



Implementasi React Vite dalam Pengembangan Antarmuka Sistem Pemesanan Tiket Pesawat dengan Metode Scrum

Tegar Alfa Rizzi¹, Rafly Aziz Abdillah², Yogi Efani Yancandra³, Melinda Wijaya⁴, Apriade Voutama⁵

^{1,2}Sistem Informasi, Universitas Singaperbangsa Karawang, Jl. HS.Ronggo Waluyo, Karawang, 41361, Indonesia.

²Sistem Informasi, Universitas Pembangunan Nasional Veteran Jakarta, Jl. R.S Fatmawati, Jakarta Selatan, 12450, Indonesia.

³Informatika, Universitas Amikom Yogyakarta, Jl. Ring Road Utara, Yogyakarta, 55283, Indonesia.

⁴Teknik Informatika, Universitas Negeri Semarang, Sekaran Kec. Gn. Pati, Semarang, 50229, Indonesia.

*Penulis Korespondensi, Email: tegaralfa18@gmail.com

Abstrak—Penelitian ini membahas mengenai pengembangan antarmuka sistem informasi pemesanan tiket pesawat. Metode yang dipakai dalam pengembangan antarmuka sistem ini adalah Scrum. Scrum dipilih karena dapat membagi proses pengembangan kedalam beberapa Sprint. disetiap akhir Sprint akan menghasilkan saran serta evaluasi untuk pengembangan berikutnya, sehingga efektivitas dari setiap proses pengembangan dapat terus meningkat dan menjadi lebih baik. Sistem yang dikembangkan menggunakan React Vite sebagai teknologi utama dapat mempercepat proses pengembangan dan meningkatkan pengalaman pengguna karena React mendukung Single Page Application yang dapat memudahkan untuk melakukan navigasi pada website yang telah dibangun. Tujuan lain yang ingin dicapai dalam penelitian ini adalah mengurangi penggunaan kertas dengan menyediakan fitur unduh tiket dalam format digital menggunakan teknologi React PDF. Selain fitur unduh tiket, ada beberapa fitur utama yang telah dikembangkan, diantaranya yaitu fitur pemesanan tiket, fitur pembayaran tiket, fitur order histori, dan fitur notifikasi. Hasil dari penelitian ini berupa dokumentasi proses pengembangan antarmuka yang berdasar kedalam metode Scrum, dimana penelitian ini menunjukkan bahwa seluruh fitur yang dikembangkan sudah berjalan dengan optimal.

Kata Kunci: Scrum; Tiket; React; Vite; Website.

Abstract—This research discusses the development of an airline ticket booking information system interface. The Scrum methodology was employed in the interface development process. Scrum was chosen for its ability to divide the development process into several Sprints. At the end of each Sprint, suggestions and evaluations for subsequent development are produced, thereby continuously enhancing the effectiveness of each development process. The system, developed using React Vite as the main technology, accelerates the development process and improves user experience through React's Single Page Application architecture, which enables smoother and more efficient navigation throughout the website. Another objective of this research was to reduce paper usage by providing a feature to download tickets in digital format using React PDF technology. Beyond this digital ticket feature, the system incorporates several core functionalities, including ticket booking, payment processing, order history tracking, and notification services. The result of this research is documentation of the interface development process based on the Scrum methodology, demonstrating that all developed features are functioning optimally.

Keywords: Scrum; Ticket; React; Vite; Website.

1. PENDAHULUAN

Teknologi informasi berkembang dengan sangat cepat. Manusia bisa dipermudah dalam melakukan berbagai hal dengan berkembangnya teknologi [1]. Manfaat dari teknologi informasi tersebut yaitu dapat menghasilkan informasi yang akurat, tepat, dan jelas [2]. Salah satu contoh dari penerapan dalam perkembangan teknologi informasi tersebut adalah sistem informasi penjualan tiket secara online. Sistem ini dapat memudahkan masyarakat untuk melakukan proses transaksi dengan lebih efektif dan efisien [3].

Di sisi lain, masih ada masyarakat yang melakukan pembelian tiket secara konvensional, seperti dalam pemesanan tiket pesawat yang memerlukan waktu dan tenaga yang lebih karena penumpang harus datang ke loket bandara dengan menunggu dan mengantri untuk melakukan pembelian tiket. Pada proses ini juga penumpang bisa beresiko kehabisan tiket, terlebih pada musim liburan atau saat permintaan pembelian tiket sedang tinggi [4]. Hal tersebut dapat diatasi dengan menyediakan sistem informasi yang mudah digunakan dan *user-friendly* agar masyarakat tidak merasa kesulitan ketika menggunakan sistem tersebut [5].

Sistem yang *user-friendly* juga tidak hanya bergantung dengan desain antarmuka yang cantik, tetapi juga pada kecepatan akses dan performanya. Namun, masih banyak website tradisional yang menggunakan sistem *Server Side Rendering* memiliki navigasi yang lebih berat dan lambat dikarenakan adanya *reload* dari *server* setiap berpindah halaman. Hal ini bukan hanya membuat pengalaman pengguna menjadi buruk tetapi juga dapat meningkatkan beban *server*, terutama saat jumlah pengguna yang mengakses sistem dalam waktu yang bersamaan cukup tinggi [6].

