

# Penerapan Progressive Web Apps untuk Pengembangan Fitur Push Notification dan Multi-Platform Installable pada Aplikasi Beasiswa

Hilmi Almuhtade Billah<sup>1</sup>, I Kadek Dwi Nuryana<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

<sup>1</sup>[hilmi.19033@mhs.unesa.ac.id](mailto:hilmi.19033@mhs.unesa.ac.id)

<sup>2</sup>[dwinuryana@unesa.ac.id](mailto:dwinuryana@unesa.ac.id)

**Abstrak**— Kecenderungan orang dalam melihat *website* yang sewaktu-waktu dan tidak rutin berpotensi untuk kehilangan informasi. Oleh karena itu diperlukan suatu cara yang dapat memberikan perhatian kepada pengguna jika terdapat informasi baru pada aplikasi web.

*Progressive Web Apps* merupakan sebuah teknologi pada aplikasi web agar memiliki kemampuan seperti aplikasi *native*. Salah satu fitur yang didukung oleh PWA adalah *push notification*. Selain itu, PWA juga mendukung fitur *multi-platform installable*. Fitur tersebut berguna agar aplikasi dapat dipasang pada perangkat yang beragam dan dapat meningkatkan pengalaman pengguna.

Tujuan dari penelitian ini adalah menerapkan PWA untuk pengembangan fitur *push notification* dan *multi-platform installable*, serta melakukan pengujian terhadap keberhasilan pengembangan dua fitur tersebut pada aplikasi beasiswa. Pengujian dilakukan dengan metode pengujian *blackbox* dan pengujian performa pengiriman notifikasi.

Hasil penelitian ini menunjukkan bahwa fitur *push notification* dan *multi-platform installable* pada aplikasi beasiswa berhasil dikembangkan dengan menerapkan PWA. Kedua fitur tersebut dapat bekerja pada sistem operasi android, windows, dan iOS. Hasil pengujian menunjukkan bahwa perangkat dengan sistem operasi android dan windows berhasil melalui seluruh skenario pengujian *blackbox*. Sedangkan perangkat dengan sistem operasi iOS gagal pada satu skenario pengujian yaitu pengiriman notifikasi dengan gambar.

Pengujian performa pengiriman notifikasi menunjukkan bahwa besar data gambar yang akan ditampilkan pada notifikasi tidak mempengaruhi durasi pengiriman notifikasi dari server menuju perangkat. Sedangkan besar data gambar mempengaruhi durasi perangkat dalam menampilkan notifikasi.

**Kata Kunci**— Progressive Web Apps, Push Notification, Multi-Platform, Firebase Cloud Messaging, Aplikasi Beasiswa.

## I. PENDAHULUAN

Pendidikan merupakan hal yang sangat penting dalam menunjang kemajuan suatu negara. Kualitas pendidikan yang baik dapat menjadi salah satu faktor yang menjadi penanda majunya suatu negara. Di era perkembangan teknologi saat ini dibutuhkan sumber daya manusia yang berkualitas, dan oleh karena itu salah satu upaya yang harus diutamakan dalam meningkatkan kualitas bangsa adalah dengan meningkatkan kualitas sumber daya manusia melalui pendidikan.

Salah satu upaya dalam menyediakan pendidikan yang merata bagi seluruh lapisan masyarakat adalah dengan memberikan beasiswa pendidikan. Beasiswa adalah dukungan biaya pendidikan yang diberikan kepada mahasiswa untuk

mengikuti dan/atau menyelesaikan Pendidikan Tinggi berdasarkan pertimbangan utama prestasi dan/atau potensi akademik [1]. Beasiswa diharapkan dapat membantu masyarakat untuk menempuh pendidikan ditengah mahalnnya biaya pendidikan. Karena manfaat dari beasiswa tersebut, banyak masyarakat yang berminat untuk menjadi penerima beasiswa. Penyedia beasiswa pun beragam mulai dari pemerintah, lembaga sosial, swasta, maupun pribadi.

Media sosial banyak digunakan oleh penyalur beasiswa untuk memberikan informasi beasiswa. Media sosial berhasil menyebarkan informasi beasiswa secara cepat dan masif. Namun, media sosial memiliki kekurangan sebagai sarana penyebaran informasi beasiswa.

Informasi beasiswa yang kini tersedia masih terbatas secara akses, dan lingkup penyebarannya. Ruang lingkup dari informasi beasiswa masih sektoral dan sesuai dengan jaringan dari penerima mandat penyebar informasi. Akun khusus penyebaran informasi beasiswa secara gratis masih terbatas pada sosial media Instagram. Selain itu, banyak beasiswa yang kredibel namun dari sisi pemasarannya kurang baik dikarenakan konsep dari poster yang tidak matang, dan bahkan beberapa informasi beasiswa tidak memiliki poster [2].

Kekurangan lain dari media sosial bagi peminat beasiswa adalah tidak dapat melakukan registrasi beasiswa sekaligus pada media sosial. Oleh karena itu diperlukan aplikasi beasiswa terpusat di mana pengguna bisa mendapatkan informasi beasiswa sekaligus melakukan permohonan beasiswa yang diminati. Pengguna dapat sekaligus melakukan monitoring terhadap status permohonan beasiswa mereka pada aplikasi beasiswa.

Teknologi *website* menjadi pilihan ideal karena dapat diakses pada perangkat yang beragam. Namun, penggunaan *website* semata sebagai penerapan aplikasi beasiswa masih belum sepenuhnya menjadi solusi. Kecenderungan mahasiswa dalam mengecek papan pengumuman dan melihat situs web yang sewaktu-waktu atau tidak rutin [3]. Hal tersebut membuat peminat beasiswa berpotensi tertinggal informasi beasiswa yang mereka minati. Oleh karena itu dibutuhkan fitur notifikasi untuk memberikan perhatian kepada peminat beasiswa. Fitur notifikasi tidak tersedia pada *website* biasa dan hanya dijumpai pada aplikasi *native*. Pengembangan dengan aplikasi *native* pada perangkat yang berbeda-beda dapat memakan biaya yang tinggi.

