

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

Nyayu Maharani Adelin¹, Meizano Ardhi Muhammad¹, Puput Budi Wintoro¹, Kaisar Akhir^{2,3}

¹ Teknik Informatika, Universitas Lampung, Jl. Prof. Soemantri Brojonegoro, Bandar Lampung 35145

² Departemen Manajemen STIE Indonesia Jakarta, Jl. Pratekan No. 9A, Jakarta Timur 13220

³ 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

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 Maritimpreneur.com, penerapan PWA berpotensi meningkatkan retensi pengguna, durasi kunjungan, dan memberikan akses *offline*. [3]

Dalam menghadapi persaingan ketat dan harapan pengguna yang tinggi, Maritimpreneur perlu mempertimbangkan penerapan teknologi terkini. Oleh karena itu, penelitian ini bertujuan untuk melakukan gap analysis terkait penerapan PWA pada website Maritimpreneur.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 Maritimpreneur

Maritimpreneur, 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 Maritimpreneur.com yang berperan sebagai sumber informasi dan platform interaktif bagi pelaku industri maritim. Layanan Maritimpreneur 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. Maritimpreneur 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.

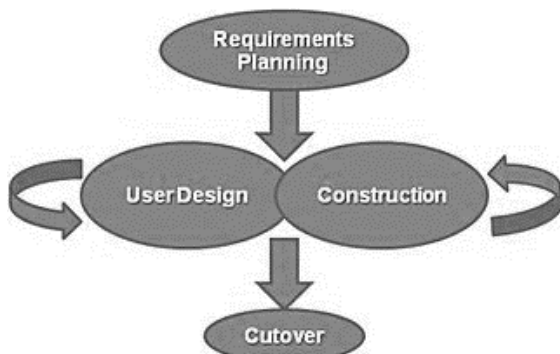
Tabel 2 Alat dan Bahan Penelitian

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	<i>Tools</i> 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 https://maritimepreneur.ghost.io	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.

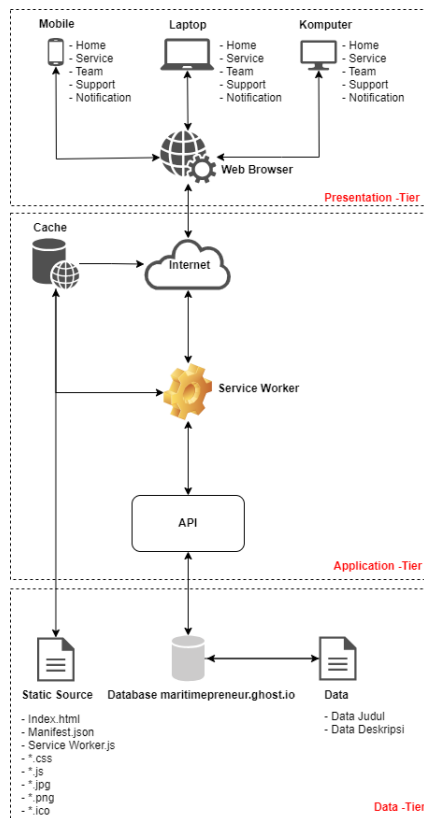


Gambar 2 Performa Website Sebelum PWA

4.2 User Design

Pada tahap *User Design* didapatkan hasil sebagai berikut:

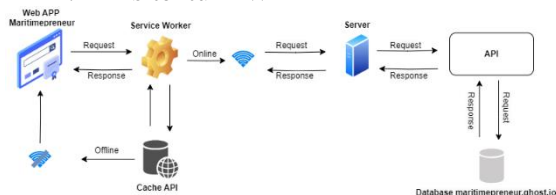
1. Arsitektur *Three Tier*



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       <div className='text-center my-20 font-bold text-primary-color text-3xl'>Media</div>
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://maritimpreneur.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               { /* Header artikel */ }
26             </div>
27             <img src={post.feature_image} alt="Avatar" className="article-image" />
28             <div className="article-title">
29               {post.title}
30             </div>
31             <div className="article-content-container">
32               { /* Isi artikel */ }
33             </div>
34           </div>
35         </Link>
36       ))}
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 { useParams } from 'react-router-dom';
4 import { getPostById } from '../utils/media';
5 import Header from './components/Header';
6 import Footer from './components/Footer';
7
8 const Artikel = () => {
9   const [postId] = useParams();
10   const [article, setArticle] = useState(null);
11   const [loading, setLoading] = useState(true);
12
13   useEffect(() => {
14     // Mengambil data artikel berdasarkan articleId (slug atau id)
15     getPostById(postId)
16       .then(data => {
17         setArticle(data);
18       })
19       .catch((err) => {
20         console.error(err);
21       });
22   }, [postId]);
23
24   if (article) {
25     return <div>...</div>;
26   }
27
28   return (
29     <div>...</div>
30   );
31
32   <div>...</div>
33
34   <div>...</div>
35
36   <div>...</div>
37
38   <div>...</div>
39
40   <div>...</div>
41
42   <div>...</div>
43
44   <div>...</div>
45
46   <div>...</div>
47
48   <div>...</div>
49
50   <div>...</div>
51
52   <div>...</div>
53
54   <div>...</div>
55
56   <div>...</div>
57
58   <div>...</div>
59
60   <div>...</div>
61
62   <div>...</div>
63
64   <div>...</div>
65
66   <div>...</div>
67
68   <div>...</div>
69
70   <div>...</div>
71
72   <div>...</div>
73
74   <div>...</div>
75
76   <div>...</div>
77
78   <div>...</div>
79
80   <div>...</div>
81
82   <div>...</div>
83
84   <div>...</div>
85
86   <div>...</div>
87
88   <div>...</div>
89
90   <div>...</div>
91
92   <div>...</div>
93
94   <div>...</div>
95
96   <div>...</div>
97
98   <div>...</div>
99
100  <div>...</div>
101
102  <div>...</div>
103
104  <div>...</div>
105
106  <div>...</div>
107
108  <div>...</div>
109
110  <div>...</div>
111
112  <div>...</div>
113
114  <div>...</div>
115
116  <div>...</div>
117
118  <div>...</div>
119
120  <div>...</div>
121
122  <div>...</div>
123
124  <div>...</div>
125
126  <div>...</div>
127
128  <div>...</div>
129
130  <div>...</div>
131
132  <div>...</div>
133
134  <div>...</div>
135
136  <div>...</div>
137
138  <div>...</div>
139
140  <div>...</div>
141
142  <div>...</div>
143
144  <div>...</div>
145
146  <div>...</div>
147
148  <div>...</div>
149
150  <div>...</div>
151
152  <div>...</div>
153
154  <div>...</div>
155
156  <div>...</div>
157
158  <div>...</div>
159
160  <div>...</div>
161
162  <div>...</div>
163
164  <div>...</div>
165
166  <div>...</div>
167
168  <div>...</div>
169
170  <div>...</div>
171
172  <div>...</div>
173
174  <div>...</div>
175
176  <div>...</div>
177
178  <div>...</div>
179
180  <div>...</div>
181
182  <div>...</div>
183
184  <div>...</div>
185
186  <div>...</div>
