

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
1 PENDAHULUAN	1
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
2 LANDASAN TEORI	5
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> [2]	28
2.3 Visualisasi Data	28
2.3.1 Line Plot	28
3 ANALISIS PENYELESAIAN MASALAH	31
3.1 Analisis Masalah	31
3.2 Data Kecil	31
3.2.1 Penyiapan Data Kecil	31
3.2.2 Pengolahan Data Kecil	32
DAFTAR REFERENSI	37
A KODE PROGRAM	39
B HASIL EKSPERIMEN	41

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 font-display	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>	29
3.1	Perkembangan sepuluh teknologi populer berdasarkan jumlah halaman <i>web</i>	34
3.2	Perkembangan sepuluh teknologi populer pada <i>client mobile</i>	35
3.3	Perkembangan sepuluh teknologi yang populer untuk <i>client desktop</i>	36

B.1	Hasil 1	41
B.2	Hasil 2	41
B.3	Hasil 3	41
B.4	Hasil 4	41

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*.

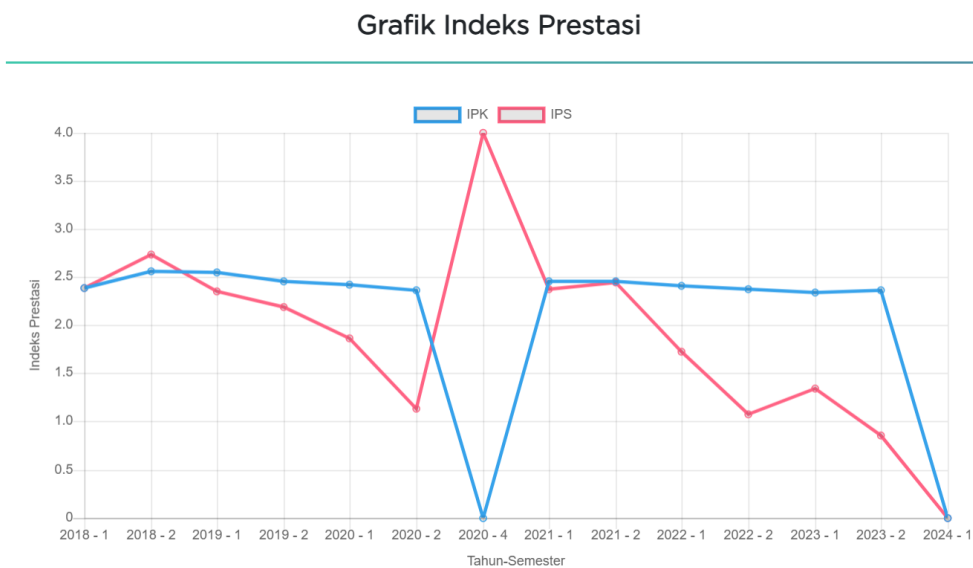
Situs *HTTP Archive* menyediakan data tentang teknologi yang digunakan dalam pembuatan *web* untuk mencatat perkembangan teknologi ini. Situs ini mengumpulkan data berdasarkan berbagai aspek, seperti pengalaman pengguna dalam mengakses *web*, kecepatan pemuatan halaman, serta tingkat aksesibilitas. Salah 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

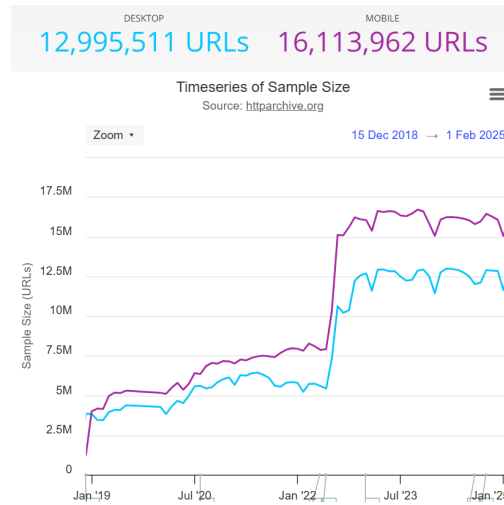
Laporan berisi informasi terperinci mengenai sumber daya yang diambil, API dan fitur platform yang digunakan, serta jejak eksekusi dari setiap halaman dari situs-situs teratas yang ada di *web*. Informasi yang telah didapatkan kemudian diolah dan dianalisis untuk melihat perkembangan tren. Laporan yang 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:

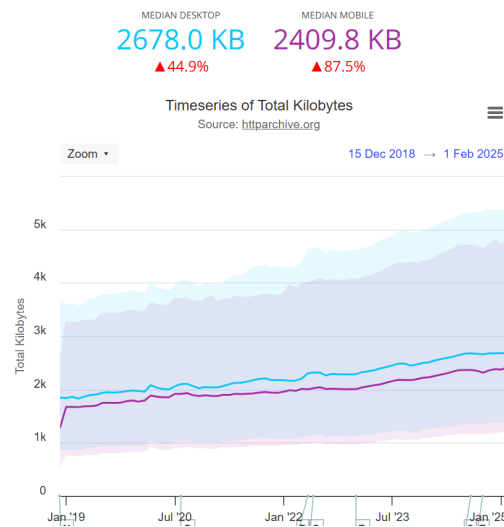
- *Sample size* yang berisi perkembangan jumlah URLs yang digunakan untuk dianalisis. Contoh visualisasi data yang dimiliki oleh laporan ini dapat dilihat pada Gambar 2.1. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025, bahwa adanya kenaikan

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



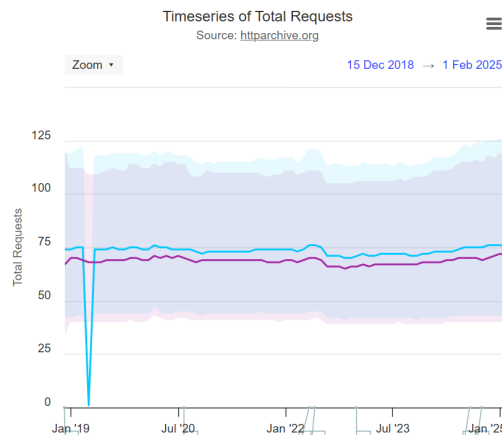
Gambar 2.1: Ukuran sample *web* yang digunakan untuk analisis

- *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.