Oleh karena itu dibutuhkan solusi untuk meningkatkan kenyamanan pengguna serta dapat meningkatkan kinerja dari sistem yang dikembangkan. Solusi yang dapat diterapkan yaitu dengan menggunakan *React Vite* yang memungkinkan pengembangan antarmuka dari website berbasis *Single Page Application*. Dari sistem antarmuka ini diharapkan agar dapat meningkatkan pengalaman pengguna yang lebih baik, responsif, dan lebih cepat. Hal ini dapat terjadi karena sistem antarmuka yang berbasis *Single Page Application* tidak memerlukan *reload* halaman sehingga dapat memudahkan navigasi yang dilakukan oleh pengguna didalam sistem. Berdasarkan hal tersebut, penelitian ini memiliki tujuan untuk membangun sistem antarmuka pemesanan tiket pesawat berbasis website menggunakan *React Vite*, karena *React* mendukung pengembangan antarmuka yang berbasis *Single Page Application*.

Berbagai penelitian sudah dilakukan untuk membuat serta merancang sistem pembelian tiket secara online. Arif Indra Satria (2020) didalam penelitiannya yang berjudul “Perancangan Sistem Informasi Manajemen Pemesanan Tiket Pesawat pada Travel Agent Wiro Karya Berbasis Web dengan Metode *Waterfall*” mengembangkan sistem pembelian tiket pesawat berbasis website untuk menggantikan sistem manual sebelumnya. Sistem yang dirancang menggunakan metode *Waterfall* dengan bahasa pemrograman *PHP* [7]. Senada dengan itu, Ari Yanti Rahmadhani (2021) dalam penelitian berjudul “Penerapan Metode *Waterfall* Pada Sistem Informasi Penjualan Tiket Pesawat Terbang” merancang sistem informasi yang dapat membantu dalam proses transaksi penjualan tiket guna menghindari kesalahan input manual. Sistem yang dibuat menggunakan *Delphi 7.0*, database *MySQL* dengan testing menggunakan *Blackbox* [8]. Adelia Guskani (2022) dalam penelitian “Perancangan Sistem Informasi Pemesanan Tiket Pesawat Domestik Berbasis Web Pada CV Jenika Group Depok” juga menggunakan metode *Waterfall* untuk membangun sistem yang dapat mempermudah admin dan pelanggan dalam proses pemesanan tiket, sistem dibangun menggunakan bahasa pemrograman *PHP* dan *Javascript* [9]. Berbeda dengan penelitian sebelumnya, pada penelitian Joice Margaretha (2023) yang berjudul “Perancangan Sistem Informasi Pemesanan Tiket Konser Musik Berbasis Web Menggunakan *Unified Modeling Language (UML)*” lebih berfokus pada sistem pemesanan tiket konser musik dengan menggunakan pendekatan perancangan *Unified Modeling Language (UML)* [10]. Sementara itu, Shafly Firmansyah (2024) dalam penelitiannya yang berjudul “Penerapan *UML* dalam Sistem Pemesanan Tiket Bioskop Berbasis Website” menerapkan metode *Software Development Life Cycle (SDLC)* dengan memanfaatkan *Unified Modeling Language (UML)* untuk mendesain dan memodelkan sistem yang akan dibangun [11].

Dari penelitian sebelumnya sistem yang dibuat masih banyak menggunakan *Server Side Rendering* yang memerlukan *reload* halaman dan dapat mengurangi pengalaman pengguna. Berbeda dengan hal tersebut, penelitian ini menerapkan *React Vite* untuk membangun sistem yang berbasis *Single Page Application*, sehingga navigasi dapat lebih cepat dan juga interaktif. Selain itu, penelitian ini juga menambahkan fitur unduh atau cetak tiket dengan kode *boarding* dalam format *PDF* menggunakan salah satu *library Javascript*, yaitu *React PDF* yang belum diimplementasikan dalam penelitian sebelumnya.

Javascript adalah bahasa pemrograman tingkat tinggi yang awalnya dikembangkan oleh Netscape Communications Corporation untuk meningkatkan interaktivitas pada halaman website. Pada saat ini *Javascript* sudah berkembang menjadi teknologi yang paling banyak digunakan untuk pengembangan sistem, baik dari sisi *server* maupun *client*. Popularitas *Javascript* didukung oleh beragamnya *framework* dan *library* yang mempermudah pengembangan aplikasi yang akan dibuat [12].

Javascript memiliki banyak *library* salah satunya adalah *React*. *React* adalah *library* yang dikembangkan oleh Facebook pada tahun 2013 yang saat ini dikenal dengan nama *META*. *React* menjadi populer karena efisien dalam menangani pembaruan data, terutama untuk aplikasi dengan *traffic* yang tinggi. *React* dikembangkan karena Facebook menghadapi kendala dalam pemeliharaan aplikasi yang semakin kompleks. Karena hal tersebut mereka mulai mengembangkan *React* dengan *Virtual DOM* yang hanya memperbarui komponen yang berubah membuat sistem yang dikembangkan menjadi lebih cepat dan efisien [13].

Masih bagian dari *library Javascript*, *React PDF* merupakan *library* yang berbasis dari *React* yang digunakan untuk merender dan menghasilkan dokumen dalam format *PDF* dilingkungan *client-side* maupun *server-side*. *Library* ini memungkinkan pengembang untuk memberikan *style* dan struktur dokumen *PDF* dengan komponen *React* yang dapat dikostumisasi dengan *style CSS*. Dengan hal ini *React PDF* memberikan fleksibilitas dalam pembuatan dokumen otomatis yang dapat disesuaikan dengan kebutuhan dari website yang akan dikembangkan [14].

