

RANCANG BANGUN APLIKASI MARKETPLACE PENATU MOBILE WEB BERBASIS RESPONSIVE WEB DAN FRAMEWORK EXPRESS.JS

Rossario Catherine Elfrida¹, Juan Farel Purba²,
Made Sudarma³, I Wayan Shandyasa⁴

^{1,2}Mahasiswa Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

^{3,4}Dosen Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana
Jl. Raya Kampus Unud Jimbaran, Kec. Kuta Sel, Kabupaten Badung, Bali 80361

rossariocatherine@student.unud.ac.id¹, juanfarelpurba@student.unud.ac.id²,

sudarmaee@unud.ac.id³, shandyasa@unud.ac.id⁴

ABSTRAK

Penelitian ini mengembangkan aplikasi *mobile web marketplace* penatu untuk mengatasi kesulitan pengguna layanan penatu, seperti menemukan lokasi terdekat, mengelola nota fisik, dan mendapatkan informasi layanan. Dengan menggunakan *responsive web design* dan teknik flexbox, *front-end* aplikasi ini dirancang agar optimal di berbagai perangkat. *Back-end* aplikasi dibangun menggunakan Express.js dengan konsep *controller-middleware-route* untuk menjaga kode tetap terorganisir. Selain itu, *Geocoding API* dan *Distance Matrix API* diimplementasikan untuk menghitung jarak tempuh antara pelanggan dan penatu. Pengujian dilakukan dengan *black box testing* oleh pengembang dan *usability test* oleh partisipan. Hasil penelitian menunjukkan bahwa *front-end* aplikasi memberikan pengalaman pengguna optimal dengan tingkat kepuasan 89,57%, dan *back-end* aplikasi efektif dalam mengelola data serta integrasi *Geocoding API* dan *Distance Matrix API* berhasil dengan tingkat kepuasan 89,22%. Penelitian ini membuktikan bahwa aplikasi *mobile web* responsif ini efektif dan efisien dalam membantu pengguna menemukan layanan penatu yang sesuai, meningkatkan kenyamanan dan efisiensi layanan.

Kata kunci : *Marketplace Penatu, Responsive Web Design, Black Box Testing, Usability Test*

ABSTRACT

This research developed a mobile web marketplace application for laundry services to address user difficulties such as finding the nearest location, managing physical receipts, and obtaining service information. Utilizing a responsive web design approach and flexbox techniques, the front end of the application was designed to be optimal across various devices. The back end was built using Express.js with the controller-middleware-route concept to maintain code organization. Furthermore, the Geocoding API and Distance Matrix API were implemented to calculate the distance between customers and laundry services. Testing was conducted using black box testing by developers and usability testing by participants. The results indicated that the front end of the application provided an optimal user experience with an overall satisfaction rate of 89.57%, and the back end was effective in managing data with middleware features and the integration of the Geocoding API and Distance Matrix API, achieving a usability satisfaction rate of 89.22%. This research demonstrates that the responsive mobile web application is effective and efficient in assisting users in finding suitable laundry services, thereby enhancing convenience and service efficiency.

Key Words : *Laundry Marketplace, Responsive Web Design, Black Box, Usability Test*

1. PENDAHULUAN

Gaya hidup *modern* yang semakin menekankan praktis dan efisiensi telah mengubah cara mencuci pakaian dari metode tradisional ke penggunaan mesin cuci. Namun, tidak semua orang memiliki akses ke mesin cuci atau waktu untuk mencuci secara manual. Dalam konteks ini, jasa penatu yang menyediakan layanan mencuci dan menyetrica pakaian menjadi solusi praktis. Meski jumlah jasa penatu terus meningkat dan banyak yang mengadopsi teknologi seperti mesin cuci otomatis dan aplikasi komunikasi, banyak pengguna masih menghadapi kesulitan dalam menemukan layanan yang sesuai dengan kebutuhan mereka, seperti lokasi, informasi harga, dan jenis layanan.

Untuk mengatasi masalah ini, aplikasi web responsif yang memanfaatkan teknik flexbox dan *framework* Express.js dapat menawarkan solusi yang lebih efisien. Aplikasi semacam ini dapat memberikan pengalaman pengguna yang optimal di berbagai perangkat[3] dan memudahkan akses informasi serta pengelolaan layanan penatu.

Penelitian ini berfokus pada "Rancang Bangun Aplikasi *Mobile Web* untuk *Marketplace* Penatu Berbasis *Responsive Web* dengan Teknik Flexbox Menggunakan *Framework* Express.js" untuk meningkatkan kenyamanan dan efisiensi dalam layanan penatu. Tujuan dari penelitian ini adalah merancang *front-end* menggunakan teknik *flexbox* agar aplikasi dapat diakses melalui *web browser* dalam berbagai ukuran layar perangkat seperti desktop, tablet, dan *smartphone*. Kemudian membangun *back-end* menggunakan *framework express.js* dalam memproses dan mengelola data dan informasi. Lalu dapat mengimplementasikan fitur yang memberikan informasi mengenai lokasi layanan penatu beserta estimasi jarak tempuh untuk menuju ke lokasi tersebut. Dengan demikian, diharapkan aplikasi ini dapat memudahkan calon pengguna jasa dalam mencari penatu sesuai kebutuhan dan memudahkan pelaku usaha dalam memasarkan layanannya.

