

## PENGEMBANGAN APLIKASI LIVE VIDEO STREAMING BERBAYAR UNTUK KONSER MUSIK INDONESIA MENGGUNAKAN PROGRESSIVE WEB APPS

Rachmad Sanuri<sup>1\*</sup>, Momon Muzakar<sup>2\*</sup>, Aris Badaruddin Thoha<sup>3</sup>, Habib Aulia Raihan<sup>4</sup>

<sup>1</sup> Program Studi Sistem Informasi, STMIK El Rahma Yogyakarta  
<sup>2,3,4</sup> Program Studi Informatika, STMIK El Rahma Yogyakarta  
e-mail: <sup>1</sup>sanuri@stmikelrahma.ac.id, <sup>2</sup>muzakkarmomon@gmail.com ,  
<sup>3</sup>thoha@stmikelrahma.ac.id, <sup>4</sup>habibauliaraihan@gmail.com,

### ABSTRACT

*Live streaming is the broadcasting of video content in real-time over the internet. Technological advancements have provided new entertainment alternatives, such as watching music concerts through live streaming technology. The live broadcasting of music concerts has become a rapidly growing trend, offering music enthusiasts worldwide the opportunity to enjoy live performances without geographical limitations. With the emergence of digital platforms and the increased accessibility of high-speed internet, live-streamed music concerts can now reach a broader audience, surpassing the limited seating capacity of traditional concert venues. The use of Progressive Web Apps (PWA) in live streaming music concerts has become highly relevant due their ability to provide a fast, efficient user experience that is accessible across various devices. PWAs eliminate the need to download specialized applications from app stores. The implementation of PWAs in live streaming music concerts has shown significant results in enhancing user experience. The application was successfully installed on smartphones using PWA technology.*

**Keywords:** *Live Streaming, Progressive Web Apps, Music Concert*

### INTISARI

*Live streaming* adalah penyiaran konten video secara langsung di internet. Kemajuan teknologi telah memungkinkan bertambahnya alternatif hiburan untuk menonton konser musik dengan teknologi *live streaming*. Penyiaran langsung konser musik telah menjadi tren yang berkembang pesat, memberikan alternatif bagi penggemar musik di seluruh dunia untuk menikmati penampilan langsung tanpa batas geografis. Dengan munculnya platform digital dan peningkatan aksesibilitas internet berkecepatan tinggi, konser musik live streaming kini dapat diakses oleh audiens yang lebih luas, melampaui kapasitas tempat duduk yang terbatas pada venue konser tradisional. Penggunaan *Progressive Web Apps* (PWA) dalam penyelenggaraan konser musik secara *live streaming* menjadi sangat relevan mengingat kemampuannya dalam menghadirkan pengalaman pengguna yang cepat, efisien, dan dapat diakses melalui berbagai perangkat. PWA mengeliminasi kebutuhan untuk mengunduh aplikasi khusus dari *apps store*. Implementasi PWA dalam penyelenggaraan konser musik secara *live streaming* telah menunjukkan hasil yang signifikan dalam meningkatkan pengalaman pengguna. Aplikasi berhasil di install pada smartphone menggunakan teknologi PWA.

**Kata kunci:** *Live Streaming, Progressive Web Apps, Konser Musik Live*

### 1. PENDAHULUAN

Kemajuan teknologi telah memungkinkan bertambahnya alternatif hiburan untuk mengisi waktu luang dengan melihat streamer pada *platform live streaming*. *Live streaming* adalah penyiaran konten video secara langsung di internet [1]. Penyiaran langsung konser musik telah menjadi tren yang berkembang pesat, memberikan alternatif bagi penggemar musik di seluruh dunia untuk menikmati penampilan langsung tanpa batas geografis. Dengan munculnya platform digital dan peningkatan aksesibilitas internet berkecepatan tinggi, konser musik live streaming kini dapat diakses oleh audiens yang lebih luas, melampaui kapasitas tempat duduk yang terbatas pada venue konser tradisional [2].

Model bisnis *Pay Per View* (PPV) adalah sebuah model bisnis yang memungkinkan pengguna membayar hanya saat mengakses konten atau acara tertentu. Model bisnis ini biasanya digunakan dalam konteks hiburan, seperti pertandingan olahraga, konser, atau tayangan film. Pengguna membayar ketika ingin menonton konten yang di pilih, sehingga memberikan fleksibilitas yang tinggi. Cara kerja PPV adalah pengguna akan membayar biaya

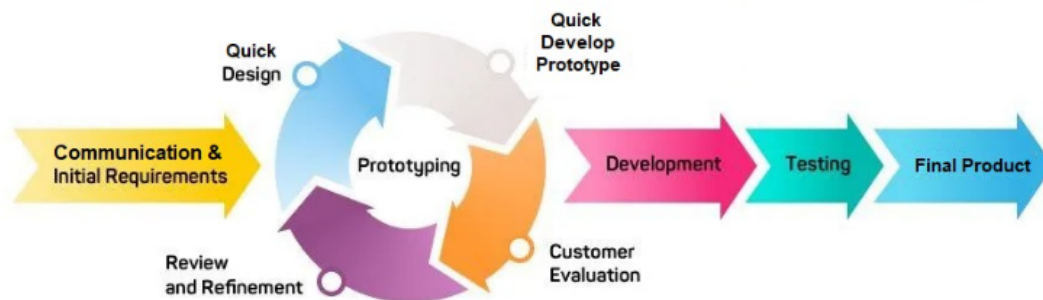
tertentu untuk mengakses konten yang diminati. Hal ini dapat dilakukan dengan berlangganan atau membayar per tayangan. Platform PPV sering kali menyediakan sistem pembayaran yang mudah dan aman untuk memfasilitasi proses ini.