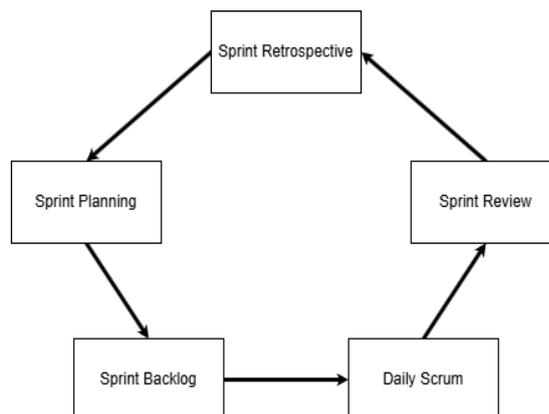
Berbeda dengan *React*, *Vite* adalah alat bantu yang dapat menyederhanakan pengembangan website modern. *Vite* digunakan untuk mengembangkan antarmuka dari sebuah website. *Vite* lebih terfokus pada *frameworks Javascript* dan juga *TypeScript*, seperti *React*, *Vue*, dan *Svelte*. *Vite* menjadi populer karena mampu mengoptimalkan kode

dengan menghapus dependensi yang tidak digunakan, sehingga kinerja dari aplikasi web dapat menjadi lebih baik [15].

Scrum merupakan metode pengembangan perangkat lunak yang dirancang untuk menjawab keterbatasan pendekatan tradisional seperti metode *Waterfall*. Termasuk kedalam kategori metode *Agile*, *Scrum* lebih berfokus pada kerja sama tim, fleksibilitas terhadap perubahan, serta proses pengembangan yang dilakukan secara bertahap. Sebagai sebuah kerangka kerja, *Scrum* mendorong kolaborasi antar anggota tim, pembangunan fitur secara inkremental, dan penggunaan proses iteratif untuk mencapai hasil yang ideal. Pendekatan ini sangat sesuai untuk digunakan dalam proyek-proyek yang kompleks dan memiliki kebutuhan yang sering berubah, seperti yang umum dijumpai dalam pengembangan website modern [16]. didalam *Scrum* terdapat *Sprint* yaitu periode kerja terstruktur yang bertujuan untuk menghasilkan fitur atau sebuah peningkatan produk yang dapat digunakan. *Sprint* memiliki durasi pengerjaan yang tetap, bisa 1 sampai 4 minggu, hal ini tergantung dari kompleksitas serta tingkat kesulitan proyek yang dikembangkan. *Sprint* terdiri dari beberapa tahapan utama, yaitu *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* [17]. Dalam penelitian ini, metode *Scrum* dipilih sebagai pendekatan pengembangan yang paling sesuai untuk antarmuka sistem pemesanan tiket pesawat. Keunggulan utama dari *Scrum* jika dibandingkan dengan metode tradisional seperti *Waterfall* adalah fleksibilitasnya dalam mengakomodasi perubahan fitur atau persyaratan. Berbeda dengan *Waterfall* yang mengharuskan semua persyaratan didefinisikan diawal, *Scrum* memungkinkan pengembangan secara bertahap dan adaptif yang sangat dibutuhkan dalam pengembangan antarmuka website modern. Hal ini penting karena antarmuka pengguna sering memerlukan penyesuaian berdasarkan *feedback* dari pengguna sistem, terutama untuk sistem pemesanan tiket yang membutuhkan pengalaman pengguna yang optimal dan ideal. Dengan adanya *Scrum*, setiap fitur seperti, login, pencarian tiket atau proses pemesanan dapat dikembangkan dan diuji dalam setiap *sprint* yang terpisah, sehingga dapat memastikan kualitas dan fungsionalitasnya berjalan dengan baik pada setiap *Sprint* sebelum masuk ke tahap berikutnya. Pendekatan ini memungkinkan pengembangan antarmuka yang lebih responsif dan hasil akhir yang lebih sesuai dengan kebutuhan dari pengguna sistem.

2. METODOLOGI PENELITIAN

Pada penelitian ini berfokus kepada pengembangan antarmuka sistem pemesanan tiket pesawat. Pengembangan dilakukan dengan menggunakan metode *Scrum*. *Scrum* dipilih karena dapat memudahkan pengembang untuk memilih fitur apa saja yang paling dibutuhkan dan diprioritaskan agar dapat dikerjakan lebih awal. *Scrum* juga dapat membantu pengembang untuk mendapatkan evaluasi berkala dari proyek yang sedang dikerjakan disetiap akhir *Sprint*. Selain itu *Scrum* sangat sesuai dengan pengembangan tim karena dapat meningkatkan fleksibilitas, efisiensi, serta komunikasi antar tim. Pengembangan antarmuka sistem pemesanan tiket pesawat ini dilakukan dengan menggunakan tahapan penelitian yang digambarkan pada gambar 1. Setiap tahapan penelitian selanjutnya dijelaskan dan sub-sub bagian dalam naskah ini.



Gambar 1. Metode pengembangan *Scrum*

2.1.1 Sprint Planning.

Tahapan pertama dari pengembangan sistem dengan metode *Scrum* adalah *Sprint Planning*. Di tahap ini tim akan mengadakan pertemuan awal dengan tujuan untuk menentukan tugas-tugas yang akan dikerjakan dalam satu siklus *Sprint* kedepan. *Sprint* sendiri ialah periode waktu tetap pengerjaan. Dimana tim berupaya untuk menyelesaikan tugas yang telah ditetapkan sebelumnya. Selama *Sprint* planning ini berlangsung, tim akan membahas berbagai aspek. Mulai dari fitur apa saja yang harus ditambahkan atau ditingkatkan, tugas-tugas yang harus diselesaikan,

serta alokasi pekerjaan diantara anggota tim. Dalam proses ini anggota tim harus memahami tujuan dari *Sprint* yang akan berjalan serta tanggung jawab masing-masing.