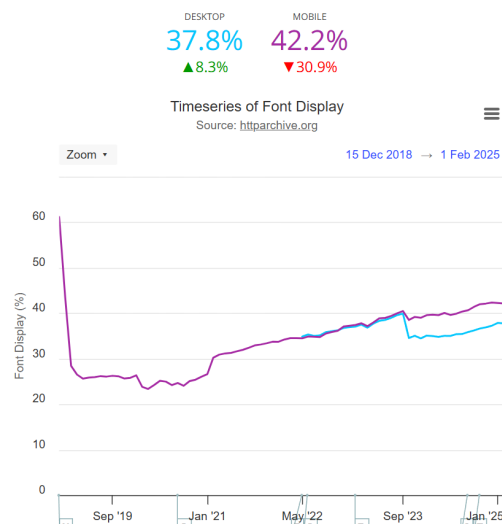


Gambar 2.2: Total *kilobyte* yang *direquest* oleh halaman *web*

- *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.

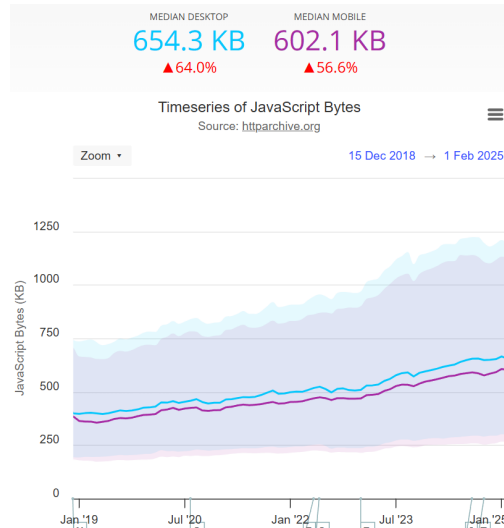
Gambar 2.4: Persentase web yang memiliki properti `font-display`

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

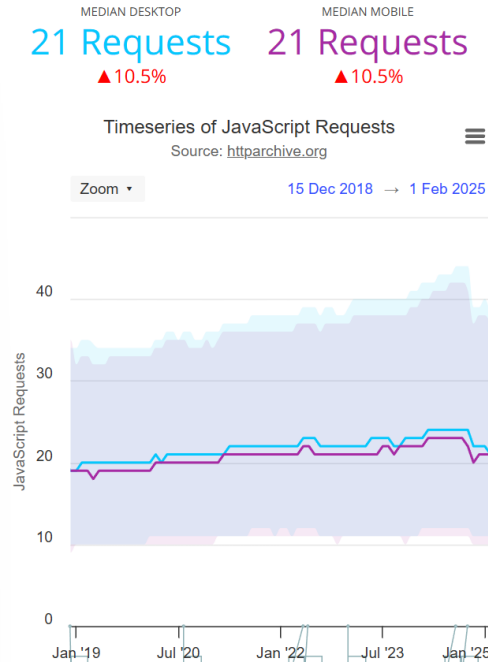
1 hal yang dianalisis adalah sebagai berikut:

- 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*.

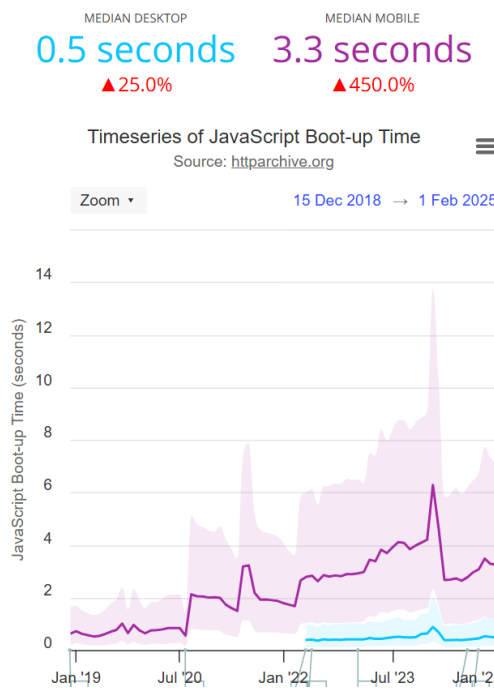


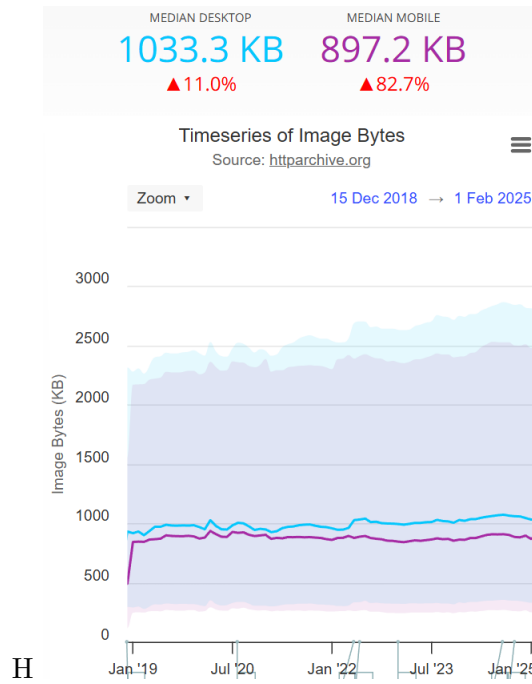
Gambar 2.5: Jumlah *kilobytes* dari *resource JavaScript*

- 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*

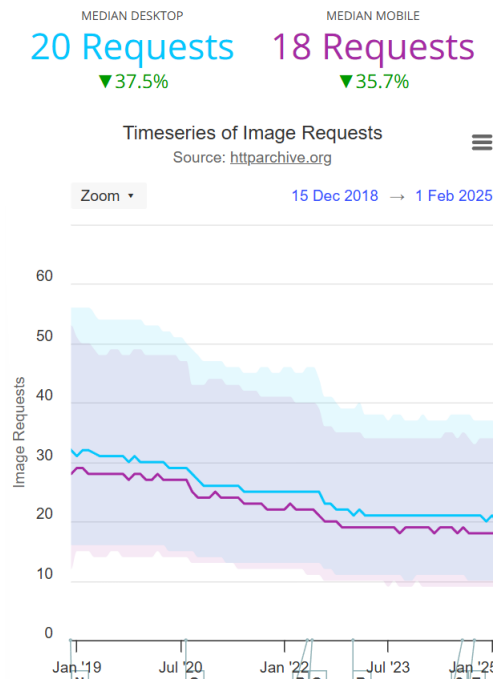


Gambar 2.8: Perkembangan perpindahan jumlah *kilobytes* yang dibutuhkan oleh gambar

