

IMPLEMENTASI PROGRESSIVE WEB APPS PADA APLIKASI PENDAFTARAN PESERTA DIDIK BARU DI SMK FARMASI MINASA MULIA CIPUTAT

Sopiyan Apandi^{1*}, Ayyub Muhammad¹

¹Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: 1*dosen02601@unpam.ac.id, 2ayyubmuhammad98@gmail.com

(* : coresponding author)

Abstrak– Penerimaan peserta didik baru adalah kegiatan yang selalu diadakan rutin setiap tahun ajaran baru. Proses penerimaan yang digunakan SMK Farmasi Minasa Mulia saat ini masih menggunakan cara datang langsung ke sekolah untuk mendaftar. Oleh karena itu, penelitian ini bertujuan untuk merancang aplikasi Penerimaan Peserta Didik Baru (PPDB) basis web pada SMK Farmasi Minasa Mulia, guna menciptakan sistem pendaftaran yang lebih efisien dan memudahkan calon siswa/siswi baru mencari informasi. Dalam pembuatannya sistem ini menerapkan konsep Progressive Web App (PWA) menggunakan metode Extreme Programming dan dalam perancangan sistem menggunakan Unified Modelling Language (UML). Hasil yang ingin dicapai dari penelitian ini adalah pihak sekolah bisa dengan mudah memberikan layanan pendaftaran dengan baik, serta memberikan informasi lebih lengkap mengenai sekolah dan memudahkan calon siswa-siswi untuk mendaftar.

Kata Kunci: Penerimaan peserta didik baru, PWA, UML, Extreme Programming.

Abstract– Admission of new learners is an activity that is always held regularly every new school year. The admission process used by SMK Farmasi Minasa Mulia currently still uses the method of coming directly to the school to register. Therefore, this study aims to design a web-based New Student Admissions (NSA) application at SMK Farmasi Minasa Mulia, in order to create a more efficient registration system and make it easier for prospective new students to find information. In its creation, this system applies the concept of Progressive Web App (PWA) using the Extreme Programming method and in designing the system using Unified Modelling Language (UML). The result to be achieved from this study is that the school can easily provide good registration services, as well as provide more complete information about the school and make it easier for prospective students to register.

Keywords: Admission of new learners, PWA, UML, Extreme Programming.

1. PENDAHULUAN

Teknologi kini semakin canggih, tidak mengenal waktu dan batasan ruang, sehingga teknologi berkembang semakin pesat dari tahun ketahun, dan memiliki banyak teknologi dengan berbagai bidang terutama website. Di era sekarang ini informasi sangat dibutuhkan guna memberikan kebutuhan informasi untuk pengguna teknologi.

SMK Farmasi Minasa Mulia adalah sekolah menengah kejuruan yang berlokasi di daerah Kecamatan Pamulang, Ciputat. Setiap tahun ajaran baru akan ada penerimaan siswa/siswi baik dari pindahan maupun pendaftar baru, untuk sistem yang digunakan sebagai proses penerimaan siswa-siswi ini pihak sekolah masih menggunakan media chat whatsapp dan langsung datang ke sekolah dengan mengisi formulir pendaftaran yang disediakan pihak sekolah, cara lama tersebut terbilang kurang efektif. Maka dari itu, untuk meningkatkan efektifitas proses penerimaan peserta didik baru, kini pihak sekolah menerapkan sistem penerimaan peserta didik baru melalui *platform* yang berbasis web agar meningkatkan ketertarikan calon siswa.

Berdasarkan permasalahan pada paragraf diatas, maka dibutuhkan sebuah aplikasi basis web sebagai media yang dapat meringankan proses penerimaan peserta didik baru sekolah menggunakan

teknologi *Progressive Web Apps* (PWA) dan menggunakan Bahasa Pemrograman *Reactjs*, dengan menerapkan teknologi tersebut kecepatan proses loading website ketika diakses akan lebih cepat walaupun dalam kondisi jaringan lemah (*low connection*). Pembuatan sistem ini menggunakan metode *Extreme Programming* (XP), serta menggunakan *Unified Modelling Language* (UML) sebagai tools untuk perancangan sistem.

