

# IMPLEMENTASI *PROGRESSIVE WEB APPS* (PWA) UNTUK MENINGKATKAN KINERJA DAN PERFORMA SITUS MARITIMEPRENEUR

Nyayu Maharani Adelin<sup>1</sup>, Meizano Ardhi Muhammad<sup>1</sup>, Puput Budi Wintoro<sup>1</sup>, Kaisar Akhir<sup>2,3</sup>

<sup>1</sup> Teknik Informatika, Universitas Lampung, Jl. Prof. Soemantri Brojonegoro, Bandar Lampung 35145

<sup>2</sup> Departemen Manajemen STIE Indonesia Jakarta, Jl. Pratekan No. 9A, Jakarta Timur 13220

<sup>3</sup> PT Mahakarya Maritim Indonesia, Jl. Pratekan No. 9A, Jakarta Timur 13220

## Riwayat artikel:

Received: 29 November 2022

Accepted: 29 Desember 2023

Published: 1 Januari 2024

## Keywords:

*Progressive Web App;*

*Maritimepreneur; RAD;*

*Performa;*

## Correspondent Email:

[nyayuadelin@gmail.com](mailto:nyayuadelin@gmail.com)

**Abstrak.** Maritimepreneur, penyedia jasa di industri maritim, menghadapi persaingan ketat dan harapan pengguna yang semakin tinggi terhadap pengalaman mereka. Untuk meningkatkan kualitas layanan, Maritimepreneur.com perlu peningkatan performa. Implementasi Progressive Web Applications (PWA) memakan waktu selama 60 hari, dengan metode Rapid Application Development dan 3 iterasi. Hasilnya menunjukkan PWA berhasil meningkatkan performa website sebesar 39%, mengurangi waktu pemuatan dari 12 detik menjadi 2 detik. Meskipun ada penurunan pada aksesibilitas dan SEO, peningkatan skor praktik terbaik sebesar 3% menunjukkan perbaikan secara keseluruhan. PWA menggabungkan keunggulan website dan aplikasi mobile, memberikan pengalaman pengguna yang lebih baik melalui performa cepat, responsif, dan fitur-fitur seperti dapat diunduh, akses offline, serta notifikasi. Dengan implementasi PWA, Maritimepreneur berhasil mencapai efisiensi dan kecepatan yang signifikan, memberikan pengalaman pengguna yang unggul.

**Abstract.** Maritimepreneur, a maritime industry service provider, faces fierce competition and increasing user expectations. To improve service quality, Maritimepreneur.com needed performance enhancements. Progressive Web Applications (PWAs) were integrated in a 60-day study using Rapid Application Development with 3 iterations. Results show a remarkable 39% improvement in website performance, reducing loading time from 12 to 2 seconds. Despite a minor decline in accessibility and SEO, a 3% rise in best practice scores indicates overall enhancement. PWAs merge website and mobile app strengths, providing users a superior experience with rapid performance, responsiveness, and features like downloadability, offline access, and notifications. Through PWA implementation, Maritimepreneur achieves notable efficiency and speed gains, ensuring an exceptional user experience.

## 1. PENDAHULUAN

Pengaruh fundamental teknologi informasi telah mengubah paradigma komunikasi dan interaksi masyarakat dengan dunia luar. Inovasi dalam teknologi informasi, khususnya di bidang komunikasi dan media, secara signifikan memengaruhi cara masyarakat memperoleh informasi, berinteraksi, dan mengelola

hubungan sosial. Media sosial menjadi alat strategis dalam pengembangan merek, komunikasi, dan pemasaran, digunakan oleh perusahaan dan pemerintah.[1]

Maritimepreneur, sebagai penyedia jasa di industri maritim, memanfaatkan teknologi melalui website Maritimepreneur.com. Terjadi

pergeseran signifikan menuju penggunaan perangkat mobile dan akses internet cepat, mendorong kebutuhan akan pengalaman pengguna yang lebih baik. *Progressive Web Applications* (PWA) muncul sebagai solusi menggabungkan keunggulan website dan aplikasi mobile, menawarkan performa cepat, responsif, dan fitur akses *offline*. [2]

Melalui studi kasus pada Alibaba Group, implementasi PWA pada AliExpress menunjukkan peningkatan konversi, kunjungan halaman, dan waktu yang dihabiskan. Dalam konteks *Maritimepreneur.com*, penerapan PWA berpotensi meningkatkan retensi pengguna, durasi kunjungan, dan memberikan akses *offline*. [3]

Dalam menghadapi persaingan ketat dan harapan pengguna yang tinggi, *Maritimepreneur* perlu mempertimbangkan penerapan teknologi terkini. Oleh karena itu, penelitian ini bertujuan untuk melakukan gap analysis terkait penerapan PWA pada website *Maritimepreneur.com*. Dengan mengeksplorasi potensi dampak positif PWA, penelitian ini bertujuan memberikan wawasan baru untuk meningkatkan kinerja dan performa website dalam mendukung pertumbuhan industri maritim.