*Progressive web apps* selanjutnya disebut PWA merupakan sebutan bagi aplikasi web yang menerapkan teknologi web

modern. PWA sebenarnya aplikasi web biasa, tapi memanfaatkan fitur perambanan yang mutakhir agar tampil layaknya aplikasi asli [6]. Di antara kemampuan tersebut adalah *push notifications*, dapat dipasang, dapat diakses dalam keadaan offline, dan memiliki interaksi seperti aplikasi *native*.

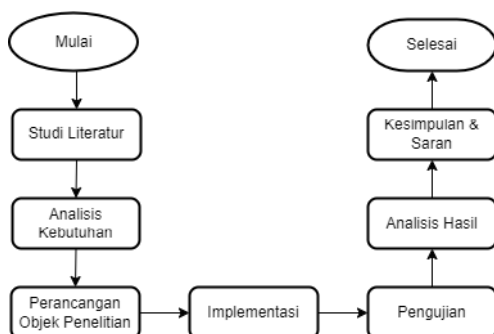
PWA memiliki kemampuan untuk mengirimkan pemberitahuan kepada pengguna pada lingkungan web. Dalam menjalankan kemampuan tersebut PWA membutuhkan Push API dan Notifications API. Keduanya adalah dua API terpisah, tetapi bekerja sama untuk menyediakan fungsionalitas pengiriman notifikasi pada aplikasi. Kedua API tersebut bekerja di luar jendela peramban, yaitu *service worker*, sehingga pembaruan dapat didorong dan pemberitahuan dapat ditampilkan saat halaman aplikasi tidak aktif atau bahkan ditutup [4].

Kemampuan lain yang dimiliki oleh PWA adalah dapat dipasang pada perangkat apapun selama perangkat tersebut mendukung *web browser*. Aplikasi dapat dipasang tanpa harus mengunduh terlebih dahulu dari toko aplikasi bawaan sistem operasi. Kemampuan tersebut membuat aplikasi yang menerapkan PWA memiliki nilai *installability* yang tinggi dibandingkan dengan aplikasi *native* yang hanya dapat dipasang pada sistem operasi tertentu. Hal tersebut berguna bagi pengguna aplikasi karena bisa memberi pengalaman pengguna yang baik dibanding aplikasi web biasa.

Kemampuan PWA yang dapat dipasang pada berbagai sistem operasi memungkinkan untuk mengembangkan aplikasi *multi-platform* dengan menerapkan PWA [7]. Sehingga pada penelitian ini penerapan PWA pada aplikasi beasiswa diharapkan dapat memenuhi kebutuhan fitur *push notifications* pada aplikasi beasiswa dan menjadikan aplikasi beasiswa memiliki kemampuan *installability* yang tinggi agar dapat dipasang pada perangkat yang beragam (*multi-platform*).

## II. METODOLOGI PENELITIAN

Metode yang diterapkan pada penelitian ini adalah *experimental design*. Penelitian ini memiliki tujuan untuk menerapkan PWA pada aplikasi beasiswa untuk mengembangkan fitur *push notification* dan *multi-platform*. Untuk mencapai tujuan tersebut maka terdapat beberapa tahapan antara lain:



Gbr. 1 Alur Penelitian

### A. Studi Literatur

Tahapan ini dilakukan untuk mengumpulkan dan mempelajari literatur yang berkaitan dengan penerapan *progressive web apps* untuk pengembangan fitur *push notification* dan *multi-platform installable*. Literatur yang sudah dipelajari tersebut diharap dapat memudahkan proses penelitian selanjutnya. Adapun pustaka studi literatur yang dipelajari dalam pelaksanaan penelitian ini adalah literatur yang berkaitan dengan topik sebagai berikut:

1. Aplikasi Web
2. Progressive Web Apps
3. Service Worker
4. Push Notification
5. Multi-platform Apps
6. Firebase Cloud Messaging
7. Caching
8. API
9. Javascript Event

### B. Analisis Kebutuhan

Penelitian yang dilakukan adalah penelitian mengenai penerapan *progressive web apps* untuk pengembangan fitur *push notification* dan *multi-platform installable* pada aplikasi beasiswa. Agar penelitian ini berjalan sesuai dengan perencanaan, maka dibutuhkan sebuah sistem yang dapat mendukung penelitian ini. Adapun kebutuhan perangkat yang digunakan adalah sebagai berikut:

#### 1. Perangkat Keras

Pada penelitian ini perangkat keras digunakan sebagai perangkat untuk pembuatan aplikasi dan juga untuk melakukan pengujian aplikasi. Adapun spesifikasi perangkat keras berupa laptop adalah sebagai berikut:

- |                |                                 |
|----------------|---------------------------------|
| A. Processor   | : Intel Core i5 1165G7 4.20 GHz |
| B. VGA         | : Nvidia mx450 2GB              |
| C. Memori      | : 16GB DDR4                     |
| D. Penyimpanan | : 256GB SSD                     |

#### 2. Perangkat Lunak

Adapun perangkat lunak yang diperlukan antara lain:

- A. Sistem operasi Windows 11 64-bit.
- B. Visual Studio Code
- C. Web Browser Google Chrome
- D. Microsoft Office
- E. Bahasa Pemrograman Javascript

Selain kebutuhan perangkat, diperlukan juga analisis kebutuhan daripada aplikasi beasiswa yang akan dikembangkan. Kebutuhan tersebut mencakup kebutuhan fungsional dan kebutuhan non-fungsional. Aplikasi akan dikembangkan sesuai dengan kebutuhan tersebut. Adapun kebutuhan fungsional aplikasi sebagai berikut:

1. Sistem dapat digunakan untuk menambah informasi beasiswa.
2. Sistem dapat digunakan untuk melakukan pendaftaran beasiswa.
3. Sistem dapat digunakan untuk mengelola informasi beasiswa.