2. METODOLOGI PENELITIAN

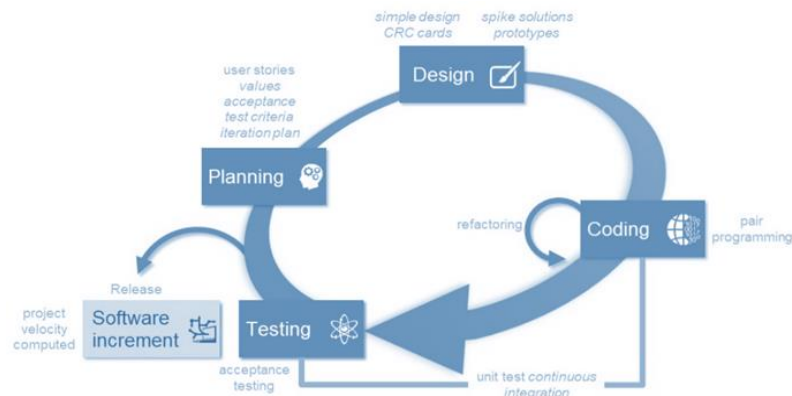
2.1 Metode Pengumpulan Data

Penelitian dilakukan langsung kelapangan pada SMK Farmasi Minasa Mulia yang bertujuan mencari sumber data *primer* yang diperlukan untuk melakukan pengembangan aplikasi PPDB dengan cara:

- a. Observasi
Tahap ini melakukan pengambilan data dan pengamatan langsung untuk mencari parameter yang dibutuhkan untuk pengembangan aplikasi lebih lanjut.
- b. Studi Pustaka
Tahap ini penulis mencari referensi dari berbagai sumber yang ada seperti buku, jurnal dan tugas akhir mengenai sistem pendaftaran *online*, serta mencari referensi baik itu dari internet maupun orang yang kompeten dari kasus penelitian ini.
- c. Wawancara
Melakukan sesi tanya - jawab kepada pihak terkait dalam hal ini yaitu; kepala sekolah SMK Farmasi Minasa Mulia mengenai objek penelitian.

2.2 Metode Pengembangan Aplikasi atau Perangkat Lunak

Metode pengembangan sistem yang akan dibuat menggunakan metode *extreme programming* dan menerapkan konsep *progressive web app*. *Extreme programming* adalah metode yang sudah disederhanakan dari beberapa tahapan pengembangan sehingga sistem yang akan dibuat menjadi lebih efisien, adaptif dan fleksibel. Berikut beberapa uraian pada tahapan metode *extreme programming*:



Gambar 1. Tahap Metode Extreme Programming

1. Perencanaan atau *Planning*
Fase ini dimulai dengan mendapatkan persyaratan dari rangkaian aktifitas suatu sistem yang membantu pengguna untuk memahami kebutuhan pokok untuk sistem dan mendapatkan gambaran yang jelas tentang fungsi utama, kegunaan dan keluaran yang diinginkan.
2. Perancangan atau *Design*
Pada fase perancangan dilakukan dengan pemodelan sistem yang berdasarkan data yang dihasilkan dari analisa kebutuhan yang didapat. Lalu membuat model basis data untuk menggambarkan hubungan antar data. Pemodelan sistem yang digunakan yaitu *Unified Modelling Language* (UML) yang terdiri dari *Activity Diagram* dan *Use Case Diagram*.

3. Pengkodean atau *Coding*
Fase ini merupakan implementasi dari hasil perancangan sistem yang dibuat kedalam kode program. Dalam pembangunan aplikasi pendaftaran peserta didik baru menggunakan bahasa pemrograman *javascript* dari *Reactjs*, *Tailwind CSS*, *HTML*, dan *ViteJs*, sedangkan untuk basis data menggunakan *Firestore Database* dari *Firebase Google*.
4. Pengujian atau *Testing*
Metode yang digunakan untuk pengujian pada aplikasi pendaftaran peserta didik baru adalah *Black-Box Testing* dengan melakukan pengujian terhadap keluaran dan masukan yang dihasilkan oleh aplikasi dan *Lighthouse Testing* untuk menguji hasil *Progressive web app*.

3. ANALISA DAN PEMBAHASAN

3.1 Perencanaan atau *Planning*

Pada tahap rancangan ini dilakukan analisa sistem yang sudah berjalan sebelumnya dengan mengumpulkan data pada masalah dan kendala yang terjadi, data ini akan digunakan untuk mengetahui kendala apa saja yang kemudian akan dilakukan perbaikan dengan menerapkan metode dan alat pendukung lainnya. Berikut adalah daftar dari permasalahan atau kendala yang terjadi, yang digambarkan pada tabel 1.

