

Implementasi Desain UI/UX pada Aplikasi Tekos Berbasis Website Menggunakan React JS

1st Ihsan Asfari Hanifan
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
ihsanasfari@students.telkomuniversity.ac.id

2nd Mira Kania Sabariah
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
mirakania@telkomuniversity.ac.id

3rd Imanuddin Hasbi
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
imanhasbi@students.telkomuniversity.ac.id

Abstrak - Kebutuhan akan tempat tinggal untuk mahasiswa baru dan mahasiswa yang sudah tidak menempati asrama di Telkom University dalam beberapa tahun terakhir mengalami peningkatan signifikan. Tekos merupakan startup baru berupa aplikasi *website* yang akan menjadi wadah untuk mahasiswa Telkom University untuk mencari tempat tinggal setelah asrama, mahasiswa bisa memperoleh kemudahan dan keterjaminan memiliki tempat tinggal pasca asrama. Sementara itu, UI/UX desainer telah merancang tampilan desain aplikasi/*website* untuk meningkatkan pengalaman pengguna, namun diperlukan peran front-end developer untuk menerjemahkan desain tersebut menjadi tampilan aplikasi *website* yang nyata dan dapat diakses oleh pengguna. Tujuan dari tugas akhir ini adalah mengimplementasikan *website* Tekos pada bagian front-end menggunakan React JS, sehingga pengguna dapat mengakses dan mengelola aplikasi *website* Tekos yang dikembangkan. Selain itu, tugas akhir ini juga bertujuan untuk meningkatkan performa aplikasi Tekos dengan mengoptimalkan kinerja front-end melalui penggunaan React JS yang efisien, memastikan kesesuaian antarmuka pengguna (UI) dengan desain yang sudah dirancang, dan menghadirkan aksesibilitas yang lebih baik pada aplikasi Tekos. Hasil utama dari eksperimen menunjukkan bahwa implementasi antarmuka dari desain UI/UX menggunakan React JS pada aplikasi Tekos telah mencapai hasil yang baik dalam hal performa dan aksesibilitas.

Kata kunci : website, tekos, front-end, react js, performa, aksesibilitas.

I. PENDAHULUAN

A. Latar Belakang

Telkom University merupakan sebuah perguruan tinggi swasta yang berlokasi di Bandung, Jawa Barat, Indonesia. Telkom University telah beberapa kali menduduki peringkat pertama Perguruan Tinggi Swasta (PTS) Terbaik di Indonesia serta masuk jajaran Perguruan Tinggi Terbaik di Indonesia. Telkom University memiliki 7289 mahasiswa baru pada tahun 2021 dan 7.868 mahasiswa baru pada tahun 2022 berdasarkan data dari Desy Dwi

Nurhandayani[1], [2]. Telkom University dituntut untuk dapat menyediakan fasilitas asrama atau tempat tinggal selama satu tahun pertama bagi mahasiswa baru di Telkom University. Terjadinya peningkatan mahasiswa baru yang signifikan pada tahun 2022 mengakibatkan adanya mahasiswa baru yang harus menempati satu ruangan untuk lima orang bahkan tidak mendapatkan fasilitas asrama dan juga kesulitan mencari kost terdekat di daerah Telkom University. Di sisi lain mahasiswa yang sudah tidak menempati asrama memiliki kesulitan untuk mencari tempat tinggal pasca asrama. UI/UX designer bertugas untuk membuat sebuah tampilan desain aplikasi/*website* sehingga dapat meningkatkan pengalaman dan kepuasan pengguna[3]. Namun untuk mewujudkan tampilan desain yang sudah dibuat, dibutuhkan seorang front-end developer yang bertugas menerjemahkannya ke dalam bentuk tampilan aplikasi *website* nyata, Sehingga pengguna dapat berinteraksi[4]. Tekos merupakan startup baru berupa aplikasi *website* yang akan menjadi wadah untuk mahasiswa Telkom University untuk mencari tempat tinggal setelah asrama, mahasiswa bisa memperoleh kemudahan dan keterjaminan memiliki tempat tinggal pasca asrama. Untuk dapat mengembangkan *website* Tekos yang ingin dibuat, Front-End merupakan salah satu bagian terpenting agar *website* dapat diakses dan dikelola oleh pengguna[4], [5].