4. Sistem dapat menampilkan informasi beasiswa yang tersedia.
5. Sistem dapat digunakan untuk mengelola pendaftaran beasiswa.

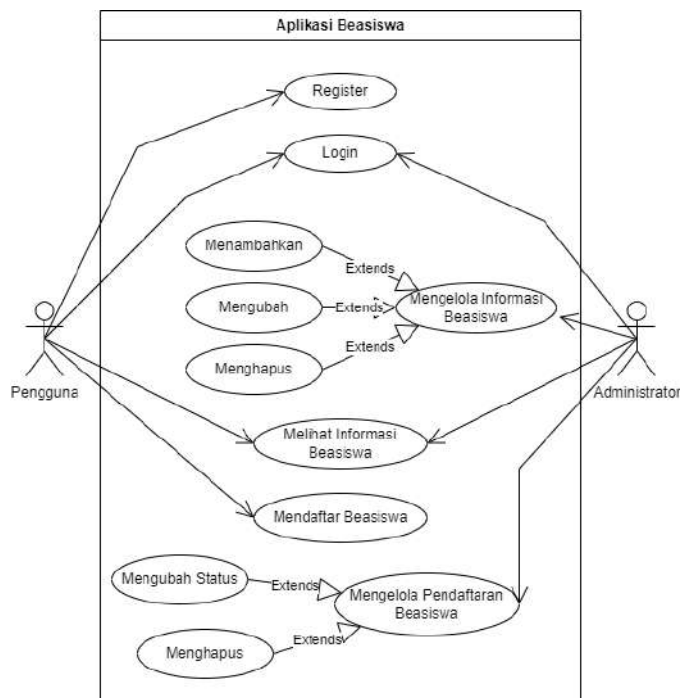
Sedangkan kebutuhan non-fungsional dari aplikasi beasiswa sebagai berikut:

1. Sistem dapat diakses oleh pengguna selama 24 jam dalam sehari sehingga dapat diakses kapan pun jika diperlukan.
2. Sistem tidak mengandung konten sensitif yang ditampilkan kepada pengguna maupun administrator.
3. Sistem yang sudah dipasang pada perangkat dapat diperbarui tanpa melakukan pemasangan ulang.
4. Sistem mudah digunakan dan intuitif bagi pengguna, dengan antarmuka pengguna yang jelas, dan navigasi yang mudah dipahami.

### C. Perancangan Objek Penelitian

Objek pada penelitian ini adalah aplikasi beasiswa. Sehingga diperlukan perancangan dan pengembangan aplikasi beasiswa sebelum PWA diimplementasikan. Aplikasi beasiswa memiliki tujuan untuk membagikan informasi beasiswa dan juga sebagai media pendaftaran beasiswa.

Terdapat dua aktor pada aplikasi ini yang mana masing-masing aktor memiliki peran yang berbeda. Gambar 2 merupakan use case diagram aplikasi beasiswa yang menggambarkan kebutuhan fungsional aplikasi beserta dengan aktor yang terlibat pada aksi tersebut.

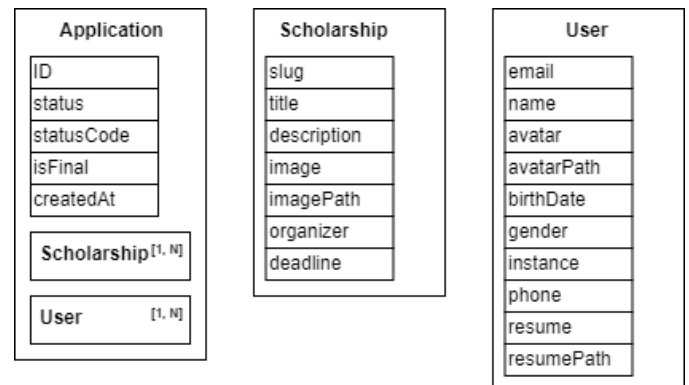


Gbr. 2 Use Case Diagram Aplikasi Beasiswa

Selain merancang aplikasi beasiswa sebagai objek penelitian, diperlukan juga perancangan basis data yang akan digunakan pada aplikasi. Database Management System pada

aplikasi ini menggunakan layanan dari Firebase yaitu Firebase Firestore. Firebase Firestore termasuk ke dalam jenis basis data NoSQL.

Gambar 3 merupakan rancangan basis data aplikasi beasiswa yang berjenis NoSQL. Setiap entitas direpresentasikan sebagai koleksi data dan setiap koleksi data dapat memiliki beberapa dokumen. Karena tidak memiliki relasi, maka jika terdapat dokumen pada salah satu koleksi yang membutuhkan data dokumen di koleksi lain, maka data dokumen pada koleksi lainnya akan disimpan di dokumen tersebut. Pada basis data aplikasi beasiswa, terdapat 3 skema dokumen yaitu dokumen permohonan, beasiswa, dan pengguna. Terdapat data beasiswa dan pengguna yang disimpan pada dokumen permohonan.



Gbr 3. Skema Dokumen Basis Data Aplikasi Beasiswa

### D. Implementasi

Aplikasi beasiswa yang dibangun pada tahap perancangan objek penelitian masih belum menerapkan PWA. Oleh karena itu pada tahap implementasi, PWA diterapkan pada aplikasi beasiswa untuk pengembangan fitur *push notification* dan *multi-platform* installable. *Push notification* pada aplikasi beasiswa akan dikirimkan jika terdapat aksi pemicu. Aksi pemicu tersebut adalah terdapat informasi beasiswa baru yang dikirimkan oleh administrator dan perubahan status permohonan.

