

Analisis Performa Aplikasi PIKOBAR Berbasis Android Native dan Progressive Web App

Qansa Alghifari¹

Jurusan Teknik Informatika, Fakultas Teknologi Informasi dan Komputer, Universitas Nusa Putra¹
Jl. Raya Cibatucisaat No.21, Cibolang Kaler, Kec. Cisaat, Kab. Sukabumi, Jawa Barat 43155
E-mail : qansa.alghifari_ti17@nusaputra.ac.id¹

Abstract -- Penyebaran kasus virus Corona di Jawa Barat pada tahun 2020 menyebabkan Pemprov Jawa Barat membuat inovasi berupa aplikasi PIKOBAR (Pusat Informasi dan Koordinasi COVID-19 Jawa Barat). Aplikasi tersebut dibuat dengan berbagai versi. Pada penelitian ini dilakukan penelitian pada aplikasi PIKOBAR berbasis *Android Native* dan *Progressive Web App*. Pengujian dilakukan pada perangkat *Smartphone Android* dengan melihat *CPU Utilization*, *Memory Usage* dan *Execution Time* saat menjalankan fitur pada kedua aplikasi. Hasil dari pengujian ini menunjukkan bahwa aplikasi PIKOBAR berbasis PWA lebih unggul dengan *CPU Utilization* dan *Memory Usage* yang lebih rendah. Namun pada *Execution Time* aplikasi PIKOBAR berbasis *Android Native* lebih unggul dengan rata-rata 1766,67 ms. Berdasarkan hasil dari keseluruhan pengujian dapat disimpulkan bahwa aplikasi PIKOBAR berbasis *Progressive Web App* memiliki performa lebih baik dibandingkan aplikasi PIKOBAR berbasis *Android Native*.

Kata Kunci: Pikobar, Android, PWA, CPU Usage, Memory Usage, Execution Time

Abstract -- The prevalence of the Corona virus making West Java provincial government to make innovations in the form of application PIKOBAR (Center for Information and Coordination COVID-19 West Java). The application is made with different versions. In this study, conducted research on the application PIKOBAR Android-based Native and Progressive Web App. Testing performed on device Android Smartphone by looking at CPU Utilization, Memory Usage and Execution Time when running the features in both applications. The results of this test show that the PIKOBAR PWA-based application is superior with lower CPU Utilization and Memory Usage. However, Execution Time the PIKOBAR application based Native Android is superior to the average 1766.67 ms. based on the results of the test can be concluded that the overall application PIKOBAR Progressive web App has performed better than Android Native based.

Keywords: Pikobar, Android, PWA, CPU Usage, Memory Usage, Execution Time

I. PENDAHULUAN

Semenjak mewabahnya virus *corona* di Indonesia khususnya di Provinsi Jawa Barat pada pertengahan tahun 2020, pemerintah Provinsi Jawa Barat meluncurkan aplikasi PIKOBAR yang merupakan singkatan dari Pusat Informasi dan Koordinasi *COVID-19* Jawa Barat. Yang berfungsi sebagai media informasi bagi warga tentang penyebaran dan penanganan kasus *Covid-19* di Jawa Barat [1]. Aplikasi ini memiliki beberapa versi diantaranya adalah versi *android native* dan *PWA*. *PWA* atau *Progressive Web App* merupakan sebuah *website* yang dibangun menggunakan *web modern* namun dapat bekerja layaknya *Mobile App* [2]. *PWA* dapat bekerja layaknya sebuah aplikasi *mobile native* yang memiliki fitur menambahkan *icon* pada beranda, dapat mengirim notifikasi hingga dapat bekerja dalam kondisi jaringan apapun. Namun *PWA* tidak disebarluaskan melalui media penyedia aplikasi seperti *Google Play Store*, *App Store* dll. *PWA* berfungsi dengan menggunakan *web browser* seperti *Google Chrome*. Penelitian ini bertujuan untuk mengetahui performa aplikasi PIKOBAR berbasis *PWA* dan *Android Native*. Sehingga dapat diketahui aplikasi dengan versi mana yang memiliki performa yang handal.

