

**SKRIPSI**

**PERKEMBANGAN TEKNOLOGI PEMBANGUN *WEB* DUNIA**



**Alfonsus Oktario Sutomo**

**NPM: 6181801010**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2025**



# DAFTAR ISI

DAFTAR ISI	iii
DAFTAR GAMBAR	v
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 <i>HTTP Archive</i> [1]	5
2.1.1 Laporan	5
2.1.2 <i>2024 Web Almanac</i>	26
2.1.3 <i>Public Dataset</i>	27
2.2 <i>Structure Query Language</i>	29
2.3 Visualisasi Data	31
2.3.1 Line Plot	31
2.4 <i>Tk Interface</i> [2]	32
<b>3 ANALISIS PENYELESAIAN MASALAH</b>	<b>35</b>
3.1 Analisis Masalah	35
3.2 Data Kecil	35
3.2.1 Penyiapan Data Kecil	35
3.2.2 Pengolahan Data Kecil	36
3.3 Eksplorasi <i>Library Python</i>	41
3.3.1 <i>Tkinter</i>	41
<b>4 PENAMBANGAN DATA</b>	<b>45</b>
4.1 Deskripsi data	45
4.2 Penyiapan data	46
4.3 Eksplorasi Data	47
4.3.1 Perkembangan sepuluh teknologi yang paling banyak digunakan di semua perangkat.	47
4.3.2 Perkembangan sepuluh teknologi yang paling banyak digunakan di perangkat <i>desktop</i> dan <i>mobile</i> .	49
4.3.3 Perbandingan teknologi yang nampak populer	52
DAFTAR REFERENSI	55
A KODE PROGRAM	57



## DAFTAR GAMBAR

1.1	Contoh <i>line chart</i> . . . . .	2
2.1	Ukuran sample <i>web</i> yang digunakan untuk analisis . . . . .	6
2.2	Total <i>kilobyte</i> yang <i>direquest</i> oleh halaman <i>web</i> . . . . .	6
2.3	Total <i>Request</i> yang dilakukan oleh halaman <i>web</i> . . . . .	7
2.4	Persentase <i>web</i> yang memiliki properti <b>font-display</b> . . . . .	7
2.5	Jumlah <i>kilobytes</i> dari <i>resource JavaScript</i> . . . . .	8
2.6	Jumlah <i>Rquest</i> dari <i>resource JavaScript</i> . . . . .	9
2.7	Perkembangan waktu yang dibutuhkan oleh setiap halaman <i>web</i> . . . . .	9
2.8	Perkembangan perpindahan jumlah <i>kilobytes</i> yang dibutuhkan oleh gambar . . . . .	10
2.9	Perkembangan jumlah <i>request</i> untuk gambar . . . . .	11
2.10	Perkembangan jumlah <i>kilobyte</i> yang dapat dihemat dalam penggunaan gambar . . . . .	12
2.11	Perkembangan jumlah <i>kilobyte</i> yang dapat dihemat dengan melakukan kompresi pada gambar . . . . .	13
2.12	Perkembangan waktu yang dibutuhkan untuk menampilkan konten utama . . . . .	14
2.13	Perkembangan waktu untuk sebuah <i>web</i> merespon sebuah navigasi . . . . .	14
2.14	Perkembangan waktu yang dibutuhkan untuk konten dapat terlihat secara jelas . . . . .	15
2.15	Perkembangan skor PWA . . . . .	16
2.16	Perkembangan persentase halaman yang membuat <i>service worker</i> bekerja . . . . .	16
2.17	Perkembangan persentase skor aksesibilitas . . . . .	17
2.18	Perkembangan persentase <i>web</i> yang memiliki nama tombol yang aksesibel . . . . .	18
2.19	Perkembangan persentase halaman yang menggunakan label yang aksesible . . . . .	18
2.20	Perkembangan persentase halaman yang memiliki <i>link canonical</i> yang valid . . . . .	19
2.21	Perkembangan persentase halaman yang memiliki <i>link</i> yang deskriptif . . . . .	20
2.22	PERkembangan jumlah <i>kilobytes</i> oleh dokumen HTML . . . . .	21
2.23	Perkembangan oersentase halaman <i>web</i> yang memiliki penilaian <i>core web vitalis</i> yang bagus . . . . .	22
2.24	Contoh perkembangan persentase halaman yang memiliki pengalaman <i>First Paint</i> yang baik . . . . .	23
2.25	Contoh perkembangan persentase halaman <i>web</i> yang memiliki pebgalaman DCL yang baik . . . . .	23
2.26	Perkembangan persentase halaman <i>web</i> yang membaca <i>clipboard</i> milik sistem. . . . .	24
2.27	Perkembangan penggunaan API <i>Notification Trigger</i> . . . . .	25
2.28	Perkembangan persentase halaman <i>web</i> penggunaan API <i>Idle Detection</i> . . . . .	25
2.29	Perkembangan persentase halaman <i>web</i> yang menggunakan teknologi <i>WordPress</i> dengan skor CWV yang baik . . . . .	26
2.30	Perkembangan persentase halaman <i>web</i> yang memiliki skor CWV yang bagus dengan berbagai teknologi . . . . .	26
2.31	Contoh visualisasi dari perkembangan IPK dan IPS dengan menggunakan <i>line plot</i> . . . . .	32
3.1	Perkembangan jumlah penggunaan sepuluh teknologi populer . . . . .	38
3.2	Perkembangan sepuluh teknologi populer berdasarkan persentase penggunaannya . . . . .	38
3.4	Perkembangan persentase penggunaan sepuluh teknologi populer pada <i>client mobile</i> . . . . .	40

3.3	Perkembangan sepuluh teknologi populer pada <i>client mobile</i> . . . . .	40
3.5	Perkembangan sepuluh teknologi yang populer untuk <i>client desktop</i> . . . . .	41
3.6	Perkembangan persentase penggunaan sepuluh teknologi populer pada <i>client desktop</i>	41
3.7	Checkbox yang berisi teknologi yang dapat dipilih untuk dilihat perkembangannya	42
3.8	<i>Textbox</i> untuk memilih tanggal mulai rentang waktu untuk melihat perkembangan teknologi . . . . .	43
3.9	<i>Radio button</i> untuk memilih mode grafik yang ingin ditampilkan . . . . .	43
3.10	Tampilan <i>GUI</i> untuk melihat perkembangan teknologi pembuatan <i>web</i> . . . . .	43
4.1	Perkembangan sepuluh teknologi yang paling banyak digunakan membangun <i>web</i> .	48
4.2	Perkembangan persentase penggunaan dari sepuluh teknologi yang paling banyak digunakan . . . . .	49
4.3	Perkembangan sepuluh teknologi paling banyak diggunakan pada perangkat <i>mobile</i>	50
4.4	Perkembangan persentase penggunaan teknologi pada perangkat <i>mobile</i> . . . . .	50
4.5	Perkembangan sepuluh teknologi paling banyak diggunakan pada perangkat <i>desktop</i>	51
4.6	Perkembangan persentase penggunaan teknologi pada perangkat <i>desktop</i> . . . . .	52
4.7	Perbandingan jumlah penggunaan teknologi <i>PHP</i> dan <i>Node.js</i> . . . . .	53
4.8	Perbandingan persentase penggunaan teknologi <i>PHP</i> dan <i>Node.js</i> . . . . .	53
4.9	Perbandingan jumlah penggunaan teknologi <i>jQuery</i> dan <i>Angular</i> . . . . .	54
4.10	Perbandingan persentase penggunaan teknologi <i>jQuery</i> dan <i>Angular</i> . . . . .	54

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi pembuatan *web* telah mengalami perubahan yang sangat pesat dalam lima tahun terakhir. Sejak kemunculannya, *web* telah menjadi platform utama dalam penyebaran informasi, komunikasi, hingga transaksi digital. Seiring meningkatnya kebutuhan pengguna terhadap kecepatan, keamanan, dan interaktivitas, berbagai teknologi baru terus bermunculan untuk mendukung pengembangan *web* yang lebih efisien dan responsif.

Internet sendiri merupakan jaringan yang menghubungkan berbagai perangkat untuk memungkinkan pertukaran informasi secara cepat. Pertukaran informasi ini diatur oleh protokol utama TCP/IP (*Transmission Control Protocol/Internet Protocol*). Namun, informasi yang dikirimkan di internet harus mudah dipahami oleh pengguna, tidak hanya dalam bentuk teks tetapi juga melalui gambar, video, dan suara. Kebutuhan inilah yang mendorong berkembangnya layanan *web* (*World Wide web*), yang memungkinkan penyajian informasi secara lebih interaktif dengan memanfaatkan protokol HTTP (*HyperText Transfer Protocol*).

Teknologi pembuatan *web* semakin beragam dalam perkembangannya, baik dari sisi *front-end* maupun *back-end*. Beberapa teknologi utama yang mendukung pengembangan *web* di antaranya adalah JavaScript, PHP, dan MySQL. Munculnya berbagai *framework* dan pustaka seperti React, Vue.js, dan Node.js juga mempercepat adopsi teknologi baru dalam pengembangan *web* modern. Perubahan ini membuat pentingnya pemantauan tren teknologi *web* agar pengembang dapat memilih teknologi yang sesuai dengan kebutuhan pengguna dan standar industri.

Penelitian ini bertujuan untuk mengetahui tren perkembangan penggunaan teknologi pembuatan *web*. Penelitian ini merupakan pengembangan dari penelitian yang telah dilakukan sebelumnya [3] yang menjawab pertanyaan berapa banyak *web* yang menggunakan teknologi yang jarang dipakai oleh kebanyakan *developer*.

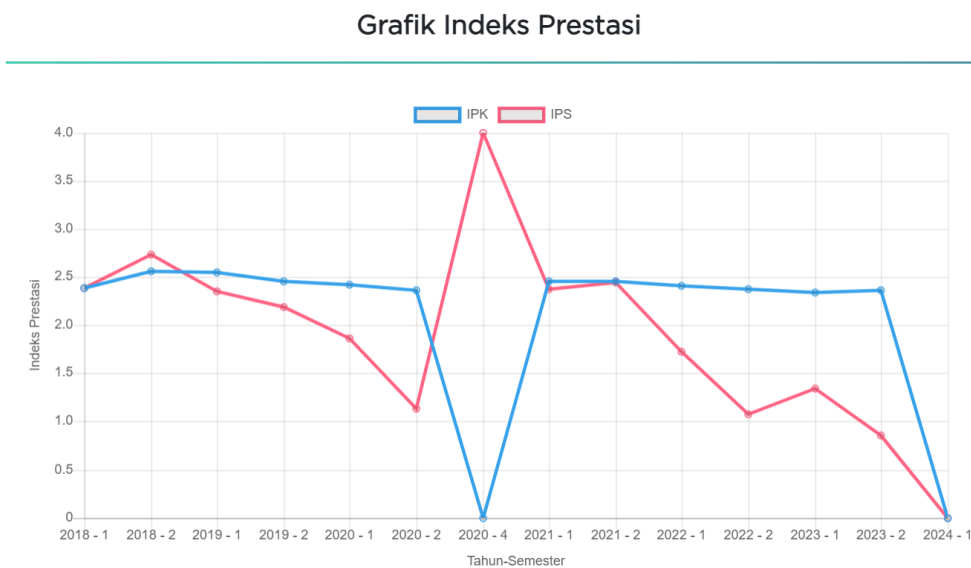
Situs *HTTP Archive* menyediakan data tentang teknologi yang digunakan dalam pembuatan *web* untuk mencatat perkembangan teknologi ini. Situs ini mengumpulkan data berdasarkan berbagai aspek, seperti pengalaman pengguna dalam mengakses *web*, kecepatan pemuatan halaman, serta tingkat aksesibilitas. Salah satu aspek utama yang diamati dalam penelitian ini adalah *Chrome User Experience Report* (CrUX), yang mengukur tingkat interaktivitas dan kecepatan pemuatan *web* berdasarkan data nyata dari pengguna peramban Google Chrome.

Data dari *HTTP Archive* kemudian disimpan dalam *Google BigQuery*, layanan penyimpanan dan analisis data berbasis *cloud* yang memungkinkan pemrosesan data dalam skala besar menggunakan *query SQL*. Dengan adanya teknologi ini, analisis terhadap perkembangan teknologi pembuatan *web*

1 dapat dilakukan secara lebih mendalam dan berbasis data yang akurat.

2 Untuk mempermudah pemahaman terhadap hasil analisis, penelitian ini akan menggunakan  
3 visualisasi data dalam bentuk grafik. Salah satu bentuk visualisasi yang digunakan adalah *line*  
4 *chart*, yang dapat menunjukkan tren perubahan teknologi dalam rentang waktu tertentu secara  
5 lebih jelas. Contoh *line chart* dapat dilihat pada gambar 1.1. Terlihat adanya kenaikan IPK pada  
6 2020-4 namun di waktu yang sama IPS juga mengalami penurunan yang sangat signifikan.

7 Perkembangan penggunaan berbagai teknologi *web* dapat divisualisasikan sehingga pola-pola  
8 perubahan dapat dikenali dengan lebih mudah dengan menggunakan *line chart*. Selain itu, bentuk  
9 visualisasi lainnya seperti *bar chart* dan *scatter plot* juga digunakan untuk memberikan perspektif  
10 tambahan terhadap data yang dianalisis.



Gambar 1.1: Contoh *line chart*

11 Penelitian ini bertujuan untuk memahami bagaimana tren teknologi pembuatan *web* berkembang  
12 dalam lima tahun terakhir, dari Oktober 2018 hingga Desember 2024. Dengan menggunakan data  
13 dari *HTTP Archive* dan *Google BigQuery*, penelitian ini akan mengeksplorasi perubahan signifikan  
14 dalam penggunaan teknologi *web* dan dampaknya terhadap pengalaman pengguna. Hasil analisis  
15 ini diharapkan dapat memberikan wawasan bagi pengembang *web* dan industri teknologi dalam  
16 memahami arah perkembangan *web* di masa depan.

## 17 1.2 Rumusan Masalah

18 Rumusan masalah yang akan diselesaikan dalam penelitian ini adalah:

- 19 1. Bagaimana perkembangan teknologi pembuatan *web* selama lima tahun terakhir?
- 20 2. Bagaimana perkembangan teknologi pembuatan *web* yang banyak digunakan oleh pembuat  
21 *web*?
- 22 3. Bagaimana cara menyajikan perkembangan teknologi pembuatan *web* kepada pengguna?



## 1.3 Tujuan

Tujuan yang akan dicapai dari penelitian ini adalah:

1. Mengetahui perkembangan teknologi pembuatan *web* selama lima tahun terakhir.
2. Mengetahui perkembangan teknologi pembuatan *web* yang banyak digunakan oleh pembuat *web*.
3. Membuat perangkat lunak untuk menyajikan perkembangan teknologi pembuatan *web*.

## 1.4 Batasan Masalah

Batasan masalah yang diterapkan pada penelitian ini adalah sebagai berikut:

1. Data yang digunakan berasal dari rentang lima tahun terakhir. Hal ini dimaksudkan untuk membatasi ukuran data agar tidak besar.
2. Data yang akan dianalisis adalah data jumlah penggunaan dan persentase penggunaan. Hal ini dilakukan agar cakupan analisis tidak terlalu besar.

## 1.5 Metodologi

Metodologi yang digunakan dalam melakukan penelitian ini adalah sebagai berikut:

- Mengumpulkan data penggunaan teknologi pembuatan *web* selama lima tahun terakhir.
- Membersihkan data dari kolom dan baris yang tidak digunakan.
- Melakukan analisis dengan menggunakan data dengan skala lebih kecil.
- Melakukan analisis dengan menggunakan data yang sebenarnya.
- Membuat perangkat lunak untuk menampilkan hasil analisis secara interaktif.

## 1.6 Sistematika Pembahasan

Sistematika pembahasan tugas akhir ini adalah:

1. Bab 1: Pendahuluan  
Membahas latar belakang, tujuan, rumusan masalah, batasan masalah, dan metodologi penelitian yang digunakan.
2. Bab 2: Landasan Teori  
Membahas *web*, *HTTP Archive*, bahasa SQL, *Google Big Query*, dan visualisasi data yang digunakan.
3. Bab 3: Analisis Masalah  
Membahas tentang analisis masalah dan solusinya dan melakukan analisis dengan menggunakan data yang skalanya lebih kecil.
4. Bab 4: Penambangan Data  
Membahas eksplorasi dan analisis data dengan menggunakan data *real*.
5. Bab 5: Pembuatan perangkat lunak dan Peluncuran Model  
Membahas tentang pembuatan perangkat lunak dan pengujian fungsional perangkat lunak untuk menampilkan hasil analisis secara interaktif.

- 1 6. Bab 6 : Kesimpulan dan Saran
- 2 Membahas tentang kesimpulan dari hasil penelitian yang telah dilakukan dan saran agar
- 3 peelitian ini lebih baik.

## BAB 2

### LANDASAN TEORI

Bab ini akan berisikan tentang beberapa teori dari metode atau hal-hal yang diperlukan dalam melakukan penelitian ini seperti apa itu *HTTP Archive*, SQL, statistika, dan visualisasi data.

#### 2.1 *HTTP Archive* [1]

*HTTP Archive* merupakan sebuah situs yang menelusuri bagaimana sebuah *web* dibuat. Situs ini menyediakan data historis yang menggambarkan bagaimana halaman-halaman *web* berevolusi. Orang-orang yang dapat menggunakan data dari *HTTP Archive* adalah bagian dari komunitas *web*, pelajar, dan pemimpin dalam industri.

Komunitas *web* menggunakan data yang terdapat di *HTTP Archive* untuk mempelajari secara lebih lanjut mengenai keadaan *web* yang terlihat pada unggahan *blog*, presentasi, dan media sosial. Pelajar menggunakannya untuk mendukung penelitian di tingkat publikasi yang besar seperti ACM dan IEEE. Sedangkan para pemimpin dalam industri menggunakan data ini untuk menyesuaikan alat yang mereka punya agar secara akurat dapat menunjukkan bagaimana halaman *web* dibuat. Contohnya, sebuah alat akan mengingatkan pengembang ketika bundel *JavaScript* yang digunakan terlalu besar seperti yang ditunjukkan oleh beberapa persentase dari semua *web*. Dalam situs *HTTP Archive* ini terdapat beberapa bagian seperti Laporan, *Web Almanac*, dan *Public Dataset*

##### 2.1.1 Laporan

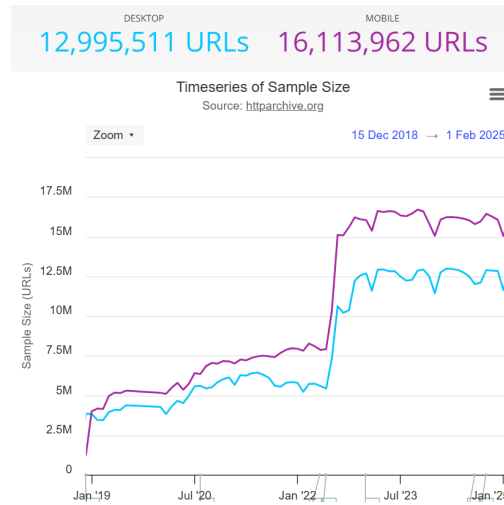
Laporan berisi informasi terperinci mengenai sumber daya yang diambil, API dan fitur platform yang digunakan, serta jejak eksekusi dari setiap halaman dari situs-situs teratas yang ada di *web*. Informasi yang telah didapatkan kemudian diolah dan dianalisis untuk melihat perkembangan tren. Laporan yang dimiliki oleh situs *HTTP Archive* dibagi menjadi beberapa kategori. Kategori tersebut adalah sebagai berikut:

##### *State of the Web*

*State of the Web* berisi Laporan yang menangkap perkembangan *web* secara jangka panjang termasuk teknik untuk efisiensi jaringan dan penggunaan standar seperti HTTPS. Laporan ini mencakup beberapa hal yaitu:

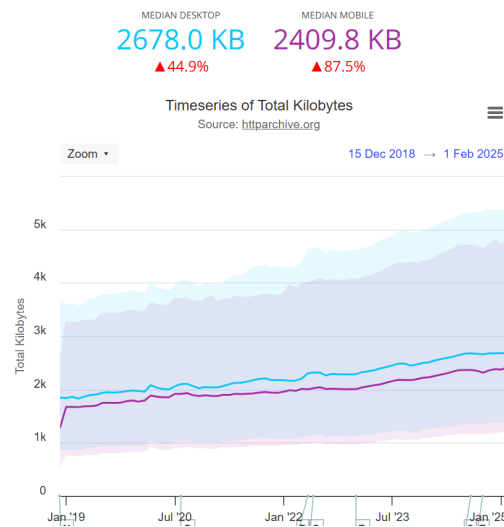
- *Sample size* yang berisi perkembangan jumlah URLs yang digunakan untuk dianalisis. Contoh visualisasi data yang dimiliki oleh laporan ini dapat dilihat pada Gambar 2.1. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025, bahwa adanya kenaikan

ukuran *sample* pada bulan juli hingga agustus 2022. Data ini juga menunjukkan pengambilan sample dari dua *client* yang berbeda yaitu *desktop* dan *mobile*.



