

SKRIPSI

PENGUKURAN APLIKASI USANG DI HTTPARCHIVE



Vinson Tandra

NPM: 2016730042

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

UNDERGRADUATE THESIS

OBSOLETE APP MEASUREMENT IN HTTPARCHIVE



Vinson Tandra

NPM: 2016730042

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2022**

ABSTRAK

Di masa teknologi saat ini, banyak perusahaan yang menggunakan *website* sebagai tempat untuk mencari informasi. Terdapat banyak *website* yang menggunakan aplikasi yang sudah usang. HTTPArchive adalah sebuah proyek yang bersifat *open source* untuk melihat bagaimana *website* dibuat. Di dalam HTTPArchive terdapat data-data historis yang disediakan untuk menunjukkan bagaimana *website* terus berkembang dan proyek ini sering digunakan untuk penelitian.

Didalam HTTPArchive terdapat dataset yang berisi jutaan *web* setiap bulan dan dapat dianalisis menggunakan teknologi BigQuery. BigQuery adalah salah satu produk dari Google yang berbasis *cloud* dan dapat digunakan untuk menganalisis data. BigQuery menggunakan SQL (*Structured Query Language*) untuk mengakses data yang diberikan dengan melakukan *query*. Maka dari itu, data dari HTTPArchive dapat dianalisis untuk mendapatkan informasi versi dari setiap aplikasi yang masih didukung dan tidak serta mencari *website* yang sudah menggunakan aplikasi yang sudah didukung. Untuk mendapatkan data-data tersebut dapat melakukan *query* pada teknologi BigQuery.

Pada skripsi ini dibuat sebuah aplikasi yang menggunakan ReactJS untuk membuat tampilan dan NodeJS untuk pengambilan data. Langkah-langkah yang dilakukan adalah mengumpulkan daftar website, mencari aplikasi yang digunakan, mencari aplikasi yang digunakan website, mengelompokkan berdasar nama semua aplikasi yang dipakai, mencari data tentang versi aplikasi yang masih didukung, dan melakukan perbandingan antara versi aplikasi yang masih dipakai sekarang dengan aplikasi yang masih didukung (*version compare*). Setelah langkah tersebut dilakukan, didapatkan hasil yaitu pada aplikasi Nginx, untuk ruang lingkup di Indonesia versi 1.10 paling banyak digunakan url, sedangkan untuk ruang lingkup dunia versi 1.14 yang paling banyak digunakan. Adapun beberapa aplikasi yang ditambahkan pada aplikasi ini yaitu Python dan JQuery Migrate. Pada aplikasi Python jumlah versi yang paling banyak digunakan adalah versi 2.7 dan pada aplikasi JQuery Migrate versi aplikasi yang paling banyak digunakan adalah versi 1.4. Data yang dihasilkan menunjukkan bahwa terdapat 4.280 *website* yang menggunakan aplikasi yang semua aplikasinya masih didukung.

Kata-kata kunci: BigQuery, HTTPArchive, Structured Query Language, Node.js, React.js, NPM, Version Compare

ABSTRACT

In today's technological era, many companies use website as a place to find information. There are many websites that use outdated applications. HTTPArchive is a open source project to see how a website is created. HTTPArchive contains historical data provided to show how website continues to evolve and this project is often used for research.

HTTPArchive contains a dataset containing millions of web every month and can be analyzed using BigQuery technology. BigQuery is a product from Google which is based on cloud and can be used to analyze data. BigQuery uses SQL (Structured Query Language) to access the given data by doing query. Therefore, data from HTTPArchive can be analyzed to get version information of each application that is still supported and not and look for website that is already using a supported application. To get these data, you can do query on BigQuery technology.

In this thesis, an application is made that uses ReactJS to create views and NodeJS to retrieve data. The steps taken are to collect a list of websites, search for applications that are used, search for applications that are used by the website, group by name of all applications used, look for data on application versions that are still supported, and make comparisons between the versions of applications that are still in use today and the application. which is still supported version compare. After these steps were taken, the results were obtained, namely the Nginx application, for the scope of Indonesia version 1.10 the most widely used url, while for the world scope version 1.14 was the most widely used. There are several applications added to this application, namely Python and JQuery Migrate. In the Python application the most widely used version is version 2.7 and in the JQuery Migrate application the most widely used version of the application is version 1.4. The resulting data shows that there are 4,280 websites that use applications where all applications are still supported

Keywords: BigQuery, HTTPArchive, Structured Query Language, Node.js, React.js, NPM, Version Compare

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	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 BigQuery	5
2.1.1 <i>Cloud Storage System</i>	5
2.1.2 <i>SQL (Structured Query Language)</i> [1]	5
2.2 HTTPArchive	10
2.3 Pengukuran Aplikasi Usang Pada Beberapa Website Populer Di Indonesia[2]	10
2.3.1 <i>Research Method</i>	11
2.3.2 Hasil Keseluruhan	12
2.4 Node.js	13
2.4.1 <i>Node Package Manager</i>	14
2.4.2 NPM CLI	14
2.5 ReactJS	14
2.5.1 JSX	14
2.5.2 Melakukan <i>Render</i> Elemen	15
2.5.3 <i>Components and Props</i>	16
2.5.4 Penanganan Event	17
2.5.5 <i>Render</i> Bersyarat	18
2.5.6 <i>List</i> dan <i>Keys</i>	19
2.5.7 Form	20
2.5.8 Hooks	21
2.5.9 NPX	23
2.6 ChartJS	24
2.7 Express.js	24
3 PERCOBAAN AWAL	25
3.1 Eksplorasi Teknologi	25
3.2 <i>Dataset</i> yang Digunakan pada HTTPArchive	33
3.3 Langkah-Langkah Query Yang Dilakukan	33
3.3.1 Mengumpulkan Daftar Website	34
3.3.2 Mencari Aplikasi Yang Digunakan <i>Website</i>	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 <i>Sample Data Apache</i>	39
4	PENGGALIAN DATA	41
4.1	Langkah-Langkah <i>Query</i> Yang Dilakukan Dengan Data Yang Lebih Besar	41
4.1.1	Mengumpulkan <i>List Website</i>	41
4.1.2	Mencari Aplikasi Yang Digunakan <i>Website</i>	42
4.1.3	Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai	43
4.1.4	Mencari Data Tentang Versi Aplikasi Yang Masih Didukung	45
4.1.5	Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai Sekarang Dengan Versi Aplikasi Yang Masih Didukung	45
4.2	Hasil <i>Sample Data</i>	46
4.2.1	Apache dan Nginx	46
4.2.2	PHP dan Python	48
4.2.3	jQuery dan jQuery Migrate	50
5	PEMBANGUNAN PERANGKAT LUNAK	53
5.1	Analisis dan Perancangan Perangkat Lunak	53
5.1.1	UseCase	53
5.1.2	Perancangan Antarmuka	54
5.2	Implementasi Perangkat Lunak	58
5.2.1	BigQuery	59
5.2.2	React.js	64
5.2.3	Node.js	69
5.3	Pengujian Fungsional	71
5.3.1	Daftar Aplikasi	71
5.3.2	Daftar URL dengan Jumlah Hasil Perbandingan	74
5.3.3	Daftar Aplikasi yang Populer	75
6	KESIMPULAN DAN SARAN	77
6.1	Kesimpulan	77
6.2	Saran	77
DAFTAR REFERENSI		79
A	DATA APLIKASI	81
B	KODE PROGRAM	107

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	40
4.1	Aplikasi Nginx	47
4.2	Aplikasi Apache	47
4.3	Aplikasi Apache dari [2]	48
4.4	Aplikasi Nginx dari [2]	48
4.5	Aplikasi PHP	49
4.6	Aplikasi Python	49
4.7	Aplikasi PHP dari [2]	50
4.8	Aplikasi jQuery	50
4.9	Aplikasi jQuery Migrate	51
4.10	Aplikasi jQuery dari [2]	51
5.1	UseCase Diagram	54
5.2	Daftar Aplikasi	55
5.3	Menampilkan Chart Ketika Menekan Tombol View	56
5.4	URL dengan Hasil Keseluruhan Aplikasi	57
5.5	Aplikasi yang Populer	58
5.6	Menampilkan <i>Chart</i>	72
5.7	Melakukan Tombol Next	73
5.8	Melakukan Tombol Prev	73
5.9	Melakukan Tombol Next	74
5.10	Melakukan Tombol Prev	75
5.11	Melakukan Tombol Next	75
5.12	Melakukan Tombol Prev	76

BAB 1

PENDAHULUAN

- ³ Pada bab ini dijelaskan latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.

5 1.1 Latar Belakang

6 Di masa teknologi saat ini, banyak perusahaan yang menggunakan *website* sebagai tempat untuk
7 mencari informasi. Terdapat banyak *website* yang menggunakan aplikasi yang sudah usang. HT-
8 TPArchive memiliki dataset penggunaan aplikasi setiap *website* dari *desktop* dan *mobile* pada bulan
9 Januari tahun 2016 sampai sekarang yang dapat dilihat menggunakan teknologi BigQuery. Dataset
10 pada HTTPArchive didapatkan dari *Chrome User Experience Report* (CrUX). CrUX merupakan
11 *public dataset* dari *user experience* data pada jutaan *website*. Pengujian pada *dataset* tersebut
12 dilakukan dengan menggunakan *Chrome* pada *desktop* dan *mobile*.

HTTPArchive¹ adalah sebuah *project* yang bersifat *open source* untuk melihat bagaimana *website* dibuat. Di dalam HTTPArchive terdapat data-data historis yang disediakan untuk menunjukkan bagaimana *website* terus berkembang dan *project* ini sering digunakan untuk penelitian. Didalam HTTPArchive terdapat *dataset* yang berisi jutaan *website* setiap bulan dan dapat dianalisis menggunakan teknologi BigQuery.

BigQuery [3] adalah salah satu produk dari Google untuk menyimpan kumpulan data yang berbasis *cloud* dan dapat digunakan untuk menganalisis data. BigQuery dapat menjalankan *query* dalam skala *terabyte* dalam hitungan detik dan *petabyte* dalam hitungan menit. BigQuery digunakan untuk menganalisis data yang besar dengan melakukan *query*.

22 Pada [2] dilakukan penelitian tentang seberapa besar penggunaan aplikasi usang pada *website* di
23 Indonesia. Data diambil dari *website* Alexa, dari 1.500 situs teratas menurut peringkat Alexa untuk
24 pengunjung situs di Indonesia dan mengidentifikasi jenisnya aplikasi yang dipakai beserta nomor
25 versinya, lebih dari setengah atau 63% aplikasi yang digunakan berhasil dibandingkan dengan skrip
26 yang telah dibuat dan hasilnya aplikasi tidak lagi didukung oleh pengelolanya.

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

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

1 Pada skripsi ini, dibuat sebuah replikasi dari [2] tetapi dengan data yang lebih besar. Data dapat
2 diambil dari HTTPArchive dengan melakukan *query* pada BigQuery. Pada penelitian ini dilakukan
3 perhitungan pada jumlah aplikasi yang sudah diberi versi dan belum diberi versi. Terdapat beberapa
4 aplikasi yang informasi versinya tidak dapat ditentukan. Versi aplikasi yang tidak dapat ditentukan
5 disebut NON-CONCLUSIVE. Versi aplikasi yang NON-CONCLUSIVE biasanya berisi simbol.
6 Selain itu terdapat beberapa versi aplikasi yang kosong. Versi aplikasi yang kosong disebut sebagai
7 NOT-VERSIONED Versi aplikasi yang dipakai setiap *website* juga dibandingkan dengan versi
8 aplikasi yang masih didukung berdasarkan *official website*-nya. Pada penelitian ini hanya berfokus
9 pada *semantic version*. *Semantic version* merupakan tata cara penentuan urutan pada aplikasi.
10 Pada *semantic version* terdapat tiga komponen pengurutannya, yaitu *major*, *minor*, dan *patch*.
11 Kemudian hasil tersebut ditampilkan dalam bentuk *bar chart*.

12 1.2 Rumusan Masalah

13 Berikut ini adalah rumusan masalah dari penelitian ini:

- 14 1. Bagaimana cara mengambil nilai informasi untuk mendukung pencarian aplikasi yang usang?
- 15 2. Bagaimana mereplikasi jurnal [2] dengan menggunakan data yang lebih besar?
- 16 3. Berapa banyak *website* pada HTTPArchive yang menggunakan aplikasi yang semua aplikasinya
17 masih didukung?

18 1.3 Tujuan

19 Berikut ini adalah tujuan dari penelitian ini:

- 20 1. Mengambil nilai informasi dengan cara melakukan query untuk mengumpulkan daftar website,
21 mencari aplikasi yang digunakan, mencari aplikasi yang digunakan website, mengelompokkan
22 berdasar nama semua aplikasi yang dipakai, mencari data tentang versi aplikasi yang masih
23 didukung, dan melakukan perbandingan antara versi aplikasi yang masih dipakai sekarang
24 dengan aplikasi yang masih didukung.
- 25 2. Mereplikasi jurnal [2] dengan menggunakan data yang lebih besar, dikarenakan jurnal [2]
26 menggunakan data yang lebih sedikit, sehingga *chart* dari aplikasi yang ditampilkan juga
27 lebih sedikit. Data pada jurnal [2] ruang lingkupnya hanya sebatas wilayah Indonesia saja,
28 sedangkan pada skripsi ini dilakukan penelitian dengan ruang lingkup global atau dunia.
- 29 3. Mencari jumlah *website* pada HTTPArchive yang menggunakan aplikasi yang semua aplikasinya
30 masih didukung.

31 1.4 Batasan Masalah

32 Berikut ini adalah batasan masalah dari penelitian ini:

- 33 1. Data HTTPArchive yang digunakan adalah data *technologies* pada bulan Agustus tahun 2020.
- 34 2. Versi aplikasi berisi simbol merupakan NON-CONCLUSIVE.
- 35 3. Pada penelitian ini hanya berfokus pada *semantic version* yang dimana digit *major*, *minor*,
36 dan *patch* adalah angka.

- 1 4. *Chart* pada aplikasi ini tidak menunjukkan data yang *UNVERSIONED* dan *NON-CONCLUSIVE*
2 karena datanya terlalu besar sehingga mengakibatkan data lain tidak terlihat.

3 **1.5 Metodologi**

- 4 Bagian-bagian pekerjaan skripsi ini adalah sebagai berikut:
- 5 1. Melakukan studi literatur mengenai HTTPArchive.
- 6 2. Melakukan studi literatur mengenai Query
- 7 3. Melakukan studi literatur mengenai BigQuery.
- 8 4. Melakukan studi literatur mengenai ReactJS.
- 9 5. Melakukan studi literatur mengenai NodeJS.
- 10 6. Melakukan studi literatur mengenai ChartJS.
- 11 7. Melakukan studi literatur mengenai *website* yang dikatakan usang.
- 12 8. Menganalisis beberapa *website* yang dikatakan usang.
- 13 9. Membandingkan versi aplikasi pada data HTTPArchive yang dipakai sekarang dengan versi
14 aplikasi yang ada pada dokumentasi.
- 15 10. Membuat perangkat lunak untuk menampilkan data.
- 16 11. Menulis dokumen skripsi.

17 **1.6 Sistematika Pembahasan**

- 18 Laporan penelitian tersusun ke dalam enam bab secara sistematis sebagai berikut.
- 19 • Bab 1 Pendahuluan
20 Berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan
21 sistematika pembahasan.
- 22 • Bab 2 Dasar Teori
23 Berisi teori BigQuery, teori HTTPArchive, teori Node.js, teori React.js, Chart.js, dan Express.js.
- 25 • Bab 3 Percobaan Awal
26 Berisi eksplorasi BigQuery, dataset yang digunakan di HTTPArchive, langkah *query* yang
27 dilakukan dengan data terbatas untuk membandingkan versi aplikasi, dan hasil *sample* data
28 Apache.
- 29 • Bab 4 Penggalian Data
30 Berisi langkah-langkah query yang dilakukan untuk mengumpulkan data yang besar, penjelasan
31 tentang sample data antara aplikasi apache dan nginx, php dan python, dan jquery dan jquery
32 migrate.
- 33 • Bab 5 Pembangunan Perangkat Lunak
34 Berisi perancangan perangkat lunak yang dibangun, masukan dan keluaran dari perangkat
35 lunak.
- 36 • Bab 6 Kesimpulan dan Saran
37 Berisi kesimpulan dari awal hingga akhir penelitian dan saran untuk penelitian berikutnya.

BAB 2

LANDASAN TEORI

3 2.1 BigQuery

4 Google memiliki salah satu produk yaitu BigQuery yang berbasis *cloud* dan dapat digunakan untuk
5 menganalisis data[4, 3]. BigQuery memaksimalkan fleksibilitas dengan memisahkan memisahkan
6 mesin komputasi yang menganalisa data. BigQuery dapat digunakan sebagai tempat penyimpanan
7 data dan data tersebut dapat dianalisis. Data dalam BigQuery dimasukkan dalam sebuah *dataset*.
8 *Dataset* berisikan tabel-tabel yang dapat dianalisis. Google meluncurkan BigQuery secara publik
9 pada tahun 2012. Saat ini BigQuery sudah berkembang menjadi penyedia penyimpanan terstruktur
10 berbasis *cloud* yang dikelola dan di-*hosting*.

2.1.1 *Cloud Storage System*

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

17 2.1.2 SQL (Structured Query Language) [1]

18 SQL adalah bahasa pemrograman menghasilkan, memanipulasi, dan mengambil informasi dari
19 *database* relasional. Salah satu *syntax sql* yang didukung BigQuery adalah *Standard SQL*. *Standard*
20 *SQL* mengambil informasi dari *database* dengan menggunakan *query* ¹. Mengambil informasi
21 dari *database* relasional harus menggunakan *query*. *Query* merupakan *syntax* atau perintah yang
22 digunakan untuk mengambil dan menghasilkan data dari *database*.

23 Query Clauses

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

- ## 26 • SELECT dan FROM

27 Fungsi dari klausula `SELECT` adalah untuk menentukan kolom dari suatu tabel yang ditampilkan
28 dalam *query result*. Fungsi dari klausula `FROM` adalah mengidentifikasi tabel yang ingin diambil

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

1 datanya. Dalam mengambil data dari *database* setidaknya minimal harus menggunakan dua
2 klausanya ini. Klausanya ini memiliki *syntax* seperti:

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

5

- 6 • WHERE

7 Fungsi dari klausanya WHERE adalah untuk membatasi jumlah baris dalam *query result* ber-
8 dasarkan kondisi tertentu. Klausanya WHERE digunakan jika terdapat beberapa kondisi yang
9 ingin dicari dari *database* tersebut. Klausanya ini memiliki *syntax* seperti:

```
10    SELECT column1, column2, ...
11    FROM table_name
12    WHERE condition
```

13

- 14 • GROUP BY

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

```
16    SELECT column1, column2, ...
17    FROM table_name
18    WHERE condition
19    GROUP BY column_name, ...
```

20

- 21 • ORDER BY

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

```
24    SELECT column1, column2, ...
25    FROM table_name
26    WHERE condition
27    GROUP BY column_name, ...
28    ORDER BY column_name, ... ASC|DESC
```

29

Query Aggregation

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

31

- 32 • MAX()

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

```
34    SELECT MAX(column_name)
35    FROM table_name
36    WHERE condition;
```

1 • MIN()

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

```
4         SELECT MIN(column_name)
5         FROM table_name
6         WHERE condition;
```

7 • AVG()

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

```
10         SELECT AVG(column_name)
11         FROM table_name
12         WHERE condition;
```

13 • COUNT()

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

```
16         SELECT COUNT(column_name)
17         FROM table_name
18         WHERE condition;
```

19 • SUM()

20 Fungsi ini bertujuan untuk mengembalikan menjumlahkan data setiap baris dari kolom sebuah
21 tabel. Fungsi SUM memiliki contoh *syntax* seperti:

```
22         SELECT SUM(column_name)
23         FROM table_name
24         WHERE condition;
```

25 • STRING_AGG()

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

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

30 **Querying Multiple Tables**

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

1 • LEFT OUTER JOIN

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

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

10

11 • RIGHT OUTER JOIN

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

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

20

21 • FULL OUTER JOIN

22 FULL OUTER JOIN merupakan gabungan dari LEFT OUTER JOIN dan RIGHT OUTER
23 JOIN. FULL OUTER JOIN memiliki *syntax* seperti:

```
24       SELECT column_name(s)
25       FROM table1
26       FULL OUTER JOIN table2
27       ON table1.column_name = table2.column_name
28       WHERE condition;
```

29

30 • INNER JOIN

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

```
33       SELECT column_name(s)
34       FROM table1
35       INNER JOIN table2
36       ON table1.column_name = table2.column_name;
```

37

1 Subquery

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

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

11 REPEAT()

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

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

15 COUNTIF()

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

```
19 COUNTIF(
20   expression
21 )
22 [ OVER over_clause ]
```

23 UNNEST()

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

26 SPLIT()

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

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

31 CASE

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

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

6 2.2 HTTPArchive

7 HTTPArchive adalah sebuah *open-source project* yang melihat bagaimana *website* dibuat. HTTPArchive
8 menyediakan data-data historis untuk melihat bagaimana *website* berkembang. HTTPArchive
9 pertama sekali dimulai pada tahun 2010 oleh Steve Souders dan di-*maintain* oleh Pat Meenan, Rick
10 Viscomi, Paul Calvano, and Barry Pollard. Data url HTTPArchive didapatkan menggunakan CrUX
11 (*Chrome User Experience Report*) kemudian url dikirimkan ke WebPageTest setiap bulannya[5].
12 CrUX adalah sebuah dataset yang bersifat publik yang berisi data *user experience* dari jutaan
13 *website*. Data ini berasal dari data yang dikumpulkan dari pengguna yang telah memilih untuk
14 mengsinkronkan *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 HTTPArchive adalah anggota komunitas web, para sarjana, dan pemimpin
19 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 HTTPArchive terdapat *dataset* yang dapat diambil menggunakan teknologi BigQuery.
27 Salah satu *dataset* yang dapat diambil adalah dataset *technologies*. *Dataset* pada *technologies* berisi
28 tabel-tabel dari bulan Januari tahun 2016 sampai dengan sekarang yang terdiri dari *website* pada
29 *desktop* dan *mobile*. *Dataset* bulan Agustus tahun 2020 pada *desktop* memiliki 61.203.638 baris dan
30 pada *mobile* memiliki 67.452.994 baris.

31 2.3 Pengukuran Aplikasi Usang Pada Beberapa *Website* Populer Di 32 Indonesia[2]

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

2.3.1 Research Method

2 Terdapat empat langkah dalam melakukan penelitian ini, yaitu:

3 1. Memilih *list website* yang populer

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

6 2. Mengidentifikasi aplikasi yang dipakai *website*

7 Untuk setiap *website* dilakukan pengidentifikasi nomor versi yang dipakai. Hal ini dibantu
8 dengan menggunakan *third party* yaitu Wappalyzer.

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

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

13 • Versi aplikasi yang didukung dapat dilihat secara eksplisit di dalam *website*.

14 • Dokumen untuk versi aplikasi tersebut masih tersedia.

15 • Aplikasi secara langsung memberikan pernyataan untuk versi yang masih didukung.

16 4. Membandingkan versi yang dipakai aplikasi saat ini dengan versi aplikasi yang didukung
17 dapat dilihat pada gambar 2.1

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

21 • *Not-versioned* berarti aplikasi yang terdeteksi oleh Wappalyzer tidak memiliki informasi
22 versi sehingga tidak dapat dibandingkan.