Tabel 1. Analisa Permasalahan

Sistem	Kendala
Pendaftaran Siswa-siswi	Calon siswa-siswi diharuskan datang ke sekolah untuk mendaftar.
Mengelola data calon siswa-siswi	Staff sekolah masih menggunakan cara umum atau menggunakan kertas yang memakan waktu dan memiliki resiko terjadinya kesalahan
Proses seleksi calon siswa-siswi	Proses seleksi dilakukan disekolah dan calon siswa-siswi diharuskan datang kesekolah pada tanggal yang ditentukan.

Berikut merupakan hasil pembahasan dan kebutuhan sistem berdasarkan hasil dari pengumpulan informasi dan data yang dibutuhkan dengan metode wawancara dan observasi kepada semua pihak sekolah terkait mengenai proses pendaftaran yang diperlukan:

Tabel 2. Sistem Usulan

Sistem berjalan	Sistem Usulan
Calon peserta didik diharuskan datang kesekolah untuk mendapatkan informasi dan melakukan pendaftaran	Calon siswa mendaftar dimana saja yang dilakukan secara online dan mendapatkan informasi detail tentang sekolah
Pihak sekolah harus melakukan proses pendataan melalui berkas-berkas yang calon pendaftar serahkan	Pihak sekolah tidak harus melakukan pendataan secara manual karena pendataan sudah dilakukan otomatis secara online
Panitia PPDB diharuskan membuat laporan berdasarkan formulir pendaftaran	Panitia tidak perlu membuat laporan lagi karena laporan dapat di unduh langsung dari website

2.1 Perancangan atau *Design*

Tahap perancangan merupakan tahap dilakukannya pembuatan struktur dan bentuk dari sistem yang akan dibuat berdasarkan hasil sistem usulan yang dibutuhkan. Pembuatan model ini dibuat menggunakan *Unified Modeling Language (UML)* yang menggambarkan suatu sistem menjadi lebih mudah dipahami selama proses desain. UML biasanya digambarkan kedalam bentuk berupa diagram dan notasi yang dapat membantu membuat model dari sistem ke berbagai perspektif.

1. Use Case Diagram

Use Case Diagram adalah permodelan perilaku (*behavior*) dari sistem informasi yang akan dibuat. *Use Case* menguraikan interaksi antara satu atau aktor lebih dari satu. Berikut permodelan usecase diagram dari sistem PPDB online:



Gambar 1. Use Case Diagram Sistem Usulan

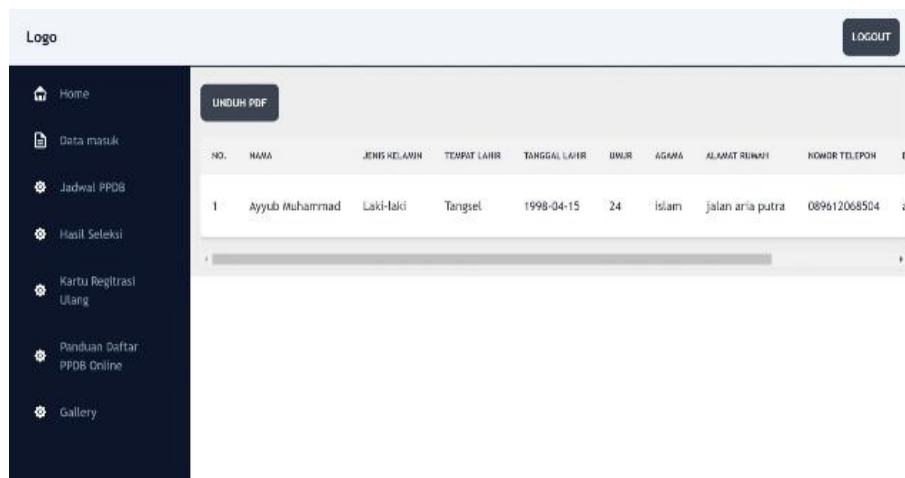
4. IMPLEMENTASI

4.1 Pengkodean atau Coding

Coding sendiri adalah tahap atau proses dimana dilakukannya penulisan program dengan menggunakan bahasa pemrograman dalam hal ini yaitu *react js*. Tahap ini difokuskan pada implementasi untuk membangun *user interface* atau tampilan antarmuka sistem.

