

RANCANG BANGUN WEB SERVICE SISTEM INFORMASI BERKAS ADMINISTRASI TUGAS AKHIR TEKNIK INFORMATIKA BERBASIS PROGRESSIVE WEB APP (ANDROID)

(Center, Bold, Times New Roman 15, Maksimum 14 kata, Huruf Awal Huruf Besar)

Mohamad Rizky Isa¹, Asmaul Husna Nasrullah, M.Kom², Serwin, M.Kom³

^{1,2,3} Ilmu Komputer, Teknik Informatika, Universitas Ichsan Gorontalo, Gorontalo, Indonesia
Email : kikiisa89@gmail.com

Abstrak - . Pengembangan sistem informasi administrasi dapat diintegrasikan dengan berbagai platform untuk memastikan kelancaran proses administrasi dan pengolahan data di suatu organisasi. Penggunaan teknologi yang tepat, seperti *web service* dan *progressive web app*, dapat membantu administrasi menjadi lebih fleksibel dalam mengakses dan memproses data dan informasi dari berbagai platform. Pengembangan web service dapat memungkinkan Sistem Informasi Berkas Administrasi Tugas Akhir Teknik Informatika (Siabti) terintegrasi dengan platform lainnya dan meningkatkan fleksibilitas sistem. Sementara itu, pengembangan *progressive web app* dapat meningkatkan aksesibilitas dan pengalaman pengguna, hasil pengujian menunjukkan bahwa aplikasi berbasis *Progressive web app Android* untuk Sistem Informasi Berkas Administrasi Tugas Akhir Teknik Informatika mendapat respons positif dari mahasiswa, dengan rata-rata skor penilaian, yaitu 9.

Kata Kunci: sistem informasi, administrasi tugas akhir, progressive web app, android
(paling sedikit 5 kata yang berhubungan dengan isi penelitian di dipisahkan dengan koma)

Abstract - . The development of administrative information systems can be integrated with various platforms to ensure the smooth running of administrative processes and data processing in an organization. The use of appropriate technologies, such as web services and progressive web apps, can help the administration become more flexible in accessing and processing data and information from various platforms. The development of web services allows the Information System for Final Project Administration Files in Informatics Engineering, called SIABTI, to be integrated with other platforms and increase the flexibility of the system. Meanwhile, the development of a progressive web app can improve accessibility and user experience. The test results indicate that the Progressive Web App Android-based application for the Informatics Engineering Final Project Administration Filing Information System has a positive response from students, with an average assessment score of 9.

Keywords: information system, final project administration, progressive web app, Android

1. PENDAHULUAN

(Center, Bold, Times New Roman 13, UPPER CASE)

Pengembangan sistem administrasi multi-platform sangat penting bagi kelancaran proses bisnis dan pengolahan data di organisasi. SIABTI adalah sistem informasi untuk upload berkas tugas akhir mahasiswa dan sebagai jembatan antara mahasiswa dengan administrasi kampus dan fakultas. Observasi di Universitas Ichsan Gorontalo menunjukkan bahwa SIABTI masih menggunakan web native yang sulit diintegrasikan dengan platform lain dan lambat ketika koneksi internet lemah. Untuk mengatasi masalah tersebut, diperlukan teknologi web yang mengikuti perkembangan IT, seperti Progressive Web App (PWA) yang dapat berjalan di platform mobile dengan kinerja cepat dan minim penggunaan storage. PWA menggunakan Service Workers untuk mengambil konten dari cache saat jaringan tidak stabil dan Web App Manifest untuk memberikan informasi aplikasi. Web service memungkinkan integrasi dengan layanan lain tanpa perlu mengembangkannya sendiri. Dalam studi kasus ini, digunakan Visual Studio Code sebagai code editor, PHP dengan CodeIgniter sebagai web service, VUE JS sebagai framework PWA (Android), dan MySQL sebagai database. Judul penelitian ini adalah "Rancang Bangun Web Service SIABTI Berbasis Progressive Web App (Android)" di Universitas Ichsan Gorontalo Fakultas Ilmu Komputer.