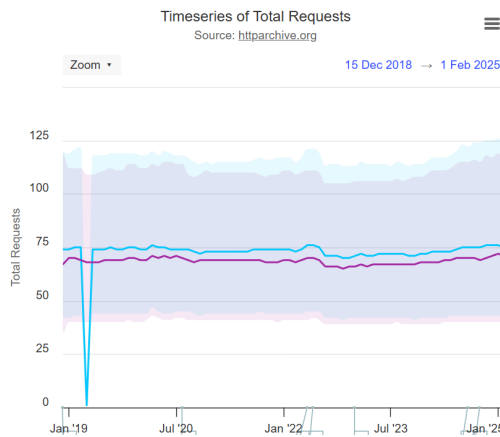
Gambar 2.1: Ukuran sample *web* yang digunakan untuk analisis

- *Total Kilobytes* yang berisi jumlah dari ukuran perpindahan *kilobytes* dari semua sumber daya yang di *request* oleh halaman *web*. Contoh visualisasi data dari laporan ini dapat dilihat pada Gambar 2.2. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa jumlah *kilobyte* yang *direquest* tidak mengalami banyak perubahan atau stabil.

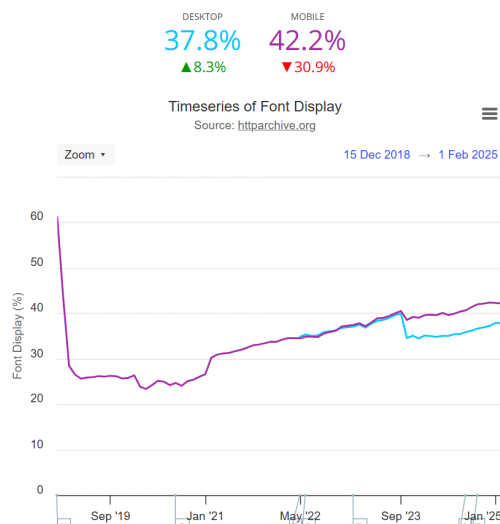


Gambar 2.2: Total *kilobyte* yang *direquest* oleh halaman *web*

- *Total Request* yang berisi jumlah *resource* yang di *request* oleh halaman *web*. Contoh hasil visualisasi data dari laporan ini dapat dilihat pada Gambar 2.3. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa pada perangkat *desktop* mengalami penurunan yang signifikan pada tanggal 1 april 2019 dengan hanya memiliki satu *request*.

Gambar 2.3: Total *Request* yang dilakukan oleh halaman *web*

- *Font Display* yang berisi persentase dari halaman yang menghindari munculnya teks tidak terlihat dengan sekejap sewaktu *web* memuat font dengan menggunakan properti CSS **font-display**. Matriks ini diukur dengan menggunakan *Lighthouse*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.4. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa data untuk perangkat *desktop* baru tersedia mulai pada tanggal 1 Mei 2022 karena *Lighthouse* baru melakukan migrasi ke perangkat *desktop*. Kemudian persentase *web* yang menggunakan properti **font-display** pada perangkat *mobile* mengalami penurunan yang signifikan mulai dari tanggal 15 Desember 2018 sampai 1 April 2019 kemudian mengalami kenaikan kembali pada 1 Februari 2021.

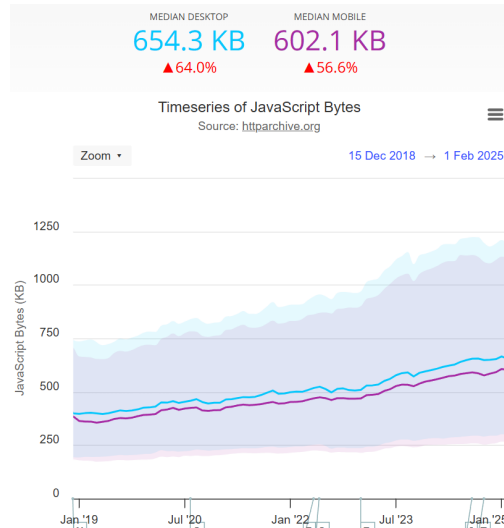
Gambar 2.4: Persentase web yang memiliki properti **font-display**

## State of JavaScript

*JavaScript* membuat halaman *web* dapat memiliki aplikasi yang kaya dan lebih interaktif. Laporan dalam kategori ini bertujuan untuk melihat penggunaan *JavaScript* dalam *web* dan adopsi serta trennya untuk perangkat *mobile*. Report ini akan menganalisis skrip eksternal. Skrip eksternal ini dimaksudkan untuk *resource file* yang menggunakan ekstensi **js** atau **json** atau sebuah tipe MIME((*Multipurpose Internet Mail Extensions*)) yang mengandung **script** atau **json**. Beberapa

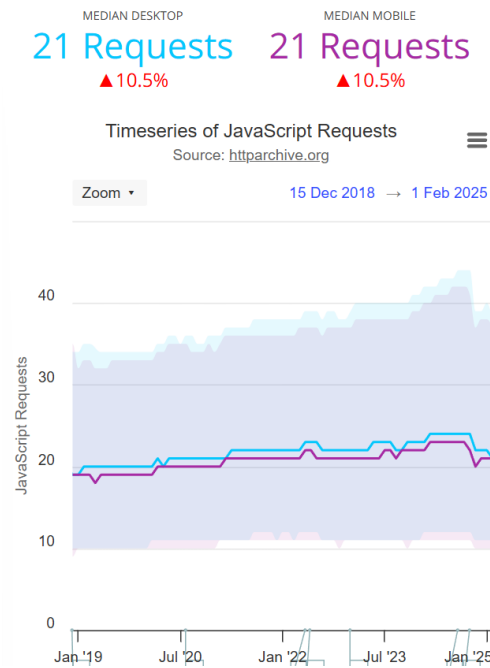
1 hal yang dianalisis adalah sebagai berikut:

- 2 • *JavaScript Bytes* yang berisi jumlah ukuran perpindahan *kilobytes* dari skrip eksternal yang  
3 di *request*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.5. Terlihat dari  
4 data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan  
5 *mobile* bahwa adanya peningkatan *kilobytes* setiap tahun nya. Hal ini menandakan bahwa  
6 semakin banyak *resource* yang berbentuk JavaScript yang digunakan oleh halaman *web*.

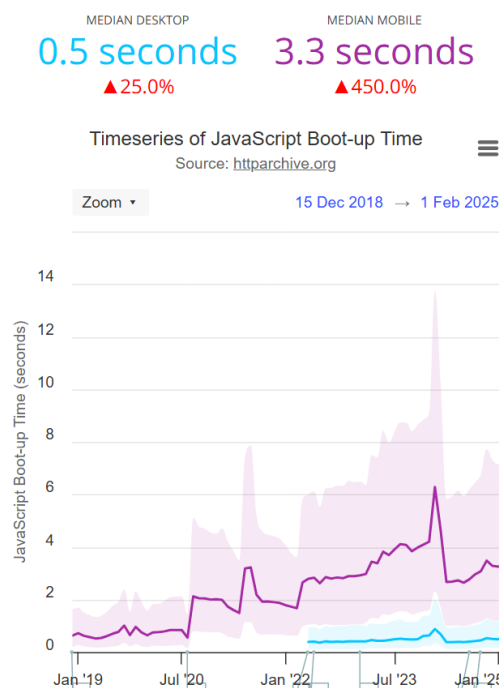


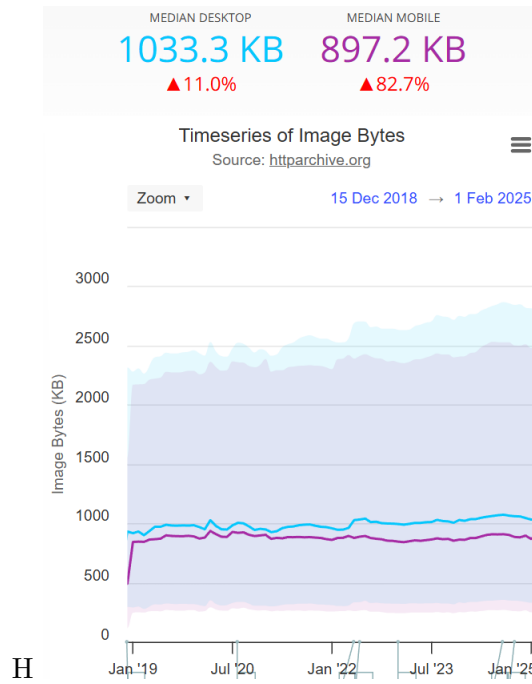
Gambar 2.5: Jumlah *kilobytes* dari *resource JavaScript*

- 7 • *JavaScript Requests* yang berisi jumlah skrip eksternal yang di *request* oleh halaman *web*.  
8 Contoh visualisasi dari data ini dapat dilihat pada Gambar 2.6. Terlihat dari data yang  
9 diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile*  
10 bahwa jumlah *request* nya cenderung stabil karena tidak adanya kenaikan maupun penurunan  
11 jumlah *request* yang signifikan.

Gambar 2.6: Jumlah Rquest dari *resource* JavaScript

- *JavaScript Boot-Up Time* yang berisi jumlah dari waktu CPU yang dibutuhkan oleh setiap *script* di setiap halaman *web*. Matriks ini diukur menggunakan *Lighthouse*. Contoh visualisasi dari data ini dapat dilihat pada Gambar 2.7. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan jumlah waktu yang dibutuhkan oleh perangkat *mobile*. Data untuk perangkat *desktop* tidak tercatat secara lengkap karena *Lighthouse* baru melakukan integrasi pada 1 Mei 2022.

Gambar 2.7: Perkembangan waktu yang dibutuhkan oleh setiap halaman *web*



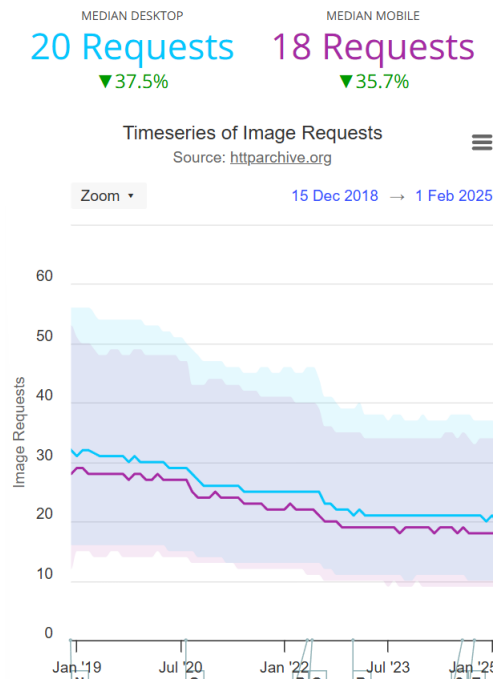
Gambar 2.8: Perkembangan perpindahan jumlah *kilobytes* yang dibutuhkan oleh gambar

### 1 *State of Images*

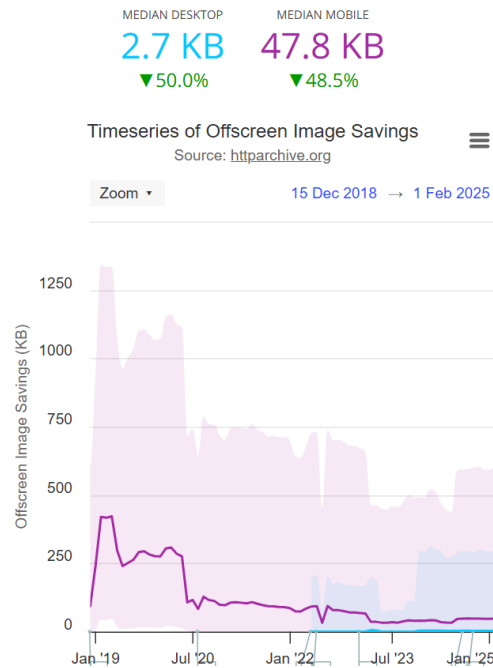
*Images* atau gambar merupakan tipe *resource* yang populer digunakan dalam *web*. Laporan ini adalah hasil analisa penggunaan gambar eksternal di seluruh *web*. Gambar eksternal adalah *resource* yang memiliki ekstensi *png*, *gif*, *jpg*, *jpeg*, *webp*, *ico*, atau *svg* atau sebuah tipe MIME(*Multipurpose Internet Mail Extensions*) yang mengandung *image*. Laporan yang masuk kategori ini adalah:

- *Image Bytes* yang berisi jumlah ukuran perpindahan *kilobytes* dari gambar eksternal yang di *request* oleh halaman *web*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.8. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa jumlah *kilobytes* yang diperlukan oleh gambar cenderung stabil karena tidak adanya kenaikan atau penurunan yang signifikan.
- *Image Request* yang berisi jumlah gambar eksternal yang di *request* oleh halaman *web*. Hasil visualisasi dari data ini dapat dilihat pada Gambar 2.9. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa jumlah *request* terus mengalami penurunan. Hal ini menandakan bahwa halaman *web* mulai mengurangi penggunaan gambar sebagai media untuk menyampaikan informasinya.



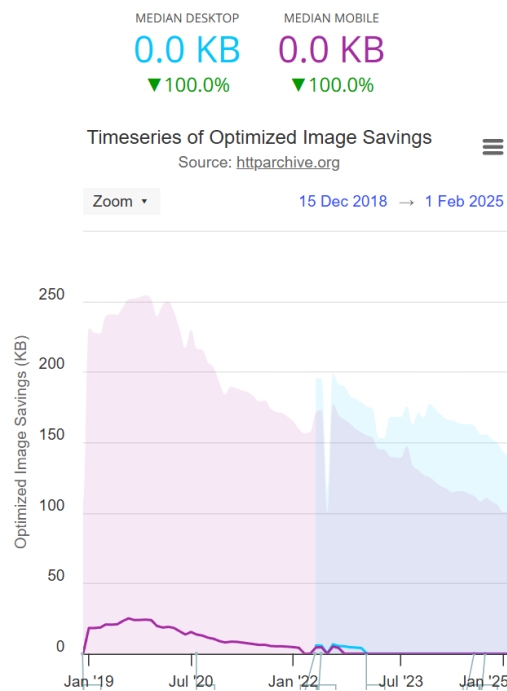
Gambar 2.9: Perkembangan jumlah *request* untuk gambar

- *Offscreen Image Save* yang berisi jumlah *kilobytes* yang dapat dihemat oleh setiap halaman menggunakan *lazy-loading offscreen* dan gambar tersembunyi. Matriks ini berasal dari *Lighthouse*. Hasil visualisasi dari data ini dapat dilihat pada Gambar 2.10. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa pada perangkat *mobile* jumlah *kilobyte* yang dapat dihemat semakin menurun. Sedangkan pada perangkat *desktop* jumlah *kilobyte* yang dapat dihemat cenderung stabil. Data untuk perangkat *desktop* dimulai dari 1 Mei 2022 karena *Lighthouse* baru melakukan integrasi dengan perangkat *desktop*.



Gambar 2.10: Perkembangan jumlah *kilobyte* yang dapat dihemat dalam penggunaan gambar

- *Optimize Image Savings* yang berisi jumlah *kilobytes* yang dapat dihemat oleh setiap halaman dengan mengatur kompresi JPEG ke 85 atau lebih kecil. Contoh visualisasi dari data ini dapat dilihat pada Gambar 2.11. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya penurunan jumlah *kilobyte*. Pada perangkat *desktop* juga mengalami hal yang sama. Hal ini dapat mengindikasikan bahwa pembuat halaman *web* tidak memerlukan lagi kompresi pada gambar yang digunakan. Data untuk perangkat *desktop* dimulai dari 1 Mei 2022 karena *Lighthouse* baru melakukan integrasi dengan perangkat *desktop*.

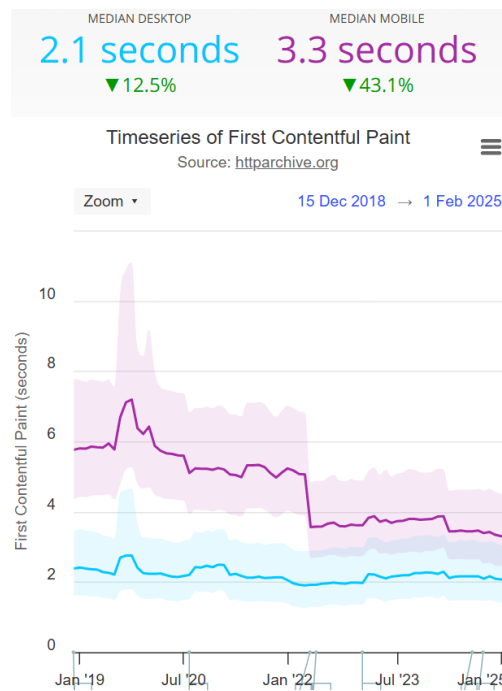


Gambar 2.11: Perkembangan jumlah kilobyte yang dapat dihemat dengan melakukan kompresi pada gambar

### 1 Loading Speed

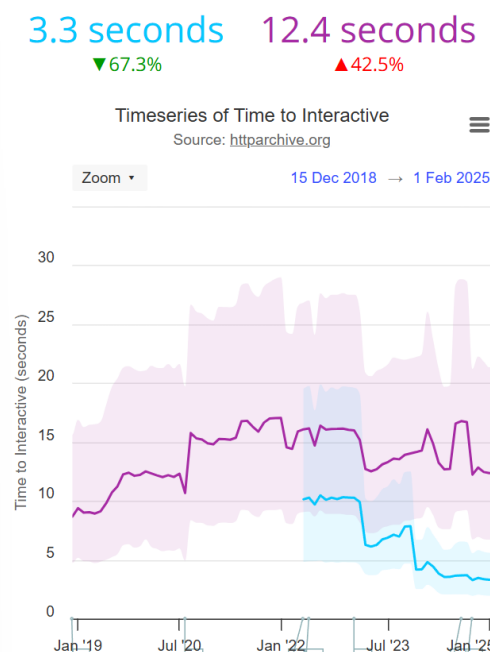
Performa *web* dapat berpengaruh langsung terhadap bisnis seperti kepuasan pengguna. Laporan ini akan menganalisis berbagai matriks performa dalam siklus pemuatan halaman *web* termasuk yang digunakan oleh aplikasi *web* modern. Kategori ini memiliki beberapa laporan sebagai berikut:

- *First Contentful Paint* yang berisi waktu dalam detik yang dibutuhkan untuk menampilkan konten utama dari sebuah *web* ke layar sejak navigasi dimulai. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.12. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya penurunan waktu pada perangkat *mobile*, ini menunjukan bahwa waktu yang dibutuhkan untuk menampilkan konten utama semakin cepat. Untuk perangkat *desktop* waktu yang dibutuhkannya cenderung stabil namun dari visualisasi yang ditunjukan terlihat bahwa perangkat *desktop* lebih cepat dalam menampilkan konten utama dibandingkan dengan perangkat *mobile*.