1) Halaman Dashboard Kelola Data Calon Siswa-Siswi

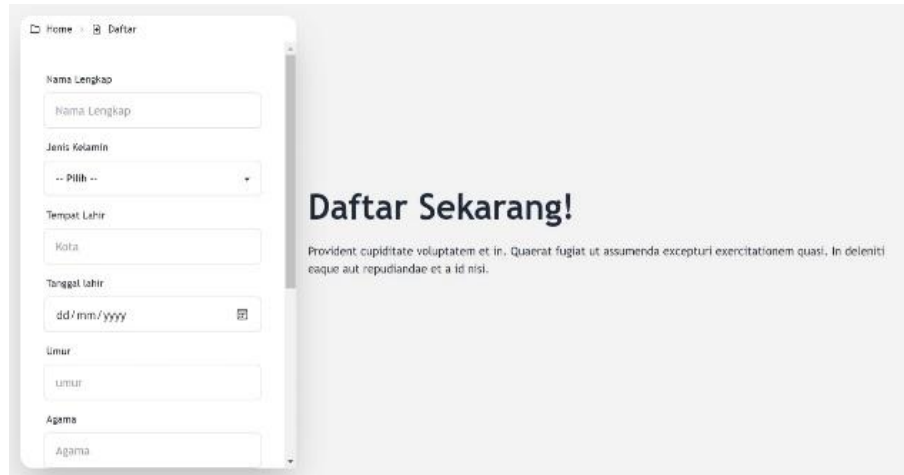
Halaman *dashboard* kelola data calon siswa-siswi ini berfungsi melakukan pendataan calon siswa-siswi yang sudah melakukan pendaftaran secara *online*.



Gambar 3. Halaman Dashboard Kelola Data Calon Siswa-Siswi

2) Halaman Pendaftaran

Halaman pendaftaran dapat diakses oleh calon siswa-siswi yang ingin melakukan pendaftaran dengan cara mengisi *form* atau formulir berdasarkan biodata yang dibutuhkan. Berikut tampilan halaman daftar pada gambar 9.



Gambar 4. Halaman Pendaftaran Online

4.2 Pengujian atau *Testing*

Pengujian merupakan sebuah proses dari bagian pengembangan aplikasi yang bertujuan untuk melakukan validasi kestabilan aplikasi pada fungsionalitas aplikasi apakah sudah berjalan sesuai dengan yang diharapkan. Proses pengujian ini menggunakan pengujian *black box testing* yang melakukan pengujian hubungan antara sistem dengan pengguna saat proses *input* dan *output*, *white box testing* melakukan pengujian yang berfokus pada kode program dengan cara melakukan perhitungan *Cyclomatic Complexity* yang berfungsi menentukan berapa banyak *independent path* dan mengukur kompleksitas struktur logis dari program serta menentukan tingkat keberhasilan kode program, dan *lighthouse testing* melakukan pengujian pada halaman website dan pengujian apakah konsep *progressive web app* (PWA) sudah berhasil diterapkan.

1. *Black box Testing*

a) *Dashboard Kelola Calon Siswa-Siswi*

Tabel 3. Pengujian Black Box Dashboard Kelola Calon Siswa-Siswi

Masukkan	Skenario	Keluaran	Hasil
Memilih menu Data masuk	menampilkan halaman dengan tabel data calon siswa/i	Dashboard dapat menampilkan halaman dengan tabel data calon siswa/i	Ok
Menekan tombol Unduh PDF	Data tabel dapat di unduh menjadi file pdf	Sistem dapat mengunduh data pada tabel menjadi file pdf	Ok
Menekan tombol ikon hapus pada baris tabel data	Menghapus satu baris data pada tabel	Sistem dapat menghapus data yang dipilih	Ok

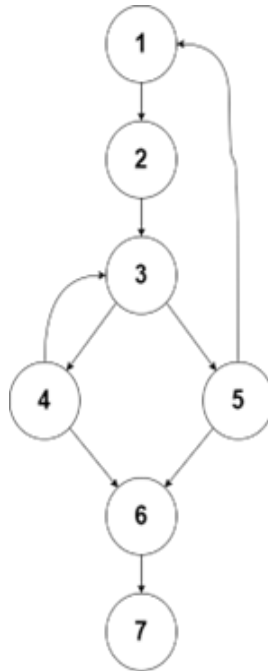
2. *White box Testing*

Pengujian *white box* dilakukan pada sistem yang berfokus pada kode program dan melakukan perhitungan *cyclomatic complexity* untuk mengetahui hasil dari kinerja program. Berikut pengujian *white box* pada pengelolaan calon siswa-siswi dan pendaftaran.