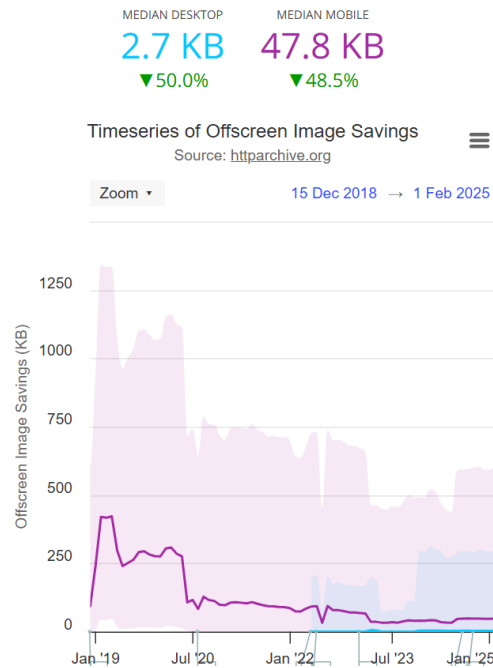
1 *State of Images*

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.

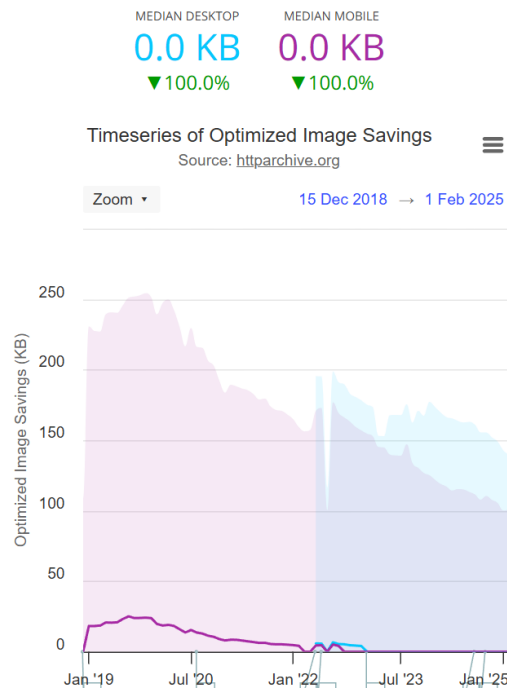
Gambar 2.9: Perkembangan jumlah *request* untuk gambar

- *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*.

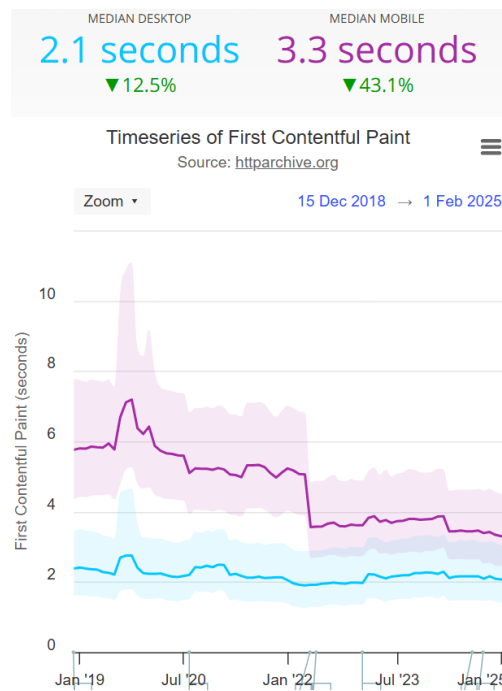


Gambar 2.11: Perkembangan jumlah kilobyte yang dapat dihemat dengan melakukan kompresi pada gambar

1 Loading Speed

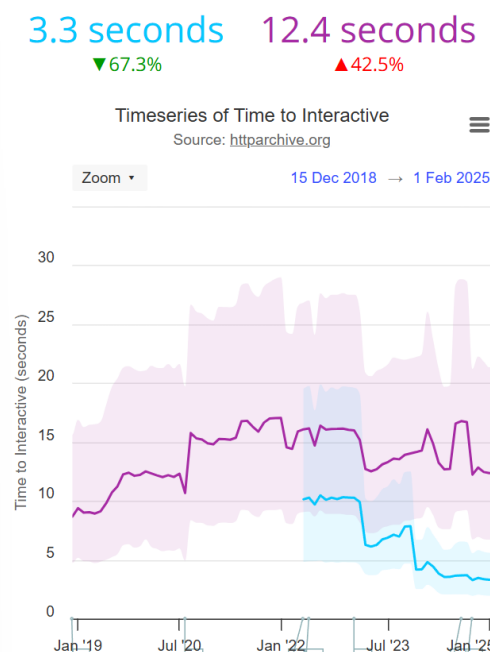
Performa *web* dapat berpengaruh langsung terhadap bisnis seperti kepuasan pengguna. Laporan ini akan menganalisis berbagai matriks performa dalam siklus pemuatan halaman *web* termasuk yang digunakan oleh aplikasi *web* modern. Kategori ini memiliki beberapa laporan sebagai berikut:

- *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*.



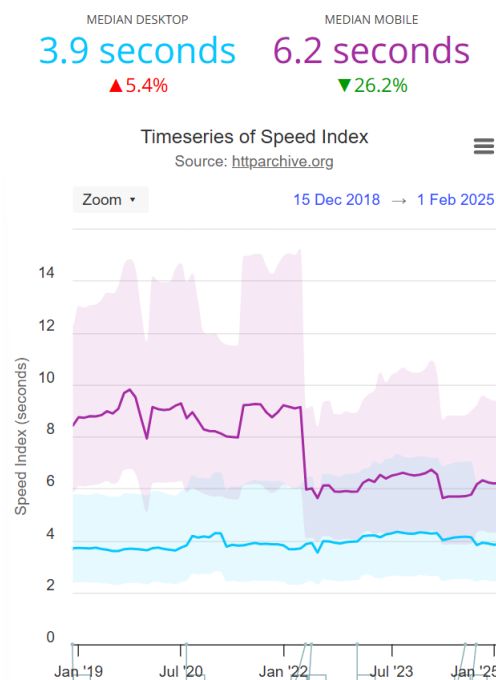
Gambar 2.12: Perkembangan waktu yang dibutuhkan untuk menampilkan konten utama

- *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.