Pengembangan fitur *push notification* dan *multi-platform* installable masing-masing memerlukan proses pengembangan yang berbeda. Alur pengembangan fitur *push notification* antara lain:

1. Mendapatkan perizinan dari pengguna untuk mengirimkan notifikasi melalui subscribe.
2. Mendapatkan identifier perangkat pengguna berupa token.
3. Mengirimkan token kepada push server untuk dilakukan subscribe terhadap topik tertentu. Token akan disimpan oleh push server.
4. Ketika terdapat aksi yang memicu notifikasi, maka aplikasi sisi client akan mengirimkan request pengiriman *push notification* kepada push server.
5. Push server akan mengirimkan *push notification* kepada seluruh perangkat yang telah melakukan subscribe.

6. Membuat *service worker* pada aplikasi sisi client untuk menangani *event push*. *Service worker* akan bereaksi dengan memunculkan notifikasi.

Pengembangan fitur multi-platform installable pada aplikasi beasiswa memungkinkan aplikasi dapat dipasang pada berbagai sistem operasi. Dengan PWA, pengembangan fitur *multi-platform installable* dapat dilakukan dengan kode sumber tunggal. Tidak diperlukan pengembangan aplikasi secara spesifik pada sistem operasi berbeda secara berulang-ulang. Aplikasi dapat dikembangkan satu kali dan dipasang pada seluruh sistem operasi yang mendukung web browser. Pengguna dapat memasang aplikasi langsung dari web browser. Jika terdapat pembaruan pada aplikasi, maka pembaruan tersebut akan terpasang secara otomatis ketika pengguna membuka aplikasi. Adapun beberapa tahapan dibawah ini perlu dilakukan untuk pengembangan fitur *multi-platform installable*:

1. Membuat *service worker* yang bertugas untuk menyimpan kerangka utama aplikasi pada memori cache.
2. Melakukan penyesuaian tampilan dan fungsionalitas aplikasi agar aplikasi dapat bekerja dengan baik pada berbagai jenis perangkat dan sistem operasi.
3. Membuat berkas manifest.json yang berisi metadata aplikasi. Berkas tersebut berguna agar perangkat yang memasang aplikasi mengetahui tindakan tepat yang harus dilakukan terhadap aplikasi.
4. Memastikan bahwa seluruh transaksi data yang berlangsung pada aplikasi berjalan pada protokol HTTPS.

#### E. Pengujian

Pengujian perangkat lunak penting dilakukan guna melakukan penilaian kualitas perangkat lunak supaya fungsionalitas aplikasi bekerja sesuai dengan harapan [5]. Pada tahapan ini dilakukan pengujian terhadap fitur *push notification* dan *multi-platform installable* yang telah dikembangkan pada aplikasi beasiswa. Pengujian yang diterapkan adalah pengujian *blackbox*.

Selain pengujian *blackbox*, dilakukan juga pengujian performa pengiriman notifikasi. Pengujian dilakukan untuk mengetahui bagaimana performa dari pengiriman notifikasi pada perangkat pengguna. Pengujian dilakukan pada berbagai perangkat dengan sistem operasi dan versi yang bervariasi. Pengujian dilakukan dengan menghitung durasi waktu yang dibutuhkan oleh server dalam mengirim notifikasi ke perangkat, dan waktu yang dibutuhkan oleh perangkat untuk menampilkan notifikasi.

#### F. Analisis Hasil

Analisis hasil dilakukan berdasarkan hasil pengujian fitur pada aplikasi beasiswa. Tujuan dilakukannya analisa adalah untuk mengetahui aspek-aspek yang mempengaruhi hasil penelitian. Pada tahap ini juga akan dinilai terkait kelebihan dan kekurangan terkait penerapan PWA untuk pengembangan fitur *push notification* dan *multi-platform installable* pada aplikasi beasiswa.

#### G. Kesimpulan dan Saran

Pada tahap ini penulis akan mempertimbangkan tahapan-tahapan sebelumnya dan hasil yang diperoleh pada pengujian dan analisis hasil. Jawaban dari rumusan masalah juga akan dijelaskan pada tahap ini. Selain itu, penulis juga memberikan saran-saran yang dapat diterapkan pada penelitian selanjutnya.

### III. HASIL DAN PEMBAHASAN

#### A. Pengembangan Aplikasi Beasiswa

Proses pengembangan aplikasi diawali dengan inisiasi proyek. Aplikasi dikembangkan dengan library Reactjs. Reactjs telah menyediakan sebuah pustaka yang dapat memudahkan pengembang dalam mengembangkan aplikasi beasiswa. Perintah yang digunakan pada terminal untuk menginisiasi proyek Reactjs ditunjukkan pada gambar 4.



Gbr. 4 Inisialisasi Proyek Aplikasi Beasiswa

Proses perancangan aplikasi web dilanjutkan dengan menghubungkan aplikasi web dengan database. Peneliti menggunakan layanan Cloud Firestore milik firebase sebagai layanan database pada penelitian ini. Diperlukan berkas bernama `firebase.js` yang digunakan untuk melakukan koneksi terhadap layanan firebase seperti `firestore`, `messaging`, `authentication`, `storage`, dll. Koneksi dilakukan dengan tujuan agar aplikasi dapat menggunakan layanan firebase tersebut. Cloud Firestore secara bawaan sudah tersimpan di awan sehingga tidak diperlukan lagi proses hosting database.