2. KAJIAN PUSTAKA

2.1 Tinjauan Mutakhir

Penelitian terbaru menunjukkan berbagai upaya untuk meningkatkan akses dan efisiensi aplikasi layanan *laundry* melalui teknologi web dan *mobile*. Budiman dan Yudianthi menggunakan *Progressive Web App* (PWA) dan *ServiceWorker* untuk pelayanan *laundry* sepatu[1]. Penelitian penulis menambahkan implementasi *Google Maps API* untuk pencarian lokasi penatu terdekat.

Febriyan Saragih dkk. membuat sistem *tracking laundry* sepatu berbasis web, sejalan dengan penelitian penulis dalam pemanfaatan teknologi web[2]. Regmi membahas pembuatan situs web responsif menggunakan *Flexible* CSS grid dan *Flexbox*[3], mirip dengan pendekatan desain penulis.

Penelitian penulis menonjol dengan integrasi *Google Maps API* untuk memudahkan pengguna menemukan lokasi penatu terdekat secara efisien dan responsif.

2.2 Penatu

Penatu menyediakan layanan cuci dan setrika kiloan, *dryclean*, dan *express*, membantu orang yang memiliki keterbatasan waktu atau fasilitas mencuci di rumah[4]. Pelanggan sering mengalami kesulitan menemukan penatu yang sesuai, kehilangan nota fisik, dan keterbatasan informasi layanan. Beberapa penatu telah mengadopsi teknologi *modern* seperti Whatsapp untuk pemesanan, tetapi upaya lebih lanjut diperlukan untuk meningkatkan efisiensi dan kenyamanan pelanggan.

2.3 Marketplace

Marketplace adalah sistem pasar elektronik yang memfasilitasi transaksi dan interaksi bisnis antara pembeli dan penjual secara virtual, sehingga kredibilitas sangat penting dalam platform ini[5]. *Marketplace* bertindak sebagai perantara yang membantu membangun hubungan dan kolaborasi bisnis secara virtual.

2.4 Aplikasi Web-Mobile

Platform *Mobile Web* memungkinkan aplikasi berjalan di semua perangkat *mobile* dengan standar yang sama seperti desktop, namun memerlukan desain responsif untuk mengakomodasi berbagai ukuran layar dan

resolusi. Kecepatan dan keterbatasan konektivitas juga harus dipertimbangkan. Keuntungan utama adalah pengguna hanya memerlukan koneksi internet untuk mengakses aplikasi[6].

2.5 Express.js

Express adalah sebuah *framework* web minimalis dan fleksibel untuk Node.js yang memberikan serangkaian fitur yang kuat untuk aplikasi web dan *mobile*. *Framework* ini menawarkan kinerja yang optimal dengan lapisan tipis fitur dasar aplikasi web, tanpa mengaburkan fitur-fitur utama Node.js yang sudah dikenal dan digunakan dengan baik. Express juga merupakan kerangka routing yang ringan dan fleksibel dengan fitur inti minimal yang dapat diperluas melalui modul-modul *middleware* express [7].

2.6 Black Box Testing

Pengujian *black box* memfokuskan pada spesifikasi fungsional perangkat lunak tanpa melihat desain atau kode program, bertujuan memastikan fungsi, masukan, dan keluaran sesuai spesifikasi. Kasus uji mencakup skenario benar dan salah, dilakukan pada tahap akhir untuk memastikan perangkat lunak berfungsi dengan baik [8].

2.7 Usability Test

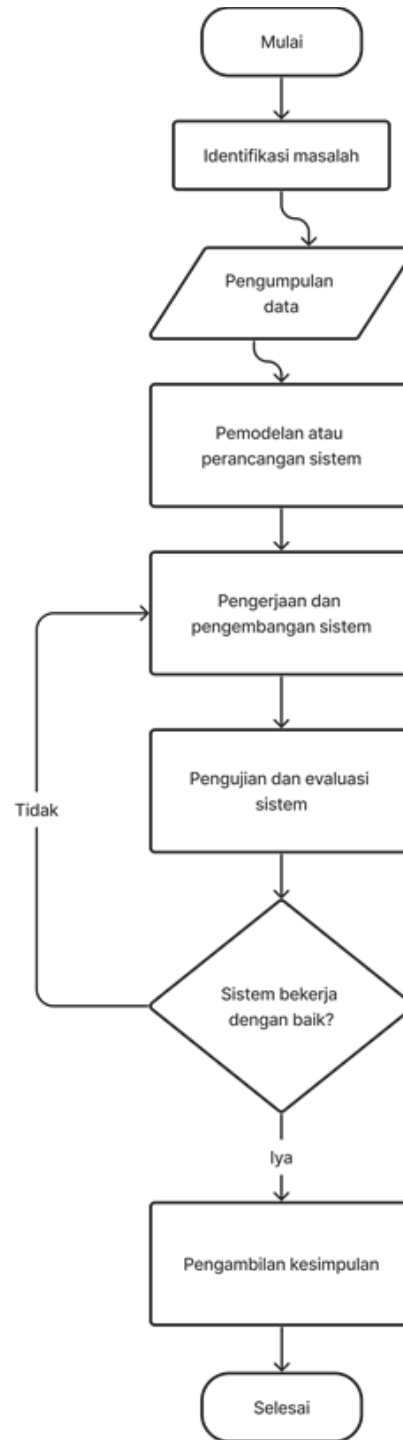
Usability test adalah metode evaluasi yang digunakan untuk mengukur efektivitas aplikasi dari perspektif pengguna, dengan fokus pada seberapa mudah aplikasi digunakan dan seberapa memuaskan pengalaman pengguna. Untuk menilai aspek *usability*, kepuasan, dan kemudahan penggunaan, digunakan kuesioner USE (*Usability, Satisfaction, and Ease of Use*) yang mengukur tiga dimensi utama dari pengalaman pengguna: kegunaan, kepuasan, dan kemudahan penggunaan[9].

3. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian mengenai *marketplace* penatu dilaksanakan di Program Studi Teknik Elektro Universitas Udayana yang terletak di Kampus Bukit Jimbaran.

Penelitiannya dimulai dari bulan Juli sampai Desember 2023. Alur pada tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Tahapan Penelitian

Berikut penjelasan dari Gambar 1.

Langkah 1. Identifikasi masalah

Penelitian ini dimulai dengan mengidentifikasi masalah yang terkait

dengan topik penelitian dan merumuskan masalah yang akan diteliti. Rumusan masalah pada penilitan ini adalah bagaimana cara membangun aplikasi marketplace penatu berbasis web untuk mencari lokasi penatu terdekat dan memperoleh informasi mengenai layanan penatu yang sesuai kebutuhan

Langkah 2. Pengumpulan Data

Pada tahapan yang kedua ini, peneliti mengumpulkan sumber-sumber seperti jurnal, artikel, maupun buku yang relevan dengan masalah penelitian. Referensi tersebut digunakan untuk mendukung penelitin dan memperdalam pemahaman terhadap topik yang dibahas, seorti penggunaan *framework* express.js hingga flexbox untuk implementasi aplikasi *marketplace* penatu yang responsif.

Langkah 3. Perancangan Sistem

Pada tahapan ini diambil keputusan untuk menggunakan metode tertentu dalam pengembangan aplikasi *marketplace* penatu. Aplikasi *marketplace* penatu akan dibangun menggunakan *framework* Express.js untuk *back-end* dan metode *Responsive Web* menggunakan teknik flexbox untuk *front-end*.

Langkah 4. Pengembangan Sistem

Dalam pembuatan aplikasi *marketplace* penatu ini, bahasa pemrograman javascript digunakan dengan menggunakan *framework* Express.js serta metode *Responsive Web*

Langkah 5. Pengujian Sistem

Penelitian ini menggunakan *black box testing* serta *usability test* menggunakan kuesioner untuk menguji kelayakan sistem.

Langkah 6. Pengambilan Kesimpulan

Tahapan ini melibatkan analisis dan penarikan kesimpulan berdasarkan hasil pengujian dan evaluasi sistem yang telah dilakukan.

3.2 Analisis Kebutuhan Pengguna

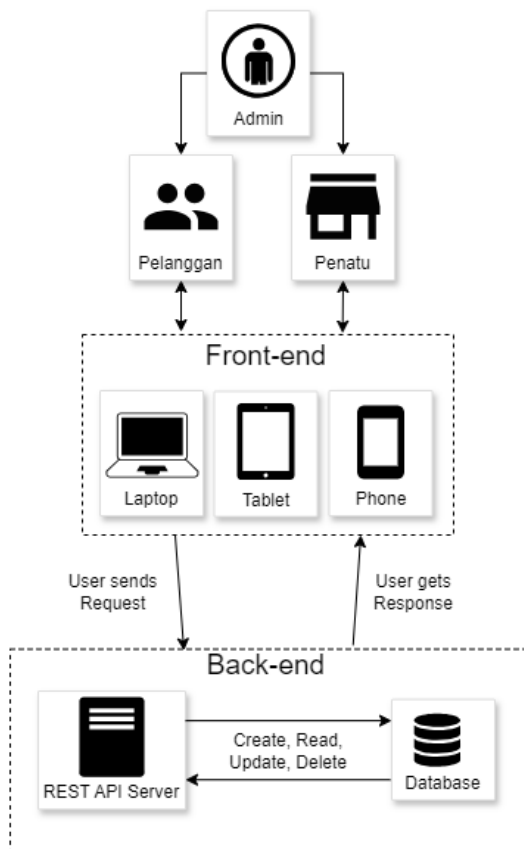
Analisis kebutuhan pengguna dilakukan untuk memastikan bahwa aplikasi dapat memenuhi harapan dan kebutuhan setiap jenis penggunanya. Dalam aplikasi ini, pengguna dibagi menjadi tiga peran utama, yaitu admin, penatu, dan pelanggan.

Tabel 1. Analisis Kebutuhan Pengguna

No.	Pengguna	Kebutuhan
1.	Admin	1. Login 2. Kelola pengguna
2.	Penatu	1. Registrasi 2. Login 3. Kelola profil 4. Mengelola daftar layanan yang tersedia 5. Mengelola status transaksi 6. Mengunduh rekap penjualan
3.	Pelanggan	1. Registrasi 2. Login 3. Kelola profil 4. Mencari penatu berdasarkan jarak terdekat dan harga termurah 5. Melihat dan membuat transaksi 6. Melihat dan/atau memperbarui status transaksi

3.3 Gambaran Umum Sistem

Aplikasi yang dikembangkan merupakan platform berbasis *website* yang dirancang khusus untuk pasar penatu. Aplikasi ini memiliki desain responsif, sehingga dapat diakses dengan mudah melalui berbagai perangkat, baik desktop, tablet, maupun *smartphone*. Dengan fitur ini, pengguna dapat mengakses aplikasi dengan nyaman dan efisien, terlepas dari perangkat yang digunakan, karena aplikasi akan menyesuaikan proporsi komponen dari *website* dengan dimensi perangkat yang digunakan oleh pengguna.