II. TINJAUAN PUSTAKA

Aplikasi PIKOBAR kepanjangan dari Pusat Informasi dan Komunikasi *Covid-19* Jawa Barat, merupakan sebuah perangkat lunak yang berisi informasi mengenai kasus penyebaran dan penanganan *Covid-19* di Jawa Barat [1]. Aplikasi ini dibangun untuk beberapa *platform* yaitu *Website/PWA*, *Android* & *Ios*. Berikut adalah tampilan aplikasi PIKOBAR berbasis *Android* & *PWA*. Aplikasi ini bertujuan dalam upaya penanganan dan penanggulangan penyebaran *COVID-19* di Jawa Barat.



Gambar 1. Tampilan Aplikasi PIKOBAR berbasis *Android Native*.



Gambar 2. Tampilan Aplikasi PIKOBAR berbasis *Progressive Web App*.

PWA merupakan kepanjangan dari *Progressive Web App* merupakan sebuah *web* yang dibangun dengan teknologi *web modern* namun dapat berlaku seperti *Mobile App* [2]. Berbagai fitur yang terdapat pada *PWA* seperti tampilan *responsive*, menggunakan *service worker*, dapat mengirim notifikasi dan bisa menambahkan *icon* pada beranda seperti aplikasi *mobile native*.

Kelebihan lainnya dari *PWA* adalah memiliki tampilan yang dapat menyesuaikan dengan perangkat yang menggunakannya, seperti *smartphone*, *tablet*, *notebook* dll [3]. Dengan demikian *PWA* dapat digunakan oleh beberapa perangkat dengan tetap dalam kondisi optimal.

Service worker adalah salah satu jenis dari *web worker*, yaitu *script* yang berjalan di belakang *browser* pengguna. *Service worker* pada dasarnya adalah berkas *Java Script* yang berjalan pada *thread* yang berbeda dengan *main thread browser*, menangani *network request*, *caching*, mengembalikan *resource* dari *cache*, dan bisa mengirimkan *push message*. [3]. Dengan teknologi ini *PWA* tetap dapat digunakan walaupun *offline*.

Add To Home Screen merupakan suatu fitur yang ada pada *PWA*. Fitur ini berfungsi sebagai penambah *icon* aplikasi pada beranda sehingga dapat berfungsi seperti aplikasi *mobile native*. Dengan ini pengguna dapat membuka aplikasi tanpa harus menuliskan alamat pada *browser*.

PWA sangat cepat dalam *loading* halaman karena menggunakan *service worker* yang telah menyimpan *cache*, sehingga aplikasi akan selalu menampilkan halaman yang terakhir diakses sebelum ditutup. *Teknologi service worker* memiliki kemampuan untuk dapat mengontrol *asset* yang akan di *cache* [3].

Aplikasi *native* atau *native app* adalah sebuah perangkat lunak yang dibangun menggunakan bahasa pemrograman tertentu untuk *platform* dengan sistem perangkat yang berbeda-beda. Aplikasi *Android Native* dibangun oleh bahasa pemrograman diantaranya *Java* dan *Kotlin*[4].

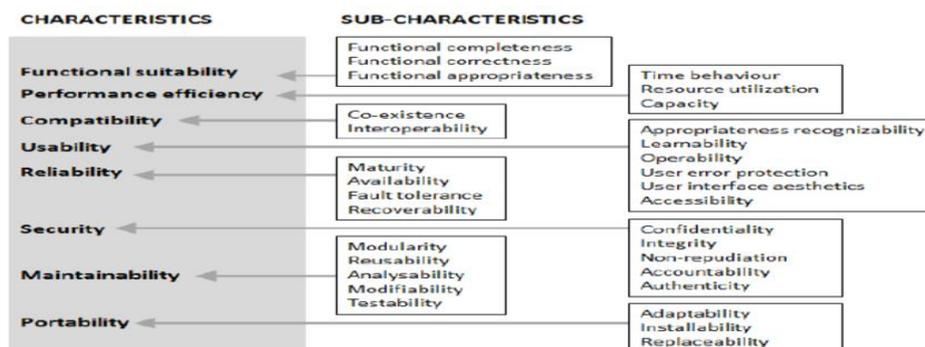
Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems*, yang saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada *C* dan *C++* namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis *java* umumnya dikompilasi ke dalam *p-code (bytecode)* dan dapat dijalankan pada berbagai Mesin Virtual *Java (JVM)* [5].

Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada Mesin Virtual *Java* ataupun menggunakan kompiler *LLVM* yang dapat pula dikompilasikan kedalam bentuk kode sumber *JavaScript*. Pengembang utamanya berasal dari tim programmer dari *JetBrains* yang bermarkas di Rusia. Meskipun sintaksisnya tidak kompatibel dengan bahasa *Java*, *Kotlin* didesain untuk dapat bekerja sama dengan kode bahasa *Java* dan bergantung kepada kode bahasa *Java* dari Kelas Pustaka *Java* yang ada, seperti berbagai *framework Java* yang ada [6].

Snapdragon Profiler adalah perangkat lunak yang berjalan pada *platform Windows, Mac, dan Linux*. Ini terhubung dengan perangkat *android* yang didukung oleh prosesor *Snapdragon* melalui *USB*. *Snapdragon Profiler* memungkinkan pengembang untuk menganalisis data CPU, GPU, DSP, memori, daya, termal, dan jaringan, sehingga dapat menemukan dan memperbaiki bug kinerja pada aplikasi. Fitur dan keuntungan yang diperoleh menggunakan *Snapdragon Profiler* adalah :

1. Terdapat tampilan *realtime* yang memudahkan untuk memantau resource pada perangkat seperti CPU, GPU, DSP, memory, power, jaringan dan lain-lain.
2. Mode *Trace Capture* yang memungkinkan pengguna untuk memvisualisasikan kejadian pada sistem dan perangkat seperti CPU, GPU dan lain-lain.
3. Melihat dan mengedit tampilan dan preview hasil capture pada perangkat.[7].

Standar performa aplikasi berdasarkan *ISO/IEC 25010* yang merupakan versi lanjutan dari *ISO/IEC 9126* dengan penambahan beberapa struktur dan bagian dari standar model kualitas. Secara keseluruhan *ISO/IEC 25010* memiliki 8 karakteristik untuk mengukur kualitas perangkat lunak secara menyeluruh, antara lain *portability, performance efficiency, reliability, security, usability, maintainability, compability* dan *functional suitability* [8]. Gambar 3 merupakan *Product quality model* berdasarkan *ISO/IEC 25010*. Dalam penelitian ini akan menggunakan aspek performa aplikasi untuk dijadikan ukuran penelitian pada aplikasi berbasis *android* dan *PWA*.



Gambar 3. *Product Quality Model* berdasarkan *ISO/IEC 25010*.

III. METODOLOGI

Metodologi adalah proses – proses yang dilakukan dalam penyelesaian penelitian yang dilakukan. Adapun tahapannya berupa studi literatur pada tiap proses yang dilakukan dalam mengumpulkan dan mempelajari literatur yang terkait dalam penelitian ini, seperti :

Melakukan analisis dan menyiapkan kebutuhan yang akan digunakan dalam penelitian. Kebutuhan yang digunakan dalam penelitian ini adalah berupa akses internet dengan kecepatan 20 Mbps, perangkat *Smartphone Android* dengan spesifikasi 4GB RAM, *Processor Snapdragon 625* dengan versi sistem operasi *Android Nougat (7.0)* & aplikasi *Snapdragon Profiler*, yang digunakan untuk mengukur performa dari aplikasi PIKOBAR.

Objek yang dijadikan penelitian adalah berupa *CPU Utilization*, *Memory Usage* & *Execution Time*, pada perangkat *Smartphone* ketika mengoperasikan fitur data penyebaran kasus *COVID-19* pada aplikasi PIKOBAR yang berbasis *Android Native* dan *Progressive Web App*.