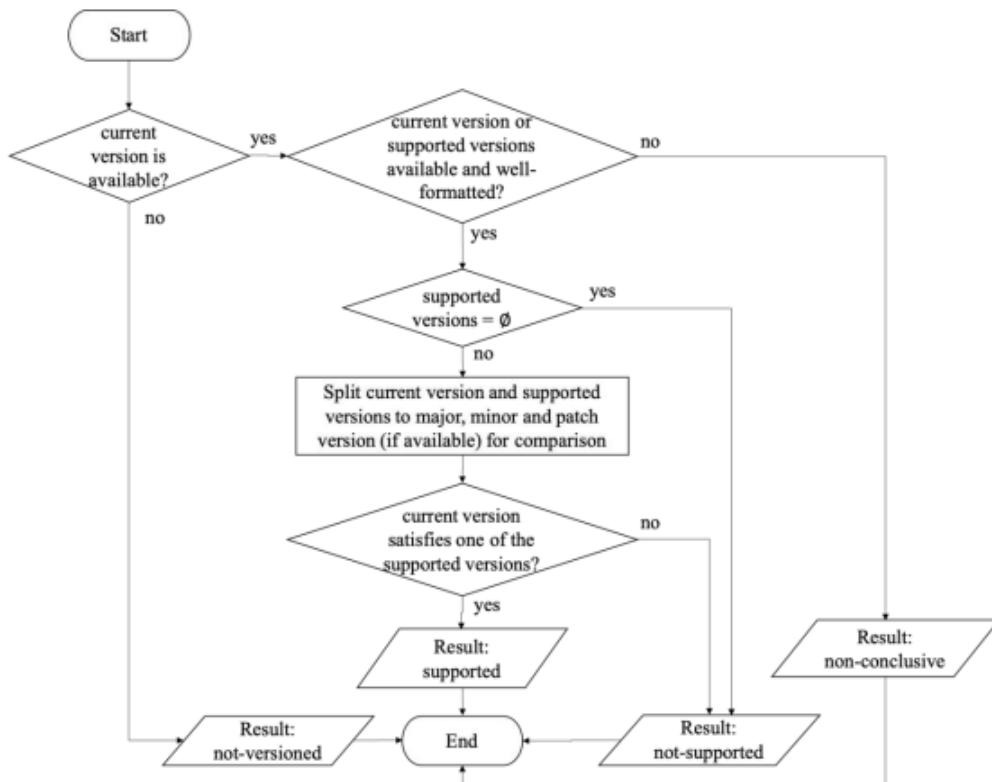
23 • Non-konklusif dapat berarti salah satu dari dua:

24 – Dapat mengambil nomor versi yang digunakan dalam aplikasi, tetapi tidak dapat
25 menentukan apakah versi tersebut masih didukung atau tidak oleh pengelola.

26 – Versi yang didukung untuk aplikasi tertentu tidak diketahui.

27 • Tidak didukung berarti dapat disimpulkan bahwa aplikasi yang digunakan menggunakan
28 nomor versi yang tidak didukung oleh pengelola.

29 • Didukung berarti dapat disimpulkan bahwa aplikasi yang digunakan menggunakan nomor
30 versi masih didukung oleh pengelola.



Gambar 2.1: Algoritma untuk membandingkan versi yang dipakai dengan versi yang masih didukung

2.3.2 Hasil Keseluruhan

- 2 Pada jurnal[2], dari 1.500 URL yang dideteksi oleh Wappalyzer, hanya 1.439 URL yang berhasil
 3 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

- 4
 5 Tabel 2.2 adalah daftar sepuluh website yang paling popular. Dari daftar tersebut terlihat
 6 banyak sekali website yang menggunakan aplikasi yang tidak ada informasi versinya. Tetapi untuk
 7 yang ada informasi versinya, terdapat beberapa aplikasi yang sudah tidak didukung. Beberapa
 8 aplikasi yang sudah tidak didukung dari sepuluh website tersebut adalah Bootstrap, Font Awesome,
 9 jQuery, dan PHP. Pada tabel 2.3 terdapat 1,500 website yang dipisahkan setiap 150 website yang
 10 diurutkan berdasarkan rank website tersebut. Untuk setiap baris pada tabel tersebut dihitung
 11 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

- 1 Pada tabel 2.4, terdapat beberapa aplikasi yang banyak digunakan. Beberapa aplikasi tersebut
 2 diambil dari 1.500 *website* teratas dan mem-*filter* aplikasi yang versinya tidak dapat diidentifikasi
 3 di 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

4 2.4 Node.js

- 5 *Node.js* adalah sebuah *asynchronous event-driven JavaScript runtime*. *Node.js* memungkinkan
 6 untuk menjalankan perintah JavaScript dari sisi server[6]. Contoh *syntax* pada *Node.js* yaitu:

7 81 const http = require('http');

```

12
13  const hostname = '127.0.0.1';
14  const port = 3000;
15
16  const server = http.createServer((req, res) => {
17    res.statusCode = 200;
18    res.setHeader('Content-Type', 'text/plain');
19    res.end('Hello World');
20  });
21
22  server.listen(port, hostname, () => {
23    console.log(`Server running at http://${hostname}:${port}/`);
24  });

```

15 2.4.1 Node Package Manager

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

- 18 • NPM *website*.
- 19 • NPM CLI (*Command Line Interface*).
- 20 • NPM *Registry*.

21 Beberapa kegunaan dari menggunakan NPM adalah:

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

25 2.4.2 NPM CLI

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

31 `npm <command> [args]`

32 2.5 ReactJS

33 ReactJS merupakan *library* yang disediakan JavaScript untuk membuat *interface*. ReactJS dibuat
34 oleh Facebook[8].

35 2.5.1 JSX

36 JSX adalah sebuah ekstensi Javascript yang dapat mengikutsertakan HTML dalam Javascript. JSX
37 menghasilkan elemen React.

38 Menyatukan Ekspresi dalam JSX

39 Berikut ini adalah contoh penggunaan JSX:

```

1  const name = 'Budi';
2  const element = <h1>Halo, {name}</h1>;
3
4  ReactDOM.render(
5    element,
6    document.getElementById('root')
7  );

```

- 10 Pada contoh diatas, variabel *name* dibungkus dengan menggunakan tanda kurung kurawal. Semua
11 ekspresi Javascript valid dalam tanda kurung kurawal di JSX.

12 Mengspesifikasikan Atribut Dengan JSX

- 13 Penulis dapat menggunakan tanda petik untuk mengspesifikasikan *string literal* sebagai atribut:

```

14 1 const element = <a href="https://www.reactjs.org"> link </a>;

```

- 15 Penulis juga dapat menggunakan kurung kurawal untuk mengspesifikasikan ekspresi Javascript di
16 dalam atribut:

```

17 1 const element = <img src={user.avatarUrl}></img>;

```

22 Menspesifikasikan Elemen Anak dengan JSX

- 23 Jika *tag* bersifat kosong atau tidak memiliki elemen anak, penulis dapat menutup *tag*-nya secara
24 langsung dengan */>*, seperti pada potongan kode dibawah ini.

```

25 1 const element = <img src={user.avatarUrl} />;

```

- 26 Didalam tag JSX memungkinkan untuk memiliki elemen anak, yang dapat dilihat pada potongan
27 kode dibawah ini. Elemen anak dalam kode dibawah ini adalah elemen *<h1>* dan *<h2>*.

```

28 1 const element = (
29 2   <div>
30 3     <h1>Halo!</h1>
31 4     <h2>Senang melihatmu di sini.</h2>
32 5   </div>
33 6 );

```

38 2.5.2 Melakukan *Render* Elemen

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

42 Melakukan *Render* Elemen Kedalam DOM

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

```

46 1 <div id="root"></div>

```

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

```

49 1 const element = <h1>Hello, world</h1>;
50 2 ReactDOM.render(element, document.getElementById('root'));

```

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

3 Memperbarui Elemen yang Di *Render*

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

```
7
8 1 function tick() {
9 2   const element = (
10 3     <div>
11 4       <h1>Hello, world!</h1>
12 5       <h2>It is {new Date().toLocaleTimeString()}.</h2>
13 6     </div>
14 7   );
15 8   ReactDOM.render(element, document.getElementById('root'));
16 9 }
17 10
18 11 setInterval(tick, 1000);
```

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

21 2.5.3 *Components and Props*

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

26 Fungsi dan Komponen Kelas

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

```
29 1 function Welcome(props) {
30 2   return <h1>Halo, {props.name}</h1>;
31 3 }
```

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

36 Melakukan *Render* pada Sebuah Komponen

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

```
39 1 const element = <Welcome name="Sara" />;
```

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

```
45 1 function Welcome(props) {
46 2   return <h1>Halo, {props.name}</h1>;
47 3 }
48 4
49 5 const element = <Welcome name="Sara" />;
50 6 ReactDOM.render(
```

```

17     element,
28   document.getElementById('root')
39 );

```

5 Menyusun Komponen

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

```

11 function Welcome(props) {
12   return <h1>Halo, {props.name}</h1>;
13 }
14
15
16 function App() {
17   return (
18     <div>
19       <Welcome name="Sara" />
20       <Welcome name="Cahal" />
21       <Welcome name="Edite" />
22     </div>
23   );
24 }
25

```

26 2.5.4 Penanganan Event

27 Menangani *events* dengan elemen React sangat mirip seperti menangani sebuah *events* pada elemen
28 DOM. Ada beberapa perbedaan sintaks:

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

31 Berikut ini adalah contoh sintaks pada HTML:

```

32 <button onclick="activateLasers()">
33   Aktivasi Laser
34 </button>
35

```

36 Sintaks HTML memiliki sedikit perbedaan dengan sintaks pada React. Berikut ini adalah contoh
37 sintaks pada React:

```

38 <button onClick={activateLasers}>
39   Aktivasi Laser
40 </button>
41

```

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

```

45 <a href="#" onClick="console.log('The link was clicked.'); return false">
46   Click me
47 </a>

```

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

```

51 function ActionLink() {
52   function handleClick(e) {
53     e.preventDefault();
54     console.log('Tautan diklik.');
55   }
56
57   return (
58
59
60

```

```

18     <a href="#" onClick={handleClick}>
19       Klik Saya
20     </a>
21   );
22 }
```

7 Pada kode diatas e merupakan *event* tiruan. React mendefinisikan *event* tiruan ini berdasarkan
 8 W3C *spec*², jadi tidak perlu khawatir kesesuaian antar lintas *browser*. *Event* dalam React tidak
 9 bekerja secara sama dengan *event native* dari *browser*.

10 Mengoper Argumen Kedalam Penanganan *Event*

11 Di dalam perulangan, umumnya ketika ingin mengoper sebuah parameter ekstra kedalam penanganan
 12 *event*. Sebagai contoh, jika id sama dengan id baris, maka salah satu dari kedua contoh berikut
 13 dapat dijalankan:

```

14 <button onClick={(e) => this.deleteRow(id, e)}>Delete Row</button>
15 <button onClick={this.deleteRow.bind(this, id)}>Delete Row</button>
16 }
```

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

23 2.5.5 *Render* Bersyarat

24 *Render* bersyarat pada React memiliki fungsi yang sama dengan operator bersyarat pada Javascript.
 25 Pada Javascript operator *if* atau operator bersyarat digunakan untuk merepresentasikan elemen
 26 pada *state* tertentu, kemudian React memperbarui UI pada *state* tersebut.

27 Contohnya seperti dua komponen berikut ini:

```

28 function UserGreeting(props) {
29   return <h1>Welcome back!</h1>;
30 }
31
32 function GuestGreeting(props) {
33   return <h1>Please sign up.</h1>;
34 }
35
36 }
```

37 Komponen diatas digunakan untuk melakukan *Greetings* berdasarkan pada apakah *user* sudah
 38 melakukan login:

```

39 function Greeting(props) {
40   const isLoggedIn = props.isLoggedIn;
41   if (isLoggedIn) {
42     return <UserGreeting />;
43   }
44   return <GuestGreeting />;
45 }
46
47 ReactDOM.render(
48   // Try changing to isLoggedIn={true}:
49   <Greeting isLoggedIn={false} />,
50   document.getElementById('root')
51 );
52
53 }
```

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

1 2.5.6 *List* dan *Keys*

- 2 Javascript dapat menggunakan fungsi *map()* untuk mengambil array *numbers* dan menggandakan
 3 angkanya. *Map()* mengembalikan nilai dalam bentuk *array* yang baru kemudian disimpan dalam
 4 sebuah variabel *doubled*. Berikut ini adalah contoh kodennya:

```
5 1 const numbers = [1, 2, 3, 4, 5];
6 2 const doubled = numbers.map((number) => number * 2);
7 3 console.log(doubled);
```

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

12 Melakukan Render pada Banyak Komponen

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

```
15 1 const numbers = [1, 2, 3, 4, 5];
16 2 const listItems = numbers.map((number) =>
17 3   <li>{number}</li>
18 4 );
```

- 21 Pada kode diatas, dilakukan perulangan pada *numbers* yang berisi *array* dengan menggunakan
 22 fungsi *map()*. Hasil yang dikeluarkan adalah sebuah elemen ** untuk setiap *item* yang kemudian
 23 dimasukkan kedalam variabel *listItem*.

```
24 1 ReactDOM.render(
25 2   <ul>{listItems}</ul>,
26 3   document.getElementById('root')
27 4 );
```

- 30 Kemudian dengan menggunakan kode diatas, *array* *listItem* tersebut dapat dimasukkan kedalam
 31 elemen ** dan me-*render*-nya kedalam DOM.

32 Daftar Komponen Dasar

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

```
35 1 function NumberList(props) {
36 2   const numbers = props.numbers;
37 3   const listItems = numbers.map((number) =>
38 4     <li>{number}</li>
39 5   );
40 6   return (
41 7     <ul>{listItems}</ul>
42 8   );
43 9 }
44 0
45 1
46 1 const numbers = [1, 2, 3, 4, 5];
47 2 ReactDOM.render(
48 3   <NumberList numbers={numbers} />,
49 4   document.getElementById('root')
50 5 );
```

- 52 Ketika penulis menjalankan kode ini, penulis mendapatkan peringatan bahwa *key* harus disediakan
 53 untuk *item* di dalam *list*. Sebuah "key" adalah atribut *string* spesial yang perlu disertakan dalam
 54 pembuatan *list* elemen. Penulis harus sertakan *key* ke dalam *list* item pada *numbers.map()* dan
 55 memperbaiki masalah *key* yang hilang.

```
 21 function NumberList(props) {
 22   const numbers = props.numbers;
 23   const listItems = numbers.map((number) =>
 24     <li key={number.toString()}>
 25       {number}
 26     </li>
 27   );
 28   return (
 29     <ul>{listItems}</ul>
 30   );
 31 }
 32
 33 const numbers = [1, 2, 3, 4, 5];
 34 ReactDOM.render(
 35   <NumberList numbers={numbers} />,
 36   document.getElementById('root')
 37 );
```

20 Key

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

```
24
25 1  const numbers = [1, 2, 3, 4, 5];
26 2  const listItems = numbers.map((number) =>
27 3  <li key={number.toString()}>
28 4  {number}
29 5  </li>
30 6  );
31
```

32 Cara terbaik untuk menentukan *key* yang digunakan adalah menggunakan *string* unik untuk
33 mengidentifikasi *item*. Biasanya menggunakan ID dari data sebagai *key*:

```
34
35 1  const todoItems = todos.map((todo) =>
36 2  <li key={todo.id}>
37 3  {todo.text}
38 4  </li>
39 5  );
40
```

⁴¹ Ketika tidak memiliki ID yang stabil untuk me-render *item*, penulis dapat menggunakan indeks
⁴² dari item sebagai *key* untuk pilihan terakhir:

```
43
44 1  const todoItems = todos.map((todo, index) =>
45 2    <li key={index}>
46 3      {todo.text}
47 4    </li>
48 5  );

```

50 2.5.7 Form

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

```
54
55 1    <form>
56 2    <label>
57 3    Name:
58 4    <input type="text" name="name" />
59 5    </label>
60 6    <input type="submit" value="Submit" />
61 7  </form>
```

⁶³ Form ini memiliki perilaku dasar dari *form* HTML biasa yakni menuju ke halaman baru ketika ⁶⁴ user mengirim *form* tersebut. Jika menginginkan perilaku seperti ini di React, ini sebenarnya

1 dapat bekerja. Namun di banyak kasus, lebih mudah untuk memiliki sebuah fungsi JavaScript yang
 2 menangani sebuah submisi dari sebuah *form* dan memiliki akses terhadap data yang dimasukkan
 3 pengguna ke dalam *form*. Cara standar untuk mencapai hal ini adalah dengan teknik yang disebut
 4 *“controlled component”*.

5 ***Controlled Component***

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

```

12  class NameForm extends React.Component {
13 1   constructor(props) {
14 2     super(props);
15 3     this.state = {value: ''};
16 5
17 6     this.handleChange = this.handleChange.bind(this);
18 7     this.handleSubmit = this.handleSubmit.bind(this);
19 8   }
20 9
21 10  handleChange(event) {
22 11    this.setState({value: event.target.value});
23 12  }
24 13
25 14  handleSubmit(event) {
26 15    alert('A name was submitted: ' + this.state.value);
27 16    event.preventDefault();
28 17  }
29 18
30 19  render() {
31 20    return (
32 21      <form onSubmit={this.handleSubmit}>
32 22        <label>
32 23          Name:
32 24          <input type="text" value={this.state.value} onChange={this.handleChange} />
32 25        </label>
32 26        <input type="submit" value="Submit" />
32 27      </form>
32 28    );
32 29  }
32 30  }
32 31
  
```

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

47 **Mengendalikan Nilai Kosong**

48 Menentukan nilai *prop* pada *controlled component* mencegah pengguna mengubah masukan. Jika
 49 sudah menetapkan nilai *value* tetapi masukan masih dapat diubah, mungkin nilai *value* yang
 50 ditetapkan bersifat *undefined* atau *null*.

51 **2.5.8 Hooks**

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

1 UseState

2 *UseState* adalah sebuah *hook* yang memungkinkan penulis untuk menambahkan *react state* ke
 3 *function components*. Cara untuk menggunakan *useState* adalah dengan meng-*import useState hook*
 4 dari react dengan menggunakan sintak:

```
5 import React, { useState } from 'react';
6
7 function Example() {
8 // ...
9 }
```

10 Berikut ini adalah beberapa informasi mengenai *useState*:

- 11 • Mendeklarasi sebuah *State Variabel*

12 Di dalam sebuah kelas, dalam menginisialisasikan variabel dalam *state* biasanya menggunakan
 13 *this.state*.

```
14
15     class Example extends React.Component {
16         constructor(props) {
17             super(props);
18             this.state = {
19                 count: 0
20             };
21         }
22     }
```

23 Di dalam sebuah *component function* tidak terdapat *this*, sehingga tidak dapat menetapkan
 24 variabel atau membaca *this.state*. Melainkan menggunakan *useState hook* didalam komponen.

```
25
26     import React, { useState } from 'react';
27
28     function Example() {
29         // Declare a new state variable, which we'll call "count"
30         const [count, setCount] = useState(0);
```

- 32 • Membaca *State*

33 Ketika ingin menampilkan *count* saat ini dalam kelas, dapat menggunakan kode *this.state.count*
 34 seperti:

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

36 Sedangkan didalam *function*, dapat menggunakan *count* secara langsung:

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

- 38 • Meng-*update State*

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

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

43 Sedangkan dalam sebuah fungsi, terdapat variabel *setCount* dan *count* sehingga tidak mem-
 44 butuhkan *this*:

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

1 UseEffect

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

```
7 1 import React, { useState, useEffect } from 'react';
8 2
9 3
10 3 function Example() {
11 4     const [count, setCount] = useState(0);
12 5
13 6     // Sama seperti componentDidMount dan componentDidUpdate:
14 7     useEffect(() => {
15 8         // Memperbarui judul dokumen menggunakan API browser
16 9         document.title = 'Anda klik sebanyak ${count} kali';
17 0     });
18 1
19 2     return (
20 3         <div>
21 4             <p>Anda klik sebanyak {count} kali</p>
22 5             <button onClick={() => setCount(count + 1)}>
23 6                 Klik saya
24 7             </button>
25 8         </div>
26 9     );
27 0 }
28 1
```

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

32 Rules of Hooks

³³ Hooks adalah fungsi JavaScript, tetapi terdapat dua aturan yang perlu diikuti keteika ingin menggunakankannya:

- ## 35 1. Hanya Panggil *Hooks* di Tingkat Atas

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

- ## 39 2. Hanya Panggil *Hooks* dari Fungsi React

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

43 2.5.9 NPX

44 NPX merupakan *execute NPM package binaries*. NPX digunakan untuk menjalankan *command* yang
45 dimiliki NPM. NPX mengeksekusi *file binary* dari *package Node.js*, baik yang sudah terinstal
46 maupun yang belum. Pembuatan *project* react dapat dilakukan dengan menggunakan sintaks:

```
47      npx create-react-app my-app  
48      cd my-app  
49      npm start
```

1 2.6 ChartJS

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

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

7 2.7 Express.js

8 *Express.js* adalah sebuah *framework* sebuah aplikasi web untuk Node.js. Untuk menginstal Express.js
9 dapat dilakukan dengan melakukan sintaks:

10 `npm install express --save`

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

13 Routing

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

16 `app.METHOD(PATH, HANDLER)`

17 Contoh perintah dalam membuat route pada Express.js adalah:

18
19 1 `app.get('/', (req, res) => {`
20 2 `res.send('hello world')`
21 3 `})`

1

BAB 3

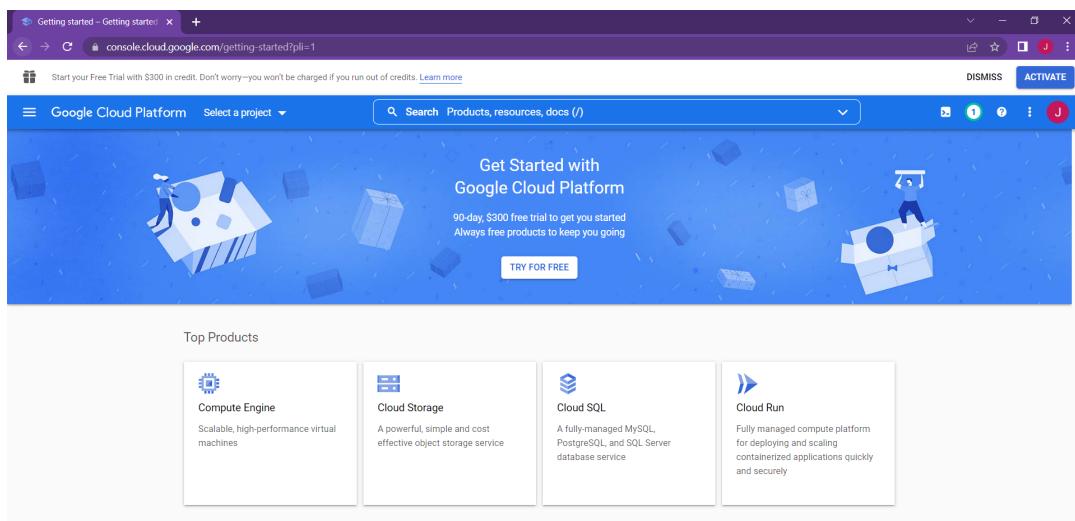
2

PERCOBAAN AWAL

- 3 Pada bab ini dijelaskan analisis masalah penelitian ini. Analisis meliputi Eksplorasi Teknologi
4 eksplorasi teknologi berisikan tentang cara mendapatkan data HTTPArchive, membuat dataset
5 baru, dan membuat tabel baru, *Dataset* Pada HTTPArchive yang digunakan, Langkah-Langkah
6 *Query* Yang Dilakukan untuk mendapatkan informasi versi, minimal versi yang masih didukung,
7 dan membandingkan versi aplikasi dengan versi aplikasi di dokumentasi sehingga dipakai data
8 yang kecil seperti melakukan batas 10 data dari tabel *technologies*, dan Hasil *Sample Data* Dengan
9 Beberapa Aplikasi yang menunjukkan contoh *chart* yang ditampilkan dari salah satu aplikasi.

10 3.1 Eksplorasi Teknologi

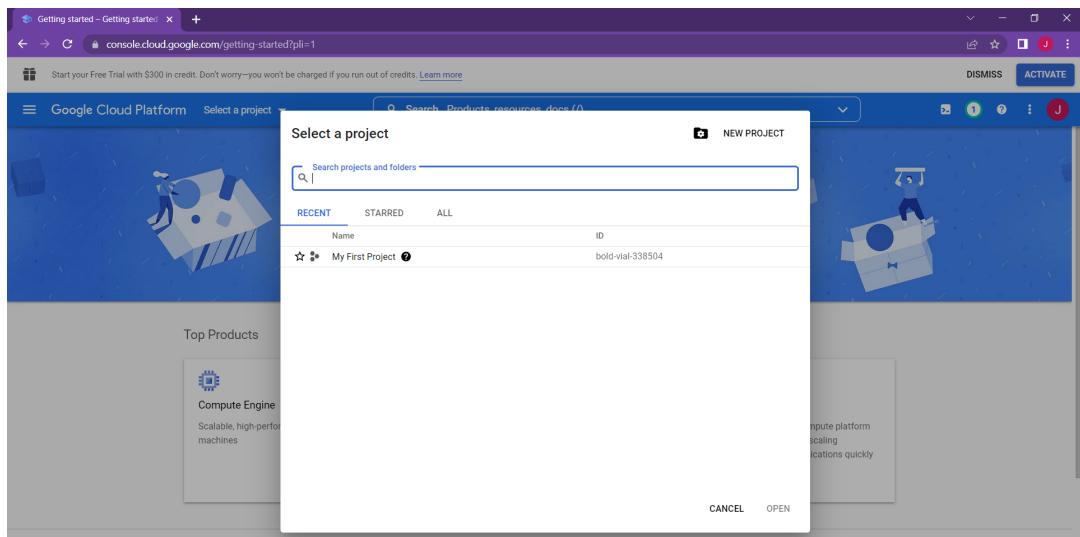
- 11 Dalam penggerjaan skripsi ini menggunakan teknologi bernama BigQuery. *Dataset* pada HTTPAr-
12 chive didapatkan dengan menggunakan teknologi BigQuery. Berikut ini adalah langkah untuk
13 mendapatkan *dataset* tersebut:
- 14 1. Membuka Google Cloud Project Page¹ dan masuk dengan menggunakan Google account.