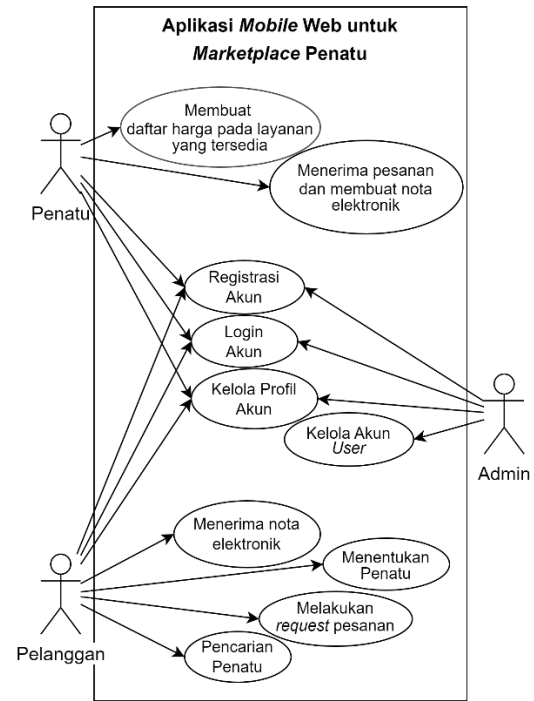


Gambar 2. Gambaran Umum Sistem

3.4 Use Case Diagram

Use Case Diagram menggambarkan bagaimana sistem berinteraksi dengan berbagai aktor dari luar. Dalam diagram ini, terdapat tiga aktor utama. Pertama, admin bertanggung jawab atas pengelolaan data pengguna, termasuk melakukan perubahan dan penghapusan data. Kemudian penatu, memiliki kemampuan untuk menentukan harga layanan, membuat nota elektronik, serta melihat riwayat nota yang telah diterbitkan. Sementara itu, pelanggan dapat mencari dan menentukan penatu sesuai kebutuhannya, melakukan pemesanan, menerima nota elektronik dan meninjau riwayat nota yang telah mereka terima. Sebelum mengakses sistem, setiap aktor

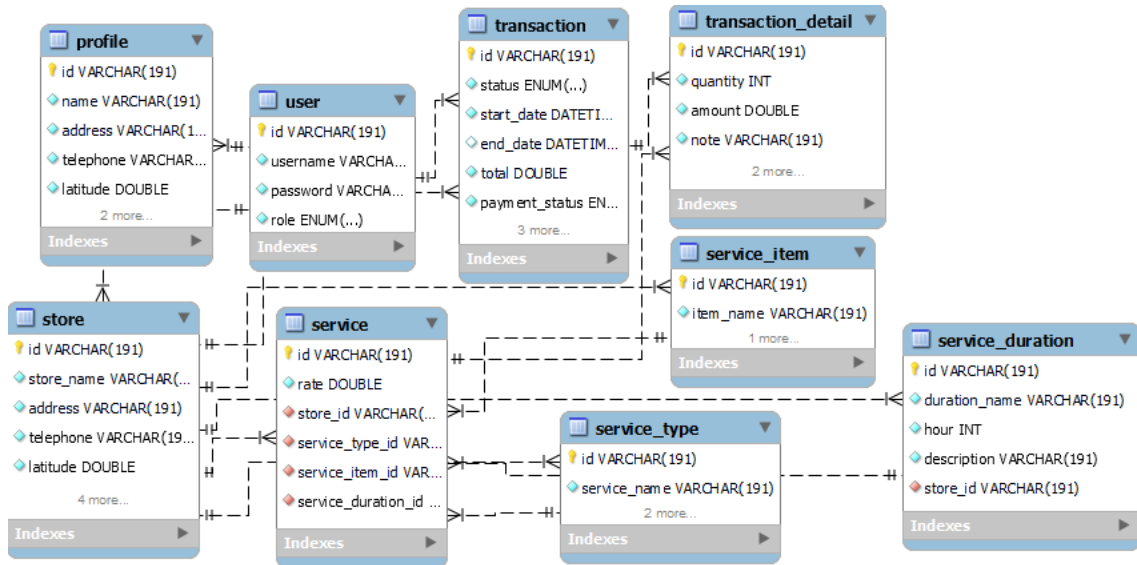
harus melakukan *login* terlebih dahulu. Jika aktor tidak dapat *login* ke aplikasi, maka aktor harus mendaftar akun terlebih dahulu. Setiap aktor juga diberikan hak untuk mengelola profil akunya sendiri. Namun, aktor admin memiliki wewenang untuk mengelola profil akun dari seluruh aktor. Use case diagram yang menggambarkan skenario penggunaan aplikasi dapat dilihat pada Gambar 3.



Gambar 3. Use Case Diagram

3.5 Rancangan Basis Data

Pada aplikasi *marketplace* penatu, terdapat sembilan tabel yang saling berelasi untuk mendukung kebutuhan berbagai pengguna secara keseluruhan. Relasi antar tabel ini dirancang untuk mengoptimalkan pengelolaan data, memastikan integritas informasi, dan memfasilitasi interaksi yang efisien antara entitas seperti pengguna, layanan, dan transaksi. Skema *database* yang menggambarkan hubungan antar tabel dapat dilihat pada Gambar 4.