B. Topik dan Batasannya

Topik dari tugas akhir ini adalah Implementasi desain UI/UX pada aplikasi Tekos berbasis *website* menggunakan React JS. Adapun batasan masalah dari tugas akhir ini adalah mengimplmentasikan tampilan website sistem layanan Tekos yang telah di desain oleh UI/UX desainer menggunakan library React JS. Selain itu memasikan kelayakan teknis website pada bagian *performace* dan *accessibility*.

C. Tujuan

Tugas akhir ini bertujuan untuk mengimplementasikan *website* Tekos pada bagian

front-end menggunakan React JS, sehingga pengguna dapat mengakses dan mengelola aplikasi *website* Tekos yang dikembangkan dengan desain yang sudah dirancang. Metodologi pengembangan scrum akan diterapkan untuk meningkatkan efisiensi dan transparansi dalam proses pengembangan aplikasi. Penelitian ini juga bertujuan untuk mengukur dan menganalisis hasil *performace testing* dan *accessibility testing* pada *website* Tekos, sehingga dapat memastikan kualitas dan keberhasilan proyek.

II. STUDI TERKAIT

A. HTML

HTML merupakan singkatan dari *HyperText Markup Language*, berfungsi untuk merancang/mengembangkan stuktur dasar halaman pada suatu *website*[6]. Kode HTML memastikan elemen teks dan gambar agar bisa ditampilkan di dalam browser.

B. CSS

CSS merupakan singkatan dari *Cascading Style Sheets*, berfungsi untuk merubah tampilan *website* seperti merubah jenis font, mengatur ukuran font, warna elemen, latar belakang halaman, dsb[7]. CSS dapat digunakan dengan tiga cara, yaitu Internal CSS, External CSS, dan Inline CSS.

C. Javascript

Javascript merupakan bahasa pemrograman yang dapat digunakan untuk beberapa fungsi, salah satunya berfungsi untuk pemrograman web[8]. Penggunaan Javascript untuk pengembangan web berfungsi untuk membuat web lebih interaktif[9].

D. React JS

ReactJs merupakan *library* Javascript, berfungsi untuk membangun tampilan antarmuka pada *website* [10]. ReactJs dapat mendesain tampilan sederhana untuk setiap level dalam aplikasi, sehingga dapat digunakan untuk mengembangkan aplikasi *website*. Salah satu keunggulan utama React JS adalah dukungannya terhadap fitur Single Page Application (SPA), yang memungkinkan pengalaman pengguna yang lebih responsif pada segi *performance* tanpa perlu memuat ulang seluruh halaman[11].

E. Tailwind

Tailwind merupakan sebuah *framework* CSS yang dirancang untuk mempermudah proses *styling* pada *website* atau aplikasi web[12]. *Framework* ini menyediakan kelas-kelas CSS yang dapat digunakan untuk mengubah tampilan dari suatu elemen HTML. Salah satu fitur unggulan dari Tailwind adalah adanya konsep "utility-first", yang berarti bahwa Tailwind menyediakan kelas-kelas CSS yang terfokus pada satu tugas tertentu, seperti menentukan warna *background* atau ukuran font[12].

F. Redux

Redux adalah sebuah *library open-source* yang digunakan untuk mengelola state atau data pada aplikasi web, khususnya pada aplikasi berbasis React [13], [14]. Redux membantu pengembang dalam mengorganisasi dan mengelola state secara terpusat, sehingga memudahkan dalam pengelolaan data yang kompleks dan memungkinkan berbagai komponen dalam aplikasi untuk mengakses data dengan mudah. Redux bekerja dengan mengikuti tiga prinsip dasar: Store, Action, dan Reducer[15].

G. API

API (*Application Programming Interface*) merupakan sebuah antarmuka yang digunakan untuk menghubungkan aplikasi atau perangkat lunak dengan sistem lain. API memungkinkan aplikasi untuk berkomunikasi dengan sistem atau layanan lain untuk bertukar informasi dan menjalankan tugas tertentu[16].

H. Lighthouse

Lighthouse merupakan sebuah *tools open source* yang dikembangkan oleh Google untuk menguji performa, aksesibilitas, dan kualitas *website*[17]. Lighthouse dapat digunakan pada *browser* Chrome atau dengan menggunakan *command line interface* (CLI) pada terminal. Lighthouse melakukan audit pada *website* dan menghasilkan laporan yang menunjukkan performa, aksesibilitas, dan kualitas *website*. Laporan tersebut menggunakan warna dan skor untuk memudahkan pengguna memahaminya. Lighthouse juga memberikan rekomendasi perbaikan untuk masalah yang ditemukan.