Gambar 3.1: Halaman Awal Google Cloud Project

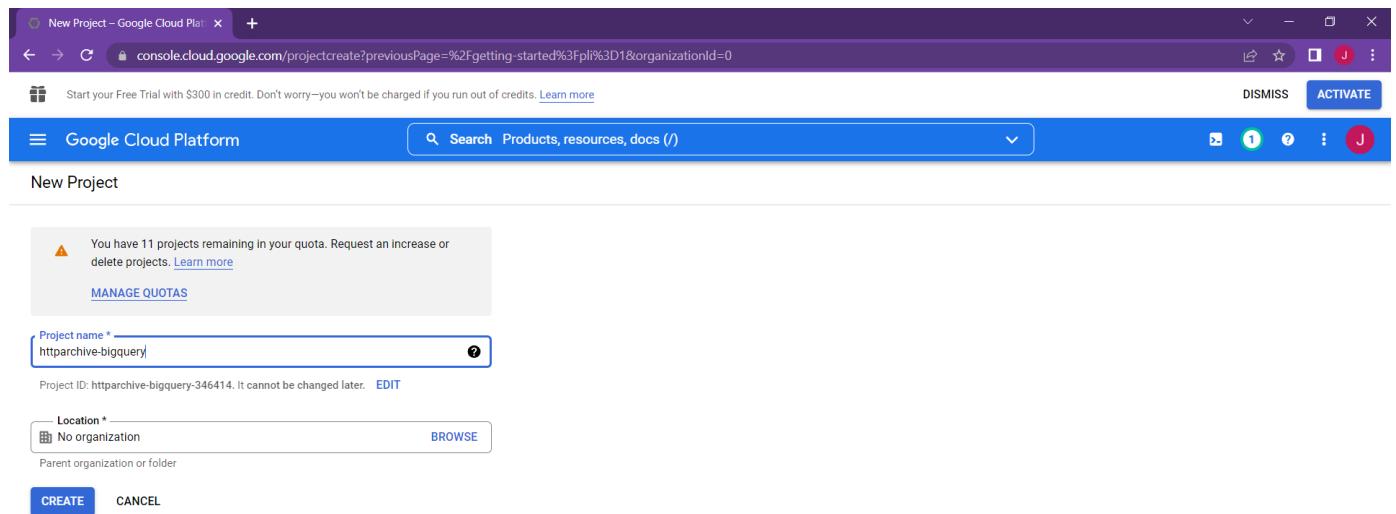
- 15 2. Memilih project kemudian "New Project"

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



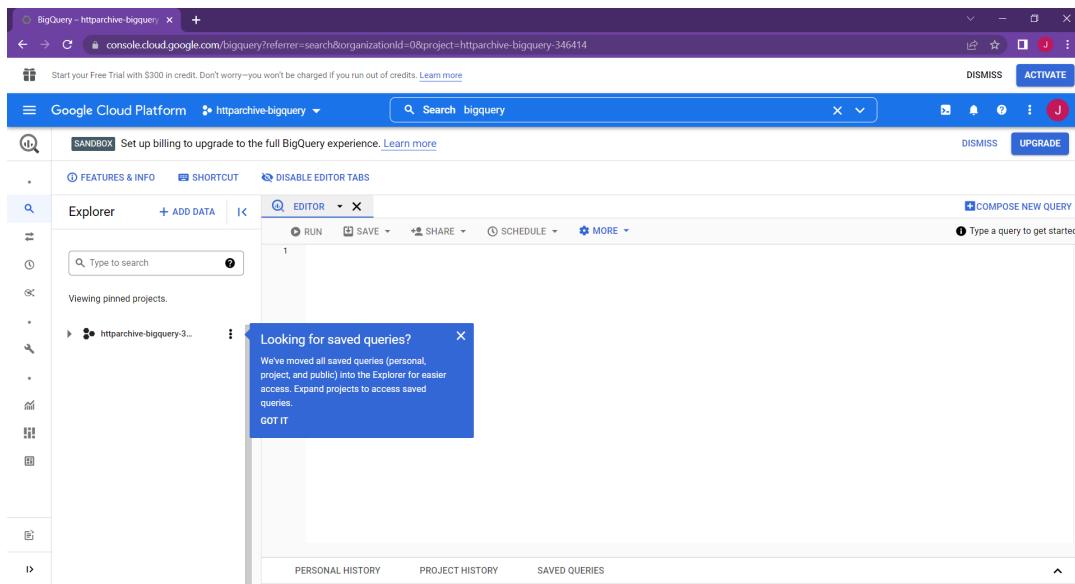
Gambar 3.2: Memilih *Project*

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

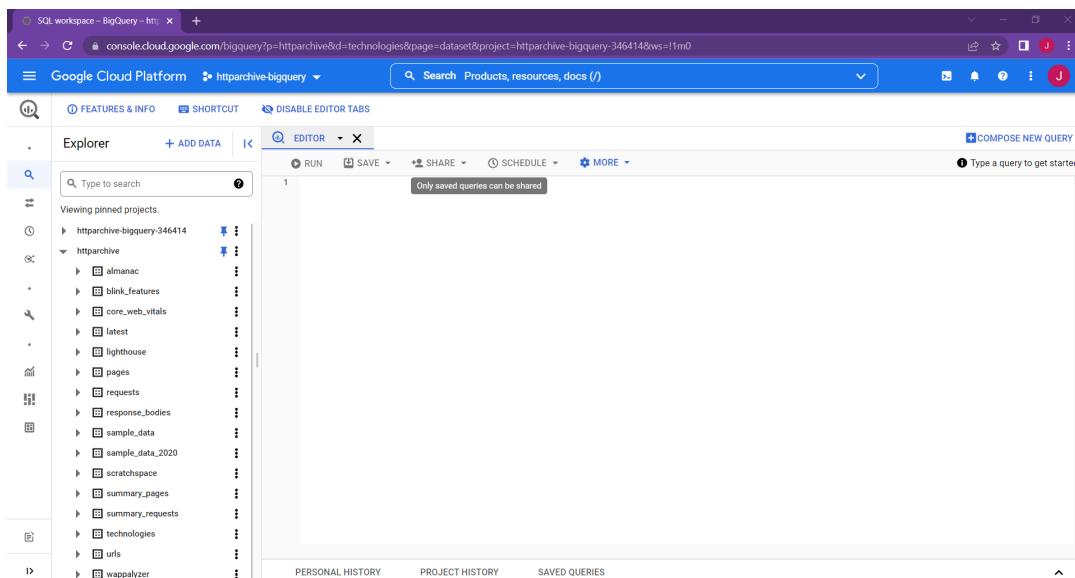


Gambar 3.3: Membuat *Project*

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.

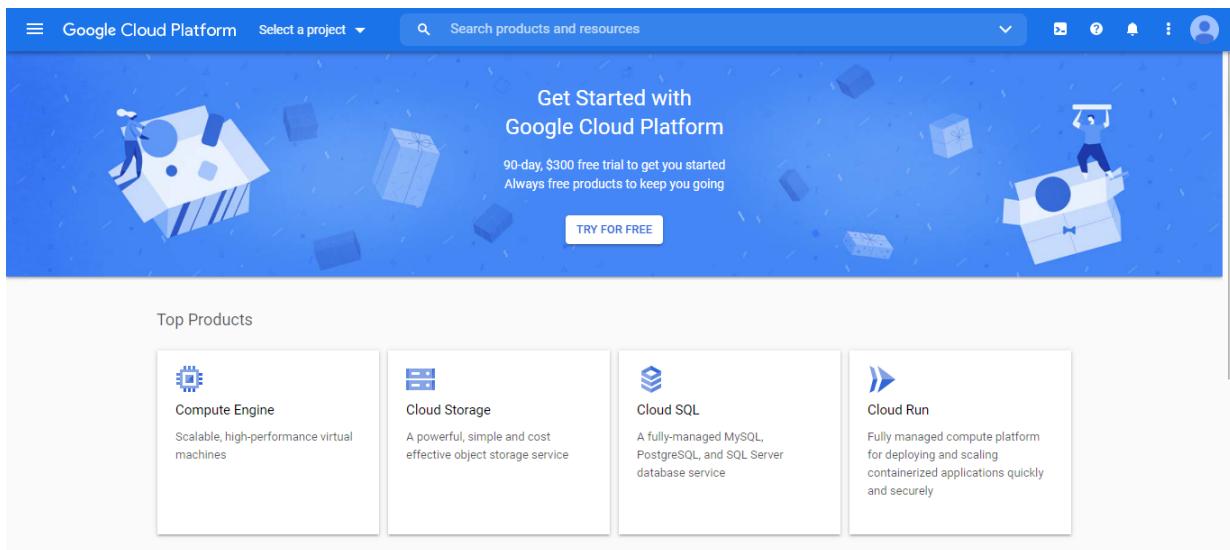


Gambar 3.5: Data Terlihat Pada Dashboard

- 3 Di dalam BigQuery, terdapat salah satu fitur yang digunakan yaitu membuat *dataset* baru.
- 4 *Dataset* bisa saja diambil dari *public dataset* maupun membuat sendiri *dataset* tersebut. *Dataset* berisi
- 5 tabel-tabel yang dianalisis. Tabel-tabel tersebut dapat dibuat secara manual maupun di-*upload*.
- 6 Berikut ini langkah-langkah dalam pembuatan dataset dan tabel:
- 7 1. Membuka Google Cloud Project Page³. Halaman yang ditampilkan dapat dilihat pada gambar
- 8 **3.6**

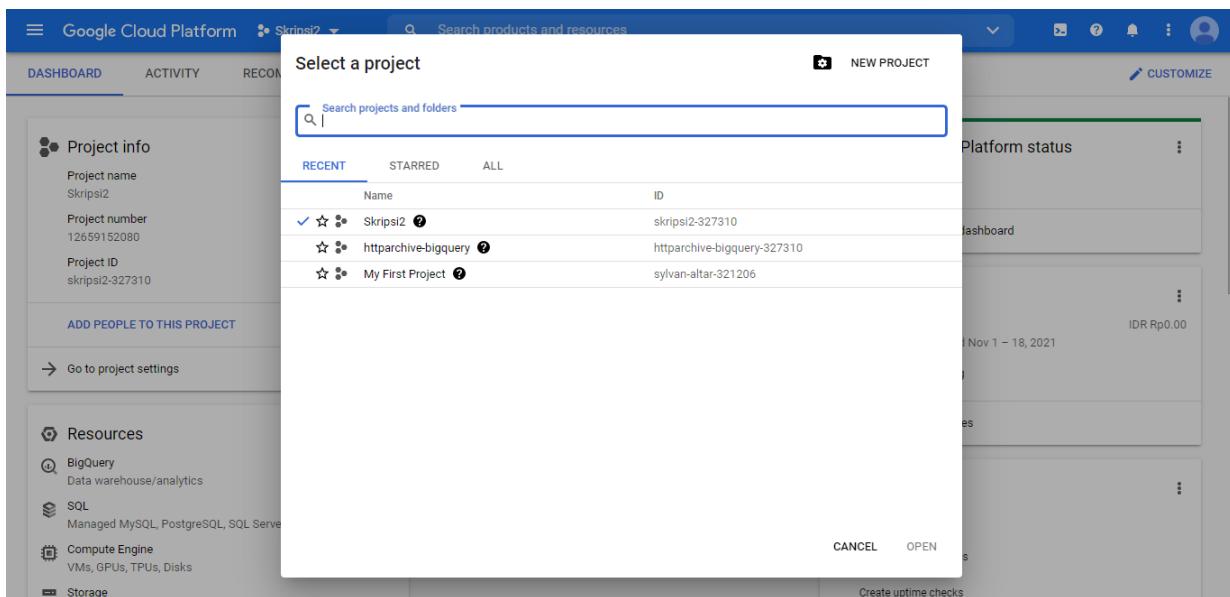
²<https://console.cloud.google.com/bigquery?p=httparchive>

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



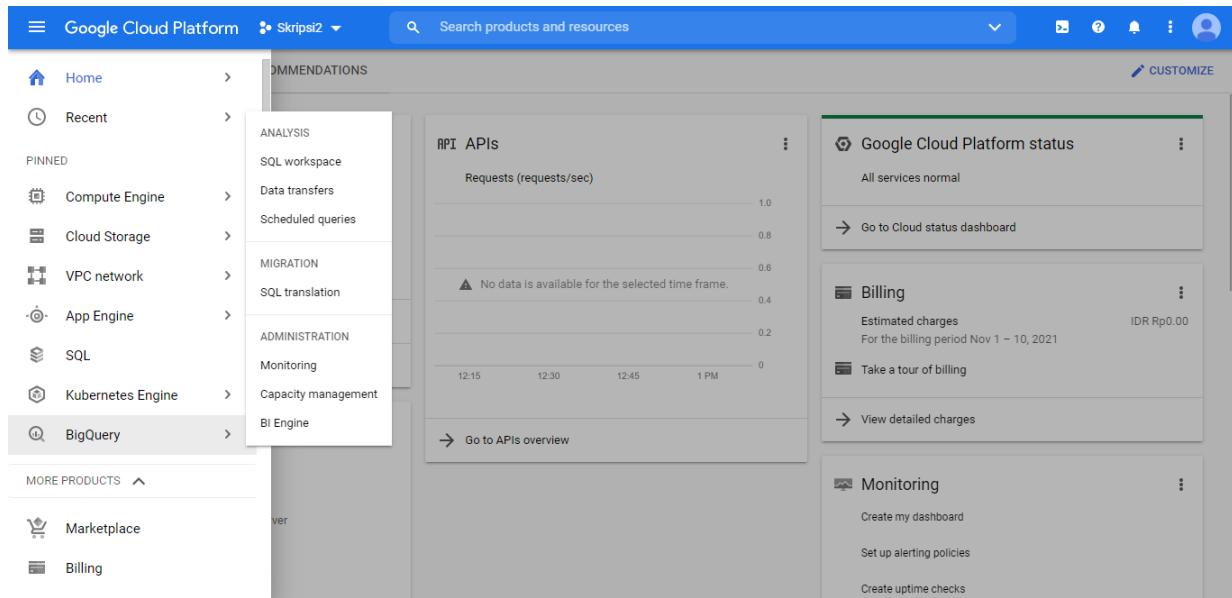
Gambar 3.6: Google Cloud Project Page

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



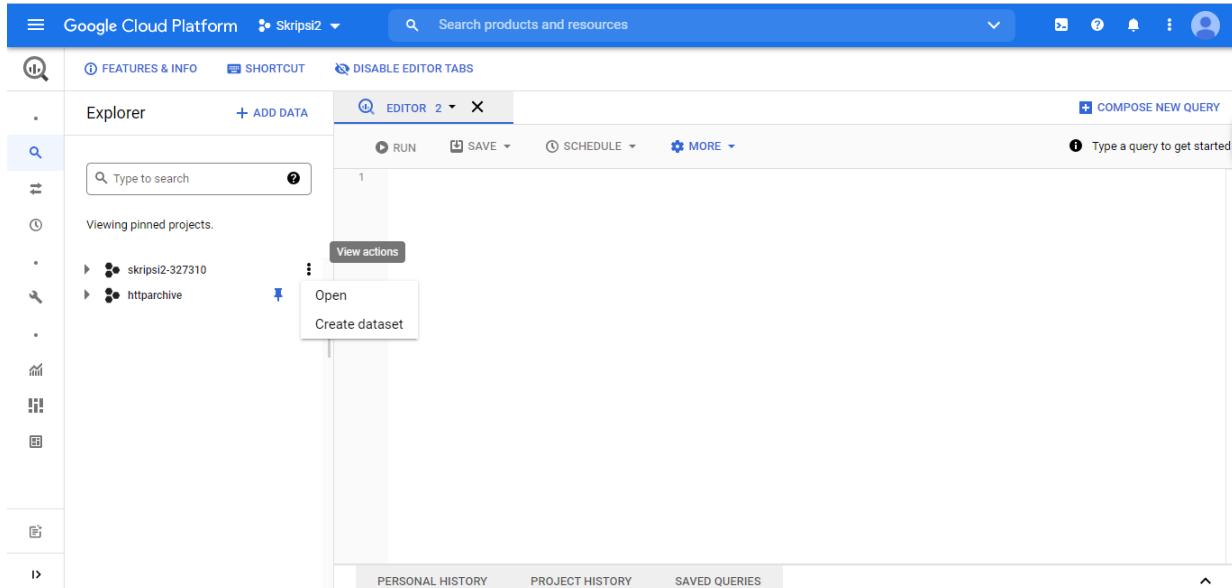
Gambar 3.7: Create atau Open Project

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



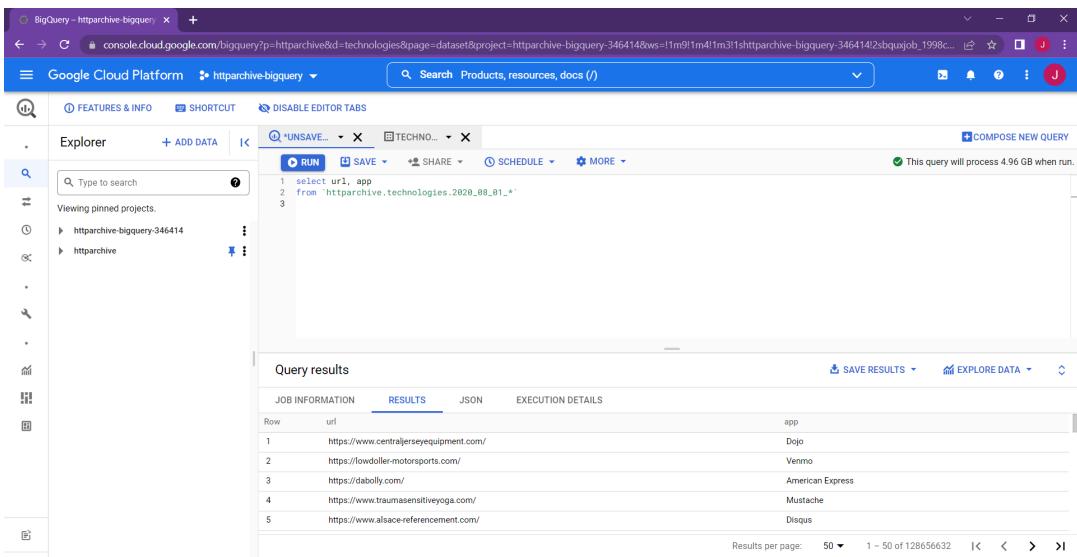
Gambar 3.8: Membuka BigQuery

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



Gambar 3.9: Membuat Dataset Baru

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



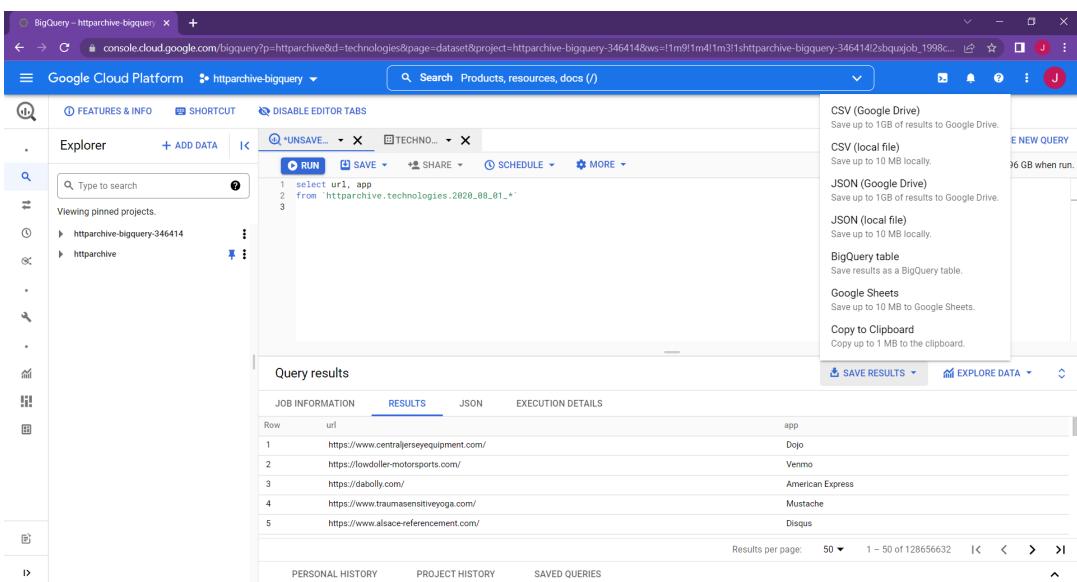
The screenshot shows the Google Cloud Platform BigQuery interface. In the top navigation bar, the project is set to 'BigQuery - httparchive-bigquery'. The main area displays a query results table. The table has a header row with 'Row', 'url', and 'app' columns. The data rows are as follows:

Row	url	app
1	https://www.centraljerseyequipment.com/	Dojo
2	https://lowdoller-motorsports.com/	Venmo
3	https://dabolly.com/	American Express
4	https://www.traumasensitiveyoga.com/	Mustache
5	https://www.alsace-reference.com/	Disqus

At the bottom of the table, it says 'Results per page: 50 1 – 50 of 128656632'.

Gambar 3.10: Membuat Tabel Baru

1 2. Memilih *save result* as BigQuery *Table*



The screenshot shows the Google Cloud Platform BigQuery interface with a query results table. The 'Save' button in the top right is highlighted, and a dropdown menu is open. The menu includes options for saving to Google Drive, local file, Google Sheets, or clipboard, as well as a 'BigQuery table' option. The 'BigQuery table' option is selected, and a tooltip indicates it will save up to 10 MB to Google Drive.