Penggunaan *Progressive Web Apps* (PWA) sebagai platform dasar pembuatan aplikasi untuk penyelenggaraan konser musik secara live streaming menjadi sangat relevan. Mengingat kemampuannya dalam menghadirkan pengalaman pengguna yang cepat, efisien, dan dapat diakses melalui berbagai perangkat. PWA mengeliminasi kebutuhan untuk mengunduh aplikasi khusus dari app store, memudahkan pengguna dalam mengakses konten dengan satu klik pada browser. Selain itu, PWA mendukung fitur *offline*, *push notifications*, dan akses yang hampir setara dengan aplikasi native [3]. Dengan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan sebuah aplikasi live video streaming pada konser musik di Indonesia dengan memanfaatkan teknologi web seperti PWA untuk meningkatkan aksesibilitas dan pengalaman pengguna.

## 2. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini mencakup metode pengumpulan data seperti wawancara, observasi dan studi literatur sedangkan metode yang digunakan untuk pengembangan aplikasi *live streaming* pada penelitian ini akan menggunakan *prototype*.

Metode *Prototype* adalah proses merancang sebuah model dari suatu aplikasi sebagai bentuk contoh awal atau standar awal untuk suatu objek yang akan dikerjakan [4]. Adapun langkah – langkah pengembangan *prototype* ditunjukkan pada Gambar 1 dengan penjelasan sebagai berikut.



Gambar 1. Tahap Pengembangan Pengembangan Metode *Prototype*

### a. *Communication Initial Requirement*

*Initial Requirement* merupakan proses awal yang menjadi landasan dalam kegiatan pengembangan aplikasi.

### b. *Iteration Prototyping*

Tahap *prototyping* bisa dilakukan berulang (iterasi) hingga *prototype* sesuai dengan kebutuhan dan persepsi pengguna. Tahap ini terdiri dari beberapa sub phase sebagai berikut.

- 1) *Quick Desain*. Tahapan desain rencanakan tampilan dan fungsi aplikasi.
- 2) *Quick Develop Development*. Tahap pengembangan awal *prototype* sesuai dengan kebutuhan pengguna.
- 3) *Customer Evaluation*. Tahap evaluasi untuk mengidentifikasi kesesuaian kebutuhan pengguna
- 4) *Review And Refinement*. Tahap Penyempurnaan *prototype* berdasar hasil evaluasi dari pengguna.

### c. *Development*

*Development* atau pengembangan merupakan tahap yang digunakan untuk proses pembangunan suatu aplikasi sesuai dengan *prototype* yang telah disepakati.

### d. *Testing*

*Testing* atau pengujian merupakan suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari aplikasi atau layanan yang sedang diuji.

### e. *Final Product*

Final produk yang akan di *deploy* adalah *Progressive Web Application*, sebuah aplikasi yang dibangun menggunakan teknologi web modern, namun dapat berjalan seperti sebuah mobile apps [5].

## 3. HASIL DAN PEMBAHASAN

Berikut adalah implementasi tahap – tahap dalam pengembangan aplikasi *live video streaming* menggunakan metode *prototype*.

### 3.1 Tahap Design and Development

#### a. Communication Initial Requirements

##### 1) Penentuan Project Scope

Salah satu tahapan dalam *communication initial requirements* adalah penentuan *project scope*. Dalam *project scope* terdapat daftar fitur – fitur yang dibutuhkan pengguna dalam aplikasi. Berikut daftar fitur – fitur yang diidentifikasi. Daftar fitur dalam Tabel 1 dibawah ini yang akan dituangkan dalam *project deliverables* dari *project scope*.

Tabel 1. Daftar Fitur Aplikasi

No.	Fitur Aplikasi
1	Aplikasi memiliki fitur authentication pengguna
2	Aplikasi bisa menampilkan katalog event – event music
3	Aplikasi bisa melakukan transaksi pemilihan event musik yang dikehendaki
4	Aplikasi bisa menampilkan konten live streaming dari event musik yang dipilih
5	Admin aplikasi bisa menambah, mengubah, menghapus data katalog event
6	Admin aplikasi bisa mengatur transaksi

##### 2) Desain Model Use Case

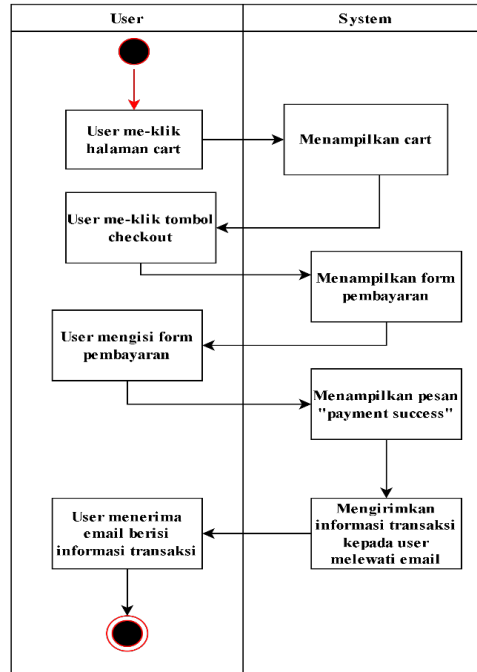
Untuk membantu mengetahui kebutuhan fungsional dari sebuah sistem digambarkan melalui *use case diagram* pada gambar 2. Use Case Diagram adalah bagian dari UML (*Unified Modeling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem[6].Diagram ini menggambarkan hubungan antara pengguna dengan sistem yang dirancang sebagai berikut



