

PROGRESSIVE WEB APP-BASED ONLINE REPAIR SHOP APPLICATION USING MAPBOX AND GEOLOCATION API

Muhammad Aryo Putra Rahmat¹, Ipung Permadi², Nofiyati^{*3}

^{1,2,3}Informatics, Faculty of Engineering, Universitas Jenderal Soedirman, Indonesia
Email: ¹muhammad.rahmat@mhs.unsoed.ac.id, ²ipung.permadi@unsoed.ac.id, ³nofiyati@unsoed.ac.id

(Article received: June 25, 2023; Revision: July 08, 2023; published: August 21, 2023)

Abstract

Motorcycle users in Indonesia continue to increase every year. This has implications for the increasing need for motorcycle repair or maintenance service providers. This need is an emergency, but this is a problem because the lack of information regarding the location of the nearest workshop makes the process of finding a repair shop consuming a lot of time and energy. This study aims to develop an Online Workshop application that can present a map with the nearest repair shop location using Mapbox and the Geolocation API to make it easier for motorbike users to find a repair shop. In addition, this application allows users to be able to call a mechanic to come to the location. The Bengkel Online application is built using Firebase as a database and applies progressive web apps technology. The results of this study indicate that the online workshop application can be used to order repair services and call a mechanic to the location properly and accurately as indicated by the results of application testing which shows that all features function properly.

Keywords: *Firestore, Geolocation, Mapbox, Motorcycle, Progressive Web Apps, Repair shop.*

APLIKASI BENGKEL ONLINE BERBASIS PROGRESSIVE WEB APP MENGUNAKAN MAPBOX DAN GEOLOCATION API

Abstrak

Pengguna sepeda motor di Indonesia terus meningkat setiap tahun. Hal ini berimplikasi pada naiknya kebutuhan terhadap penyedia layanan perbaikan atau perawatan sepeda motor. Kebutuhan ini bersifat darurat, namun hal ini menjadi masalah karena kurangnya informasi mengenai lokasi bengkel terdekat membuat proses pencarian bengkel menghabiskan banyak waktu dan energi. Hal tersebut penting untuk dilakukan karena banyak pengendara yang membutuhkan layanan bengkel secara online dalam mensolusi permasalahan di jalan. Penelitian ini bertujuan untuk mengembangkan aplikasi Bengkel Online yang dapat menyajikan peta dengan titik lokasi bengkel terdekat menggunakan *Mapbox* dan *Geolocation* API untuk mempermudah para pengguna sepeda motor dalam mencari bengkel. Selain itu aplikasi ini memungkinkan pengguna untuk dapat memanggil montir datang ke lokasi. Aplikasi Bengkel Online dibangun menggunakan *Firestore* sebagai basis data serta menerapkan teknologi *progressive web apps*. Hasil penelitian ini menunjukkan bahwa aplikasi bengkel online dapat digunakan untuk memesan layanan bengkel dan memanggil montir ke lokasi dengan baik dan akurat yang ditunjukkan dengan hasil pengujian aplikasi yang menunjukkan fungsi semua fitur dapat berjalan dengan baik.

Kata kunci: *Bengkel, Firestore, Geolocation, Mapbox, Progressive Web Apps, Sepeda Motor.*

1. PENDAHULUAN

Dewasa ini sepeda motor sudah menjadi kebutuhan bagi masyarakat dalam mendukung kegiatan sehari-harinya. Bukan tanpa alasan, pelayanan sistem transportasi umum di Indonesia yang dinilai masih belum baik dan sering terjadinya kemacetan di jalan menjadi alasan mengapa masyarakat lebih memilih menggunakan kendaraan pribadi salah satunya sepeda motor karena dianggap lebih efisien dan efektif. Pada tahun 2020, jumlah motor yang terdata oleh BPS ada 115,023,039 unit.

Jika dibandingkan dengan jumlah pada tahun 2018, setidaknya ada kenaikan sekitar 9 juta unit dalam jangka waktu dua tahun [1]. Peningkatan jumlah pengguna sepeda motor ini berimplikasi pada naiknya kebutuhan masyarakat terhadap layanan bengkel. Bengkel merupakan suatu usaha jenis wirausaha kecil dan menengah yang bergerak dalam bidang jasa pelayanan perbaikan, baik itu sepeda motor atau mobil [2]. Seringkali kebutuhan terhadap bengkel bersifat darurat dan harus segera dipenuhi. Kurangnya informasi mengenai lokasi bengkel

terdekat membuat pengendara motor kesulitan untuk mencari bengkel.

Masalah keterbatasan informasi ini seharusnya bisa diselesaikan dengan adanya teknologi informasi yang tengah berkembang pesat di Indonesia. *We Are Social* dan Hootsuite yang dikutip oleh databoks.katadata.co.id mengemukakan bahwa jumlah pengguna internet di Indonesia terus meningkat dalam lima tahun terakhir. Pada Januari 2022 setidaknya ada 204,7 juta pengguna internet di Indonesia yang jika dibandingkan dengan tahun 2018, jumlah pengguna internet nasional sudah melonjak sekitar 54,25% [3]. Sebuah penelitian lain mengemukakan fakta menarik bahwa aplikasi *mobile web* memiliki jangkauan jauh lebih tinggi dibandingkan dengan aplikasi *native*. Ada sekitar 11,4 juta pengunjung unik tiap bulan pada *mobile web* dibandingkan dengan 4 juta pengunjung pada aplikasi *native* [4]. Dengan demikian bisa dikatakan bahwa kebanyakan orang mengakses konten digital menggunakan perangkat *mobile* nya melalui *mobile web*.

Direktur *Product Management Google Chrome*, Rahul RoyChowdury, pada gelaran *Google I/O 2016* menjelaskan mengenai sebuah teknologi baru yang dapat membuat *mobile web* memiliki pengalaman lebih baik dengan fitur dan tampilan yang hampir sama dengan aplikasi *native*, teknologi tersebut adalah *Progressive web app* [5].