Gambar 3.11: Memilih Save Result As BigQuery Table

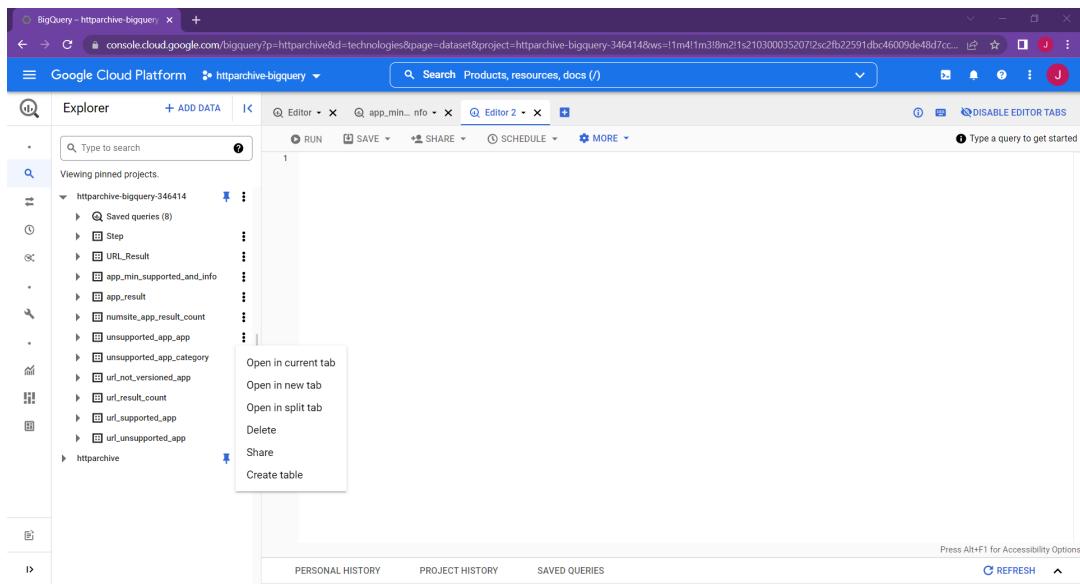
2 3. Memilih lokasi atau *dataset* dan nama tabel untuk disimpan kemudian *export*

Gambar 3.12: Export Table

1. 4. Lokasi tabel dapat dilihat pada *dashboard*

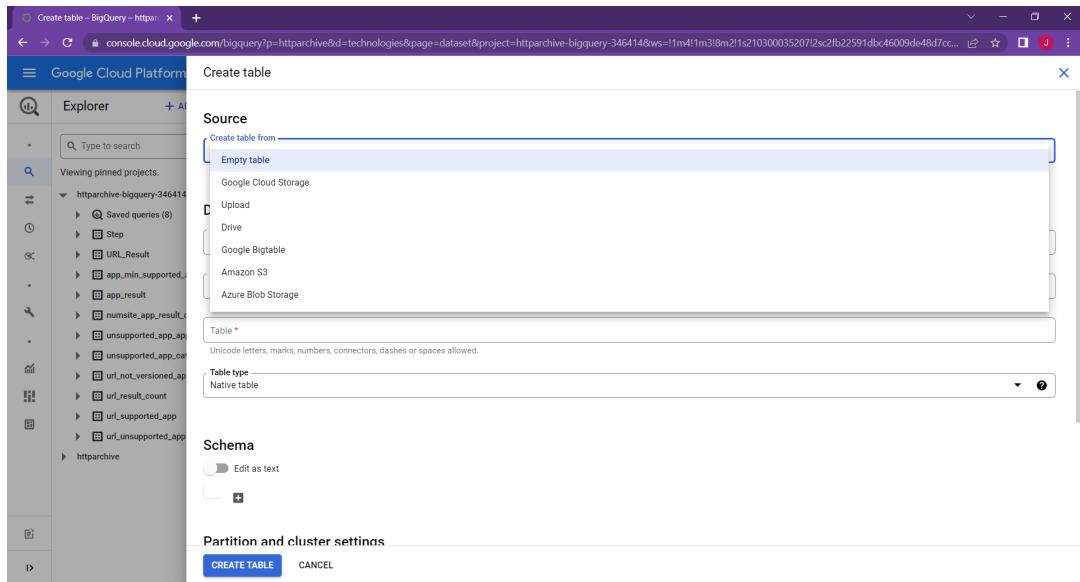
Gambar 3.13: Dashboard Table

2. Selain itu, *table* dapat dibuat dengan meng-*upload* 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??.



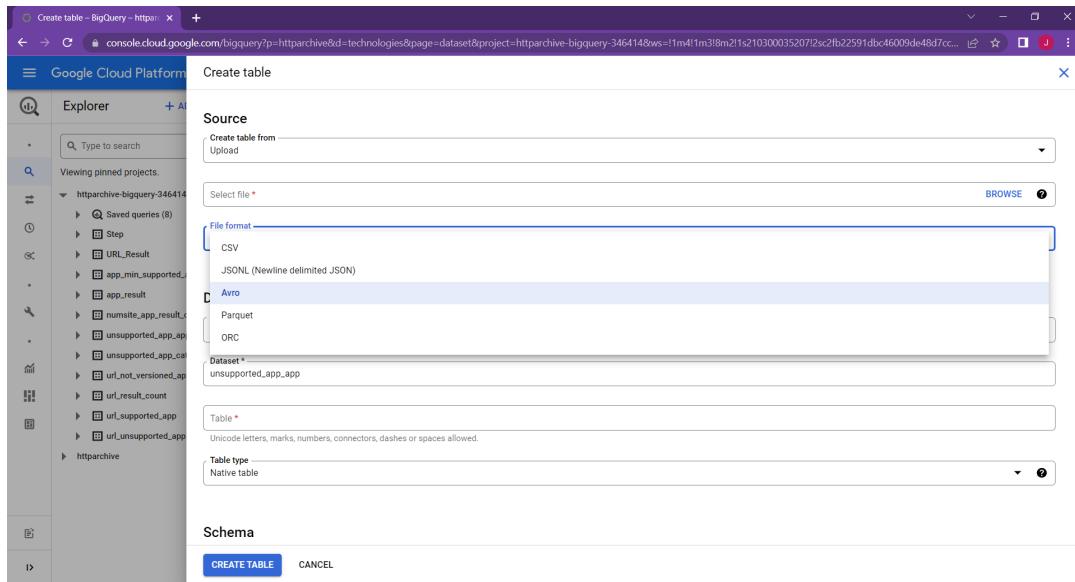
Gambar 3.14: Create Table

- 1 2. Kemudian pilih *upload* pada *field create table from* yang dapat dilihat pada gambar3.15



Gambar 3.15: Pilih Upload

- 2 3. Pilih lokasi *file*, pilih format, dan nama tabel *file* yang dapat dilihat pada gambar3.16



Gambar 3.16: Pilih Lokasi dan Format File

3.2 Dataset yang Digunakan pada HTTPArchive

- Dataset pada HTTPArchive 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.palaciadaslampadas.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

- 6 Pada section ini dijelaskan tentang langkah-langkah *query* yang dilakukan dalam memperoleh data.
- 7 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. *Query* yang digunakan untuk mengumpulkan daftar *website* dapat dilihat pada Gambar 3.1.

Kode 3.1: Mencari Daftar *Website* Limit 10

```

51 SELECT
52     url
53 FROM
54     'httparchive.technologies.2020_08_01_'
55 GROUP BY
56     url
57 LIMIT 10

```

Pada *query* diatas dilakukan pemilihan pada kolom url dengan menggunakan perintah SELECT dari project httparchive dataset *technologies* tabel 2020_08_01_* dengan menggunakan perintah FROM. Mengelompokan pada kolom url yang dilakukan dengan menggunakan perintah GROUP BY sehingga tidak ada nama url yang sama. Kolom dibatasi sebanyak 10 baris dengan menggunakan perintah LIMIT. Sepuluh contoh hasil keluaran dari *query* diatas dapat dilihat pada Tabel 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* dicari aplikasi apa saja yang digunakan dalam pembangunan *website* tersebut dari aplikasi yang dipakainya. *Query* yang digunakan dapat dilihat pada Gambar 3.2.

Kode 3.2: Mencari Aplikasi yang Digunakan Website Limit 10

```

231 SELECT DISTINCT
232     url, app
233 FROM
234     'httparchive.technologies.2020_08_01_'
235 ORDER BY
236     url asc
237 LIMIT 10

```

Pada *query* diatas 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 diurutkan berdasarkan url secara *ascending*. Kolom 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*. *Query* yang digunakan dapat dilihat pada Gambar 3.3.

Kode 3.3: Mengelompokan Semua Aplikasi yang Dipakai

```

4 51  SELECT
6 52    tabelName.app, num.num_sites , versioned.versioned_count , unversioned.unversioned_count
7 53  FROM
8 54  (SELECT DISTINCT
9 55    app
10 56  FROM
11 57    'httparchive.technologies.2020_08_01_*' ) tabelName
12 58
13 59  LEFT JOIN
14 60
15 61  (SELECT
16 62    tabel1.app, count(app) AS versioned_count
17 63  FROM
18 64    'httparchive.technologies.2020_08_01_*' AS tabel1
19 65  WHERE
20 66    tabel1.app!="" AND tabel1.info != ""
21 67  GROUP BY
22 68    tabel1.app) AS versioned
23 69
24 70  ON(versioned.app = tabelName.app)
25 71
26 72  LEFT JOIN
27 73
28 74  (SELECT
29 75    tabel2.app, count(app) AS unversioned_count
30 76  FROM
31 77    'httparchive.technologies.2020_08_01_*' AS tabel2
32 78  WHERE
33 79    tabel2.app!="" AND tabel2.info = ""
34 80  GROUP BY
35 81    tabel2.app) AS unversioned
36 82
37 83  ON (unversioned.app = tabelName.app)
38 84
39 85  LEFT JOIN
40 86
41 87  (SELECT
42 88    app, count(url) AS num_sites
43 89  FROM
44 90    'httparchive.technologies.2020_08_01_*'
45 91  GROUP BY
46 92    app) AS num
47 93
48 94  ON (tabelName.app = num.app)
49 95  LIMIT 10

```

- Pada *query* diatas dibuat beberapa tabel baru yang bersifat sementara. Pada baris ke 3 dan 4, *query* mengembalikan tabel yang berisi semua app yang ada pada tabel menggunakan perintah

1 SELECT dan menggunakan DISTINCT agar app yang ditampilkan hanya keluar satu kali. Data
 2 diambil dari *project httparchive dataset technologies* tabel 2020_08_01_* dengan menggunakan
 3 perintah FROM. Kemudian pada baris ke 8 sampai 11, *query* mengembalikan tabel yang berisi
 4 app dan jumlah app yang memiliki info tidak kosong atau memiliki informasi versi. Pada baris
 5 17 sampai 22, *query* mengembalikan tabel yang berisi app dan jumlah app yang tidak memiliki
 6 informasi versi. Pada baris 26 sampai dengan baris 28, *query* mengembalikan tabel app, jumlah url
 7 yang menggunakan app tersebut. Kemudian semua tabel tersebut digabungkan dengan perintah
 8 LEFT JOIN. Kemudian dengan menggunakan perintah SELECT, dipanggil beberapa variabel dari
 9 setiap kolom dari setiap tabel. Kolom yang diambil berupa: app, jumlah situs yang dipakai aplikasi
 10 (num_sites), jumlah aplikasi yang memiliki versi (versioned_count), dan jumlah aplikasi yang
 11 tidak memiliki versi (unversioned_count). Kolom dibatasi sebanyak 10 baris dengan menggunakan
 12 perintah LIMIT. Sepuluh contoh hasil keluaran dari *query* diatas dapat dilihat pada 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*

13 Pada [2], jumlah data yang digunakan lebih sedikit sehingga jumlah keseluruhan data juga
 14 berbeda. Terdapat beberapa aplikasi yang sama sehingga dapat dibandingkan datanya. Tabel pada
 15 [2] 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	?
Word Press	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 [2]

1 3.3.4 Mencari Data Tentang Versi Aplikasi Yang Masih Didukung

- 2 Sebelum menentukan suatu aplikasi usang atau tidak, kita harus mencari versi dari setiap aplikasi
 3 secara manual. Versi setiap aplikasi dapat dilihat di *official documentation* dari setiap aplikasi.
 4 Hasil pencarian dari aplikasi yang masih didukung dapat dilihat pada Tabel A.

5 3.3.5 Melakukan Perbandingan Antara Versi Aplikasi Yang Masih Dipakai 6 Sekarang Dengan Versi Aplikasi Yang Masih Didukung

- 7 Setelah mendapatkan data versi minimal dari setiap aplikasi, data tersebut dibandingkan dengan
 8 versi aplikasi yang dipakai *url*. *Supported* adalah versi aplikasi dari yang dipakai *url* masih
 9 mendukung atau diatas atau sama dengan versi yang didukung di dokumen. *unsupported* adalah
 10 versi aplikasi dari yang dipakai *url* sudah tidak mendukung atau dibawah versi yang didukung
 11 didokumen. *not_versioned* adalah versi aplikasi dari *url* tidak ditampilkan. *non_conclusive* adalah
 12 versi aplikasi tidak dapat ditentukan. *Query* yang digunakan dapat dilihat pada Gambar 3.4,
 13 Gambar 3.5, Gambar 3.6, dan Gambar 3.7

Kode 3.4: Normalize Semantic Version

```

14 CREATE TEMP FUNCTION normalizedSemanticVersion(semanticVersion STRING)
15 1 AS (((
16 2   SELECT STRING_AGG(
17 3     IF(isDigit, REPEAT('0', 100 - LENGTH(chars)) || chars, chars) ORDER BY grp
18 4   )
19 5   )
20 6   FROM (
21 7     SELECT
22 8       grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,
23 9     FROM (

```

```

10| SELECT
11|   OFFSET, char, isDigit,
12|   COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
13|   FROM
14|     UNNEST(SPLIT(semanticVersion, '')) AS char WITH OFFSET,
15|     UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit
16|
17|   GROUP BY
18|     grp, isDigit
19|   );

```

Kode 3.5: Compare Semantic Version

```

12| CREATE TEMP FUNCTION compareSemanticVersions(
13|   normSemanticVersion1 STRING,
14|   normSemanticVersion2 STRING)
15|   AS (
16|     SELECT CASE
17|       WHEN info < min_supported
18|         THEN 'UNSUPPORTED'
19|       ELSE
20|         'SUPPORTED'
21|       END
22|     FROM UNNEST([STRUCT(
23|       normaizedSemanticVersion(normSemanticVersion1) AS info,
24|       normaizedSemanticVersion(normSemanticVersion2) AS min_supported
25|     )]);

```

Kode 3.6: Membuat Tabel Sementara untuk Grouping Version

```

29| WITH test AS (
30|   SELECT
31|     url, category, app, if (array_length(split(info , ".") > 2, split(info , ".")[offset(0)] || "." || split(info , ".")[offset
32|       (1)], info) as info, min_supported
33|   FROM
34|     'httparchive-bigquery-346414.app_min_supported_and_info.app_min_supported_and_info'
35|   WHERE
36|     info != "\\"
37|   GROUP BY
38|     url, category, app, info, min_supported
39|   )
40| )

```

Kode 3.7: Menampilkan Hasil dari Perbandingan Versi

```

42| SELECT
43|   url, category, app, info, min_supported, if(info = '', "NOT VERSIONED", if(min_supported = '?', 'NON CONCLUSIVE',
44|     compareSemanticVersions(info, min_supported)) ) as result
45|   FROM
46|     test
47|   ORDER BY
48|     url
49| )

```

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

55 Setelah melakukan perbandingan versi tersebut, data diteliti dan berikut ini adalah hasil sepuluh 56 data yang dapat dilihat pada Tabel 3.6. Data diambil berdasarkan banyak aplikasi yang dipakai 57 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

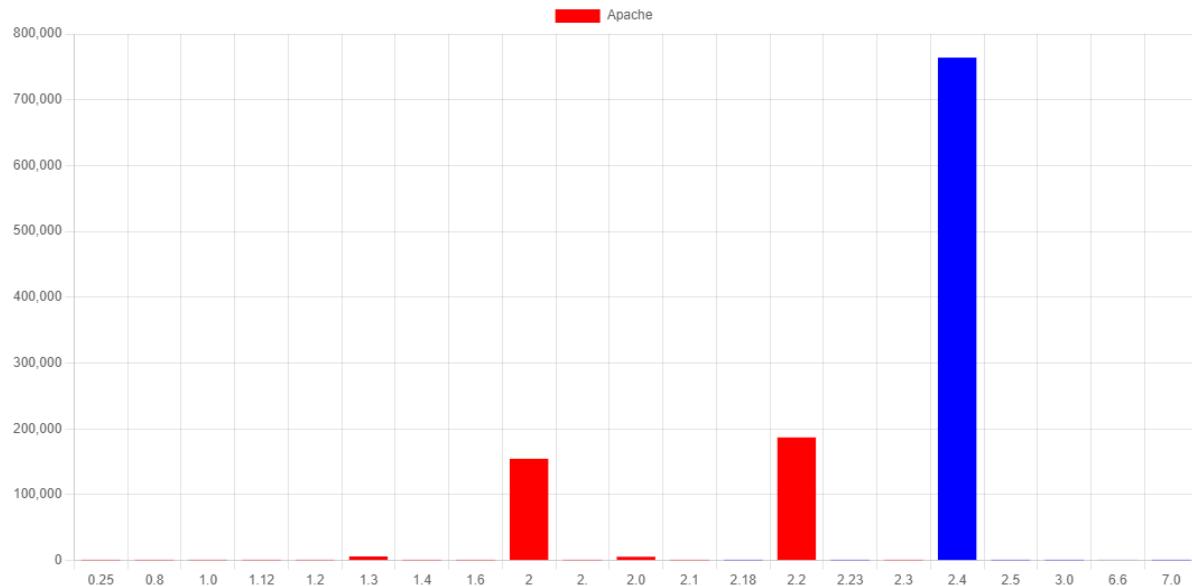
1 Data juga dibandingkan berdasarkan aplikasi tertentu. Data yang dihasilkan adalah *num_sites*
 2 atau jumlah url yang menggunakan aplikasi tertentu, *app*, *supported* atau aplikasi yang masih
 3 didukung, *unsupported* atau aplikasi yang sudah tidak didukung, *not_versioned* atau aplikasi yang
 4 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
6 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 Apache

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

3 Pada bab ini dijelaskan analisis masalah penelitian ini. Analisis meliputi langkah-langkah *query*
4 yang dilakukan dengan data yang lebih besar. *query* yang dilakukan sama dengan bab sebelumnya
5 [3.3](#) tetapi tidak menggunakan *limit*. Pada bab ini juga menampilkan beberapa chart dari aplikasi
6 untuk menunjukkan jumlah penggunaan aplikasi dengan versi yang paling banyak digunakan.]

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

9 Pada *section* ini dijelaskan tentang langkah-langkah *query* yang dilakukan dalam memperoleh data.
10 Data yang diambil adalah semua data yang didapatkan dengan menggunakan *query*. Data yang
11 diambil merupakan *dataset* dari tabel *technologies* 2020-08-01:

¹² 4.1.1 Mengumpulkan *List Website*

13 Langkah pertama yang dilakukan yaitu mengumpulkan *website*. *Website* yang dicari tidak berda-
14 sarkan berdasarkan *rank* karena tidak tersedia pada *dataset* tersebut. *Query* yang digunakan untuk
15 mengumpulkan *list website* dapat dilihat pada Gambar 4.1.

Kode 4.1: Mendapatkan Daftar *Website*

```
16
17 1 SELECT
18 2     url
19 3 FROM
20 4     'httparchive.technologies.2020_08_01_*'
21 5 ORDER BY
22 6     url ASC
```

24 Pada *query* diatas dilakukan pemilihan pada kolom url dengan menggunakan perintah SELECT
25 dari project httparchive dataset *technologies* tabel 2020_08_01_* dengan menggunakan perintah
26 FROM. Mengelompokan pada kolom url yang dilakukan dengan menggunakan perintah GROUP
27 BY sehingga tidak ada nama url yang sama. 10 contoh halaman awal yang dapat dilihat pada Tabel
28 [4.1](#) dan 10 contoh halaman akhir dari hasil keluaran yang dapat dilihat pada Tabel [4.2](#) dari *query*
29 diatas:

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 4.1: 10 Halaman Awal Hasil Pengumpulan Daftar Website

Row	url
7515741	https://zypa.ru/
7515742	https://zyrardow.eglos.pl/
7515743	https://zyrtec.pl/
7515744	https://zytecgermbuster.ca/
7515745	https://zythom.blogspot.com/
7515746	https://zzsso27.wordpress.com/
7515747	https://zzslot8.com/
7515748	https://zzzttt.me/
7515749	https://zzzzzztema.itch.io/
7515750	https://zzzzzzzz9.pixnet.net/

Tabel 4.2: 10 Halaman Akhir Hasil Pengumpulan Daftar Website

4.1.2 Mencari Aplikasi Yang Digunakan *Website*

- Setiap *website* dicari aplikasi apa saja yang digunakan dalam pembangunan *website* tersebut dan versi dari aplikasi yang dipakainya. *Query* yang digunakan dapat dilihat pada Gambar 4.2.

Kode 4.2: Mencari Aplikasi yang Digunakan Website

```

4
51 SELECT
52     url, app
53 FROM
54     'httparchive.technologies.2020_08_01_'
55 ORDER BY
56     url ASC

```

- Pada *query* diatas 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 diurutkan berdasarkan url secara *ascending*. 10 contoh halaman awal yang dapat dilihat pada Tabel 4.3 dan 10 contoh halaman akhir dari hasil keluaran yang dapat dilihat pada Tabel 4.4 dari *query* diatas:

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 4.3: 10 Halaman Awal Contoh Aplikasi Yang Digunakan Website

Row	url	app
75873774	https://zzzzzzzz9.pixnet.net/	Criteo
75873775	https://zzzzzzzz9.pixnet.net/	Babel
75873776	https://zzzzzzzz9.pixnet.net/	YouTube
75873777	https://zzzzzzzz9.pixnet.net/	SWFObject
75873778	https://zzzzzzzz9.pixnet.net/	YouTube
75873779	https://zzzzzzzz9.pixnet.net/	Mixpanel
75873780	https://zzzzzzzz9.pixnet.net/	Google Analytics
75873781	https://zzzzzzzz9.pixnet.net/	Google Plus
75873782	https://zzzzzzzz9.pixnet.net/	jQuery
75873783	https://zzzzzzzz9.pixnet.net/	VideoJS

Tabel 4.4: 10 Halaman Akhir Contoh Aplikasi Yang Digunakan Website

4.1.3 Mengelompokkan Berdasarkan Nama Semua Aplikasi Yang Dipakai

- 2 Pengelompokan aplikasi dapat dilakukan dengan menggunakan *query*. *query* yang digunakan dapat dilihat pada Gambar 4.3.

Kode 4.3: Mengelompokan Semua Aplikasi yang Dipakai

```

4
5 1  SELECT
6 2      tabelName.app, num.num_sites , versioned.versioned_count , unversioned.unversioned_count
7 3  FROM (
8 4      SELECT DISTINCT
9 5          app
10 6      FROM
11 7          'httparchive.technologies.2020_08_01_*' ) tabelName
12 8
13 9  LEFT JOIN
14 0
15 1  (
16 2      SELECT
17 3          tabel1.app, count(app) AS versioned_count
18 4      FROM
19 5          'httparchive.technologies.2020_08_01_*' AS tabel1
20 6      WHERE
21 7          tabel1.app!="" AND tabel1.info != ""
22 8      GROUP BY
23 9          tabel1.app) AS versioned
24 0
25 1  ON(versioned.app = tabelName.app)
26 2
27 3  LEFT JOIN
28 4  (

```

```

26   SELECT
27     tabel2.app, count(app) AS unversioned_count
28   FROM
29     'httparchive.technologies.2020_08_01_*' AS tabel2
30   WHERE
31     tabel2.app!="" AND tabel2.info = ""
32   GROUP BY
33     tabel2.app) AS unversioned
34
35   ON (unversioned.app = tabelName.app)
36
37   LEFT JOIN
38
39   (
40   SELECT
41     app, count(url) AS num_sites
42   FROM
43     'httparchive.technologies.2020_08_01_*'
44   GROUP BY
45     app) AS num
46
47   ON (tabelName.app = num.app)

```

Pada *query* diatas dibuat beberapa tabel baru yang bersifat sementara. Pada baris ke 3 dan 4, *query* 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 8 sampai 11, *query* mengembalikan tabel yang berisi app dan jumlah app yang memiliki info tidak kosong atau memiliki informasi versi. Pada baris 17 sampai 22, *query* mengembalikan tabel yang berisi app dan jumlah app yang tidak memiliki informasi versi. Pada baris 26 sampai dengan baris 28, *query* mengembalikan tabel app, jumlah url yang menggunakan app tersebut. Kemudian semua tabel tersebut digabungkan dengan perintah LEFT JOIN. Kemudian dengan menggunakan perintah SELECT, dipanggil beberapa variabel dari setiap kolom dari setiap tabel. Kolom yang diambil berupa: app, jumlah situs yang dipakai aplikasi (num_sites), jumlah aplikasi yang memiliki versi (versioned_count), dan jumlah aplikasi yang tidak memiliki versi (unversioned_count). 10 contoh halaman awal yang dapat dilihat pada Tabel 4.5 dan 10 contoh halaman akhir dari hasil keluaran yang dapat dilihat pada Tabel 4.6 dari *query* diatas:

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 4.5: 10 Halaman Awal Hasil Pengelompokan Aplikasi Beserta Jumlah *Versioned* Dan *Unversioned*

Row	app	num_sites	versioned_count	unversioned_count
1327	Raphael	2	null	2
1328	Avangate	2	null	2
1329	GoAhead	3	null	3
1330	Banshee	2	null	2
1331	Veoxa	7	null	7
1332	jQuery Sparkline	2	null	2
1333	animate.cs	1	null	1
1334	jQuery U	1	null	1
1335	Pars Elecom Portal	1	null	1
1336	Highchart	1	null	1

Tabel 4.6: 10 Halaman Akhir Hasil Pengelompokan Aplikasi Beserta Jumlah *Versioned* Dan *Unversioned*

1 4.1.4 Mencari Data Tentang Versi Aplikasi Yang Masih Didukung

- 2 Sebelum menentukan suatu aplikasi usang atau tidak, kita harus mencari versi dari setiap aplikasi
 - 3 secara manual. Versi setiap aplikasi dapat dilihat di-*official documentation* dari setiap aplikasi.
 - 4 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

- 7 Setelah mendapatkan data versi minimal dari setiap aplikasi, data tersebut dibandingkan dengan
8 versi aplikasi yang dipakai *url. supported* adalah versi aplikasi dari yang dipakai url masih mendukung
9 atau diatas atau sama dengan versi yang didukung didokumen. *unsupported* adalah versi aplikasi
10 dari yang dipakai url sudah tidak mendukung atau dibawah versi yang didukung didokumen.
11 *not_versioned* adalah versi aplikasi dari url tidak ditampilkan. *non_conclusive* adalah versi aplikasi
12 tidak dapat ditentukan. Data diambil berdasarkan banyak aplikasi yang dipakai oleh url tertentu.
13 Data yang sudah dibandingkan juga digunakan untuk mencari jumlah *website* yang jumlah semua
14 aplikasinya yang masih didukung. Terdapat 4.280 jumlah aplikasi yang digunakan *website*. *Query*
15 yang digunakan untuk mencari datanya dapat diliha pada Gambar 4.4:

Kode 4.4: Jumlah Website yang Jumlah Semua Aplikasi *Supported*

```
16
17 1 SELECT
18 2     count(url1.url) as jumlah
19 3 FROM
20 4 (
21 5 SELECT
22 6     url, count(app) AS jumlah1
23 7 FROM
24 8     'httparchive-bigquery-346414.app_result.app_result'
25 9 WHERE
26 10     result = "SUPPORTED"
27 11 GROUP BY
28 12     url
29 13 ORDER BY
30 14     url ASC
31 15 ) AS url1
32 16
33 17 JOIN
34 18
35 19 (
36 20 SELECT
37 21     url, count(app) AS jumlah2
38 22
39 23
40 24
41 25
42 26
43 27
44 28
45 29
46 30
47 31
48 32
49 33
50 34
51 35
52 36
53 37
54 38
55 39
56 40
57 41
58 42
59 43
60 44
61 45
62 46
63 47
64 48
65 49
66 50
67 51
68 52
69 53
70 54
71 55
72 56
73 57
74 58
75 59
76 60
77 61
78 62
79 63
80 64
81 65
82 66
83 67
84 68
85 69
86 70
87 71
88 72
89 73
90 74
91 75
92 76
93 77
94 78
95 79
96 80
97 81
98 82
99 83
100 84
101 85
102 86
103 87
104 88
105 89
106 90
107 91
108 92
109 93
110 94
111 95
112 96
113 97
114 98
115 99
116 100
117 101
118 102
119 103
120 104
121 105
122 106
123 107
124 108
125 109
126 110
127 111
128 112
129 113
130 114
131 115
132 116
133 117
134 118
135 119
136 120
137 121
138 122
139 123
140 124
141 125
142 126
143 127
144 128
145 129
146 130
147 131
148 132
149 133
150 134
151 135
152 136
153 137
154 138
155 139
156 140
157 141
158 142
159 143
160 144
161 145
162 146
163 147
164 148
165 149
166 150
167 151
168 152
169 153
170 154
171 155
172 156
173 157
174 158
175 159
176 160
177 161
178 162
179 163
180 164
181 165
182 166
183 167
184 168
185 169
186 170
187 171
188 172
189 173
190 174
191 175
192 176
193 177
194 178
195 179
196 180
197 181
198 182
199 183
200 184
201 185
202 186
203 187
204 188
205 189
206 190
207 191
208 192
209 193
210 194
211 195
212 196
213 197
214 198
215 199
216 200
217 201
218 202
219 203
220 204
221 205
222 206
223 207
224 208
225 209
226 210
227 211
228 212
229 213
230 214
231 215
232 216
233 217
234 218
235 219
236 220
237 221
238 222
239 223
240 224
241 225
242 226
243 227
244 228
245 229
246 230
247 231
248 232
249 233
250 234
251 235
252 236
253 237
254 238
255 239
256 240
257 241
258 242
259 243
260 244
261 245
262 246
263 247
264 248
265 249
266 250
267 251
268 252
269 253
270 254
271 255
272 256
273 257
274 258
275 259
276 260
277 261
278 262
279 263
280 264
281 265
282 266
283 267
284 268
285 269
286 270
287 271
288 272
289 273
290 274
291 275
292 276
293 277
294 278
295 279
296 280
297 281
298 282
299 283
300 284
301 285
302 286
303 287
304 288
305 289
306 290
307 291
308 292
309 293
310 294
311 295
312 296
313 297
314 298
315 299
316 300
317 301
318 302
319 303
320 304
321 305
322 306
323 307
324 308
325 309
326 310
327 311
328 312
329 313
330 314
331 315
332 316
333 317
334 318
335 319
336 320
337 321
338 322
339 323
340 324
341 325
342 326
343 327
344 328
345 329
346 330
347 331
348 332
349 333
350 334
351 335
352 336
353 337
354 338
355 339
356 340
357 341
358 342
359 343
360 344
361 345
362 346
363 347
364 348
365 349
366 350
367 351
368 352
369 353
370 354
371 355
372 356
373 357
374 358
375 359
376 360
377 361
378 362
379 363
380 364
381 365
382 366
383 367
384 368
385 369
386 370
387 371
388 372
389 373
390 374
391 375
392 376
393 377
394 378
395 379
396 380
397 381
398 382
399 383
400 384
401 385
402 386
403 387
404 388
405 389
406 390
407 391
408 392
409 393
410 394
411 395
412 396
413 397
414 398
415 399
416 400
417 401
418 402
419 403
420 404
421 405
422 406
423 407
424 408
425 409
426 410
427 411
428 412
429 413
430 414
431 415
432 416
433 417
434 418
435 419
436 420
437 421
438 422
439 423
440 424
441 425
442 426
443 427
444 428
445 429
446 430
447 431
448 432
449 433
450 434
451 435
452 436
453 437
454 438
455 439
456 440
457 441
458 442
459 443
460 444
461 445
462 446
463 447
464 448
465 449
466 450
467 451
468 452
469 453
470 454
471 455
472 456
473 457
474 458
475 459
476 460
477 461
478 462
479 463
480 464
481 465
482 466
483 467
484 468
485 469
486 470
487 471
488 472
489 473
490 474
491 475
492 476
493 477
494 478
495 479
496 480
497 481
498 482
499 483
500 484
501 485
502 486
503 487
504 488
505 489
506 490
507 491
508 492
509 493
510 494
511 495
512 496
513 497
514 498
515 499
516 500
517 501
518 502
519 503
520 504
521 505
522 506
523 507
524 508
525 509
526 510
527 511
528 512
529 513
530 514
531 515
532 516
533 517
534 518
535 519
536 520
537 521
538 522
539 523
540 524
541 525
542 526
543 527
544 528
545 529
546 530
547 531
548 532
549 533
550 534
551 535
552 536
553 537
554 538
555 539
556 540
557 541
558 542
559 543
560 544
561 545
562 546
563 547
564 548
565 549
566 550
567 551
568 552
569 553
570 554
571 555
572 556
573 557
574 558
575 559
576 560
577 561
578 562
579 563
580 564
581 565
582 566
583 567
584 568
585 569
586 570
587 571
588 572
589 573
590 574
591 575
592 576
593 577
594 578
595 579
596 580
597 581
598 582
599 583
600 584
601 585
602 586
603 587
604 588
605 589
606 590
607 591
608 592
609 593
610 594
611 595
612 596
613 597
614 598
615 599
616 600
617 601
618 602
619 603
620 604
621 605
622 606
623 607
624 608
625 609
626 610
627 611
628 612
629 613
630 614
631 615
632 616
633 617
634 618
635 619
636 620
637 621
638 622
639 623
640 624
641 625
642 626
643 627
644 628
645 629
646 630
647 631
648 632
649 633
650 634
651 635
652 636
653 637
654 638
655 639
656 640
657 641
658 642
659 643
660 644
661 645
662 646
663 647
664 648
665 649
666 650
667 651
668 652
669 653
670 654
671 655
672 656
673 657
674 658
675 659
676 660
677 661
678 662
679 663
680 664
681 665
682 666
683 667
684 668
685 669
686 670
687 671
688 672
689 673
690 674
691 675
692 676
693 677
694 678
695 679
696 680
697 681
698 682
699 683
700 684
701 685
702 686
703 687
704 688
705 689
706 690
707 691
708 692
709 693
710 694
711 695
712 696
713 697
714 698
715 699
716 700
717 701
718 702
719 703
720 704
721 705
722 706
723 707
724 708
725 709
726 710
727 711
728 712
729 713
730 714
731 715
732 716
733 717
734 718
735 719
736 720
737 721
738 722
739 723
740 724
741 725
742 726
743 727
744 728
745 729
746 730
747 731
748 732
749 733
750 734
751 735
752 736
753 737
754 738
755 739
756 740
757 741
758 742
759 743
760 744
761 745
762 746
763 747
764 748
765 749
766 750
767 751
768 752
769 753
770 754
771 755
772 756
773 757
774 758
775 759
776 760
777 761
778 762
779 763
780 764
781 765
782 766
783 767
784 768
785 769
786 770
787 771
788 772
789 773
790 774
791 775
792 776
793 777
794 778
795 779
796 780
797 781
798 782
799 783
800 784
801 785
802 786
803 787
804 788
805 789
806 790
807 791
808 792
809 793
810 794
811 795
812 796
813 797
814 798
815 799
816 800
817 801
818 802
819 803
820 804
821 805
822 806
823 807
824 808
825 809
826 810
827 811
828 812
829 813
830 814
831 815
832 816
833 817
834 818
835 819
836 820
837 821
838 822
839 823
840 824
841 825
842 826
843 827
844 828
845 829
846 830
847 831
848 832
849 833
850 834
851 835
852 836
853 837
854 838
855 839
856 840
857 841
858 842
859 843
860 844
861 845
862 846
863 847
864 848
865 849
866 850
867 851
868 852
869 853
870 854
871 855
872 856
873 857
874 858
875 859
876 860
877 861
878 862
879 863
880 864
881 865
882 866
883 867
884 868
885 869
886 870
887 871
888 872
889 873
890 874
891 875
892 876
893 877
894 878
895 879
896 880
897 881
898 882
899 883
900 884
901 885
902 886
903 887
904 888
905 889
906 890
907 891
908 892
909 893
910 894
911 895
912 896
913 897
914 898
915 899
916 900
917 901
918 902
919 903
920 904
921 905
922 906
923 907
924 908
925 909
926 910
927 911
928 912
929 913
930 914
931 915
932 916
933 917
934 918
935 919
936 920
937 921
938 922
939 923
940 924
941 925
942 926
943 927
944 928
945 929
946 930
947 931
948 932
949 933
950 934
951 935
952 936
953 937
954 938
955 939
956 940
957 941
958 942
959 943
960 944
961 945
962 946
963 947
964 948
965 949
966 950
967 951
968 952
969 953
970 954
971 955
972 956
973 957
974 958
975 959
976 960
977 961
978 962
979 963
980 964
981 965
982 966
983 967
984 968
985 969
986 970
987 971
988 972
989 973
990 974
991 975
992 976
993 977
994 978
995 979
996 980
997 981
998 982
999 983
1000 984
1001 985
1002 986
1003 987
1004 988
1005 989
1006 990
1007 991
1008 992
1009 993
1010 994
1011 995
1012 996
1013 997
1014 998
1015 999
1016 1000
1017 1001
1018 1002
1019 1003
1020 1004
1021 1005
1022 1006
1023 1007
1024 1008
1025 1009
1026 1010
1027 1011
1028 1012
1029 1013
1030 1014
1031 1015
1032 1016
1033 1017
1034 1018
1035 1019
1036 1020
1037 1021
1038 1022
1039 1023
1040 1024
1041 1025
1042 1026
1043 1027
1044 1028
1045 1029
1046 1030
1047 1031
1048 1032
1049 1033
1050 1034
1051 1035
1052 1036
1053 1037
1054 1038
1055 1039
1056 1040
1057 1041
1058 1042
1059 1043
1060 1044
1061 1045
1062 1046
1063 1047
1064 1048
1065 1049
1066 1050
1067 1051
1068 1052
1069 1053
1070 1054
1071 1055
1072 1056
1073 1057
1074 1058
1075 1059
1076 1060
1077 1061
1078 1062
1079 1063
1080 1064
1081 1065
1082 1066
1083 1067
1084 1068
1085 1069
1086 1070
1087 1071
1088 1072
1089 1073
1090 1074
1091 1075
1092 1076
1093 1077
1094 1078
1095 1079
1096 1080
1097 1081
1098 1082
1099 1083
1100 1084
1101 1085
1102 1086
1103 1087
1104 1088
1105 1089
1106 1090
1107 1091
1108 1092
1109 1093
1110 1094
1111 1095
1112 1096
1113 1097
1114 1098
1115 1099
1116 1100
1117 1101
1118 1102
1119 1103
1120 1104
1121 1105
1122 1106
1123 1107
1124 1108
1125 1109
1126 1110
1127 1111
1128 1112
1129 1113
1130 1114
1131 1115
1132 1116
1133 1117
1134 1118
1135 1119
1136 1120
1137 1121
1138 1122
1139 1123
1140 1124
1141 1125
1142 1126
1143 1127
1144 1128
1145 1129
1146 1130
1147 1131
1148 1132
1149 1133
1150 1134
1151 1135
1152 1136
1153 1137
1154 1138
1155 1139
1156 1140
1157 1141
1158 1142
1159 1143
1160 1144
1161 1145
1162 1146
1163 1147
1164 1148
1165 1149
1166 1150
1167 1151
1168 1152
1169 1153
1170 1154
1171 1155
1172 1156
1173 1157
1174 1158
1175 1159
1176 1160
1177 1161
1178 1162
1179 1163
1180 1164
1181 1165
1182 1166
1183 1167
1184 1168
1185 1169
1186 1170
1187 1171
1188 1172
1189 1173
1190 1174
1191 1175
1192 1176
1193 1177
1194 1178
1195 1179
1196 1180
1197 1181
1198 1182
1199 1183
1200 1184
1201 1185
1202 1186
1203 1187
1204 1188
1205 1189
1206 1190
1207 1191
1208 1192
1209 1193
1210 1194
1211 1195
1212 1196
1213 1197
1214 1198
1215 1199
1216 1200
1217 1201
1218 1202
1219 1203
1220 1204
1221 1205
1222 1206
1223 1207
1224 1208
1225 1209
1226 1210
1227 1211
1228 1212
1229 1213
1230 1214
1231 1215
1232 1216
1233 1217
1234 1218
1235 1219
1236 1220
1237 1221
1238 1222
1239 1223
1240 1224
1241 1225
1242 1226
1243 1227
1244 1228
1245 1229
1246 1230
1247 1231
1248 1232
1249 1233
1250 1234
1251 1235
1252 1236
1253 1237
1254 1238
1255 1239
1256 1240
1257 1241
1258 1242
1259 1243
1260 1244
1261 1245
1262 1246
1263 1247
1264 1248
1265 1249
1266 1250
1267 1251
1268 1252
1269 1253
1270 1254
1271 1255
1272 1256
1273 1257
1274 1258
1275 1259
1276 1260
1277 1261
1278 1262
1279 1263
1280 1264
1281 1265
1282 1266
1283 1267
1284 1268
1285 1269
1286 1270
1287 1271
1288 1272
1289 1273
1290 1274
1291 1275
1292 1276
1293 1277
1294 1278
1295 1279
1296 1280
1297 1281
1298 1282
1299 1283
1300 1284
1301 1285
1302 1286
1303 1287
1304 1288
1305 1289
1306 1290
1307 1291
1308 1292
1309 1293
1310 1294
1311 1295
1312 1296
1313 1297
1314 1298
1315 1299
1316 1300
1317 1301
1318 1302
1319 1303
1320 1304
1321 1305
1322 1306
1323 1307
1324 1308
1325 1309
1326 1310
1327 1311
1328 1312
1329 1313
1330 1314
1331 1315
1332 1316
1333 1317
1334 1318
1335 1319
1336 1320
1337 1321
1338 1322
1339 1323
1340 1324
1341 1325
1342 1326
1343 1327
1344 1328
1345 1329
1346 1330
1347 1331
1348 1332
1349 1333
1350 1334
1351 1335
1352 1336
1353 1337
1354 1338
1355 1339
1356 1340
1357 1341
1358 1342
1359 1343
1360 1344
1361 1345
1362 1346
1363 1347
1364 1348
1365 1349
1366 1350
1367 1351
1368 1352
1369 1353
1370 1354
1371 1355
1372 1356
1373 1357
1374 1358
1375 1359
1376 1360
1377 1361
1378 1362
1379 1363
1380 1364
1381 1365
1382 1366
1383 1367
1384 1368
1385 1369
1386 1370
1387 1371
1388 1372
1389 1373
1390 1374
1391 1375
1392 1376
1393 1377
1394 1378
1395 1379
1396 1380
1397 1381
1398 1382
1399 1383
1400 1384
1401 1385
1402 1386
1403 1387
1404 1388
1405 1389
1406 1390
1407 1391
1408 1392
1409 1393
1410 1394
1411 1395
1412 1396
1413 1397
1414 1398
1415 1399
1416 1400
1417 1401
1418 1402
1419 1403
1420 1404
1421 1405
1422 1406
1423 1407
1424 1408
1425 1409
1426 1410
1427 1411
1428 1412
1429 1413
1430 1414
1431 1415
1432 1416
1433 1417
1434 1418
1435 1419
1436 1420
1437 1421
1438 1422
1439 1423
1440 1424
1441 1425
1442 1426
1443 1427
1444 1428
1445 1429
1446 1430
1447 1431
1448 1432
1449 1433
1450 1434
1451 1435
1452 1436
1453 1437
1454 1438
1455 1439
1456 1440
1457 1441
1458 1442
1459 1443
1460 1444
1461 1445
1462 14
```

```

22| FROM
23|   'httparchive-bigquery-346414.app_result.app_result'
24| GROUP BY
25|   url
26| ORDER BY
27|   url ASC
28| ) AS url2
29|
30| ON url1.url = url2.url
31| WHERE
32|   url1.jumlah1 = url2.jumlah2

```

13 *Project* httparchive-bigquery-346414 dengan nama *dataset* app_result dan tabel app_result adalah
 14 sebuah tabel pembantu. Tabel ini berasal dari hasil *version compare* pada 3.3.5. *Project* httparchive-
 15 bigquery-346414 ini dibuat berdasarkan data dari *project* httparchive, *dataset technologies*, dan
 16 tabel 2020_08_01_* yang kemudian dibuat tabel baru agar *query* tidak dipanggil beberapa kali.
 17 Pada *query* diatas awalnya dibuat sebuah tabel yang bersifat sementara. Tabel diambil dari *project*
 18 httparchive-bigquery-346414 dengan nama *dataset* app_result dan tabel app_result. Pada tabel ini
 19 dicari url dan data dengan informasi versi dari aplikasi yang masih didukung url tersebut, tabel
 20 diberi nama url1. Kemudian tabel digabungkan dengan tabel lain yang bersifat sementara. Pada
 21 tabel ini dicari semua url dan jumlah aplikasi yang dipakai oleh url tersebut, tabel diberi nama
 22 url2. Hasil akhir dari *query* ini berupa url yang dan jumlah dari tabel url1 dan tabel url2. Hasil
 23 dari *query* dapat dilihat pada Tabel 4.7

Row	jumlah
1	4.280

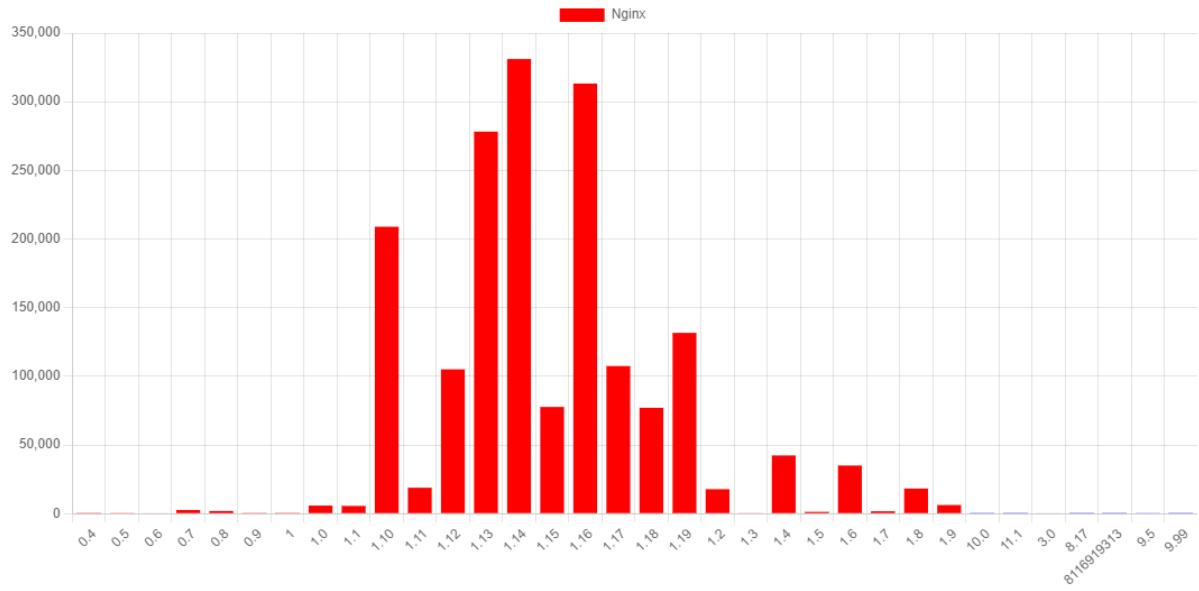
Tabel 4.7: Jumlah *Website* yang Jumlah Semua Aplikasi *Supported*

24 4.2 Hasil Sample Data

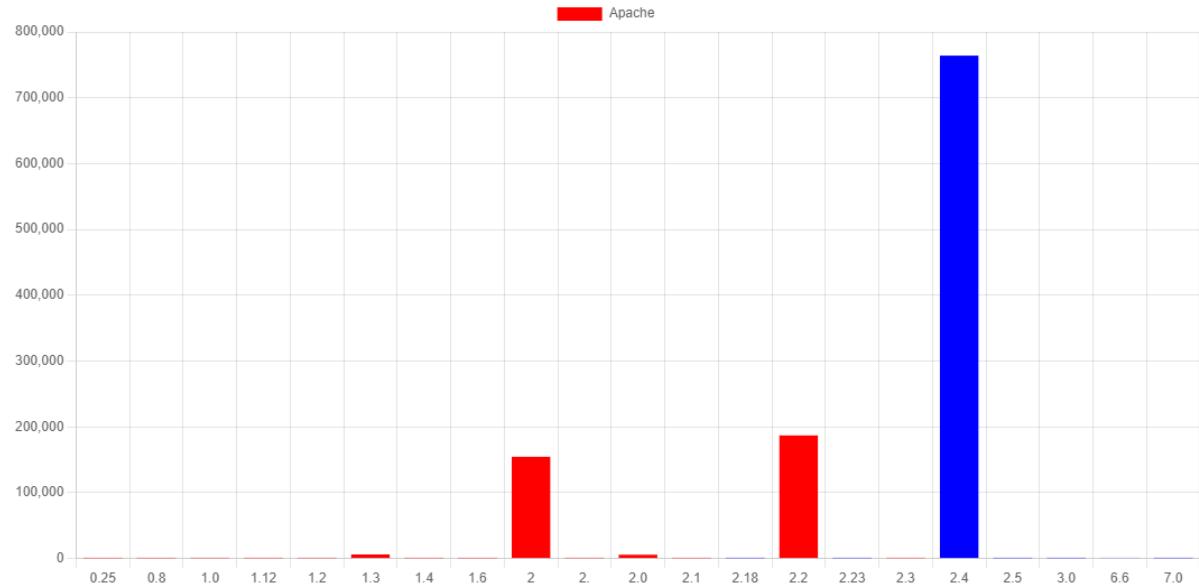
25 Data yang ditampilkan adalah data beberapa aplikasi yang sudah dipisahkan berdasarkan aplikasi
 26 dan nomor versi dari aplikasi yang dipakai serta jumlahnya dalam bentuk *chart*.

27 4.2.1 Apache dan Nginx

28 Apache dan Nginx merupakan dua web servers yang paling banyak digunakan. Pada dua web
 29 server ini, aplikasi Apache memiliki lebih banyak jumlah yang supported daripada aplikasi Nginx.
 30 Pada aplikasi Nginx terdapat 5.440.268 aplikasi yang *unversioned*. Versi pada aplikasi Nginx yang
 31 paling banyak digunakan adalah versi 1.14 dengan jumlah 267.102. Pada aplikasi Apache terdapat
 32 2.949.180 aplikasi yang *unversioned*. Versi pada aplikasi Apache yang paling banyak digunakan
 33 adalah versi 2.4 dengan jumlah 154.533. Berikut ini adalah chart yang dapat dilihat pada Gambar
 34 4.1 dan Gambar 4.2:

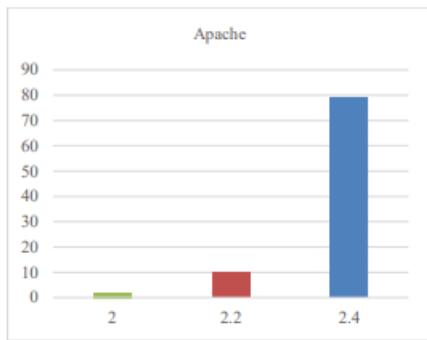


Gambar 4.1: Aplikasi Nginx

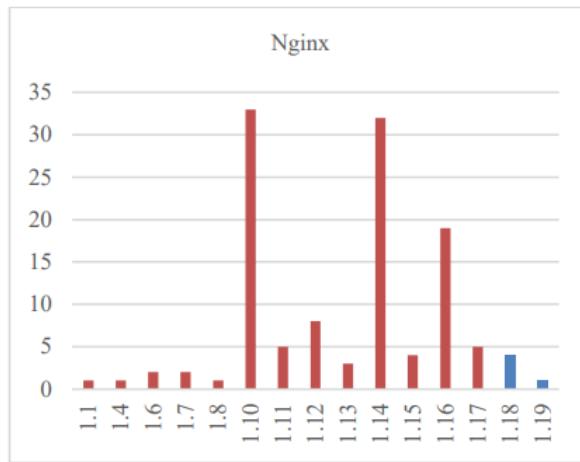


Gambar 4.2: Aplikasi Apache

- 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.