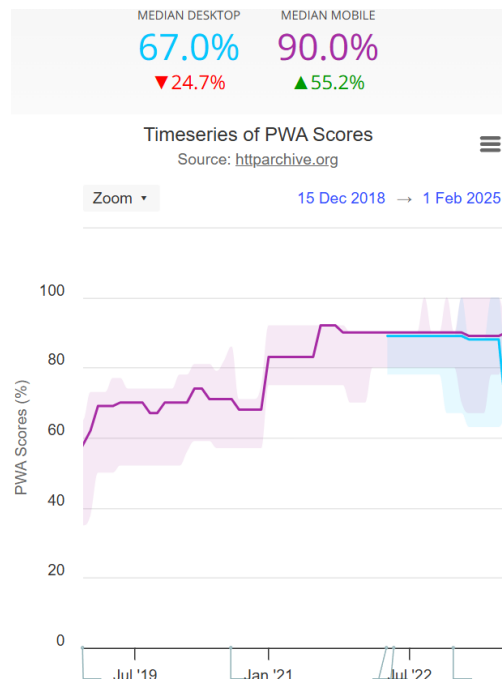


Gambar 2.14: Perkembangan waktu yang dibutuhkan untuk konten dapat terlihat secara jelas

Progressive Web App

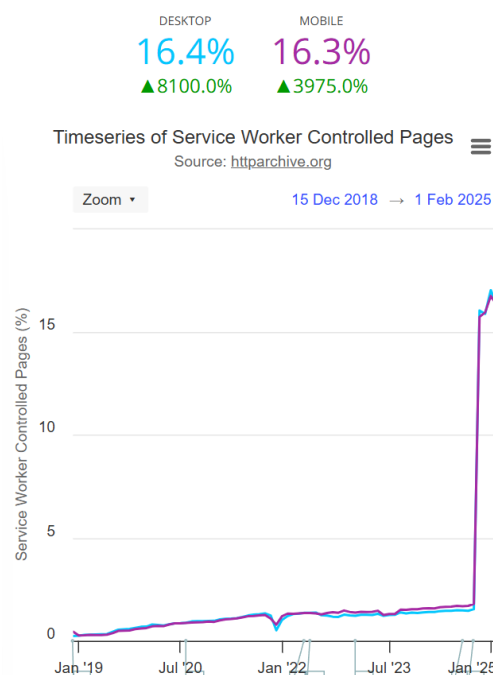
Laporan ini akan mengkaji status dari *Progressive Web App*. *Progressive Web App* adalah kelas baru dari aplikasi *web* yang disediakan oleh *Service Workers APIs*. *Service Workers* memungkinkan aplikasi untuk mendukung proses muat jaringan secara independen, menerima *push notifications* untuk menyinkronkan 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

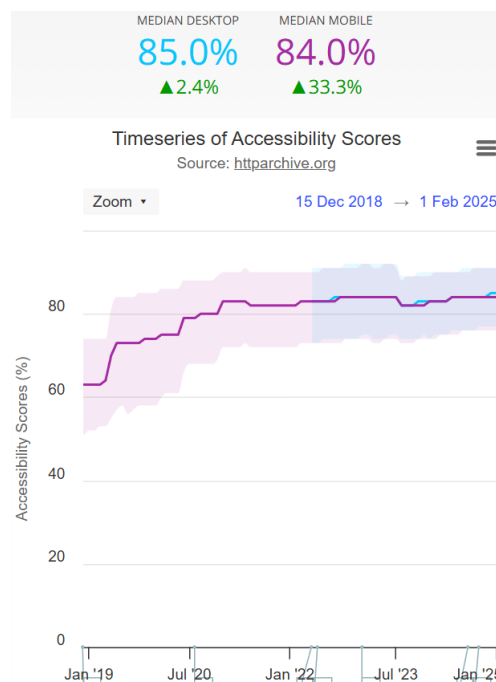
- *Service Worker Controlled Pages* yang berisi persentase dari jumlah halaman yang telah memicu `ServiceWorkerControlledPage use counter` yang aktif ketika sebuah halaman *web* dikendalikan oleh *service worker*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.16. Terlihat dari data yang 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

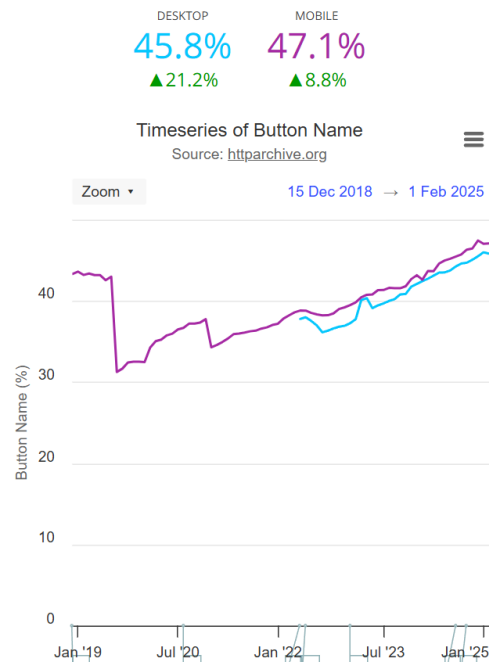
Laporan ini menjelaskan tingkat aksesibilitas dari sebuah halaman *web*. Penilaian ini dilakukan oleh *Lighthouse*. Kategori ini berisi beberapa laporan seperti:

- *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.



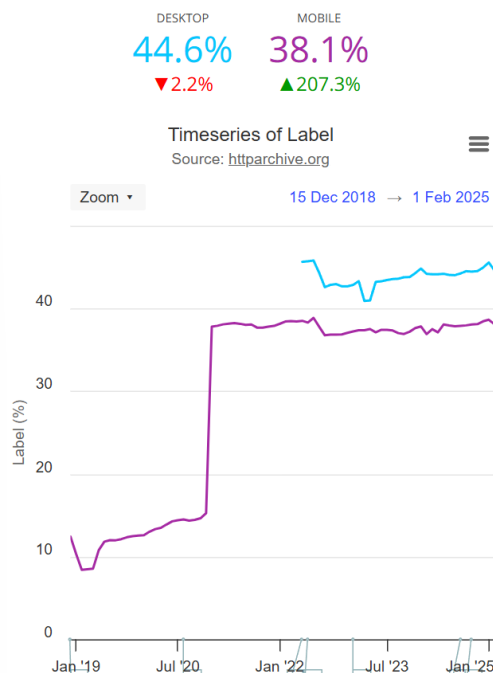
Gambar 2.17: Perkembangan persentase skor aksesibilitas

- *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.