2. TINJAUAN PUSTAKA

2.1. Web Service

web service merupakan perangkat lunak yang dikembangkan untuk mendukung Interaksi antar mesin melalui jaringan. *Web Service* menyediakan *interface* yang memungkinkan aplikasi yang dikembangkan dapat berinteraksi dengan berbagai jenis perangkat lunak yang memiliki platform dan sistem operasi yang sama ataupun berbeda.

2.2. Rest (Representational State Exchange)

Representational State Exchange dapat berupa seperangkat standar teknik yang mengirimkan informasi melalui antarmuka standar seperti *HTTP*. *REST API* bekerja seperti aplikasi web biasa. Klien dapat mengirim permintaan ke server melalui konvensi *HTTP* dan setelah itu server beraaksi kembali ke klien. *REST* dibuat oleh Roy Handing yang merupakan *Co-Founder* dari *Apache HTTP Server Project*[2] Adapun beberapa keunggulan yang dimiliki oleh *REST*.

2.3 Progressive Web App (PWA)

Pada dasarnya teknologi *Progressive Web App (PWA)* bekerja layaknya aplikasi berbasis web pada biasanya, yang membedakan *PWA* dengan aplikasi website lainnya adalah *PWA* yang bekerja dengan konektivitas yang independen. Artinya, aplikasi *progressive web app* dapat bekerja pada jaringan yang kurang stabil dengan adanya *service worker* [3], selain itu *progressive web app* juga menggunakan teknologi *instant loading* yang membuat aplikasi website tersebut berjalan dengan cepat, *Screenhome* dimana aplikasi website tersebut dapat dijadikan *icon* pada *dekstop* atau *homescreen*.

2.4 Spa (Single Page Application)

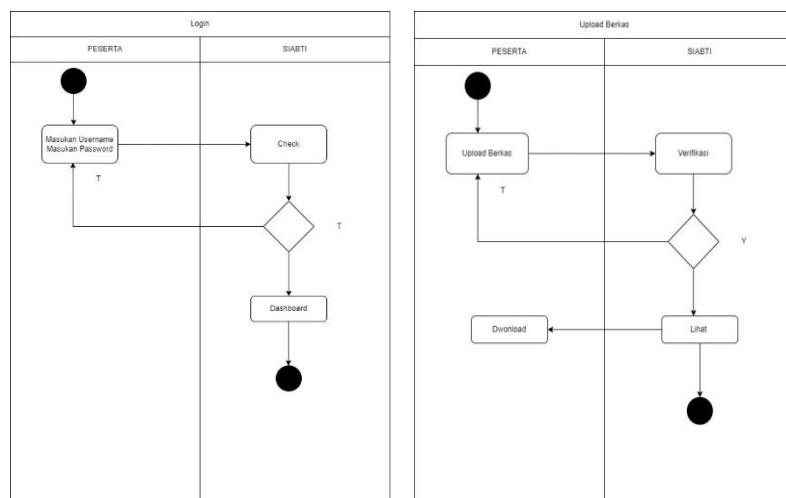
Single Page Application (SPA) adalah aplikasi web yang muat pada satu halaman web dengan aktivitas yang energik tanpa menghidupkan kembali halaman (*page refresh*). Interaksi Aplikasi Halaman Tunggal Aplikasi dapat diurus tanpa datang ke server, dan dapat ditingkatkan eksekusinya dalam beberapa cara seperti pengaturan waktu, penggunaan *AJAX*, navigasi yang mudah ke halaman dan sebagainya. Klien kesimpulan *conclusion client* akan lebih nyaman, karena sangat mudah untuk dijelajahi atau dijelajahi melalui halaman web dan saluran konten yang beragam. Penerapan inovasi *Single Page Application (SPA)* yang digunakan ditujukan untuk mengurangi beban kerja server ketika permintaan informasi dari klien terjadi, dan meminimalkan pemanfaatan sumber daya oleh server. Perbandingan Halaman *Web Non-SPA* dengan menggunakan pedoman kerja SPA.