2.1.2 Sprint Backlog

Setelah *Sprint Planning* selesai, tahapan yang akan dikerjakan selanjutnya adalah *Sprint Backlog*. *Sprint Backlog* berisi catatan dari tugas-tugas yang harus dikerjakan selama periode *Sprint* berlangsung. Tugas-tugas ini berasal dari hasil diskusi yang dilakukan pada tahap *Sprint Planning* sebelumnya. Guna memastikan pekerjaan yang telah direncanakan berjalan dengan baik, tim pengembang dapat menggunakan alat bantu seperti *Scrum Board*. *Scrum Board* adalah papan visual yang membantu dalam mengelola dan memantau perkembangan dari tugas-tugas selama *Sprint* berlangsung. Dengan alat bantu ini dapat memungkinkan tim untuk melacak kemajuan pekerjaan secara sistematis, memantau tugas yang sedang dikerjakan, tugas yang telah selesai, serta dapat melihat kendala yang muncul selama *Sprint* berlangsung.

2.1.3 Daily Scrum

Daily Scrum merupakan pertemuan harian yang dilakukan oleh tim pengembang selama *Sprint* berlangsung. Tujuannya ialah untuk mengevaluasi perkembangan pekerjaan, mengidentifikasi masalah yang muncul, serta mengatur pekerjaan agar tetap selaras dengan tujuan dari *Sprint* yang telah ditentukan. Disetiap *Daily Scrum* anggota tim akan menyampaikan laporan singkat mengenai pekerjaan yang telah mereka kerjakan, serta kendala atau hambatan yang sedang dihadapi. Dengan adanya *Daily Scrum* ini, tim dapat menyesuaikan pekerjaan secara cepat jika ditemukan kendala atau perubahan yang perlu diterapkan dalam *Sprint* yang sedang berlangsung. *Daily Scrum* juga berperan penting dalam meningkatkan kolaborasi antar anggota tim dengan memastikan transparansi dalam proses pengembangan sistem, serta membantu tim untuk tetap fokus pada tujuan *Sprint*. Pertemuan pada *Daily Scrum* hanya dilakukan secara singkat dan dilakukan pada waktu yang sama setiap harinya agar seluruh anggota tim dapat berpartisipasi dengan baik.

2.1.4 Sprint Review

Setelah tahap pengembangan sistem selesai ada tahap yang perlu dilakukan, yaitu *Sprint Review*. *Sprint Review* dilakukan untuk mengevaluasi hasil pekerjaan yang telah dicapai dari satu siklus *Sprint*. Dalam tahapan ini tim akan melakukan presentasi ke pemegang kepentingan mengenai fitur-fitur yang telah dikembangkan dalam satu siklus *Sprint* sebelumnya. Tujuan dari *Sprint Review* ini yaitu untuk mendapatkan *feedback* mengenai fitur yang telah dikembangkan. Jika terdapat saran perbaikan atau perubahan, tim dapat mencatat hal tersebut dan bisa dimasukkan kedalam *Sprint Planning* untuk *Sprint* berikutnya. Selain saran perbaikan, *Sprint Review* juga berfungsi untuk memastikan fitur yang dikembangkan sesuai dengan ekspektasi dari pengguna atau pemangku kepentingan. Dengan adanya tahap ini juga pengembangan sistem dapat berjalan lebih adaptif dengan kebutuhan yang mungkin berubah dari waktu ke waktu.

2.1.5 Sprint Retrospective

Tahap ini adalah tahap terakhir dari siklus *Scrum*. *Sprint Retrospective* memiliki tujuan untuk merefleksikan seluruh proses yang telah berjalan selama *Sprint* berlangsung. Di tahap ini tim pengembang akan melakukan evaluasi internal untuk mengetahui hal apa saja yang sudah berjalan dengan baik, tantangan dan kendala yang dihadapi, serta menentukan langkah-langkah yang efektif dan efisien pada *Sprint* berikutnya. Dalam *Sprint Retrospective*, evaluasi dilakukan baik dalam aspek teknis maupun non-teknis. Dalam aspek teknis mencakup metode coding, strategi testing, serta efektivitas alat yang digunakan dalam pengembangan proyek. Sementara dalam aspek non-teknis, meliputi komunikasi antar anggota tim, efisiensi dari kolaborasi, dan cara meningkatkan produktivitas kerja. Dengan ini, tim dapat terus melakukan perbaikan berkelanjutan sehingga *Sprint* berikutnya dapat berjalan dengan lebih baik.

3. HASIL DAN PEMBAHASAN

Pengembangan antarmuka sistem pemesanan tiket pesawat dilakukan dengan metode *Scrum* yang terdiri dari 3 siklus *Sprint* dengan durasi waktu dari masing-masing *Sprint* adalah 2 minggu

3.1 Sprint Planning

Dalam *Sprint Planning* pertama, diskusi dilakukan untuk menentukan rencana awal pengembangan proyek. Langkah pertama adalah menciptakan repository untuk menginisialisasi proyek menggunakan *React Vite*.

Selanjutnya, menyusun *Scrum Board* sebagai alat untuk mengatur tugas-tugas yang akan dilaksanakan dalam *Sprint* pertama. Kemudian, *Sprint Backlog* diatur untuk memecah antarmuka menjadi beberapa halaman utama, yaitu halaman beranda, halaman tiket, halaman biodata pelanggan, halaman pemesanan tiket, dan halaman pembayaran. Halaman-halaman tersebut akan di*Slicing* untuk diterapkan ke dalam web yang sedang dikembangkan.