Gambar 5 merupakan baris kode pada berkas `firebase.js`. Diperlukan pemasangan library firebase untuk menghubungkan aplikasi dengan layanan firebase. Variabel `firebaseConfig` menyimpan identitas proyek firebase yang telah dibuat sebelumnya. Fungsi `initializeApp` milik library firebase merupakan fungsi yang digunakan untuk menginisialisasi objek firebase pada aplikasi. Setelah proses inisialisasi tersebut, maka layanan firebase sudah dapat digunakan. Contohnya adalah pada baris selanjutnya fungsi `getAuth` digunakan untuk menggunakan layanan autentikasi firebase pada aplikasi. Fungsi `getFirestore` digunakan untuk menggunakan layanan basis data `firestore` pada aplikasi, dan seterusnya.

```
import { initializeApp } from "firebase/app";
import { getAuth } from "firebase/auth";
import { getFirestore } from "firebase/firestore";
import { getStorage } from "firebase/storage";

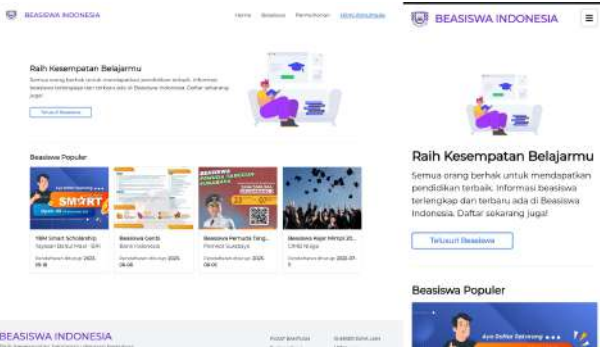
const firebaseConfig = {
  apiKey: "AIzaSyxxqk-xxKfpmARWpt55lUZGc",
  authDomain: "beasiswa-indonesia.firebaseio.com",
  projectId: "beasiswa-indonesia",
  storageBucket: "beasiswa-indonesia.appspot.com",
  appId: "1:235318415040:web:xxbc056xx25"
};

const firebaseApp = initializeApp(firebaseConfig);
const auth = getAuth(firebaseApp);
const db = getFirestore(firebaseApp);
const storage = getStorage(firebaseApp);

export {
  auth,
  db,
  storage,
}
```

Gbr. 5 Koneksi Layanan Firebase

Tahapan pengembangan dilanjutkan dengan membuat komponen tampilan aplikasi yang diperlukan. Komponen dibuat sesuai dengan kebutuhan pada antarmuka aplikasi. Kumpulan komponen yang disatukan kemudian membentuk halaman web yang utuh. Antarmuka dan fungsionalitas aplikasi yang selesai dibuat menghasilkan aplikasi utuh yang terdiri dari beberapa halaman. Halaman pada aplikasi memuat informasi dan tugas yang berbeda-beda. Gambar 6 adalah halaman utama aplikasi beasiswa yang selesai dibuat pada perangkat desktop dan mobile.



Gbr. 6 Halaman Utama Aplikasi Beasiswa

## B. Implementasi

Tahap pertama dari penerapan PWA adalah pendaftaran berkas *service worker*. *Service worker* didaftarkan saat aplikasi pertama kali dijalankan pada browser. Sehingga baris kode untuk pendaftaran *service worker* disimpan pada berkas *index.js*. Berkas *service worker* yang ditulis pada gambar 7 memiliki tugas untuk menyimpan kerangka utama aplikasi

(*app-shell*) ke dalam memori cache. Proses penyimpanan tersebut bertujuan agar *app-shell* dapat dibuka kembali tanpa koneksi internet pada pengaksesan yang kedua kalinya.

```
import { clientsClaim } from 'workbox-core';
import { precacheAndRoute,
  createHandlerBoundToURL } from 'workbox-precaching';
import { registerRoute } from 'workbox-routing';

precacheAndRoute(self.__WB_MANIFEST);

registerRoute(
  ({ request, url }) => {
    if (request.mode !== 'navigate') {
      return false;
    }
    if (url.pathname.startsWith('/_')) {
      return false;
    }
    return true;
  },
  createHandlerBoundToURL(url + '/index.html')
);
```

Gbr. 7 Berkas Service Worker

Setelah *service worker* berhasil didaftarkan, tahapan selanjutnya adalah mendapatkan perizinan pengguna untuk mengirimkan notifikasi. Gambar 8 merupakan fungsi yang digunakan untuk meminta perizinan pengguna. Jika pengguna setuju untuk mengizinkan aplikasi mengirim notifikasi, maka aplikasi akan mendapatkan token dari firebase. Token tersebut merupakan *identifier* dari perangkat pengguna dan akan dikirim kepada push server untuk proses *subscribing*. Proses *subscribing* bertujuan agar perangkat nantinya akan menerima *push notification* dari *push server*. Hal tersebut terjadi karena hanya perangkat dengan token terdaftar yang akan menerima *push notification*.

```
function requestPermission() {
  Notification.requestPermission().then(async
  (permission) => {
    if (permission === 'granted') {
      const messaging = getMessaging();
      const token = await getToken(messaging, {
        vapidKey: 'xxxxxxxx'
      })
      axios.post('https://url/subscribe', {
        token,
        topic: 'beasiswa-indonesia'
      }, {
        headers: {
          'Content-Type': 'application/json'
        }
      })
      return token;
    }
  });
}
```