Demi menyajikan layanan aplikasi yang cepat kepada pengguna, aplikasi ini menerapkan *realtime database* menggunakan layanan *Firebase Cloud Firestore*. Selain itu, aplikasi ini juga menggunakan *Mapbox* yang memungkinkan pengembang untuk menyajikan *maps* di *web browser*, menambahkan *user interactivity* dan mengatur sendiri tampilan dan perilaku dari *maps* [6], serta *Geolocation API* yang memungkinkan pengguna untuk menyediakan lokasi mereka kepada aplikasi *web* jika pengguna mengizinkan [7]. Cepat atau lambat nya aplikasi web yang diakses menjadi salah satu tantangan dalam pengembangan web. Berdasarkan dari data yang dihimpun oleh Google, sebanyak 53% konsumen akan meninggalkan halaman web jika memakan waktu lebih dari 3 detik untuk memuat halaman web [8]. Dibutuhkan koneksi internet yang baik untuk mengakses aplikasi web dan aplikasi web biasa tidak bisa diunduh ke dalam perangkat *mobile* layaknya aplikasi *native*, sehingga Google menciptakan teknologi baru dalam pengembangan web untuk mengatasi masalah tersebut, yaitu *Progressive Web Application (PWA)* [9].

PWA adalah metode dalam pengembangan software dengan mengkombinasikan aplikasi *native*, aplikasi web, dan desktop. Di dunia kosmetik ada istilah “*your lips but better*”. Di dunia IT, PWA dikenal sebagai “*your website but better*” [10]. PWA juga dapat mengirim *push notification* yang relevan kepada pengguna, dapat memiliki icon aplikasi di layar beranda ponsel, dan hanya membutuhkan

sedikit ruang memori [11]. Penerapan metode PWA dalam pembuatan sebuah web perlu diuji untuk membandingkan kecepatan akses antara web PWA dengan web standar dan membandingkan penghematan memori dalam mengakses sebuah CSS yang dirender saat web dibuka. Penelitian lainnya Majchrzak, Hansen dan Gronli [12]-[13], mempelajari apakah PWA nantinya bisa menjadi teknologi pemersatu dalam pengembangan web dan aplikasi *native* dimana PWA sudah mendekati sebuah metode pengembangan web secara modern dan memiliki teknologi serta manfaat yang luar biasa.

Peneliti lain yang dilakukan oleh Nurwanto [14] mengimplementasikan metode PWA pada situs web *e-commerce*, dimana hasilnya PWA merupakan metode pengembangan web yang luar biasa karena pengguna dapat menjalankan web *e-commerce* tersebut layaknya aplikasi *native (push notification)*, adanya icon aplikasi pada beranda ponsel, dan dapat berjalan saat offline).

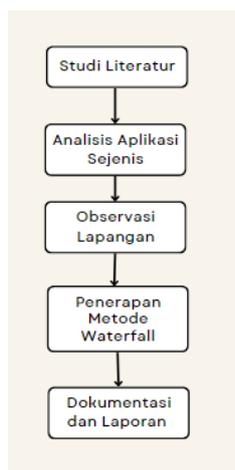
Selain menerapkan *Progressive web app*, dalam penelitian ini juga mengimplementasikan *Geolocation API*. Menurut [15] *Geolocation API* menentukan antarmuka tingkat tinggi ke informasi lokasi yang hanya terkait dengan perangkat yang menghosting implementasi. Sumber umum informasi lokasi meliputi *Global Positioning System (GPS)* dan lokasi yang disimpulkan dari sinyal jaringan seperti alamat IP, RFID, WiFi dan alamat MAC Bluetooth, dan ID sel GSM/CDMA, serta input pengguna.

Dengan mempertimbangkan semua data diatas, maka penelitian ini penting untuk dilakukan dalam memenuhi kebutuhan terhadap bengkel online yang bersifat darurat dan harus segera dipenuhi dengan melakukan melakukan perancangan aplikasi Bengkel Online untuk mencari lokasi bengkel terdekat dan memesan layanan bengkel berbasis *progressive web apps*. Aplikasi ini nantinya dapat menampilkan peta dengan beberapa lokasi bengkel terdekat kepada pengendara motor. Kemudian pengendara motor dapat memanggil montir dari bengkel yang dipilih.

2. METODE PENELITIAN

Secara lebih detail kegiatan penelitian yang dilakukan mengikuti tahapan penelitian dengan mengimplementasikan metode yang digunakan yaitu Waterfall yang ditunjukkan pada Gambar 1.

Penelitian ini diawali dengan melakukan studi literatur, kemudian melakukan analisis aplikasi sejenis dan observasi di lapangan mengenai bagaimana sebuah bengkel beroperasi, bagaimana kebiasaan pengendara motor ketika mencari bengkel, serta fitur apa saja yang biasanya terdapat dalam aplikasi pendeteksi lokasi serupa.