Pada *Sprint Planning* yang kedua, perhatian utama terletak pada *slicing* antarmuka untuk halaman autentikasi yang mencakup halaman login dan halaman register. Selain itu *slicing* juga dilakukan untuk halaman histori order. Pada *Sprint 2*, *API* juga akan diintegrasikan untuk halaman yang sebelumnya sudah dibuat serta memastikan halaman beranda dapat diakses oleh pengguna dengan login maupun tanpa login terlebih dahulu.

Fokus utama dalam *Sprint Planning* ketiga adalah penambahan fitur-fitur baru seperti fitur untuk notifikasi dan fitur untuk unduh tiket. Disamping itu optimalisasi tampilan juga dilakukan agar website menjadi *responsive* diberbagai ukuran layar. Selain itu integrasi *API* dari keseluruhan fitur juga dikerjakan. Dalam tahap ini juga perbaikan dilakukan terhadap *bug* yang terdeteksi selama proses pengembangan untuk memastikan sistem dapat beroperasi dengan baik dan berfungsi secara optimal.

3.2 Sprint Backlog

Didalam *Sprint Backlog* berisi daftar dari berbagai fitur yang sudah dibahas dan disepakati pada tahap *Sprint Planning* sebelumnya. *Sprint Backlog* berisi berbagai macam tugas yang akan dilakukan pada *Sprint* yang sedang berlangsung. Berikut adalah daftar *Sprint Backlog* dari *Sprint* pertama hingga *Sprint* ketiga.

Tabel 1. Daftar Pekerjaan dalam *Sprint Backlog*

<i>Sprint</i>	Fitur Yang Dikembangkan
<i>Sprint</i> Ke-1	Inisialisasi proyek, <i>slicing</i> halaman beranda, <i>slicing</i> halaman tiket, <i>slicing</i> halaman biodata pelanggan, dan <i>slicing</i> halaman pemesanan tiket.
<i>Sprint</i> Ke-2	<i>Slicing</i> halaman login dan register, <i>slicing</i> halaman histori order, <i>slicing</i> halaman <i>reset password</i> , integrasi <i>API</i> pada halaman yang sudah dibuat sebelumnya, pastikan hanya halaman beranda yang bisa diakses tanpa login terlebih dahulu.
<i>Sprint</i> Ke-3	Penambahan fitur notifikasi, fitur unduh tiket, integrasi keseluruhan <i>API</i> , membuat website menjadi <i>responsive</i> di berbagai ukuran layar, dan melakukan <i>fixing bug</i> .

3.3 Daily Scrum

Dalam Pengembangan antarmuka sistem pemesanan pesawat ini, pertemuan diadakan empat kali dalam seminggu, yaitu pada hari senin, selasa, kamis, dan jum'at. Pertemuan berlangsung selama 3 jam, mulai dari pukul 19:00 sampai 22:00. Setiap pertemuan diadakan, tim akan saling memantau dan saling membantu jika ada yang mengalami hambatan atau masalah dalam pengerjaan proyek.

3.4 Sprint Review

Pada *Sprint Review* yang pertama, fitur yang sudah dibuat telah berjalan dengan baik. Implementasi *slicing* halaman sudah sesuai dengan *Sprint Backlog*. Untuk perbaikan pada *Sprint* berikutnya, disarankan untuk lebih berfokus ke pengembangan sistem autentikasi sehingga pengguna harus login terlebih dahulu sebelum melakukan pemesanan.

Saran perbaikan dari *Sprint Review* kedua adalah memperbaiki *bug* yang ditemukan, yaitu *bug* pada sistem autentikasi token *JWT*. Hal tersebut dapat membuat pengguna bisa mengakses halaman yang seharusnya membutuhkan proses login. Untuk mengatasinya disarankan menggunakan *JWT Verify*. Selain daripada itu *slicing* tampilan halaman yang dibuat sudah sesuai dengan standar *requirement* yang diminta.

Presentasi di *Sprint Review* ketiga berjalan dengan sangat baik dengan penyampaian fitur yang jelas dan terarah. Dari sisi teknis, website diuji menggunakan perangkat *mobile*. Dari hasil pengujiannya website terbukti *responsive* dan minim *bug* yang ditemukan. Penambahan animasi loading memberi pengalaman pengguna yang lebih baik. Fitur-fitur utama seperti pemesanan dan unduh tiket sudah berfungsi secara optimal. Selain itu sistem notifikasi sudah berjalan dengan baik, notifikasi yang diberikan akan tampil jika pengguna memesan serta melakukan pembelian tiket pada sistem.

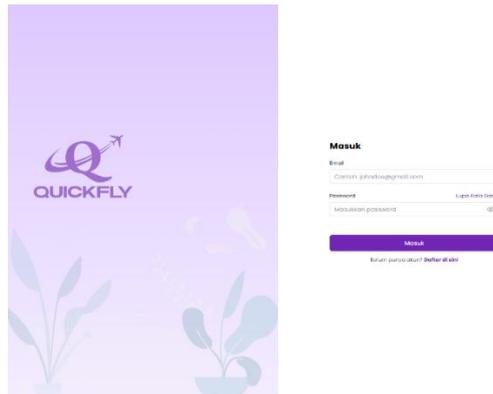
3.5 Sprint Retrospective

Berdasarkan hasil evaluasi dari diskusi *Sprint Retrospective*, terdapat beberapa aspek yang perlu ditingkatkan untuk menjaga efisiensi dan efektivitas kerja tim. Komunikasi antar anggota perlu diperkuat agar koordinasi lebih optimal. Selain hal tersebut, penting untuk membangun budaya saling membantu dalam menyelesaikan kendala

yang muncul selama masa pengembangan, baik dalam menghadapi kendala internal maupun kendala eksternal. Sesi *Brainstorming* juga perlu dimaksimalkan agar ide dan solusi yang dihasilkan lebih baik dan komprehensif. *Bonding* antar anggota tim juga harus lebih diperkuat guna menciptakan lingkungan kerja yang lebih kolaboratif. Terakhir, perlu adanya kebiasaan untuk menginformasikan secara berkala kepada tim terkait fitur yang sudah selesai dikembangkan sebelum *dipush* ke repository proyek.