187
188  <div>...</div>
189
190  <div>...</div>
191
192  <div>...</div>
193
194  <div>...</div>
195
196  <div>...</div>
197
198  <div>...</div>
199
200  <div>...</div>
201
202  <div>...</div>
203
204  <div>...</div>
205
206  <div>...</div>
207
208  <div>...</div>
209
210  <div>...</div>
211
212  <div>...</div>
213
214  <div>...</div>
215
216  <div>...</div>
217
218  <div>...</div>
219
220  <div>...</div>
221
222  <div>...</div>
223
224  <div>...</div>
225
226  <div>...</div>
227
228  <div>...</div>
229
230  <div>...</div>
231
232  <div>...</div>
233
234  <div>...</div>
235
236  <div>...</div>
237
238  <div>...</div>
239
240  <div>...</div>
241
242  <div>...</div>
243
244  <div>...</div>
245
246  <div>...</div>
247
248  <div>...</div>
249
250  <div>...</div>
251
252  <div>...</div>
253
254  <div>...</div>
255
256  <div>...</div>
257
258  <div>...</div>
259
260  <div>...</div>
261
262  <div>...</div>
263
264  <div>...</div>
265
266  <div>...</div>
267
268  <div>...</div>
269
270  <div>...</div>
271
272  <div>...</div>
273
274  <div>...</div>
275
276  <div>...</div>
277
278  <div>...</div>
279
280  <div>...</div>
281
282  <div>...</div>
283
284  <div>...</div>
285
286  <div>...</div>
287
288  <div>...</div>
289
290  <div>...</div>
291
292  <div>...</div>
293
294  <div>...</div>
295
296  <div>...</div>
297
298  <div>...</div>
299
300  <div>...</div>
301
302  <div>...</div>
303
304  <div>...</div>
305
306  <div>...</div>
307
308  <div>...</div>
309
310  <div>...</div>
311
312  <div>...</div>
313
314  <div>...</div>
315
316  <div>...</div>
317
318  <div>...</div>
319
320  <div>...</div>
321
322  <div>...</div>
323
324  <div>...</div>
325
326  <div>...</div>
327
328  <div>...</div>
329
330  <div>...</div>
331
332  <div>...</div>
333
334  <div>...</div>
335
336  <div>...</div>
337
338  <div>...</div>
339
340  <div>...</div>
341
342  <div>...</div>
343
344  <div>...</div>
345
346  <div>...</div>
347
348  <div>...</div>
349
350  <div>...</div>
351
352  <div>...</div>
353
354  <div>...</div>
355
356  <div>...</div>
357
358  <div>...</div>
359
360  <div>...</div>
361
362  <div>...</div>
363
364  <div>...</div>
365
366  <div>...</div>
367
368  <div>...</div>
369
370  <div>...</div>
371
372  <div>...</div>
373
374  <div>...</div>
375
376  <div>...</div>
377
378  <div>...</div>
379
380  <div>...</div>
381
382  <div>...</div>
383
384  <div>...</div>
385
386  <div>...</div>
387
388  <div>...</div>
389
390  <div>...</div>
391
392  <div>...</div>
393
394  <div>...</div>
395
396  <div>...</div>
397
398  <div>...</div>
399
400  <div>...</div>
401
402  <div>...</div>
403
404  <div>...</div>
405
406  <div>...</div>
407
408  <div>...</div>
409
410  <div>...</div>
411
412  <div>...</div>
413
414  <div>...</div>
415
416  <div>...</div>
417
418  <div>...</div>
419
420  <div>...</div>
421
422  <div>...</div>
423
424  <div>...</div>
425
426  <div>...</div>
427
428  <div>...</div>
429
430  <div>...</div>
431
432  <div>...</div>
433
434  <div>...</div>
435
436  <div>...</div>
437
438  <div>...</div>
439
440  <div>...</div>
441
442  <div>...</div>
443
444  <div>...</div>
445
446  <div>...</div>
447
448  <div>...</div>
449
450  <div>...</div>
451
452  <div>...</div>
453
454  <div>...</div>
455
456  <div>...</div>
457
458  <div>...</div>
459
460  <div>...</div>
461
462  <div>...</div>
463
464  <div>...</div>
465
466  <div>...</div>