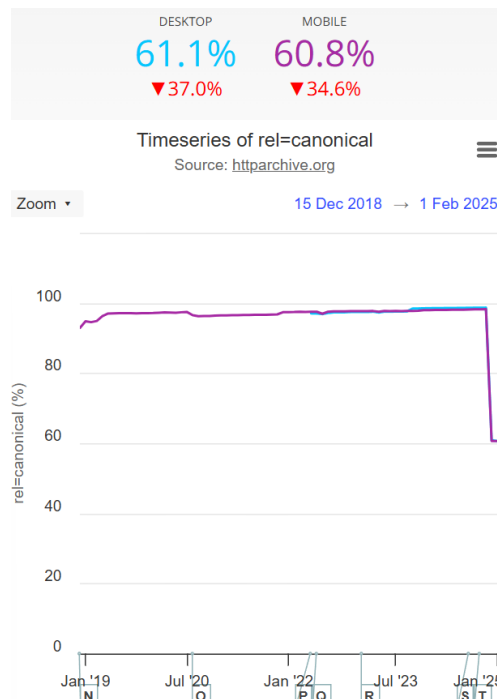


Gambar 2.19: Perkembangan persentase halaman yang menggunakan label yang aksesible

1 SEO

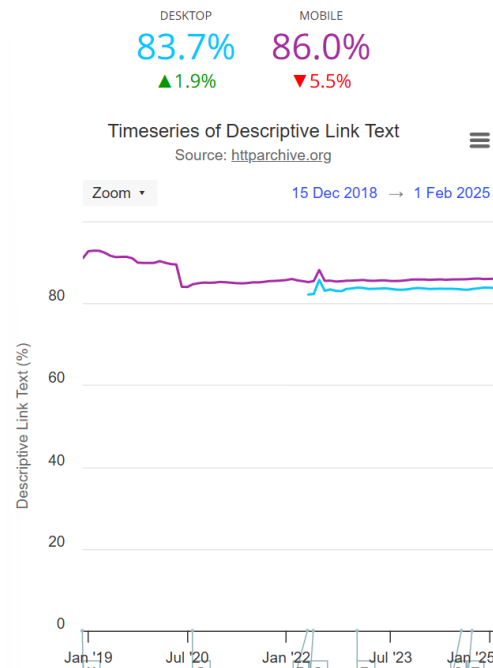
2 Laporan yang akan menelusuri penggunaan beberapa teknik agar halaman *web* dapat dikenali oleh
3 mesin pencarian secara lebih baik. Kategori ini memiliki beberapa laporan seperti:

- 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.

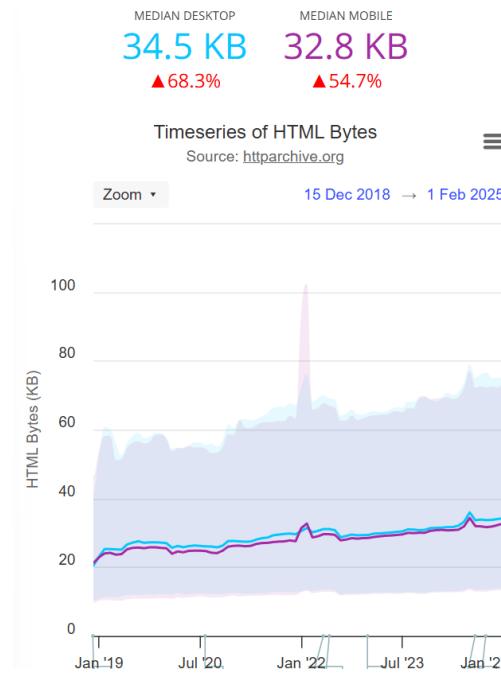


Gambar 2.21: Perkembangan persentase halaman yang memiliki *link* yang deskriptif

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.

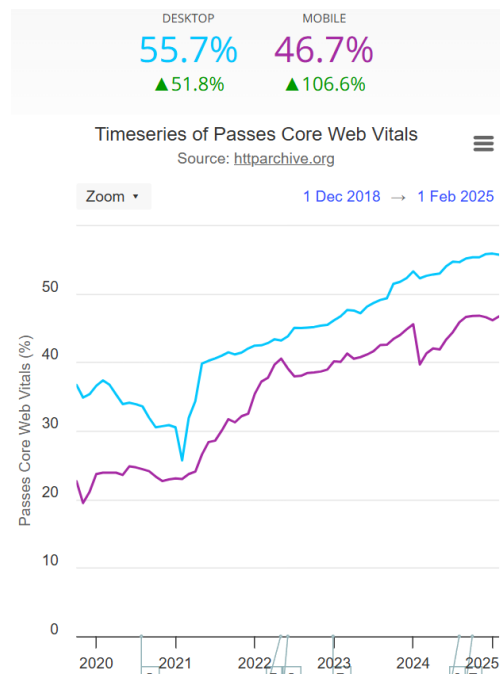


Gambar 2.22: PERkembangan jumlah *kilobytes* oleh dokumen HTML

1 *CrUx*

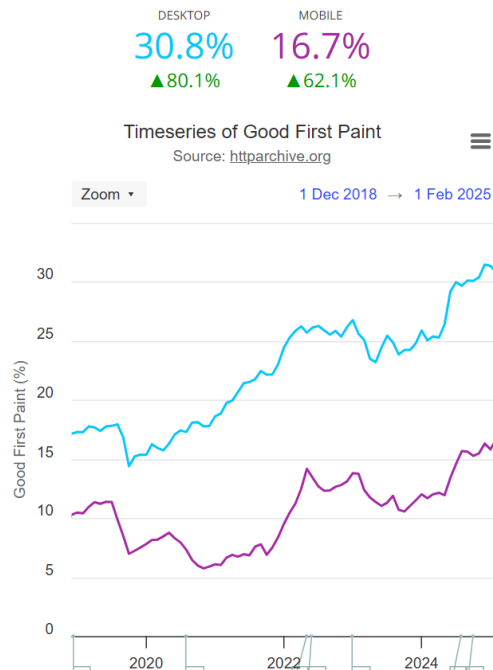
2 Laporan ini akan menelusuri tingkat interaktivitas dan proses muat dari pengguna *Chrome* di
3 dunia nyata melalui berbagai kondisi perangkat keras dan jaringan. Kategori ini memiliki beberapa
4 laporan seperti:

- 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.