3. METODE PENELITIAN

3.1 Jenis Metode, Objek, Waktu, dan Lokasi Penelitian

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka Penelitian ini menggunakan metode penelitian eksperimen. Dengan demikian jenis penelitian ini adalah penelitian eksperimental. Objek penelitian ini *web SIABTI* penelitian ini dimulai dari bulan November 2022 sampai dengan bulan Desember 2022, yang berlokasi pada Universitas Ichsan Gorontalo Fakultas Ilmu Komputer.

3.2 Pengembangan Sistem



3.1.1 Analisis Sistem

Analisis Sistem menggunakan pendekatan berorientasi objek yang digambarkan dalam bentuk

- a. Functional Modelling, menggunakan alat bantu UML, dalam bentuk :
 - *Use Case Diagram.*
 - *Activity Diagram.*
- b. Structural Modelling, menggunakan alat bantu UML, dalam bentuk:
 - *Class Diagram.*
- c. Behavioral Modelling, menggunakan alat bantu UML, dalam bentuk :
 - *Sequence Diagram.*

3.3.2 Analisis Sistem

Desain Sistem menggunakan pendekatan berorientasi objek yang di gambarkan dalam bentuk

- a. *Architecture Design*, menggunakan alat bantu vscode dalam bentuk:
 - Model jaringan dari sistem *Client dan Server.*
- Spesifikasi *Hardware* dan *software* yang di rekomendasikan.
- b. *Interfaxe Design*, menggunakan alat bantu vscode dalam bentuk:
 - Mekanisme User.
 - Mekanisme Navigasi.
 - Mekanisme Input.
 - Mekanisme Output (Report).

3.3.2 Konstruksi Sistem

Pada tahap ini menerjemahkan hasil pada tahap ini menerjemahkan hasil pada tahap analisis dan desain ke dalam kode kode program komputer kemudian membangun sistemnya. Alat bantu yang digunakan pada tahap ini adalah Vscode sebagai code editor dengan Bahasa Pemrograman PHP, Node Js dan Javascript. Dan Mysql sebagai Basis Data.

3.3.3 Black Box

Software yang telah direkayasa kemudian diuji dengan metode *White Box Testing* pada kode program proses penerapan metodenya/modelnya. Kode program tersebut dibuatkan *flowchart* programnya, kemudian dipetakan kedalam bentuk *flowgraph* (bagan alir kontrol) yang tersusun dari beberapa node dan edge. Berdasarkan *flowgraph*, ditentukan jumlah Region dan Cyclomatic Complexity (CC). Apabila *independent path* = $V(G)$ = (CC) = Region, di mana setiap *path* hanya dieksekusi sekali dan sudah benar, maka sistem dinyatakan efisien dari segi kelayakan logika pemrograman

3.3.4 White Box

Software yang telah direkayasa kemudian diuji dengan metode *White Box Testing* pada kode program proses penerapan metodenya/modelnya. Kode program tersebut dibuatkan *flowchart* programnya, kemudian dipetakan kedalam bentuk *flowgraph* (bagan alir kontrol) yang tersusun dari beberapa node dan edge. Berdasarkan *flowgraph*, ditentukan jumlah Region dan Cyclomatic Complexity (CC). Apabila *independent path* = $V(G)$ = (CC) = Region, di mana setiap *path* hanya dieksekusi sekali dan sudah benar, maka sistem dinyatakan efisien dari segi kelayakan logika pemrograman