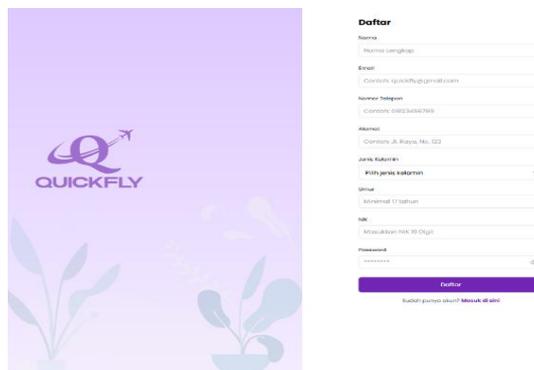
3.6 Tampilan Aplikasi

Jika pengguna ingin melakukan pemesanan tiket maka pengguna diharuskan untuk login terlebih dahulu. Pada halaman ini pengguna dapat memasukan email dan *password* untuk masuk kedalam website dari akun yang sudah dibuat sebelumnya.



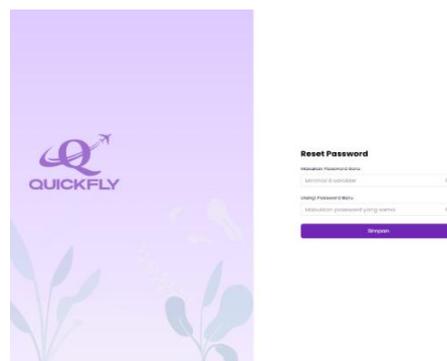
Gambar 2. Halaman Login

Jika pengguna belum memiliki akun, maka bisa melakukan registrasi terlebih dahulu dengan memasukan nama, email, nomor telepon, alamat, jenis kelamin, umur, NIK, dan *password* untuk akun yang akan dibuat.



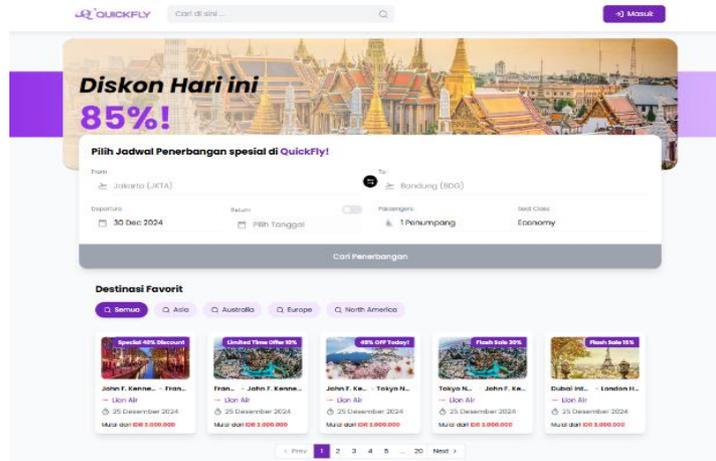
Gambar 3. Halaman Register

Adapun jika pengguna lupa *password* dari akun yang sudah dibuat, pengguna bisa melakukan *reset password* pada halaman *reset password*.



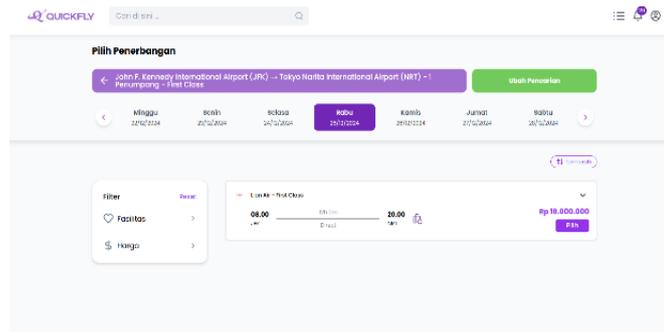
Gambar 4. Halaman Reset Password

Halaman utama dalam website ini adalah halaman beranda, halaman ini bisa diakses oleh pengguna baik dengan login maupun tanpa melakukan login. Pada halaman beranda terdapat list dari tujuan penerbangan yang bisa dipilih oleh pengguna. Selain itu pengguna juga bisa memasukkan *filter* pada jadwal penerbangannya.



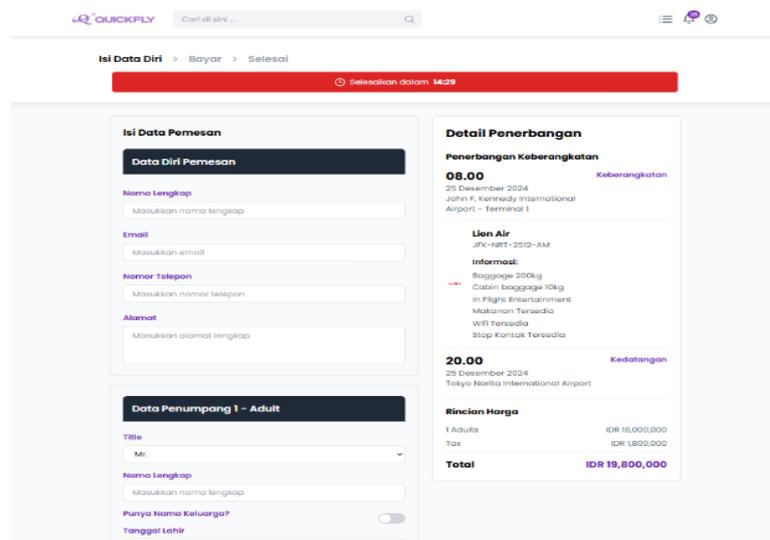
Gambar 5. Halaman Beranda

Pada halaman tiket pengguna bisa menggunakan *filter* untuk mencari tiket penerbangan yang diinginkan, fitur *filter* terdiri dari hari penerbangan, harga tiket, fasilitas, hingga lama dari penerbangan.