Gambar 2. Tahap Pengembangan

3) Desain Proses Bisnis

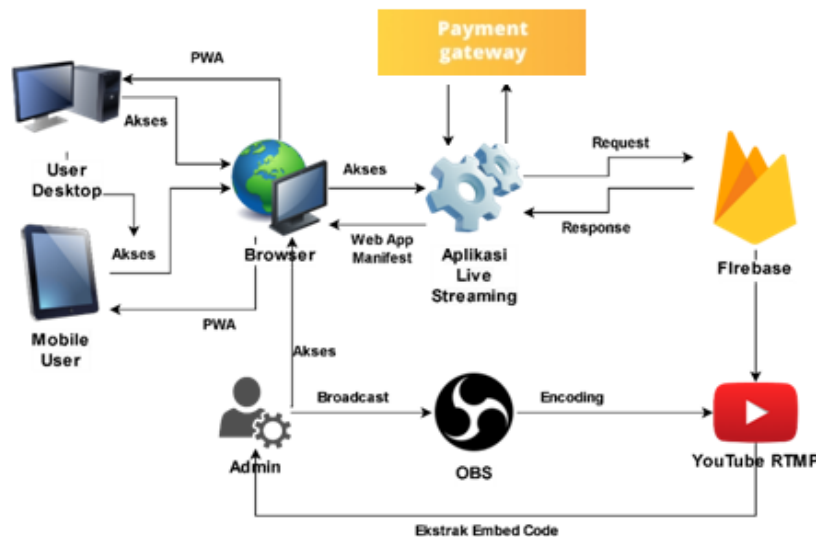
Desain proses bisnis menggunakan digambarkan melalui *activity diagram* berdasarkan model bisnis *Pay Per View (PPV)*. Pada gambar 3 berikut adalah proses bisnis utama PPV pada aplikasi *live streaming*. Alur *activity user* ketika melakukan *checkout* transaksi .



Gambar 3. Activity Diagram Proses Bisnis Pemilihan Video Dan User Checkout

4) Desain Arsitektur Sistem

Desain arsitektur sistem menunjukkan bagaimana sistem aplikasi bekerja. Desain arsitektur sistem ditunjukkan pada gambar 4.



Gambar 4. Desain Arsitektur Sistem Aplikasi Live Streaming.

Pada gambar 4 *user* bisa menggunakan *desktop* dan *mobile device* untuk mengakses aplikasi. *User* mengakses *browser* untuk mengakses aplikasi *live streaming*. Aplikasi *live streaming* menggunakan *Firebase* sebagai *back-end server*. Aplikasi mengirim *request* berupa *login authentication* untuk *user* dan akses *database*. Selanjutnya *Firebase* mengirim *response* yang diminta oleh aplikasi.

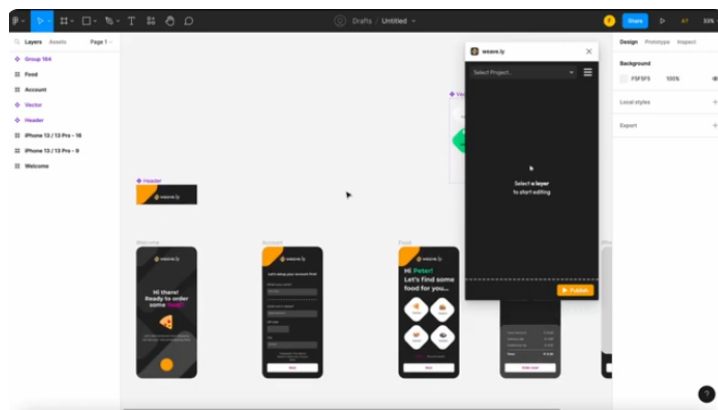
Aplikasi akan terhubung ke sistem pembayaran *online* melalui API *Payment Gateway*. Aplikasi mempunyai data *web app manifest* menggunakan teknologi *Progressive Web Apps (PWA)* yang dimiliki *browser* untuk membuat *user desktop* dan *mobile* bisa mengunduh aplikasi *live streaming*. Aplikasi yang akan dibangun menggunakan *React*. *React* berisi kumpulan snippet kode *JavaScript* (disebut 'komponen') yang bisa digunakan berulang kali untuk mendesain antarmuka pengguna [7].

b. **Prototyping**

*Prototyping* dilakukan dengan dengan dua iterasi untuk menemukan kesesuaian dengan kebutuhan pengguna