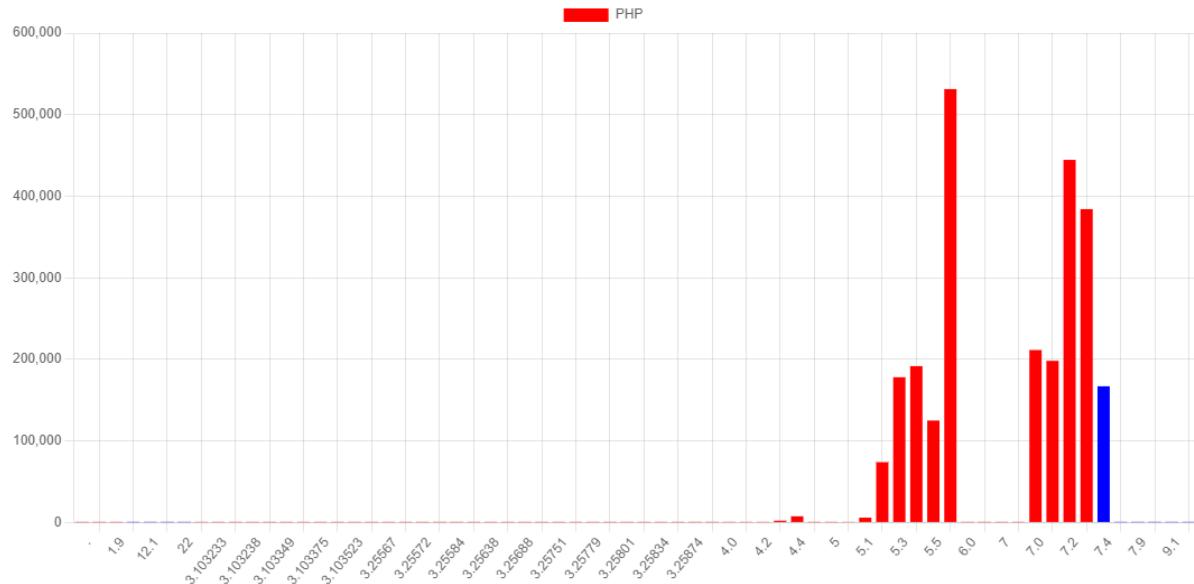
Gambar 4.3: Aplikasi Apache dari [2]



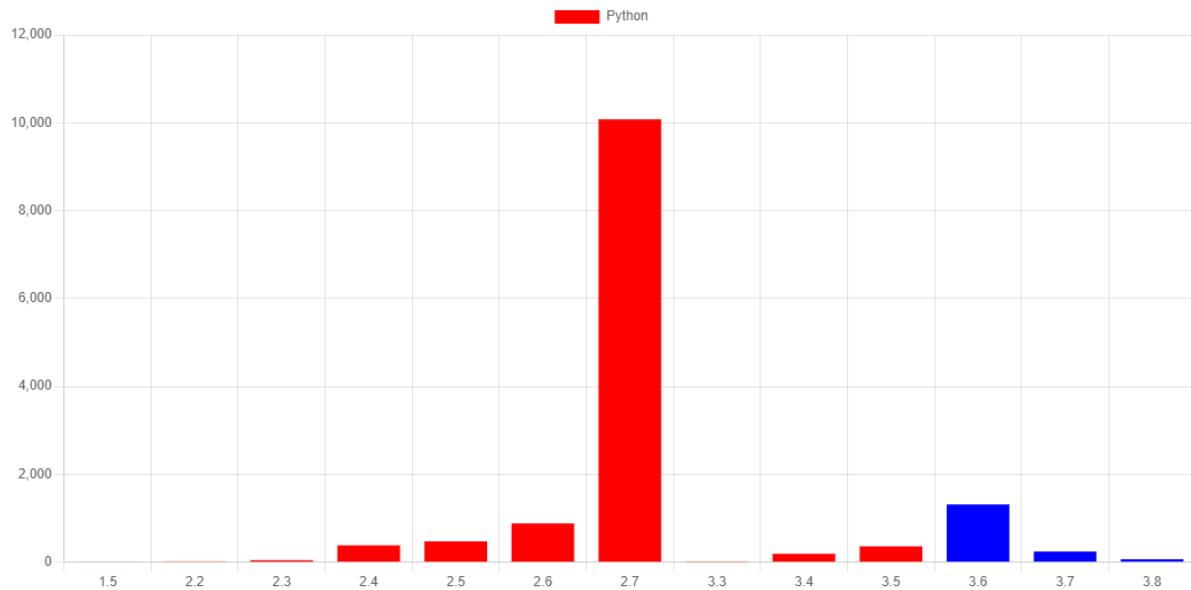
Gambar 4.4: Aplikasi Nginx dari [2]

¹ 4.2.2 PHP dan Python

² PHP merupakan bahasa pemograman yang digunakan dalam pembuatan *website*. PHP menjadi ³ bahasa pemograman 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 pemograman tingkat tinggi dan berorientasi objek. ⁶ Python adalah bahasa pemograman 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 Gambar 4.6:

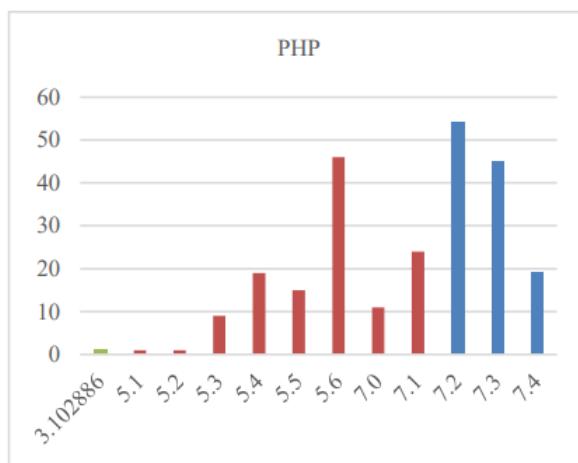


Gambar 4.5: Aplikasi PHP



Gambar 4.6: Aplikasi Python

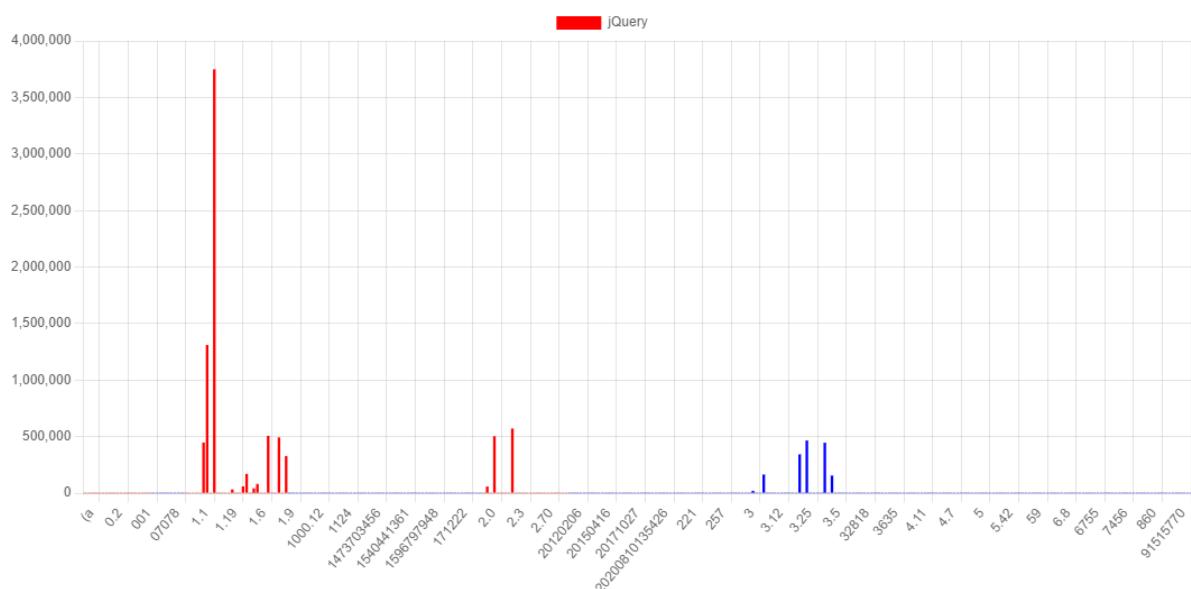
- ¹ Berdasarkan penelitian dengan aplikasi yang sama, didapatkan hasil dalam bentuk *chart*. *Chart*
- ² yang dibandingkan dapat dilihat pada gambar 4.7.



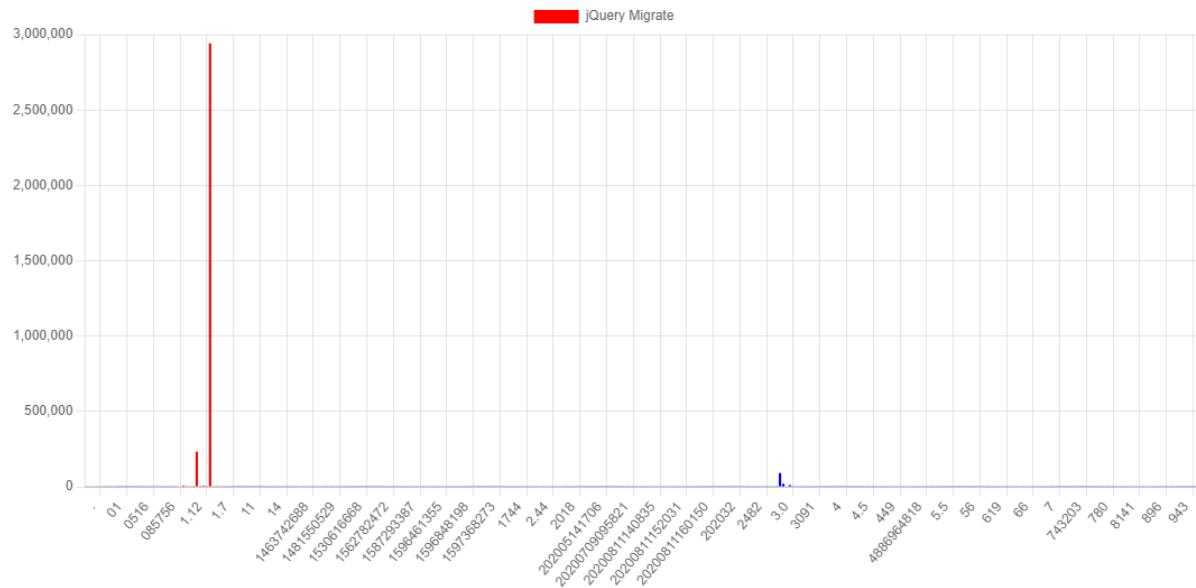
Gambar 4.7: Aplikasi PHP dari [2]

4.2.3 jQuery dan jQuery Migrate

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

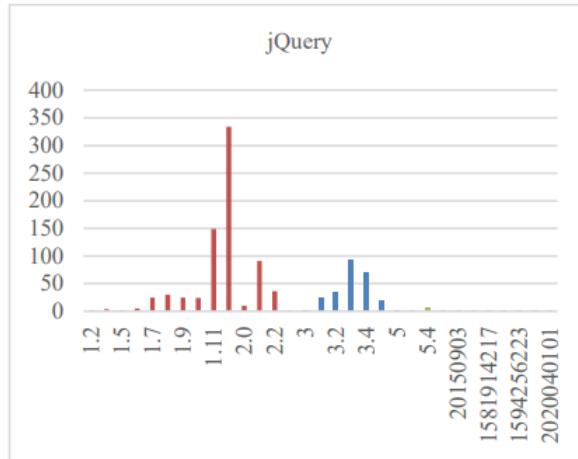


Gambar 4.8: Aplikasi jQuery



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 [2]

BAB 5

PEMBANGUNAN PERANGKAT LUNAK

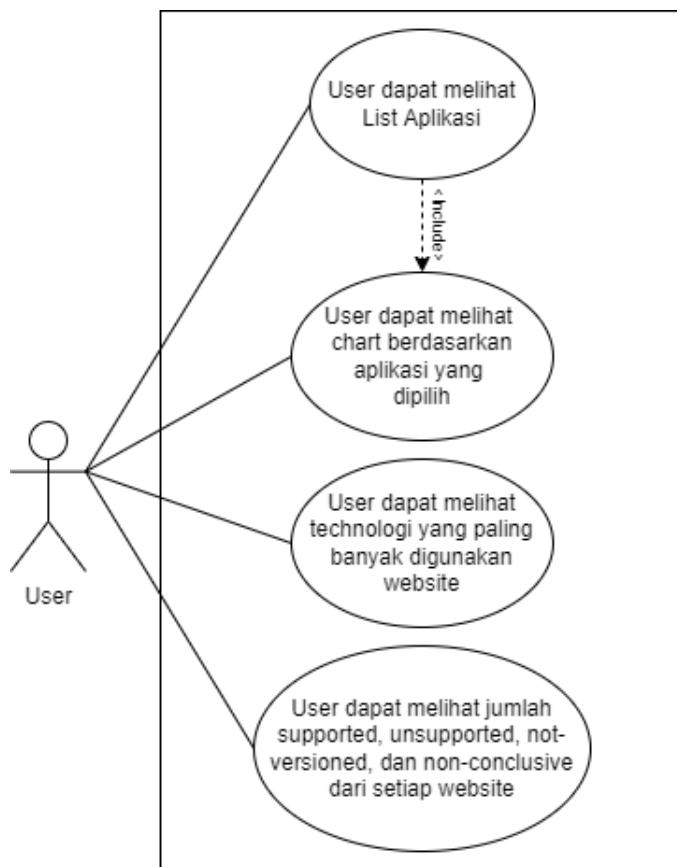
- 3 Pada bab ini dijelaskan tentang analisis dan perancangan perangkat lunak, implementasi perangkat
4 lunak, dan pengujian. Perangkat lunak yang dibangun bertujuan untuk menampilkan data dari
5 HTTPArchive. Data tersebut menunjukkan versi aplikasi yang masih didukung dan tidak didukung
6 dengan menampilkan *chart*. Perangkat lunak juga menampilkan aplikasi yang populer dan url
7 dengan jumlah hasil perbandingan dari setiap aplikasi.

5.1 Analisis dan Perancangan Perangkat Lunak

- ⁹ Pada perangkat lunak dibutuhkan sebuah antarmuka untuk menjalankan fitur-fitur yang sudah
¹⁰ dibuat.

11 5.1.1 UseCase

- 12 Interaksi yang dapat dilakukan dapat dilihat pada *usecase* berikut yang berada pada Gambar 5.1:



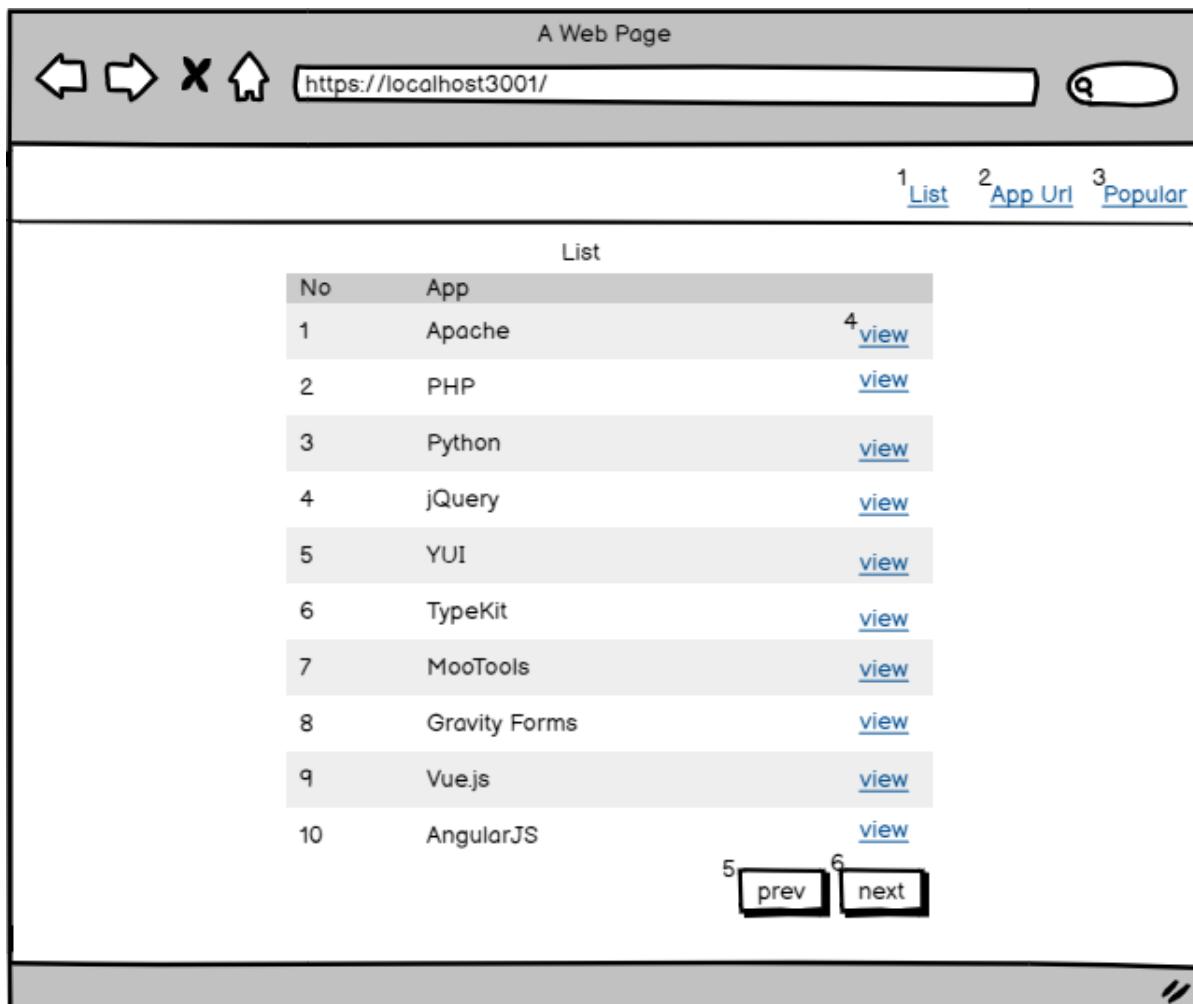
Gambar 5.1: UseCase Diagram

1 Pada aplikasi ini terdapat empat buah aksi, yaitu:

- 2 1. Pengguna dapat melihat daftar aplikasi.
- 3 2. Setelah daftar aplikasi ditampilkan, pengguna dapat melihat chart berdasarkan aplikasi yang dipilih.
- 4 3. Pengguna dapat melihat teknologi yang paling banyak digunakan *website*.
- 5 4. Pengguna dapat melihat jumlah *supported*, *unsupported*, *not-versioned*, dan *non-conclusive* dari setiap *website*

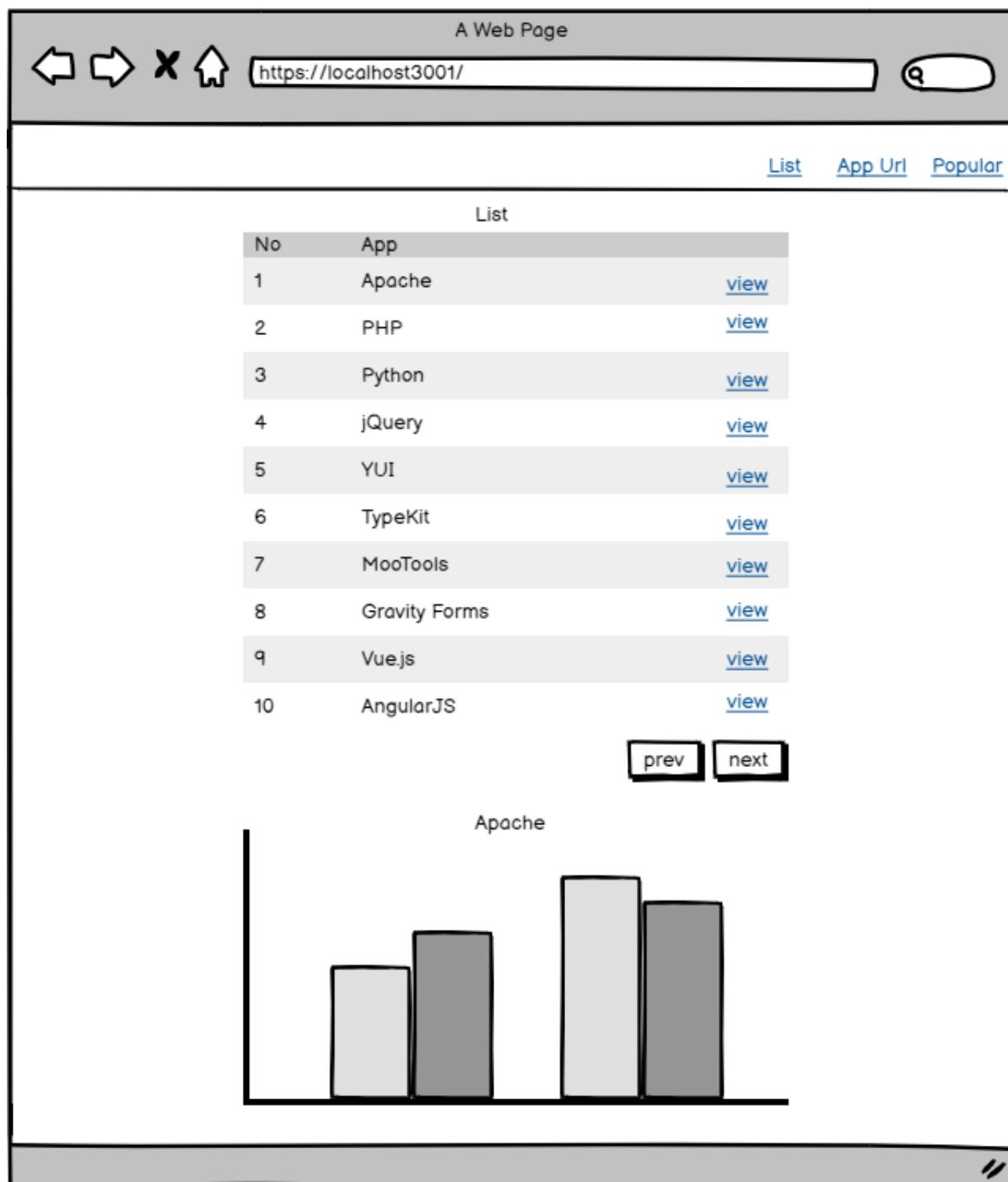
8 5.1.2 Perancangan Antarmuka

9 Antarmuka yang dibuat digunakan untuk mempermudah interaksi antara pengguna dan perangkat lunak yang dapat dilihat pada Gambar 5.2, Gambar 5.4, dan Gambar 5.5



Gambar 5.2: Daftar Aplikasi

- 1 Berdasarkan rancangan diatas, berikut adalah fungsi dari setiap komponen dalam antarmuka.
- 2 1. List: berfungsi untuk membuka halaman list aplikasi.
- 3 2. App Url: berfungsi untuk membuka halaman url dengan jumlah aplikasi yang *supported*, *unsupported*, *not-versioned*, dan *non-conclusive*.
- 4 3. Popular: berfungsi untuk membuka halaman aplikasi yang populer.
- 5 4. View: berfungsi untuk melihat *chart* aplikasi tertentu yang dapat dilihat pada Gambar 5.3.
- 6 5. Prev: berfungsi untuk menunjukkan halaman sebelumnya pada tabel.
- 7 6. Next: berfungsi untuk menunjukkan halaman selanjutnya pada tabel.



