

Monitoring Sistem Keluhan Mahasiswa Menggunakan Progressive Web Application Pada Politeknik Negeri Lhokseumawe

Aiyatul Mu'in¹, Muhammad Nasir^{2*}, Amri³, Anwar⁴

^{1,2,3,4}Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe Jln.
B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹Bocahsukes43@gmail.com

²Muhnasir.tmj@pnl.ac.id

³Amriaceh72@gmail.com

⁴Anwar551@yahoo.com

Abstrak— Perkembangan teknologi informasi saat ini sangat pesat. Kebutuhan manusia dalam sumber data sangatlah penting. Salah satu teknologi *web* yang hadir saat ini adalah *progressive web application*. Penelitian ini bertujuan untuk merancang sebuah sistem yang dapat mempermudah mahasiswa ketika ingin menyampaikan keluhan fasilitas ruangan atau Laboratorium. Biasanya keluhan dapat disampaikan melalui sebuah kotak saran dan *kuisisioner* yang hanya terbatas beberapa pertanyaan saja. Pelayanan dalam dunia pendidikan merupakan unsur penting untuk terciptanya suasana akademik yang kondusif untuk terlaksananya proses belajar mengajar di jurusan Teknologi Informasi Dan Komputer. Oleh karena itu dibutuhkan sebuah aplikasi untuk mempermudah mahasiswa menyampaikan keluhan tertentu. Berdasarkan diatas, penelitian ini akan membangun sebuah aplikasi *Monitoring system* keluhan mahasiswa menggunakan *PWA* pada jurusan Teknologi Informasi dan Komputer. Hasil dari penelitian ini adalah sistem keluhan mahasiswa dapat mengadukan keluhan tidak hanya terbatas pada pertanyaan dan kotak saran melainkan dengan meng *upload file*, gambar, *video*, *audio* beserta deskripsi. Data keluhan yang bersumber dari sebuah *API (Application Programming Interface)* kemudian ditampilkan, Selama dalam keadaan jaringan aktif. *Service worker* melakukan proses *caching* sehingga data yang sudah disimpan bisa diakses pada jaringan tidak aktif. *Service worker* mampu melakukan proses *caching data*. Meskipun eksekusi waktu yang dibutuhkan dalam mengakses aplikasi lebih lama karena pemasangan *service worker*, namun aplikasi yang diakses lebih cepat ketika dalam keadaan *offline* karena data dimuat dalam *cache service worker*. Nilai performa 72/100 dan aspek – aspek *PWA* pada aplikasi keluhan mahasiswa sangat baik dari 10 kriteria pengujian dengan hasil persentasi nilai 91/100.

Kata kunci - *Progressive Web Application, RESTful API, Service Worker, Cache, Laravel, React Js.*

Abstract— The development of information technology is currently very fast. Human needs in data sources are very important. One of the current web technologies is a progressive web application. This study aims to design a system that can make it easier for students when they want to submit complaints about room or laboratory facilities. Usually complaints can be submitted via a suggestion box and a questionnaire which is only limited to a few questions. Service in the world of education is an important element for creating an academic atmosphere that is conducive to the implementation of the teaching and learning process in the Department of Information and Computer Technology. Therefore an application is needed to make it easier for students to submit certain complaints. Based on the above, this study will build a student complaint monitoring system application using PWA in the Information Technology and Computer department. The result of this research is that the student complaint system can file complaints not only limited to questions and suggestion boxes but by uploading files, images, videos, audio along with descriptions. Complaint data that comes from an API (Application Programming Interface) is then displayed, as long as the network is active. The service worker performs a caching process so that the saved data can be accessed on an inactive network. Service workers are capable of caching data. Although the execution time required to access the application is longer due to the installation of the service worker, the application is accessed faster when it is offline because the data is loaded in the service worker cache. The 72/100 performance score and the PWA aspects of the student complaint application are very good out of the 10 testing criteria with the percentage score of 91/100.

Keywords - *Progressive Web Application, RESTful API, Service Worker, Cache, Laravel, React Js.*

I. PENDAHULUAN

Mahasiswa masih mengalami kesulitan ketika ingin menyampaikan keluhan terkait fasilitas di jurusan Teknologi Informasi Dan Komputer kerusakan ruangan atau Laboratorium. Biasanya keluhan dapat disampaikan melalui sebuah kotak saran dan *kuisisioner* yang hanya terbatas pada beberapa pertanyaan saja.

Pelayanan dalam dunia pendidikan merupakan unsur penting

untuk terciptanya suasana akademik yang kondusif demi kesuksesan proses belajar mengajar pada jurusan Teknologi Informasi Dan Komputer. Oleh karena itu dibutuhkan sebuah aplikasi yang dapat mempermudah mahasiswa untuk menyampaikan keluhan tersebut.