I. Performance Testing

Performance testing merupakan jenis pengujian perangkat lunak yang bertujuan untuk mengevaluasi performa suatu aplikasi atau sistem dalam berbagai kondisi dan skenario penggunaan yang berbeda. Tujuan utama dari *performance testing* adalah untuk mengukur kemampuan suatu aplikasi atau sistem dalam menangani beban dan *volume* data yang tinggi dalam situasi yang terus berubah[18].

J. Accessibility Testing

Accessibility testing merupakan jenis pengujian perangkat lunak yang bertujuan untuk mengevaluasi sejauh mana suatu aplikasi atau sistem dapat diakses dan digunakan oleh pengguna[19].

K. Scrum

Scrum merupakan metode untuk mengelola dan mengembangkan produk dengan pendekatan agile[20]. Scrum digunakan untuk memfasilitasi tim dalam melakukan kerja kolaboratif, transparan, dan adaptif dalam menghadapi perubahan yang terjadi dalam pengembangan produk[21].

L. Trello

Trello merupakan sebuah alat manajemen proyek dengan *kanban board* untuk mengorganisasi suatu alur kerja proyek[22]. Dengan menggunakan Trello pengguna dapat membuat daftar status tugas, serta suatu tugas dalam bentuk kartu[23]. Kartu-kartu tugas tersebut dapat dipindahkan secara fleksibel dari satu daftar ke daftar lain.

M. SKPL

Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan suatu dokumen yang menguraikan secara terperinci semua kebutuhan fungsional dan non-fungsional yang harus dipenuhi oleh sebuah perangkat lunak yang akan dikembangkan [6]. Dokumen SKPL menjadi landasan bagi tim pengembang dalam merancang, mengimplementasikan, dan menguji perangkat lunak sehingga sesuai dengan harapan dan kebutuhan pengguna serta lingkungan kerja yang diinginkan[24].

N. Front-end Developer

Front-end merupakan sebuah peran bidang dalam pengembangan perangkat lunak yang bertanggung jawab untuk merancang dan mengimplementasikan tampilan antarmuka pengguna pada suatu aplikasi[25]. Tugas utamanya meliputi merubah desain UI/UX yang telah dirancang oleh desainer menjadi kode-kode yang dapat dijalankan menjadi sebuah tampilan website.

O. UI/UX

UI/UX (User Interface/User Experience) merupakan aspek-aspek desain dan pengalaman pengguna dalam pengembangan perangkat lunak. UI berkaitan dengan desain visual dan tata letak elemen-elemen antarmuka pengguna, yang bertujuan menciptakan antarmuka intuitif untuk memudahkan pengguna dalam berinteraksi dengan aplikasi. Sementara UX, berfokus pada pengalaman keseluruhan pengguna saat berinteraksi dengan aplikasi[26].

III. SISTEM YANG DIBANGUN

A. Scrum

Dalam melakukan pengembangan aplikasi Tekos berbasis *website* menggunakan React js, scrum digunakan sebagai metodologi pengembangan perangkat lunaknya. Scrum dipilih karena fleksibilitas yang memungkinkan tim Tekos untuk mengimplementasikan proyek Tekos sesuai dengan kebutuhan backlog dan ataupun jika ada kebutuhan perubahan dan perbaikan. Pada proyek ini tim tekos menggunakan Trello sebagai alat manajemen proyek Kanban board. Pada metode pengembangan ini terdiri dari beberapa elemen, seperti Product Backlog, dan Sprint Planning.

Product backlog merupakan daftar dari seluruh pekerjaan yang perlu dilakukan dalam proyek pengembangan perangkat lunak Tekos. Daftar ini mencakup pekerjaan yang ditentukan berdasarkan diskusi kelompok. Dalam proyek Tekos ini, product backlog dibuat dan dikelola oleh tim pengembang Tekos.

Sprint planning dilakukan untuk merencanakan apa yang perlu dikerjakan pada sprint berikutnya dengan memilih item dari Product backlog yang akan diambil ke dalam sprint selanjutnya. Sprint planning ini dilakukan online maupun offline bersama tim Tekos. Berikut merupakan jadwal sprint dari tim Tekos pada TABEL 1.