## 2. TINJAUAN PUSTAKA

### 2.1 *Progressive Web Apps*

*Progressive Web Apps* (PWA) merupakan aplikasi web yang menggunakan teknologi API terkini untuk memberikan fungsionalitas, ketergantungan, dan kemudahan pemasangan yang superior. PWA memungkinkan akses universal, tanpa terbatas oleh lokasi atau perangkat, melalui basis kode tunggal. PWA dirancang agar pengembang web dapat mengubah situs web yang sudah ada menjadi pengalaman serupa aplikasi mobile tanpa memerlukan penambahan program khusus. Keunggulan PWA mencakup akses *offline*, pemasangan pada *homescreen* mirip aplikasi asli, sinkronisasi data ke server, pengiriman *push notification*, dan berbagai kemampuan lainnya. [4][5]

### 2.2 *Maritimepreneur*

*Maritimepreneur*, didirikan oleh PT Mahakarya Maritim Indonesia, bertujuan mengembangkan sumber daya alam dan industri kelautan melalui teknologi digital, acara, konsultasi, pendidikan, dan pelatihan. Merek dagang ini terdaftar di Direktorat Jenderal Kekayaan Intelektual Republik Indonesia. Melalui situs web *Maritimepreneur.com* yang berperan sebagai sumber informasi dan platform interaktif bagi pelaku industri maritim. Layanan *Maritimepreneur* melibatkan sertifikasi untuk pengelolaan sumber daya yang berkelanjutan, strategi teknologi digital, penyelenggaraan acara, konsultasi bisnis, serta pendidikan dan pelatihan yang mencakup kepemimpinan dan keterampilan teknis. *Maritimepreneur* berkomitmen untuk mendukung pembangunan bangsa dan menjaga kelestarian laut. [6]

### 2.3 *Web App Manifest*

*Web App Manifest* adalah dokumen JSON yang sederhana dan mengatur tampilan serta eksekusi aplikasi oleh pengguna. File manifest ini umumnya dinamai *manifest.json* dan diinterpretasikan oleh browser saat pengguna membuka aplikasi di dalamnya. Selanjutnya, browser akan mengambil semua sumber daya yang dibutuhkan, menampilkan konten, dan melaksanakan proses lainnya. [4]

### 2.4 *Service Worker*

*Script* yang berbeda dari *script* lain di halaman web browser disebut *service worker*. *Service Worker* pada dasarnya berfungsi sebagai server proxy yang menghubungkan aplikasi web, peramban, dan, jika tersedia, jaringan. Dengan bantuan *service worker*, konten tetap dapat ditampilkan bahkan saat jaringan *offline*, karena *service worker* mencegah permintaan jaringan, mengambil tindakan yang sesuai dengan ketersediaan jaringan, dan memperbarui aset server. [7]

### 2.5 *Ghost*

*Ghost* adalah sistem manajemen konten (CMS) open-source yang dibuat untuk memenuhi kebutuhan penulis dan penerbit untuk membuat dan mengelola konten dengan mudah. [8]

### 2.6 OneSignal

OneSignal memungkinkan pengembang dan pemilik situs web untuk menyediakan layanan notifikasi *push* untuk aplikasi mobile dan situs web. Dengan platform ini, pengembang dapat mengirim pemberitahuan *push* kepada pengguna di perangkat seluler dan desktop untuk berbagai tujuan, seperti mengingatkan mereka tentang konten baru, promosi, pembaruan, atau pemberitahuan penting lainnya. Banyak fitur notifikasi didukung oleh OneSignal, termasuk notifikasi geolokasi, pesan disegmentasi, dan analitik notifikasi. OneSignal memberikan pengalaman pengguna yang lebih interaktif dalam aplikasi dan situs web.[9]

### 2.7 Lighthouse

Lighthouse adalah alat *open-source* yang dikembangkan oleh Google untuk melakukan analisis dan evaluasi kualitas dan performa situs web. Alat ini memberikan laporan terperinci tentang berbagai elemen situs web, seperti kinerja, aksesibilitas, SEO, dan praktik terbaik. Lighthouse dapat digunakan sebagai ekstensi browser atau melalui Chrome DevTools. Ini menggunakan metrik seperti performa, aksesibilitas, praktik terbaik, SEO, dan Progressive Web Apps (PWA) untuk membantu pengembang web dan pemilik situs meningkatkan pengalaman pengguna dan kinerja situs mereka.[10]

## 3. METODE PENELITIAN

### 3.1 Waktu dan Tempat Penelitian

Penelitian dilakukan pada bulan Agustus 2023 sampai dengan Desember 2023 yang bertempat di Maritim Muda Nusantara. Pada table 1 menunjukkan agenda kegiatan penelitian yang dilakukan.