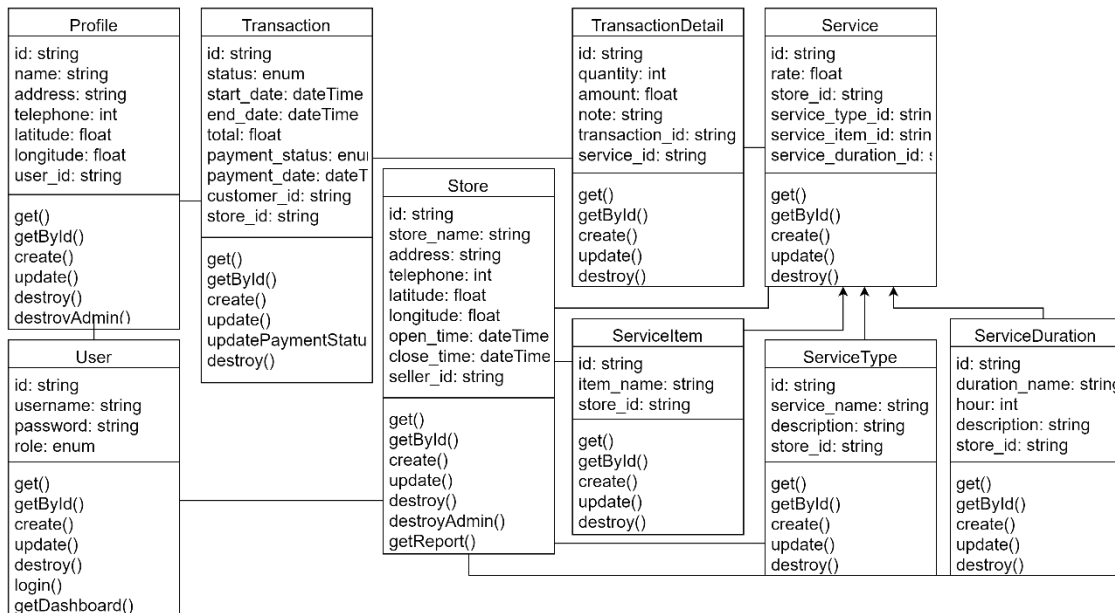


Gambar 4. Skema Database

3.6 Class Diagram

Pada penelitian ini terdapat berbagai entitas yang berinteraksi satu sama lain untuk menyediakan layanan yang lengkap dan responsif bagi pengguna. Diagram kelas ini memvisualisasikan struktur dan

hubungan antara entitas-entitas utama dalam sistem. Diagram kelas yang dapat dilihat pada Gambar 5. mencakup entitas seperti *User*, *Profile*, *Store*, *ServiceType*, *ServiceItem*, *ServiceDuration*, *Service*, *Transaction*, dan *TransactionDetail*.



Gambar 5. Class Diagram

3.7 Perancangan REST API

Dasar implementasi API yang digunakan dalam penelitian ini menggunakan metode REST. Dalam penelitian ini, perancangan REST API mencakup total 48 endpoint yang telah

dirancang untuk masing-masing fungsi dari entitas yang ada. Tabel 2. menampilkan daftar endpoint dan metode HTTP yang digunakan.

Tabel 2. Endpoint API

Endpoint	Method
----------	--------

/api/v1/auth/login	POST
/api/v1/users	GET, POST
/api/v1/users/:id	GET, PUT, DELETE
/api/v1/profiles	GET, POST, PUT, DELETE
/api/v1/profiles/:id	GET
/api/v1/profiles/:username	DELETE
/api/v1/stores/:storeid/services	GET
/api/v1/stores/:storeid/services/:id	GET
/api/v1/stores	GET, POST, PUT, DELETE
/api/v1/stores/:id	GET
/api/v1/stores/:username	DELETE
/api/v1/services	GET, POST
/api/v1/services/:id	GET, PUT, DELETE
/api/v1/servicedurations	GET, POST
/api/v1/servicedurations/:id	GET, PUT, DELETE
/api/v1/serviceitems	GET, POST
/api/v1/serviceitems/:id	GET, PUT, DELETE
/api/v1/servicetypes	GET, POST
/api/v1/servicetypes/:id	GET, PUT, DELETE
/api/v1/transactions	GET
/api/v1/transactions/:id	GET, PUT, PATCH, DELETE
/api/v1/stores/:store_id/transactions	POST
/api/v1/report	GET
/api/v1/dashboard	GET