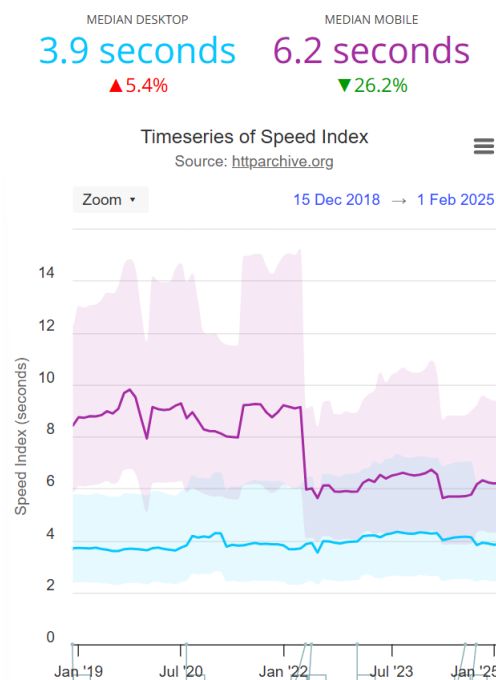
Gambar 2.12: Perkembangan waktu yang dibutuhkan untuk menampilkan konten utama

- *Time to interactive* yang berisi waktu yang dibutuhkan agar CPU stabil kembali setidaknya lima detik. Matriks ini diambil dari *Lighthouse*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.13. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan waktu pada perangkat *mobile*. Sedangkan untuk perangkat *desktop* terlihat penurunan yang signifikan. Data untuk perangkat *desktop* dimulai dari 1 Mei 2022 karena *Lighthouse* baru melakukan integrasi dengan perangkat *desktop*.



Gambar 2.13: Perkembangan waktu untuk sebuah *web* merespon sebuah navigasi

- *JavaScript Boot-up Time* yang berisi jumlah waktu CPU yang dibutuhkan setiap skrip dari setiap halaman. Matriks penilaian ini berasal dari *Lighthouse*. Contoh visualisasi dari data ini dapat dilihat pada Gambar 2.7. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan jumlah waktu yang dibutuhkan oleh perangkat *mobile*. Data untuk perangkat *desktop* tidak tercatat secara lengkap karena *Lighthouse* baru melakukan integrasi pada 1 Mei 2022.
- *Speed Index* yang berisi seberapa cepat konten sebuah halaman terlihat secara jelas. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.14. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya penurunan waktu yang dibutuhkan oleh perangkat *mobile*. Hal ini menunjukkan adanya peningkatan kecepatan untuk konten dapat terlihat jelas. Sedangkan untuk perangkat *desktop* waktu yang dibutuhkan cenderung stabil.

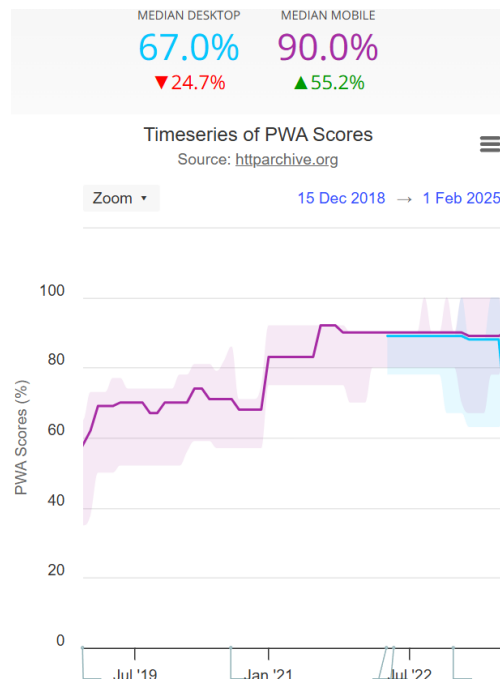


Gambar 2.14: Perkembangan waktu yang dibutuhkan untuk konten dapat terlihat secara jelas

### Progressive Web App

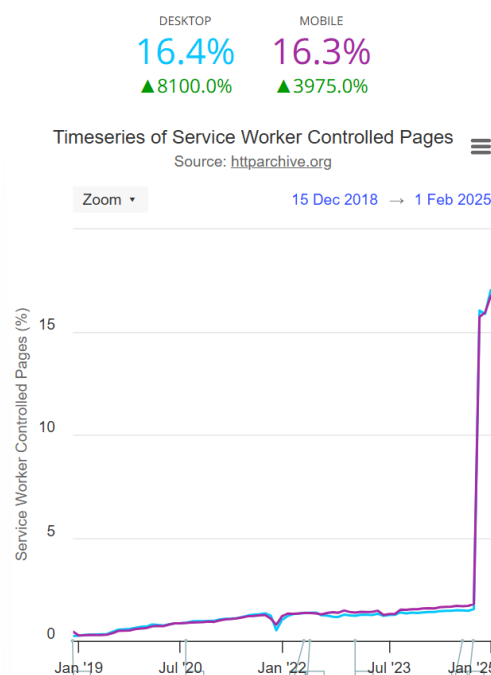
Laporan ini akan mengkaji status dari *Progressive Web App*. *Progressive Web App* adalah kelas baru dari aplikasi *web* yang disediakan oleh *Service Workers APIs*. *Service Workers* memungkinkan aplikasi untuk mendukung proses muat jaringan secara independen, menerima *push notifications* untuk menyinkronkan data di *background*. Kategori ini mencakup beberapa laporan seperti:

- *PWA Scores* yang berisi median dari skor PWA yang terdapat pada *Lighthouse*. *Lighthouse* adalah alat otomatis yang dapat digunakan untuk meningkatkan performa *web*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.15. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa skor PWA pada perangkat *mobile* mengalami peningkatan. Sedangkan pada perangkat *desktop* mengalami penurunan. Data untuk perangkat *desktop* tidak tercatat secara lengkap karena *Lighthouse* baru melakukan integrasi pada 1 Mei 2022.



Gambar 2.15: Perkembangan skor PWA

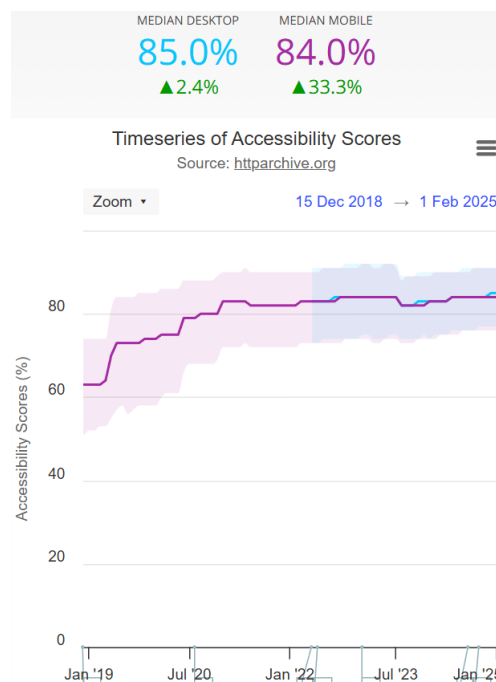
- *Service Worker Controlled Pages* yang berisi persentase dari jumlah halaman yang telah memicu `ServiceWorkerControlledPage use counter` yang aktif ketika sebuah halaman *web* dikendalikan oleh *service worker*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.16. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan yang sangat signifikan pada 1 Oktober 2024. Pada titik ini halaman *web* yang didukung IPv6 baru ditambahkan.

Gambar 2.16: Perkembangan persentase halaman yang membuat *service worker* bekerja

## Accessibility

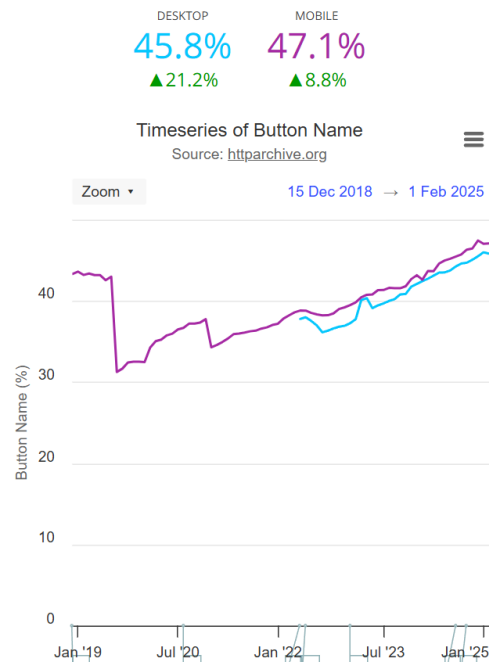
Laporan ini menjelaskan tingkat aksesibilitas dari sebuah halaman *web*. Penilaian ini dilakukan oleh *Lighthouse*. Kategori ini berisi beberapa laporan seperti:

- *Accessibility Score* yang berisi sebaran nilai kategori aksesibilitas dalam *Lighthouse*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.17. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan pada perangkat *mobile* dan *desktop*. Hal ini menandakan bahwa halaman *web* yang dibuat memiliki tingkat aksesibilitas yang semakin baik. Data untuk perangkat *desktop* tidak tercatat secara lengkap karena *Lighthouse* baru melakukan integrasi pada 1 Mei 2022.



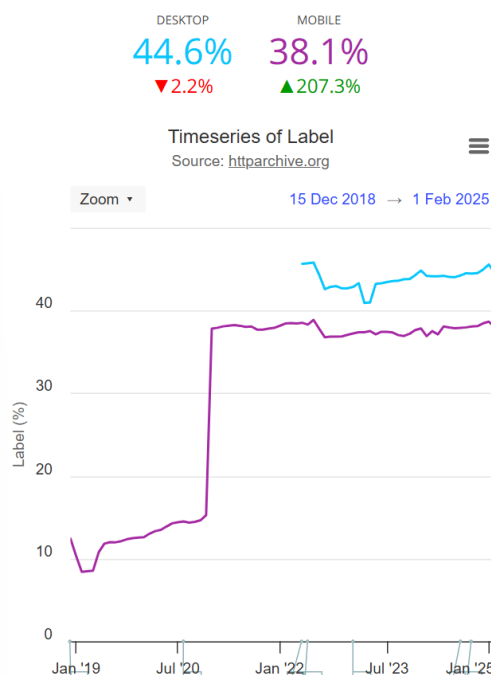
Gambar 2.17: Perkembangan persentase skor aksesibilitas

- *Button Name* yang berisi persentase halaman yang berhasil melalui audit *Lighthouse* yang memeriksa apakah *buttons* atau tombol memiliki nama yang aksesibel. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.18. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya penurunan yang signifikan 1 Agustus 2019 namun kemudian mengalami peningkatan di bulan-bulan selanjutnya. Pada perangkat *desktop* persentasenya terus meningkat. Hal ini menandakan bahwa nama yang digunakan untuk tombol memiliki tingkat aksesibel yang semakin tinggi. Data untuk perangkat *desktop* tidak tercatat secara lengkap karena *Lighthouse* baru melakukan integrasi pada 1 Mei 2022.



Gambar 2.18: Perkembangan persentase *web* yang memiliki nama tombol yang aksesibel

- *Label* yang berisi persentase halaman yang berhasil melalui audit dari *Lighthouse* yang memeriksa apakah semua elemen memiliki label yang terkait. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.19. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan yang signifikan pada 1 Januari 2021 pada perangkat *mobile*. Pada perangkat *desktop* persentasenya cenderung stabil. Data untuk perangkat *desktop* tidak tercatat secara lengkap karena *Lighthouse* baru melakukan integrasi pada 1 Mei 2022.



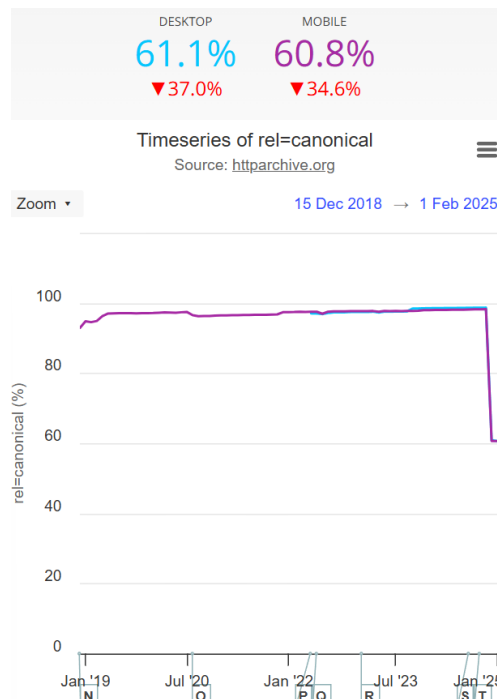
Gambar 2.19: Perkembangan persentase halaman yang menggunakan label yang aksesible



## 1 SEO

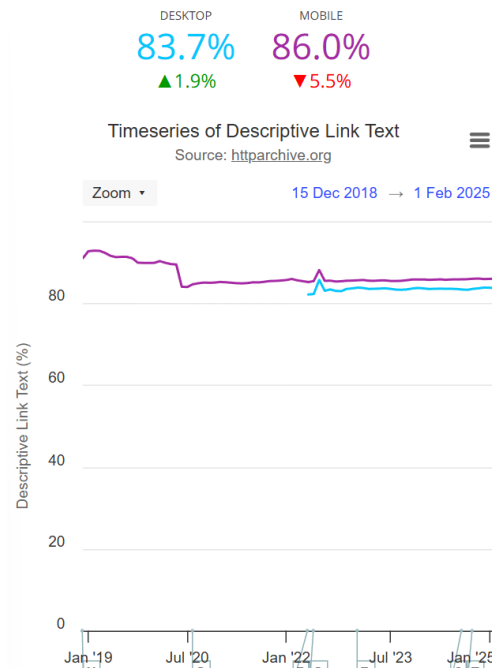
2 Laporan yang akan menelusuri penggunaan beberapa teknik agar halaman *web* dapat dikenali oleh  
3 mesin pencarian secara lebih baik. Kategori ini memiliki beberapa laporan seperti:

- 4 • *rel=canonical* yang berisi persentase halaman yang memiliki *link* kanonikal yang valid. Halam-  
5 an yang kanonikal dideteksi oleh *Lighthouse*. Contoh visualisasi untuk data ini dapat dilihat  
6 pada Gambar 2.20. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari  
7 2025 pada perangkat *desktop* dan *mobile* bahwa perkembangan halaman yang memiliki *link*  
8 yang *canonical* untuk perangkat *desktop* dan *mobile* cenderung stabil. Data untuk perangkat  
9 *desktop* tidak tercatat secara lengkap karena *Lighthouse* baru melakukan integrasi pada 1 Mei  
10 2022.



Gambar 2.20: Perkembangan persentase halaman yang memiliki *link canonical* yang valid

- 11 • *Descriptive Link Text* yang berisi persentase halaman yang memiliki *link* yang deskriptif.  
12 Tingkat *deskriptif* sebuah link diukur oleh *Lighthouse*. Contoh visualisasi untuk data ini  
13 dapat dilihat pada Gambar 2.21. Terlihat dari data yang diambil dari 15 Desember 2018  
14 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa persentase halaman yang  
15 memiliki link deskriptif cenderung stabil.

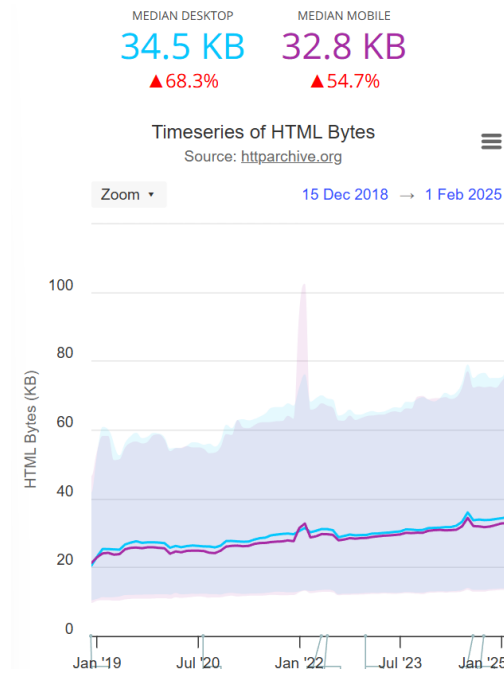


Gambar 2.21: Perkembangan persentase halaman yang memiliki *link* yang deskriptif

## 1 Page Weight

Laporan ini menelusuri ukuran dan banyaknya *resource* dari banyak halaman *web* populer. Ukuran dalam hal ini merepresentasikan jumlah *byte* yang dikirimkan melalui jaringan. Kategori ini memiliki beberapa laporan seperti:

- *Total Kilobytes* yang berisi jumlah kilobyte dari semua *resource* yang diminta oleh halaman *web*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.2. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa jumlah *kilobyte* yang *direquest* tidak mengalami banyak perubahan atau stabil.
- *JavaScript Bytes* yang berisi jumlah kilobyte yang diminta oleh skrip eksternal dari sebuah halaman *web*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.5. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan *kilobytes* setiap tahun nya. Hal ini menandakan bahwa semakin banyak *resource* yang berbentuk JavaScript yang digunakan oleh halaman *web*.
- *HTML Bytes* yang berisi jumlah kilobyte yang diminta oleh dokumen HTML dari sebuah halaman *web*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.22. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa jumlah *kilobytes* pada perangkat *mobile* dan *desktop* mengalami peningkatan namun meningkat secara perlahan.

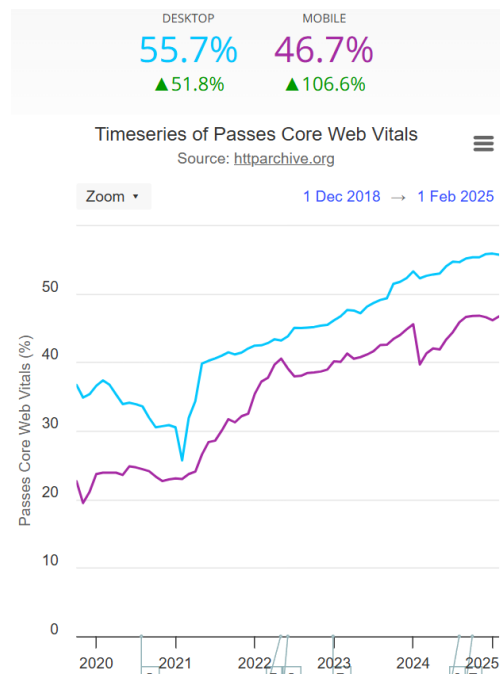


Gambar 2.22: PERkembangan jumlah *kilobytes* oleh dokumen HTML

## 1 *CrUx*

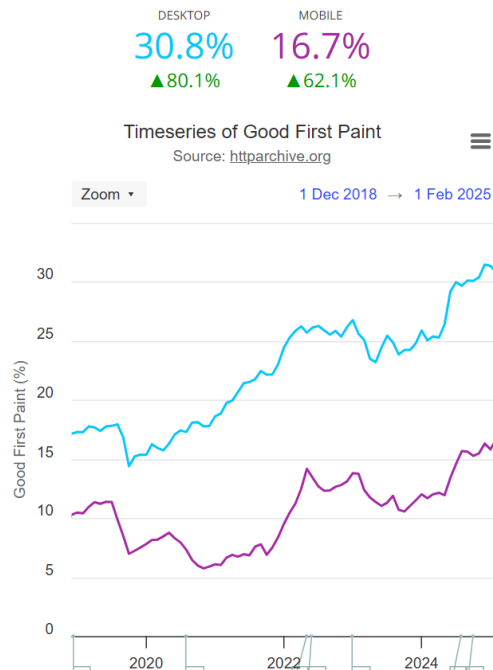
2 Laporan ini akan menelusuri tingkat interaktivitas dan proses muat dari pengguna *Chrome* di  
3 dunia nyata melalui berbagai kondisi perangkat keras dan jaringan. Kategori ini memiliki beberapa  
4 laporan seperti:

- 5 • *Passes Core Web Vitals* yang bersisi persentase halaman *web* yang berhasil lolos degan penilaian  
6 baik dari tiga matriks penilaian *Core Web Vitals*. Contoh visualisasi untuk data ini dapat  
7 dilihat pada Gambar 2.23. Terlihat dari data yang diambil dari 1 Desember 2018 hingga 1  
8 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa pada perangkat *desktop* dan *mobile*  
9 mengalami peningkatan. Ini menunjukkan bahwa kualitas halaman *web* semakin baik.