Tabel 1 Waktu Penelitian

No	Aktivitas	Agustus 2023				September 2023				Oktober 2023				November 2023			
		Minggu ke-															
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur	■	■	■													
2	Perencanaan Kebutuhan			■	■	■											
3	Desain Sistem					■	■	■	■	■							
4	Pengembangan (Iterasi 1)							■	■	■	■						
5	Pengembangan (Iterasi 2) dan Implementasi										■	■	■	■			
6	Pengembangan (Iterasi 3) dan Final Testing													■	■	■	■
7	Penulisan Laporan																■

### 3.2 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan pada penelitian ini adalah sebagai berikut.

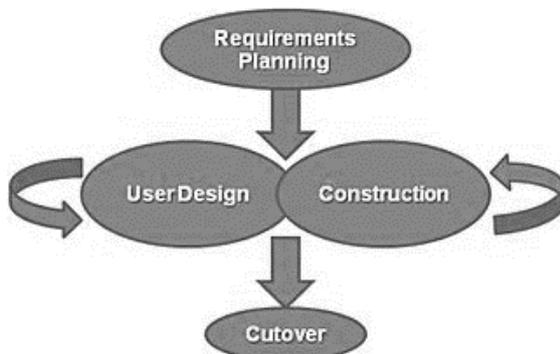
Tabel 2 Alat dan Bahan Penelitian

No	Alat / Bahan	Deskripsi
1	Laptop Lenovo IdeaPad 130-14IKB Notebook	Perangkat keras yang digunakan selama penelitian.
2	Samsung Galaxy A50	Perangkat keras yang digunakan untuk pengujian.
3	Visual Studio Code	Perangkat Lunak yang digunakan selama proses pengkodean penelitian.
4	Lighthouse by Google	Tools yang digunakan untuk menguji performa dan kinerja website.
5	React.js	Framework yang digunakan di penelitian.
6	Ghost.org	Alat publikasi <i>open-source</i> yang digunakan sebagai database untuk halaman media.
7	OneSignal	Sebagai platform untuk mengintegrasikan pemberitahuan <i>push</i> ke dalam website.
8	Jagoan Hosting	Penyedia layanan web hosting yang

		berguna untuk menyediakan infrastruktur teknis yang diperlukan untuk menjalankan website.
9	Artikel pada website <a href="https://maritimepreneur.ghost.io">https://maritimepreneur.ghost.io</a>	Sebagai sumber yang dipergunakan untuk halaman media dan notifikasi.

### 3.3 Tahapan Penelitian

Penelitian ini menggunakan metodologi SDLC *Rapid Application Development (RAD)* yang meliputi *Requirements Planning*, *User Design*, *Construction*, dan *Cutover*, kemudian selama proses penelitian dilakukan Penulisan Laporan.[11] Tahapan Penelitian menggunakan RAD ini disajikan dalam gambar berikut:[12]



Gambar 1 Tahapan Penelitian

## 4. HASIL DAN PEMBAHASAN

### 4.1 Requirements Planning

Tahapan ini dimulai dengan mengidentifikasi masalah, langkah ini kemudian menghasilkan perencanaan rinci tentang apa yang akan dibuat dan bagaimana kebutuhan tersebut akan dipenuhi. Tahap *Requirements Planning* dilakukan dengan 3 cara yaitu observasi, analisis perbandingan dan studi literatur. Hasil dari tahapan ini adalah sebagai berikut:

1. Pengguna dari website ini merupakan pelaku industri maritim yang membutuhkan informasi eksklusif dan akurat mengenai berita bisnis di dunia maritim.

2. Pengguna memerlukan sebuah situs website yang dapat diakses *offline* untuk bisa membaca berita mengenai industri maritim sekalipun tanpa jaringan internet.
3. Pengguna memerlukan berita terbaru mengenai bisnis di dunia maritim.
4. Pengguna memerlukan situs website yang dapat memberitahu pengguna apabila ada berita terbaru mengenai bisnis di dunia maritim.

Didapati pula hasil analisis menggunakan *Lighthouse* yang menunjukkan bahwa performa website sebelum implementasi PWA menggunakan *Lighthouse* berada pada skor sebesar 58%, hasil evaluasi ini menunjukkan bahwa performa pada website ini masih rendah (tidak baik) dan belum diterapkan PWA.