Gambar 1. Alur Penelitian

Tahapan analisis dan definisi kebutuhan merupakan tahapan yang berguna untuk mengetahui apa saja kebutuhan pengguna pada aplikasi. Dengan menganalisis kebutuhan yang diperlukan nantinya aplikasi akan mencapai keadaan sempurna seperti yang diinginkan pengguna. Dalam mengumpulkan kebutuhan pengguna dan sistem, terdapat beberapa metode yang akan digunakan, antara lain:

a. Studi Literatur

Studi literatur merupakan salah satu cara untuk mengkaji penelitian yang sudah dilakukan sebelumnya ataupun yang terpampang pada sebuah artikel dengan menggunakan bahasa dan pengertian dari diri sendiri sebagai landasan dasar pembuatan aplikasi.

b. Analisis Aplikasi Sejenis

Mempelajari beberapa aplikasi yang sejenis dengan aplikasi yang akan dikembangkan untuk mendapatkan data dan informasi yang berguna untuk pengembangan aplikasi. Aplikasi sejenis yang dijadikan sebagai referensi adalah Gojek, Grab, dan Google Maps. Analisis dilakukan dengan melihat dan mengamati *user interface* dan fitur serta kegunaannya yang ada pada beberapa aplikasi tersebut.

c. Observasi

Pada penelitian ini observasi dilakukan dengan memperhatikan bagaimana cara bengkel beroperasi dan bagaimana kebiasaan pengendara motor dalam mencari bengkel untuk mendapatkan data kebutuhan pengguna yang diperlukan.

d. Penerapan Metode Waterfall

Metode pengembangan sistem menggunakan metode *waterfall* yang memiliki beberapa tahapan, yaitu analisis dan definisi kebutuhan, desain sistem dan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem, operasi dan pemeliharaan sistem.

e. Dokumentasi dan Laporan

Langkah terakhir dalam penelitian ini yaitu membuat dokumentasi hasil penelitian yang dilakukan serta membuat laporan hasil penelitian yang salah satunya membuat artikel jurnal ilmiah.

3. HASIL DAN PEMBAHASAN

3.1. Analisis dan Definisi Kebutuhan

Berdasarkan pengumpulan data yang dilakukan dengan studi literatur, analisis aplikasi sejenis, dan observasi di lapangan, diperoleh siapa saja pengguna yang akan berperan dalam aplikasi Bengkel Online pada penelitian ini beserta kebutuhannya.

Pengguna tersebut adalah pihak bengkel, montir dan pengendara motor. Ketiga pengguna tersebut memiliki kebutuhan yang berbeda. Adapun kebutuhan dari ketiganya akan dijelaskan pada Tabel 1. Selanjutnya kebutuhan sistem ditunjukkan pada Tabel 2.

Tabel 1. Kebutuhan Pengguna

Pengguna	Kebutuhan	Kode
Pengendara Motor	1. Pengguna dapat melihat peta dengan titik lokasi pengguna	[URS.BOP WA.01]
	2. Pengguna dapat melihat peta dengan titik lokasi bengkel terdekat	[URS.BOP WA.02]
	3. Pengguna dapat melihat informasi mengenai bengkel yang dipilih	[URS.BOP WA.03]
	4. Pengguna dapat melihat rute terpendek dari lokasi pengguna menuju lokasi bengkel yang dipilih	[URS.BOP WA.04]
	5. Pengguna dapat memanggil montir bengkel untuk datang ke lokasi pengguna	[URS.BOP WA.05]
	6. Pengguna dapat menerima keterangan jika montir sudah sampai di lokasi pengguna	[URS.BOP WA.06]
	7. Pengguna dapat menerima informasi total biaya perbaikan	[URS.BOP WA.07]
	8. Pengguna dapat memberikan ulasan setelah menerima pelayanan	[URS.BOP WA.08]
	9. Pengguna dapat mengubah profil	[URS.BOP WA.09]
	10. Pengguna dapat memasang aplikasi web pada perangkat <i>smartphone</i> milik pengguna	[URS.BOP WA.10]
Bengkel	1. Pengguna dapat mengubah status menjadi "buka" jika ingin menerima panggilan dari pengendara motor	[URS.BOP WA.11]
	2. Pengguna dapat mengubah status menjadi "tutup" jika sedang tidak ingin menerima panggilan dari pengendara motor	[URS.BOP WA.12]
	3. Pengguna dapat mengubah status ketersediaan tiap montir di bengkel	[URS.BOP WA.13]
	4. Pengguna dapat menerima notifikasi jika ada panggilan dari pengendara motor	[URS.BOP WA.14]
	5. Pengguna dapat melihat informasi beserta lokasi pengendara motor yang melakukan panggilan	[URS.BOP WA.15]
	6. Pengguna dapat melihat rute terdekat menuju lokasi pengendara motor yang melakukan panggilan.	[URS.BOP WA.16]

Pengguna	Kebutuhan	Kode
	7. Pengguna dapat menerima atau menolak panggilan dari pengendara motor	[URS.BOP WA.17]
	8. Pengguna dapat memilih montir yang akan dikirim ke lokasi pengendara motor yang melakukan panggilan	[URS.BOP WA.18]
	9. Pengguna dapat melihat informasi hasil pengecekan masalah motor oleh montir	[URS.BOP WA.19]
	10. Pengguna dapat memasukkan total biaya perbaikan motor	[URS.BOP WA.20]
	11. Pengguna dapat melihat semua panggilan yang sedang diproses	[URS.BOP WA.21]
	12. Pengguna dapat melihat semua panggilan yang sukses diproses	[URS.BOP WA.22]
	13. Pengguna dapat menambahkan montir	[URS.BOP WA.23]
	14. Pengguna dapat menghapus montir	[URS.BOP WA.24]
	15. Pengguna dapat mengubah profil	[URS.BOP WA.25]
	16. Pengguna dapat memasang aplikasi web pada perangkat smartphone milik pengguna	[URS.BOP WA.26]
Montir	1. Pengguna dapat menerima tugas dari Bengkel untuk berangkat ke lokasi pengendara motor	[URS.BOP WA.27]
	2. Pengguna dapat melihat informasi beserta lokasi pengendara motor yang melakukan panggilan	[URS.BOP WA.28]
	3. Pengguna dapat melihat rute terdekat menuju lokasi pengendara motor yang melakukan panggilan	[URS.BOP WA.29]
	4. Pengguna dapat memberi informasi sudah mulai berangkat ke lokasi pengendara motor	[URS.BOP WA.30]
	5. Pengguna dapat memberi informasi sudah sampai di lokasi penendara motor	[URS.BOP WA.31]
	6. Pengguna dapat memberi input hasil pengecekan masalah motor untuk dilakukan perhitungan biaya oleh bengkel	[URS.BOP WA.32]
	7. Pengguna dapat melakukan konfirmasi jika pembayaran telah dilakukan.	[URS.BOP WA.33]