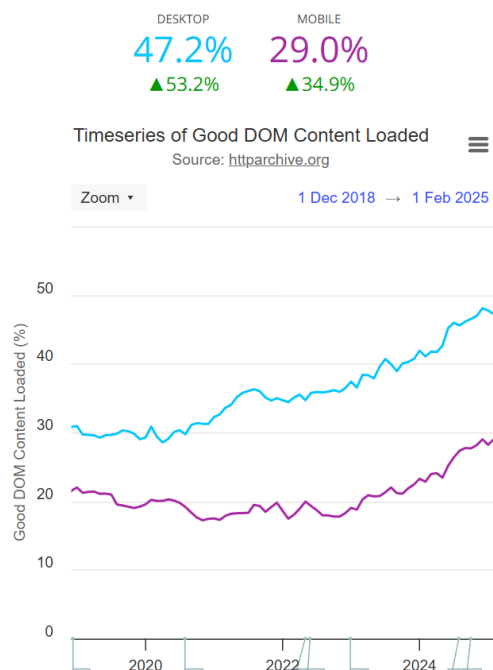
Gambar 2.23: Perkembangan oersentase halaman *web* yang memiliki penilaian *core web vitalis* yang bagus

- *Good First Paint* yang berisi persentase halaman *web* yang memiliki pengalaman *First Paint* yang “baik”. Waktu yang dikatakan baik adalah saat halaman *web* mendapat skor FP kurang dari sama dengan satu detik. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.24. Terlihat dari data yang diambil dari 1 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa persentase halaman yang memiliki penilaian yang baik meningkat. Hal ini menunjukkan bahwa semakin banyak halaman *web* memiliki pengalaman *First Paint* yang baik.



Gambar 2.24: Contoh perkembangan persentase halaman yang memiliki pengalaman *First Paint* yang baik

- *Good DOM Content Loaded* yang berisi persentase halaman *web* yang memiliki pengalaman DCL(DOMContentLoaded) yang “baik”. Nilai yang dikatakan baik adalah halamanya yang berhasil dimuat kurang dari sama dengan satu detik. Contoh visualisasi untuk data ini dapat dilihat pada Gambar ???. Terlihat dari data yang diambil dari 1 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa adanya peningkatan persentase halaman *web* yang memiliki pengalaman DCL yang baik pada perangkat *desktop* dan *mobile*.

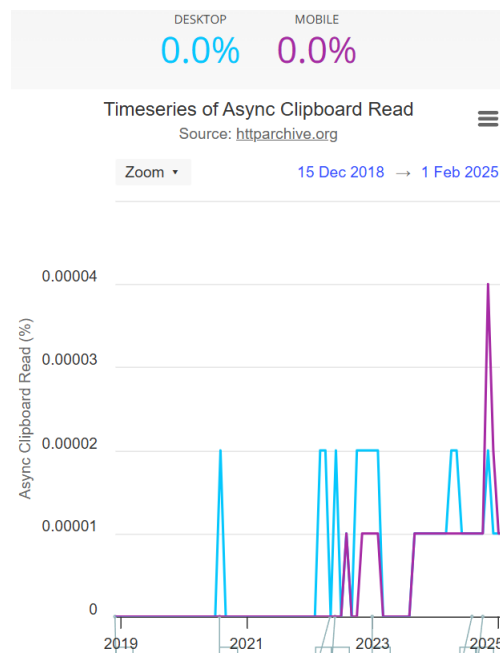


Gambar 2.25: Contoh perkembangan persentase halaman *web* yang memiliki pengalaman DCL yang baik

## 1 *Capabilities*

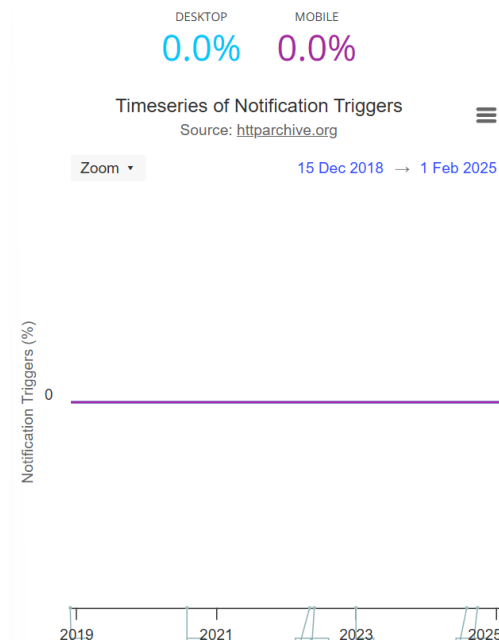
2 *Capabilities Project* adalah usaha yang dilakukan oleh *Google* dengan perusahaan lain untuk  
 3 memungkinkan sebuah aplikasi *web* dapat melakukan hal yang dapat dilakukan oleh aplikasi bawaan  
 4 dari sistem operasi sambil tetap mempertahankan keamanan pengguna, privasi, kepercayaan, dan  
 5 prinsip dasar lainnya dari *web*. Kategori ini memiliki beberapa laporan seperti:

- 6 • *Async Clipboard Read* yang berisi persentase halaman yang membaca data dari *clipboard* yang  
 7 dimiliki sistem melalui API *Async Clipboard*. Contoh visualisasi untuk data ini dapat dilihat  
 8 pada Gambar 2.26. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari  
 9 2025 pada perangkat *desktop* dan *mobile* bahwa perkembangannya sangat tidak stabil karena  
 10 hasil visualisasinya memperlihatkan garis yang naik turun secara ekstrem.

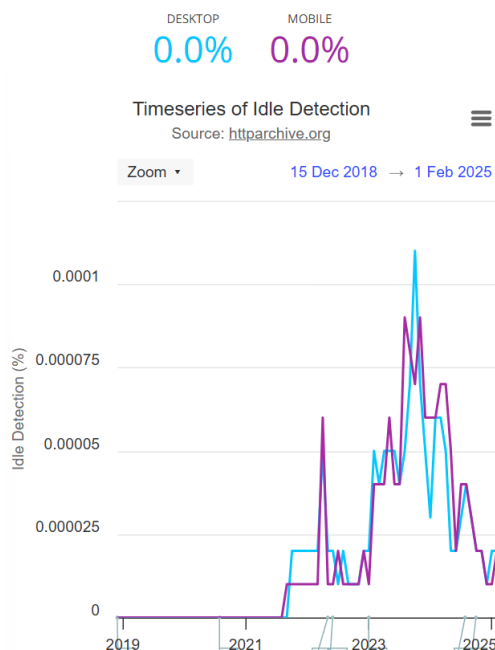


Gambar 2.26: Perkembangan persentase halaman *web* yang membaca *clipboard* milik sistem.

- 11 • *Notification Triggers* yang berisi persentase halaman yang menggunakan pemberitahuan  
 12 terjadwal dengan menggunakan API *Notification Trigger*. Contoh visualisasi untuk data ini  
 13 dapat dilihat pada Gambar 2.27. Terlihat dari data yang diambil dari 15 Desember 2018  
 14 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa tidak ada halaman *web*  
 15 yang menggunakan API *Notification Trigger*.

Gambar 2.27: Perkembangan penggunaan API *Notification Trigger*

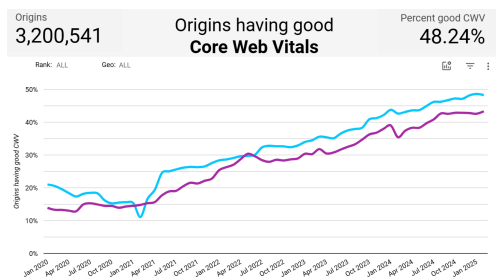
- *Idle Detection* yang berisi persentase halaman yang mendeteksi penggunaanya sedang melakukan *idle* melalui API *Idle Detection*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.28. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa persentase halaman yang menggunakan API ini sangat fluktuatif. Hal ini ditunjukkan dari bentuk visualisasi yang menunjukkan peningkatan dan penurunan dalam waktu yang cepat dan sangat ekstrem.

Gambar 2.28: Perkembangan persentase halaman *web* penggunaan API *Idle Detection*

## 1 Core Web Vitals Technology Report

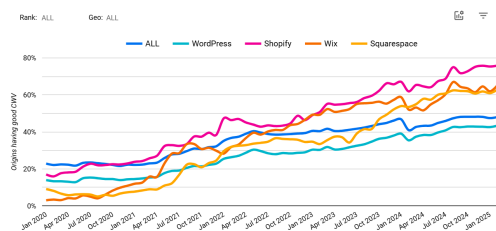
Laporan *Core Web Vitals Technology* merupakan dasbor yang berisi gabungan dari pengalaman nyata pengguna *chrome*(CrUX) dengan pendeteksi teknologi *web* yang dimiliki *HTTP Archive*, yang memungkinkan analisis terhadap cara sebuah *web* dibangun dan pengalaman saat *web* tersebut digunakan. Kategori ini memiliki beberapa laporan seperti:

- *Technology Drilldown* yang berisi informasi yang lebih detail mengenai satu teknologi. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.29. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* dan *mobile* bahwa persentase halaman yang memiliki skor CWV yang baik dengan menggunakan teknologi *WordPress* semakin meningkat.



Gambar 2.29: Perkembangan persentase halaman *web* yang menggunakan teknologi *WordPress* dengan skor CWV yang baik

- *Technology Comparasion* yang bersi informasi yang lebih detail mengenai dua sampai sepuluh teknologi. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.30. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *mobile* bahwa semua teknologi memiliki peningkatan skor CWV yang baik.



Gambar 2.30: Perkembangan persentase halaman *web* yang memiliki skor CWV yang bagus dengan berbagai teknologi

- *Category* yang berisi informasi yang lebih detail mengenai sebuah kategori dan teknologi yang dimilikinya.

### 2.1.2 2024 Web Almanac

*Web Almanac* merupakan kombinasi dari statistik yang masih mentah dan tren yang ada di *HTTP Archive* dengan keahlian komunitas *web*. *Web Almanac* memiliki 19 bab yang berisi hal berikut ini:

1. *JavaScript* Bab yang berisi hasil evaluasi peran *JavaScript* dalam dunia *web*.
2. *Markup* Bab yang berisi tentang analisis dari halaman HTML yang dimiliki oleh *web* yang dievaluasi.



3. *Structured Data* Bab yang berisi hasil analisis tren di tahun lalu dan memeriksa perkembangan cepat yang terjadi.
4. *Fonts* Bab yang berisi hasil analisis penggunaan *web font*.
5. *Media* Bab yang berisi hasil analisis tentang HTTP/1.1, HTTP/2 and HTTP/3 manajemen media seperti video dan gambar.
6. *Third Parties* Bab yang berisi hasil analisis empiris yang menjelaskan praktik penggunaan pihak ketiga pada *web*.
7. *SEO* Bab yang menjelaskan mengenai elemen dan pengaturan yang penting agar *web* dapat terlihat pada pencarian.
8. *Accessibility* Bab yang berisi solusi yang ditawarkan agar memiliki tingkat aksesibilitas yang baik.
9. *Performance* Bab yang berisi hasil analisis performa berdasarkan skor *core web vitals*.
10. *Privacy* Bab yang berisi ringkasan mengenai *online tracking*.
11. *Security* Bab yang berisi hasil analisis proteksi dan praktek keamanan yang digunakan oleh halaman *web* saat ini.
12. *CMS* Bab yang berisi hasil analisis terhadap variasi dalam sistem manajemen konten yang dimiliki berbagai halaman *web*.
13. *Ecommerce* Bab yang berisi ringkasan mengenai ekosistem *e-commerce* dalam *web*
14. *JamStack* Bab yang berisi analisis mengenai penggunaan tiga arsitektur halamannya yang digunakan pada *web*.
15. *Sustainability* Bab yang berisi bagaimana mengurangi dampak dari ekosistem *web* terhadap lingkungan.
16. *Page Weight* Bab yang berisi hasil analisis terhadap bobot halaman yang dapat dibuka oleh semua pengguna dalam berbagai kondisi.
17. *CDN* Bab yang berisi hasil kajian terhadap *Content Delivery Network* dan peran pentingnya dalam ekosistem digital saat ini.
18. *HTTP* Bab yang berisi hasil analisis penggunaan HTTP/1.1, HTTP/2, dan HTTP/3 serta perkembangannya saat ini.
19. *Cookies* Bab yang berisi hasil analisis struktur *cookie* dari halaman *web*.

Metode yang digunakan oleh *HTTP Archive* dalam mengumpulkan data perkembangan pembuatan *web* sejak 2010 adalah dengan menggunakan *WebPageTest* dan *Lighthouse*. *Lighthouse* adalah sebuah alat *open-source* yang disediakan oleh *Google* untuk meningkatkan kualitas halaman *web* [4]. *Lighthouse* dapat melakukan pemeriksaan terhadap performa, aksesibilitas, dan SEO (*Search Engine Optimization*) dari sebuah halaman *web*.

### 2.1.3 Public Dataset

*HTTP Archive* memberikan akses terhadap informasi yang lebih detail mengenai hal-hal yang ada di setiap *website*, seperti *metadata* dari *request* dan *response*, *response bodies*, jejak eksekusi, dan lain-lain. *HTTP Archive* memberikan beberapa tabel yang terbagi ke dalam beberapa *dataset* seperti *crawl*, *sample\_data*, dan *wappalyzer*. *Dataset* sendiri adalah unit logis utama untuk mengelompokkan dan mengatur *view*, *resource*, dan tabel pada *Google BigQuery* dataset-dataset tersebut kemudian memiliki beberapa tabel yang dapat digunakan seperti berikut ini:

- 1 • *pages* adalah tabel yang terdapat pada dataset *crawl* dan berisi detail mengenai halaman yang  
2 dites oleh *HTTP Archive*. Data ini memiliki kolom sebagai berikut:
  - 3 – *date*
  - 4 – *client*
  - 5 – *page*
  - 6 – *is\_root\_page*
  - 7 – *root\_page*
  - 8 – *rank*
  - 9 – *wptid*
  - 10 – *payload*
  - 11 – *summary*
  - 12 – *custom\_metric*
  - 13 – *lighthouse*
  - 14 – *features*
  - 15 – *technologies*
  - 16 – *metadata*
- 17 • *pages\_10k* adalah tabel yang terdapat pada dataset *sample\_data* yang merupakan potongan  
18 dari tabel *pages* pada dataset *crawl* yang berisi halaman-halaman yang dites dan memiliki  
19 peringkat popularitas diatas 10.000 berdasarkan CrUX. Tabel ini memiliki kolom sebagai  
20 berikut:
  - 21 – *date*
  - 22 – *client*
  - 23 – *page*
  - 24 – *is\_root\_page*
  - 25 – *root\_page*
  - 26 – *rank*
  - 27 – *wptid*
  - 28 – *payload*
  - 29 – *summary*
  - 30 – *custom\_metric*
  - 31 – *lighthouse*
  - 32 – *features*
  - 33 – *technologies*
  - 34 – *metadata*
- 35 • *requests* adalah tabel yang terdapat pada dataset *crawl* yang berisi detail *request* dari halaman  
36 yang diuji oleh *HTTP Archive*. TAbel ini memiliki kolom sebagai berikut:
  - 37 – *date*
  - 38 – *client*
  - 39 – *page*
  - 40 – *is\_root\_page*
  - 41 – *root\_page*
  - 42 – *url*

- *is\_main\_document*
- *type*
- *index*
- *payload*
- *summary*
- *request\_header*
- *reponse\_header*
- *response\_body*
- *technologies* adalah tabel yang berisi *dataset wappalyzer* yang berisi mengenai detail dari teknologi yang dipakai oleh berbagai *web*. Tabel ini memiliki tabel sebagai berikut:
  - *technologies*
  - *categories*
  - *info*

## 2.2 Structure Query Language

*Structure Query Language* atau yang biasa disebut SQL merupakan bahasa pemrograman yang bertujuan untuk menganalisis dan mengolah basis data relasional. Lakshman dalam bukunya menjelaskan bahwa SQL sangat cocok digunakan dalam lingkungan analitik yang besar seperti dalam lingkungan *Big Query* [5]. Sintaks yang dapat dipakai adalah sebagai berikut:

- **SELECT** merupakan sintaks yang berguna untuk memilih bagian data yang dibutuhkan dari tabel tertentu. Sintaks ini memiliki beberapa ekspresi seperti
  - Ekspresi ini akan mengembalikan sebuah kolom keluaran untuk semua kolom yang terlihat setelah menjalankan semua kueri.
  - *expression* Ekspresi ini akan mengembalikan sebuah nilai dan menghasilkan sebuah kolom keluaran.
  - *expression*. \* Ekspresi ini akan menghasilkan satu kolom keluaran untuk setiap kolom atau bidang ekspresi tingkat atas.

Kode 2.1: contoh penggunaan sintaks **SELECT**

```

1  SELECT
2     date, client
3  FROM
4     'httparchive.crawl.pages'
```

Kueri pada potongan Kode 2.1 akan mengembalikan tabel yang berisi kolom *date* dan *client* dari tabel *httparchive.crawl.pages*.

- **FROM** merupakan sintaks yang berguna untuk menunjukkan asal tabel yang kolomnya dipilih oleh sintaks **SELECT**. **FROM** biasanya diikuti dengan nama tabel yang sudah ada seperti pada Kode 2.1 data yang dipilih berasal dari struktur 'httparchive', dataset 'crawl', dan tabel 'pages'.
- **WHERE** adalah sintaks yang berguna untuk memberikan kondisi tertentu dalam memilih data. Kondisi yang diinginkan bisa lebih dari satu maka dari itu untuk menggabungkan beberapa kondisi dapat digunakan fungsi **AND**, **OR**, atau **BETWEEN**. Sintaks ini digunakan dengan cara **SELECT nama, kolom FROM nama.tabel WHERE kondisi1 AND kondisi2**

Kode 2.2: contoh penggunaan sintaks `WHERE`

```
1 SELECT
2   date, client
3 FROM
4   'httparchive.crawl.pages'
5 WHERE
6   date = "2024-01-01"
```

Kueri pada potongan Kode 2.2 akan mengembalikan tabel yang berisi *date* dan *client* dari tabel *httparchive.crawl.pages* dengan kondisi yang diinginkan adalah nilai dalam kolom *date* yang diambil adalah "2024-01-01".

- **GROUP BY** adalah sintaks yang berguna untuk mengelompokkan data berdasarkan kelas tertentu yang terdapat dalam data. Pengelompokan ini dapat dilakukan berdasarkan lebih dari satu kelas. Sintaks ini digunakan dengan cara `SELECT nama, kolom, agregat FROM nama.tabel WHERE kondisi1`. Kolom agregat adalah kolom yang akan dipakai sebagai kategori pemisah data.