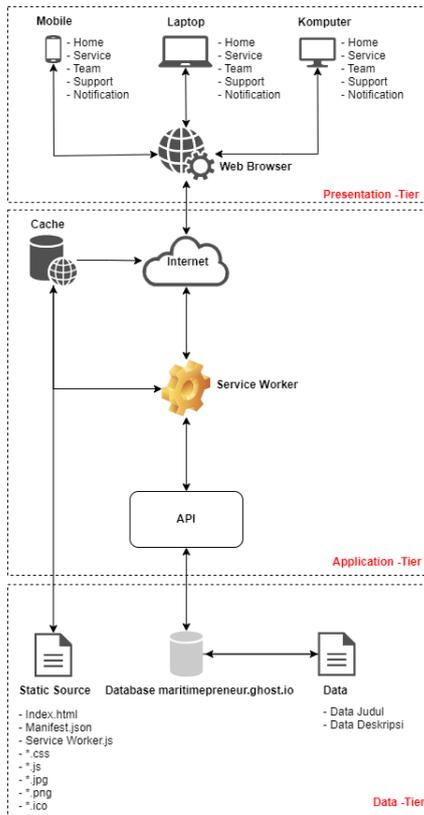


Gambar 2 Performa Website Sebelum PWA

### 4.2 User Design

Pada tahap *User Design* didapatkan hasil sebagai berikut:

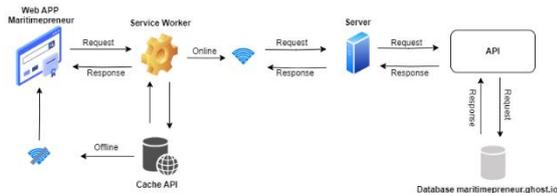
1. Arsitektur *Three Tier*



Gambar 3 Arsitektur Three tier

Gambar 3 menggambarkan arsitektur *Three-tier*, terdiri dari *presentation-tier*, *application-tier*, dan *data-tier*. *Presentation-tier* memiliki tiga *user interface* untuk akses melalui mobile, laptop, dan komputer, diakses melalui web browser. *Application-tier* sebagai *server side* mengontrol pertukaran data antara *client* dan server melalui *service worker*, yang mengelola file statis dan menyimpannya dalam *cache browser client*. *Data-tier* berisi basis data di database server dan *static resource* di cache browser pengguna, termasuk judul dan deskripsi berita dalam database server, serta file index, webmanifest, css, js, dan format gambar di *static resource* pada cache browser.

## 2. Arsitektur PWA



Gambar 4 Arsitektur PWA

Pada gambar 4, *service worker* akan pertama kali mengecek respons jaringan; jika itu

berhasil, data akan dikembalikan ke halaman web. Jika tidak, *service worker* akan mengembalikannya dari cache.

## 4.3 Construction

Tahap ini fokus pada pembangunan aplikasi berdasarkan desain yang telah dibuat pada tahap sebelumnya. Gambar berikut menunjukkan *source code* system.

### 1. Penambahan Halaman Media

```

1 import React, { lazy, Suspense } from 'react';
2 import Footer from '../component/Footer';
3 import NavBar from '../component/Navbar';
4
5 const HeaderMedia = lazy(() => import('../component/Mediapage/HeaderMedia'));
6 const ListArtikel = lazy(() => import('../component/Mediapage/ListArtikel'));
7
8
9 const MediaPage = () => {
10
11   return (
12     <>
13       <NavBar />
14       <Suspense fallback=<div>Loading...</div>>
15       <HeaderMedia />
16       <h1 className="text-center my-20 font-bold text-primary-color text-3xl">Media</h1>
17       <ListArtikel />
18       <Footer />
19     </Suspense>
20   </>
21 );
22 }
23
24 export default MediaPage;

```

Gambar 5 Source code Halaman Media

```

1 import GhostContentAPI from '@tryghost/content-api';
2
3 const api = new GhostContentAPI({
4   url: 'https://maritimepreneur.ghost.io',
5   key: 'b49aaf79b4cd8bd9df29e2e9a7',
6   version: 'v5.0',
7 });
8
9 async function getAllPosts() {
10   try {
11     const posts = await api.posts.browse({ limit: 'all', include: 'tags,authors' });
12     return posts;
13   } catch (error) {
14     console.error(error);
15     return [];
16   }
17 }
18
19 async function getPostById(postId) {
20   try {
21     const post = await api.posts.read({ id: postId, include: 'tags,authors' });
22     return post;
23   } catch (error) {
24     console.error(error);
25     return null;
26   }
27 }
28
29 export { getAllPosts, getPostById };