Berdasarkan permasalahan diatas penelitian ini akan membangun sebuah aplikasi *monitoring system* keluhan mahasiswa menggunakan *PWA* di jurusan Teknologi Informasi Dan Komputer. Mahasiswa dapat mengadukan keluhan tidak

hanya terbatas pada pertanyaan dan kotak saran melainkan dengan meng *upload* gambar, *video*, *audio* beserta *text* secara langsung. Mahasiswa dapat melihat sejauh mana tindak lanjut keluhan oleh pihak *staff*, selain itu mahasiswa dapat melihat juga keluhan yang disampaikan tersebut sudah dibaca atau belum oleh pihak *staff*. Sistem yang akan dirancang menggunakan *Progressive Web App (PWA)* yang memudahkan mahasiswa dalam mengakses sistem. *PWA* sudah menggunakan *service worker* dengan cukup baik, teknologi *service worker* memiliki kemampuan untuk dapat mengontrol *asset* yang akan di *cache*. Teknologi ini juga menyediakan *custome* permintaan jaringan sehingga dapat memberikan layanan meski dalam keadaan *offline*. Namun untuk dapat menggunakan sistem ini tetap harus mengakses website terlebih dahulu. Setelah program *cache* terbaca maka selanjutnya tetap dapat menikmati layanan meski dalam keadaan *offline* sekalipun. Pengguna tidak akan menemukan gambar dinosaurus meski jaringan sedang terputus sehingga memudahkan pengguna dan sistem ini dapat diakses menggunakan *PC*, *Laptop* maupun *smartphone*.

II. METODE PENELITIAN

1. Progressive Web Apps

Progressive Web Apps (PWAs) menggunakan kemampuan web modern untuk menghadirkan pengalaman *mobile web* yang cepat, menarik dan dapat diandalkan yang bagus untuk para pengguna dan pebisnis [1].

2. RESTful API

RESTful API dapat menjadi jembatan antara database dengan client (program yang akan mengkonsumsi data) [2]

3. HTTPS

Hypertext Transfer Protocol Secure memiliki pengertian yang sama dengan *http* hanya saja *https* memiliki kelebihan fungsi di bidang keamanan (*secure*) [3].

4. Lighthouse

Lighthouse adalah alat (bantu) sumber terbuka otomatis untuk meningkatkan kualitas aplikasi web. bisa menjalankannya sebagai ekstensi Chrome atau dari baris perintah. Anda beri *lighthouse* sebuah *URL* yang ingin Anda audit, maka *Lighthouse* menjalankan serangkaian pengujian terhadap laman tersebut, kemudian menghasilkan sebuah laporan mengenai seberapa bagus laman itu menjalaninya [4].

5. React Js

React Js merupakan kerangka kerja open source yang menggunakan *library javascript* untuk membuat user interface dan *React* biasa digunakan untuk menangani pengembangan pada aplikasi *single-page* dan aplikasi *mobile*. *React Js* memiliki keunggulan dimana kerangka kerja ini memberikan kecepatan, *simplicity*, dan *scalability*. *React* yang dikembangkan oleh facebook [5].

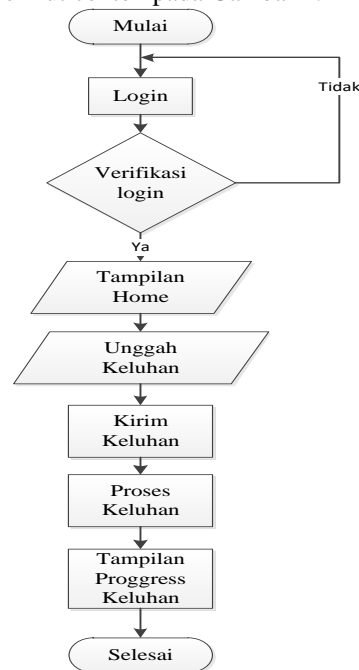
6. Framework Laravel

Framework laravel adalah sebuah kerangka kerja open source yang diciptakan oleh *Taylor Otwell*. *Laravel* merupakan *framework bundle*, migrasi dan *artisan CLI (Command Line Interface)* yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari kerangka kerja seperti *Codeigniter*, *Yii*, *ASP.NET MVC*, *Ruby on Rails*, *Sinatra* dan lain-lain [6].

Tahapan Selanjutnya adalah proses dari perancangan sistem yang digunakan dengan *Flowchart*, *Use Case Diagram*, *Conteks Diagram (CD)*, Dan *Skema Database*. Pengujian sistem menggunakan metode *Black Box*.