TABEL 1.
Jadwal Sprint Planning Tekos

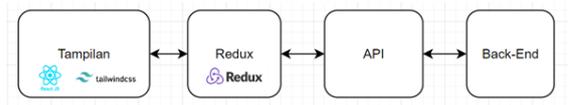
Sprint	Direncanakan Berakhir Pada Bulan
1	Februari
2	Maret
3	April
4	Mei
5	Juni
6	Juli
7	Agustus

Sprint review merupakan pertemuan yang dilakukan tim Tekos dengan pihak owner Tekos untuk mempresentasikan hasil dan progres dari sprint yang telah selesai. Tim Tekos mempresentasikan demo aplikasi yang dikembangkan dan mendapatkan masukan untuk perbaikan lebih lanjut.

Sprint retrospective merupakan pertemuan yang dilakukan secara online maupun offline oleh tim pengembang Tekos di akhir sprint untuk mengkaji sprint yang telah berlalu. Tim Tekos membahas apa yang berjalan baik dan apa yang perlu ditingkatkan untuk sprint berikutnya. Diskusi ini memastikan apa yang akan dikerjakan selanjutnya dan perbaikan dari proses yang telah dilalui.

B. Desain Sistem

Dalam perancangan aplikasi Tekos pada bagian Front-End, terdapat beberapa komponen Front-End yang digambarkan pada GAMBAR 1.



GAMBAR 1.
Desain Sistem

Pada bagian Implementasi, tugas akhir ini, berfokus untuk mengimplementasikan desain UI/UX yang telah dirancang oleh UI/UX desainer pada aplikasi Tekos. Implementasi ini dilakukan untuk memastikan bahwa desain UI/UX yang sudah dibuat UI/UX desainer telah dirancang dan dapat di

implementasikan dengan baik dalam bentuk aplikasi *website*.

Dalam bagian implementasi ini, pada bagian tampilan dilakukan pembuatan komponen-komponen UI/UX pada aplikasi *website* menggunakan ReactJs dan juga framework CSS Tailwind. Setiap Komponen UI/UX yang sudah dirancang diubah kedalam bentuk kodingan agar dapat diimplementasikan pada aplikasi *website* Tekos. Dengan menggunakan ReactJs dan framework CSS Tailwind, kode pada aplikasi *website* dapat dibuat secara fleksibel, sehingga mengembangkan aplikasi *website* dapat disesuaikan dengan kebutuhan dan tampilan yang diinginkan. Tailwind menyediakan kelas-kelas CSS yang dapat disesuaikan dengan kebutuhan, sementara React JS memungkinkan pengembangan *website* dengan menggunakan konsep komponen, sehingga aplikasi *website* dapat diatur secara modular dan dapat digunakan kembali.

Pada bagian selanjutnya, dilakukan integrasi API dengan menggunakan Redux untuk menyambungkan data dari bagian Back-End Tekos dengan tampilan aplikasi *website* Tekos. Dengan pengimplementasian ini, Redux dapat digunakan untuk menerima dan mengirim permintaan ke server Back-end melalui API dan menyimpan hasilnya kedalam state Redux. Sehingga Redux dapat digunakan untuk mengelola data dari API pada aplikasi *website* dengan lebih interaktif dan dinamis.

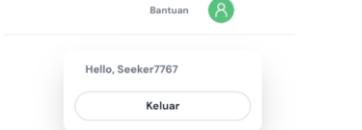
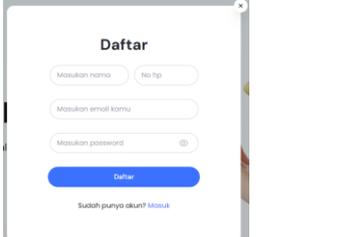
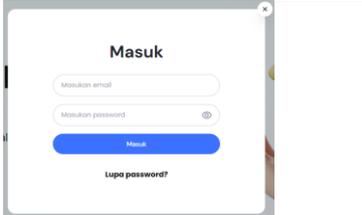
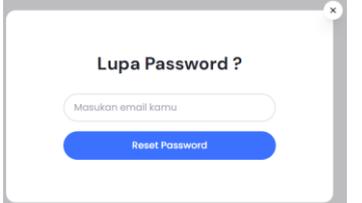
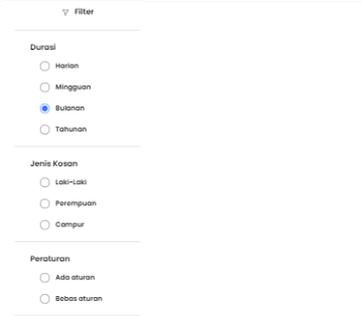
IV. HASIL DAN PEMBAHASAN