Kode 2.3: contoh penggunaan sintaks `GROUP BY`

```
1 SELECT
2   date, client
3 FROM
4   'httparchive.crawl.pages'
5 WHERE
6   date = "2024-01-01"
7 GROUP BY
8   client
```

Kueri pada potongan Kode 2.3 akan mengembalikan tabel yang berisi *date* dan *client* dari tabel *httparchive.crawl.pages* dengan kondisi yang diinginkan adalah nilai dalam kolom *date* yang diambil adalah "2024-01-01". Namun tabel hasil kueri ini akan berisi data yang sudah dikelompokkan berdasarkan kelas *client*.

- **UNNEST** adalah fungsi yang mengembalikan elemen dari sebuah *array* menjadi sebuah baris atau *row*. Sintaks ini digunakan dengan cara `SELECT nama, kolom, elemen FROM nama.tabel, UNNEST(arr`. Array adalah nama *array* yang elemen nya ingin dikembalikan menjadi baris dan elemen adalah bagian dari *array* yang dikembalikan menjadi baris.

Kode 2.4: contoh penggunaan sintaks `UNNEST`

```
1 SELECT
2   date, client, t.technology
3 FROM
4   'httparchive.crawl.pages',
5   UNNEST (technologies) as t
6 WHERE
7   date = "2024-01-01"
8 GROUP BY
9   client
```

Kueri pada potongan Kode 2.4 akan mengembalikan tabel yang berisi *date*, *client*, dan *technology* dari tabel *httparchive.crawl.pages* dengan kondisi yang diinginkan adalah nilai dalam kolom *date* yang diambil adalah "2024-01-01". Namun tabel hasil kueri ini akan berisi data yang sudah dikelompokkan berdasarkan kelas *client*. Kueri ini juga mengembalikan isi array *technologies* menjadi baris-baris baru

- **COUNT** adalah fungsi agregat yang berguna untuk menghitung berapa banyak row yang memiliki nilai tertentu. Sintaks ini digunakan dengan cara `SELECT nama, kolom, COUNT(agregat) FROM nama.tabel`. `COUNT(agregat)` akan menghitung jumlah agregat berdasarkan nilai yang ada di kolom agregat.

Kode 2.5: contoh penggunaan sintaks **COUNT**

```

1  SELECT
2  date, client, t.technology, COUNT(t.technology)
3  FROM
4  'httparchive.crawl.pages,
5  UNNEST (technologies) as t
6  WHERE
7  date = "2024-01-01"
8  GROUP BY
9  client, t.technology

```

Kueri pada potongan Kode 2.5 akan mengembalikan tabel yang berisi *date*, *client*, *technology*, dan jumlah dari setiap teknologi dari tabel *httparchive.crawl.pages* dengan kondisi yang diinginkan adalah nilai dalam kolom *date* yang diambil adalah "2024-01-01". Namun tabel hasil kueri ini akan berisi data yang sudah dikelompokkan berdasarkan kelas *client*. Kueri ini juga mengembalikan isi array *technologies* menjadi baris-baris baru.

- **BETWEEN** adalah sintaks yang berguna untuk memberikan rentang pada tipe data *date*. Sintaks ini digunakan dengan cara **SELECT** nama, kolom, agregat **FROM** nama.tabel **WHERE** tanggal **BETWEEN** tanggal1 dan tanggal2 adalah rentang tanggal dimana data akan dikembalikan.

Kode 2.6: contoh penggunaan sintaks **BETWEEN**

```

1  SELECT
2  date, client, t.technology, COUNT(t.technology)
3  FROM
4  'httparchive.crawl.pages,
5  UNNEST (technologies) as t
6  WHERE
7  date BETWEEN "2024-01-01" and "2025-01-01"
8  GROUP BY
9  client, t.technology

```

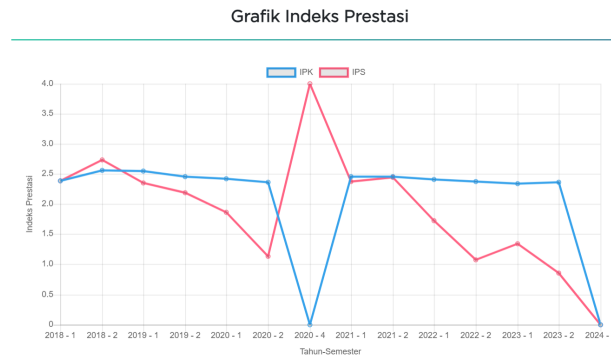
Kueri pada potongan Kode 2.5 akan mengembalikan tabel yang berisi *date*, *client*, *technology*, dan jumlah dari setiap teknologi dari tabel *httparchive.crawl.pages* dengan kondisi yang diinginkan adalah nilai dalam kolom *date* yang diambil berkisar antara "2024-01-01" dan "2025-01-01". Namun tabel hasil kueri ini akan berisi data yang sudah dikelompokkan berdasarkan kelas *client*. Kueri ini juga mengembalikan isi array *technologies* menjadi baris-baris baru

- **BETWEEN** adalah sintaks yang berguna untuk memberikan rentang pada tipe data *date*. sintaks-sintaks tersebut merupakan sebagian kecil dari sintaks yang dimiliki oleh bahasa SQL.

## 2.3 Visualisasi Data

### 2.3.1 Line Plot

*Line plot* merupakan teknik visualisasi data yang menggunakan batang vertikal atau horizontal untuk menunjukkan nilai-nilai dari data. Visualisasi ini berguna untuk menunjukkan pengukuran statistik sebuah data secara terpisah. *Line Plot* memiliki elemen utama yaitu sumbu *x* dan sumbu *y*. Gambar 2.31 merupakan contoh penggunaan *line plot* untuk memvisualisasikan data di mana pada contoh ini perkembangan tren dari IPK dan IPS yang dianalisis terlihat adanya penurunan IPS secara signifikan pada tahun 2020 semester 04 namun pada waktu yang sama IPK mengalami kenaikan yang signifikan.



Gambar 2.31: Contoh visualisasi dari perkembangan IPK dan IPS dengan menggunakan *line plot*

## 2.4 Tk Interface [2]

*Tk Interface* adalah sebuah antarmuka *python* standar untuk membuat aplikasi antarmuka grafis yang merupakan *toolkit* dari gabungan bahasa Tcl dan Antarmuka Grafis Tk (Tcl/Tk). Tcl/Tk bukan *library* yang berdiri sendiri melainkan gabungan dari beberapa *module* yang memiliki fungsi berbeda. *TK Interface* sendiri memiliki arsitektur sebagai berikut:

- Tcl

Tcl merupakan sebuah bahasa pemrograman seperti *Python*. Bahasa ini dapat digunakan sebagai bahasa pemrograman dengan tujuan yang umum. Bahasa ini biasanya terdapat dalam aplikasi C sebagai mesin *scripting* atau sebagai antarmuka untuk TK. *Library Tcl* memiliki antarmuka C untuk membuat atau mengelola satu atau lebih *interpreter Tcl*, menjalankan perintah dan skrip Tcl, dan menambahkan perintah kustom untuk dijalankan dalam C atau Tcl. Setiap interpreter memiliki antrian *events* dan terdapat fasilitas untuk mengirimkan *events* ke dalamnya untuk kemudian memprosesnya. Tidak seperti *Python* model eksekusi Tcl dirancang berdasarkan *multitasking* kooperatif dan *Tkinter* menjembatani perbedaan ini.

- Tk

Tk adalah paket Tcl yang diimplementasikan dalam C yang menambahkan perintah kustom untuk membuat atau memanipulasi *widget* antarmuka grafis. Setiap objek tk memiliki interpreter Tcl nya sendiri dengan Tk yang dimuat ke dalamnya. Tk menggunakan antrian *event* milik Tcl untuk membuat dan memproses *event* dari *GUI*.

- Ttk

Ttk atau *Themed tk* adalah bagian *widget* Tk yang baru yang memberikan penampilan yang lebih baik pada *platform* yang berbeda. Ttk didistribusikan sebagai bagian dari Tk.

Ketika aplikasi *Python* menggunakan kelas *Tkinter* seperti untuk membuat *widget*, modul *Tkinter* akan terlebih dahulu menyusun urutan skrip *Tcl/Tk*. Tk juga memiliki beberapa konsep penting seperti:

- *widget*

Antarmuka pengguna Tkinter dibuat dari *widget* individual. Setiap *widget* direpresentasikan sebagai objek *Python*. yang dibuat dari kelas-kelas seperti `ttk.Frame`, `ttk.Label`, dan `ttk.Button`

- *widget hierarchy*

Setiap *widget* disusun dalam sebuah hierarki. Ketika membuat setiap *child widget*, *parent widget* nya dilewati sebagai argumen pertama dalam konsturktor *widget* nya

- *Geometry Management*

Setiap *widget* yang dibuat tidak langsung ditambahkan ke antarmuka pengguna namun *Geometry Manager* seperti **grid** mengatur dimana *widget* tersebut akan diletakan di antarmuka pengguna.





## BAB 3

### ANALISIS PENYELESAIAN MASALAH

Pada bab analisis masalah ini akan dibahas mengenai masalah yang akan diselesaikan beserta solusi yang ditawarkan. Selain itu bab ini juga akan membahas tentang contoh penyelesaian masalah dengan data yang skalanya lebih kecil.

#### 3.1 Analisis Masalah

Perkembangan dunia *web* selama lima tahun terakhir sangat pesat. Pesatnya perkembangan dunia *web* ini didukung oleh munculnya berbagai macam teknologi yang dapat digunakan untuk membuatnya. Banyaknya teknologi yang bermunculan ini tentunya digunakan untuk membuat *web* semakin baik dari sisi performa maupun pengalaman pengguna *web*. Perkembangan dari penggunaan teknologi ini kemudian dicatat oleh sebuah situs bernama *HTTP Archive*. Situs ini menyediakan data yang mencatat berbagai matriks penilaian yang dapat mengukur baik maupun buruknya performa sebuah *web*. Data yang disajikan tentunya tidak mudah untuk dimengerti oleh semua orang.

Solusi yang ditawarkan untuk mempermudah pengguna data untuk mengerti data yang disajikan adalah dengan cara melakukan visualisasi yang sesuai. Visualisasi yang dilakukan juga dapat disesuaikan dengan kebutuhan dari pengguna sehingga visualisasi yang digunakan akan berupa visualisasi yang interaktif. Data yang sudah didapatkan kemudian diolah dengan menggunakan bahasa SQL untuk kemudian hasil *query*, yang menghasilkan potongan data yang dibutuhkan, divisualisasikan ke dalam bentuk visualisasi yang dibutuhkan.

#### 3.2 Data Kecil

Bagian ini akan berisi tentang penyiapan data kecil kemudian setelah itu akan dilanjutkan dengan melakukan pengolahan dan visualisasi dengan menggunakan data kecil.

##### 3.2.1 Penyiapan Data Kecil

Data yang digunakan merupakan data *sample* yang diberikan oleh *HTTP Archive* yang berasal dari bagian *public dataset* dan menggunakan tabel *pages* seperti yang sudah dijelaskan pada [2.1.3](#). Data yang digunakan berasal dari data yang sudah diperbaharui oleh *HTTP Archive*. Data yang didapatkan adalah data dari tanggal 1 Februari 2024 sampai dengan 1 Februari 2025. Data yang digunakan akan lebih berfokus pada tanggal, teknologi, *client*, dan url utama dari halaman *web*,

sehingga kolom dan baris lainnya tidak akan digunakan. Pengumpulan data dilakukan dengan melakukan *query* dengan menggunakan *Google Big Query*. Hasil dari *query* tadi kemudian disimpan ke dalam format CSV (*Comma Separated Value*). Salah satu *query* yang dapat digunakan untuk mengumpulkan data dapat dilihat pada Kode 3.1. Kode tersebut akan mengembalikan data yang berisi kolom *date*, *client*, *root\_page*, dan *technology* yang berasal dari tabel *pages* dan data yang diambil berasal dari rentang waktu yang sudah disebutkan sebelumnya, namun data dibatasi 10.000.000 baris.

Kode 3.1: Kode untuk mengumpulkan data kecil

```

8
9 1
10 2 SELECT date, client, t.technology, COUNT(t.technology),
11 3 FROM 'httparchive.crawl.pages', UNNEST(technologies) as t
12 4 WHERE date BETWEEN "2024-02-01" and "2025-02-01" limit 100000000
13

```

### 3.2.2 Pengolahan Data Kecil

Setelah data yang akan digunakan siap, hal selanjutnya adalah mengolah data tersebut. Pengolahan data ini akan menggunakan bahasa *Python*. Dalam pengolahan ini perspektif yang dilihat adalah perkembangan sepuluh teknologi populer pada semua *client*, perkembangan sepuluh teknologi populer untuk *client mobile*, dan perkembangan sepuluh teknologi untuk *client desktop*.

#### Perkembangan Sepuluh Teknologi Populer

Hal pertama yang dilakukan adalah dengan memuat data *sample*. Contoh dari data *sample* yang dimuat dapat dilihat pada Tabel 3.1. Data tersebut adalah hasil dari menjalankan kode berisikan tanggal diambilnya data, perangkat yang digunakan untuk mengambil data, url utama dari halaman *web* yang dites, dan teknologi yang digunakan oleh halaman *web* yang dites. Data tersebut didapatkan dengan menjalankan *query* pada potongan Kode 3.1

Tabel 3.1: Lima baris data *sample*

date	client	technology	jumlah
2025-02-01	desktop	Google Analytics	12.370
2025-02-01	mobile	HSTS	131
2025-02-01	mobile	Klaviyo	
2025-02-01	mobile	Gatsby	268
2025-02-01	mobile	YouTube	20.2669
...	...	...	...

Setelah itu kemudian data dikelompokkan berdasarkan teknologinya dengan menggunakan Kode `df_gb = df.groupby(["technology"]).sum('f0_')`. Contoh hasil dari pengelompokan data ini dapat dilihat pada Tabel 3.2. Terlihat bahwa untuk tanggal yang sama ada berbagai teknologi dengan jumlah pemakaian dari berbagai halaman *web* yang beragam. Data yang dikelompokkan juga berasal dari penggunaan teknologi untuk perangkat *mobile* dan *desktop*.

Tabel 3.2: Contoh hasil pengelompokan data *sample* berdasarkan teknologi yang digunakan

date	technology	jumlah pemakaian
2024-02-01	Datadog	4.591
2024-02-01	Node.js	14.794
2024-02-01	Google Tag Manager	124.814
2024-02-01	Sendgrid	12.152
2024-02-01	Dojo	2.087
...	...	...

1 Kemudian data yang sudah terkelompok diurutkan berdasarkan penggunaan paling banyak.  
 2 Kode `df_sorted = df_gb.sort_values(by='f0_', ascending=False)` digunakan untuk mengu-  
 3 rutkan data. Contoh hasil pengurutan ini dapat dilihat pada Tabel 3.3. Terlihat bahwa teknologi  
 4 yang paling banyak digunakan pada rentang waktu 1 Februari 2024 sampai 1 Februari 2025 adalah  
 5 *jQuery* dengan 4.464.436 halaman *web* yang menggunakan teknologi ini untuk membangunnya.

Tabel 3.3: Contoh data penggunaan teknologi yang sudah diurutkan berdasarkan pemakaian paling banyak

technology	Jumlah Pemakaian
jQuery	4.464.436
Open Graph	3.834.676
Google Analytics	3.323.217
PHP	3.294.059
Google Font API	2.802.138
core-js	2.573.059
...	...

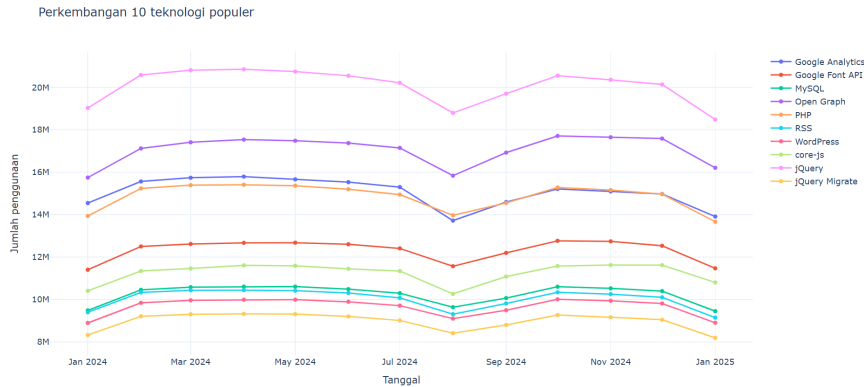
6 Setelah itu sepuluh teknologi yang paling banyak digunakan akan diambil. Kode yang digunakan  
 7 untuk mencari sepuluh teknologi yang paling banyak digunakan adalah `df_10 = df_sorted.head(10)`.  
 8 Sepuluh teknologi dengan penggunaan paling banyak dapat dilihat pada tabel 3.4. Terlihat bahwa  
 9 dua teknologi dari *jQuery* masuk sebagai sepuluh yang paling banyak dipakai di semua *client*

Tabel 3.4: Sepuluh teknologi dengan jumlah penggunaan paling banyak untuk semua *client*

technology	Jumlah Pemakaian
jQuery	4.464.436
Open Graph	3.834.676
Google Analytics	3.323.217
PHP	3.294.059
Google Font API	2.802.138
core-js	2.573.059
MySQL	2.338.106
RSS	2.236.824
WordPress	2.112.285
jQuery Migrate	2.041.754

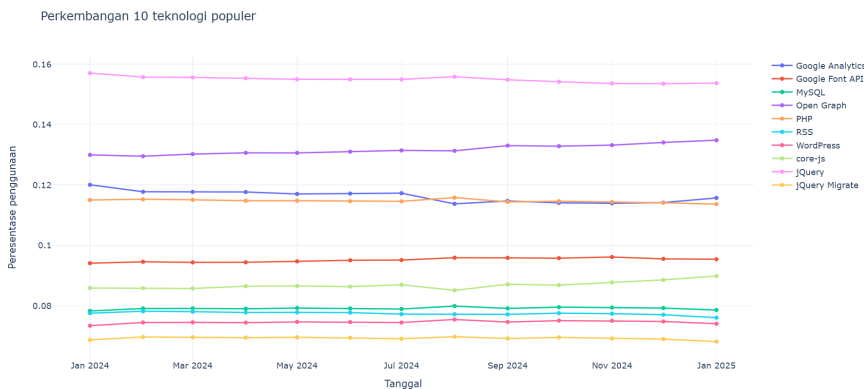
10 Setelah sepuluh teknologi yang populer sudah didapatkan, hal selanjutnya yang dicari ada-  
 11 lah penggunaan sepuluh teknologi tersebut dalam rentang waktu yang sudah ditentukan. Ko-

de `df_pivot = df_baru.pivot(index="date", columns='technology', values='f0_')` digunakan untuk melakukan hal ini. Hasilnya kemudian divisualisasikan dan hasilnya dapat dilihat pada Gambar 3.1. Terlihat bahwa pola yang dihasilkan untuk semua teknologi sama, sehingga jumlah penggunaan ini akan dibagi dengan jumlah pemakaian semua teknologi di bulan yang sama.



Gambar 3.1: Perkembangan jumlah penggunaan sepuluh teknologi populer

Hal ini dilakukan agar lebih terlihat persentase pemakaiannya. Setelah itu hal yang dilakukan adalah memvisualisasikan data yang sudah didapatkan. Hasil visualisasinya dapat dilihat pada Gambar 3.2. Terlihat dari hasil visualisasinya bahwa untuk semua teknologi mengalami penurunan pemakaian pada tanggal 1 Agustus 2024. Selain itu semua teknologi terlihat memiliki perkembangan yang hampir mirip dan stabil.