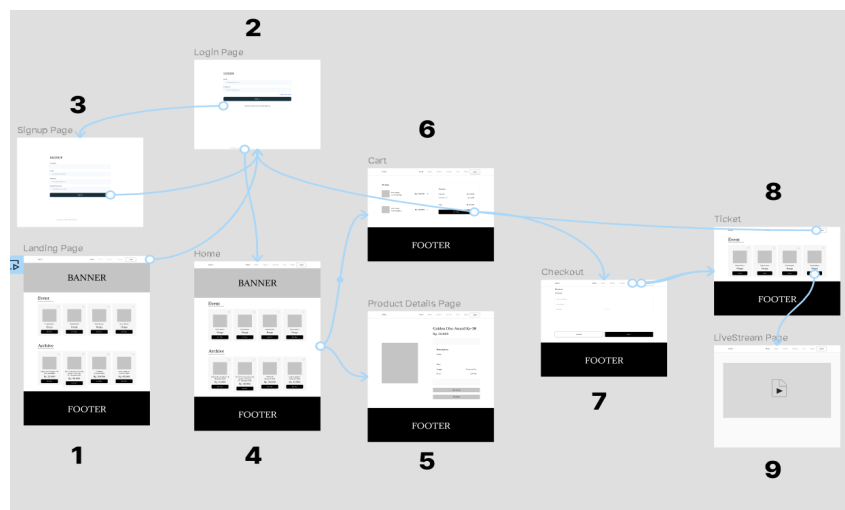
1) **Prototyping Iterasi Pertama**

*Prototyping* pertama pengembangan *User Interface (UI)* menggunakan aplikasi *Figma* sebagaimana pada gambar 5, untuk pembuatan *prototype* aplikasi dengan menggabungkan model *prototype low-fidelity* dan *digital* sebagai dasar komunikasi untuk memahami keinginan pengguna.



Gambar 5. Aplikasi Figma

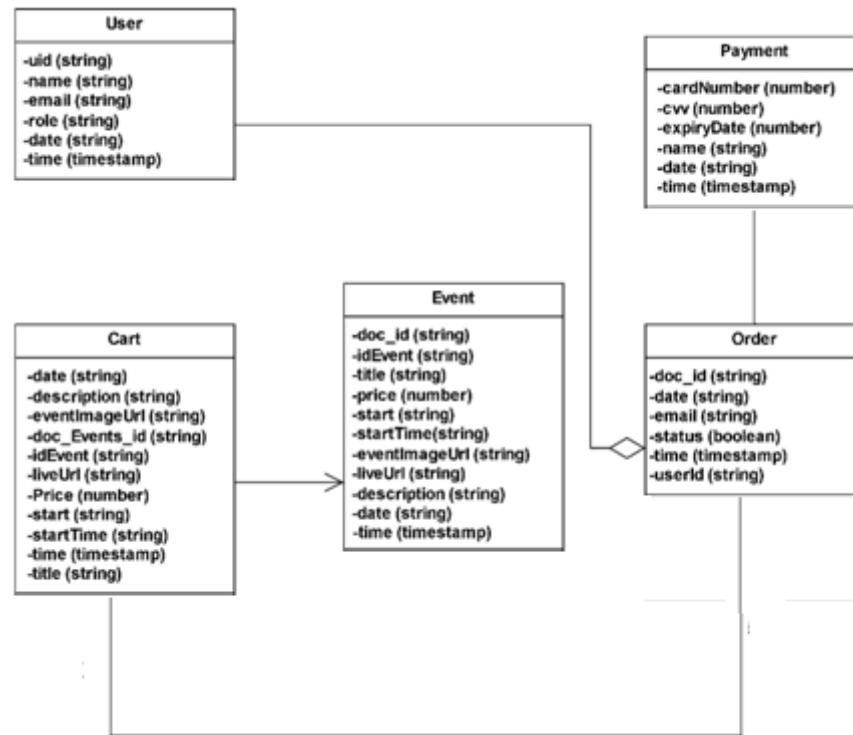
Pada gambar 6 adalah desain dan *mapping* UI secara keseluruhan dengan *landingpage(1)* sebagai tampilan awal, dari *landingpage* akan menuju halaman *login(2)*. Jika *user* belum memiliki akun, maka dari halaman *login* menuju halaman *signup(3)*. Setelah proses *signup* *user* akan kembali menuju halaman *login*. Setelah proses *login* *user* akan diarahkan ke halaman *home (4)*. Pada halaman *home*, *user* memiliki dua opsi, opsi pertama *user* akan diarahkan menuju halaman *detail event (5)* dan opsi kedua *user* akan diarahkan menuju halaman *cart(6)*. Pada halaman *cart*, *user* akan diarahkan pada halaman pembayaran yang berupa *pop-up window*. Setelah melakukan pembayaran, *user* menuju halaman *ticket(8)*. pada halaman *ticket* terdapat tombol *watch* yang dimana *user* akan diarahkan menuju halaman untuk menonton melalui sesi *livestreaming(9)*.



Gambar 6. Desain *Prototype* Aplikasi *Live Stream*

## 2) Prototyping Iterasi Kedua

Setelah prototyping pertama untuk menyesuaikan dengan kebutuhan pengguna dan mendesain tampilan, maka iterasi berikutnya adalah tahap mendesain database untuk menyimpan data aplikasi sebagai iterasi kedua. Rancangan *collection* dan *documents* sebagai NoSQL Database disesuaikan dengan form – form yang dirancang pada iterasi pertama disajikan pada gambar 7 sebagai berikut.



