.hyber.org/	?	https://github.com/erlyaws/yaws/releases

GX WebManager	80	67	13	https://www.gxsoftware.com/en/products/web-content-management.htm	?	
wisyCMS	12		12	https://wisy.3we.de/home-15-de	?	https://wisy.3we.de/change-log-225-de
SimpleHTTP	16	4	12	http://simple-http.com/	?	https://repo.maven.apache.org/maven2/com/simple-http/simple-http/
ImpressPages	729	717	12	https://www.impresspages.org/	?	https://www.impresspages.org/blog2
Public CMS	11		11	http://www.publiccms.com/	?	http://www.publiccms.com/download.html
Virtuoso	14	3	11	https://virtuoso.openlinksw.com/	?	https://github.com/openlink/virtuoso-opensource/releases
Bugzilla	18	7	11	https://www.bugzilla.org/	?	https://www.bugzilla.org/download/
phpSQLiteCMS	10		10	https://phpsqlitecms.hoschek.com/	?	https://github.com/ilosuna/phpsqlitecms
Blade	10		10	https://lets-blade.com/	?	
LinkSmart	11	2	9	https://www.viglink.com/linksmart/	?	
JAlbum	14	5	9	https://jalbum.net/en/	?	https://jalbum.net/en/software/release-notes
uKnowva	18	9	9	https://uknowva.com/	?	https://docs.uknowva.com/release-notes
Graffiti CMS	8		8	https://archive.codeplex.com/?p=graffiticms	?	https://github.com/motoyugota/GraffitiCMS/releases
Saber	7		7	https://saber.land/	?	https://forcivity.com/solutions/applications/saber-component-pack-for-lightning/release-notes-saber/

Kohana	7		7	http://kohanaframework.org/	?	https://github.com/kohana/kohana/releases
phpMyAdmin	11	4	7	https://www.phpmyadmin.net/	?	https://www.phpmyadmin.net/downloads/
SquirrelMail	96	89	7	https://squirrelmail.org/	?	https://www.squirrelmail.org/download.php
Livefyre	505	498	7	https://business.adobe.com/products/experience-manager/sites/aem-sites.html	?	https://experienceleague.adobe.com/docs/livefyre/using/release-notes/c-rn.html?lang=en
Exhibit	1,165	1,158	7	http://simile-widgets.org/exhibit/	?	https://github.com/simile-widgets/exhibit/tags
mini_httpd	6		6	http://acme.com/software/mini_httpd/	?	http://acme.com/software/mini_httpd/
DirectAdmin	6		6	https://www.directadmin.com/	?	https://www.directadmin.com/versions.php
CppCMS	6		6	http://cppcms.com/wikipp/en/page/main	?	http://cppcms.com/wikipp/en/page/releases
BaseHTTP	6		6		?	
Artifactory Web Server	6		6	https://jfrog.com/open-source/#os-arti	?	https://www.jfrog.com/confluence/display/JFROG/Artifactory+Release+Notes
Gitea	28	23	5	https://gitea.io/en-us/	?	https://blog.gitea.io/
Planet	640	635	5	http://planetplanet.org/	?	https://developers.planet.com/changelog/
Blessing Skin	4		4	https://github.com/bs-community/blessing-skin-server	?	https://github.com/prinsss/blessing-skin-server/releases
Indy	10	6	4	https://www.indyproject.org/	?	https://www.indyproject.org/documentation/
Oracle Commerce	2,641	2,637	4	https://www.oracle.com/cx/	?	

Django	28,506	28,502	4	https://djangoproject.com	?	https://docs.djangoproject.com/en/3.2/releases/
Gogs	13	10	3	https://gogs.io/	?	https://github.com/gogs/gogs/releases
ef.js	72	69	3	https://ef.js.org/#!home	?	https://github.com/TheNeuronProject/ef.js/releases
Snap	2		2	http://snapframework.com/	?	http://snapframework.com/blog
Scenari	2		2	https://github.com/zoncoen/scenarigo	?	https://github.com/zoncoen/scenarigo/releases
MochiWeb	2		2	https://github.com/mochi/mochiweb	?	https://github.com/mochi/mochiweb/releases
Laterpay	2		2	https://www2.laterpay.net/	?	
Jenkins	2		2	https://www.jenkins.io/	?	https://www.jenkins.io/changelog-stable/
Koobi	4	2	2	http://dream4.de/cms	?	
Analysys Ark	65	63	2	https://www.analysysdata.com/	?	
OpenGrok	1		1	http://hub.opensolaris.org/bin/view/Project+opengrok/WebHome	?	https://github.com/opengrok/releases
gitweb	1		1	http://git-scm.com/	?	https://github.com/yoannfleurydev/gitweb/releases
gitlist	1		1	http://gitlist.org/	?	https://github.com/klaussilveira/gitlist/releases
Cecil	1		1	https://cecil.app/	?	https://github.com/jbevain/cecil/releases
Xitami	2	1	1	http://www.xitami.com/	?	https://imatix-legacy.github.io/xitami.com/download.htm
FreeTextBox	3	2	1			