```

Gambar 6 Source Code Fetch API Halaman Media

```

1 import React, { useEffect, useState } from 'react';
2 import { Link } from 'react-router-dom';
3 import { getAllPosts } from '../utils/media';
4 import './Media.css';
5
6 const ListArtikel = () => {
7   const [posts, setPosts] = useState([]);
8
9   useEffect(() => {
10    getAllPosts()
11      .then((posts) => {
12        setPosts(posts);
13      })
14      .catch((err) => {
15        console.error(err);
16      });
17   }, []);
18
19   return (
20     <div className="card-container">
21       <posts.map(post) => (
22         <div className="article-container" key={post.id}>
23           <link to="/artikel/${post.id}" className="article-link">
24             <div className="article-header">
25               <h3>Header artikel </h3>
26             </div>
27             <img src={post.feature_image} alt="Avatar" className="article-image" />
28             <div className="article-title">
29               <h2>{post.title}</h2>
30             </div>
31             <div className="article-content-container">
32               <h3>Isi artikel </h3>
33             </div>
34           </div>
35         </Link>
36       </div>
37     </div>
38   );
39 };
40
41 export default ListArtikel;

```

Gambar 7 Source Code List Artikel Halaman Media

```

1 // Artikel.js
2 import React, { useEffect, useState, useRef } from 'react';
3 import { useState } from 'react-router-dom';
4 import { getAllPosts } from '../utils/media';
5 import Header from './components/Header';
6 import Navbar from './components/Navbar';
7
8 const Artikel = () => {
9   const [artikel, setArtikel] = useState(null);
10   const [artikel, setData] = useState(null);
11
12   useEffect(() => {
13     // Panggil data artikel berdasarkan arsitektur (suaing atau id)
14     getAllPosts()
15       .then((data) => {
16         setData(data);
17       })
18       .catch((err) => {
19         console.error(err);
20       });
21   }, [isLoading]);
22
23   if (artikel) {
24     return <div>...</div>;
25   }
26
27   return (
28     <div>...</div>
29   );
30
31   <div>...</div>
32   <div>...</div>
33   <div>...</div>
34   <div>...</div>
35   <div>...</div>
36   <div>...</div>
37   <div>...</div>
38   <div>...</div>
39   <div>...</div>
40   <div>...</div>
41   <div>...</div>
42   <div>...</div>
43   <div>...</div>
44   <div>...</div>
45   <div>...</div>
46   <div>...</div>
47   <div>...</div>
48   <div>...</div>
49   <div>...</div>
50   <div>...</div>
51   <div>...</div>
52   <div>...</div>
53   <div>...</div>
54   <div>...</div>
55   <div>...</div>
56   <div>...</div>
57   <div>...</div>
58   <div>...</div>
59   <div>...</div>
60   <div>...</div>
61   <div>...</div>
62   <div>...</div>
63   <div>...</div>
64   <div>...</div>
65   <div>...</div>
66   <div>...</div>
67   <div>...</div>
68   <div>...</div>
69   <div>...</div>
70   <div>...</div>
71   <div>...</div>
72   <div>...</div>
73   <div>...</div>
74   <div>...</div>
75   <div>...</div>
76   <div>...</div>
77   <div>...</div>
78   <div>...</div>
79   <div>...</div>
80   <div>...</div>
81   <div>...</div>
82   <div>...</div>
83   <div>...</div>
84   <div>...</div>
85   <div>...</div>
86   <div>...</div>
87   <div>...</div>
88   <div>...</div>
89   <div>...</div>
90   <div>...</div>
91   <div>...</div>
92   <div>...</div>
93   <div>...</div>
94   <div>...</div>
95   <div>...</div>
96   <div>...</div>
97   <div>...</div>
98   <div>...</div>
99   <div>...</div>
100  <div>...</div>

```

Gambar 8 Source Code Detail Artikel Halaman Media

## 2. Implementasi Progressive Web App

```

1 {
2   "name": "Maritimepreneur",
3   "description": "Building the Nations Sustaining Our Oceans",
4   "short_name": "Maritimepreneur",
5   "icons": [
6     {
7       "src": "/android-chrome-192x192.png",
8       "sizes": "192x192",
9       "type": "image/png",
10      "purpose": "any"
11    },
12    {
13      "src": "/android-chrome-512x512.png",
14      "sizes": "512x512",
15      "type": "image/png",
16      "purpose": "maskable"
17    }
18  ],
19  "start_url": ".",
20  "theme_color": "#143F83",
21  "background_color": "#ffffff",
22  "display": "standalone",
23  "splash_pages": null
24 }

```

Gambar 9 Source Code Web App Manifest

```

1 if ("serviceWorker" in navigator) {
2   window.addEventListener("load", async function () {
3     navigator.serviceWorker.register("../service-worker.js").then(
4       function (registration) {
5         // Registration was successful
6         console.log("ServiceWorker registration successful with scope: ", registration.scope);
7       },
8       function (err) {
9         // registration failed :(
10        console.log("ServiceWorker registration failed: ", err);
11      });
12   });
13 }