Tabel 1. Kebutuhan Sistem

No.	Kebutuhan	Kode
1.	Sistem menyediakan form <i>login</i> dan <i>register</i> menggunakan nomor telepon dan verifikasi kode OTP.	[SRS.BOPWA.01]
2.	Sistem menyajikan peta dengan titik lokasi bengkel terdekat	[SRS.BOPWA.02]
3.	Sistem menampilkan informasi bengkel	[SRS.BOPWA.03]
4.	Sistem menampilkan rute terdekat menuju bengkel	[SRS.BOPWA.04]
5.	Sistem mengirim notifikasi ke perangkat pengguna	[SRS.BOPWA.05]
6.	Sistem menampilkan informasi montir sudah sampai	[SRS.BOPWA.06]

No.	Kebutuhan	Kode
7.	Sistem menampilkan form input hasil pengecekan motor	[SRS.BOPWA.07]
8.	Sitem menampilkan informasi hasil pengecekan motor	[SRS.BOPWA.08]
9.	Sistem menampilkan form input total biaya perbaikan motor	[SRS.BOPWA.09]
10.	Sistem menampilkan informasi total biaya perbaikan motor	[SRS.BOPWA.10]
11.	Sistem menyajikan tombol konfirmasi pembayaran	[SRS.BOPWA.11]
12.	Sistem menyajikan form beri ulasan	[SRS.BOPWA.12]
13.	Sistem menyajikan form ubah profil pengendara motor	[SRS.BOPWA.13]
14.	Sistem menyajikan tombol untuk memasang aplikasi pada perangkat pengguna	[SRS.BOPWA.14]
15.	Sistem menyajikan tombol untuk ubah status bengkel	[SRS.BOPWA.15]
16.	Sistem menyajikan tombol untuk ubah status ketersediaan montir	[SRS.BOPWA.16]
17.	Sistem menampilkan notifikasi panggilan pada perangkat pengguna	[SRS.BOPWA.17]
18.	Sistem menampilkan informasi panggilan seperti lokasi dan data pengendara motor	[SRS.BOPWA.18]
19.	Sistem menampilkan rute terdekat menuju pengendara motor	[SRS.BOPWA.19]
20.	Sistem menyajikan tombol tolak atau terima panggilan	[SRS.BOPWA.20]
21.	Sistem menyajikan form pilih montir yang akan ditugaskan melayani panggilan	[SRS.BOPWA.21]
22.	Sistem menampilkan informasi panggilan yang sedang diproses	[SRS.BOPWA.22]
23.	Sistem menampilkan informasi panggilan yang telah sukses diproses	[SRS.BOPWA.23]
24.	Sistem menyajikan form tambah montir	[SRS.BOPWA.24]
25.	Sistem menyajikan tombol hapus montir	[SRS.BOPWA.25]
26.	Sistem menyajikan form ubah profil bengkel	[SRS.BOPWA.26]
27.	Sistem menyajikan tombol konfirmasi montir berangkat	[SRS.BOPWA.27]

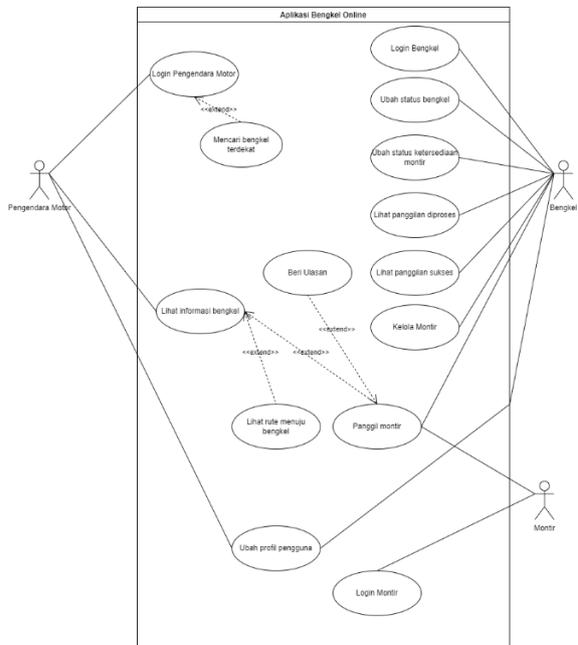
3.2. Desain Sistem dan Perangkat Lunak

Tahap selanjutnya yang dilakukan setelah tahap analisis dan definisi kebutuhan adalah tahap desain sistem dan perangkat lunak. Proses yang dikerjakan pada tahap desain sistem dan perangkat lunak diantaranya pembuatan *use case diagram*, *activity diagram* dan penggambaran skema penyimpanan pada *Firestore*.

Use case diagram merupakan diagram yang menjelaskan interaksi antara sistem dengan aktor yang berperan sebagai pengguna dari sistem [16] - [17]. Aktor yang berperan dalam Aplikasi Bengkel Online yaitu Pengendara Motor, Bengkel dan Montir.

Pengendara Motor merupakan pengguna yang berperan sebagai pelanggan, Bengkel berperan sebagai penyedia jasa yang membawahi beberapa Montir, dan Montir berperan sebagai karyawan dari Bengkel. Semua aktor berinteraksi satu sama lain pada case tertentu.

Gambar 2 menunjukkan *use case diagram* aplikasi Bengkel Online.