Gambar 7. Desain Collection Database NoSQL Aplikasi Live Stream

### c. Development

Dalam pengembangan aplikasi ada beberapa configuration setup yang perlu dipersiapkan saat masuk tahap *production*. Berikut adalah langkah *configuration setup* dan pengkodean yang dilakukan.

#### 1) Pemasangan *NodePackageManager* (NPM)

Pemasangan NPM langkah pertama yang perlu dilakukan sebagai *Node Package Manager* sebagai dasar pengembangan dengan *Javascript*

#### 2) Pemasangan *React Vite*.

*React Vite* adalah salah satu *framework javascript* yang digunakan untuk membangun aplikasi dengan cepat dan mudah. *React Vite* adalah sebuah alat modern dan inovatif dalam ekosistem pengembangan *front-end JavaScript*, dirancang untuk mempercepat dan mempermudah proses pembangunan aplikasi. Alternatif pengganti untuk bundler seperti *Webpack* dan *Parcel* [8]

#### 3) Pemasangan *Tailwind CSS*.

*Tailwind CSS* kerangka kerja CSS yang mengutamakan utilitas yang dibuat untuk mempercepat dan menyederhanakan pembuatan aplikasi yang menggunakan desain unik.

#### 4) Konfigurasi *Web AppManifest*

*Web AppManifest* adalah sebuah file JSON yang menyediakan informasi tentang aplikasi web agar bisa diinstal di perangkat pengguna dan berfungsi seperti aplikasi native

#### 5) Setup *Firebase*.

*Firebase* adalah salah satu platform *Backend as a Service* (BaaS) dari *Google* dibuat untuk mempermudah pengembangan aplikasi. Pada *firebase* disediakan berbagai layanan untuk autentikasi, database, pengujian dan analitik yang semuanya dirancang untuk mendukung pengembangan aplikasi untuk platform mobile dan platform web agar lebih efektif dan efisien[9].

#### 6) Setup *Emailjs*.

*Emailjs* adalah pustaka *JavaScript* yang menyederhanakan proses pengiriman email dari aplikasi sisi klien, seperti aplikasi web atau aplikasi seluler.

#### 7) Setup *Open BroadcastSoftware* (OBS).

*Open Broadcast Software* (OBS) digunakan sebagai encoder untuk menampilkan *live streaming* kedalam *YouTube*,

8) *Setup YouTubeCreator Studio*.

*YouTubeCreatorStudio* digunakan untuk menyiarkan *live streaming* dari OBS menuju *YouTube* RTMP (*real-time messaging protocol*) terdapat pada *YouTubeCreator Studio*

9) Pengkodean.

Pengkodean dilakukan menggunakan framework *ReactJS* dengan IDE *Visual Studio Code*, suatu editor atau IDE gratis ini dapat dijalankan pada perangkat desktop pada lingkungan *Linux*, *Microsoft Windows*, dan *Mac OS* yang mana dikembangkan oleh *Microsoft*. Aplikasi *visual studio code* adalah sebuah software editor powerful namun tetap ringan saat digunakan [10]

**d. Testing**

*Testing* atau pengujian diperlukan untuk mengetahui sistem sudah sesuai dengan kebutuhan pengguna. Pengujian dilakukan menggunakan *Blackbox testing* dan *Lighthouse*.

1) **Blackbox Testing**

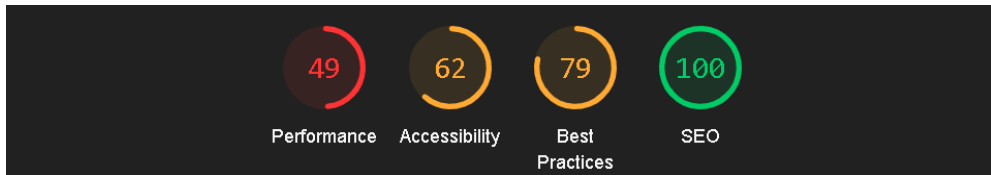
Pengujian dilakukan dengan metode *blackbox testing* yaitu dengan melihat fungsi masukan dan keluaran sistem aplikasi apakah sudah sesuai dengan kebutuhan pengguna atau masih memiliki *bug* atau *error*. Tabel 2 merupakan hasil pengujian aplikasi *live streaming* menggunakan *progressive web app*. Berdasarkan hasil pengujian menggunakan *blackbox testing* menunjukkan bahwa tidak ditemukan adanya *bug* atau *error*.