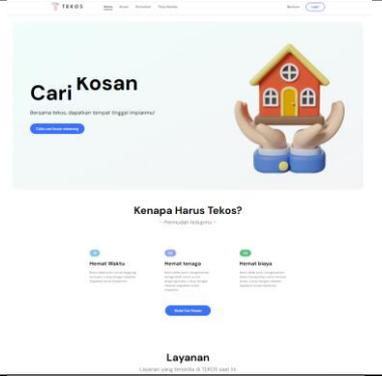
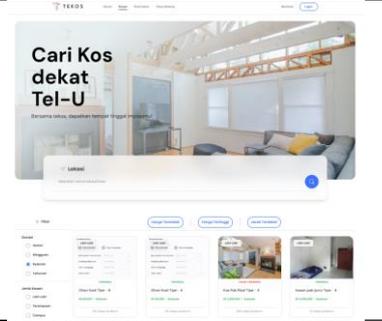
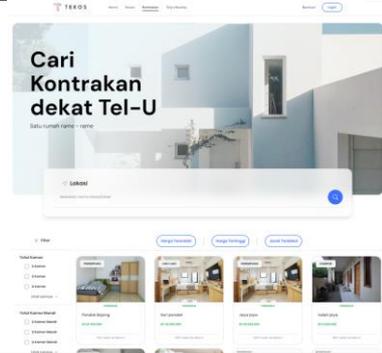
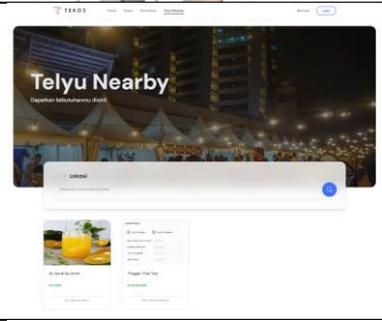
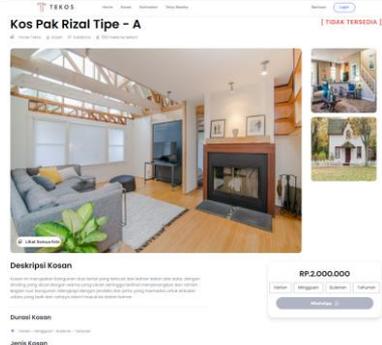
A. Hasil Implementasi Antarmuka

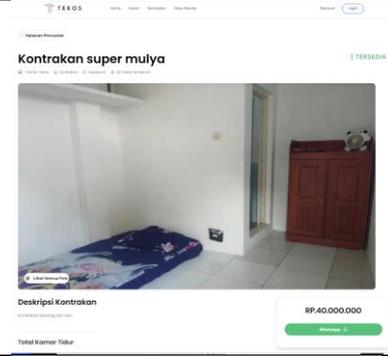
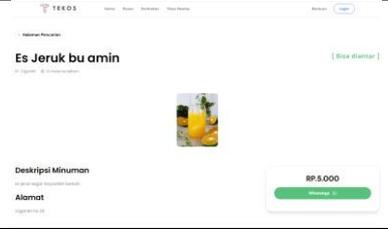
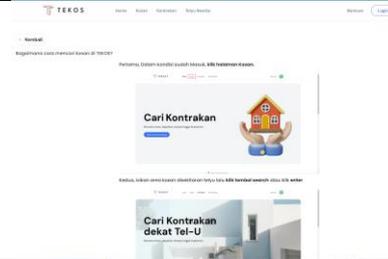
Pada bagian hasil implementasi antarmuka, akan ditampilkan tangkapan layar dari visualisasi komponen-komponen dan halaman-halaman Tekos, yang diimplementasikan sesuai dengan kebutuhan fungsional untuk pencari pada SKPL dan desain antarmuka UI/UX pada figma. Hasil tampilan yang diimplementasikan tersebut sudah di integrasi dengan api dari back-end, dan responsif untuk tampilan *website desktop* dan *mobile*. Berikut merupakan tangkapan layar visualisasi antarmuka Tekos pada TABEL 2.