467
468  <div>...</div>
469
470  <div>...</div>
471
472  <div>...</div>
473
474  <div>...</div>
475
476  <div>...</div>
477
478  <div>...</div>
479
480  <div>...</div>
481
482  <div>...</div>
483
484  <div>...</div>
485
486  <div>...</div>
487
488  <div>...</div>
489
490  <div>...</div>
491
492  <div>...</div>
493
494  <div>...</div>
495
496  <div>...</div>
497
498  <div>...</div>
499
500  <div>...</div>
501
502  <div>...</div>
503
504  <div>...</div>
505
506  <div>...</div>
507
508  <div>...</div>
509
510  <div>...</div>
511
512  <div>...</div>
513
514  <div>...</div>
515
516  <div>...</div>
517
518  <div>...</div>
519
520  <div>...</div>
521
522  <div>...</div>
523
524  <div>...</div>
525
526  <div>...</div>
527
528  <div>...</div>
529
530  <div>...</div>
531
532  <div>...</div>
533
534  <div>...</div>
535
536  <div>...</div>
537
538  <div>...</div>
539
540  <div>...</div>
541
542  <div>...</div>
543
544  <div>...</div>
545
546  <div>...</div>
547
548  <div>...</div>
549
550  <div>...</div>
551
552  <div>...</div>
553
554  <div>...</div>
555
556  <div>...</div>
557
558  <div>...</div>
559
560  <div>...</div>
561
562  <div>...</div>
563
564  <div>...</div>
565
566  <div>...</div>
567
568  <div>...</div>
569
570  <div>...</div>
571
572  <div>...</div>
573
574  <div>...</div>
575
576  <div>...</div>
577
578  <div>...</div>
579
580  <div>...</div>
581
582  <div>...</div>
583
584  <div>...</div>
585
586  <div>...</div>
587
588  <div>...</div>
589
590  <div>...</div>
591
592  <div>...</div>
593
594  <div>...</div>
595
596  <div>...</div>
597
598  <div>...</div>
599
600  <div>...</div>
601
602  <div>...</div>
603
604  <div>...</div>
605
606  <div>...</div>
607
608  <div>...</div>
609
610  <div>...</div>
611
612  <div>...</div>
613
614  <div>...</div>
615
616  <div>...</div>
617
618  <div>...</div>
619
620  <div>...</div>
621
622  <div>...</div>
623
624  <div>...</div>
625
626  <div>...</div>
627
628  <div>...</div>
629
630  <div>...</div>
631
632  <div>...</div>
633
634  <div>...</div>
635
636  <div>...</div>
637
638  <div>...</div>
639
640  <div>...</div>
641
642  <div>...</div>
643
644  <div>...</div>
645
646  <div>...</div>
647
648  <div>...</div>
649
650  <div>...</div>
651
652  <div>...</div>
653
654  <div>...</div>
655
656  <div>...</div>
657
658  <div>...</div>
659
660  <div>...</div>
661
662  <div>...</div>
663
664  <div>...</div>
665
666  <div>...</div>
667
668  <div>...</div>
669
670  <div>...</div>
671
672  <div>...</div>
673
674  <div>...</div>
675
676  <div>...</div>
677
678  <div>...</div>
679
680  <div>...</div>
681
682  <div>...</div>
683
684  <div>...</div>
685
686  <div>...</div>
687
688  <div>...</div>
689
690  <div>...</div>
691
692  <div>...</div>
693
694  <div>...</div>
695
696  <div>...</div>
697
698  <div>...</div>
699
700  <div>...</div>
701
702  <div>...</div>
703
704  <div>...</div>
705
706  <div>...</div>
707
708  <div>...</div>
709
710  <div>...</div>
711
712  <div>...</div>
713
714  <div>...</div>
715
716  <div>...</div>
717
718  <div>...</div>
719
720  <div>...</div>
721
722  <div>...</div>
723
724  <div>...</div>
725
726  <div>...</div>
727
728  <div>...</div>
729
730  <div>...</div>
731
732  <div>...</div>
733
734  <div>...</div>
735
736  <div>...</div>
737
738  <div>...</div>
739
740  <div>...</div>
741
742  <div>...</div>
743
744  <div>...</div>
745
746  <div>...</div>
747