Gambar 5.3: Menampilkan Chart Ketika Menekan Tombol View

A Web Page					
1 List 2 App Url 3 Popular					
N	url	App Url	support	unsupport	not_version
1	https://www.mentorbaseball.com/		0	2	15
2	https://www.menu.hr/		0	6	8
3	https://www.meokhealthspa.com.		0	1	10
4	https://www.mepa.de/		1	1	9
5	https://www.mercatsantacaterina		0	2	12
6	https://www.mercedes-benz-dur		1	3	14
7	https://www.mercedes-benz.hu/		0	1	8
8	https://www.merhalsa.jp/		0	3	8
9	https://www.meritspor.com.tr/		0	2	9
10	https://www.merlana.lt/		0	2	4
					1
4			prev	next	5

Gambar 5.4: URL dengan Hasil Keseluruhan Aplikasi

1. List: berfungsi untuk membuka halaman list aplikasi.
2. App Url: berfungsi untuk membuka halaman url dengan jumlah aplikasi yang *supported*, *unsupported*, *not-versioned*, dan *non-conclusive*.
3. Popular: berfungsi untuk membuka halaman aplikasi yang populer.
4. Prev: berfungsi untuk menunjukkan halaman sebelumnya pada tabel.
5. Next: berfungsi untuk menunjukkan halaman selanjutnya pada tabel.

A Web Page

<https://localhost3001/>

1 [List](#) 2 [App Url](#) 3 [Popular](#)

N	numsite	app	Popular			
			support	unsupport	not_version	non_conclusi
1	1000303	jQuery	160483	8374167	24029	0
2	8190668	Google Analytics	0	0	8190668	0
3	749464	WordPress	350	4891016	2603276	0
4	7230612	Nginx	652	1789692	5440268	0
5	5977790	PHP	167095	2355525	3455170	0
6	5481111	Google Font API	0	0	5481111	0
7	452982	Google Tag Mana	0	0	4529823	0
8	406738	Apache	764705	353495	2949180	0
9	404734	MySQL	0	0	4047343	0
10	382703	Font Awesome	106643	1719907	1040702	0

4 [prev](#) [next](#) 5

Gambar 5.5: Aplikasi yang Populer

1. List: berfungsi untuk membuka halaman list aplikasi.
2. App Url: berfungsi untuk membuka halaman url dengan jumlah aplikasi yang *supported*, *unsupported*, *not-versioned*, dan *non-conclusive*.
3. Popular: berfungsi untuk membuka halaman aplikasi yang populer.
4. Prev: berfungsi untuk menunjukkan halaman sebelumnya pada tabel.
5. Next: berfungsi untuk menunjukkan halaman selanjutnya pada tabel.

5.2 Implementasi Perangkat Lunak

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

- BigQuery

Bagian ini adalah representasi dari data. Pada BigQuery 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.

- 1 • Node.js
 2 Bagian ini adalah sebuah penghubung antara data dari Bigquery dan web-client.

3 **5.2.1 BigQuery**

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

7 **Minimal Supported Data**

- 8 Minimal supported data didapatkan dengan mencari sendiri informasi versi dari setiap aplikasi
 9 berdasarkan dokumen resminya. Kemudian data-data tersebut dimasukkan kedalam excel dan
 10 didownload sebagai csv. Data-data csv tersebut di-upload menggunakan fitur yang ada pada
 11 BigQuery dan terbentuk tabel yang berisi csv tersebut. Tabel dari excel tersebut dapat dilihat pada
 12 lampiranA

13 **Menyatukan Tabel Informasi Versi dan Minimal Supported**

Kode 5.1: Menyatukan Tabel Informasi Versi dan Minimal Supported

```

14
15 1  SELECT
16 2      jumlah.url, app1.category, app1.app, app1.info, sup.min_supported
17 3  FROM(
18 4      SELECT
19 5         url, count(app) as jumlah
20 6      FROM
21 7         'httparchive.technologies.2020_08_01_*'
22 8      GROUP BY
23 9         url
24 0      ORDER BY
25 1         jumlah desc
26 2  ) AS jumlah
27 3
28 4  JOIN
29 5
30 6  (
31 7      SELECT
32 8         url, category, app, info
33 9      FROM
34 0         'httparchive.technologies.2020_08_01_*'
35 1  ) AS app1
36 2  ON jumlah.url = app1.url
37 3
38 4
39 5  JOIN
40 6
41 7  (
42 8      SELECT DISTINCT
43 9         app, min_supported
44 0      FROM
45 1         'httparchive-bigquery-346414.Step.min_supported_app'
46 2  ) AS sup
47 3  ON app1.app = sup.app
48 4
49 5  ORDER BY
50 6      jumlah.jumlah desc

```

- 52 Query pada Gambar 5.1 terdapat beberapa *query* yang disatukan sehingga membentuk suatu tabel
 53 yang menyatukan informasi versi yang dipakai aplikasi dengan *minimal supported*.

- 54 • Mencari url dan jumlah app yang dipakai url tersebut. *Query* yang dilakukan dapat dilihat
 55 pada Gambar 5.2

Kode 5.2: Mencari URL dan Jumlah APP yang Dipakai

```

1  SELECT
2      url, count(app) as jumlah
3  FROM
4      'httparchive.technologies.2020_08_01_*'
5  GROUP BY
6      url
7  ORDER BY
8      jumlah desc

```

- Mencari url, kategori, app, informasi versi. *Query* yang dilakukan dapat dilihat pada Gambar 5.3

Kode 5.3: Mencari URL, Kategori, APP, dan Informasi Versi

```

1  SELECT
2      url, category, app, info
3  FROM
4      'httparchive.technologies.2020_08_01_*'

```

- Mencari app dan minimal supported. *Query* yang dilakukan dapat dilihat pada Gambar 5.4

Kode 5.4: Mencari APP dan Minimal Supported

```

1  SELECT DISTINCT
2      app, min_supported
3  FROM
4      'httparchive-bigquery-346414.Step.min_supported_app'

```

26 Version Compare

Kode 5.5: Version Compare Query

```

27 1 CREATE TEMP FUNCTION normalizedSemanticVersion(semanticVersion STRING)
28 2 AS (
29 3     SELECT STRING_AGG(
30 4         IF(isDigit, REPEAT('0', 100 - LENGTH(chars)) || chars, chars) ORDER BY grp
31 5     )
32 6     FROM (
33 7         SELECT
34 8             grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,
35 9         FROM (
36 10            SELECT
37 11                OFFSET, char, isDigit,
38 12                COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
39 13            FROM (
40 14                UNNEST(SPLIT(semanticVersion, '')) AS char WITH OFFSET,
41 15                UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit
42 16            )
43 17            GROUP BY
44 18                grp, isDigit
45 19        ))));
46 20 CREATE TEMP FUNCTION compareSemanticVersions(
47 21     normSemanticVersion1 STRING,
48 22     normSemanticVersion2 STRING)
49 23 AS (
50 24     SELECT CASE
51 25     WHEN info < min_supported
52 26     THEN 'UNSUPPORTED'
53 27     ELSE
54 28     'SUPPORTED'
55 29     END
56 30     FROM (
57 31         UNNEST([STRUCT(
58 32             normalizedSemanticVersion(normSemanticVersion1) AS info,
59 33             normalizedSemanticVersion(normSemanticVersion2) AS min_supported
60 34         ]));
61 35     );
62 36     WITH test AS (
63 37         SELECT
64 38             url, category, app, if (array_length(split(info, ".") > 2, split(info, ".")[offset(0)] || "." || split(info, ".")[offset(1)]), info, min_supported
65 39
66 40

```

```

19 FROM
20   'httparchive-bigquery-346414.app_min_supported_and_info.app_min_supported_and_info'
21 WHERE
22   info != "\\"
23 )
24 SELECT
25   url, category, app, info, min_supported, if(info = '', "NOT VERSIONED", if(min_supported = '?', 'NON CONCLUSIVE',
26     compareSemanticVersions(info, min_supported)) ) as result
27 FROM test
28 ORDER BY url

```

Query pada Gambar 5.5 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:

- Memeriksa Digit Dari Versi. *Query* yang dilakukan dapat dilihat pada Gambar 5.6

Kode 5.6: Memeriksa Digit dari Versi

```

19
20   SELECT
21     OFFSET, char, isDigit,
22     COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
23   FROM
24     UNNEST(SPLIT('3.14', '')) AS char WITH OFFSET,
25     UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit

```

Pada *query* ini 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. Hasil dapat dilihat pada Tabel 5.1:

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

Tabel 5.1: Hasil Memotong Setiap Char

- Memotong Setiap Char Dari Version. *Query* dapat dilihat pada Gambar 5.7

Kode 5.7: Memotong Char dari Setiap Version

```

32
33   SELECT
34     grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,
35   FROM (
36     SELECT
37       OFFSET, char, isDigit,
38       COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
39     FROM
40       UNNEST(SPLIT('3.14', '')) AS char WITH OFFSET,
41       UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit
42     )
43   GROUP BY grp, isDigit

```

Pada *query* ini mengembalikan jumlah boolean yang bernilai *false*, *boolean* yang menandakan apakah input merupakan digit atau bukan, dan input yang dibagi-bagi dalam bentuk *string*. Hasil *query* dapat dilihat pada Tabel 5.2:

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

Tabel 5.2: Hasil Memotong Char yang Dipisahkan Titik

- Normalisasi Informasi Versi. *Query* dapat dilihat pada Gambar 5.8

Kode 5.8: Normalisasi Informasi Versi

```

2 CREATE TEMP FUNCTION normalizedSemanticVersion(semanticVersion STRING)
3 AS (
4   SELECT STRING_AGG(
5     IF(isDigit, REPEAT('0', 100 - LENGTH(chars)) || chars, chars) ORDER BY grp
6   )
7   FROM (
8     SELECT
9       grp, isDigit, STRING_AGG(char, '' ORDER BY OFFSET) chars,
10      FROM (
11        SELECT
12          OFFSET, char, isDigit,
13          COUNTIF(NOT isDigit) OVER(ORDER BY OFFSET) AS grp
14        FROM
15        UNNEST(SPLIT(semanticVersion, '')) AS char WITH OFFSET,
16        UNNEST([char IN ('1','2','3','4','5','6','7','8','9','0')]) isDigit
17      )
18      GROUP BY
19        grp, isDigit
20    ));
21

```

Pada *query* ini menghasilkan informasi versi yang sudah dinormalisasikan dengan membuat panjang dari setiap versi sama yang dipisahkan oleh *isDigit*. Pada Tabel 5.3, parameter yang digunakan akalah "3.14". Hasil *query* dapat dilihat pada Tabel 5.3:

Row	f0_
1	00 00 00 00 0014

Tabel 5.3: Hasil Normalisasi Versi

2. Version Compare Function

Pada bagian ini dibuat sebuah fungsi yang digunakan untuk membuat version compare dari informasi versi dari tabel yang sudah dibuat. *Query* dapat dilihat pada Gambar 5.9:

- Membuat Fungsi Untuk Membandingkan Versi

Kode 5.9: Fungsi Version Compare

```

30 CREATE TEMP FUNCTION compareSemanticVersions(
31   normSemanticVersion1 STRING,
32   normSemanticVersion2 STRING)
33 AS (
34   SELECT CASE
35     WHEN info < min_supported
36     THEN 'UNSUPPORTED'
37     ELSE
38       'SUPPORTED'
39     END
40   FROM
41   UNNEST([STRUCT(
42     normSemanticVersion(normSemanticVersion1) AS info,
43

```

```

1      normalizedSemanticVersion(normSemanticVersion2) AS min_supported
2  ]])
3 );

```

5 Pada *query* ini menghasilkan UNSUPPORTED jika info atau informasi versi lebih kecil
6 daripada minimal supported yang sudah ditentukan dan mengembalikan SUPPORTED
7 jika info atau informasi versi lebih besar daripada minimal supported yang sudah
8 ditentukan.

9 3. Membuat Tabel Sementara

10 Pada bagian ini dibuat sebuah tabel sementara yang beguna membuat group untuk menun-
11 jukkan versi yang dihasilkan hanya versi major dan versi minor. *Query* dapat dilihat pada
12 Gambar 5.10

Kode 5.10: Membuat Tabel Sementara

```

13 WITH test AS (
14 SELECT
15   url, category, app, if (array_length(split(info , ".") > 2, split(info , ".")[offset(0)] || "." || split(info , ".")[
16     offset(1)], info) as info, min_supported
17   FROM
18     'httparchive-bigquery-346414.app_min_supported_and_info.app_min_supported_and_info'
19   WHERE
20     info != "\\\""
21 )
22

```

24 Pada *query* ini membuat tabel sementara yag mengembalikan url, kategori dari aplikasi,
25 aplikasi yang dipakai, informasi versi dari aplikasi, dan minimal supported dari aplikasi
26 yang sudah ditentukan. Pada bagian informasi versi sudah dilakukan group sehingga yang
27 dihasilkan hanya *major version* dan *minor version*.

28 4. Menampilkan Semua Hasil Pada bagian ini untuk menampilkan tabel hasil akhir dari *query*
29 yang dilakukan. *Query* dapat dilihat pada Gambar 5.11

Kode 5.11: Menampilkan Semua Hasil

```

30 SELECT
31   url, category, app, info, min_supported, if(info = '', 'NOT VERSIONED', if(min_supported = '?', 'NON CONCLUSIVE',
32     compareSemanticVersions(info, min_supported)) ) as result
33   FROM
34     test
35   ORDER BY
36     url
37

```

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

42 Ketika semua *query* disatukan, berikut adalah 10 contoh hasilnya dapat dilihat pada Tabel ??:

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

2 5.2.2 React.js

3 Bagian ini merupakan bagian tampilan atau *web-client*. Pada bagian ini bertujuan untuk me-
4 nampilkan hasil data atau visualisasi data. Terdapat beberapa file dalam perangkat lunak ini,
5 yaitu:

- Components

7 Didalam foler components terdapat beberapa fungsi yaitu:

- AppUrl.js

9 Pada fungsi ini digunakan untuk membuat tabel yang menampilkan url dengan jumlah
10 dari setiap result (*supported*, *unsupported*, *not versioned*, dan *non conclusive*). Kode
11 dapat dilihat pada Gambar 5.12:

Kode 5.12: Membuat Tabel yang Menampilkan URL dengan Jumlah Setiap Result

```

1 import React, { useEffect, useState } from "react";
2
3 const MAX = 10;
4
5 export default function AppUrl() {
6     const [step, setStep] = useState(0);

```

```

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

```

– List.js

Pada fungsi ini digunakan untuk membuat tabel yang berisi daftar app dan *chart* app yang dipilih. Kode dapat dilihat pada Gambar 5.13:

Kode 5.13: Membuat Tabel dan Menampilkan Chart Berdasarkan App yang Dipilih

```

63
64  1 import React, { useEffect, useState } from "react";
65  2 import { CategoryScale } from "chart.js";
66  3 import { Bar } from "react-chartjs-2";
67  4 import Chart from "chart.js/auto";
68  5
69  6 const MAX = 10;
70  7
71  8 export default function List() {
72  9  const [step, setStep] = useState(0);
73 10  const [data, setData] = useState([]);
74 11  const [isLoading, setIsLoading] = useState(true);
75 12  const [selectedData, setSelectedData] = useState();
76 13
77 14  function color(arr) {

```

```

1      15     let temp = [];
2      16     for (let i = 0; i < arr.length; i++) {
3      17       const dataType = arr[i].result;
4      18       switch (dataType) {
5      19         case "SUPPORTED":
6      20           temp.push("blue");
7      21           break;
8      22         case "UNSUPPORTED":
9      23           temp.push("red");
10     24           break;
11     25         default:
12     26           temp.push("green");
13     27       }
14     28     }
15     29     return temp;
16   30   }
17   31   useEffect(() => {
18   32     Chart.register(CategoryScale);
19   33   }, []);
20
21   35   function getListData(page) {
22     36     setIsLoading(true);
23     37     fetch('http://localhost:3000/get/app/type?limit=${MAX}&offset=${page}')
24     38       .then((res) => res.json())
25     39       .then((data) => {
26     40         setData(data);
27     41         setIsLoading(false);
28     42       });
29   43   }
30
31   45   function getData(name) {
32     46     fetch('http://localhost:3000/get/app/name/${name}')
33     47       .then((res) => res.json())
34     48       .then((data) => {
35     49         setSelectedDate({
36       50           name,
37         data: {
38           52             labels: data.map((val) => String(val.info)),
39           53             datasets: [
40             54               {
41               55                 label: name,
42                 data: data.map((val) => String(val.jumlah)),
43                 backgroundColor: color(data),
44               58             },
45               59             ],
46             60           },
47           61         });
48         62       });
49   63   }
50
51   65   useEffect(() => {
52     66     getListData(step);
53   }, [step]);
54   68   return (
55     69     <div className="container">
56     70       <h2 className="title">List APP</h2>
57     71       {data[0] && (
58         72           <>
59           73             <table className="table table-hover">
60             74               <thead>
61               75                 <tr>
62                 76                   <th className="no">No</th>
63                 77                   <th>Name</th>
64                 78                   <th className="action-head"></th>
65               79                 </tr>
66             80               </thead>
67             81               <tbody>
68               82                 {data.map((val, index) => (
69                 83                   <tr key={index} onClick={() => getData(val.app)}>
70                   84                     <th cla>{index + 1 + MAX * step}</th>
71                     85                     <td>{val.app}</td>
72                     86                     <td>View</td>
73                     87                     </tr>
74                   88                 )));
75               89             </tbody>
76             90           </table>
77           91           <div className="action">
78             92               <button
79               93                 onClick={() => setStep((prev) => prev - 1)}
80                 94                 disabled={step <= 0 || isLoading}
81               95               >
82                 96                 Prev

```

```

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

```

– Popular.js

Pada fungsi ini digunakan untuk membuat tabel yang berisi app yang popular berdasarkan jumlah url yang menggunakan app tersebut. Kode yang digunakan dapat dilihat pada Gambar 5.14:

Kode 5.14: Membuat Tabel App yang Paling Populer

```

26 1 import React, { useEffect, useState } from "react";
27 2
28 3 const MAX = 10;
29 4
30 5 export default function Popular() {
31 6   const [step, setStep] = useState(0);
32 7   const [data, setData] = useState([]);
33 8   const [isLoading, setIsLoading] = useState(true);
34 9   function getPopularData() {
35 10     setIsLoading(true);
36 11     fetch('http://localhost:3000/get/app/popular?limit=${MAX}&offset=${step}')
37 12       .then((res) => res.json())
38 13       .then((data) => {
39 14         setData(data);
40 15         setIsLoading(false);
41 16       });
42 17   }
43 18   useEffect(() => {
44 19     getPopularData();
45 20   }, [step]);
46 21   return (
47 22     <div className="container">
48 23       <h2 className="title">Popular</h2>
49 24       {data[0] && (
50 25         <>
51 26           <table className="table">
52 27             <thead>
53 28               <tr>
54 29                 <th>No</th>
55 30                 {Object.keys(data[0]).map((val) => (
56 31                   <th>{val}</th>
57 32                   )));
58 33               </tr>
59 34             </thead>
60 35             <tbody>
61 36               {data.map((val, index) => (
62 37                 <tr key={index}>
63 38                   <td>{index + 1 + MAX * step}</td>
64 39                   {Object.values(val).map((val) => (
65 40                     <td>{val}</td>
66 41                     )));
67 42                   </tr>
68 43                 )));
69 44             </tbody>
70 45           </table>
71 46           <div className="action">
72 47             <button
73 48               onClick={() => setStep((prev) => prev - 1)}
74 49               disabled={step <= 0 || isLoading}
75 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. Kode yang digunakan dapat dilihat pada Gambar ??:

Kode 5.15: Membuat menu bar

```

19 import React from "react";
20 import { Link } from "react-router-dom";
21
22
23 export default function Navbar() {
24   return (
25     <ul className="menu">
26       <li className="item">
27         <a href="/">List</a>
28       </li>
29       <li className="item">
30         <a href="/app-url">App Url</a>
31       </li>
32       <li className="item">
33         <a href="/popular">Popular</a>
34       </li>
35     </ul>
36   );
37 }
38

```

• App.js

Dalam fungsi ini memanggil components yang dibuat untuk ditampilkan. Kode yang digunakan dapat dilihat pada Gambar 5.16:

Kode 5.16: Komponen Utama

```

42 import * as React from "react";
43 import Navbar from "./components/Navbar";
44 import { BrowserRouter, Routes, Route } from "react-router-dom";
45 import List from "./components/List";
46 import Popular from "./components/Popular";
47 import AppUrl from "./components/AppUrl";
48
49
50 function App() {
51   return (
52     <main className="app">
53       <Navbar />
54       <BrowserRouter>
55         <Routes>
56           <Route path="/" element={<List />} />
57           <Route path="/app-url" element={<AppUrl />} />
58           <Route path="/popular" element={<Popular />} />
59         </Routes>
60       </BrowserRouter>
61     </main>
62   );
63 }
64
65 export default App;
66

```

1 5.2.3 Node.js

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

4 1. Features

5 Bagian ini merupakan sebuah folder yang berisi media untuk berkomunikasi dengan BigQuery.
6 Dalam Features terdapat kelas GetApplication.js. Kelas GetApplication.js memiliki beberapa
7 function untuk mendapatkan data dari BigQuery. Kode yang digunakan dapat dilihat pada
8 Gambar 5.17:

Kode 5.17: Media untuk Komunikasi dengan BigQuery

```

9 1 const {BigQuery} = require('@google-cloud/bigquery');
10 2 const options = {
11 3     keyFilename: 'gsm-bigquery-credentials.json',
12 4     projectId: 'httparchive-bigquery-346414',
13 5 };
14 6 const bigquery = new BigQuery(options)
15 7
16 8
17 9 async function getApplications(app = "Apache") {
18 10     const getAppSql = 'select app, info, count(app) as jumlah, result from httparchive-bigquery-346414.app_result.
19 20         app_result where app = "${app}" and (result != "NON CONCLUSIVE" and result != "NOT VERSIONED")
21 21     group by app, info, result order by info ASC'
22 22     const options = {
23 23         query: getAppSql,
24 24         location: 'US',
25 25     };
26 26     const [job] = await bigquery.createQueryJob(options);
27 27     const [rows] = await job.getQueryResults();
28 28     return rows.filter(item => !item.info.includes("\\")),
29 29 }
30 30
31 31
32 32 async function getApplicationsType(limit = 5 , offset = 1) {
33 33     const getAppSql = 'select app from httparchive-bigquery-346414.Step.app_result where info != '' group by app limit ${
34 34         limit} offset ${offset}'
35 35     const options = {
36 36         query: getAppSql,
37 37         location: 'US',
38 38     };
39 39     const [job] = await bigquery.createQueryJob(options);
40 40     const [rows] = await job.getQueryResults();
41 41     return rows;
42 42 }
43 43
44 44 async function getApplicationsUrl(limit = 10, offset = 1) {
45 45     const getAppSql = 'select * from \'httparchive-bigquery-346414.URL_Result.url_result\' limit ${limit} offset ${offset}'
46 46
47 47     const options = {
48 48         query: getAppSql,
49 49         location: 'US',
50 50     };
51 51     const [job] = await bigquery.createQueryJob(options);
52 52     const [rows] = await job.getQueryResults();
53 53     return rows;
54 54 }
55 55
56 56 async function getPopularTech(limit = 10, offset = 1) {
57 57     const getAppSql = 'select * from \'httparchive-bigquery-346414.numsite_app_result_count.numsite_app_result_count\' '
58 58         limit ${limit} offset ${offset} '
59 59
60 60     const options = {
61 61         query: getAppSql,
62 62         location: 'US',
63 63     };
64 64     const [job] = await bigquery.createQueryJob(options);
65 65     const [rows] = await job.getQueryResults();
66 66     return rows;
67 67 }
68 68
69 69 module.exports = {getApplications, getApplicationsType, getApplicationsUrl, getPopularTech}

```

71 Berikut ini adalah penjelasan setiap fungsi:

- function getApplications(app = "Apache")
Pada fungsi ini mengembalikan app, info, jumlah app, result. Pada fungsi terdapat parameter untuk menentukan app yang ingin ditampilkan. Data tidak menampilkan result yang UNVERSIONED dan NON CONCLUSIVE.
- function getApplicationType(limit = 5, offset = 1)
Pada fungsi ini 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 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 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. Kode yang digunakan dapat dilihat pada Gambar 5.18:

Kode 5.18: Penghubung Backend Logic dengan Web-Client