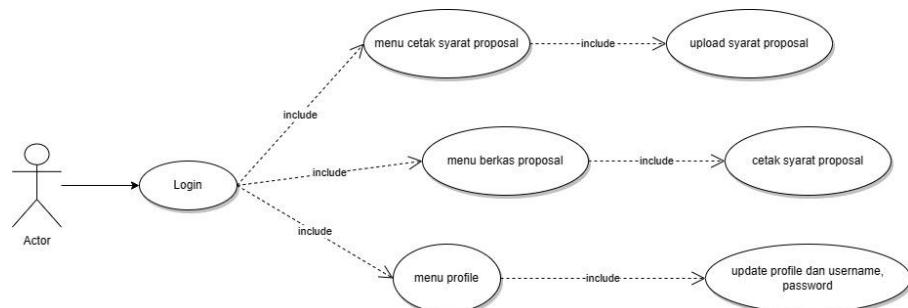
3.3.5 User Acceptance Testing

Setelah pengujian *White Box* dan *Black Box* telah selesai dilakukan, kemudian dilakukan instalasi. Selanjutnya pengujian terakhir yaitu User Acceptance Test. Pengujian ini akan memberikan gambaran tentang penerimaan oleh pengguna, *software* kemudian di test oleh beberapa pengguna untuk menjawab kuesioner yang diberikan. Kemudian data kuesioner tersebut diolah dengan pendekatan statistic guna memperoleh tingkat efektifitas sistem

4. HASIL DAN PEMBAHASAN

4.1 Perancangan Aplikasi

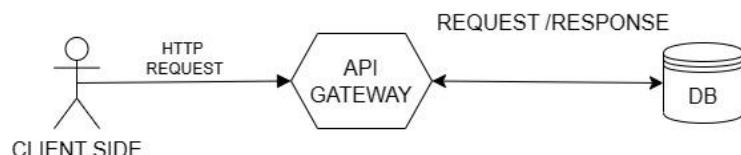
Pada tahapan ini mulai menerapkan konsep dan alur media yang ada serta merancang tampilan interface yang akan dibuat. Dengan menggunakan use case.



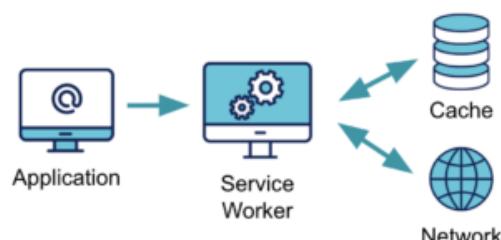
4.2 Hasil Rancangan Desain antar muka.



4.3 Hasil Rancangan Desain Arsitektur Web Service.



4.4 Hasil Rancangan Desain Arsitektur Progressive Web App (PWA).



4.4 Hasil Pengujian White Box

Dari *Flowgraph* tersebut, didapatkan:

$$\begin{aligned}
 \text{Diketahui} & \quad \text{Region}(R) = 4 \\
 & \quad \text{Node}(N) = 11 \\
 & \quad \text{Edge}(E) = 13 \\
 & \quad \text{Predicate Node}(P) = 3 \\
 & \quad \text{Rumus: } V(G) = E - N + 2 \text{ dan } V(G) = P + 1 \\
 \text{Penyelesaian : } & \quad V(G) = (13 - 11) + 2 = 4 \\
 & \quad V(G) = 3 + 1 = 4
 \end{aligned}$$

NO	PATH	KET
1.	1-2-9-10-11	OK
2.	1-2-3-7-8-9-10-11	OK
3.	1-2-3-4-7-8-9-10-11	OK
4.	1-2-3-4-5-6-7-8-9-10-11	OK

4.5 Hasil Pengujian Black Box

Pengujian *black box* dalam penelitian ini dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