```

Gambar 10 Source Code Registrasi Service Worker

```

1 self.addEventListener("install", (event) => {
2   console.log("Installing service worker...");
3
4   event.waitUntil(
5     caches
6       .open(CACHE_NAME)
7       .then((cache) => cache.addAll(assetsToCache))
8       .catch((error) => {
9         console.error("Fatal message service worker!", error);
10      })
11   );
12 });
13
14 // es-lint-disable-next-line no-restricted-globals
15 self.addEventListener("activate", (event) => {
16   console.log("Activating service worker...");
17
18   event.waitUntil(
19     caches
20       .keys()
21       .then((cacheNames) => Promise.all(cacheNames.map((name) => name !== CACHE_NAME ? notFound(name) : caches.delete(cacheName)))
22       .catch((error) => {
23         console.error("Fatal message service worker!", error);
24       })
25   );
26 });
27
28 // es-lint-disable-next-line no-restricted-globals
29 self.addEventListener("fetch", (event) => {
30   event.respondWith(
31     caches
32       .match(event.request)
33       .then((response) => {
34         if (response) {
35           return response;
36         }
37         return fetch(event.request).then((response) => {
38           if (response.status !== 200 || response.type !== "basic") {
39             return response;
40           }
41           const responseToCache = response.clone();
42           caches.open(CACHE_NAME).then((cache) => {
43             cache.put(event.request, responseToCache);
44           });
45           return response;
46         });
47       })
48       .catch((error) => {
49         console.error("Fatal message service worker!", error);
50       })
51   );
52 });
53
54 self.addEventListener("push", (event) => {
55   const options = {
56     body: event.data.text(),
57     icons: ["/img-no-icon/icon.png"],
58   };
59
60   event.waitUntil(self.registration.showNotification("Jurnal Maritim", options));
61 });
62
63 console.log("Notifikasi diterima", event);
64 });

```

Gambar 11 Source Code Implementasi Service Worker

```

1 const version = "1.0.0";
2 const CACHE_NAME = "MediaMaritim-${version}";
3 const assetsToCache = [
4   "/",
5   "/index.js",
6   "/static/css/main.a5767e4.css",
7   "/static/js/main.546a0d19.js",
8   "/static/js/291.f19f7120.chunk.js",
9   "/static/js/989.67f34d1f.chunk.js",
10  "/static/js/980.a25f51ff.chunk.js",
11  "/static/js/406.532552c9.chunk.js",
12  "/static/js/522.f3982266.chunk.js",
13  "/static/js/819.1dfc266a.chunk.js",
14  "/static/js/240.4329fbaa.chunk.js",
15  "/static/js/247.49fd8052.chunk.js",
16  "/static/js/469.813f3883.chunk.js",
17  "/static/js/932.5e62459b.chunk.js",
18  "/static/js/977.325e2e61.chunk.js",
19  "/static/js/524.d3a67361.chunk.js",
20  "/static/css/463.e6003347.chunk.css",
21  "/static/js/463.17760024.chunk.js",
22  "/static/js/602.4b651718.chunk.js",
23  "/static/media/Indonesian Ocean Policy.F589d2155bac7b689066.pdf",
24  "/static/media/Blue Economy Company Index - Proposal.91abb98add99a5db8f6.pdf",
25  "/static/media/Company Profile.29c573fc191034048982.pdf",
26  "/static/media/bg-peraturan.47c0816984556e31710.png",
27  "/static/media/Indonesia - Law of the Sea.7eeffbf63e45b3ad1256.pdf",
28  "/static/media/Blue Economy Company Index - Research Article.21342f4649778e5137c1.pdf",
29  "/static/media/bg-home.5c49488c80d58b5520b.png",
30  "/static/media/bt-et.6a8200636ce9f6a6bb7b.png",
31  "/static/media/log1.d87dbacc7a347e08205.png",
32  "/static/media/bcedex.facc68be4c640cf96ea.png",

```

Gambar 12 Source Code Implementasi Cache API

### 3. Penambahan *Push Notification*

```

1  importScripts("https://cdn.onesignal.com/sdks/web/v16/OneSignalSDK.sw.js");
2
3  <script src="https://cdn.onesignal.com/sdks/web/v16/OneSignalSDK.page.js" defer></script>
4  <script>
5  window.OneSignalDeferred = window.OneSignalDeferred || [];
6  OneSignalDeferred.push(function (OneSignal) {
7    OneSignal.init({
8      appId: "6a565311-bb5f-46f4-9b7e-3d147414c129",
9      safari_web_id: "web.onesignal.auto.49889676-9f7f-4916-ae63-fd94958742a1",
10     notifyButton: {
11       enable: true,
12     },
13     allowLocalhostAsSecureOrigin: true,
14     serviceWorkerPath: "../OneSignalSDKWorker.js",
15   });
16 });
17 </script>

```

Gambar 13 Source Code Penambahan *Push Notification*

### 4.4 Cutover

Tahap terakhir adalah *cutover*, di mana aplikasi yang telah dikonstruksi dipindahkan dari lingkungan pengembangan ke lingkungan produksi. Proses ini melibatkan uji coba dan implementasi sistem secara penuh. Pada gambar menunjukkan hasil dari beberapa tampilan website yang telah selesai dikerjakan.