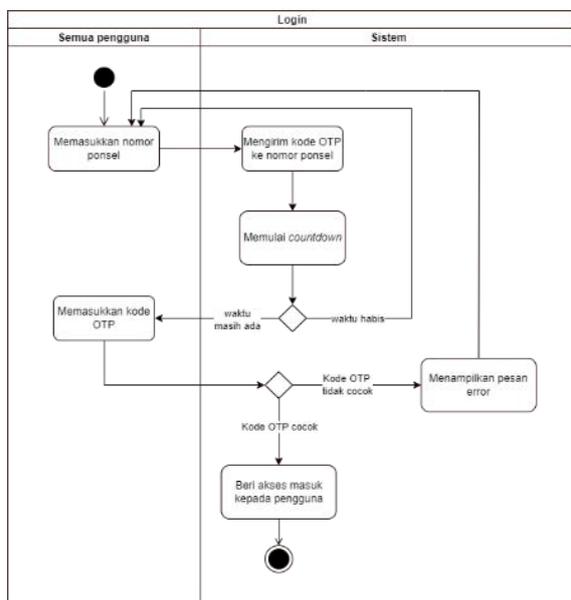


Gambar 2. Use Case Diagram

1. Activity Diagram

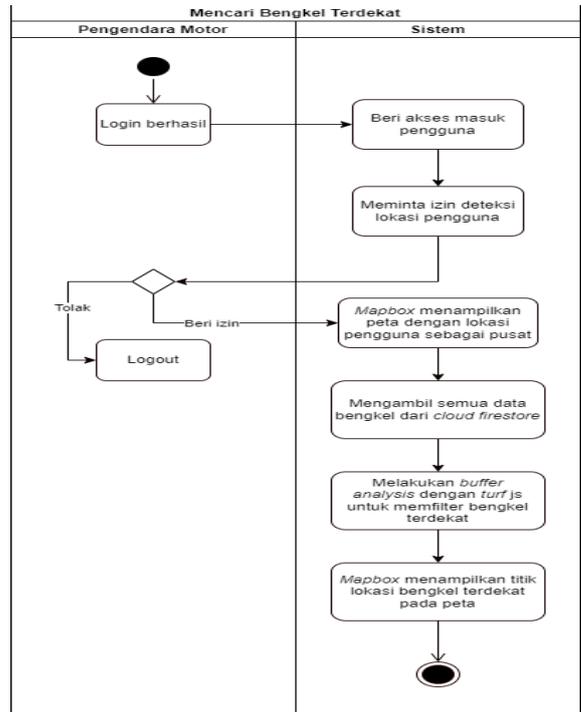
Activity diagram atau diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. *Activity diagram* memberikan *model* aktivitas dari sistem yang menunjukkan alur kerja dari satu aktivitas ke aktivitas lainnya[18].

Gambar 3 menunjukkan *activity diagram login*.



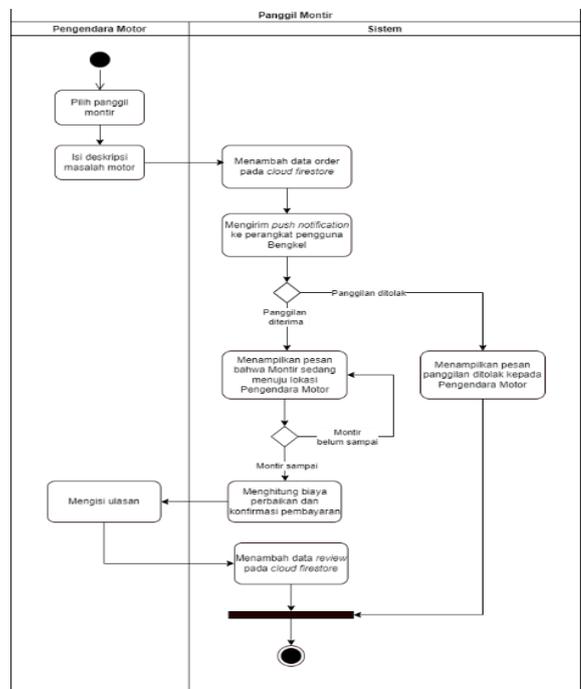
Gambar 3. Activity Diagram Login

Gambar 4 menunjukkan *activity diagram* mencari bengkel terdekat.



Gambar 4. Activity Diagram Mencari Bengkel Terdekat

Gambar 5 menunjukkan *activity diagram* panggilan montir.



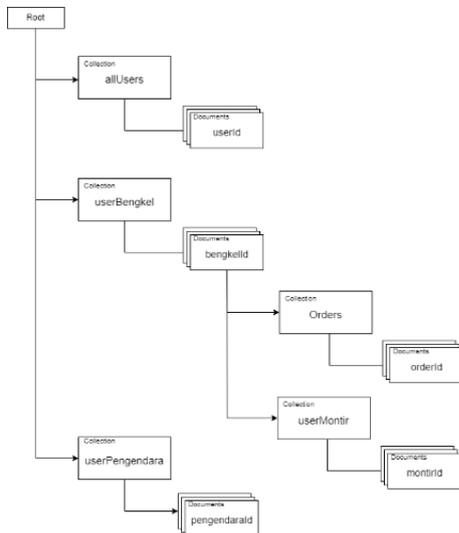
Gambar 5. Activity Diagram Panggil Montir

2. Firestore Database Design

Firestore Database menerapkan *NoSql database*, yaitu datanya *non-relasional* dan tidak menggunakan SQL sebagai bahasa *query*[19]. Firestore merupakan *document-based database*,

dimana data disimpan dalam tiap *document* dalam bentuk *key-value pair* dan tiap *document* dikelompokkan di dalam sebuah *collection*.

Gambar 6 menunjukkan diagram untuk mengilustrasikan struktur penyimpanan data pada Aplikasi Bengkel Online menggunakan *Firestore*.