Pengujian dilakukan terhadap aplikasi PIKOBAR berbasis *Android Native & Progressive Web App*. Tabel 1 merupakan variabel performa yang akan diukur berdasarkan ISO/IEC 25010 tentang performa aplikasi. Pengujian dilakukan dengan cara mencoba fitur yang ada pada aplikasi PIKOBAR dari tiap versi sebanyak 3 kali percobaan. Aplikasi PIKOBAR dengan *CPU Utilization*, *Memory Usage* & *Execution Time* yang paling rendah, merupakan aplikasi dengan performa yang lebih baik.

Tabel 1. Variabel performa program yang diuji.

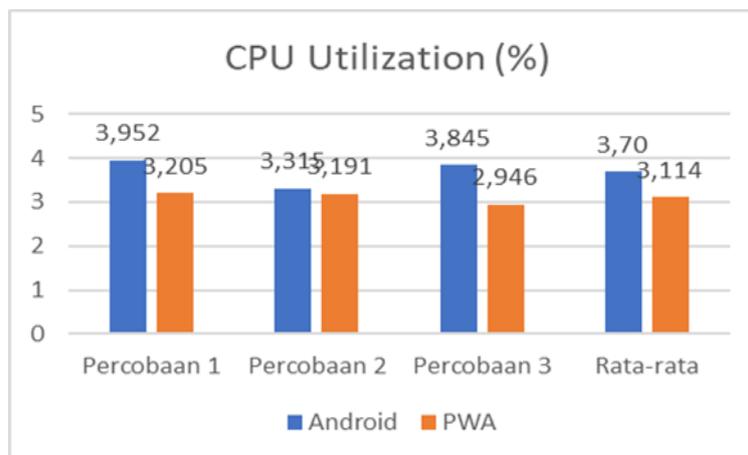
| NO | Variabel | Satuan | Keterangan |
|----|------------------------|-------------------------|--------------------------------------|
| 1 | <i>CPU Utilization</i> | Persentase (%) | Penggunaan <i>CPU</i> pada perangkat |
| 2 | <i>Memory Usage</i> | <i>Mega Byte (MB)</i> | Penggunaan memori pada perangkat |
| 3 | <i>Execution Time</i> | <i>Millisecond (ms)</i> | Waktu eksekusi program |

Mengulas tentang hasil penelitian yang dilakukan sesuai dengan rancangan objek kajian penelitian. Membahas kesimpulan serta saran yang didapatkan dari penelitian yang dilakukan pada *CPU Utilization*, *Memory Usage*, dan *Execution Time* pada perangkat saat membuka fitur-fitur pada kedua aplikasi.

IV. ANALISIS DAN PEMBAHASAN

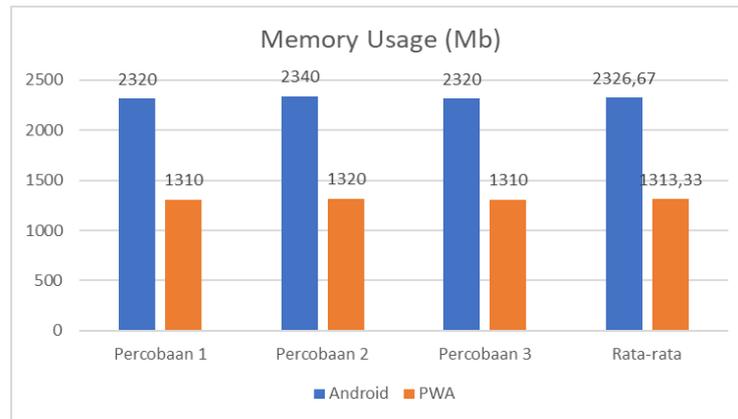
Pada bagian ini dilakukan ulasan data dari hasil penelitian yang telah dilakukan. Dari data yang didapatkan, maka akan diketahui performa dari dua aplikasi yang telah diuji. Kemudian akan dilihat dari data analisisnya aplikasi versi mana yang lebih baik untuk digunakan.