Gambar 3.2: Perkembangan sepuluh teknologi populer berdasarkan persentase penggunaannya

### Perkembangan Sepuluh Teknologi pada *Client mobile*

Hal pertama yang dilakukan adalah dengan memuat data *sample*. Contoh dari data *sample* yang dimuat dapat dilihat pada Tabel 3.1. Data tersebut berisikan tanggal diambilnya data, perangkat yang digunakan untuk mengambil data, url utama dari halaman *web* yang dites, dan teknologi yang digunakan oleh halaman *web* yang dites. Data tersebut didapatkan dengan menjalankan query pada potongan Kode 3.1

Kemudian data yang memiliki *client mobile* diambil untuk dilakukan pengolahan lebih lanjut. Kode yang digunakan untuk memisahkan data ini adalah `df_mobile = df[df['client'] == 'mobile']`. Kemudian data yang sudah dipisahkan dikelompokkan berdasarkan teknologi yang dipakai menggu-

1 nakan kode `df_mobile_gb = df_mobile.groupby(["technology"]).sum('f0_')`. Proses yang  
 2 dilakukan masih sama seperti pada Bagian 3.2.2 namun ada sedikit perbedaan pada sepuluh tekno-  
 3 logi yang paling banyak digunakan seperti yang terlihat pada Tabel 3.5. Pada *client mobile* terlihat  
 4 bahwa teknologi PHP lebih banyak digunakan.

Tabel 3.5: Sepuluh teknologi dengan jumlah penggunaan paling banyak untuk *client mobile*

technology	Jumlah Pemakaian
jQuery	286.441.430
Open Graph	245.369.426
PHP	214.645.943
Google Analytics	212.471.833
Google Font API	176.342.896
core-js	158.670.473
MySQL	147.561.365
RSS	144.642.248
WordPress	139.451.571
jQuery Migrate	128.788.203

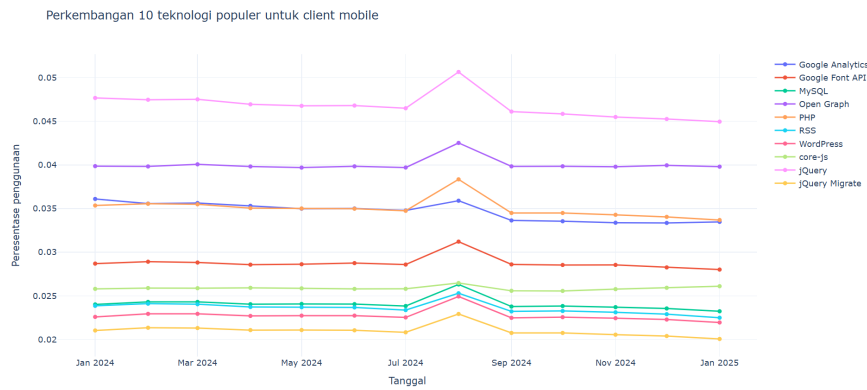
5 Setelah menemukan sepuluh teknologi yang paling banyak digunakan pada *client mobile* hal  
 6 yang selanjutnya dilakukan adalah melakukan visualisasi. Visualisasi ini dilakukan dengan cara  
 7 yang sama seperti pada Bagian 3.2.2, yaitu memvisualisasikan data berdasarkan jumlah penggunaan,  
 8 lalu memvisualisasikan data berdasarkan persentase penggunaan nya. Visualisasi dilakukan dengan  
 9 menjalankan Kode 3.2. Kode tersebut membuat visualisasi dari data yang telah dibuat menjadi  
 10 pivot tabel. Kode tersebut juga membuat visualisasi ke dalam bentuk HTML.

Kode 3.2: Kode untuk membuat visualisasi pada *client mobile*

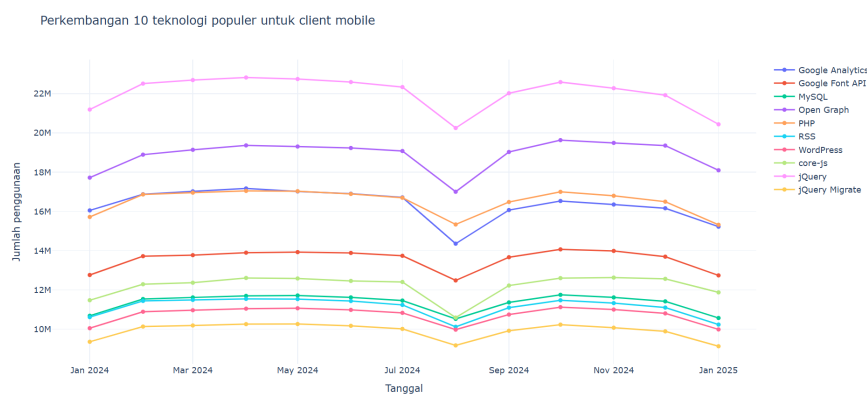
```

11
12 1
13 2     fig3 = go.Figure()
14 3     for col in df_pivot_mobile.columns:
15 4         fig3.add_trace(go.Scatter(x=df_pivot_mobile.index, y=df_pivot_mobile[col], mode='lines+markers', name=col))
16 5
17 6     fig3.update_layout(
18 7         title="Perkembangan_10_teknologi_populer_pada_client_mobile",
19 8         xaxis_title="Tanggal",
20 9         yaxis_title="Persentase_penggunaan",
21 0         template="plotly-white"
22 1     )
23 2     fig3.show()
24 3     fig3.write_html('sample_mobile.html')
```

26 Hasil visualisasi untuk data yang menggunakan jumlah penggunaan teknologi dapat dilihat pada  
 27 Gambar 3.3. Hasil visualisasinya memperlihatkan hal yang hampir sama seperti pada Bagian 3.2.2.  
 28 Semua teknologi mengalami penurunan penggunaan pada 1 Agustus 2024 dan penggunaan semua  
 29 teknologi terlihat stabil.



Gambar 3.4: Perkembangan persentase penggunaan sepuluh teknologi populer pada *client mobile*

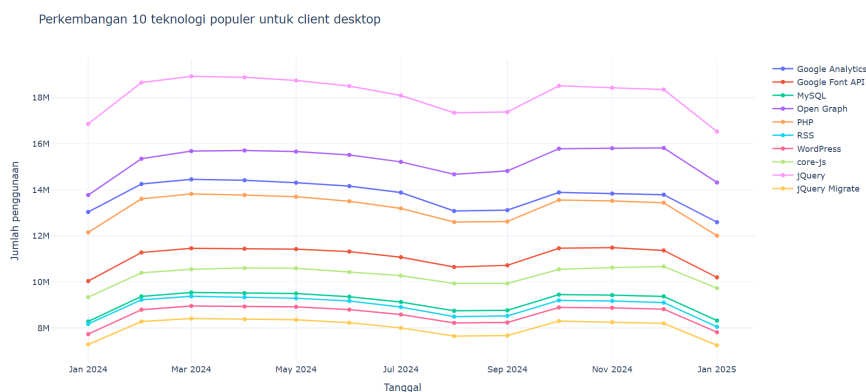


Gambar 3.3: Perkembangan sepuluh teknologi populer pada *client mobile*

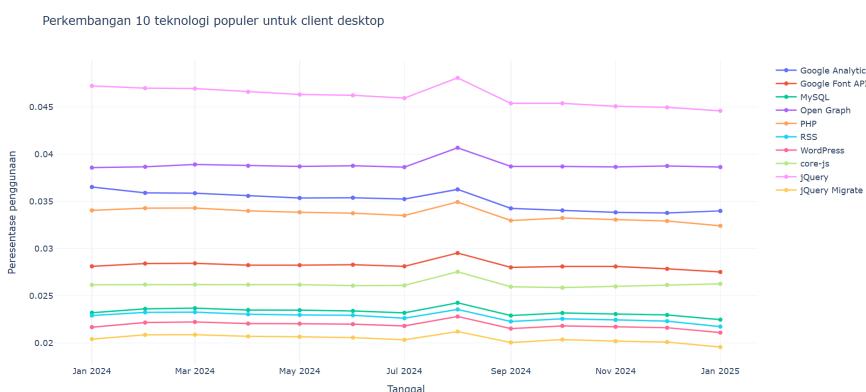
- 1 Hasil visualisasi untuk data berdasarkan persentase penggunaan dapat dilihat pada Gambar 3.4.
- 2 Hasilnya menunjukkan untuk semua teknologi adanya kenaikan persentase penggunaan pada bulan
- 3 Agustus 2024. Hal ini dikarenakan total jumlah penggunaan teknologi pembuatan *web* di bulan
- 4 tersebut merupakan yang paling kecil.

#### 5 Perkembangan Sepuluh Teknologi pada *Client desktop*

- 6 Metode yang dilakukan untuk melihat perkembangan sepuluh teknologi yang paling banyak digunak-
- 7 an untuk *client desktop* sama seperti pada Bagian 3.2.2 yaitu memisahkan data yang memiliki *client*
- 8 *desktop* terlebih dahulu kemudian melakukan pengolahan sehingga mendapatkan hasil visualisasi
- 9 yang diinginkan. Visualisasi yang dilakukan didasarkan pada dua hal yaitu jumlah penggunaan
- 10 teknologi dan persentase penggunaan teknologi. Hasil visualisasi berdasarkan jumlah penggunaan
- 11 nya dapat dilihat pada Gambar 3.5. Pada *client desktop* penggunaan teknologi PHP tidak sebanyak
- 12 pada perangkat *mobile*. Kemudian pada tanggal 1 Agustus 2024 juga semua teknologi mengalami
- 13 penurunan penggunaan. Namun untuk *client desktop* penggunaan semua teknologi tidak langsung
- 14 mengalami kenaikan di bulan september namun baru meningkat kembali di bulan Oktober.

Gambar 3.5: Perkembangan sepuluh teknologi yang populer untuk *client desktop*

- 1 Hasil visualisasi untuk persentase penggunaan pada *client desktop* dapat dilihat pada Gambar 3.6.  
 2 Hasil nya menunjukkan bahwa adanya kenaikan persentase penggunaan teknologi populer pada bulan  
 3 Agustus 2024. Hal ini dikarenakan jumlah penggunaan total pada bulan Agustus nilai nya paling  
 4 rendah dibandingkan dengan bulan yang lain sehingga persentase untuk semua teknologi akan naik.

Gambar 3.6: Perkembangan persentase penggunaan sepuluh teknologi populer pada *client desktop*

## 5 3.3 Eksplorasi *Library Python*

### 6 3.3.1 *Tkinter*

7 *Library* ini digunakan untuk membuat tampilan antarmuka grafis yang menggunakan Tcl/Tk sebagai  
 8 dasarnya seperti yang sudah dijelaskan dalam 2.4. Ada beberapa hal yang dapat dilakukan dengan  
 9 menggunakan library ini seperti contohnya membuat tampilan grafis untuk menampilkan grafik  
 10 perkembangan teknologi pembuatan *web*.

11 Tampilan yang dapat dibuat tentunya harus memiliki fitur yang dapat digunakan oleh pengguna  
 12 nantinya seperti pengguna dapat memilih teknologi yang ingin dilihat perkembangannya, pengguna  
 13 dapat memilih rentang waktu untuk melihat perkembangan teknologinya, dan pengguna juga dapat  
 14 melihat perkembangan teknologi berdasarkan jumlah penggunaannya atau persentase pengguna-  
 15 annya. Hal ini dapat dicapai dengan beberapa cara. Untuk pemilihan teknologi, pengguna dapat  
 16 memilih satu atau lebih teknologi yang ingin dilihat.

17 *Tkinter* memiliki beberapa cara untuk memenuhi kebutuhan-kebutuhan yang sudah disebutkan,

- 1 misalnya untuk memilih teknologi yang ingin dilihat perkembangannya dapat digunakan *checkbox*.
- 2 Kode 3.3 adalah kode yang digunakan agar dapat menampilkan *checkbox* yang berisi teknologi yang
- 3 ingin dilihat perkembangannya.

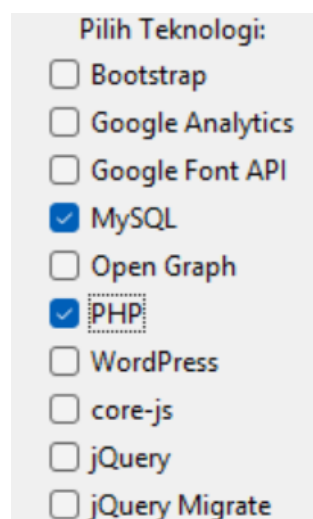
Kode 3.3: Kode untuk menampilkan *checkbox* berisi teknologi

```

4
51
62 self.checkbox_frame = ttk.Frame(root)
73 self.checkbox_frame.pack()
84
95 self.tech_vars = {}
106 for tech in df.columns:
117     var = tk.BooleanVar(value=False)
128     cb = ttk.Checkbutton(self.checkbox_frame, text=tech, variable=var)
139     cb.pack(anchor='w')
140     self.tech_vars[tech] = var

```

- 16 Kode `self.checkbox_frame = ttk.Frame(root)` adalah kode untuk membuat *frame* yang
- 17 akan menjadi wadah bagi sekumpulan *checkbox* tersebut. Parameternya diisi `root` karena *frame*
- 18 ini merupakan *child widget* dari *widget root*, seperti yang sudah dijelaskan pada Bagian 2.4. Baris
- 19 kode `self.checkbox_frame.pack()` berguna untuk meletakkan *widget frame* di dalam jendela
- 20 tampilan secara otomatis, dalam kata lain baris kode ini berperan sebagai *geometry manager*. Baris
- 21 selanjutnya adalah pengulangan yang dilakukan untuk memasukkan teknologi ke dalam *checkbox*.
- 22 Hasil dari menjalankan Kode 3.3 dapat dilihat pada Gambar 3.7

Gambar 3.7: *Checkbox* yang berisi teknologi yang dapat dipilih untuk dilihat perkembangannya

- 23 Selain memilih teknologi yang ingin dilihat, pengguna juga dapat memilih rentang waktu yang
- 24 ingin dilihat perkembangannya. Salah satu yang dapat digunakan adalah dengan menggunakan *text*
- 25 *box*. Kode ?? dapat dijalankan untuk memenuhi kebutuhan ini.

Kode 3.4: Kode untuk menampilkan *textbox* berisi tanggal

```

26
271
282 ttk.Label(frame, text="Dari_YYYY-MM:").grid(row=0, column=0)
293 self.start_var = tk.StringVar(value=df.index[0])
304 self.start_entry = ttk.Entry(frame, textvariable=self.start_var, width=10)
315 self.start_entry.grid(row=0, column=1)

```

- 33 Baris kode `self.start_var = tk.StringVar(value=df.index[0])` berguna untuk mengisi *te-*
- 34 *xtbox* dengan tanggal pertama yang ada di dalam data yang dimiliki yang kemudian diubah ke dalam



1 bentuk *string*. Baris kode `self.start_entry = ttk.Entry(frame, textvariable=self.start_var, width=`  
 2 `berguna untuk membuat textbox untuk memilih tanggalnya. Parameter frame digunakan karena te-`  
 3 `xtbox ini adalah child widget dari widget frame. Baris kode self.start_entry.grid(row=0, column=1)  
 4 berperan sebagai geometry manager yang akan menempatkan textbox pada baris 0 dan kolom 1  
 5 pada widget frame. Hasil yang didapat setelah menjalankan kode 3.4 dapat dilihat pada Gambar 3.8`

Dari (YYYY-MM): 2019-01-01

Gambar 3.8: *Textbox* untuk memilih tanggal mulai rentang waktu untuk melihat perkembangan teknologi

6 Kemudian pengguna juga dapat memilih perkembangan yang ingin dilihat berdasarkan jumlah  
 7 penggunaan atau persentase penggunaan teknologi. Untuk memenuhi hal tersebut maka salah satu  
 8 caranya adalah dengan menggunakan *radio button*. Kode 3.5 adalah kode yang dapat digunakan  
 9 untuk membuat *radio button* ini.

Kode 3.5: Kode untuk menampilkan *radio button* untuk memilih grafik yang dilihat

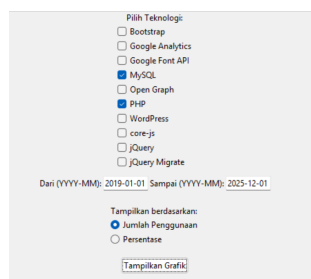
```
10
11 1
12 2 self.mode_var = tk.StringVar(value="jumlah")
13 3 rb1 = ttk.Radiobutton(mode_frame, text="Jumlah Penggunaan", variable=self.mode_var, value="jumlah")
14 4 rb2 = ttk.Radiobutton(mode_frame, text="Persentase", variable=self.mode_var, value="persentase")
15 5 rb1.pack(anchor='w')
16 6 rb2.pack(anchor='w')
```

18 Baris kode `self.mode_var = tk.StringVar(value="jumlah")` merupakan variabel yang me-  
 19 nyimpan mode yang akan dipakai. Kode `rb1 = ttk.Radiobutton(mode_frame, text="Jumlah Penggunaan",`  
 20 `berguna untuk membuat radio button yang membuat grafik yang dilihat merupakan grafik berda-`  
 21 `sarkan jumlah penggunaan. Kode rb1.pack(anchor='w') berguna untuk menempatkan radio`  
 22 `button pada jendela tampilan secara otomatis. Hasil dari kode 3.5 dapat dilihat pada Gambar 3.9`

Tampilkan berdasarkan:  
☒ Jumlah Penggunaan  
☐ Persentase

Gambar 3.9: *Radio button* untuk memilih mode grafik yang ingin ditampilkan

23 Hasil secara utuh dapat dilihat pada Gambar 3.10. Terlihat adanya tombol “Tampilkan Grafik”  
 24 yang akan mengarahkan pengguna ke *browser* untuk melihat grafik perkembangan teknologi yang  
 25 sudah dipilih.



Gambar 3.10: Tampilan *GUI* untuk melihat perkembangan teknologi pembuatan *web*



## BAB 4

### PENAMBANGAN DATA

Bab ini akan membahas deskripsi data, penyiapan data, dan eksplorasi data dengan menggunakan data *real*

#### 4.1 Deskripsi data

Data yang digunakan adalah data yang memiliki format *Comma Separated Value* dengan nama file “data\_real.csv”. Data ini didapatkan dengan menjalankan kueri pada Kode 4.1 di *platform* bernama *Google BigQuery*. Kueri yang ada akan mengembalikan tabel yang berisi empat kolom dan memiliki 330.781 baris. Data ini diambil dalam rentang waktu 1 Januari 2024 sampai 31 Desember 2024. Sebagai gambaran Tabel 4.1 merupakan sepuluh baris pertama dari data yang akan digunakan.

Kode 4.1: Kode untuk mengumpulkan data *real*

```
11 SELECT date, client, t.technology, count(t.technology) as jumlah
12 FROM 'httparchive.crawl.pages', UNNEST(technologies) as t
13 WHERE date BETWEEN "2019-01-01" and "2024-12-31"
14 GROUP BY t.technology, date, client
```

Tabel 4.1: Sepuluh Baris pertama dari data *real*

date	client	technology	jumlah
2024-04-01	desktop	gunicorn	12.370
2024-04-01	desktop	Iterable	131
2024-04-01	desktop	AntV G2	946
2024-04-01	desktop	Rocketfy	1
2024-11-01	desktop	Platform.sh	16.634
2024-11-01	desktop	Brimble	2
2024-10-01	desktop	Tawk.to	202.669
2024-10-01	desktop	One.com	92.276
2024-10-01	desktop	Bootstrap Table	8.180
2024-10-01	desktop	Saleor	8

Kolom-kolom tersebut adalah sebagai berikut:

1. *date* yang memiliki tipe data *date*. Kolom ini berisi waktu pengetesan dilakukan.
2. *client* yang memiliki tipe data *string*. Kolom ini berisi perangkat yang digunakan untuk melakukan tes kepada halaman *web*. Kolom ini memiliki dua *value* yaitu *desktop* dan *mobile*.

3. *technology* yang memiliki tipe data *string*. Kolom ini berisi nama-nama teknologi yang digunakan oleh halaman *web* dalam pembuatan atau pengembangan halaman *web* tersebut. Kolom ini merupakan elemen dari *array technologies*.
4. *jumlah* yang memiliki tipe data *integer*. Kolom ini berisi jumlah penggunaan setiap teknologi.