Gambar 6. Struktur Penyimpanan Data

3.3. Implementasi dan Pengujian Unit

Tahap ini merupakan tahap penerjemahan desain sistem yang telah dibuat ke dalam bentuk perintah-perintah yang dimengerti oleh komputer dengan menggunakan bahasa pemrograman, dalam proyek ini bahasa pemrograman yang digunakan adalah *Javascript*.

Pengujian unit merupakan kegiatan menguji setiap fungsionalitas dari tiap unit program pada aplikasi ini. Pengujian unit dilakukan oleh penulis selama tahap implementasi yaitu saat proses pengkodean atau pengembangan aplikasi ini.

Gambar 7 menunjukkan implementasi tampilan halaman *login*.



Gambar 7. Implementasi Halaman *Login*

Gambar 8 menunjukkan implementasi tampilan halaman utama pengendara motor. Pada halaman inilah terjadinya pencarian lokasi bengkel terdekat.



Gambar 8. Implementasi Halaman Utama Pengendara Motor

Gambar 9 menunjukkan implementasi tampilan halaman informasi bengkel. Halaman ini akan tampil ketika pengendara motor memilih salah satu bengkel pada peta.



Gambar 9. Implementasi Halaman Informasi Bengkel

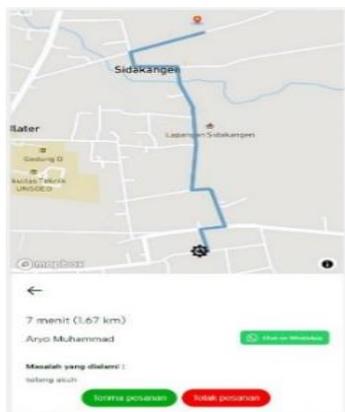
Gambar 10 menunjukkan tampilan form deskripsi masalah motor yang muncul ketika pengendara motor memilih tombol “panggil montir”.



Gambar 10. Implementasi Form Deskripsi Masalah Motor

Gambar 11 menunjukkan tampilan informasi panggilan. Jika bengkel memilih tombol “terima panggilan”, maka bengkel akan diminta memilih

salah satu montir yang akan ditugaskan untuk melayani panggilan.



Gambar 11. Implementasi Halaman Informasi Panggilan

Gambar 12 menunjukkan tampilan halaman konfirmasi keberangkatan montir. Jika montir memilih tombol “berangkat”, aplikasi akan mendeteksi lokasi montir secara *realtime*.



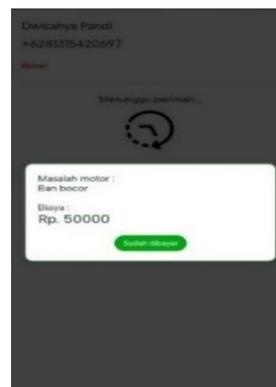
Gambar 12. Implementasi Halaman Konfirmasi Keberangkatan Montir

Gambar 13 menunjukkan tampilan form pengecekan motor oleh montir. Form ini akan menjadi acuan bengkel untuk menentukan biaya perbaikan motor.



Gambar 13. Implementasi Form Pengecekan Motor

Gambar 14 menunjukkan *card* total biaya yang ditentukan oleh bengkel.



Gambar 14. Implementasi *Card* Total Biaya

Gambar 15 menunjukkan form beri ulasan yang tampil setelah pembayaran dilakukan.



Gambar 15. Implementasi Form Beri Ulasan.

3.4. Integrasi dan Pengujian Sistem

Pengujian yang dilakukan terhadap aplikasi Bengkel Online yaitu pengujian *black box* untuk menguji fungsionalitas sistem apakah dapat berfungsi dengan baik atau tidak [20] - [21], dan pengujian kelayakan penerapan teknologi *progressive web apps* yang akan dilakukan menggunakan *lighthouse*.

Hasil pengujian *black box* dapat dilihat pada Tabel 3, 4 dan 5. Dari hasil pengujian menunjukkan bahwa semua fitur dapat berjalan dengan baik sesuai kebutuhan pengguna.

Tabel 3. Hasil Pengujian Black Box oleh Pengendara Motor

Kode	Detail Pengujian	Hasil
TEST.01	Sistem menyediakan form <i>login</i> dan <i>register</i> menggunakan nomor telepon dan verifikasi kode OTP.	Terpenuhi
TEST.02	Sistem menyajikan peta dengan titik lokasi bengkel terdekat	Terpenuhi
TEST.03	Sistem menampilkan informasi bengkel	Terpenuhi
TEST.04	Sistem menampilkan rute terdekat menuju bengkel	Terpenuhi
TEST.05	Sistem mengirim notifikasi ke perangkat pengguna	Terpenuhi
TEST.06	Sistem menampilkan informasi montir sudah sampai	Terpenuhi
TEST.07	Sistem menampilkan informasi total biaya perbaikan	Terpenuhi

Kode	Detail Pengujian	Hasil
TEST.08	Sistem menyajikan form beri ulasan	Terpenuhi
TEST.09	Sistem menyajikan form ubah profil pengendara motor	Terpenuhi
TEST.10	Sistem menyajikan tombol untuk memasang aplikasi pada perangkat pengguna	Terpenuhi