a. Pengujian White Box Pengelolaan Data Calon Siswa-Siswi

Tabel 4. Pengujian White Box Kelola Data Calon Siswa-siswi

Node	Kode Program
1	<pre>import React, { useState, useEffect } from "react"; import TableDaftar from "../Components/Table"; import { getDocs, deleteDoc, doc } from "firebase/firestore"; import { daftarDb } from "../Service/dbDaftar"; import { db } from "../Service/configFirebase"; import { toast } from "react-toastify";</pre>
2	<pre>const [dbdaftar, setDbdaftar] = useState([]);</pre>
3	<pre>useEffect(() => { getDataDaftar(); }, []);</pre>
4	<pre>function getDataDaftar() { getDocs(daftarDb).then((res) => { const tampil = res.docs.map((dox) => ({ data: dox.data(), id: dox.id, })); setDbdaftar(tampil); }) .catch((err) => console.log(err.message)); }</pre>
5	<pre>const deletData = (id) => { let dbDoc = doc(db, "daftar", id); deleteDoc(dbDoc) .then((res) => { toast.info("Delete Data Berhasil", {position:toast.POSITION.BOTTOM_RIGHT,}); }) .catch((err) => { console.log(err);}); };</pre>
6	<pre>{Object.values(dbdaftar).map((c, ix) => (<tr className="hover" key={c.id}> <td>{ix + 1}</td> <td>{c.data.inputs.nama}</td> ... <td> <button onClick={() => { deletData(c.id); }} className="btn btn-warning rounded-full"> Hapus </button> </td> </tr>))}</pre>
7	Selesai



Gambar 5. Flowgraph Diagram Mengelola Calon Siswa- Siswi

Pada gambar *flowgraph* diatas menunjukkan ada beberapa *node* dan *edge* yang dapat dilakukannya perhitungan *Cyclomatic Complexity* dibawah ini

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 9 - 7 + 2 \\ &= 4 \end{aligned}$$

Keterangan:

E = Total *Edge Flowgraph*

N = Total *Node Flowgraph*

Hasil diatas menunjukkan terdapat 4 path yang dapat dihitung sebagai berikut:

Path 1 = 1-2-3-4-6-7

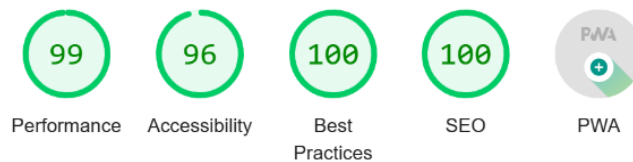
Path 2 = 1-2-3-5-6-7

Path 3 = 1-2-3-4-3-4-6-7

Path 4 = 1-2-3-5-1-...-5-6-7

3. Lighthouse Testing

Pengujian *Lighthouse* merupakan fitur atau salah satu alat yang terdapat pada browser modern seperti *Chrome*, *Edge Browser*, dan *Mozilla Firefox*. Pengujian ini dilakukan untuk mengetahui kinerja halaman website dan *progressive web apps*. Berikut hasil pengujian *lighthouse* dapat dilihat pada gambar 13.