TABEL 2.
Hasil Implementasi Antarmuka

Nama Antarmuka Beserta Kode Functional Requirement Pencari Kosan pada SKPL (Jika ada)	Tangkapan layar
Navbar	

Tombol Keluar (FR1-03)	
Modal Daftar (FR1-01)	
Modal Masuk	
Modal Bantuan	
Modal Lupa Password (FR1-04)	
Modal Notifikasi	
Kolom Pencarian (FR1-05, FR1-06, FR1-07)	
Sidebar Filter (FR1-08, FR1-09)	
Sort Button (FR1-20)	

Halaman Beranda	
Halaman Pencari Kosan (FR1-10)	
Halaman Pencarian Kontrakan (FR1-11)	
Halaman Pencarian Telyu Nearby (FR1-12)	
Halaman Detail Kosan (FR1-13, FR1-18, FR1-16, FR1-17, FR1-21)	

Halaman Detail Kontrakan (FR1-14, FR1-18, FR1-16, FR1-17)	
Halaman Detail Telyu Nearby (FR1-15, FR1-19, FR1-17)	
Halaman Bantuan	
Halaman Not Found	

B. Hasil Pengujian Performa

Pada bagian pengujian performa, alat Lighthouse digunakan sebagai alat pengujian. Pengujian dilakukan menggunakan server web lokal dengan mengkompilasi file menggunakan perintah “npm run build” untuk menghasilkan versi produksi dari aplikasi Tekos dengan file yang lebih dioptimalkan untuk tahap produksi. Berikut merupakan skor kecepatan memuat halaman aplikasi Tekos dari hasil pengujian performa yang digambarkan pada TABEL 3.

TABEL 3.
Hasil Skor Kecepatan Memuat Halaman

Halaman	Kecepatan memuat halaman (detik)
Halaman Beranda	0,5
Halaman Pencarian Kosan	0,4
Halaman Pencarian Kontrakan	0,5
Halaman Pencarian Telyu Nearby	0,4

Halaman Detail Kosan	0,4
Halaman Detail Kontrakan	0,5
Halaman Detail Telyu Nearby	0,4
Halaman Bantuan Kosan	0,5
Halaman Bantuan Kontrakan	0,5
Halaman Bantuan Telyu Nearby	0,4
Halaman Bantuan Autentikasi	0,4

C. Hasil Pengujian Aksesibilitas

Accessibility testing dilakukan per halaman untuk memastikan bahwa setiap halaman pada aplikasi Tekos dapat memenuhi standar *accessibility* dan dapat diakses dengan baik oleh pengguna dari berbagai latar belakang dan kemampuan. Berikut merupakan hasil skor pengujian aksesibilitas menggunakan alat Lighthouse dari aplikasi Tekos

TABEL 4 digambarkan pada

TABEL 4.
Hasil Skor Aksesibilitas Lighthouse

Halaman	Skor Hasil Pengujian Aksesibilitas dari Lighthouse
Halaman Beranda	97
Halaman Pencarian Kosan	97
Halaman Pencarian Kontrakan	97
Halaman Pencarian Telyu Nearby	97
Halaman Detail Kosan	97
Halaman Detail Kontrakan	97
Halaman Detail Telyu Nearby	97
Halaman Bantuan Kosan	97
Halaman Bantuan Kontrakan	97
Halaman Bantuan Telyu Nearby	97
Halaman Bantuan Autentikasi	97

D. Analisis Hasil

Melalui proses pengembangan aplikasi Tekos yang dilakukan dengan menggunakan metodologi scrum, tugas akhir ini berhasil mengimplementasikan desain UI/UX pada sisi front-end aplikasi Tekos berbasis *website* menggunakan React JS. Berdasarkan hasil implementasi yang ditampilkan,

antarmuka *website* yang dikembangkan berhasil memenuhi semua kebutuhan fungsional sisi pencari dalam SKPL.

Setelah mengimplementasikan desain UI/UX aplikasi Tekos pada sisi front-end, dilakukan pengujian performa dan aksesibilitas menggunakan alat Lighthouse untuk memastikan kelayakan aplikasi Tekos.

Setelah dilakukannya Pengujian performa dengan menggunakan alat Lighthouse. Berdasarkan TABEL 3 kecepatan memuat dari seluruh halaman aplikasi Tekos yang diimplementasikan, angka-angka tersebut sudah melebihi ekspektasi dari NFR-01 yang menyatakan, perangkat lunak harus memberikan respons permintaan penggunaan dalam waktu kurang dari 5 detik. Performa yang cukup baik dalam pengujian tersebut sebagian besar disebabkan oleh pemanfaatan fitur Single Page Application (SPA) yang disediakan oleh React JS.