Tabel 4. Hasil Pengujian Black Box oleh Bengkel

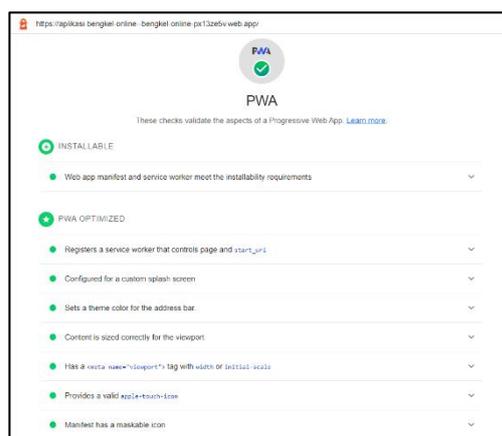
Kode	Detail Pengujian	Hasil
TEST.11	Sistem menyediakan form <i>login</i> dan <i>register</i> menggunakan nomor telepon dan verifikasi kode OTP.	Terpenuhi
TEST.12	Sistem menyajikan tombol untuk memasang aplikasi pada perangkat pengguna	Terpenuhi
TEST.13	Sistem menyajikan tombol untuk ubah status bengkel	Terpenuhi
TEST.14	Sistem menyajikan tombol untuk ubah status ketersediaan montir	Terpenuhi
TEST.15	Sistem menampilkan notifikasi panggilan pada perangkat pengguna	Terpenuhi
TEST.16	Sistem menampilkan informasi panggilan seperti lokasi dan data pengendara motor	Terpenuhi
TEST.17	Sistem menampilkan rute terdekat menuju pengendara motor	Terpenuhi
TEST.18	Sistem menyajikan tombol tolak atau terima panggilan	Terpenuhi
TEST.19	Sistem menyajikan form pilih montir yang akan ditugaskan melayani panggilan	Terpenuhi
TEST.20	Sistem menampilkan informasi hasil pengecekan motor	Terpenuhi
TEST.21	Sistem menyajikan form input total biaya perbaikan	Terpenuhi
TEST.22	Sistem menampilkan informasi panggilan yang sedang diproses	Terpenuhi
TEST.23	Sistem menampilkan informasi panggilan yang telah sukses diproses	Terpenuhi
TEST.24	Sistem menyajikan form tambah montir	Terpenuhi
TEST.25	Sistem menyajikan tombol hapus montir	Terpenuhi
TEST.26	Sistem menyajikan form ubah profil bengkel	Terpenuhi

Tabel 5. Hasil Pengujian Black Box oleh Montir

Kode	Detail Pengujian	Hasil
TEST.27	Sistem menyediakan form <i>login</i> dan <i>register</i> menggunakan nomor telepon dan verifikasi kode OTP.	Terpenuhi
TEST.28	Sistem menampilkan informasi montir sudah sampai	Terpenuhi
TEST.29	Sistem menampilkan informasi panggilan seperti lokasi dan data pengendara motor	Terpenuhi
TEST.30	Sistem menampilkan rute terdekat menuju pengendara motor	Terpenuhi
TEST.31	Sistem menyajikan tombol konfirmasi montir berangkat	Terpenuhi
TEST.32	Sistem menampilkan form input hasil pengecekan motor	Terpenuhi
TEST.33	Sistem menampilkan informasi total biaya perbaikan motor	Terpenuhi
TEST.34	Sistem menyajikan tombol konfirmasi pembayaran	Terpenuhi

Berdasarkan hasil pengujian yang dilakukan oleh semua aktor yang dapat menggunakan aplikasi menunjukkan bahwa semua fitur yang dibuat telah memenuhi kebutuhan pengguna dan aplikasi dapat digunakan untuk menemukan bengkel terdekat dan memanggil montir ke lokasi pengendara motor yang mengalami kendala.

Selain pengujian *blackbox*, pengujian kelayakan penerapan teknologi *progressive web apps* juga dilakukan menggunakan *lighthouse*. Hasil pengujian menunjukkan bahwa penerapan *progressive web apps* telah berhasil dan dapat digunakan. Hasil pengujian ditunjukkan pada Gambar 16.



Gambar 16. Hasil Pengujian Lighthouse

4. DISKUSI

Rico Maulana melakukan penelitian untuk membangun aplikasi untuk mencari lokasi bengkel terdekat menggunakan *Google Maps API* pada perangkat android. Aplikasi ini memungkinkan pengguna untuk mencari bengkel yang berada pada radius terdekat dari lokasi pengguna[22].

Resya Septian Hamsyah melakukan penelitian untuk membangun aplikasi untuk mencari dan memanggil teknisi tambal ban terdekat. Aplikasi yang dibangun memungkinkan pengguna untuk mencari tambal ban terdekat menggunakan *Google Maps API*. Kelebihan dari aplikasi ini adalah pada aplikasi ini pengguna dapat memanggil teknisi tambal ban yang kemudian teknisi tambal ban dapat melihat lokasi pengguna yang memanggil nya[23].

Nabila Pratiwi Kiswanto, dkk melakukan penelitian untuk membangun aplikasi *E-log book* penangkapan ikan dengan menerapkan teknologi *Progressive web app*. Aplikasi ini memungkinkan pengguna untuk melakukan pencatatan tangkapan ikan seperti ikan apa yang ditangkap, kapan ikan ditangkap, siapa nelayan yang menangkap dan di wilayah mana ikan ditangkap. Aplikasi ini menerapkan teknologi *Progressive web app* karena PWA memiliki fitur seperti aplikasi *native* namun lebih ringan dan mudah untuk digunakan oleh banyak orang[24].

Berdasarkan hasil penelitian serupa yang dilakukan dengan menerapkan teknologi *Progressive*

web app pada pembangunan aplikasi bengkel online dapat diterapkan dan mendukung pengguna untuk memanggil teknisi dan melihat lokasi pengguna, sedangkan pada penelitian ini selain dapat memanggil teknisi bengkel untuk datang ke lokasi pengguna juga dapat mendeteksi lokasi bengkel terdekat yang sedang on untuk melayani pengendara dengan mendeteksi lokasi bengkel terdekat menggunakan *Mapbox* dan *Geolocation* API serta dapat memanggil teknisi bengkel yang dipilih.