Microsoft PowerPoint	12	11	1	https://www.microsoft.com/id-id/microsoft-365/powerpoint?legRedirect=true&CorrelationId=9f3cb8b1-7d9b-4064-8cad=cbc7e6a111c1&rtc=1	?	https://docs.microsoft.com/en-us/officeupdates/current-channel
git	105	104	1	http://git-scm.com/	?	https://mirrors.edge.kernel.org/pub/software/scm/git/
Lua	258,294	258,293	1	http://www.lua.org/	?	https://www.lua.org/versions.html
MediaElement.js	2	2		http://www.mediaelementjs.com	?	https://github.com/mediaelement/releases
Raphael	2	2		https://dmitrybaranovskiy.github.io/raphael/	?	https://github.com/DmitryBaranovskiy/raphael/releases
MediaWiki	5	5		https://www.mediawiki.org	?	https://www.mediawiki.org/wiki/Release_notes
Gatsby	4	4		https://www.gatsbyjs.org/	3	https://github.com/gatsbyjs/gatsby#contributing-to-gatsby-v1
React	2	2		https://reactjs.org	?	https://reactjs.org/versions/
Bootstrap	10	10		https://getbootstrap.com	4	https://github.com/twbs/release
Underscore.js 1.8.2	1	1				
UIKit	1	1				
TYPO3CMS	1	1		https://typo3.org/		
Swiftlet	1	1				
Pars Elecom Portal	1	1		http://parselecom.com/		

Tabel A.1: App Data

LAMPIRAN B

KODE PROGRAM

Kode B.1: GetApplications.js

```
1 const {BigQuery} = require('@google-cloud/bigquery');
2 const options = {
3   keyFilename: 'gsm-bigquery-credentials.json',
4   projectId: 'httparchive-bigquery-346414',
5 };
6 const bigquery = new BigQuery(options)
7
8
9 async function getApplications(app = "Apache") {
10   const getAppSql = 'select app, info, count(app) as jumlah, result from httparchive-bigquery-346414.app_result.app_result where
11     app = "${app}" and (result != "NON CONCLUSIVE" and result != "NOT VERSIONED")
12   group by app, info, result order by info ASC'
13   const options = {
14     query: getAppSql,
15     location: 'US',
16   };
17   const [job] = await bigquery.createQueryJob(options);
18   const [rows] = await job.getQueryResults();
19   return rows.filter(item => !item.info.includes("\\"));
20 }
21
22 async function getApplicationsType(limit = 5, offset = 1) {
23   const getAppSql = 'select app from httparchive-bigquery-346414.Step.app_result where info != '' group by app limit ${limit}
24     offset ${offset}'
25   const options = {
26     query: getAppSql,
27     location: 'US',
28   };
29   const [job] = await bigquery.createQueryJob(options);
30   const [rows] = await job.getQueryResults();
31   return rows;
32 }
33
34 async function getApplicationsUrl(limit = 10, offset = 1) {
35   const getAppSql = 'select * from \'httparchive-bigquery-346414.URL_Result.url_result\' limit ${limit} offset ${offset} '
36   const options = {
37     query: getAppSql,
38     location: 'US',
39   };
40   const [job] = await bigquery.createQueryJob(options);
41   const [rows] = await job.getQueryResults();
42   return rows;
43 }
44
45 async function getPopularTech(limit = 10, offset = 1) {
46   const getAppSql = 'select * from \'httparchive-bigquery-346414.numsite_app_result_count.numsite_app_result_count\' limit ${
47     limit} offset ${offset} '
48   const options = {
49     query: getAppSql,
50     location: 'US',
51   };
52   const [job] = await bigquery.createQueryJob(options);
53   const [rows] = await job.getQueryResults();
54   return rows;
55 }
56 module.exports = {getApplications, getApplicationsType, getApplicationsUrl, getPopularTech}
```