A. Flowchart Mahasiswa

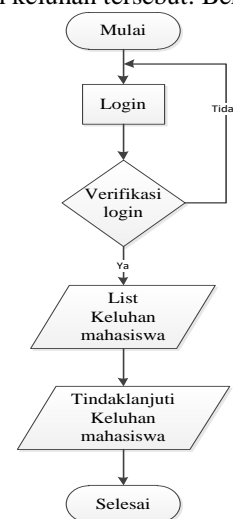
Perancangan sistem yang digunakan pada penelitian ini adalah *flowchart*, mahasiswa bisa melakukan *login* ketika sudah di *register* oleh pihak *staff* kemudian mahasiswa dapat mengupload keluhan berupa *file* yang akan menampilkan ke halaman home dan mahasiswa dapat melihat sejauh mana sudah proses keluhan yang berikan di halaman *progress* keluhan. Berikut contoh pada Gambar 1.



Gambar 1. Flowchart Mahasiswa

B. Flowchart Staf

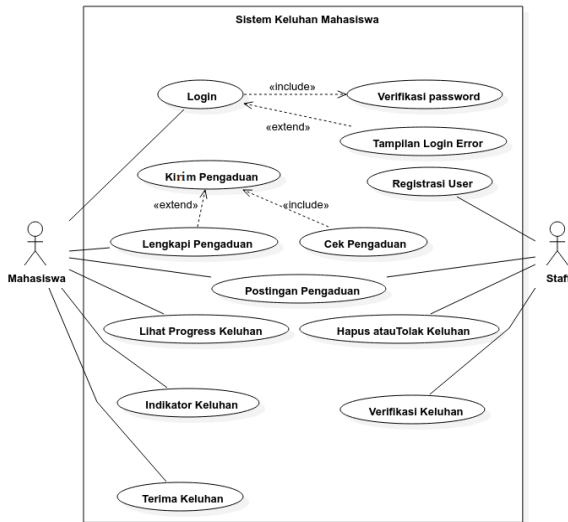
Perancangan sistem yang digunakan pada penelitian ini adalah *flowchart*, semua keluhan yang diberikan oleh mahasiswa, *staff* dapat melihat dan menindak lanjuti keluhan tersebut. Berikut Gambar 2.



Gambar 2 Flowchart Staff

C. Use Case Diagram

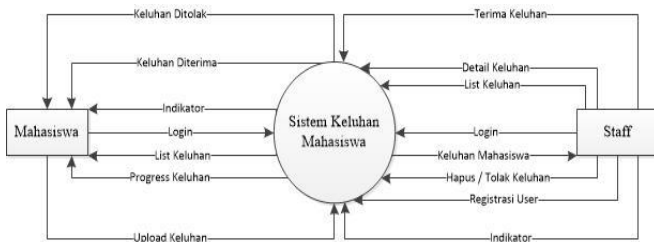
Pada bagian ini menggunakan *use case* untuk mengetahui perilaku dari *aktor* yang menggunakan aplikasi. Disini dibagi 2 *aktor* yaitu aktor *staff* dan mahasiswa. Berikut merupakan *use case* yang diperankan oleh aktor mahasiswa dan *staff* dalam menggunakan sistem aplikasi monitoring keluhan mahasiswa ditunjukkan pada Gambar 3.



Gambar 3 Use Case

D. Konteks Diagram

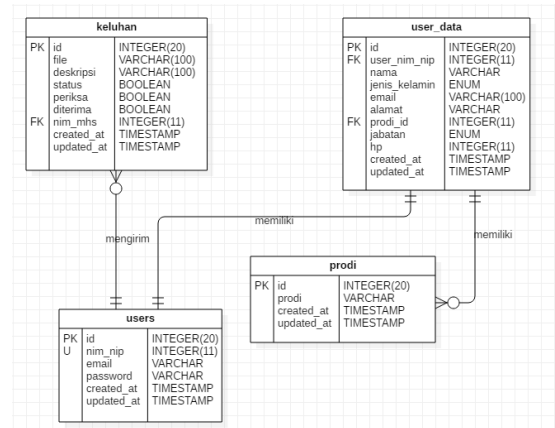
Perancangan sistem yang digunakan adalah diagram konteks yang menggambarkan keseluruhan dari suatu sistem dan merupakan bagian dari *level* tertinggi dari diagram konteks. Adapun yang bisa mengakses sistem ini ialah mahasiswa dan *staff* jurusan Teknologi Informasi dan Komputer dan *staff* bisa mengakses semua kecuali mengupload keluhan, bentuk desain diagram konteks ada pada Gambar 4.