**Tabel 2.** Hasil Pengujian Menggunakan *Blackbox Testing*

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1.	Email dan <i>Password</i> salah saat <i>login</i>	Muncul dialog " <i>loginfailed</i> "	<i>Valid</i>
2.	Email dan <i>Password</i> benar saat <i>login</i>	Muncul dialog " <i>LoginSuccessful</i> "	<i>Valid</i>
3.	Admin memberi <i>input</i> kosong saat menambah <i>event</i>	Muncul dialog " <i>allfield are required</i> "	<i>Valid</i>
4.	Admin menambah <i>event</i>	Muncul dialog " <i>addeventsuccessful</i> "	<i>Valid</i>
5.	Admin mengedit data <i>event</i>	Muncul dialog " <i>updateeventsuccessful</i> "	<i>Valid</i>
6.	Admin menghapus data <i>event</i>	Muncul dialog " <i>Delete Event Successful</i> "	<i>Valid</i>
7.	<i>User</i> melakukan <i>payment</i>	Muncul dialog " <i>PaymentSuccessful</i> "	<i>Valid</i>
8.	Admin menyetujui transaksi	Muncul dialog " <i>Order Approved</i> " dan dialog " <i>Notification Email SendSuccessfully</i> "	<i>Valid</i>
9.	Admin menolak transaksi	Muncul dialog " <i>Order Rejected</i> " dan dialog " <i>Notification Email SendSuccessfully</i> "	<i>Valid</i>
10.	<i>Order</i> disetujui admin	<i>Event</i> muncul pada halaman <i>Ticket</i>	<i>Valid</i>
11.	<i>Order</i> ditolak admin	<i>Event</i> tidak muncul pada halaman <i>Ticket</i>	<i>Valid</i>
12.	<i>Notification</i> dikirim sesuai dengan waktu yang ditentukan	Mengirimkan notifikasi "event 15 menit lagi dimulai"	<i>Valid</i>

2) **Lighthouse**

*Lighthouse* adalah alat otomatis yang digunakan untuk mengukur kualitas aplikasi *web*. Alat ini menyediakan audit untuk kinerja, aksesibilitas, aplikasi *web* progresif (PWA), SEO, dan lainnya. Skor dari pengujian menggunakan *Lighthouse* ditunjukkan pada Gambar 8.



Gambar 8. Hasil Lighthouse Report

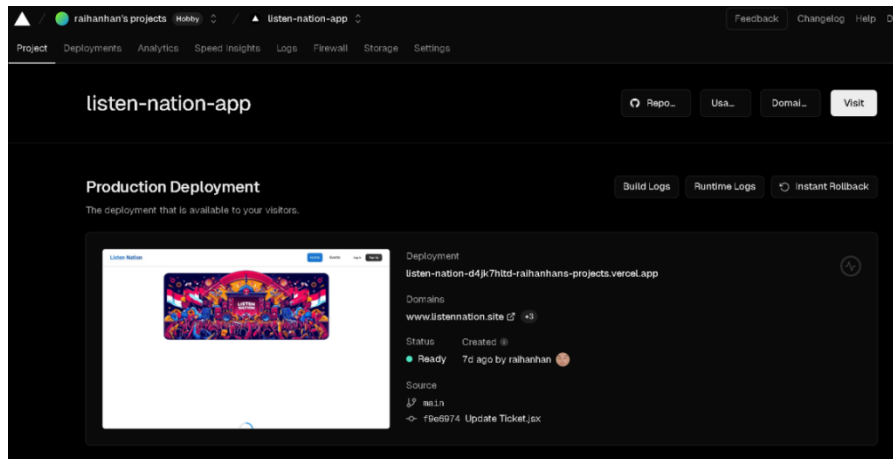
Skor 49 untuk *Performance* menunjukkan bahwa kinerja aplikasi *web* berada di bawah standar. Hal ini disebabkan oleh beberapa faktor seperti waktu pemuatan yang lambat, ukuran *file* yang besar, dan terlalu banyak *request network*. Skor 62 untuk *Accessibility* menunjukkan bahwa aksesibilitas aplikasi *web* memerlukan peningkatan. Berarti bahwa beberapa pengguna, terutama dengan disabilitas, kesulitan mengakses atau menggunakan situs *web*. Skor 79 untuk *Best Practice* menunjukkan bahwa aplikasi *web* sebagian besar sudah mengikuti praktik terbaik dalam pengembangan *web*, tetapi masih ada beberapa perbaikan.

### 3.2. Tahap Final Project

Tahap final project adalah implementasi dari tahapan – tahapan sebelumnya

#### a. Tahap *deployment*

Dalam pengembangan perangkat lunak *deployment* adalah proses pengiriman aplikasi yang sudah selesai dikembangkan ke lingkungan produksi atau pengguna akhir. Aplikasi *live streaming* berhasil di *deploy* menggunakan *Vercel.com*. *Vercel* adalah platform *hosting* dan layanan *cloud* yang dirancang khusus untuk aplikasi *web* dan situs statis, hasil *deploy* ditunjukkan pada Gambar 9.