Penggunaan *CPU*. Berikut adalah hasil dari pengujian penggunaan *CPU* pada perangkat saat menjalankan kedua aplikasi. Gambar 4 menunjukkan hasil penggunaan *CPU*.



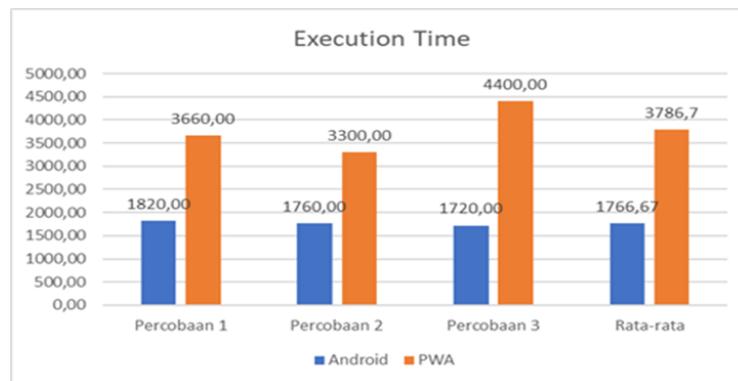
Gambar 4. Hasil pengujian *CPU Utilization*.

Penggunaan memori. Berikut adalah hasil dari pengujian penggunaan memori pada perangkat saat menjalankan kedua aplikasi. Gambar 5 menunjukkan hasil penggunaan memori.



Gambar 5. Hasil pengujian *Memory Usage*.

Waktu eksekusi. Berikut adalah hasil dari pengujian waktu eksekusi pada perangkat saat menjalankan kedua aplikasi. Gambar 6 menunjukkan hasil penggunaan memori.



Gambar 6. Hasil pengujian *Execution Time*.

Dengan mengamati hasil pengujian yang telah dilakukan pada *CPU Utilization, Memory Usage & Execution Time* pada perangkat saat membuka aplikasi. Untuk mengetahui versi aplikasi mana yang memiliki performa yang lebih baik untuk digunakan pada perangkat *Smartphone Android* maka hasil pengukuran tersebut dirangkum seperti berikut. Pada tabel 2 adalah hasil rekapitulasi dari seluruh pengukuran yang dilakukan. Hasil pengujian pada aplikasi di rata-ratakan sebagai berikut :

Tabel 1. Variabel performa program yang diuji.

| Pengujian | Android Native | PWA |
|----------------------------|----------------|---------|
| <i>CPU Utilization (%)</i> | 3,70 | 3,114 |
| <i>Memory Usage (Mb)</i> | 2326,67 | 1313,33 |
| <i>Execution Time (ms)</i> | 1766,67 | 3786,7 |

Berdasarkan hasil pengujian yang telah dilakukan terlihat bahwa aplikasi PIKOBAR berbasis *PWA* unggul pada *CPU Utilization* dan *Memory Usage* , sedangkan yang berbasis *Android Native* unggul pada *Execution Time*. Dengan demikian aplikasi PIKOBAR berbasis *PWA* lebih unggul. Hasil ini dipengaruhi oleh faktor bahwa *PWA* memiliki kelebihan karena menggunakan teknologi berupa *Service Worker*, yang mana dapat menyimpan data aplikasi yang telah dibuka sebelumnya sehingga membuat aplikasi dapat dibuka dalam kondisi jaringan *offline* sekalipun, dan membuat aplikasi *PWA* lebih rendah pada aspek *CPU Utilization & Memory Usage* pada perangkat

saat melakukan *loading* data dari *server*. Kemudian aplikasi PWA pun memiliki tampilan yang *responsive* dan dapat menambahkan *icon* pada beranda saat digunakan serta dapat mengirim notifikasi .