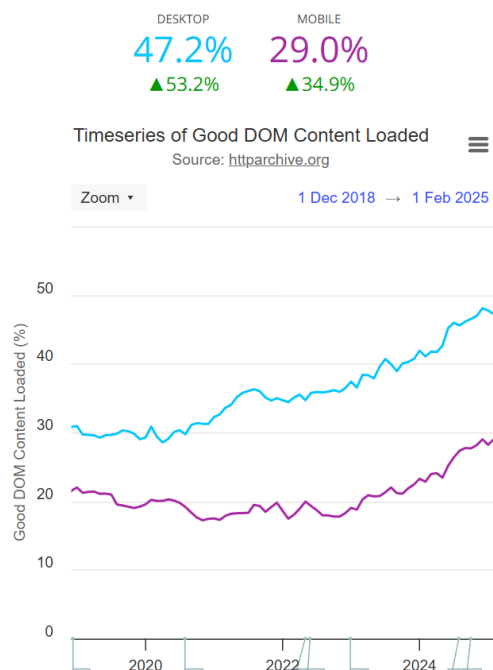
Gambar 2.23: Perkembangan oersentase halaman *web* yang memiliki penilaian *core web vitalis* yang bagus

- *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*.

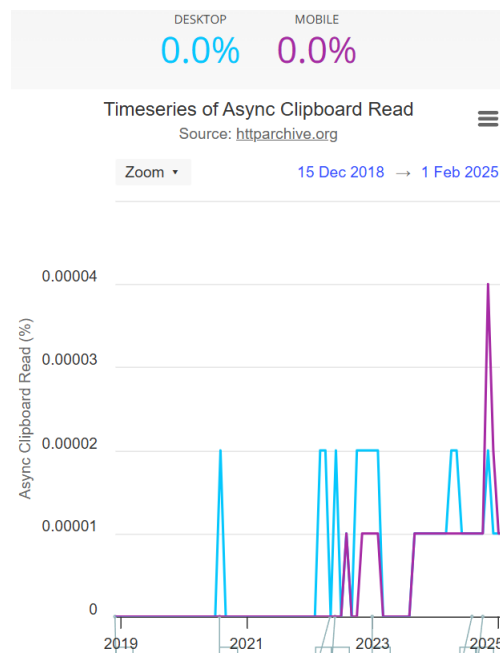


Gambar 2.25: Contoh perkembangan persentase halaman *web* yang memiliki pengalaman DCL yang baik

1 *Capabilities*

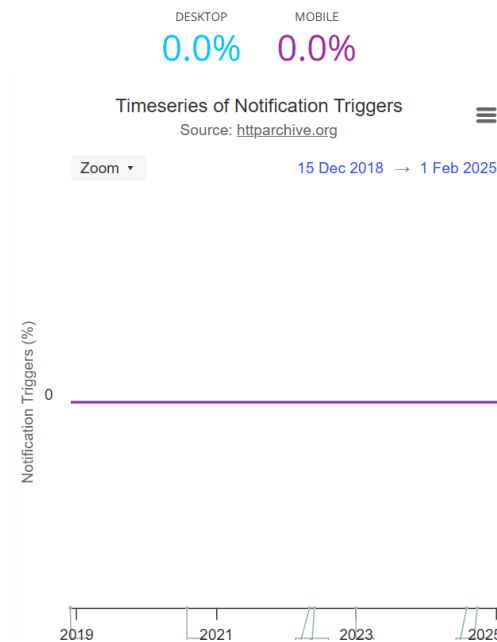
2 *Capabilities Project* adalah usaha yang dilakukan oleh *Google* dengan perusahaan lain untuk
 3 memungkinkan sebuah aplikasi *web* dapat melakukan hal yang dapat dilakukan oleh aplikasi bawaan
 4 dari sistem operasi sambil tetap mempertahankan keamanan pengguna, privasi, kepercayaan, dan
 5 prinsip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

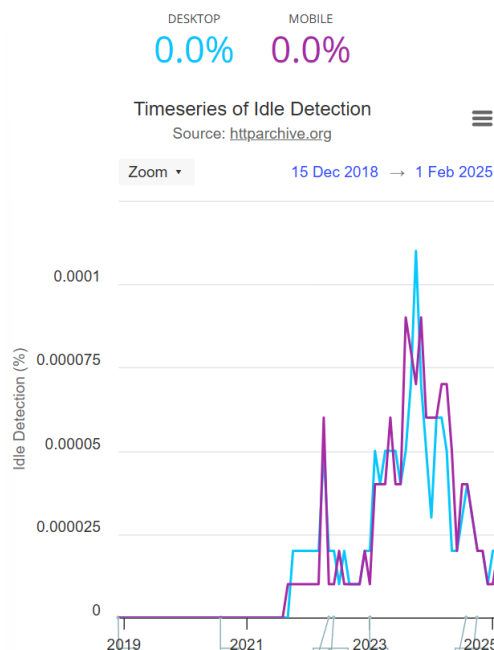


Gambar 2.26: Perkembangan persentase halaman *web* yang membaca *clipboard* milik sistem.

- 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*

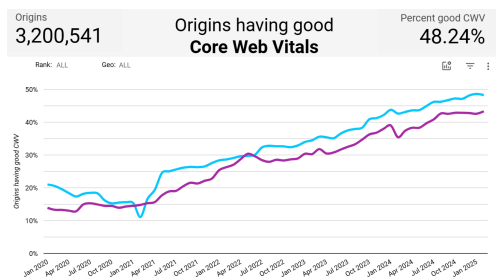
- *Idle Detection* yang berisi persentase halaman yang mendeteksi penggunaanya sedang melakukan *idle* melalui API *Idle Detection*. Contoh visualisasi untuk data ini dapat dilihat pada Gambar 2.28. Terlihat dari data yang diambil dari 15 Desember 2018 hingga 1 Februari 2025 pada perangkat *desktop* 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

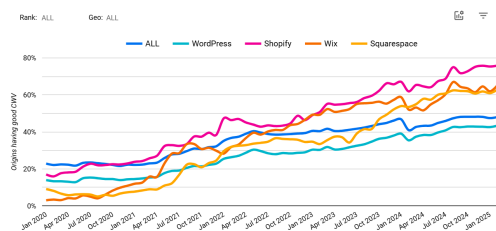
Laporan *Core Web Vitals Technology* merupakan dasbor yang berisi gabungan dari pengalaman nyata pengguna *chrome*(CrUX) dengan pendeteksi teknologi *web* yang dimiliki *HTTP Archive*, yang memungkinkan analisis terhadap cara sebuah *web* dibangun dan pengalaman saat *web* tersebut digunakan. Kategori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

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

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

- *crawl* adalah tabel yang telah dipartisi dan dikelompokkan yang berisi halaman yang dites oleh *HTTP Archive*.
- *sample_data* adalah tabel yang merupakan potongan dari tabel *crawl* yang dapat digunakan

untuk menjalankan *query* dengan waktu eksekusi yang lebih cepat.