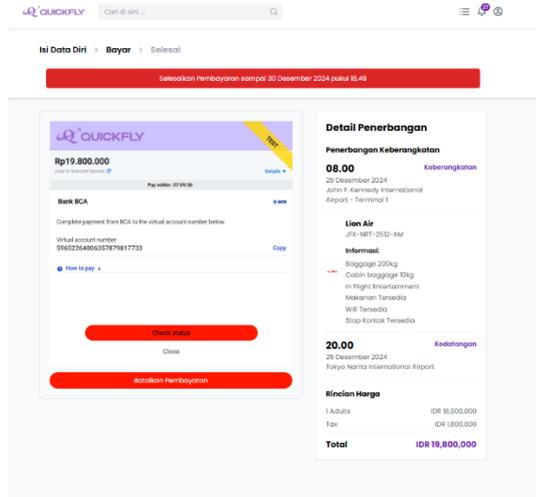
Gambar 6. Halaman Tiket

Setelah pengguna memilih tiket yang diinginkan, pengguna akan diarahkan ke halaman biodata, di halaman ini pengguna diminta untuk mengisi biodata pemesan serta penumpang dari tiket yang ingin dibeli.



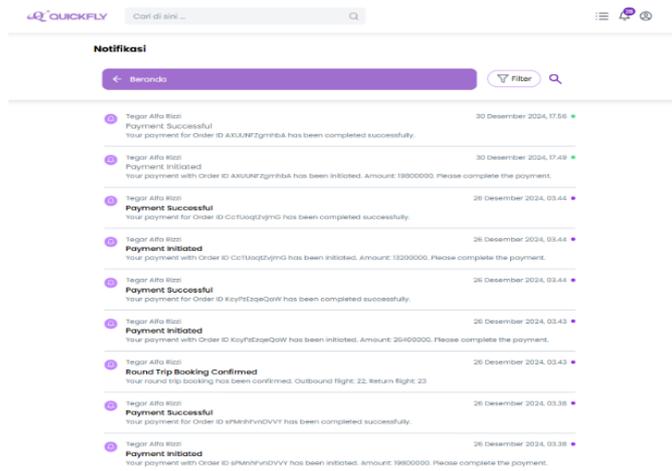
Gambar 7. Halaman Biodata Pengguna

Untuk melakukan pembayaran, pengguna akan diarahkan ke halaman pembayaran. Di halaman ini pengguna bisa memilih berbagai jenis metode pembayaran, mulai dari transfer bank hingga *e-wallet*.



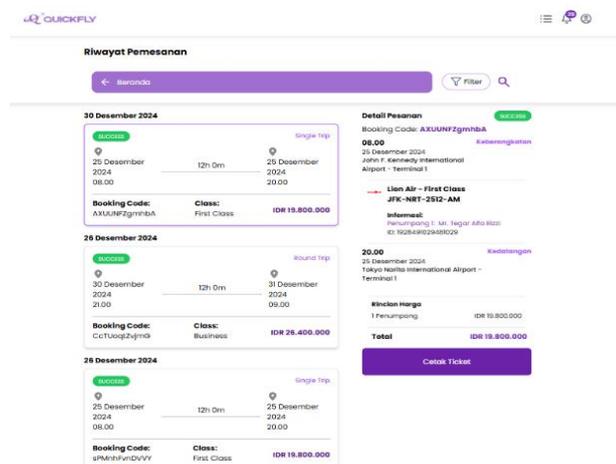
Gambar 8. Halaman Pembayaran

Notifikasi akan muncul setelah pengguna melakukan pemesanan dan pembayaran dari tiket yang sudah dibeli.



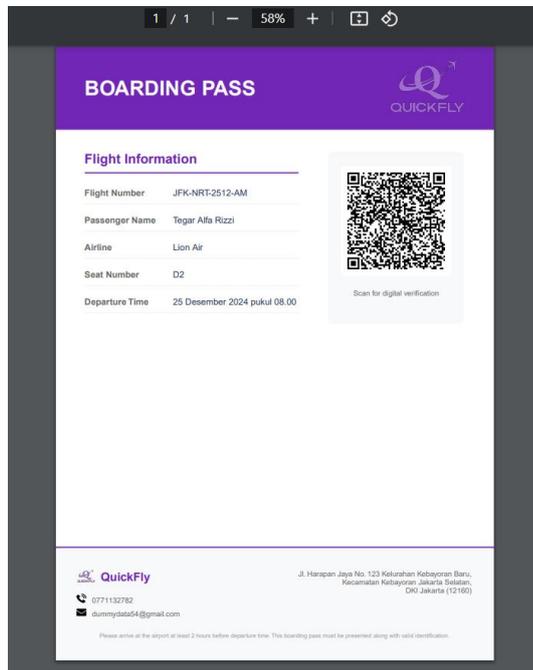
Gambar 9. Fitur Notifikasi

Pengguna bisa melihat tiket yang sudah dibeli pada halaman order histori, halaman ini menampilkan detail-detail dari tiket yang telah dibeli, pengguna juga bisa mengunduh tiket dengan menekan tombol cetak tiket.