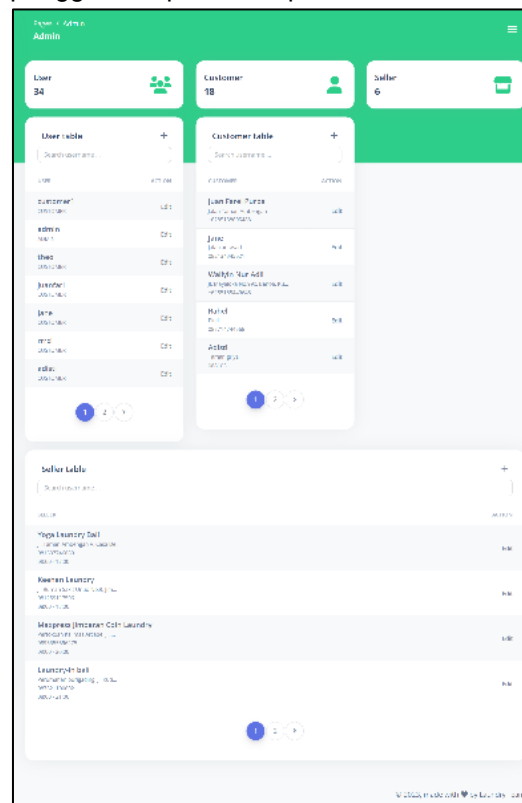
4. HASIL DAN PEMBAHASAN

4.1 Hasil Antarmuka Aplikasi Admin

Pada tahap ini akan menjelaskan fitur aplikasi *marketplace* penatu *role* admin.

1. Kelola Pengguna

Kelola pengguna merupakan fitur utama pada *role* admin. Pada halaman ini admin dapat mengawasi jumlah pengguna pelanggan dan penatu yang sudah terdaftar di aplikasi beserta data yang telah terdaftar diaplikasi. Tidak sampai disitu, admin juga dapat menambahkan mengedit, serta menghapus pengguna yang terdaftar pada aplikasi *marketplace* penatu. Kelola pengguna dapat dilihat pada Gambar 6.



Gambar 6. Kelola Pengguna

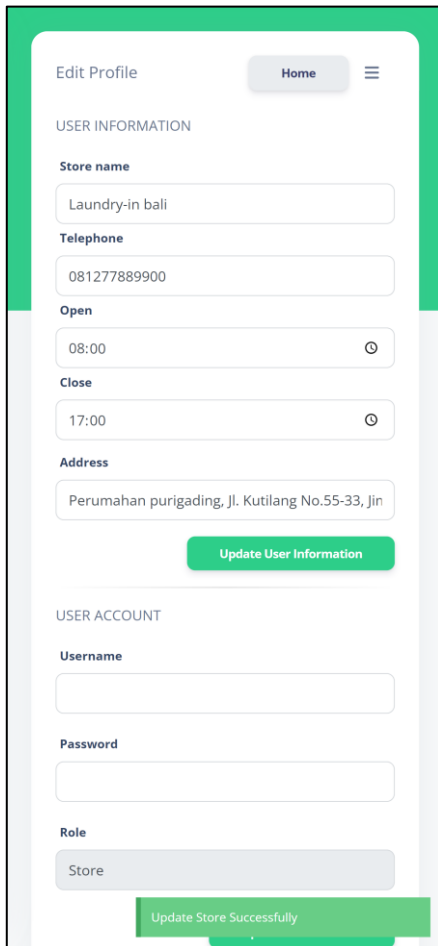
4.2 Hasil Antarmuka Penatu

Berikut adalah penjelasan fitur aplikasi *marketplace* penatu dengan *role* penatu.

1. Kelola Profil

Halaman profil penatu yang dapat dilihat pada Gambar 7. berisikan informasi yaitu nama, nomor telepon, waktu operasional toko, hingga alamat penatu. Informasi tersebut juga dapat diperbarui dengan menekan tombol *Update User Information*. Selain informasi toko, pada

halaman ini juga tersedia tombol untuk melakukan penggantian *username* maupun *password*.



Gambar 7. Halaman Kelola Profil Penatu

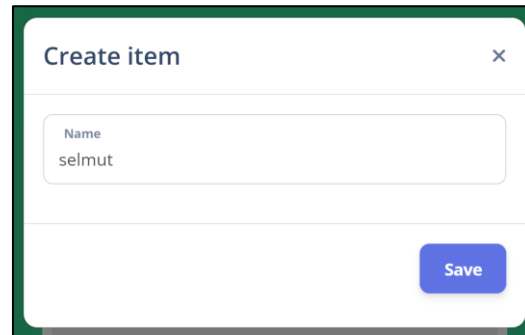
2. Kelola Layanan

Pada halaman kelola layanan, penatu melakukan manajemen layanan yang terdiri dari 3 kriteria yaitu *item* layanan, tipe layanan dan durasi layanan. Penatu dapat melakukan CRUD (*create, read, update, delete*) pada daftar *item* layanan, tipe layanan dan durasi layanan.

Pada Gambar 8. adalah tampilan dari daftar *item* layanan. Untuk membuat satu *item* layanan, pengguna harus mengisi informasi dari nama *item*, dapat dilihat pada Gambar 9.



Gambar 8. Daftar *Item*

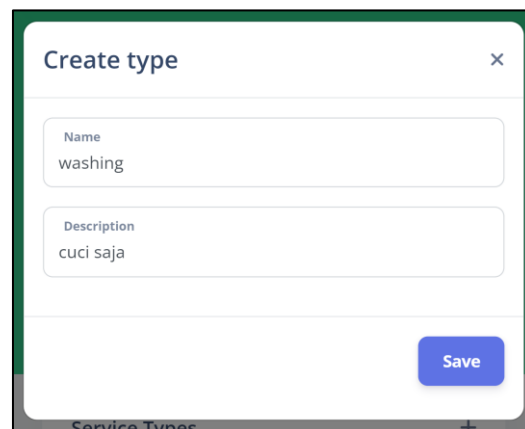


Gambar 9. Halaman *Create Item*

Pada Gambar 10. adalah tampilan dari daftar tipe layanan. Untuk membuat satu tipe layanan, pengguna harus mengisi informasi dari nama dan deskripsi tipe, dapat dilihat pada Gambar 11.