- *technologies* yang berisi mengenai detail dari teknologi yang dipakai oleh berbagai *web*.

2.2 *Structure Query Language* [2]

Structure Query Language atau yang biasa disebut SQL merupakan bahasa pemrograman yang bertujuan untuk memanipulasi atau mengubah basis data. Ben Forta dalam bukunya menjelaskan bahwa bahasa ini didesain untuk mengerjakan sebuah perintah dengan tepat dan benar agar proses pembuatan atau pengambilan data berjalan lebih efisien [5]. Data disimpan dalam bentuk tabel ke dalam basis data, untuk mengakses data tersebut SQL menyediakan beberapa sintaks yang bisa dipakai. Sintaks yang dapat dipakai adalah sebagai berikut:

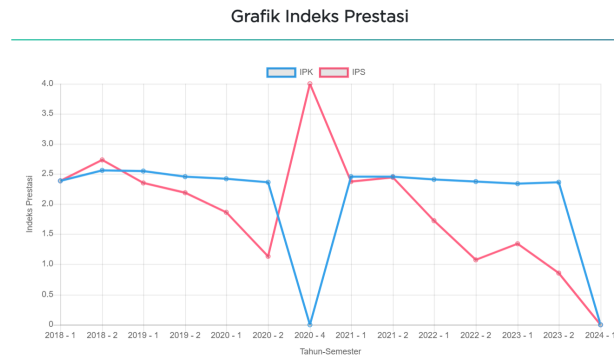
- **SELECT** dan **FROM** merupakan sintaks yang berguna untuk memilih bagian data yang dibutuhkan dari tabel tertentu.
- **WHERE** adalah sintaks yang berguna untuk memberikan kondisi tertentu dalam memilih data. Kondisi yang diinginkan bisa lebih dari satu maka dari itu untuk menggabungkan beberapa kondisi dapat digunakan fungsi **AND** atau **OR**.
- **GROUP BY** adalah sintaks yang berguna untuk mengelompokan data berdasarkan kelas tertentu yang terdapat dalam data. Pengelompokan ini dapat dilakukan berdasarkan lebih dari satu kelas.
- **UNNEST** adalah fungsi yang mengembalikan elemen dari sebuah array menjadi sebuah baris atau *row*.
- **COUNT** adalah fungsi agregat yang berguna untuk menghitung berapa banyak *row* yang memiliki nilai tertentu.
- **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*

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 kecil yang digunakan merupakan data *sample* yang diberikan oleh *HTTP Archive*. 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

```

6
7 1 SELECT DISTINCT date, client, root_page, t.technology,
8 2 FROM 'httparchive.crawl.pages', UNNEST(technologies) as t
9 3 WHERE date BETWEEN "2024-02-01" and "2025-02-01" limit 100000000
10 4

```

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 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	root_page	technology
2025-02-01	desktop	https://www.schilling-rechtsanwalt.com/	Google Analytics
2025-02-01	mobile	https://kiztopia.co.id/	HSTS
2025-02-01	mobile	https://ca.brixton.com/	Klaviyo
2025-02-01	mobile	https://spinfinity.casino/	Gatsby
2025-02-01	mobile	http://namhaerun.com/	YouTube
...

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
...	...

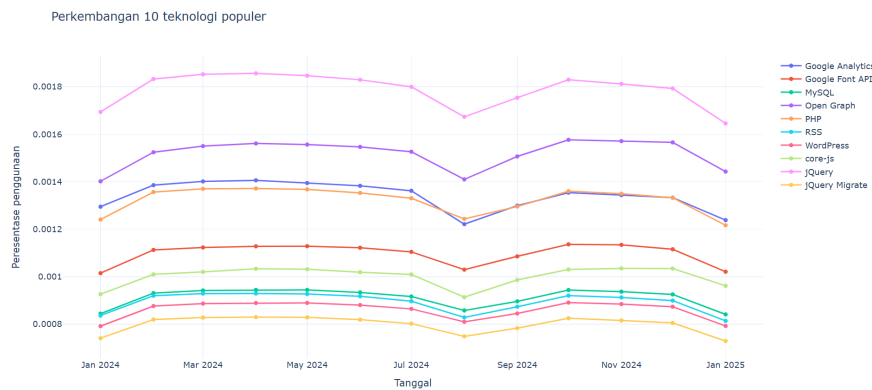
6 Setelah itu sepuluh teknologi yang paling banyak digunakan akan diambil. Kode yang digunakan
 7 untuk mencari sepuluh teknologi yang paling banyak digunakan adalah `df_10 = df_sorted.head(10)`.
 8 Sepuluh teknologi dengan penggunaan paling banyak dapat dilihat pada tabel 3.4. Terlihat bahwa
 9 dua 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. Jumlah penggunaan untuk setiap teknologi juga dibagi dengan jumlah pemakaian untuk semua teknologi. 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 3.1. 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 3.1: Perkembangan sepuluh teknologi populer berdasarkan jumlah halaman *web*

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 menggunakan kode `df_mobile_gb = df_mobile.groupby(["technology"]).sum('f0_')`. Proses yang dilakukan masih sama seperti pada Bagian 3.2.2 namun ada sedikit perbedaan pada sepuluh teknologi yang paling banyak digunakan seperti yang terlihat pada Tabel 3.5. Pada *client mobile* terlihat 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

Setelah menemukan sepuluh teknologi yang paling banyak digunakan pada *client mobile* hal yang selanjutnya dilakukan adalah melakukan visualisasi. Visualisasi ini dilakukan dengan cara yang sama seperti pada Bagian 3.2.2, yaitu dengan membagi jumlah penggunaan setiap teknologi dengan jumlah pemakaian semua teknologi. Visualisasi dilakukan dengan menjalankan Kode 3.2. Kode tersebut membuat visualisasi dari data yang telah dibuat menjadi pivot tabel. Kode tersebut juga membuat visualisasi ke dalam bentuk HTML.