Gbr. 8 Fungsi Permintaan Izin Pengiriman Notifikasi



```

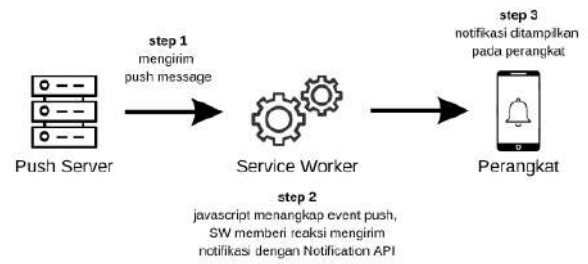
app.post("/subscribe", async (req, res) => {
  const { token, topic } = req.body;
  await
  firestore.collection("subscription").add({
    token, topic
  })
  messaging.subscribeToTopic([token], topic)
  .then((response) => {
    res.json({
      message: "subscribe berhasil",
      status: 200
    })
  })
  .catch((error) => {
    res.json({
      message: "subscribe gagal",
      status: 500
    })
  });
});

app.post("/send", (req, res) => {
  const message = {
    webpush: {
      notification: {
        title: req.body.title,
        body: req.body.body,
        image: req.body.imageUrl,
      },
    },
    topic: req.body.topic
  };
  messaging.send(message)
  .then((response) => {
    res.json({
      message: "push notif terkirim",
      status: 200
    })
  })
  .catch((error) => {
    res.json({
      message: "push notif tidak terkirim",
      status: 500
    })
  });
});

```

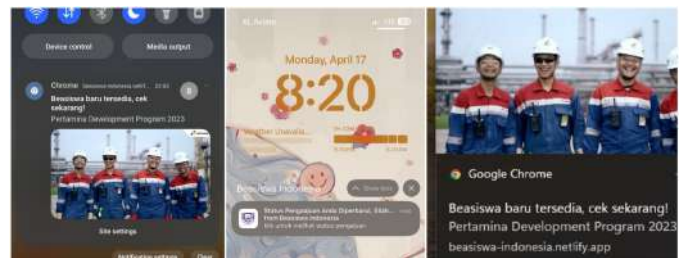
Gbr. 9 API pada Aplikasi Sisi Server

Proses *subscribing* dan *request* pengiriman *push notification* hanya tersedia melalui pustaka *firebase admin SDK* yang tersedia pada aplikasi sisi server. Oleh karena itu diperlukan sebuah API pada aplikasi sisi server untuk menangani *request subscribe* dan pengiriman *push notification* dari aplikasi sisi client. Gambar 9 merupakan API pada aplikasi sisi server untuk menangani *request* tersebut. Melalui API tersebut, aplikasi sisi client dapat melakukan proses *subscribe* dan pengiriman *push notification*.



Gbr. 10 Alur Service Worker Menangani Event Push

*Push notification* yang dikirim oleh *push server* akan memicu javascript untuk membangkitkan *event push*. Event tersebut akan ditangkap oleh *service worker* dan *service worker* akan bereaksi dengan menampilkan notifikasi kepada perangkat yang telah melakukan *subscribe*. Gambar 10 merupakan alur *service worker* dalam menangani event push.



Gbr. 11 Notifikasi Terkirim pada Berbagai Perangkat

Gambar 11 merupakan tampilan notifikasi pada perangkat Android, iOS, dan windows. Notifikasi tersebut dipicu oleh aksi penambahan informasi beasiswa baru oleh administrator. Perangkat yang telah melakukan *subscribe* akan menerima notifikasi tersebut.

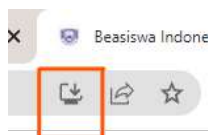
Pengembangan fitur *multi-platform* installable diawali dengan membuat berkas *manifest.json*. Berkas tersebut berguna untuk memberikan informasi yang diperlukan oleh browser dan perangkat lunak untuk mengenali aplikasi web sebagai PWA. Berkas tersebut berisi metadata seperti nama aplikasi, deskripsi, ikon, tema warna, dan pengaturan layar awal. Metadata ini digunakan oleh browser ataupun sistem operasi untuk menampilkan ikon aplikasi pada layar perangkat pengguna. Gambar 12 merupakan berkas *manifest.json* aplikasi beasiswa.

```

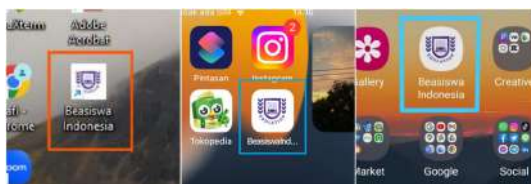
{
  "theme_color": "#3b82f6",
  "background_color": "#3b82f6",
  "display": "standalone",
  "scope": "/",
  "start_url": "/",
  "name": "Beasiswa Indonesia",
  "short_name": "Beasiswa Indonesia",
  "description": "Aplikasi beasiswa untuk seluruh masyarakat Indonesia",
  "icons": [
    {
      "src": "/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/icon-256x256.png",
      "sizes": "256x256",
      "type": "image/png"
    },
    {
      "src": "/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
    
```

Gbr. 12 Berkas manifest.json Aplikasi Beasiswa

Pendaftaran manifest.json yang berhasil ditandai dengan munculnya ikon unduh aplikasi pada kolom URL browser yang ditunjukkan oleh gambar 13. Manifest.json merupakan syarat utama agar aplikasi dapat dipasang pada perangkat. Jika berkas manifest.json tidak ditemukan, maka aplikasi tidak akan dikenali sebagai aplikasi web yang dapat dipasang pada perangkat pengguna. Manifest.json berguna agar perangkat dapat menentukan perilaku terhadap aplikasi dengan baik. Gambar 14 merupakan ikon aplikasi beasiswa yang telah terpasang pada perangkat pengguna.



Gbr. 13 Ikon Unduh Aplikasi Pada Web Browser



Gbr. 14 Ikon Aplikasi pada Layar Utama Perangkat Windows, iOS, dan Android

C. Hasil Pengujian

Berikut merupakan hasil pengujian pada fitur *push notification* dan *multi-platform installable* menggunakan metode pengujian *blackbox* dan pengujian performa pengiriman notifikasi. Pengujian dilakukan pada perangkat

dengan sistem operasi Android, iOS, dan Windows yang telah memasang aplikasi beasiswa.