Gambar 4 Konteks Diagram (CD)

E. Skema Database

Berikut adalah skema database atau relasi tabel yang menunjukkan *relasi* antar tiap tabel pada perancangan yang dilakukan pada aplikasi sistem keluhan mahasiswa. Skema database di tunjukkan pada Gambar 5.



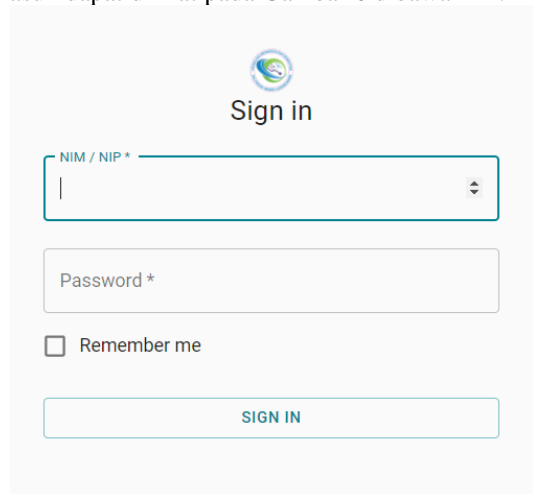
Gambar 5 Skema Database

III. HASIL DAN PEMBAHASAN

A. Hasil User Interface

Sistem keluhan mahasiswa yang dibuat terdapat *user interface* sebagai visual dari *website*. Aplikasi ini memiliki beberapa halaman seperti *form* masuk, daftar, beranda, *upload*, dan halaman lainnya. Penjelasan penggunaan dari masing-masing halaman tersebut akan diuraikan sebagai berikut.

1) Halaman Masuk: Pada halaman ini *user* melakukan proses autentikasi masuk pada aplikasi keluhan mahasiswa. Untuk melakukan proses autentikasi, *user* diminta memasukkan nim atau nip dan juga *password* untuk dapat masuk pada halaman utama keluhan mahasiswa, jika admin berhasil langsung mengarahkan ke halaman *admin*. Tampilan *form* masuk dapat dilihat pada Gambar 6 dibawah ini.

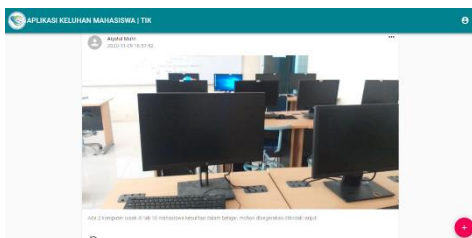


Gambar 6 Halaman Masuk

2) Tampilan Halaman Daftar Untuk Mahasiswa: Halaman daftar berfungsi untuk menyimpan informasi atau data *user* untuk bergabung dengan aplikasi tersebut. Form daftar untuk mahasiswa terdiri dari nama, nim, *password*, email, program studi, jabatan, jenis kelamin, alamat dan *hp*. Tampilan halaman daftar dapat dilihat ada Gambar 7.

Gambar 7 Tampilan Halaman Daftar Untuk Mahasiswa

3) Tampilan Halaman *Beranda*: Pada halaman ini ditampilkan *file – file* dan deskripsi yang di *publish* oleh mahasiswa, adapun *file* yang berupa *video, audio, image* dan *teks* sebagai informasi lebih jelas. Mahasiswa dapat juga melihat keluhan yang diupload sudah terbaca, ditolak dan diterima keluhan pada *icon*, tombol dan mahasiswa juga dapat melihat tindak lanjut keluhan pada tombol *progress* ketika diklik maka akan muncul sebuah *form progress*. Tampilan halaman beranda dapat dilihat pada Gambar 8.



Gambar 8 Tampilan Halaman Beranda

4) Tampilan Halaman *Admin*: Pada halaman ini ditampilkan semua keluhan mahasiswa yang telah mengupload dan *admin* bisa menindak lanjuti keluhan seperti, melihat *detail*, tolak dan terima keluhan sehingga mahasiswa dapat melihat sudah terbaca, diterima, hapus atau ditolak keluhan dapat dilihat pada Gambar 9.