Pada pengujian aksesibilitas menggunakan alat

Lighthouse, sebagaimana digambarkan pada

TABEL 4 didapatkan hasil skor sebesar 97 dari setiap halamannya. Angka tersebut termasuk pada kategori skor kinerja “bagus” didalam rentang 90-100. Sehingga dapat dinilai sudah sesuai dengan NFR-05 yang menyatakan, antarmuka yang diimplementasikan telah memenuhi ekspektasi faktor-faktor seperti navigasi yang jelas, penempatan elemen yang logis, serta instruksi yang mudah dipahami.

Hasil pengujian *performance testing* dan *accessibility testing* pada aplikasi Tekos menunjukkan implementasi antarmuka dari desain UI/UX menggunakan React JS memberikan pengalaman responsif dan memenuhi kebutuhan non-fungsional NFR-01 dan NFR-05 pada SKPL. Meskipun demikian, ditemukan beberapa masalah dalam kedua pengujian. Dalam *performance testing*, terdeteksi masalah *Cumulative Layout Shift* (CLS) sebesar 0,212 pada halaman detail kosan dan kontrakan. Sedangkan dalam *accessibility testing*, terdapat masalah kontras warna pada elemen-elemen seperti navbar, footer, dan elemen pada halaman tertentu yang dapat ditingkatkan lagi.

Penelitian ini telah berhasil diatasi dengan melakukan implementasi tampilan aplikasi *website* Tekos, dengan memenuhi semua kebutuhan fungsional sisi pencari dan kebutuhan non fungsional (NFR-01 dan NRF-05) yang telah diuraikan dalam SKPL.

V. KESIMPULAN

Melalui pengembangan aplikasi Tekos dengan metodologi scrum, tugas akhir ini berhasil mengimplementasikan desain UI/UX pada sisi front-end menggunakan React JS. Keberhasilan dalam mengimplementasikan tampilan aplikasi sudah sesuai dengan kebutuhan fungsional sisi pencari dan

kebutuhan non fungsional (NFR-01 dan NFR-05) yang telah diuraikan dalam SKPL.

Sebagaimana dinyatakan pada NFR-01, perangkat lunak harus memberikan respons permintaan penggunaan dalam waktu kurang dari 5 detik. Dari hasil antarmuka Tekos yang diimplementasikan kedalam bentuk *website*, waktu memuat pada seluruh halaman sudah berhasil melebihi ekspektasi tersebut. Selanjutnya pada pengujian aksesibilitas menggunakan alat Lighthouse, menghasilkan skor pengujian sebesar 97 pada setiap halaman. Nilai 97 tersebut menunjukkan antarmuka yang diimplementasikan telah memenuhi ekspektasi faktor-faktor seperti navigasi yang jelas, penempatan elemen yang logis, serta instruksi yang mudah dipahami, sebagaimana dinyatakan dalam NFR-05. Nilai angka yang didapatkan termasuk kedalam kategori “bagus” didalam rentang 90-100 pada alat Lighthouse.

Secara keseluruhan dapat disimpulkan bahwa permasalahan pada penelitian ini, telah menghasilkan produk yang sesuai. Namun demikian, dalam proses analisis masih terdapat beberapa kelemahan, seperti masalah *Cumulative Layout Shift* (CLS) pada halaman detail kosan dan detail kontrakan, serta masalah kontras warna pada beberapa elemen. Meskipun demikian, penemuan masalah ini menunjukkan pentingnya melakukan perbaikan dan peningkatan lebih lanjut untuk memastikan bahwa aplikasi Tekos mencapai tingkat kualitas dan aksesibilitas yang lebih optimal bagi pengguna.