## 4.2 Penyiapan data

Data yang telah didapatkan kemudian disiapkan agar lebih mudah dalam melakukan analisis. Salah satu cara yang dapat dilakukan adalah dengan melakukan pembersihan data. Pada data ini ditemukan beberapa baris yang mengalami kesalahan saat proses penarikan data. Sehingga data yang dihasilkan menjadi tidak jelas. Contoh dari masalah ini dapat dilihat pada Tabel 4.2. Untuk mengatasi masalah ini maka baris yang mengalami kesalahan tadi akan dihapus dari data. Hal ini dilakukan karena jumlah data yang mengalami kesalahan tidak mencapai 10 % dari total seluruh data.

Tabel 4.2: Contoh kesalahan saat penarikan data.

date	client	technology	jumlah
2022-09-01	desktop	2022	2
2019-05-01	desktop	255);	1
2019-11-01	mobile	255); border-style: none; margin: 0px; border-radius: 0px; padding: 0px;	1
2020-12-01	mobile	b) => (a.length > b.length ? a : b)1.7.2	1
2020-11-01	mobile	yeah	1

Setelah data dibersihkan, kemudian data diubah ke dalam pivot tabel untuk lebih mudah dalam melakukan eksplorasi. Hal ini bertujuan agar jumlah penggunaan setiap teknologi untuk setiap bulan dapat dengan mudah didapatkan. Tabel 4.3 adalah tabel pivot yang telah dibuat dan menunjukkan jumlah penggunaan untuk masing-masing teknologi setiap bulannya. Selain jumlah penggunaan setiap teknologi untuk setiap bulannya tabel ini juga memiliki kolom jumlah yang merupakan jumlah penggunaan semua teknologi untuk setiap bulannya.

Tabel 4.3: Tabel pivot untuk mengetahui jumlah penggunaan teknologi setiap bulan

date	Bootstrap	Google Analytics	Google Font API	...
2019-01-01	10.333.295	27.675.700	15.906.025	...
2019-02-01	9.989.325	26.912.725	15.522.950	...
2019-03-01	9.960.270	26.805.405	15.518.165	...
2019-04-01	11.872.205	31.344.745	18.993.060	...
2019-05-01	12.341.080	32.557.230	19.735.740	...
2019-06-01	12.368.135	32.596.160	19.771.645	...
2019-07-01	13.182.590	34.160.625	21.019.160	...
2019-08-01	13.146.940	34.057.865	20.922.670	...
2019-09-01	13.116.225	33.929.805	20.620.630	...
2019-10-01	13.081.440	33.779.350	20.555.285	...
...	...	...	...	...

Setelah data disiapkan hal yang dapat dilakukan selanjutnya adalah melakukan eksplorasi data seperti melihat perkembangan sepuluh teknologi yang paling banyak digunakan untuk semua

- 1 perangkat, perkembangan sepuluh teknologi yang paling banyak digunakan untuk perangkat *mobile*
- 2 dan *desktop*, dan melihat perkembangan beberapa teknologi yang nampaknya populer.

### 3 4.3 Eksplorasi Data

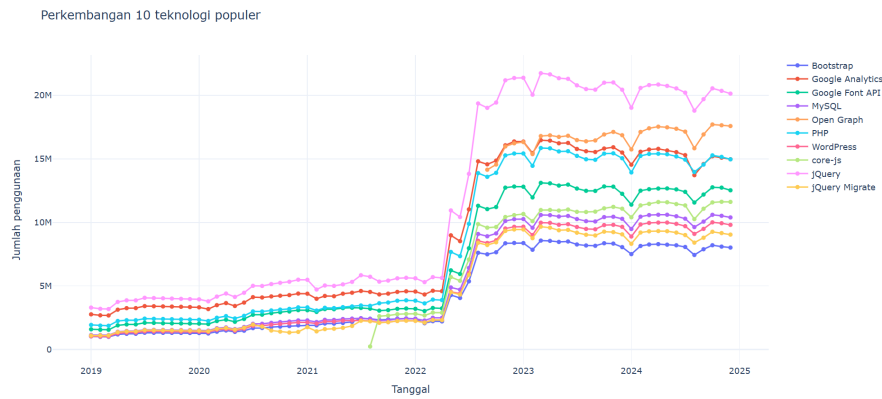
#### 4 4.3.1 Perkembangan sepuluh teknologi yang paling banyak digunakan di semua 5 perangkat.

6 Hal pertama yang akan dilakukan adalah melihat perkembangan sepuluh teknologi yang paling  
7 banyak digunakan untuk membangun atau mengembangkan halaman *web*. Proses ini dilakukan  
8 dengan cara mengelompokkan data berdasarkan teknologinya dan menghitung jumlah penggunaan  
9 untuk setiap teknologi. Data yang sudah dikelompokkan kemudian diurutkan mulai dari teknologi  
10 yang jumlah penggunaannya paling banyak. Setelah diurutkan, kemudian sepuluh teknologi yang  
11 memiliki jumlah penggunaan paling banyak akan dianalisis lebih lanjut. Sepuluh teknologi tersebut  
12 dapat dilihat pada Tabel 4.4. Terlihat bahwa teknologi yang paling banyak digunakan untuk  
13 mengembangkan atau membangun halaman *web* adalah *jQuery*.

Tabel 4.4: Sepuluh teknologi dengan jumlah penggunaan paling banyak

technology	jumlah
jQuery	1.635.240.294
Google Analytics	1.266.635.371
PHP	1.155.199.795
Google Font API	969.323.383
Open Graph	931.262.211
MySQL	776.720.413
WordPress	728.930.962
core-js	713.142.531
jQuery Migrate	691.494.400
Bootstrap	636.813.294

14 Kemudian sepuluh teknologi yang paling banyak digunakan ini divisualisasikan perkembangannya.  
15 Agar dapat divisualisasikan dengan mudah, maka pekerjaan ini diselesaikan dengan membuat tabel  
16 pivot. Tabel pivot seperti pada Tabel 4.3 berguna agar dapat mengetahui jumlah penggunaan setiap  
17 teknologi dalam setiap bulan selama lima tahun. Hasil visualisasi terhadap sepuluh teknologi yang  
18 paling banyak digunakan dapat dilihat pada Gambar 4.1.



Gambar 4.1: Perkembangan sepuluh teknologi yang paling banyak digunakan membangun *web*

Visualisasi yang dihasilkan menunjukkan adanya peningkatan jumlah penggunaan untuk semua teknologi. Peningkatan paling besar terjadi pada bulan Mei 2022 hingga Agustus 2022. Hal ini kemungkinan terjadi karena adanya peningkatan jumlah *web* yang diuji oleh halaman *HTTP Archive*. Hal ini terlihat dari jumlah penggunaan semua teknologi yang melonjak secara signifikan di antara Mei 2022 sampai Agustus 2022 seperti yang terlihat pada tabel 4.5 terlihat adanya perbedaan 90.000.000 antara bulan April 2022 dan Mei 2022. Hal ini yang menyebabkan semua teknologi mengalami peningkatan pada bulan tersebut.

Tabel 4.5: Perubahan jumlah sample

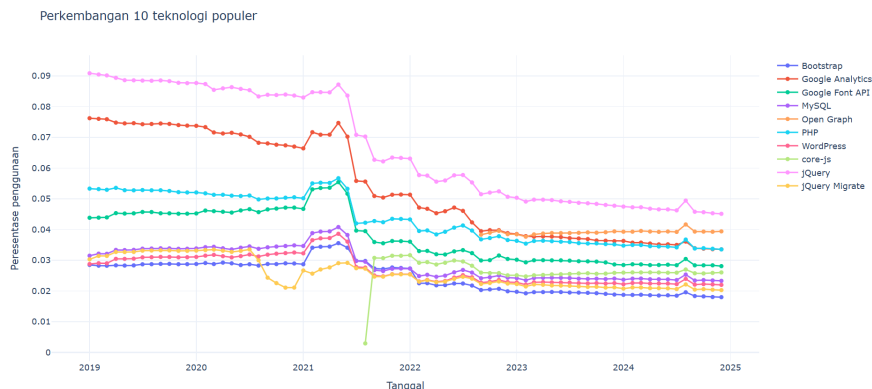
date	jumlah
2022-01-01	88.849.500
2022-02-01	91.860.601
2022-03-01	98.950.365
2022-04-01	101.368.693
2022-05-01	195.567.380
2022-06-01	180.854.521
2022-07-01	239.389.984
2022-08-01	349.980.357

Terlihat bahwa *core-js* adalah teknologi yang baru masuk sebagai teknologi yang paling banyak digunakan mulai dari Agustus 2021 namun langsung menjadi teknologi terbanyak dipakai urutan ke-6 di bulan selanjutnya. Selain *core-js* ada juga teknologi *Open Graph* yang baru muncul di bulan September 2022 dan langsung menempati urutan ketiga sebagai teknologi yang paling banyak digunakan dalam membangun dan mengembangkan *web* yang kemudian menyalip *Google Analytics* pada bulan Maret 2023.

Hasil yang didapatkan dinilai kurang proporsional, maka visualisasi diubah menjadi visualisasi berdasarkan persentase penggunaan teknologi bukan lagi berdasarkan jumlah penggunaan. Hal ini dilakukan karena total jumlah penggunaan untuk semua teknologi setiap bulannya tidak sama sehingga agar lebih proporsional. Hasil visualisasi berdasarkan persentase penggunaan ini dapat dilihat pada Gambar 4.2.

Terlihat adanya penurunan persentase penggunaan untuk semua teknologi yang paling banyak digunakan. Penurunan ini kemungkinan terjadi karena total penggunaan semua teknologi semakin meningkat. Penurunan paling rendah terjadi untuk teknologi *jQuery Migrate* pada bulan November

- 1 2020. yang hanya dipakai sebanyak 0,02017%. Penurunan pada setiap teknologi juga terpengaruh
- 2 dari jumlah penggunaannya yang semakin tinggi setiap bulannya.



Gambar 4.2: Perkembangan persentase penggunaan dari sepuluh teknologi yang paling banyak digunakan

### 4.3.2 Perkembangan sepuluh teknologi yang paling banyak digunakan di perangkat *desktop* dan *mobile*.

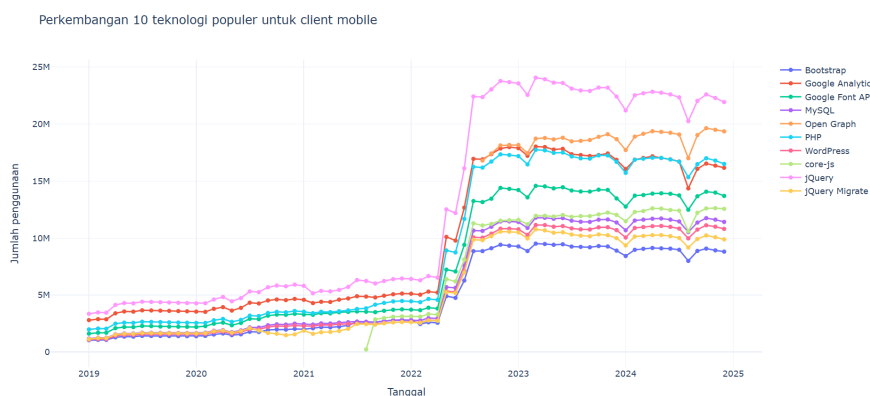
Data yang didapatkan merupakan data yang berasal dari dua perangkat yang berbeda yaitu *mobile* dan *desktop*. Pada bagian ini yang akan dilihat adalah perkembangan sepuluh teknologi yang paling banyak digunakan pada masing-masing perangkat. Perangkat pertama yang akan ditinjau adalah perangkat *mobile*. Metode yang digunakan masih sama yaitu mencari terlebih dahulu sepuluh teknologi yang paling banyak digunakan pada perangkat *mobile*. Setelah dikelompokkan dan diurutkan, sepuluh teknologi yang paling banyak digunakan pada perangkat *mobile* dapat dilihat pada Tabel 4.6. Terlihat bahwa sepuluh teknologi yang populer di perangkat *mobile* masih sama dengan yang terdapat pada Tabel 4.4.

Tabel 4.6: Sepuluh teknologi dengan jumlah penggunaan paling banyak pada perangkat *mobile*

technology	jumlah
jQuery	907.815.523
Google Analytics	694.557.335
PHP	649.411.801
Google Font API	541.528.239
Open Graph	520.578.213
MySQL	435.952.366
WordPress	410.439.160
core-js	392.176.760
jQuery Migrate	388.203.382
Bootstrap	355.050.603

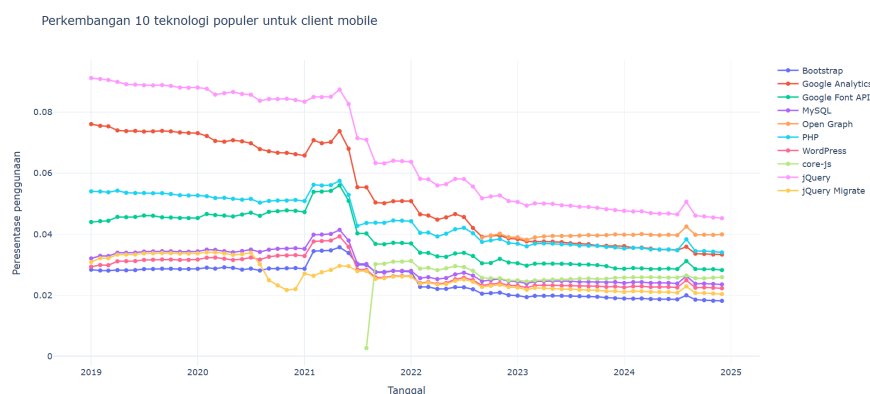
Setelah sepuluh teknologi yang paling banyak digunakan pada perangkat *mobile* didapatkan, kemudian divisualisasikan dengan cara yang sama seperti pada bagian 4.3.1. Visualisasi pertama dilakukan berdasarkan jumlah penggunaan teknologi. Kemudian dilanjutkan dengan visualisasi berdasarkan persentase penggunaannya. Untuk visualisasi berdasarkan jumlah penggunaannya, hasilnya dapat dilihat pada Gambar 4.3. Terlihat bahwa perkembangannya semakin meningkat

- 1 untuk semua teknologi seperti yang sudah dijelaskan pada 4.3.1 bahwa adanya peningkatan jumlah
- 2 *web* yang diuji, namun untuk teknologi *jQuery* jumlah penggunaan pada perangkat *mobile* memiliki
- 3 rentang yang lebih tinggi.



Gambar 4.3: Perkembangan sepuluh teknologi paling banyak digunakan pada perangkat *mobile*

- 4 Setelah melihat perkembangan jumlah penggunaan, hal yang selanjutnya akan ditinjau adalah
- 5 persentase penggunaan teknologi pada perangkat *mobile*. Hasil visualisasi untuk persentase penggu-
- 6 naan ini dapat dilihat pada Gambar 4.4 Terlihat bahwa visualisasi yang dihasilkan tidak berbeda
- 7 jauh dengan Gambar 4.2. Yang berbeda adalah rentang persentasenya lebih rendah.



Gambar 4.4: Perkembangan persentase penggunaan teknologi pada perangkat *mobile*

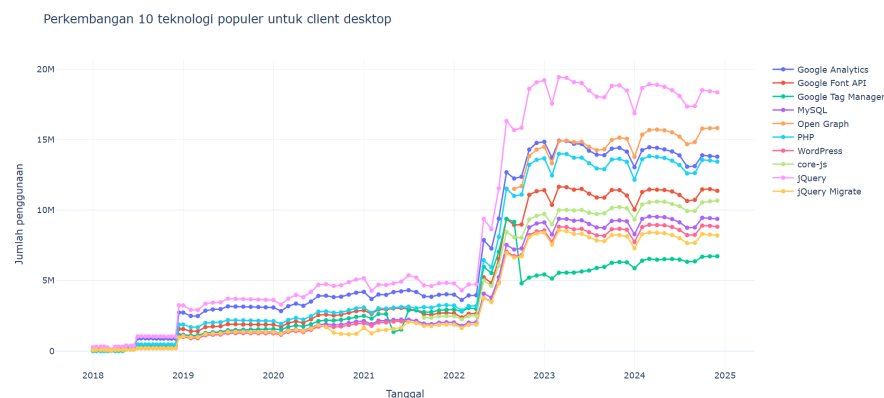
- 8 Perangkat selanjutnya yang akan dianalisis adalah perangkat *desktop*. Metode yang digunakan
- 9 pada analisis ini masih sama yaitu dengan mencari sepuluh teknologi yang paling banyak digunak-
- 10 an. Setelah itu melakukan visualisasi berdasarkan jumlah penggunaan teknologi dan persentase
- 11 penggunaan teknologi tersebut. Sepuluh teknologi yang paling banyak digunakan pada perangkat
- 12 *desktop* dapat dilihat pada Tabel 4.7. Terlihat adanya teknologi baru yang masuk yaitu *Google Tag*
- 13 *Manager*. Terlihat juga dari jumlah penggunaan masing-masing teknologi lebih rendah daripada
- 14 jumlah penggunaan teknologi pada perangkat *mobile*.



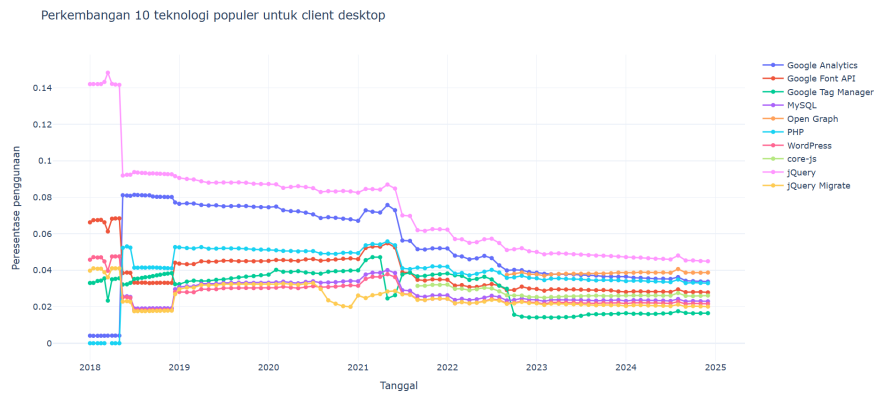
Tabel 4.7: Sepuluh teknologi yang paling banyak digunakan pada perangkat *desktop*

technology	jumlah
jQuery	745.620.826
Google Analytics	585.708.398
PHP	513.344.386
Google Font API	435.039.011
Open Graph	410.683.998
MySQL	344.462.488
WordPress	322.822.568
core-js	320.965.771
jQuery Migrate	307.386.712
Google Tag Manager	288.083.449

Hasil visualisasi berdasarkan jumlah penggunaan dari sepuluh teknologi yang paling banyak digunakan pada perangkat *desktop* dapat dilihat pada Gambar 4.5. Terlihat bahwa pada perangkat *desktop* juga terdapat peningkatan pada semua teknologi. Namun untuk teknologi *Google Tag Manager* perkembangannya sangat tidak stabil. Hal ini terlihat dari jumlah penggunaan yang naik dan turun secara signifikan. Pada perangkat *desktop* juga teknologi *Open Graph* baru mulai digunakan pada bulan September 2022.