Gambar 10. Daftar Tipe

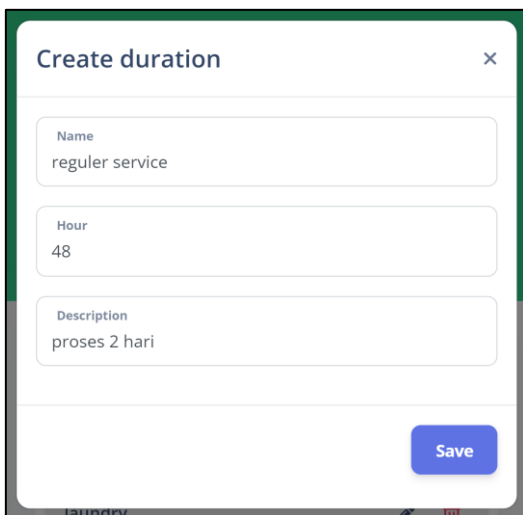


Gambar 11. Halaman *Create Tipe*

Pada Gambar 12. merupakan tampilan dari daftar durasi layanan pada penatu. Untuk membuat satu durasi layanan, pengguna harus mengisi informasi dari nama durasi, dan waktu durasi, dapat dilihat pada Gambar 13.



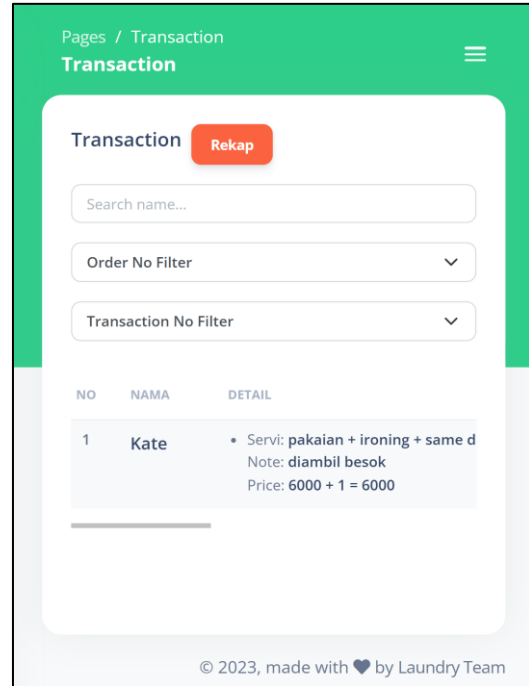
Gambar 12. Daftar Durasi



Gambar 13. Halaman Create Durasi

3. Kelola Transaksi

Pada halaman transaksi akan memuat semua pesanan yang masuk dari pihak pelanggan beserta status pesanannya. Halaman ini juga menyediakan tombol-tombol *assist* berupa pencarian nama hingga *filter*. Adapun tombol berwarna jingga untuk mengunduh hasil rekap transaksi yang akan memudahkan pengguna seperti pada Gambar 14.



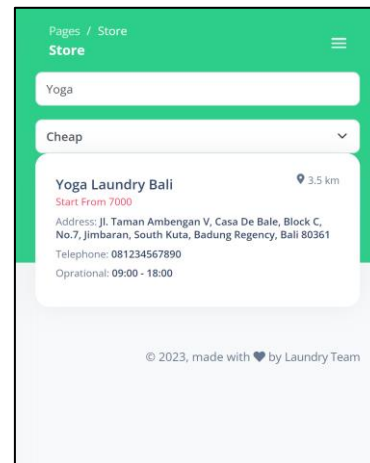
Gambar 14. Halaman Transaksi Penatu

4.3 Hasil Antarmuka Pelanggan

Pada tahap ini akan menjelaskan fitur aplikasi *marketplace* penatu *role* pelanggan.

1. Fitur Pencarian Penatu

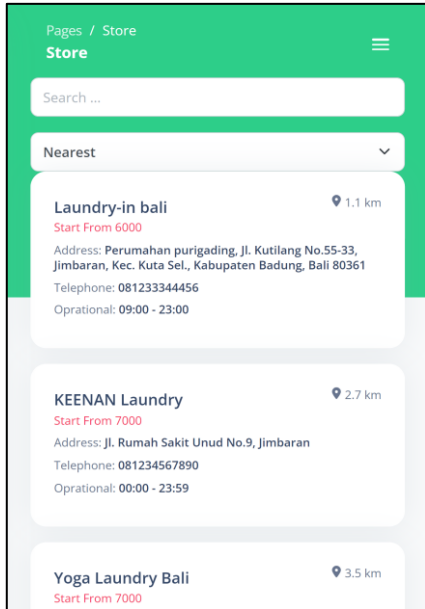
Pada fitur ini, pelanggan dapat mencari nama penatu memasukkan kata kunci pada kolom pencarian. Dapat dilihat pada Gambar 15.