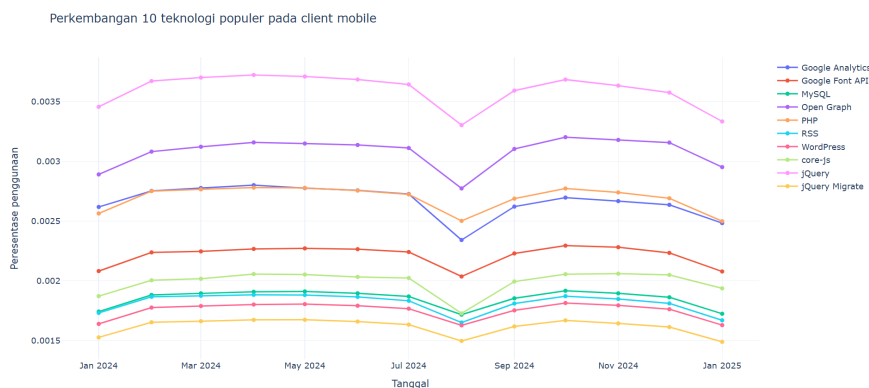
Kode 3.2: Kode untuk membuat visualisasi pada *client mobile*

```

7
81
92 fig3 = go.Figure()
103 for col in df_pivot_mobile.columns:
114     fig3.add_trace(go.Scatter(x=df_pivot_mobile.index, y=df_pivot_mobile[col], mode='lines+markers', name=col))
125
136 fig3.update_layout(
147     title="Perkembangan_10_teknologi_populer_pada_client_mobile",
158     xaxis_title="Tanggal",
169     yaxis_title="Peresentase_penggunaan",
170     template="plotly_white"
181 )
192 fig3.show()
203 fig3.write_html('sample_mobile.html')

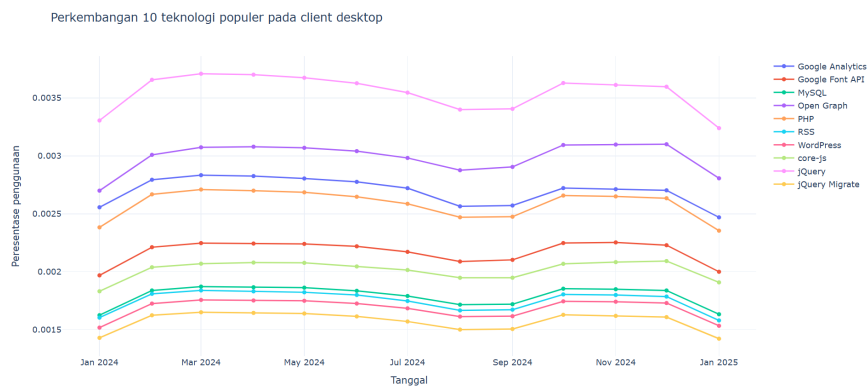
```

Hasil visualisasi dapat dilihat pada Gambar 3.2. Hasil visualisasinya memperlihatkan hal yang hampir sama seperti pada Bagian 3.2.2. Semua teknologi mengalami penurunan penggunaan pada 1 Agustus 2024 dan penggunaan semua teknologi terlihat stabil.

Gambar 3.2: Perkembangan sepuluh teknologi populer pada *client mobile*

1 Perkembangan Sepuluh Teknologi pada *Client desktop*

2 Metode yang dilakukan untuk melihat perkembangan sepuluh teknologi yang paling banyak digunak-
 3 an untuk *client desktop* sama seperti pada Bagian 3.2.2 yaitu memisahkan data yang memiliki *client*
 4 *desktop* terlebih dahulu kemudian melakukan pengolahan sehingga mendapatkan hasil visualisasi
 5 yang diinginkan. Jumlah penggunaan untuk setiap teknologi yang populer juga dibagi dengan
 6 jumlah penggunaan semua teknologi. Hasil visualisasi dapat dilihat pada Gambar 3.3. Pada *client*
 7 *desktop* penggunaan teknologi PHP tidak sebanyak pada perangkat *mobile*. Kemudian pada tanggal
 8 1 Agustus 2024 juga semua teknologi mengalami penurunan penggunaan. Namun untuk *client*
 9 *desktop* penggunaan semua teknologi tidak langsung mengalami kenaikan di bulan september namun
 10 baru meningkat kembali di bulan Oktober.



Gambar 3.3: Perkembangan sepuluh teknologi yang populer untuk *client desktop*

DAFTAR REFERENSI

- [1] Viscomia, R., Calvano, P., Chen, S., Hobbs, M., dan Demir, N. (2024) HTTP Archive about. Technical report. HTTP Archive.
- [2] Lakshmanan, J., Valliappa; Tigani (2020) *Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale*, first edition edition. O'Reilly Media.
- [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] Forta, B. (2019) *SQL in 10 Minutes a Day, Sams Teach Yourself*, 5th edition. Sams Publishing.

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf

```

Kode A.2: MyCode.java

```

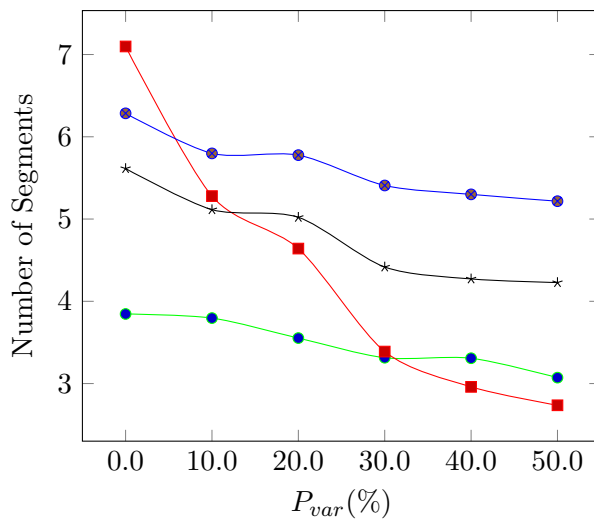
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }

```


LAMPIRAN B

HASIL EKSPERIMEN

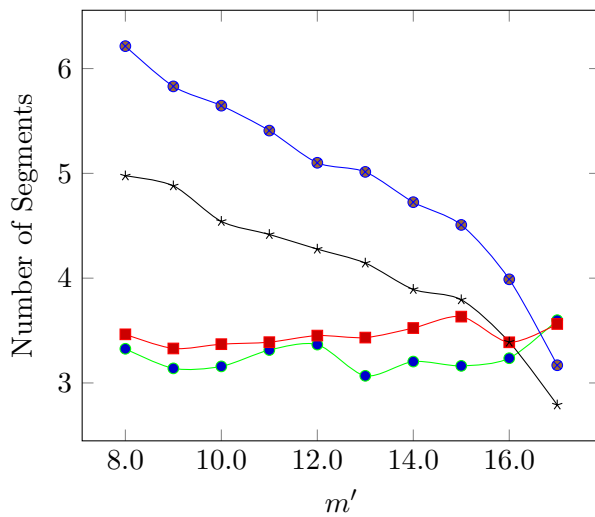
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4