V. KESIMPULAN

Berdasarkan pada penelitian yang telah dilakukan dapat disimpulkan sebagai berikut;

1. Pengujian dilakukan pada aplikasi PIKOBAR milik Pemprov Jawa Barat. Kemudian dilakukan pengujian performa aplikasi saat dioperasikan pada perangkat *Smartphone Android* ,komponen yang diuji berupa *CPU Utilization*, *Memory Usage* dan *Execution Time*.
2. Pada hasil pengujian *CPU Utilization* pada kedua aplikasi menunjukkan hasil bahwa aplikasi berbasis PWA lebih unggul dengan hasil penggunaan *CPU* rata-rata 3,11 %, sedangkan aplikasi *Android native* memiliki nilai penggunaan *CPU* rata-rata sebesar 3,704 % yang digunakan pada perangkat saat membuka fitur pada aplikasi.
3. Hasil pengujian *Memory Usage* pada kedua aplikasi menunjukkan hasil bahwa aplikasi berbasis PWA lebih unggul pada komponen *Memory Usage* dengan hasil rata-rata sebesar 1313,33 Mb. Kemudian aplikasi *Android Native* memiliki hasil pengujian *Memory Usage* rata-rata sebesar 2326,67 Mb yang digunakan pada perangkat saat membuka fitur pada aplikasi. Dengan demikian aplikasi PWA lebih unggul dalam komponen *Memory Usage*.
4. Untuk hasil pengujian komponen *Execution Time* pada kedua aplikasi menunjukkan hasil bahwa aplikasi berbasis *Android Native* lebih unggul dengan hasil rata-rata 1766,67 ms, sedangkan aplikasi PWA memiliki hasil rata-rata 3786,7 ms pada perangkat untuk membuka fitur aplikasi. Dengan demikian aplikasi *Android Native* lebih unggul pada komponen *Execution Time*.
5. Berdasarkan semua hasil pengujian yang sudah dilakukan pada kedua aplikasi, maka didapatkan hasil bahwa aplikasi PIKOBAR berbasis *Progressive Web App* memiliki performa yang lebih baik daripada aplikasi berbasis *Android Antive*. Dan bisa menjadi pilihan untuk digunakan karena tidak perlu menginstal aplikasi pada perangkat, penggunaan *CPU & Memory* yang rendah pada perangkat, namun dapat memiliki *icon* pada beranda dan dapat mengirimkan notifikasi seperti layaknya aplikasi *Android Native*.

DAFTAR PUSTAKA

- [1] P. Jabar, "PIKOBAR," 2020. <https://pikobar.jabarprov.go.id/>.
- [2] A. Faisol and S. Noertjahjono, "(SIPERDIT) BERBASIS PROGRESSIVE WEB APP (PWA) Faisol | Sidik," vol. 2, no. 2, pp. 1–4, 2019.
- [3] K. Syaifudin, E. Pranata, W. N. A, and A. R. Dian, "Analisis Usability pada Perbandingan Web-Native dengan Web Berbasis Progressive Web App," *Informatika*, 2019.
- [4] A. Oliver, "Memahami Lebih Jauh Apa Itu Aplikasi Native," 2020. <https://glints.com/id/lowongan/aplikasi-native-adalah/#.X8pli2gzbIU> (accessed Nov. 29, 2020).
- [5] Wikipedia, "Java." <https://id.wikipedia.org/wiki/Java> (accessed Nov. 26, 2020).
- [6] Wikipedia, "Kotlin." [https://id.wikipedia.org/wiki/Kotlin_\(bahasa_pemrograman\)](https://id.wikipedia.org/wiki/Kotlin_(bahasa_pemrograman)) (accessed Nov. 28, 2020).
- [7] N. S. Sibarani, "Analisis Performa Aplikasi Native Android Menggunakan Bahasa Pemrograman Java dan Kotlin," 2018.
- [8] F. H. Wattiheluw, S. Rochimah, and C. Fatichah, "Klasifikasi Kualitas Perangkat Lunak Berdasarkan Iso/Iec 25010 Menggunakan Ahp Dan Fuzzy Mamdani Untuk Situs Web E-Commerce," *JUTI J. Ilm. Teknol. Inf.*, vol. 17, no. 1, p. 73, 2019, doi: 10.12962/j24068535.v17i1.a820.