Id	Keluhan	Detail	Tindakan	Tgl. Diterima
1	Keluhan 1	Detail 1	Tolak	2023-11-09 17:00:00
2	Keluhan 2	Detail 2	Diterima	2023-11-09 17:00:00
3	Keluhan 3	Detail 3	Ditolak	2023-11-09 17:00:00
4	Keluhan 4	Detail 4	Diterima	2023-11-09 17:00:00

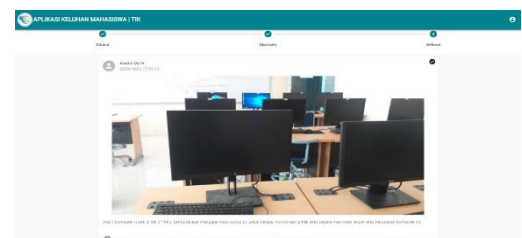
Gambar 9 Tampilan Halaman Admin

5) Tampilan *Form Upload Keluhan*: Pada halaman *form* ini, untuk melakukan proses *upload file* pada sistem, mahasiswa dapat mengupload berupa *gambar, video, audio* dan

juga deskripsi untuk keterangan keluhan. *file* yang diupload ditampilkan ke halaman beranda yang kemudian ditindak lanjuti oleh *admin* dapat dilihat pada Gambar 10.

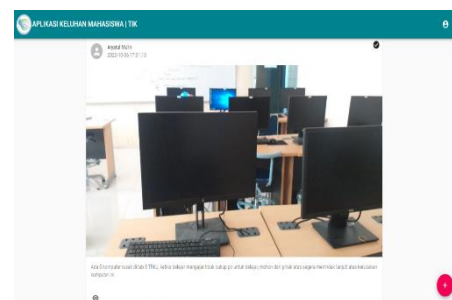
Gambar 10 Tampilan Form Upload Keluhan

6) Tampilan *progress* Keluhan: Berikut *progress* untuk melihat lebih detail keluhan supaya mahasiswa tahu sejauh mana sudah ditindak lanjut keluhan yang telah di *upload* dapat dilihat pada Gambar 11.



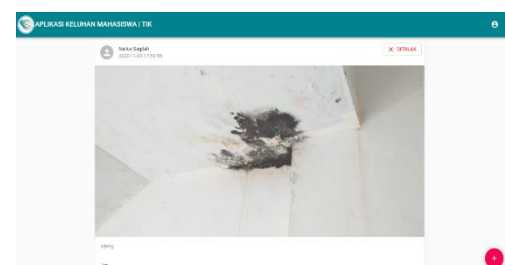
Gambar 11 Tampilan progress Keluhan

7) Tampilan Diterima Keluhan: Berikut tampilan sudah diterima keluhan oleh pihak *admin*, berubah menjadi *icon* centang



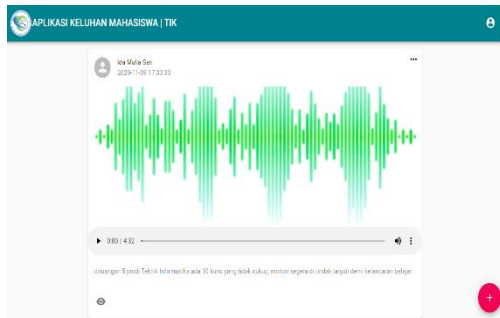
Gambar 12 Tampilan Halaman Diterima Keluhan

8) Tampilan Ditolak Keluhan: Berikut tampilan ditolak keluhan, berarti keluhan yang diberikan tidak cocok atau kurang lengkap. dapat dilihat pada Gambar 13.



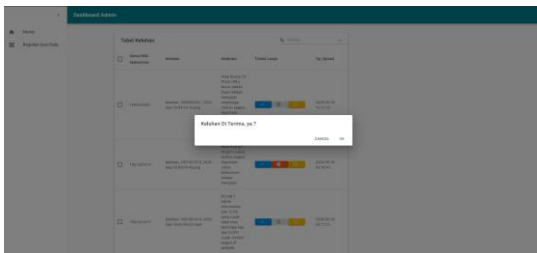
Gambar 13 Tampilan Di tolak Keluhan

9) Tampilan Sudah Dibaca Keluhan: Berikut tampilan sudah dibaca keluhan oleh pihak *admin*, maka akan berubah *icon* dibawah deskripsi dapat dilihat pada Gambar 14.



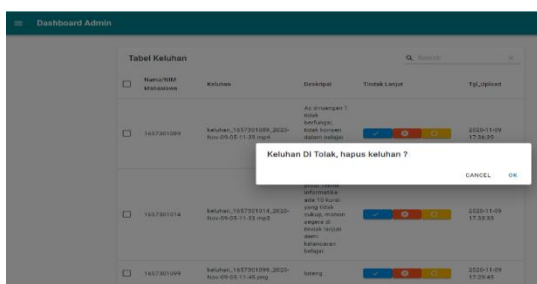
Gambar 14 Tampilan Sudah Dibaca Keluhan

10) Tampilan Terima Keluhan Pada halaman ini admin dapat menerima keluhan yang di *upload* oleh mahasiswa, setelah mengeklik maka secara *otomatis* berubah status icon centang berarti pihak *admin* sudah menerima keluhan yang di *upload* oleh mahasiswa dapat dilihat pada Gambar 15.