748  <div>...</div>
749
750  <div>...</div>
751
752  <div>...</div>
753
754  <div>...</div>
755
756  <div>...</div>
757
758  <div>...</div>
759
760  <div>...</div>
761
762  <div>...</div>
763
764  <div>...</div>
765
766  <div>...</div>
767
768  <div>...</div>
769
770  <div>...</div>
771
772  <div>...</div>
773
774  <div>...</div>
775
776  <div>...</div>
777
778  <div>...</div>
779
780  <div>...</div>
781
782  <div>...</div>
783
784  <div>...</div>
785
786  <div>...</div>
787
788  <div>...</div>
789
790  <div>...</div>
791
792  <div>...</div>
793
794  <div>...</div>
795
796  <div>...</div>
797
798  <div>...</div>
799
800  <div>...</div>
801
802  <div>...</div>
803
804  <div>...</div>
805
806  <div>...</div>
807
808  <div>...</div>
809
810  <div>...</div>
811
812  <div>...</div>
813
814  <div>...</div>
815
816  <div>...</div>
817
818  <div>...</div>
819
820  <div>...</div>
821
822  <div>...</div>
823
824  <div>...</div>
825
826  <div>...</div>
827
828  <div>...</div>
829
830  <div>...</div>
831
832  <div>...</div>
833
834  <div>...</div>
835
836  <div>...</div>
837
838  <div>...</div>
839
840  <div>...</div>
841
842  <div>...</div>
843
844  <div>...</div>
845
846  <div>...</div>
847
848  <div>...</div>
849
850  <div>...</div>
851
852  <div>...</div>
853
854  <div>...</div>
855
856  <div>...</div>
857
858  <div>...</div>
859
860  <div>...</div>
861
862  <div>...</div>
863
864  <div>...</div>
865
866  <div>...</div>
867
868  <div>...</div>
869
870  <div>...</div>
871
872  <div>...</div>
873
874  <div>...</div>
875
876  <div>...</div>
877
878  <div>...</div>
879
880  <div>...</div>
881
882  <div>...</div>
883
884  <div>...</div>
885
886  <div>...</div>
887
888  <div>...</div>
889
890  <div>...</div>
891
892  <div>...</div>
893
894  <div>...</div>
895
896  <div>...</div>
897
898  <div>...</div>
899
900  <div>...</div>
901
902  <div>...</div>
903
904  <div>...</div>
905
906  <div>...</div>
907
908  <div>...</div>
909
910  <div>...</div>
911
912  <div>...</div>
913
914  <div>...</div>
915
916  <div>...</div>
917
918  <div>...</div>
919
920  <div>...</div>
921
922  <div>...</div>
923
924  <div>...</div>
925
926  <div>...</div>
927
928  <div>...</div>
929
930  <div>...</div>
931
932  <div>...</div>
933
934  <div>...</div>
935
936  <div>...</div>
937
938  <div>...</div>
939
940  <div>...</div>
941
942  <div>...</div>
943
944  <div>...</div>
945
946  <div>...</div>
947
948  <div>...</div>
949
950  <div>...</div>
951
952  <div>...</div>
953
954  <div>...</div>
955
956  <div>...</div>
957
958  <div>...</div>
959
960  <div>...</div>
961
962  <div>...</div>
963
964  <div>...</div>
965
966  <div>...</div>
967
968  <div>...</div>
969
970  <div>...</div>
971
972  <div>...</div>
973
974  <div>...</div>
975
976  <div>...</div>
977
978  <div>...</div>
979
980  <div>...</div>
981
982  <div>...</div>
983
984  <div>...</div>
985
986  <div>...</div>
987
988  <div>...</div>
989
990  <div>...</div>
991
992  <div>...</div>
993
994  <div>...</div>
995
996  <div>...</div>
997
998  <div>...</div>
999
1000 <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("Error adding assets to cache", error);
10      })
11   );
12 });
13
14 // eslint-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) => {
22         if (name !== CACHE_NAME) {
23           return caches.delete(name);
24         }
25       })))

```


3. Penambahan *Push Notification*