Gambar 4.5: Perkembangan sepuluh teknologi paling banyak diggunakan pada perangkat *desktop*

Selain dari jumlah penggunaannya, visualisasi juga dilakukan berdasarkan persentase penggunaan dari sepuluh teknologi yang paling banyak digunakan. Hasil visualisasi berdasarkan persentase penggunaan ini dapat dilihat pada Gambar 4.6. Terlihat adanya perbedaan yang sangat signifikan di antara semua teknologi, namun kembali stabil pada Mei 2018. Persentase penggunaan untuk semua teknologi mengalami penurunan. Teknologi yang paling tidak stabil perkembangannya adalah *Google Tag Manager*. Visualisasi ini juga menunjukkan untuk teknologi *jQuery* persentase penggunaan pada bulan Januari 2018 – Mei 2018 merupakan persentase penggunaan paling tinggi kemudian untuk teknologi *Google Analytic* pada rentang waktu yang sama merupakan persentase penggunaan yang paling rendah.



Gambar 4.6: Perkembangan persentase penggunaan teknologi pada perangkat *desktop*

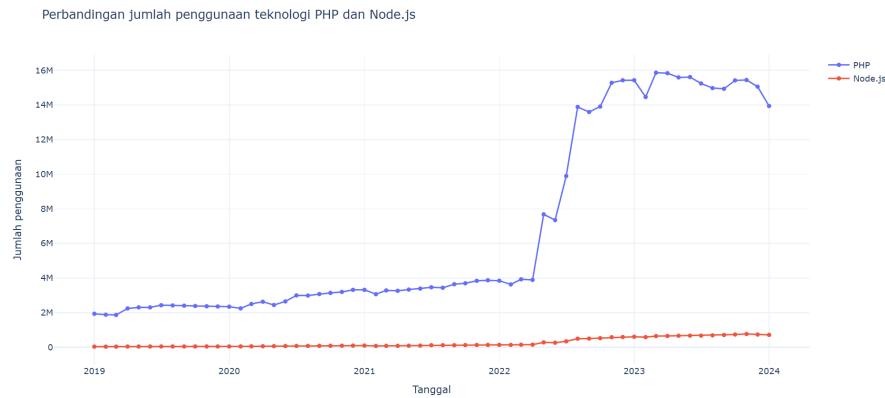
### 4.3.3 Perbandingan teknologi yang nampak populer

Bagian ini akan membandingkan beberapa teknologi yang paling banyak digunakan dengan beberapa teknologi yang nampak populer dalam semua perangkat. Teknologi-teknologi ini diambil dari daftar teknologi pengembangan *web* yang populer dari halaman <https://www.geeksforgeeks.org/top-web-development-trends/#most-popular-web-development-technologies>. Perbandingan akan dilakukan terhadap teknologi yang dipakai pada *back-end* maupun *front-end* dalam pengembangan *web*.

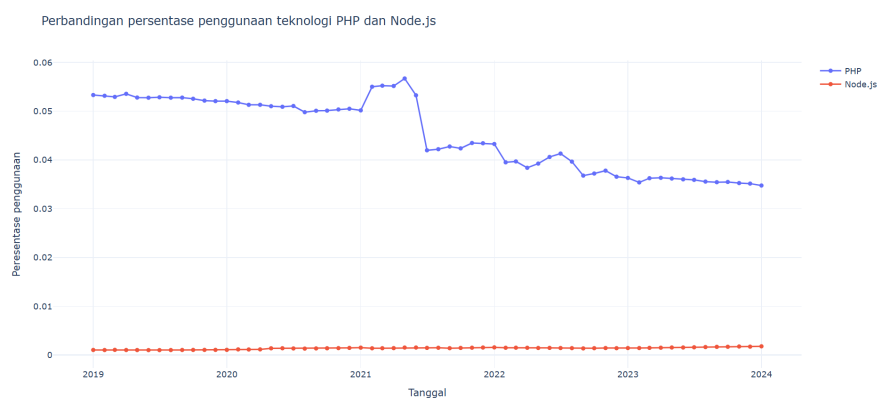
#### PHP dengan Node.js

PHP dan Node.js merupakan dua teknologi yang dipakai pada bagian *back-end* dalam pengembangan *web*. Node.js merupakan teknologi yang cukup populer. Namun dari data yang didapatkan seperti yang terlihat pada Gambar 4.7 teknologi Node.js tidak digunakan sebanyak teknologi PHP, sehingga Node.js tidak masuk ke dalam daftar teknologi yang paling banyak digunakan. Perbedaan jumlah penggunaan juga sangat besar terlihat dari penggunaan PHP yang berkisar di antara 20 juta penggunaan sedangkan Node.js hanya berkisar di antara 600.000 penggunaan.

Sementara itu dari segi persentase penggunaan menunjukkan hal yang sama. Terlihat pada Gambar 4.8 bahwa persentase penggunaan dari teknologi PHP lebih tinggi jika dibandingkan dengan teknologi Node.js. Terlihat bahwa penggunaan teknologi PHP berkisar antara 0.035-0.055% sedangkan teknologi Node.js penggunaannya tidak lebih dari 0.01%. Kemungkinan penyebab teknologi Node.js kurang banyak diminati oleh pengembang *web* adalah karena JavaScript menyediakan banyak varian teknologi yang dapat dipilih oleh pengembang. Sehingga jumlah penggunaannya terbagi ke teknologi dari JavaScript yang lain.



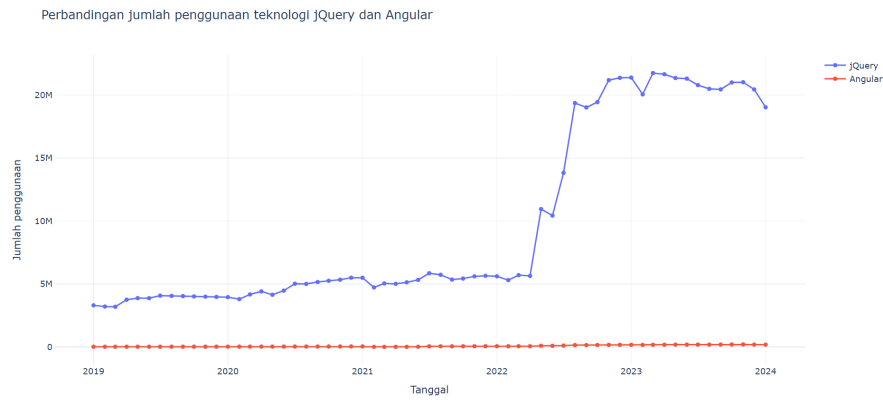
Gambar 4.7: Perbandingan jumlah penggunaan teknologi *PHP* dan *Node.js*



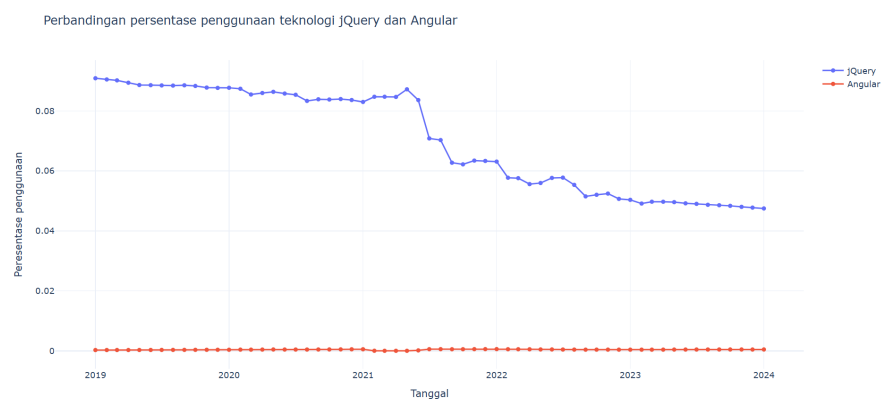
Gambar 4.8: Perbandingan persentase penggunaan teknologi *PHP* dan *Node.js*

#### 1 *jQuery* dengan *Angular*

2 *jQuery* dan *Angular* merupakan teknologi yang dipakai pada bagian *front-end* dalam pengembangan  
3 *web*. *Angular* merupakan teknologi yang cukup populer, tetapi berdasarkan data yang didapatkan,  
4 *Angular* memiliki jumlah penggunaan yang sangat kecil seperti yang dapat dilihat pada Gam-  
5 bar 4.9. Terlihat bahwa jumlah halaman *web* yang menggunakan teknologi *Angular* tidak lebih dari  
6 200.000. Hal ini juga berpengaruh terhadap persentase penggunaan seperti yang dapat dilihat pada  
7 Gambar 4.10. Teknologi *Angular* tidak memiliki persentase yang lebih dari 0.01%. Kemungkinan  
8 penyebab rendahnya jumlah penggunaan teknologi *Angular* adalah karena pengembang tidak terlalu  
9 tertarik untuk menggunakan *Angular*.



Gambar 4.9: Perbandingan jumlah penggunaan teknologi *jQuery* dan *Angular*



Gambar 4.10: Perbandingan persentase penggunaan teknologi *jQuery* dan *Angular*

## DAFTAR REFERENSI

- [1] Viscomia, R., Calvano, P., Chen, S., Hobbs, M., dan Demir, N. (2024) HTTP Archive about. Technical report. HTTP Archive.
- [2] Version 3.13.3 (2025) *tkinter — Python interface to Tcl/Tk*. Python Software Foundation.
- [3] Nugroho, P. A., Putra, R. C., Maulana, R. C., dan Tandra, V. (2024) Usage of unsupported technologies in websites worldwide. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, **10**, 332–344.
- [4] Version 3.0 (2016) *Introduction to Lighthouse*. Google for developers.
- [5] Lakshmanan, J., Valliappa; Tigani (2020) *Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale*, first edition edition. O'Reilly Media.



# LAMPIRAN A

## KODE PROGRAM

Kode A.1: Kode untuk data kecil

```
1 #-*- coding: utf-8 -*-
2 """
3 Spyder Editor
4 """
5
6 import pandas as pd
7 import plotly.graph_objects as go
8
9 df = pd.read_csv("data_sample2.csv")
10 df_pivot_baru = df.pivot_table(index='date', columns='technology', values='f0_')
11 df_gb = df.groupby(["technology"]).sum('f0_')
12 df_sorted = df_gb.sort_values(by='f0_', ascending=False)
13 df_10 = df_sorted.head(10)
14 df_baru = df.loc[df['technology'].isin(df_10.index)]
15 df_baru = df_baru[['date', 'technology', 'f0_']]
16 df_baru.drop_duplicates(subset='date')
17 df_pivot = df_baru.pivot_table(index="date", columns='technology', values='f0_')
18 fig = go.Figure()
19 for col in df_pivot.columns:
20     fig.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
21
22 fig.update_layout(
23     title="Perkembangan_10_teknologi_populer",
24     xaxis_title="Tanggal",
25     yaxis_title="Jumlah_penggunaan",
26     template="plotly_white"
27 )
28 fig.show()
29 fig.write_html("sample_jumlah.html")
30 df_pivot_baru['jumlah'] = df_pivot_baru.sum(axis=1)
31 df_pivot['jumlah'] = df_pivot_baru['jumlah']
32 df_pivot_baru.to_excel("10_teknologi_populer.xlsx")
33 for i in df_pivot.columns:
34     df_pivot[i] = df_pivot[i]/df_pivot['jumlah']
35 df_pivot = df_pivot.drop(columns='jumlah')
36 df_filled = df_pivot.fillna('0')
37
38 fig2 = go.Figure()
39 for col in df_pivot.columns:
40     fig2.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
41
42 fig2.update_layout(
43     title="Perkembangan_10_teknologi_populer",
44     xaxis_title="Tanggal",
45     yaxis_title="Persentase_penggunaan",
46     template="plotly_white"
47 )
48 fig2.show()
49 fig2.write_html("sample_persentase.html")
50
51 def visualisasi(client):
52     df2 = df[df['client'] == client]
53     df2_pivot = df2.pivot_table(index='date', columns='technology', values='f0_')
54     df_gb = df2.groupby(["technology"]).sum('f0_')
55     df_sorted = df_gb.sort_values(by='f0_', ascending=False)
56     df_10 = df_sorted.head(10)
57     df_baru = df2.loc[df2['technology'].isin(df_10.index)]
58     df_baru = df_baru[['date', 'technology', 'f0_']]
59     df_baru.drop_duplicates(subset='date')
60     df_pivot = df_baru.pivot_table(index="date", columns='technology', values='f0_')
61     fig = go.Figure()
62     for col in df_pivot.columns:
63         fig.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
64
65     fig.update_layout(
66         title=f"Perkembangan_10_teknologi_populer_untuk_client_{client}",
67         xaxis_title="Tanggal",
68         yaxis_title="Jumlah_penggunaan",
69         template="plotly_white"
70     )
71     fig.show()
72     fig.write_html(f"sample_jumlah_{client}.html")
73     df2_pivot['jumlah'] = df2_pivot.sum(axis=1)
74     df_pivot['jumlah'] = df2_pivot['jumlah']
75     df2_pivot.to_excel(f"10_teknologi_populer_{client}.xlsx")
```

```

76     for i in df_pivot.columns:
77         df_pivot[i] = df_pivot[i]/df_pivot['jumlah']
78     df_pivot = df_pivot.drop(columns = 'jumlah')
79
80     fig2 = go.Figure()
81     for col in df_pivot.columns:
82         fig2.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
83
84     fig2.update_layout(
85         title=f"Perkembangan_10_teknologi_populer_untuk_client_{client}",
86         xaxis_title="Tanggal",
87         yaxis_title="Persentase_penggunaan",
88         template="plotly_white"
89     )
90     fig2.show()
91     fig2.write_html(f"sample_persentase_{client}.html")
92
93
94 visualisasi("mobile")
95 visualisasi("desktop")

```

### Kode A.2: Kode untuk proses menggunakan data real

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon May 5 18:31:17 2025
4
5  @author: frans
6  """
7
8  import pandas as pd
9  import plotly.graph_objects as go
10
11 df = pd.read_csv('data_real.csv')
12 df = df[df['technology'] != "_"]
13 df = df[df['technology'] != "_0"];border-style:none;margin:0px;border-radius:0px;padding:0px;"
14 df = df[df['technology'] != "_255"]
15 df = df[df['technology'] != "_1.11.3"]
16 df = df[df['technology'] != "_2017"]
17 df = df[df['technology'] != "_2018-"]
18 df = df[~df['date'].str.contains('2018-')]
19 df_gb_date = df.groupby('date').sum('f0_')
20 df_pivot_baru = df.pivot_table(index='date', columns='technology', values='f0_')
21 df3 = df_pivot_baru
22 df_gb = df.groupby(["technology"]).sum('f0_')
23 df_sorted = df_gb.sort_values(by='f0_', ascending=False)
24 df_10 = df_sorted.head(10)
25 df_baru = df.loc[df['technology'].isin(df_10.index)]
26 df_baru = df_baru[['date', 'technology', 'f0_']]
27 df_baru.drop_duplicates(subset='date')
28 df_pivot = df_baru.pivot_table(index="date", columns='technology', values='f0_')
29 fig = go.Figure()
30 for col in df_pivot.columns:
31     fig.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
32
33 fig.update_layout(
34     title="Perkembangan_10_teknologi_populer",
35     xaxis_title="Tanggal",
36     yaxis_title="Jumlah_penggunaan",
37     template="plotly_white"
38 )
39 fig.show()
40 fig.write_html("sample_jumlah_real.html")
41
42 #persentase
43 df_pivot_baru['jumlah'] = df_pivot_baru.sum(axis=1)
44 df_pivot['jumlah'] = df_pivot_baru['jumlah']
45
46 for i in df_pivot.columns:
47     df_pivot[i] = df_pivot[i]/df_pivot['jumlah']
48
49 df_pivot = df_pivot.drop(columns = 'jumlah')
50 df_filled = df_pivot.fillna('0')
51
52 fig2 = go.Figure()
53 for col in df_pivot.columns:
54     fig2.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
55
56 fig2.update_layout(
57     title="Perkembangan_10_teknologi_populer",
58     xaxis_title="Tanggal",
59     yaxis_title="Persentase_penggunaan",
60     template="plotly_white"
61 )
62 fig2.show()
63 fig2.write_html("sample_persentase_real.html")
64
65 def visualisasi(client):
66     df2 = df[df['client'] == client]
67     df2_pivot = df2.pivot_table(index='date', columns='technology', values='f0_')
68     df_gb = df2.groupby(["technology"]).sum('f0_')
69     df_sorted = df_gb.sort_values(by='f0_', ascending=False)
70     df_10 = df_sorted.head(10)
71     df_baru = df2.loc[df2['technology'].isin(df_10.index)]
72     df_baru = df_baru[['date', 'technology', 'f0_']]
73     df_baru.drop_duplicates(subset='date')
74     df_pivot = df_baru.pivot_table(index="date", columns='technology', values='f0_')

```



```

75 fig = go.Figure()
76 for col in df_pivot.columns:
77     fig.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
78
79 fig.update_layout(
80     title=f"Perkembangan_10_teknologi_populer_untuk_client_{client}",
81     xaxis_title="Tanggal",
82     yaxis_title="Jumlah_penggunaan",
83     template="plotly_white"
84 )
85 fig.show()
86 fig.write_html(f"sample_jumlah_{client}.html")
87
88 #Persentase
89 df2_pivot['jumlah'] = df2_pivot.sum(axis=1)
90 df_pivot['jumlah'] = df2_pivot['jumlah']
91 df2_pivot.to_excel(f"10_teknologi_populer_{client}.xlsx")
92 for i in df_pivot.columns:
93     df_pivot[i] = df_pivot[i]/df_pivot['jumlah']
94 df_pivot = df_pivot.drop(columns = 'jumlah')
95
96 fig2 = go.Figure()
97 for col in df_pivot.columns:
98     fig2.add_trace(go.Scatter(x=df_pivot.index, y=df_pivot[col], mode='lines+markers', name=col))
99
100 fig2.update_layout(
101     title=f"Perkembangan_10_teknologi_populer_untuk_client_{client}",
102     xaxis_title="Tanggal",
103     yaxis_title="Persentase_penggunaan",
104     template="plotly_white"
105 )
106 fig2.show()
107 fig2.write_html(f"sample_persentase_{client}_real.html")
108
109 visualisasi('mobile')
110 visualisasi('desktop')
111
112 def perbandingan(tek1,tek2):
113
114     df_pivot_baru = df3
115     fig2 = go.Figure()
116
117     fig2.add_trace(go.Scatter(x=df_pivot_baru.index, y=df_pivot_baru[f'{tek1}'], mode='lines+markers', name=f'{tek1}'))
118     fig2.add_trace(go.Scatter(x=df_pivot_baru.index, y=df_pivot_baru[f'{tek2}'], mode='lines+markers', name=f'{tek2}'))
119
120     fig2.update_layout(
121         title=f"Perbandingan_jumlah_penggunaan_teknologi_{tek1}_dan_{tek2}",
122         xaxis_title="Tanggal",
123         yaxis_title="Jumlah_penggunaan",
124         template="plotly_white"
125     )
126     fig2.show()
127     fig2.write_html(f"sample_{tek1,tek2}.html")
128
129 #persentase
130
131 df_pivot_baru['jumlah'] = df_pivot_baru.sum(axis=1)
132 #df2_pivot.to_excel(f"10_teknologi_populer_{client}.xlsx")
133 for i in df_pivot_baru.columns:
134     df_pivot_baru[i] = df_pivot_baru[i]/df_pivot_baru['jumlah']
135
136 fig2 = go.Figure()
137
138 fig2.add_trace(go.Scatter(x=df_pivot_baru.index, y=df_pivot_baru[f'{tek1}'], mode='lines+markers', name=f'{tek1}'))
139 fig2.add_trace(go.Scatter(x=df_pivot_baru.index, y=df_pivot_baru[f'{tek2}'], mode='lines+markers', name=f'{tek2}'))
140 fig2.update_layout(
141     title=f"Perbandingan_persentase_penggunaan_teknologi_{tek1}_dan_{tek2}",
142     xaxis_title="Tanggal",
143     yaxis_title="Persentase_penggunaan",
144     template="plotly_white"
145 )
146 fig2.show()
147 fig2.write_html(f"sample_persentase_{tek1,tek2}.html")
148
149 perbandingan('jQuery', 'Angular')

```