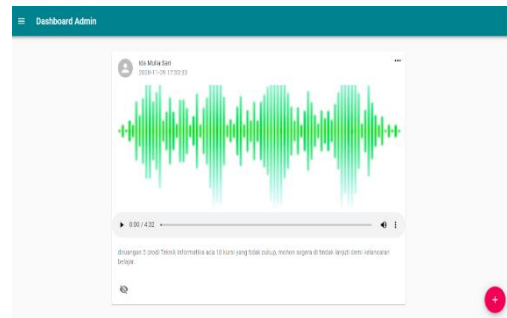
Gambar 15 Halaman Tambah Data Mukim

11) Tampilan Tolak Dan Hapus Keluhan: Pada halaman ini *admin* dapat menolak keluhan yang di *upload* oleh mahasiswa, ada dua pilihan di tombol tolak, jika mengeklik *OK* maka keluhan akan terhapus dan jika mengeklik *cancel* berarti keluhan ditolak. dapat dilihat pada Gambar 16.



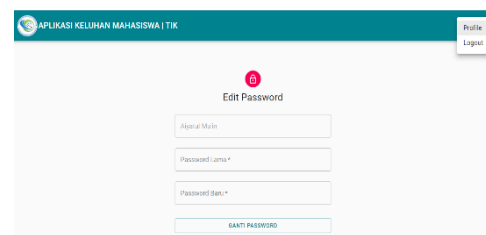
Gambar 16 Tampilan Tolak Dan Hapus Keluhan

12) Tampilan Detail Keluhan: Pada halaman ini menampilkan file upload id yang bisa dilihat oleh admin secara satu persatu berdasarkan id keluhan, kemudian keluhan yang di *upload* user ini akan berubah secara *otomatis icon* di halaman beranda sudah terbaca keluhan beserta akan berubah status di halaman *progress* atau sudah dibaca dapat dilihat pada Gambar 17.



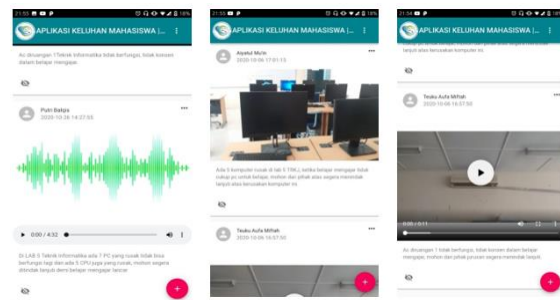
Gambar 17 Tampilan Detail Keluhan

13) Tampilan *Edit Password*: Halaman edit *password* berfungsi untuk mengedit *password user* supaya keamanan lebih terjaga dapat dilihat pada Gambar 18.



Gambar 17 Halaman Lihat Dana

14) Tampilan *Home Page* di Android Pada tampilan ini menampilkan gambar yang berisi suara dan deskripsi, Tampilan *home page* pada *android* dapat dilihat ada Gambar 18



Gambar 18 Tampilan Home Page di Android

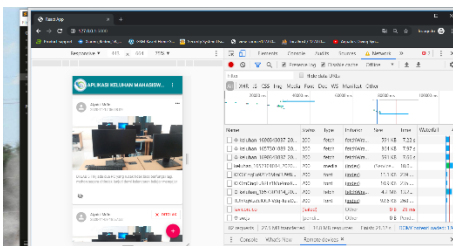
15) Tampilan *Upload* pada *Android* Pada halaman ini mahasiswa dapat meng *upload* keluhan secara langsung seperti foto, *video* dan merekam secara langsung dan dapat juga meng *upload* file. Tampilan halaman *upload* pada *android* dapat dilihat ada di Gambar 19.



Gambar 19. Tampilan Apload Pada Android

16) Tampilan Halaman *Home Offline*

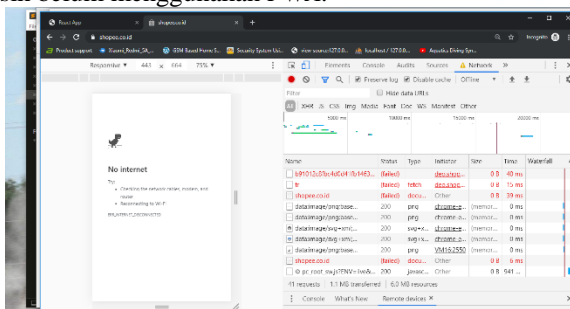
Pada halaman ini menunjukkan bahwa aplikasi web ini telah bisa berjalan Ketika tidak ada jaringan (*offline*)



Gambar 20. Tampilan Halaman Pada Home Offline

17) Tampilan halaman *shopee offline*

Pada halaman ini terlihat bahwa *website* shopee belum dapat berjalan dalam keadaan jaringan *offline* atau *website* ini masih belum menggunakan *PWA*.