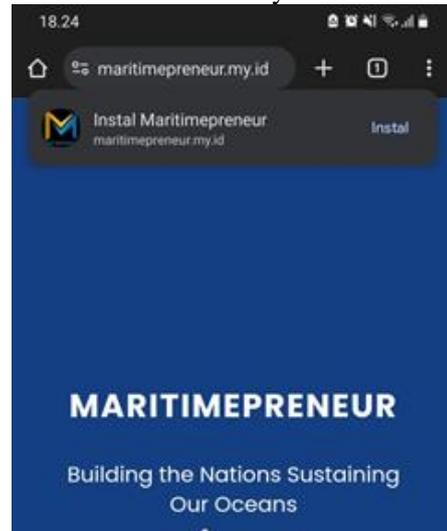
#### 1. Halaman Media



Gambar 14 Tampilan Halaman Media

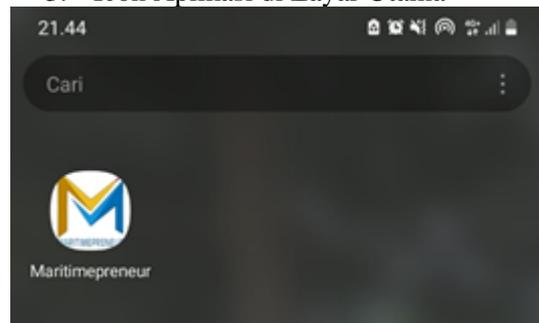


#### 2. *Install PWA* ke Layar Utama



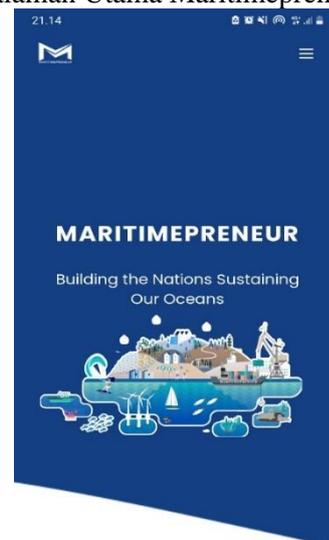
Gambar 15 Tampilan Instalasi *PWA* ke Layar Utama

#### 3. Icon Aplikasi di Layar Utama



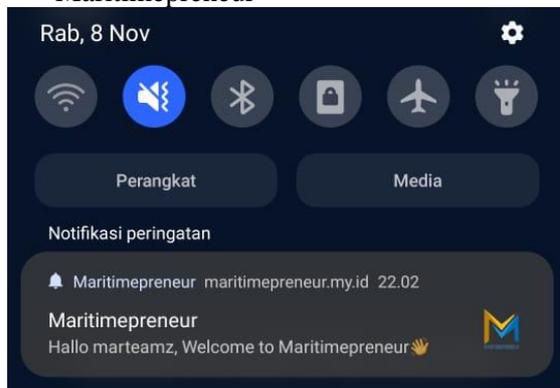
Gambar 16 Icon Aplikasi pada Layar Utama Android

#### 4. Halaman Utama *Maritimepreneur*



Gambar 17 Tampilan Halaman *Maritimepreneur* pada Android

5. *Push Notification* pada *Maritimepreneur*



Gambar 18 Tampilan *Push Notification* yang berhasil diterima

4.5 *Lighthouse Testing*

Pengujian kriteria PWA menggunakan *Lighthouse* didapatkan hasil seperti ditampilkan pada Tabel di bawah ini:

Tabel 3 Hasil Pengujian Baseline PWA Checklist

N o.	Variabel	Sebelum Implementasi	Setelah Implementasi
<b>Installable</b>			
1.	Web app manifest atau service worker memenuhi persyaratan instalasi aplikasi	Tidak Terpenuhi	Terpenuhi
<b>PWA Optimized</b>			
2.	Mendaftarkan service worker yang mengontrol halaman dan start_url	Tidak Terpenuhi	Terpenuhi
3.	Konfigurasi untuk layar pembuka khusus	Terpenuhi	Terpenuhi
4.	Menyetel warna tema untuk kolom URL	Terpenuhi	Terpenuhi
5.	Ukuran konten tepat untuk area pandang	Tidak Terpenuhi	Terpenuhi
6.	Memiliki tag <meta name="viewpo	Terpenuhi	Terpenuhi

	rt"> dengan width atau initial-scale		
7.	Manifest memiliki maskable icon	Tidak Terpenuhi	Terpenuhi
<b>Pengujian Manual</b>			
8.	Situs dapat berjalan lintas browser (Chrome, Edge, dan Safari)	Terpenuhi	Terpenuhi
9.	Transisi halaman seperti tidak diblokir di jaringan	Terpenuhi	Terpenuhi
10	Setiap halaman memiliki URL	Terpenuhi	Terpenuhi