Gambar 10. Halaman Order Histori

Hasil dari tiket yang sudah didownload berupa dokumen dengan format *PDF*, dari dokumen ini pengguna dapat menggunakan *qr code* yang terlampir untuk melakukan *boarding*. Hal ini dapat menghemat penggunaan kertas karena pengguna hanya perlu menunjukkan dokumen tiket tanpa perlu mencetaknya.



Gambar 11. Hasil Cetak Tiket Digital

4. KESIMPULAN

Penerapan metode *Scrum* dalam pengembangan antarmuka sistem informasi pemesanan tiket pesawat memungkinkan proses pengembangan menjadi lebih optimal dan efektif, hal ini dapat terjadi karena *Scrum* mendukung proses evaluasi dan *review* pada setiap akhir *Sprint*. Selain itu, komunikasi dan kolaborasi antar tim pengembang dapat menjadi lebih meningkat, sehingga proses pengembangan sistem antarmuka yang dibangun bisa menjadi lebih maksimal. Penggunaan *React Vite* juga dapat mempercepat dan menyederhanakan proses pengembangan. Dengan dukungan *Single Page Application* pada *React* memungkinkan proses navigasi pada website dapat lebih lancar tanpa perlu melakukan *reload* halaman, sehingga dapat mengurangi beban *server* serta dapat meningkatkan pengalaman pengguna. Dengan dukungan dari berbagai *library Javascript* fitur unduh tiket dapat dikembangkan dengan menggunakan *React PDF*, memungkinkan pengembang untuk melakukan *styling* pada dokumen. Kemudian untuk pengembangan lebih lanjut, disarankan untuk sistem yang dibuat dilengkapi dengan fitur login atau registrasi menggunakan *Google Oauth*, dengan adanya fitur ini proses autentifikasi pengguna dapat berjalan dengan lebih mudah, sehingga dapat meningkatkan kenyamanan serta kemudahan dalam mengakses sistem.

REFERENCES

- [1] Farhatun Nisaul Ahadiyah, "Perkembangan Teknologi Infomasi Terhadap Peningkatan Bisnis Online," *INTERDISIPLIN: Journal of Qualitative and Quantitative Research*, vol. 1, no. 1, hlm. 41–49, Des 2023, doi: 10.61166/interdisiplin.v1i1.5.
- [2] C. A. Cholik, "Perkembangan Teknologi Informasi Komunikasi / ICT Dalam Berbagai Bidang," 2021.
- [3] S. Hadijah dan F. Yunita, "Sistem Informasi Penjualan Tiket Boat Berbasis Web," 2020.
- [4] T. Putra Kurnianda dan H. Purwanto, "Analisis Kepuasan Penumpang Terhadap Pemesanan Tiket Melalui Tiket Online Dan Pemesanan Tiket Secara Offline Di Bandara XYZ," 2021.
- [5] I. Santosa *dkk.*, "Perancangan Dan Pengembangan Sistem Informasi Aplikasi Berbasis Website Pemesanan Tiket Konser Dengan Menerapkan Pemodelan Unified Modeling Language," 2024.
- [6] S. N. Atifah, A. Primajaya, dan D. Yusup, "Penerapan Single Page Application Pada Pengembangan Aplikasi E-Learning Nusabot," 2024.
- [7] A. I. Satria, L. Andrawina, dan H. D. Anggana, "Perancangan Sistem Informasi Manajemen Pemesanan Tiket Pesawat Pada Travel Agent Wiro Karya Berbasis Web Dengan Metode Waterfall Designing Information System of Aircraft Ticket Booking Management on the Wiro Agent Travel Web-Based Work With Waterfall Method," vol. 7, no. 1, hlm. 1890, 2020.

- [8] A. Y. Rahmadhani, I. Heildayana,) Sistem, dan I. Akuntansi, “Penerapan Metode Waterfall Pada Sistem Informasi Penjualan Tiket Pesawat Terbang,” 2021.
- [9] A. Guskani dan D. A. Muthia, “Perancangan Sistem Informasi Pemesanan Tiket Pesawat Domestik Berbasis Web Pada CV Jenika Group Depok,” 2022.
- [10] J. Margaretha dan A. Voutama, “Perancangan Sistem Informasi Pemesanan Tiket Konser Musik Berbasis Web Menggunakan Unified Modeling Language (UML),” *JOINS (Journal of Information System)*, vol. 8, no. 1, hlm. 20–31, Jun 2023, doi: 10.33633/joins.v8i1.7107.
- [11] S. Firmansyah *dkk.*, “Penerapan UML Dalam Sistem Pemesanan Tiket Bioskop Berbasis Website,” 2024.
- [12] A. Wirfs-Brock dan B. Eich, “JavaScript: The First 20 Years,” *Proceedings of the ACM on Programming Languages*, vol. 4, no. HOPL, Jun 2020, doi: 10.1145/3386327.
- [13] A. Khati Chhetri, “Developing a Front-end web app using React,” 2024.
- [14] W. Maj, “React-PDF,” npmjs. Diakses: 20 Maret 2025. [Daring]. Tersedia pada: <https://www.npmjs.com/package/react-pdf>
- [15] T. A. Nguyen, “A comparative analysis of Webpack and Vite as build tools for JavaScript.”
- [16] A. Andipradana dan K. D. Hartomo, “Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum,” 2021.
- [17] K. Schwaber dan J. Sutherland, “Panduan Scrum,” 2020.