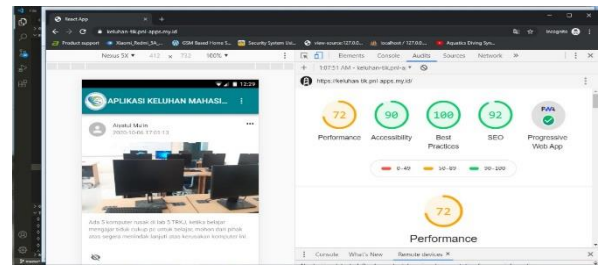
Gambar 21. Tampilan halaman *shopee offline*

B. Pengujian *Blackbox* Otomatis Menggunakan *Lighthouse*

Pengujian sistem merupakan proses yang dilakukan untuk mengetahui apakah sistem telah berjalan sesuai yang diharapkan. Berikut adalah proses pengujian *Blackbox* Otomatis Menggunakan *Lighthouse* di browser *Google Chrome* pada sistem yang dibuat.

1) Pengujian *PWA* Otomatis Menggunakan *Lighthouse*

Pada pengujian ini *PWA* tercentang yang berarti aspek - aspek pada *PWA* terpenuhi, dapat dilihat pada gambar 20.



Gambar 22 Pengujian *PWA* Otomatis Menggunakan *Lighthouse*

2) Pengujian *Performance*

Pada pengujian ini untuk memproses *audit* yang menilai kinerja web seperti seberapa lama web diakses dan membuat pengguna menunggu, kemudian halaman sebuah *web* dapat diakses atau tidak sama sekali, selain itu tidak *responsive* terhadap input dan interaksi yang dilakukan oleh pengguna di dalam web.

TABEL I
PENGUJIAN PERFORMANCE
PERFORMANCE

<i>First Contentful paint</i>	1,9 S
<i>Speed Index</i>	2,3 S
<i>Time To Interaktive</i>	4,7 S
<i>First Meaningful Paint</i>	4,5 S
<i>Total Blocking Time</i>	140 MS
<i>Cumulative Layout Shift</i>	0,176

3) Pengujian Apliasi

Dari hasil pengujian aspek *PWA* dapat dihitung tingkat keberhasilan dengan membagi rata tiap – tiap aspek *PWA*.

$$\text{Poin/aspek (p)} = \frac{100}{a} = \frac{100}{11} = 9,1$$

Dengan:

Point/aspek *PWA* (p) = nilai poin satu aspek dari aspek - aspek *PWA*.

a= jumlah seluruh aspek *PWA*.

Dari penerapan rumus diatas dalam hasil pengujian aspek *PWA* didapatkan hasil seperti ditunjukkan pada Tabel 2

TABEL II
ASPEK – ASPEK *PWA* MENGGUNAKAN *LIGHTHOUSE*

NO	PENGUJIAN	STATUS	NILAI
<i>Fast And Reliable</i>			
1	Halaman termuat cukup cepat dalam jaringan <i>mobile</i>	<i>Passed</i>	9,1
2	Halaman terkini direspon dengan kode 200 ketika <i>offline</i>	<i>Passed</i>	9,1
3	<i>Start_url</i> dapat direspon dengan kode 200 ketika <i>offline</i>	<i>Passed</i>	9,1
<i>Installable</i>			
4	<i>Website</i> Menggunakan <i>HTTPS</i>	<i>Passed</i>	9,1
5	Mendaftarkan <i>service worker</i> yang mengontrol	<i>Passed</i>	9,1

halaman dan *start_url*

6	Manifest aplikasi web memenuhi syarat instalasi aplikasi	Passed	9,1
<i>PWA Optimized</i>			
7	Mengalihkan trafik HTTP ke HTTPS	Passed	9,1
8	Terkonfigurasi menggunakan splash screen seperti yang diinginkan	Passed	9,1
9	Mengandung beberapa konten ketika javascript tidak tersedia	Passed	9,1
10	Memiliki tag <meta name="viewport"> dengan atribut width atau initial-scale	Passed	9,1
10	Manifest tidak memiliki icon latar belakang berwarna	Failed	0
TOTAL			91

4) Pengujian Blackbox Manual Menggunakan Chrome Dev Tools

Pada pengujian ini untuk membandingkan kecepatan aplikasi dalam mengambil dan menampilkan data ketika online dan offline pada pengujian ini dilakukan menggunakan chrome pada pc / laptop dan juga android.

a. Chrome Pada PC/Laptop

TABEL III
PROSES CACHING FILE OLEH SERVICE WORKER DALAM JARINGAN AKTIF (ONLINE).