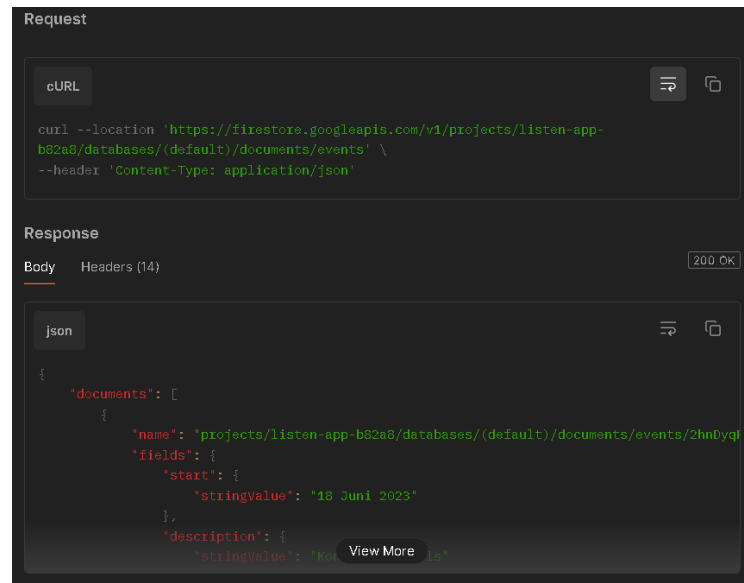


Gambar 9. Deployment Dengan Vercel

#### b. Implementasi *RESTful API Backend*

*RESTful API* (*Representational State Transfer API*) adalah jenis arsitektur API yang menggunakan prinsip-prinsip REST untuk memungkinkan komunikasi antara klien dan server. Aplikasi *livestream* menggunakan *RESTful API* dari *Firebase* yang memungkinkan aplikasi untuk melakukan operasi *CRUD* (*Create, Read, Update, Delete*) menggunakan *HTTP methods* (*GET, POST, PUT, DELETE*). Berikut adalah dokumentasi salah satu fungsi API dari aplikasi *livestream* yang sudah diuji menggunakan *Postman* pada gambar 10.



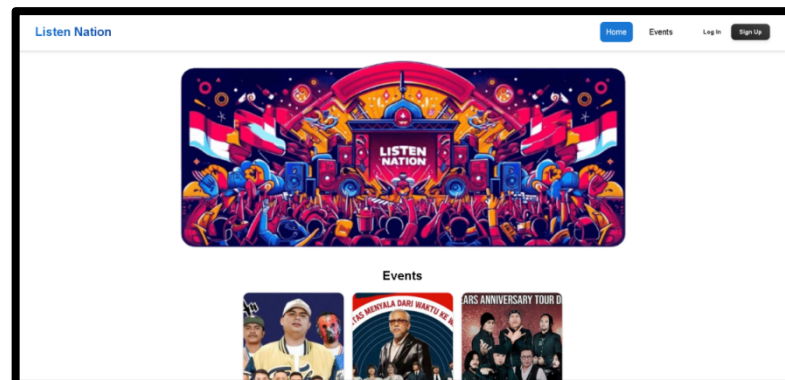


Gambar 10. Hasil Pengujian *Get All Events*.

1

c. Hasil Pengembangan Aplikasi

Berikut ini tampilan antarmuka dari aplikasi *livestreaming* sesuai dengan perancangan yang sebelumnya sudah dibuat. PWA adalah jenis aplikasi web yang memadukan keunggulan aplikasi web dan aplikasi mobile dalam satu paket yang kuat. Tampilan gambar 11 merupakan hasil ketika aplikasi dibuka menggunakan browser pada PC atau sejenisnya. Sedangkan pada gambar 12 merupakan hasil ketika aplikasi dibuka menggunakan perangkat *mobile*.

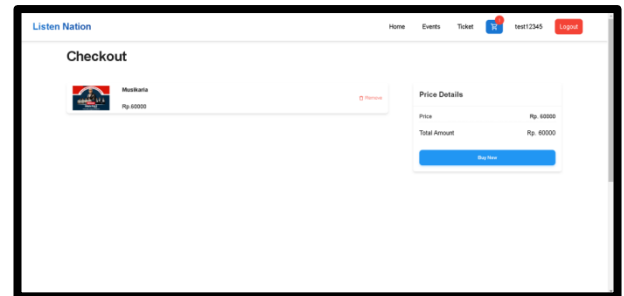
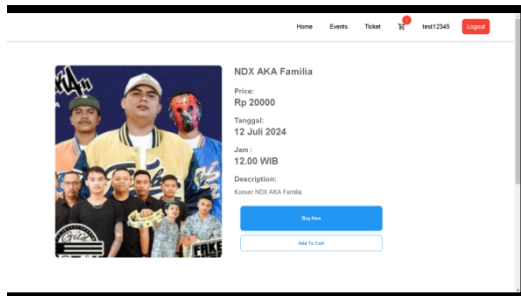


Gambar 11. Tampilan *Home* Sebelum *UserLogin* Pada Aplikasi Web



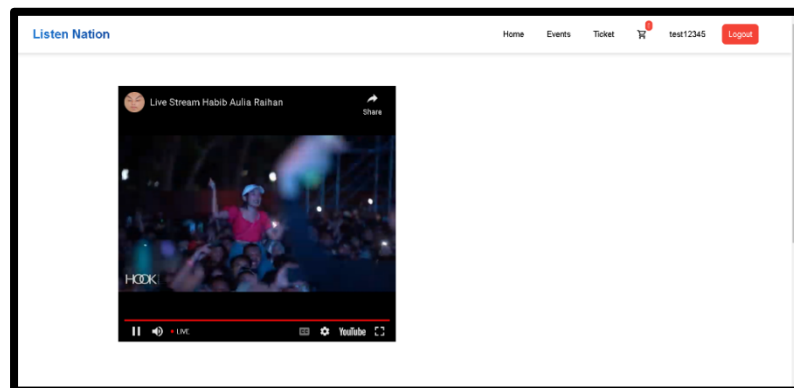
Gambar 12. Tampilan Home Sebelum UserLogin Pada Aplikasi Mobile

Halaman *detailevent* pada gambar 13 berisi item acara poster konser, judul konser, harga tiket, tanggal, jam dan deskripsi konser yang memuat *gueststar* dan informasi tambahan lainnya. Halaman *cart* menampilkan *event* yang siap *checkout*. Terdapat juga informasi *detail* harga seperti *price* dan *total amount*.