Tabel 4. 1 Hasil Pengujian Blackbox

Nama Pengujian	Tujuan	Skenario	Hasil yang di harapkan	Ket
Membuka Aplikasi	Mengetahui aplikasi dapat berjalan tanpa error	Mencoba membuka aplikasi	Aplikasi berjalan dengan baik menampilkan flash screen dan menuju menu utama	Ok
Membuka menu beranda	Untuk melihat isi menu beranda	Menekan tombol beranda	Mendapatkan info aplikasi yang ada di menu beranda	Ok
Menu upload proposal	Untuk melihat apakah ketika user mengupload dokumen proposal dapat terkirim atau tidak	Menekean tombol upload	Mendapatkan status informasi apakah terupload atau tidak	Ok
Menu Profile	Untuk mengupdate informasi profile atau Username Dan Password	Melakukan perubahan nama dan username atau password	Mendapatkan informasi apakah ketika kita mengganti profile atau mengubah username dan password berhasil atau tidak	Ok

Logout	Untuk keluar sesion sebagai pengguna	Menekan tombol logout	Bisa keluar dan kembali ke halaman Login	Ok
--------	--------------------------------------	-----------------------	--	----

4.6 Hasil Pengujian User Acceptance Test (UAT)

Pengujian UAT dalam penelitian ini dilakukan dengan membagikan kuisioner pertanyaan kepada pengguna/responden. Adapun jumlah responden yaitu diambil sampai sebanyak 50 responden dengan jumlah pertanyaan sebanyak 10 pertanyaan.

A. Formulasi Rumus UAT (User Acceptance Test)

$$\% = ((\text{Total} / \text{User}) / 5) * 100$$

% >= 90%	A
80 <= % <= 89	B
70 <= % <= 79	C
60 <= % <= 69	D
50 <= % <= 59	E

B. Implementasi Penggunaan Formulasi dalam penelitian ini dapat dilihat

Pada tabel 4.4 dan 4.5

Tabel 4. 2 Data Jawaban Kuisioner Aplikasi Siabti berbasis PWA

KODE	5	4	3	2	1	5	4	3	2	1
P1	36	7	6	1	0	72	14	12	2	0
P2	38	7	4	1	0	76	14	8	2	0
P3	36	8	5	1	0	72	16	10	2	0
P4	38	7	4	1	0	76	14	8	2	0
P5	39	4	6	1	0	78	8	12	2	0
P6	38	6	5	1	0	76	12	10	2	0
P7	36	6	6	1	0	72	12	12	2	0
P8	38	5	7	0	0	76	10	14	0	0
P9	38	6	6	0	0	76	12	12	0	0
P10	39	6	5	0	0	78	12	10	0	0

Data yang di dapatkan diolah dengan mengalikan setiap poin jawaban dengan bobot yang sudah ditentukan sesuai dengan table bobot nilai jawaban dari hasil perhitungan dengan mengalikan setiap jawaban bobot yang sudah ditentukan maka di dapat hasil berikut :

Tabel 4. 3 Data Jawaban Kuisioner Aplikasi Siabti berbasis PWA

Soal	Nilai (Jumlah * bobot)					Total	Total / User	%	Skor Akhir (Rata-Rata)
	5	4	3	2	1				
P1	180	28	18	2	0	228	4,56	9,12	9,1
P2	190	28	12	2	0	232	4,64	9,28	9,3
P3	180	32	15	2	0	229	4,58	9,16	9,2
P4	190	28	12	2	0	232	4,64	9,28	9,3
P5	195	16	18	2	0	231	4,62	9,24	9,2
P6	190	24	15	2	0	231	4,62	9,24	9,2
P7	180	24	18	2	0	224	4,48	8,96	9,0

P8	190	20	21	0	0	231	4,62	9,24	9,2
P9	190	24	18	0	0	232	4,64	9,28	9,3
P10	195	24	15	0	0	234	4,68	9,36	9,4

Nilai skor akhir (rata-rata) pengujian *user acceptance testing* yaitu jumlah % dibagi 10. Adapun rentang nilai yang digunakan yaitu :

Nilai % $\geq 90\%$ maka nila nya A termasuk kategori sangat baik

$80 \leq \% \leq 89,99$ maka nilainya B termasuk kategori baik

$70 \leq \% \leq 79,99$ maka nilainya C termasuk kategori cukup baik

$60 \leq \% \leq 69,99$ maka nilainya D termasuk kategori kurang

$50 \leq \% \leq 59,99$ maka nilainya E termasuk kategori sangat kurang.