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan maka menghasilkan aplikasi Bengkel Online berbasis *progressive web app* dengan mengimplementasikan *Mapbox*, *Turf Js* dan *Geolocation* API yang dapat digunakan dalam pengembangan aplikasi Bengkel Online untuk membuat peta interaktif beserta fitur deteksi lokasi secara *realtime* sehingga memudahkan pengendara motor dalam mencari bengkel terdekat dan memesan layanan bengkel, serta memudahkan pihak bengkel dalam memberi layanan kepada pelanggan.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, "Badan Pusat Statistik." <https://www.bps.go.id/indicator/17/57/1/jumlah-kendaraan-bermotor.html> (accessed Apr.05, 2022).
- [2] Amri M., Dedi, Mutiara, "Manajemen Penjualan Sparepart Sepeda Motor Pada Bengkel Sinar Jaya Motor Berbasis Online," *Jurnal Tren Bisnis Global*, Vol. 2 No. 1, PP 39-45, 2022.
- [3] C. M. Annur, "Ada 204,7 Juta Pengguna Internet di Indonesia Awal 2022 | Databoks," 2022. <https://databoks.katadata.co.id/datapublish/2022/03/23/ada-2047-juta-pengguna-internet-di-indonesia-awal-2022> (accessed Apr. 06, 2022).
- [4] S. S. Tandel and A. Jamadar, "Impact of Progressive Web Apps on Web App Development," *Int. J. Innov. Res. Sci.*, vol. 7, no. 9, 2018, doi: 10.15680/IJRSET.2018.0709021.
- [5] R. RoyChowdury, "Google I/O 2016 - Keynote - YouTube," 2016. <https://www.youtube.com/watch?v=862r3XS2YB0> (accessed Apr. 05, 2022).
- [6] Mapbox, "Mapbox GL JS," 2022. <https://docs.mapbox.com/mapbox-gl-js/guides/> (accessed Jul. 06, 2022).
- [7] Mozilla, "Geolocation API - Web APIs," 2022. https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API (accessed Jul. 06, 2022).
- [8] D. Kirkpatrick, "Google: 53% of mobile users abandon sites that take over 3 seconds to load", *Marketing Dive*. [Online]. Available: <https://www.marketingdive.com/news/google-53-of-mobile-users-abandonsites-that-take-over-3-seconds-to-load/426070/>. [Accessed: 10- Juni 2023].
- [9] Kurniawan A.A, Analisis Performa Progressive Web Application (PWA) Pada Perangkat Mobile, *Jurnal Ilmiah Informatika Komputer* [Vol 25, No 1, 2020](#), p-ISSN: 0853-8638, e-ISSN:2089-8045, DOI: <http://dx.doi.org/10.35760/ik.2020.v25i1.2510>
- [10] SoftwareSeni, "Membuat Aplikasi Dengan Progressive Web Application (PWA)", *SoftwareSeni*, 2019. [Online]. Available: <https://www.softwareseni.co.id/progressive-web-application-pwa/>. [Accessed: 15- Juni 2023].
- [11] R. Mishra, "Progressive WEBAPP:Review", *International Research Journal Of Engineering and Technology (IRJET)*, vol. 3, no. 6, 2016.
- [12] S. Tandel and A. Jamadar, "Impact of Progressive Web Apps on Web App Development", *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 7, no. 9, 2018. [Accessed 4 April 2020].
- [13] T. Majchrzak, A. Hansen and T. Gronli, "Progressive Web Apps: the Definite Approach to Cross-Platform Development?," in *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.
- [14] Nurwanto, "Penerapan Progressive Web Application (PWA) pada ECommerce", *Techno.Com*, vol. 18, no. 3, pp. 227-235, 2019. Available: 10.33633/tc.v18i3.2400.
- [15] W3C, 2022, [Geolocation API - World Wide Web Consortium \(W3C\)](#) <https://www.w3.org/TR/geolocation/#introduction> [Akses 20 Juni 2023]
- [16] D. Rombach, "Engineering software & software engineering," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1998, vol. 1521, no. 2, pp. 240–243. doi: 10.1007/3-540-49477-4_16.
- [17] Kurniawan T. A, "Pemodelan Use Case (Uml):Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 5, No. 1 pp. 77–586, 2018, DOI: 10.25126/jtiik.201851610

- [18] Tabrani M, Suhardi, Priyandanu H., “Sistem Informasi Manajemen Berbasis Website Pada Unl Studio Dengan Menggunakan Framework Codeigniter,” *Jurnal Ilmiah M-Progress.*, vol. 11, No. 1, pp. 13–21, 2021, [journal.universitassuryadarma.ac.id > index](http://journal.universitassuryadarma.ac.id/index)
- [19] G. Vaish, “Getting Started with NoSQL Your guide to the world and technology of NoSQL,” 2013. [Online]. Available: www.packtpub.com
- [20] Dhawan D., Heena, “A Study Of White Box And Black Box Software Testing,” *JETIR*, Volume 5, Issue 6, www.jetir.org (ISSN-2349-5162), 2018.
- [21] Putri SIP., Rahayu SP., Nofiyati, Web-Based Information System Design Of Inventory And Covid-19 Vaccine Distribution In Banyumas Regency, *Jurnal Teknik Informatika (JUTIF)*, Vol. 4, No. 1, Februari 2023, hlm. 1-10, DOI:<https://doi.org/10.20884/1.jutif.2023.4.1.529>
- [22] R. Maulana, “Aplikasi Pencarian Bengkel aktif dengan Google Map API berbasis Web,” *Journal of Computer and Information Technology*, Vol. 4, No. 2, February 2021, Pages 61-69, <http://e-journal.unipma.ac.id/index.php/doubleclick>.
- [23] R. S. Hamsyah, “Rancang Bangun Aplikasi Go-Ban untuk Mencari dan Memanggil Teknisi Tambal Ban Menggunakan Google Maps API,” 2018.
- [24] N. P. Kiswanto, S. D. E. Paturusi, and V. Tulenan, “Aplikasi E-Log Book Penangkapan Ikan Menggunakan Progressive Web App,” *J. Tek. Inform. vol. 15 no. 2 April. 2020, hal. 93-100*, 2020.