Sedangkan untuk hasil pengujian performa yang didapatkan menggunakan *tools Lighthouse* dapat dilihat pada gambar dibawah ini:



Gambar 19 Hasil Pengujian Performa Website

4.6 **Capaian Penelitian**

Penelitian ini berlangsung selama 60 hari dengan total 3 iterasi dan pada setiap iterasi menunjukkan perubahan pada website yang dapat dilihat pada grafik dibawah ini:



Gambar 20 Grafik Peningkatan Skor

5. **KESIMPULAN**

- Website *Maritimepreneur* dikembangkan sebanyak 3 kali iterasi sehingga didapatkan halaman baru, diimplementasikan *Progressive Web Apps*, dan diterapkan *push notification*.
- Penerapan PWA pada website *Maritimepreneur* telah berhasil memenuhi semua kriteria PWA. Termasuk kemampuan untuk diakses melalui layar

utama smartphone, berfungsi secara *offline* di berbagai peramban, dan meningkatkan performa website dibandingkan sebelum penggunaan PWA.

- c. Implementasi *Progressive Web App* (PWA) pada website Maritimpreneur mengakibatkan peningkatan performa sebesar 39%, membawa situs dari 58% sebelum implementasi PWA menjadi 97% setelah implementasi PWA. Meskipun terdapat penurunan kecil pada matriks aksesibilitas (-3%) dan SEO (-1%), peningkatan skor praktik terbaik sebesar 3% (dari 83% menjadi 86%) menunjukkan perbaikan dalam aspek-aspek terkait praktik terbaik. Secara keseluruhan, dampak positif PWA terutama terlihat pada peningkatan performa dan praktik terbaik, sementara pemantauan lebih lanjut diperlukan untuk memahami implikasi perubahan pada aksesibilitas dan SEO.
- d. Dengan adanya PWA pada website Maritimpreneur, waktu pemuatan website dipangkas drastis dari 12 detik menjadi hanya 2 detik. Peningkatan tersebut mencerminkan efektivitas PWA dalam meningkatkan kecepatan dan efisiensi situs, memberikan pengalaman pengguna yang lebih baik.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

#### DAFTAR PUSTAKA

- [1] A. M. A. Saputra, L. P. I. Kharisma, A. A. Rizal, M. I. Burhan, and N. W. Purnawati, *Teknologi Informasi: Peranan TI dalam Berbagai Bidang*. Jambi: PT. Sonpedia Publishing Indonesia, 2023.
- [2] J. M. Wargo, *Learning Progressive Web Apps*. Addison-Wesley Professional, 2020.
- [3] Chrome DevRel, "aliexpress study case," 2016. Accessed: Aug. 31, 2023. [Online]. Available: <https://web.dev/aliexpress/>
- [4] C. Love, *Progressive Web Application Development by Example: Develop fast, reliable, and engaging user experiences for the web*. Packt Publishing Ltd, 2018.
- [5] M. Ardhi Muhammad, J. Sumantri Brojonegoro No, and B. Lampung, "Mobile Dictionary Aksara Lampung Berbasis Teknologi SPA (Single Webpage Application)," 2017.
- [6] PT. Mahakarya Maritim Indonesia, "Service: Maritimpreneur." Accessed: Sep. 01, 2023. [Online]. Available: <http://maritimpreneur.com/services>
- [7] D. Hume, *Progressive Web Apps*. Simon and Schuster, 2017.
- [8] "Ghost: About," Ghost.org. Accessed: Oct. 25, 2023. [Online]. Available: <https://ghost.org/about/>
- [9] L. Safitri, "Sistem Informasi Jadwal Pengiriman Gas LPG Berbasis Push Notification Android," *Bangkit Indonesia*, vol. XI, no. 01, 2022.
- [10] "Documentation: Lighthouse," Chrome Developers. Accessed: Aug. 26, 2023. [Online]. Available: <https://developer.chrome.com/docs/lighthouse/overview/>
- [11] G. Revanol, W. E. Sulistyono, H. D. Septama, Y. Mulyani, and M. Pratama, "Sistem Informasi Portal Prodi Modul Pendaftaran Seminar Akademik di Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung menggunakan metode Rapid Application Development (RAD)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 1, Jan. 2023, doi: 10.23960/jitet.v11i1.2864.
- [12] F. N. Hasanah and R. S. Untari, *Buku Ajar Rekayasa Perangkat Lunak*, vol. 1. 2020.