Dari hasil pengujian *user acceptance testing* yang telah dilakukan dilihat dari hasil rata rata yaitu dengan nilai rata rata 9 termasuk kategori sangat baik sehingga dapat disimpulkan bahwa aplikasi Sibati Berbasis *Progressive Web App (Android)* mudah dipahami, mudah dioperasikan, mendukung kebijakan membantu/memudahkan, aplikasi ini baik, dokumentasi baik, teknologi aplikasi canggih, bebas dari error dan perlu diimplementasikan.

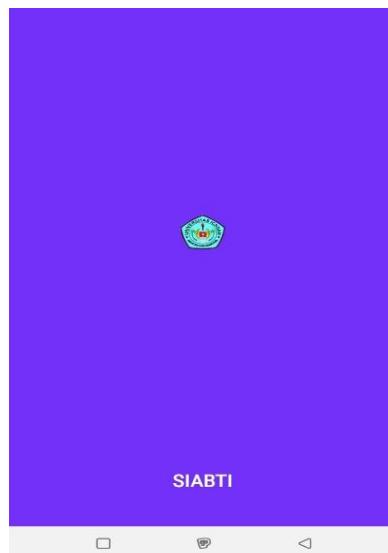
4.7 Implementasi Web Service.

Berikut ini beberapa *endpoint web service* yang sudah di implementasikan sebagai protokol pertukaran data antara Web Siabti dengan PWA (Android).

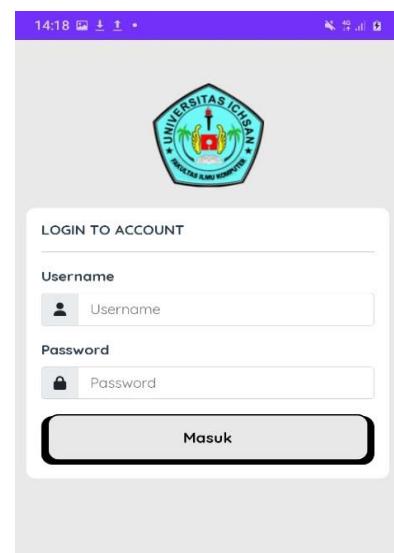
Url Endpoint	Type Method	Keterangan
siabti.com/api/loginMahasiswa	POST	melakukan request login
siabti.com/api/infoDashboard	GET	melakukan request untuk mengambil info statistic user
siabti.com/api/getProfile	GET	melakukan request untuk mengambil info user
siabti.com/api/updateProfile	POST	melakukan request agar data dapat user dapat terupdate
siabti.com/api/verifyLogin	GET	melakukan request untuk memverifikasi apakah user memegang token JWT atau tidak
siabti.com/api/infoBerkas	GET	melakukan request untuk menampilkan data berkas mahasiswa
siabti.com/api/Cetaksyaratproposalmahasiswa	POST	melakukan request untuk mengunduh laporan Persyaratan proposal
siabti.com/api/Katproposal	GET	melakukan request untuk menampilkan kategori persyaratan proposal
siabti.com/api/Apiuploadproposal	POST	melakukan request untuk mengupload file persyaratan proposal