Gambar 13. Tampilan Halaman *Detail Event* Dan *Cart Checkout*

Tampilan halaman *live* ditunjukkan pada gambar 14. *Live page* menampilkan *media live streaming* yang di *mirror* dari platform *YouTube* yang bisa diakses oleh *user* apabila *user* telah membeli tiket *event*.



Gambar 14. Tampilan Tampilan *Live Page*

Halaman *total event* pada gambar 15 berisi tabel keseluruhan *event* dan berisi informasi seperti gambar, judul *event*, *idevent*, harga, tanggal mulai, jam tayang, tanggal ditambahkan *event* dan *action* untuk mengedit dan menghapus *event* dari *database*.



S.No.	Image	Title	Event id	Price	Start Date	Start Time	Date	Action	Action
1.		30 Years Anniversary Dewa 19	141829e4-3d28-4134-b33a-6329564cf8cc	Rp.50000	12 Juli 2024	12.00 WIB	Jul 10, 2024	Edit	Delete
2.		NDX.AKA Famila	434e97df-0bee-4981-be43-a8876b6a4a30	Rp.20000	12 Juli 2024	12.00 WIB	Jul 11, 2024	Edit	Delete
3.		Musikana Iwan Fals	49d059f-d474-494e-a408-75e30a99877b	Rp.80000	18 Juni 2023	14.00 WIB	Jul 11, 2024	Edit	Delete

Gambar 15. Tampilan Total Events

#### 4. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengembangkan sebuah aplikasi *live video streaming* berbayar yang memanfaatkan teknologi *Progressive Web App* (PWA). Teknologi PWA dipilih karena kemampuannya memberikan pengalaman pengguna yang hampir setara dengan aplikasi native, dengan kinerja yang baik dan aksesibilitas yang mudah melalui *web browser*. Desain antarmuka pengguna (UI) menjadi salah satu fokus utama dalam penelitian ini, menekankan pentingnya tampilan yang menarik dan interaktif untuk meningkatkan keterlibatan pengguna, serta menciptakan pengalaman yang lebih menyenangkan dan intuitif. Penerapan teknologi PWA dalam aplikasi ini telah terbukti berhasil, memungkinkan aplikasi untuk diunduh tanpa melalui app store dan berfungsi dengan baik pada perangkat *smartphone*. Walaupun pengembangan aplikasi *live streaming* pada penelitian ini sudah berhasil tetapi memerlukan pengembangan lebih lanjut untuk mendukung teknologi *blockchains* guna autentifikasi konten dan heterogenitas metode pembayaran, serta masih belum memaksimalkan penggunaan *caching* sebagai upaya untuk meningkatkan performa aplikasi.

#### DAFTAR PUSTAKA

- Agustina, L.(2019). Live Video Streaming Sebagai Bentuk Perkembangan Fitur Media Sosial. *Jurnal Media dan Komunikasi*, vol. 1, no. 1, 17–23.
- Zakaria, F., Wijaya, G.(2022). Dari Pertunjukan Ke Media: Konvergensi Festival Musik Dalam Merespons Pandemi Covid – 19, *Jurnal Media dan Komunikasi Indonesia*, vol. 3, no. 1, 4–19.
- Biørn-Hansen, A., Majchrzak, T.A., & Grønli, T.M. (2019). Progressive Web Apps: The Possible Web-Native Unifier For Mobile Development, *Proceedings of the 13th International Conference on Web Information Systems and Technologies*, vol. 1, no. 1, pp. 344–351. *WEBIST*
- Maraya.(2020), *Metode Prototype : Kelebihan, Kekurangan & Tahapan Model*, diakses tanggal 3 Juli 2024 pukul 23:18, dari <https://Dosenit.Com/Software/Metode-Prototype>.
- Santoso, H. (2019). Membangun Aplikasi Mobile dengan PWA (Progressive Web App), *Lokomedia*, Vol. 1.
- Dicoding (2021). *Apa Itu UML? Beserta Pengertian dan Contohnya*. diakses tanggal 16 Juni 2024 pukul 17:10, dari <https://www.dicoding.com/Blog/Node Package-Manager>.
- Faradila, A.(2023), *Apa Itu React? Pengertian dan Cara Kerja ReactJS*. diakses tanggal 1 Juli 2024 pukul 20:14, dari <https://www.hostinger.co.id/tutorial/apa-itu-react>.
- Solikhul, M.(2024), *Cara Install dan Membuat Project React Menggunakan Vite*, diakses tanggal 3 Juni 2024 pukul 16:10, dari <https://medium.Com/@batuamin123/cara-install-dan-membuat-project-react-menggunakan-vite-0ab931307482>.
- Dicoding. (2020), *Apa Itu Firebase? Pengertian, Jenis – Jenis dan Fungsi Kegunaanya*, diakses tanggal 21 Juni 2024 pukul 20:10, dari <https://www.dicoding.com/blog/apa-itu-javascript-fungsi-can-contohnya>.
- Huda,N.(2022), *Visual Studio Code: Pengertian, Fitur, Keunggulan dan Jenisnya*, diakses tanggal 3 Juli 2024 pukul 23:18, dari <https://www.dewaweb.com/blog/mengenal-visual-studio-code>.