JUMLAH DATA	TRANSFER RATE	DATA SIZE
9	11 MS	3,1 KB
29	8,36 MS	7,4 KB
41	2,15 S	9,9 KB
61	4,35 S	14,0 KB

TABEL IV
PROSES CACHING FILE OLEH SERVICE WORKER DALAM JARINGAN TIDAK AKTIF (OFFLINE).

JUMLAH DATA	TRANSFER RATE	DATA SIZE
10	4,93 MS	3,3 KB
29	6,0 MS	7,4 KB
41	1,86 S	9,9 KB
61	6,60 S	14,0 KB

TABEL V
PROSES UPLOAD FILE KELUHAN OLEH SERVICE WORKER DALAM JARINGAN AKTIF (ONLINE)

TIPE DATA	TRANSFER RATE	TRANSFER SITE	FILE SIZE
IMAGE	2,36 S	683 B	315 KB
VIDEO	2,22 S	683 B	1,593 KB
AUDIO	14,43 S	683 B	15,798 KB

b. Chrome pada Android

TABEL VI
PROSES CACHING FILE OLEH SERVICE WORKER DALAM JARINGAN AKTIF (ONLINE) SESUDAH WEBSITE AKTIF MENGGUNAKAN ANDROID REDMI NOTE 4.

JUMLAH DATA	TRANSFER RATE	RESOURCE SIZE
9	1,16 S	2,3 KB
31	1,36 S	7,3 KB

TABEL VII
PROSES CACHING FILE OLEH SERVICE WORKER DALAM JARINGAN TIDAK AKTIF (OFFLINE) SESUDAH WEBSITE AKTIF MENGGUNAKAN ANDROID REDMI NOTE 4.

JUMLAH DATA	TRANSFER RATE	RESOURCE SIZE
9	10 MS	2,5 KB
31	10 MS	7,3 KB

IV. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah diuraikan sebelumnya, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dengan menerapkan *web app* manifest pada *PWA*, pengguna dapat menjalankan aplikasi keluhan mahasiswa layaknya aplikasi native yang dapat dijalankan dengan mengklik ikon pada layar utama *smartphone*. Aplikasi ini juga dapat diakses pada berbagai platform dengan menggunakan *browser* seluler maupun desktop pada keadaan *offline* dengan disisipkannya *service worker*, sehingga pengguna dapat mengakses keluhan mahasiswa tanpa jaringan internet.
2. Pengujian Otomatis Menggunakan *Lighthouse checklist*, nilai *Performance*, *Fast And Reliable*, *Installable* dan *PWA Optimized* pada aplikasi keluhan mahasiswa tercentang semua yang berarti penerapan *PWA* sangat baik dari 10 kriteria pengujian dengan hasil persentasi nilai 91/100.
3. Pengujian manual tingkat transfer lebih cepat *offline* dari pada *online* yang berukuran data sama.
4. Proses *caching file* oleh *service worker* mempercepat halaman web keluhan mahasiswa saat dimuat untuk kedua kalinya dan seterusnya atau lebih cepat *offline* dari pada *online* dan dapat diakses ketika *offline*.

V. REFERENSI

- [1] Google. 2019. Secure your site with HTTPS. dilihat 4 Desember 2019 <https://support.google.com/webmasters/answer/6073543>
- [2] Prakoso, G. L. (2019). No Title <https://www.galihprakoso.com/2019/04/apa-itu-rest-ful-api-pengertian-penjelasan.html>
- [3] Zabar, A. A., & Novianto, F. (2015). Keamanan Http Dan Https Berbasis Web Menggunakan Sistem Operasi Kali Linux. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, 4(2), 69–74. <https://doi.org/10.34010/komputa.v4i2.2427>
- [4] Adi, L., Akbar, R. J., & Khotimah, W. N. (2018). Platform e-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps. *Jurnal Teknik ITS*, 6(2), 2–6.
- [5] Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri). *J-Ptiik.Ub.Ac.Id*, 4(1), 46–55. <http://j-ptiik.ub.ac.id>
- [6] Handika, I. G., & Purbasari, A. (2018). Pemanfaatan Framework Laravel Dalam Pembangunan Aplikasi E-Travel Berbasis Website. *Konferensi Nasional Sistem Informasi STMIK Atma Luhur Pangkalpinang*, 1329–1334.