TABEL I  
HASIL PENGUJIAN BLACKBOX FITUR PUSH NOTIFICATION

| No | Skenario Pengujian  | Test Case  | Hasil yang Diharapkan   | Status  |
|----|---|--|---|---|
| 1  | Menguji pengiriman notifikasi pada perangkat saat aplikasi dalam keadaan aktif.                             | Membuka aplikasi dan membiarkan aplikasi dalam keadaan aktif.<br>Membuat beasiswa baru.                  | Perangkat yang aplikasinya dalam keadaan aktif akan menerima notifikasi informasi beasiswa baru.                | iOS: Berhasil<br>Android: Berhasil<br>Windows: Berhasil |
| 2  | Menguji pengiriman notifikasi pada perangkat saat aplikasi dalam keadaan tidak aktif.                       | Membiarkan aplikasi dalam keadaan tidak aktif.<br>Membuat beasiswa baru.                                 | Perangkat pengguna yang aplikasinya dalam keadaan tidak aktif akan menerima notifikasi informasi beasiswa baru. | iOS: Berhasil<br>Android: Berhasil<br>Windows: Berhasil |
| 3  | Menguji pengiriman notifikasi pada perangkat yang belum melakukan subscribe.                                | Membiarkan perangkat belum melakukan subscribe.<br>Membuat beasiswa baru.                                | Notifikasi tidak muncul karena perangkat belum melakukan subscribe.   | iOS: Berhasil<br>Android: Berhasil<br>Windows: Berhasil |
| 4  | Menguji pengiriman notifikasi pada perangkat yang sudah melakukan subscribe.                                | Melakukan subscribe perangkat pada aplikasi.<br>Membuat beasiswa baru.                                   | Notifikasi muncul karena perangkat sudah melakukan subscribe.   | iOS: Berhasil<br>Android: Berhasil<br>Windows: Berhasil |
| 5  | Menguji pengiriman notifikasi terkait pembaruan status permohonan pada perangkat yang bukan milik pengguna. | Melakukan pembaruan status permohonan pada akun yang tidak melakukan subscribe pada perangkat pengujian. | Perangkat yang bukan milik pengguna tidak mendapat notifikasi pembaruan status permohonan.                      | iOS: Berhasil<br>Android: Berhasil<br>Windows: Berhasil |
| 6  | Menguji pengiriman notifikasi terkait   | Mengubah status dari salah satu permohonan   | Perangkat pengguna akan menerima  | iOS: Berhasil<br>Android:                               |

|   |   |   |  |   |
|---|---|---|--|---|
|   | pembaruan status permohonan pada perangkat milik pengguna.            | beasiswa.   | notifikasi perubahan status permohonan beasiswa.   | Berhasil<br><br>Windows:<br>Berhasil                                  |
| 7 | Mengirim notifikasi dengan body berisi teks dan poster berupa gambar. | Membuat informasi beasiswa baru dengan poster gambar oleh administrator | Perangkat pengguna akan menerima notifikasi informasi beasiswa baru beserta dengan gambar poster beasiswa. | iOS:<br>Gagal<br><br>Android:<br>Berhasil<br><br>Windows:<br>Berhasil |

Terdapat 5 skenario positif dan 2 skenario negatif pada pengujian *blackbox* fitur *push notification*. Hasil pengujian pada tabel 1 menunjukkan bahwa aplikasi yang dipasang pada perangkat dengan sistem operasi Android dan Windows berhasil lolos pada seluruh skenario pengujian fitur *push notification*. Sedangkan aplikasi yang dipasang pada sistem operasi iOS gagal pada skenario pengiriman notifikasi dengan gambar dan berhasil pada skenario pengujian lainnya.

TABEL III  
HASIL PENGUJIAN PERFORMA PENGIRIMAN NOTIFIKASI

|                                  |            | Body Notifikasi               |                     |                     |                     |  |
|----------------------------------|------------|-------------------------------|---------------------|---------------------|---------------------|--|
|                                  |            | Oppo A39 (Android 5.1)        |                     |                     |                     |  |
|                                  | T          | T + G<br>142<br>kb            | T + G<br>1106<br>kb | T + G<br>2431<br>kb | T + G<br>3104<br>kb |  |
| Server ke Perangkat              | 1,65 detik | 1,40 detik                    | 1,43 detik          | 1,39 detik          | 1,54 detik          |  |
| Perangkat Menampilkan Notifikasi | 1,40 detik | 2,63 detik                    | 4,32 detik          | 7,11 detik          | 7,66 detik          |  |
|                                  |            | Samsung A53 (Android 13)      |                     |                     |                     |  |
| Server ke Perangkat              | 1,27 detik | 1,44 detik                    | 1,44 detik          | 1,41 detik          | 1,41 detik          |  |
| Perangkat Menampilkan Notifikasi | 2,02 detik | 3,00 detik                    | 5,38 detik          | 6,59 detik          | 11,36 detik         |  |
|                                  |            | Lenovo Ideapad (Windows 10)   |                     |                     |                     |  |
| Server ke Perangkat              | 1,35 detik | 1,63 detik                    | 1,54 detik          | 1,45 detik          | 1,41 detik          |  |
| Perangkat Menampilkan Notifikasi | 1,32 detik | 2,50 detik                    | 4,72 detik          | 10,16 detik         | 12,25 detik         |  |
|                                  |            | Lenovo Thinkbook (Windows 11) |                     |                     |                     |  |

| Server ke Perangkat              | 1,67 detik | 1,44 detik | 1,44 detik | 1,41 detik | 1,41 detik |
|----------------------------------|------------|------------|------------|------------|------------|
| Perangkat Menampilkan Notifikasi | 1,05 detik | 1,84 detik | 3,22 detik | 3,55 detik | 5,69 detik |
| iPhone 12 Mini (ios 16.4)        |            |            |            |            |            |
| Server ke Perangkat              | 1,84 detik | 1,44 detik | 1,51 detik | 1,42 detik | 1,79 detik |
| Perangkat Menampilkan Notifikasi | 2,87 detik | 2,83 detik | 2,88 detik | 3,22 detik | 3,55 detik |