```

16 1 const express = require('express')
17 2 const {GetPopularTech} = require("./Controllers/GetPopularTech");
18 3 const {GetAppRecap} = require("./Controllers/GetRecap");
19 4 const {GetAppType} = require("./Controllers/GetType");
20 5 const {GetAppByName} = require("./Controllers/GetApplications");
21 6 const cors = require('cors')
22 7 const app = express()
23 8 const port = 3000
24 9 app.use(cors())
25 10 app.get('/get/app/:name', GetAppByName())
26 11 app.get('/get/app/type', GetAppType())
27 12 app.get('/get/app/url', GetAppRecap())
28 13 app.get('/get/app/popular', GetPopularTech())
29 14 app.listen(port, () => {
30 15   console.log('Listening at http://localhost:${port}')
31 16 })

```

3. Controllers Bagian ini berfungsi untuk mengimplementasikan *feature* berdasarkan *use-case* yang di berikan. Kode yang digunakan dapat dilihat pada Gambar 5.19:

- GetApplications.js

Kode 5.19: Mendapatkan Aplikasi dan Hasilnya

```

37 1 const {getApplications} = require("../Features/GetApplications");
38 2 const GetAppByName = () => {
39 3   return (req, res) => {
40 4     const {name} = req.params
41 5     getApplications(name).then((rows) => {
42 6       res.send(rows);
43 7     }).catch((e) => {
44 8       res.send(e.message)
45 9     });
46 10   };
47 11 }
48 12 module.exports = {GetAppByName}

```

- GetPopularTech.js

Kode 5.20: Mendapatkan Aplikasi yang Populer

```

52 1 const {getPopularTech} = require("../Features/GetApplications");
53 2 const GetPopularTech = () => {
54 3   return (req, res) => {
55 4     const {limit, offset} = req.query
56 5     getPopularTech(limit, offset).then((rows) => {

```

```

1      6|         res.send(rows);
2      7|     }).catch((e) => {
3      8|         res.send(e.message)
4      9|     });
5     10| };
6     11| }
7     12| module.exports = {GetPopularTech}

```

9 • GetRecap.js

Kode 5.21: Mendapatkan URL dengan Jumlah Jumlah Result dari Aplikasi

```

10 10| const {getApplicationsUrl} = require("../Features/GetApplications");
11 11| const GetAppRecap = () => {
12 12|     return (req, res) => {
13 13|         const {limit, offset} = req.query
14 14|         getApplicationsUrl(limit, offset).then((rows) => {
15 15|             res.send(rows);
16 16|         }).catch((e) => {
17 17|             res.send(e.message)
18 18|         });
19 19|     };
20 20| }
21 21| module.exports = {GetAppRecap}
22 22|

```

24 • GetType.js

Kode 5.22: Mendapatkan Daftar Aplikasi

```

25 25| const {getApplicationsType} = require("../Features/GetApplications");
26 26| const GetAppType = () => {
27 27|     return (req, res) => {
28 28|         const {limit, offset} = req.query
29 29|         getApplicationsType(limit, offset).then((rows) => {
30 30|             res.send(rows);
31 31|         }).catch((e) => {
32 32|             res.send(e.message)
33 33|         });
34 34|     };
35 35| }
36 36| module.exports = {GetAppType}
37 37|

```

39 **5.3 Pengujian Fungsional**

40 Pada bagian ini, pengujian dilakukan untuk menguji perangkat lunak yang dibangun sudah berjalan
41 dengan sesuai. Terdapat beberapa halaman pada perangkat lunak ini, yaitu:

42 **5.3.1 Daftar Aplikasi**

43 Berikut ini beberapa fitur yang ada pada perangkat lunak:

44 1. Melihat *chart* dari setiap aplikasi.

45 Fitur ini bertujuan untuk melihat *chart* dari setiap aplikasi yang dipilih yang dapat dilihat
46 pada Gambar 5.6



Gambar 5.6: Menampilkan *Chart*

- 1 2. Melihat halaman selanjutnya dengan tombol next
- 2 Fitur ini bertujuan untuk melihat halaman selanjutnya pada tabel yang dapat dilihat pada
- 3 Gambar 5.7

List APP		
No	Name	
21	PHP	View
22	Typekit	View
23	Apache	View
24	Vue.js	View
25	Moment.js	View
26	Microsoft ASP.NET	View
27	Bluefish	View
28	XRegExp	View
29	AngularJS	View
30	Microsoft Word	View

[Prev](#)
[Next](#)

Gambar 5.7: Melakukan Tombol Next

1. Melihat halaman sebelumnya dengan tombol prev
2. Fitur ini bertujuan untuk melihat halaman sebelumnya pada tabel yang dapat dilihat pada Gambar 5.8
- 3.

List APP		
No	Name	
11	MooTools	View
12	PHP	View
13	Typekit	View
14	Apache	View
15	Vue.js	View
16	Moment.js	View
17	Microsoft ASP.NET	View
18	Bluefish	View
19	XRegExp	View
20	AngularJS	View

[Prev](#)
[Next](#)

Gambar 5.8: Melakukan Tombol Prev

5.3.2 Daftar URL dengan Jumlah Hasil Perbandingan

- 2 Berikut ini beberapa fitur yang ada pada perangkat lunak:
 1. Melihat halaman selanjutnya dengan tombol next
- 4 Fitur ini bertujuan untuk melihat halaman selanjutnya pada tabel yang dapat dilihat pada Gambar 5.9

No	url	supported	unsupported	not_versioned	non_conclusive
21	https://www.meokhealthspa.com.hk/	0	1	10	4
22	https://www.mepa.de/	1	1	9	0
23	https://www.mercatsantacaterina.com/	0	2	12	2
24	https://www.mercedes-benz-durham.ca/	1	3	14	3
25	https://www.mercedes-benz.hu/	0	1	8	2
26	https://www.merhalsa.jp/	0	3	8	0
27	https://www.meritspor.com.tr/	0	2	9	2
28	https://www.merlana.lt/	0	2	4	1
29	https://www.mersinhaber.com/	2	3	7	0
30	https://www.mertegra.eu/	1	0	1	0

[Prev](#) [Next](#)

Gambar 5.9: Melakukan Tombol Next

- 6 2. Melihat halaman sebelumnya dengan tombol prev
- 7 Fitur ini bertujuan untuk melihat halaman sebelumnya pada tabel yang dapat dilihat pada Gambar 5.10

App Url						
No	url	supported	unsupported	not_versioned	non_conclusive	
11	https://www.menu.hr/	0	6	8	2	
12	https://www.meokhealthspa.com.hk/	0	1	10	4	
13	https://www.mepa.de/	1	1	9	0	
14	https://www.mercatsantacaterina.com/	0	2	12	2	
15	https://www.mercedes-benz-durham.ca/	1	3	14	3	
16	https://www.mercedes-benz.hu/	0	1	8	2	
17	https://www.merhalsa.jp/	0	3	8	0	
18	https://www.meritspor.com.tr/	0	2	9	2	
19	https://www.merlana.lt/	0	2	4	1	
20	https://www.mersinhaber.com/	2	3	7	0	

Prev **Next**

Gambar 5.10: Melakukan Tombol Prev

1 5.3.3 Daftar Aplikasi yang Populer

2 Berikut ini beberapa fitur yang ada pada perangkat lunak:

- 3 1. Melihat halaman selanjutnya dengan tombol next

4 Fitur ini bertujuan untuk melihat halaman selanjutnya pada tabel yang dapat dilihat pada
5 Gambar 5.11

Popular						
No	numsites	app	supported	unsupported	not_versioned	non_conclusive
21	7494642	WordPress	350	4891016	2603276	0
22	7230612	Nginx	652	1789692	5440268	0
23	5977790	PHP	167095	2355525	3455170	0
24	5481111	Google Font API	0	0	5481111	0
25	4529823	Google Tag Manager	0	0	4529823	0
26	4067380	Apache	764705	353495	2949180	0
27	4047343	MySQL	0	0	4047343	0
28	3827039	Font Awesome	1066430	1719907	1040702	0
29	3597289	jQuery Migrate	128236	3200091	268962	0
30	3391648	Bootstrap	694234	1828777	868637	0

Prev **Next**

Gambar 5.11: Melakukan Tombol Next

- 1
2. Melihat halaman sebelumnya dengan tombol prev
- 2 Fitur ini bertujuan untuk melihat halaman sebelumnya pada tabel yang dapat dilihat pada
- 3 Gambar 5.12

Popular						
No	numsites	app	supported	unsupported	not_versioned	non_conclusive
11	8190668	Google Analytics	0	0	8190668	0
12	7494642	WordPress	350	4891016	2603276	0
13	7230612	Nginx	652	1789692	5440268	0
14	5977790	PHP	167095	2355525	3455170	0
15	5481111	Google Font API	0	0	5481111	0
16	4529823	Google Tag Manager	0	0	4529823	0
17	4067380	Apache	764705	353495	2949180	0
18	4047343	MySQL	0	0	4047343	0
19	3827039	Font Awesome	1066430	1719907	1040702	0
20	3597289	jQuery Migrate	128236	3200091	268962	0

Prev **Next**

Gambar 5.12: Melakukan Tombol Prev

BAB 6

KESIMPULAN DAN SARAN

- ³ Pada bab ini membahas kesimpulan berdasarkan implementasi, serta saran-saran untuk pengembangan berikutnya.

5 6.1 Kesimpulan

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

7 1. Mengambil nilai informasi dengan cara melakukan query untuk mengumpulkan daftar *website*,
8 mencari aplikasi yang digunakan, mencari aplikasi yang digunakan *website*, mengelompokkan
9 berdasar nama semua aplikasi yang dipakai, mencari data tentang versi aplikasi yang masih
10 didukung, dan melakukan perbandingan antara versi aplikasi yang masih dipakai sekarang
11 dengan aplikasi yang masih didukung.

12 2. Berdasarkan penelitian ini dikarena ruang lingkup lebih besar, sehingga terdapat perbedaan
13 seperti pada aplikasi Nginx, untuk ruang lingkup di Indonesia versi 1.10 paling banyak
14 digunakan url, sedangkan untuk ruang lingkup dunia versi 1.14 yang paling banyak digunakan.
15 Adapun beberapa aplikasi yang ditambahkan pada aplikasi ini yaitu Python dan JQuery
16 Migrate. Pada aplikasi Python jumlah versi yang paling banyak digunakan adalah versi 2.7
17 dan pada aplikasi JQuery Migrate versi aplikasi yang paling banyak digunakan adalah versi
18 1.4

19 3. Setelah diolah, terdapat 4.280 *website* yang menggunakan aplikasi yang semua aplikasinya
20 masih didukung. Hasil didapatkan dengan membuat *query* yang dapat dilihat pada Listing 4.4

21 6.2 Saran

- 22 Penulis memiliki beberapa saran untuk pengembangan aplikasi selanjutnya:

23 1. Dapat menggunakan metode klasifikasi untuk menentukan apakah suatu teknologi atau

24 aplikasi *supported* atau *unsupported*

DAFTAR REFERENSI

- [1] Beaulieu, A. (2005) *Learning SQL*, 1st ed edition. O'Reilly Media.
- [2] Nugroho, P. A. dan Steven, H. (2013) Measuring unsupported applications in indonesia popular websites. *JITEKI*, **66**, 595–614.
- [3] Google Developer Bigquery. <https://cloud.google.com/bigquery/docs/introduction>.
- [4] Tigani, J. dan Naidu, S. (2014) *Google BigQuery Analytics*, 1 edition. Wiley.
- [5] Souders, S. Http archive. <https://httparchive.org/faq>.
- [6] Dahl, R. nodejs. <https://nodejs.org/en/about/>.
- [7] Dahl, R. npmjs. <https://docs.npmjs.com/about-npm>.
- [8] Walke, J. Reactjs. <https://reactjs.org/docs/getting-started.html>.

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 Migrator	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/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/fancyapps/fancybox/releases
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/mediaelement/releases
MediaElement.js	420,046	33	420,013	http://www.mediaelementjs.com	?	https://github.com/mediaelement/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
Revslder	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/opensslstrat.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.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/
jQuery Mobile	105,150	402,192	99,392	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/releasenotes/index.html
Fingerprintjs	86,871	16,625	70,246	https://valve.github.io/fingerprintjs/	?	https://github.com/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://ken Wheeler.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.fivetran.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/szepviktor/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 <small>mod_ssl</small>	46,424	11,498	34,926	http://tomcat.apache.org	8.5	http://tomcat.apache.org/whichversion.html
<small>script.aculo.us</small>	33,928	22	33,906	http://modssl.org	?	http://www.modssl.org/
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/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/1ts
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/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.
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%202Mins%20Read%20In%20August%2C,new%20features.%20It%20requires%20Shopware%206.3.0%20or%20newer.
CodeMirror	16,677	404	16,273	https://codemirror.net/	?	https://github.com/felixhusse/codemirror-addon

DNN	20,994	4,928	16,066	http://dnnssoftware.com	8.5	https://www.dnnssoftware.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/javascript-client-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.fivetrans.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/
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%28Official,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/jsp.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	4	https://www.spip.net/en_article6499.html
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/	7.1	https://docs.trafficserver.apache.org/en/latest/release-notes/roadmap.en.html
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/benoitc/gunicorn/releases
Mobirise	4,272		4,272	https://mobirise.com/	?	https://mobirise.com/history.html

WebSite X5	4,261	4,261	https://www.websitem5.com/en/	?	https://www.websitem5.com/changelog/v2019.1/en.php?ed=&_ga=2.8172977.285220766.1636086202-530522610.1634454942
Discourse	3,997	14	3,983	https://discourse.org/	?
Next.js	55,570	51,606	3,964	https://nextjs.org	https://meta.discourse.org/c/releases/30
Socket.io	37,349	33,459	3,890	https://socket.io	https://github.com/vercel/next.js/releases
Apollo	3,751	2	3,749	https://www.apollographql.com	https://github.com/socketio/socket.io/releases
Nette Framework	18,202	14,464	3,738	https://nette.org/	https://github.com/ApolloAuto/apollo/releases
OpenLayers	6,527	2,912	3,615	https://openlayers.org/	https://files.nette.org/releases/
Jekyll	4,873	1,511	3,362	http://jekyllrb.com/	https://github.com/openlayers/openlayers/releases/
WP Rocket	214,557	211,230	3,327	https://wp-rocket.me/	https://jekyllrb.com/news/releases/
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/	?
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/releases

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://github.com/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/v1.0/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/javaeeserverfaces-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/w/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/releasenotes?version
ADPLAN	1,003	383	620	https://adplan.gafcodes.com/	?	
GlassFish	1,676	1,065	611	https://javeae.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/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	https://chevereto.com/	?	https://releases.chevereto.com/3.X/3.20/3.20.12.html#links
imperia CMS	450	64	https://www.piroweb-imperia.com/de/produkte/produktuebersicht/imperia-cms	?	http://imperiamucms.com/release-notes/
Contensis	330	330	https://www.contensis.com/	?	https://github.com/zengenti/contensis-react-base
Sarika-SPiP	329	1	http://www.sarika-spip.net/	4.0	https://www.spip.net/en_article6499.html
Phaser	329	2	https://phaser.io/	?	https://github.com/photonstorm/phaser/releases
GitBook	359	32	https://www.gitbook.com/	?	
govCMS	323	323	https://www.govcms.gov.au/	?	https://github.com/govCMS/govCMS8/releases
GeneXus	356	38	https://www.genexus.com/en/	?	https://www.genexus.com/en/developers/downloadcenter
Sphinx	3,010	2,701	https://www.sphinx-doc.org/en/master/	?	https://github.com/sphinxsearch/sphinx/releases
A-Frame	302	12	https://aframe.io/	?	https://github.com/aframevr/aframe/releases
Transifex	289	289	https://www.transifex.com/	?	https://github.com/transifex-client/releases
CKAN	301	13	https://ckan.org/	?	https://github.com/KSP-CKAN/CKAN/releases

TornadoServer	286	286	https://www.tornadoweb.org/en/stable/releases.html	?	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/	?
Oracle Commerce Cloud	255	2	253	https://cloud.oracle.com/commerce-cloud	?
Akka HTTP	226	226	https://akka.io/	?	https://doc.akka.io/docs/akka-http/current/plugins/oracle-commerce-cloud/release-notes
X-Cart	1,065	842	223	https://www.x-cart.com/	?
Typecho	227	7	220	http://typecho.org/	?
OpenUI5	219		219	https://openui5.org/	?
phpwind	344	142	202	https://www.phpwind.net/	?
Highlight.js	35,637	35,437	200	https://highlightjs.org/	?
CherryPy	194	2	192	https://www.cherrypy.org/	?
CMSSimple	254	62	192	https://www.cmsimple.org/en/	?
Kooboo CMS	247	56	191	https://www.kooboo.com/	?
mod_auth_pam	197	20	177	http://pam.sourceforge.net/mod_auth_pam/	?
Coppermine	174		174	https://coppermine-gallery.net/	?
				https://github.com/coppermine-gallery/cpg1.6.x/releases	

Serendipity	16'2	8	154	https://github.com/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.html1
FreeBSD	14,647	14,517	130	https://www.freebsd.org/releases/	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?legRedir=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 Infovis Toolkit	283	197	86	https://philogb.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/?subId1=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.totals.js.com/components/
Microsoft Excel	76	20	56	https://www.microsoft.com/id-id/microsoft-365/excel?legRedir=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/	?	https://www.hiawatha-webserver.org/about
Backdrop	716	667	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	https://milkeml.github.io/decimal.js/	?	https://www.npmjs.com/package/decimal.js/v/3.0.0
Adobe Robo- Help	63	15	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	http://www.methods.co.nz/asciidoc	?	https://github.com/asciidoctor/releases
Koken	43	43	https://www.koken.me/	?	https://www.koken.me/
Atlassian Bitbucket	46	3	https://www.atlassian.com/software/bitbucket	?	https://developer.atlassian.com/server/bitbucket/reference/api-changelog/
MODX	8,319	8,277	https://modx.com/	?	https://modx.com/blog/category/release/
PHP-Fusion	408	369	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/development/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-Org/Semantic-UI/releases
Coaster CMS	30		30	https://www.coastercms.org/	?	https://www.coastercms.org/blog/
cgit	30		30	https://git.zx2c4.com/cgit	?	https://git.zx2c4.com/cgit
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-fightingcryptojacking=anddoinggoodthings-with=contentsecurity=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/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/	?
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/releases
phpMyAdmin	11	4	https://www.phpmyadmin.net/	?	https://www.phpmyadmin.net/downloads/
SquirrelMail	96	89	https://squirrelmail.org/	?	https://www.squirrelmail.org/download.php
Livefyre	505	498	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	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	https://jfrog.com/open-source/#os-arti	?	https://www.jfrog.com/confluence/display/JFROG/Artifactory+Release+Notes
Gitea	28	23	https://gitea.io/en-us/	?	https://blog.gitea.io/
Planet	640	635	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	https://www.indyproject.org/	?	https://www.indyproject.org/documentation/
Oracle Commerce	2,641	2,637	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/oracle/opengrok/releases
gitweb	1		1	http://git-scm.com/	?	https://github.com/yannfleurydev/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?legRedir=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/w/index.php?title=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	4	https://reactjs.org/versions/release
Bootstrap	10	10		https://getbootstrap.com		
Underscore.js	1	1				
1.8.2						
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
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.numsite_app_result_count\' limit ${
48     limit} offset ${offset} '
49
50   const options = {
51     query: getAppSql,
52     location: 'US',
53   };
54   const [job] = await bigquery.createQueryJob(options);
55   const [rows] = await job.getQueryResults();
56   return rows;
57 }
58
59 module.exports = {getApplications, getApplicationsType, getApplicationsUrl, getPopularTech}
```

Kode B.2: 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|
13|
14| module.exports = {GetAppByName}
```

Kode B.3: 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
13
14 module.exports = {GetPopularTech}
```

Kode B.4: 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) => {
8             res.send(e.message)
9         });
10    };
11 }
12
13
14 module.exports = {GetAppRecap}
```

Kode B.5: GetType.js

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

Kode B.6: AppUrl.js

```
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                 {Object.keys(data[0]).map((val) => (
31                   <th>{val}</th>
32                 ))}
33             </thead>
34             <tbody>
35               {data[0].map((val) => (
36                 <tr>
37                   <td>{val}</td>
38                   <td>{val}</td>
39                 </tr>
40               ))}
41             </tbody>
42           </table>
43         </>
44       )
45     </div>
46   );
47 }
48
49 <AppUrl />
```

```

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    >
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 }

```

Kode B.7: List.js

```

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);
10 const [data, setData] = useState([]);
11 const [isLoading, setIsLoading] = useState(true);
12 const [selectedData, setSelectedDate] = useState();
13
14 function color(arr) {
15  let temp = [];
16  for (let i = 0; i < arr.length; i++) {
17    const dataType = arr[i].result;
18    switch (dataType) {
19      case "SUPPORTED":
20        temp.push("blue");
21        break;
22      case "UNSUPPORTED":
23        temp.push("red");
24        break;
25      default:
26        temp.push("green");
27    }
28  }
29  return temp;
30}
31 useEffect(() => {
32  Chart.register(CategoryScale);
33}, []);
34
35 function getListData(page) {
36  setIsLoading(true);
37  fetch('http://localhost:3000/get/app/type?limit=${MAX}&offset=${page}')
38  .then((res) => res.json())
39  .then((data) => {
40    setData(data);
41    setIsLoading(false);
42  });
43}
44
45 function getData(name) {
46  fetch('http://localhost:3000/get/app/name/${name}')
47  .then((res) => res.json())
48  .then((data) => {
49    setSelectedDate({
50      name,
51      data: {
52        labels: data.map((val) => String(val.info)),
53        datasets: [
54          {
55            label: name,
56            data: data.map((val) => String(val.jumlah)),
57            backgroundColor: color(data),
58          },
59        ],
60      },
61    });
62 }

```

```

62    });
63
64  useEffect(() => {
65    getListData(step);
66  }, [step]);
67  return (
68    <div className="container">
69      <h2 className="title">List APP</h2>
70      {data[0] && (
71        <>
72          <table className="table table-hover">
73            <thead>
74              <tr>
75                <th className="no">No</th>
76                <th>Name</th>
77                <th className="action-head"></th>
78              </tr>
79            </thead>
80            <tbody>
81              {data.map((val, index) => (
82                <tr key={index} onClick={() => getData(val.app)}>
83                  <th cla>{index + 1 + MAX * step}</th>
84                  <td>{val.app}</td>
85                  <td>View</td>
86                </tr>
87              ))}
88            </tbody>
89          </table>
90          <div className="action">
91            <button
92              onClick={() => setStep((prev) => prev - 1)}
93              disabled={step <= 0 || isLoading}
94            >
95              Prev
96            </button>
97            <button
98              onClick={() => setStep((prev) => prev + 1)}
99              disabled={isLoading}
100            >
101              Next
102            </button>
103          </div>
104        </>
105      )})
106    );
107
108  {selectedData && (
109    <div className="selected">
110      <h3 className="title">{selectedData.name}</h3>
111      <Bar data={selectedData.data} />
112    </div>
113  )}
114  </div>
115 );
116 }

```

Kode B.8: Navbar.js

```

1 import React from "react";
2 import { Link } from "react-router-dom";
3
4 export default function Navbar() {
5   return (
6     <ul className="menu">
7       <li className="item">
8         <a href="/">List</a>
9       </li>
10      <li className="item">
11        <a href="/app-url">App Url</a>
12      </li>
13      <li className="item">
14        <a href="/popular">Popular</a>
15      </li>
16    </ul>
17  );
18 }

```

Kode B.9: Popular.js

```

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
19   useEffect(() => {
20     getPopularData();
21   }, [step]);
22
23   return (
24     <div>
25       <h2>Popular APP</h2>
26       <table>
27         <thead>
28           <tr>
29             <th>No</th>
30             <th>Name</th>
31             <th>Action</th>
32           </tr>
33         </thead>
34         <tbody>
35           {data.map((val, index) => (
36             <tr key={index}>
37               <td>{index + 1}</td>
38               <td>{val.name}</td>
39               <td>
40                 <button onClick={() => handleEdit(index)}>Edit</button>
41                 <button onClick={() => handleDelete(index)}>Delete</button>
42               </td>
43             </tr>
44           ))}
45         </tbody>
46       </table>
47     </div>
48   );
49 }

```

```

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            >
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 }

```

Kode B.10: App.js

```

1 import * as React from "react";
2 import Navbar from "./components/Navbar";
3 import { BrowserRouter, Routes, Route } from "react-router-dom";
4 import List from "./components/List";
5 import Popular from "./components/Popular";
6 import AppUrl from "./components/AppUrl";
7
8 function App() {
9  return (
10    <main className="app">
11      <Navbar />
12      <BrowserRouter>
13        <Routes>
14          <Route path="/" element={<List />} />
15          <Route path="/app-url" element={<AppUrl />} />
16          <Route path="/popular" element={<Popular />} />
17        </Routes>
18      </BrowserRouter>
19    </main>
20  );
21}
22
23 export default App;

```