REFERENSI

- [1] Telkom University, “Pembukaan PKKMB 2022 Untuk Mahasiswa Baru Telkom University,” Sep. 13, 2022.
- [2] Telkom University, “Telkom University Kukuhkan 7.554 Mahasiswa Baru,” Sep. 17, 2021.
- [3] D. K. Safitri and A. Andrianingsih, “Analisis UI/UX untuk Perancangan Ulang Front-End Web Smart-SITA dengan Metode UCD dan UEQ,” *Techno.Com*, vol. 21, no. 1, 2022, doi: 10.33633/tc.v21i1.5639.
- [4] F. Hasbi, “Pengembangan Front-End Aplikasi UIIPortofolio Berbasis Web,” *Universitas Islam Indonesia*, 2018.
- [5] A. Babaei, G. Locatelli, and T. Sainati, “What is wrong with the front-end of infrastructure megaprojects and how to fix it: A systematic literature review,” *Project Leadership and Society*, vol. 2, 2021. doi: 10.1016/j.plas.2021.100032.
- [6] K. D. Pertiwi and Y. Kurniawan, “SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK SISTEM INFORMASI AKADEMIK UNIVERSITAS MA CHUNG MALANG,” *Simetris : Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 7, no. 2, 2016, doi: 10.24176/simet.v7i2.775.
- [7] Koesheryatin and Suryana Taryana, “Pengertian CSS,” *Aplikasi Internet Menggunakan HTML, CSS, dan JavaScript*. 2014.
- [8] K. Peguero and X. Cheng, “CSRF protection in JavaScript frameworks and the security of JavaScript applications,” *High-Confidence Computing*, vol. 1, no. 2, 2021, doi: 10.1016/j.hcc.2021.100035.
- [9] H. Reactor, “What is JavaScript Used For ? What is JavaScript ? What is JavaScript used for?,” *Hackreactor*, 2021.
- [10] P. S. Maratkar and P. Adkar, “React JS - An Emerging Frontend JavaScript Library,” *IRE Journals*, vol. 4, no. 12, 2021.
- [11] M. F. Santoso, “TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP,” *Jurnal Khatulistiwa Informatika*, vol. 9, no. 2, 2021, doi: 10.31294/jki.v9i2.11357.
- [12] Yogiokto Agae, “Mengenal Tailwind CSS,” *medium.com*, 2020.
- [13] Khuat and Tung, “Developing a frontend application using ReactJS and Redux,” *Laurea University of Applied Sciences*, 2018.
- [14] A. Banks and E. Porcello, *Functional Web Development with React and Redux*, vol. 7, no. 2. 2017.
- [15] A. Freeman, *Pro React 16*. 2019. doi: 10.1007/978-1-4842-4451-7.
- [16] M. Benmoussa, “API ‘Application Programming Interface’ Banking: A Promising Future for Financial Institutions (International Experience),” *Revue des Sciences Commerciales*, vol. 18, no. 2, 2019.
- [17] Google Developers, “Google Lighthouse,” *Google Developers*, 2017.
- [18] M. S. Sharmila and E. Ramadevi, “Analysis of Performance Testing on Web Applications,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, 2014.
- [19] M. Campoverde-Molina, S. Lujan-Mora, and L. Valverde, “Process Model for Continuous Testing of Web Accessibility,” *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3116100.
- [20] A. A. F. Amarta and I. G. Anugrah, “Implementasi Agile Scrum Dengan Menggunakan Trello Sebagai Manajemen Proyek Di PT Andromedia,” *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, vol. 4, no. 6, 2021, doi: 10.32672/jnkti.v4i6.3702.

- [21] C. Rawis, S. D. S. Karouw, and Sherwin R. U. A. Sompie, "Software Requirement Specification Academic Information System of Sam Ratulangi University," *Jurnal Teknik Elektro dan Komputer*, vol. 10, no. 2, 2021.
- [22] M. H. A. Johnson, "Trello," *Journal of the Medical Library Association*, vol. 105, no. 2, Apr. 2017, doi: 10.5195/JMLA.2016.49.
- [23] T. Minchew, "Who's on first?: License team workflow tracking with trello," *Serials Review*, vol. 41, no. 3, 2015, doi: 10.1080/00987913.2015.1065946.
- [24] P. G. Manek and D. Siahaan, "NOISE DETECTION IN SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT USING SPECTRAL CLUSTERING," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 17, no. 1, 2019, doi: 10.12962/j24068535.v17i1.a771.
- [25] S. Mufti Prasetyo, M. Ivan Prayogi Nugroho, R. Lima Putri, and O. Fauzi, "Pembahasan Mengenai Front-End Web Developer dalam Ruang Lingkup Web Development," *Jurnal Multidisiplin Ilmu*, vol. 1, no. 6, 2022.
- [26] R. Gutama and T. Dirgahayu, "Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP)," *Informatics Departement Universitas Islam Indonesia*, vol. Vol 2, 2021.