Keterangan:

- T = teks
- G = gambar

Pengujian performa notifikasi menunjukkan bahwa durasi server dalam mengirim notifikasi kepada perangkat berada pada kisaran 1 detik hingga 2 detik. Pengiriman notifikasi dari server ke perangkat cenderung stabil dalam semua kondisi bahkan pada pengiriman gambar berukuran besar sekalipun. Hal tersebut dikarenakan data yang dikirimkan server kepada perangkat hanya berupa teks. Termasuk gambar yang nantinya ditampilkan pada notifikasi hanya dikirim dalam bentuk url.

Durasi perangkat dalam menampilkan notifikasi adalah dinamis. Bergantung terhadap ukuran data pada body notifikasi yang harus diunduh oleh perangkat terlebih dahulu sebelum menampilkan notifikasi. Termasuk ke dalam data yang harus diunduh adalah url gambar yang dikirimkan oleh server. Semakin besar ukuran data yang harus diunduh, maka semakin lama durasi yang dibutuhkan oleh perangkat dalam menampilkan notifikasi.

#### IV. KESIMPULAN

Aplikasi beasiswa yang dibangun dengan menerapkan PWA dapat mengakomodasi kebutuhan fitur yang didukung oleh aplikasi native. Peran aplikasi beasiswa yaitu membagikan informasi beasiswa dan sebagai platform pendaftaran beasiswa dapat ditunjang dengan baik.

Hasil pengujian menunjukkan bahwa fitur *push notification* dan *multi-platform installabile* berhasil dikembangkan dengan menerapkan PWA. Perangkat dengan sistem operasi windows, android, dan iOS berhasil melalui pengujian *blackbox* untuk menguji fitur *push notification* dan *multi-platform installabile*. Perangkat windows dan android berhasil lolos pada 100% skenario pengujian *blackbox*. Sedangkan perangkat iOS gagal pada salah satu skenario pengujian yaitu pengiriman notifikasi dengan gambar. Hal tersebut terjadi dikarenakan sistem operasi iOS hingga saat ini masih belum mendukung pengiriman web *push notification* dengan penyematan gambar.

Pada pengujian performa pengiriman notifikasi, didapatkan temuan bahwa durasi pengiriman notifikasi dari server menuju perangkat cenderung stabil di antara 1 sampai 2 detik. Sedangkan durasi perangkat dalam menampilkan notifikasi adalah dinamis. Semakin besar data pada body notifikasi, maka semakin lama juga durasi perangkat dalam



menampilkan notifikasi. Web browser juga berpengaruh pada durasi perangkat dalam menampilkan notifikasi. Hal tersebut dikarenakan perangkat perlu mengunduh data tersebut terlebih dahulu sebelum menampilkan notifikasi. Sedangkan pengiriman notifikasi dari server menuju perangkat cenderung stabil karena data yang memiliki ukuran besar dikirimkan dalam bentuk url yang nantinya diunduh oleh perangkat ketika push message berhasil dikirimkan.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan YME atas segala berkah, rahmat, dan hidayah, sehingga penulis dapat menyelesaikan artikel ini. Penulis juga berterima kasih kepada orang tua dan saudara yang senantiasa memberi dukungan secara penuh. Terima kasih kepada Dosen Pembimbing dan Dosen penguji yang banyak memberi masukan dan motivasi sehingga penelitian ini menjadi lebih baik lagi.

#### REFERENSI

- [1] Ramadhon, Raka, Riswan Jaenudin, and Siti Fatimah. 2018. "Pengaruh beasiswa terhadap motivasi belajar mahasiswa pendidikan ekonomi Universitas Sriwijaya." *Jurnal PROFIT: Kajian Pendidikan Ekonomi dan Ilmu Ekonomi* 4.2 : 203-213.
- [2] Hadini, Bella Citra, and Baroto Tavip Indrojarwo. 2021. "Perancangan Kampanye Media Sosial Instagram 'Youthoffers' sebagai Channel Informasi Beasiswa Pendidikan Mahasiswa Indonesia." *Jurnal Sains dan Seni ITS* 9.2. F191-F196.
- [3] Sulistiyasni, Sulistiyasni, Joko Purnomo, and Chandra Bagas Pambudi. 2020. "Aplikasi Mobile dengan Fitur Push Notification Berbasis Android sebagai Media Informasi Pengumuman dan Kegiatan Mahasiswa di Stmik Widya Utama." *Teknikom: Teknologi Informasi, Ilmu Komputer dan Manajemen* 4.1. 9-16.
- [4] Mozilla Developer. 2022. How to make PWAs re-engageable using Notifications and Push. [Daring]. Diakses pada 28 Januari 2023. Tersedia: [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps/Re-engageable\\_Notifications\\_Push](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Re-engageable_Notifications_Push).
- [5] Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti, Y. 2019. Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *J. Inform. Univ. Pamulang*, 4(4), 143.
- [6] Kurniawan, A., Areni, I. S., & Achmad, A. 2017. Implementasi Progressive Web Application pada Sistem Monitoring Keluhan Sampah Kota Makassar. *Jurnal JPE*, 21(02), 34–38.
- [7] Mole, P. V., & Mole, P. V. 2020. Progressive Web Apps: A Novel Way for Cross-Platform Development. no. September.