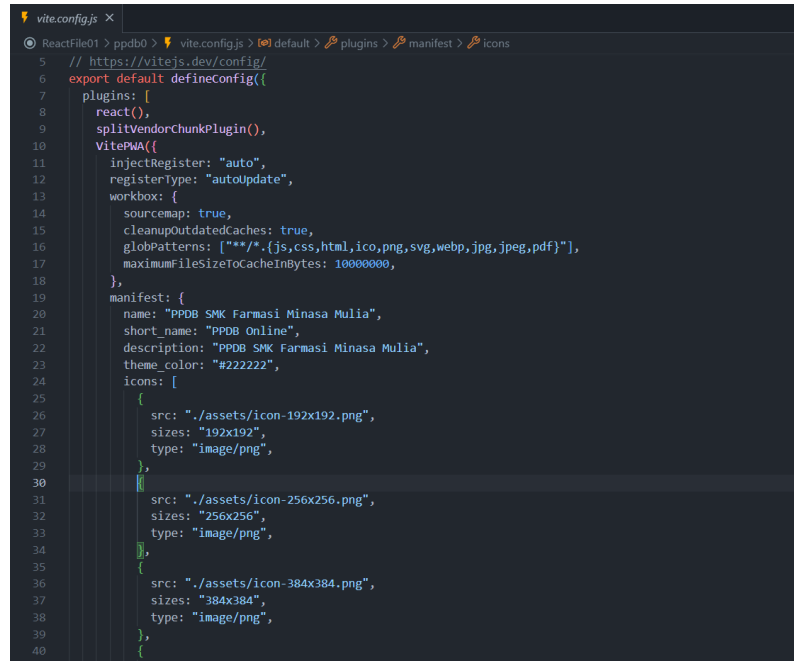


Gambar 6. Pengujian Lighthouse

Pada gambar diatas menunjukkan hasil pengujian *lighthouse* sudah memenuhi persyaratan atau kriteria dari *progressive web app*. Hasil tes tersebut menunjukkan skor 99 untuk indikator kinerja, skor 96 untuk indikator aksesibilitas, skor 100 untuk indikator praktik pada program atau best practices, dan skor 100 pada indikator *search engine optimization* (SEO).

4.3 Implementasi Progressive web apps (PWA) Pada Web PPDB

Langkah penerapan PWA pada web ppdb berbeda dengan konsep PWA pada umumnya, cara yang digunakan adalah cukup menambahkan konfigurasi berikut ke dalam satu file konfigurasi *javascript*:



```

5 // https://vitejs.dev/config/
6 export default defineConfig({
7   plugins: [
8     react(),
9     splitVendorChunkPlugin(),
10    vitePWA({
11      injectRegister: "auto",
12      registerType: "autoUpdate",
13      workbox: {
14        sourceMap: true,
15        cleanupOutdatedCaches: true,
16        globPatterns: ["**/*.{js,css,html,ico,png,svg,webp,jpg,jpeg,pdf}"],
17        maximumFileSizeToCacheInBytes: 10000000,
18      },
19      manifest: {
20        name: "PPDB SMK Farmasi Minasa Mulia",
21        short_name: "PPDB Online",
22        description: "PPDB SMK Farmasi Minasa Mulia",
23        theme_color: "#222222",
24        icons: [
25          {
26            src: "./assets/icon-192x192.png",
27            sizes: "192x192",
28            type: "image/png",
29          },
30          {
31            src: "./assets/icon-256x256.png",
32            sizes: "256x256",
33            type: "image/png",
34          },
35          {
36            src: "./assets/icon-384x384.png",
37            sizes: "384x384",
38            type: "image/png",
39          },
40        ],
41      },
42    })
43  ],
44 })

```

Gambar 7. Potongan Dari Konfigurasi PWA PPDB

5. KESIMPULAN

Berdasarkan hasil *planning*, *design*, *coding* dan *testing* pada aplikasi maka dapat disimpulkan, pengembangan aplikasi ppdb *online* menggunakan metode *extreme programming* cukup efektif dan penerapan konsep *progressive web app* berjalan dengan baik. Website ppdb *online* juga dapat diakses dalam kondisi jaringan tidak stabil dan dapat berjalan secara *offline* yang cukup membantu pengguna melakukan pendaftaran.

REFERENCES

Chan Chan Bayu Bahari, Y. S. (2019). Penerapan Progressive Web Apps Pada Aplikasi Lowongan Pekerjaan Dosen Universitas Perjuangan. *Informatics And Digital Expert (INDEX)*, 25-31.

Dini Hari Pertiwi, H. E. (2022). Penerapan Extreme Programming (XP) pada Sistem Informasi Pembayaran Perbaikan Kendaraan di CV Tiara Persada Berbasis Web. *Jurnal Informatika Global*, 123-130.

Iwan Aslan, H. B. (2022). Pengembangan Website Fakultas Teknik Universitas Hamzanwadi Berbasis Progressive WEB APP (PWA). *Infotek : Jurnal Informatika dan Teknologi*, 99-107.

Santoso, M. F. (2021). TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP. *JURNAL KHATULISTIWA INFORMATIKA*, 107-114.

Yusnia Budiarti, R. (2020). Implementasi Metode Extreme Programming Untuk Merancang Sistem Informasi Pendaftaran Siswa Baru Berbasis Web Pada Smk Multimedia Mandiri Jakarta. *Informatika : Jurnal Ilmiah Fakultas Sains dan Teknologi*, 2-3.