Tabel 5. 1 Spesifikasi Endpoint Api dari Web Service

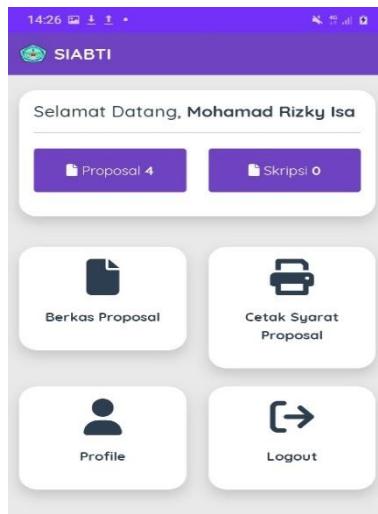
4.7 Implementasi antar muka berbasis Progressive web app (PWA)



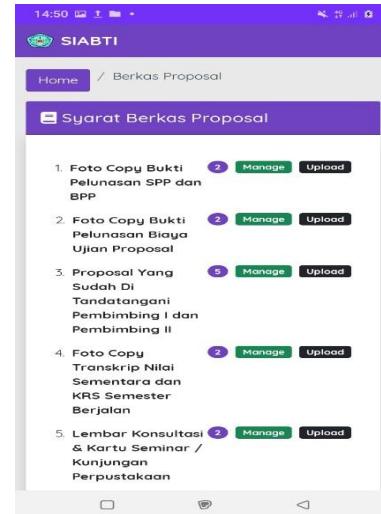
Tampilan Layar Depan (FlashScreen)



Tampilan Login Siabti



Tampilan Main Menu Siabti



Tampilan management proposal

5. KESIMPULAN

Rancangan *Web Service* SIABTI menghasilkan sebuah layanan yang dapat di integrasikan dengan berbagai system operasi, yang terutama service tersebut akan digunakan pada SIABTI berbasis PWA (Android). Aplikasi berbasis pwa juga dapat dijalankan di berbagai platform seperti desktop, web, dan mobile. Respon mahasiswa mengenai aplikasi siabti berbasis pwa android ini cukup baik sesuai dengan uji hasil responden yang telah dilakukan dan mendapatkan rata rata skor penilaian yaitu mendapatkan nilai 9 yang termasuk dalam kategori sangat baik atau nilai predikat A.

DAFTAR PUSTAKA

- [1] Kukuh Syaifudin1 , Ego Pranata2 , Wafa Nafisah A.3 , Arum Rawining Dian (2017) Analisis Usability pada Perbandingan Web-Native dengan Web Berbasis Progressive Web App
- [2] Akbar, M. (2018). Pengembangan RESTful API untuk Application Specific High Level Location Service.
- [3] Dean Hume (2017). Progressive Web Apps
- [4] Richard Murch, The Software Development Lifecycle - A Complete Guide
- [5] Dr. Henderi Dr. Untung Rahardja Efana Rahwanto, M.T.I · 2022 UML POWERED DESIGN SYSTEM USING VISUAL PARADIGM
- [6] Ali Zaky SmitDev Community · 2013, 36 Menit Belajar Komputer PHP dan MySQL
- [7] Wardana, S.Hut, M.Si Menjadi Master PHP dengan Framework Codeigniter.
- [8] Rismon Hasiholan Sianipar · 2017 Belajar Cepat Pemrograman Query dengan MySQL
- [9] Callum Macrae, 2018 Vue.js: Up and Running
- [10] A. P. Kusuma and K. A. Prasetya, “Perancangan Dan Implementasi E-Commerce Untuk Penjualan Baju Online Berbasis Android,” *Antivirus J. Ilm. Tek. Inform.*, vol. 11, no. 1, pp. 1–11, 2017, doi: 10.35457/antivirus.v11i1.194.
- [11] B. A. Priyaungga, D. B. Aji, M. Syahroni, N. T. S. Aji, and A. Saifudin, “Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 3, p. 150, 2020, doi: 10.32493/jtsi.v3i3.5343.
- [12] T. Mutiara, Achmad Benny, Muslim, A, Oswari, “Testing Implementasi Website Rekam Medis Elektronik,” *Pros. Semin. Ilm. Nas. Komput. dan Sist. Intelijen*, vol. 8, no. October, pp. 1–7, 2014.