```

1 importScripts("https://cdn.onesignal.com/sdks/web/v16/OneSignalSDK.sw.js");
2
3
4 <script src="https://cdn.onesignal.com/sdks/web/v16/OneSignalSDK.page.js" defer></script>
5 <script>
6 window.OneSignalDeferred = window.OneSignalDeferred || [];
7 OneSignalDeferred.push(function (OneSignal) {
8   OneSignal.init({
9     appId: "6a565311-bb5f-46f4-9b7e-3d147414c129",
10    safari_web_id: "web.onesignal.auto.49889676-9f7f-4916-aef3-fd94958742a1",
11    notifyButton: {
12      enable: true,
13    },
14    allowLocalhostAsSecureOrigin: true,
15    serviceWorkerPath: "/OneSignalSDKWorker.js",
16  });
17 });
18 </script>

```

Gambar 13 Source Code Penambahan *Push Notification*

4.4 Cutover

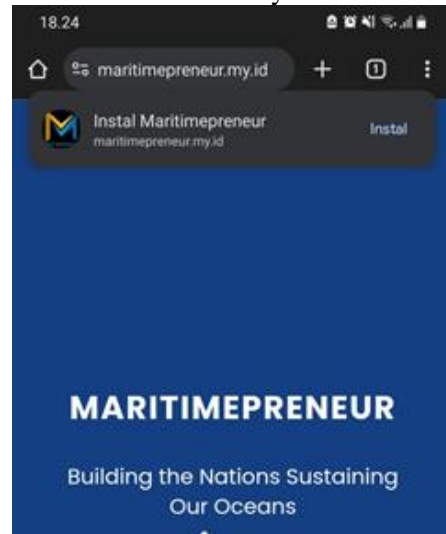
Tahap terakhir adalah *cutover*, di mana aplikasi yang telah dikonstruksi dipindahkan dari lingkungan pengembangan ke lingkungan produksi. Proses ini melibatkan uji coba dan implementasi sistem secara penuh. Pada gambar menunjukkan hasil dari beberapa tampilan website yang telah selesai dikerjakan.

1. Halaman Media



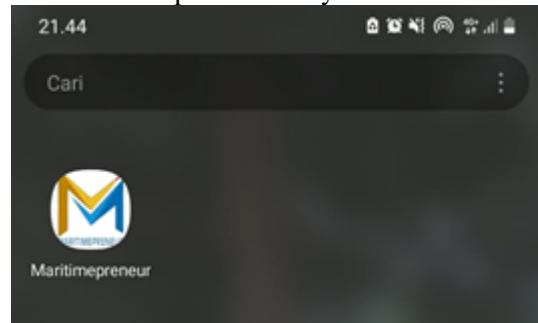
Gambar 14 Tampilan Halaman Media

2. *Install PWA* ke Layar Utama



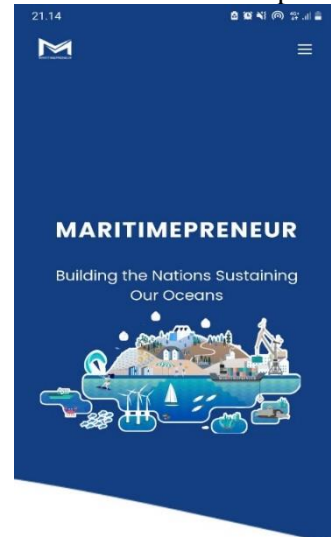
Gambar 15 Tampilan Instalasi PWA ke Layar Utama

3. Icon Aplikasi di Layar Utama



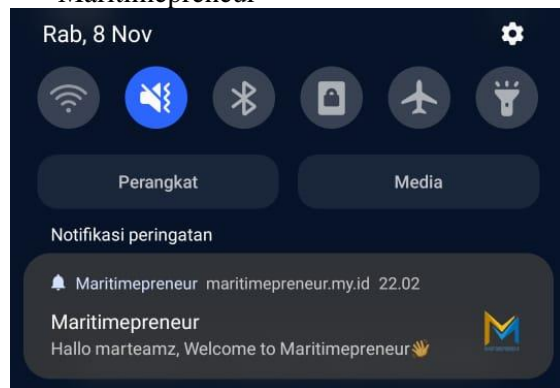
Gambar 16 Icon Aplikasi pada Layar Utama Android

4. Halaman Utama Maritimepreneur



Gambar 17 Tampilan Halaman Maritimepreneur pada Android

5. Push Notification pada Maritimpreneur



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
Installable			
1.	Web app manifest atau service worker memenuhi persyaratan instalasi aplikasi	Tidak Terpenuhi	Terpenuhi
PWA Optimized			
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
Pengujian Manual			
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:

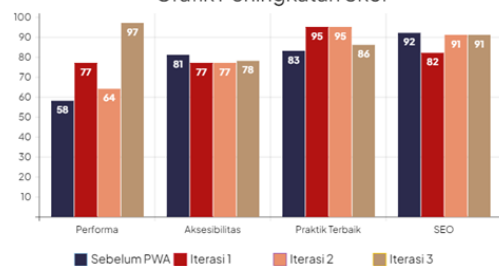


Gambar 19 Hasil Pengujian Performa Website

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:

Grafik Peningkatan Skor



Gambar 20 Grafik Peningkatan Skor

5. KESIMPULAN

- Website Maritimpreneur dikembangkan sebanyak 3 kali iterasi sehingga didapatkan halaman baru, diimplementasikan Progressive Web Apps, dan diterapkan push notification.
- Penerapan PWA pada website Maritimpreneur 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.