Gambar 15. Pencarian Penatu

2. Fitur *Filtering* Jarak

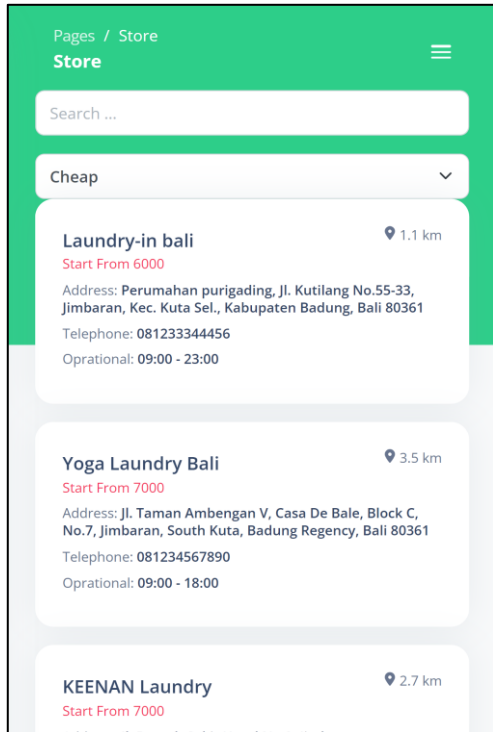
Fitur ini dapat mempermudah pelanggan untuk menemukan penatu berdasarkan jarak terdekat dari lokasi terkini pengguna. Seperti pada Gambar 16.



Gambar 16. Filtering Jarak Terdekat

3. Fitur *Filtering* Harga Termurah

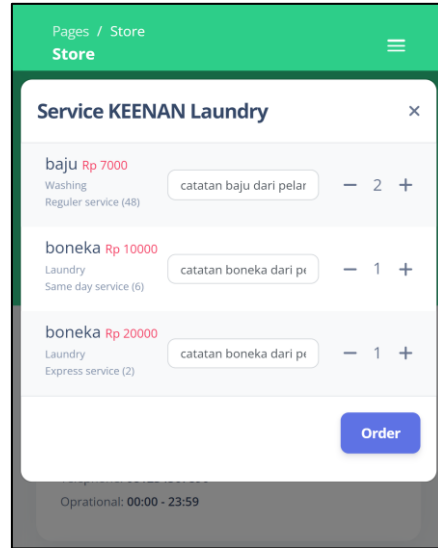
Fitur ini dirancang untuk menampilkan penatu berdasarkan harga layanan termurah. Dapat dilihat pada Gambar 17.



Gambar 17. Filtering Harga Termurah

4. Fitur Pemesanan

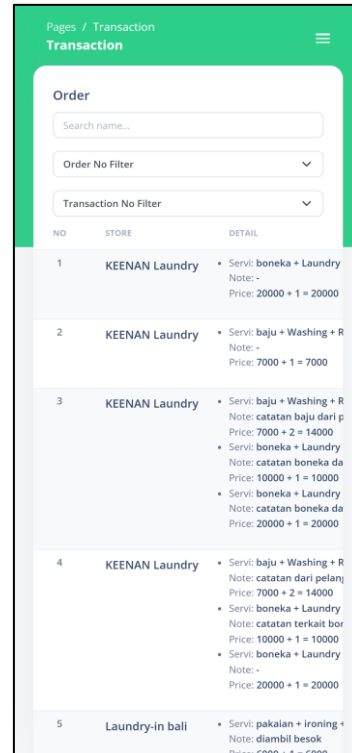
Melakukan pemesanan merupakan fitur tak terpisahkan dalam sebuah aplikasi *marketplace*. Halaman pemesanan dapat dilihat pada Gambar 18.



Gambar 18. Halaman Pemesanan

5. Melihat Riwayat Transaksi

Pada halaman ini, pelanggan dapat melihat riwayat transaksi yang sedang berjalan maupun yang telah selesai. Disamping itu untuk transaksi yang sedang berjalan, pelanggan juga dapat melihat status proses pengerjaan pemesanan mereka. Dapat dilihat pada Gambar 19.



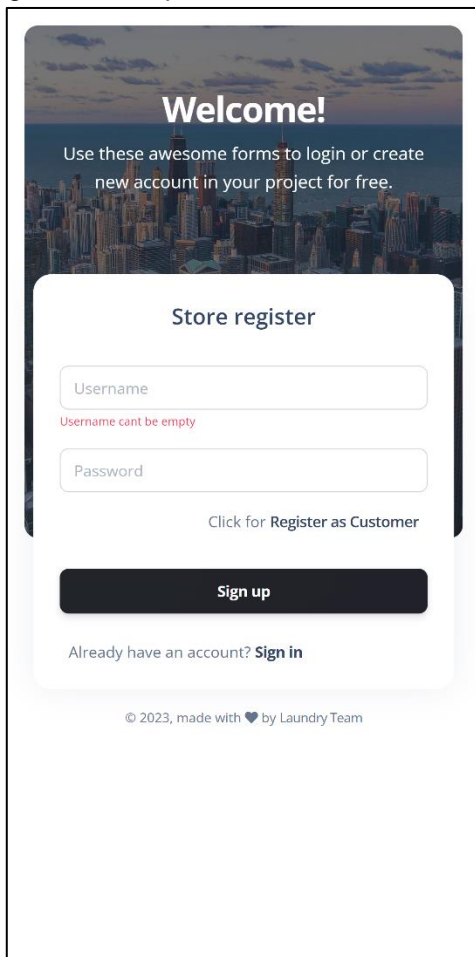
Gambar 19. Riwayat Transaksi Pelanggan

4.4 Black Box Testing

Pengujian antarmuka pengguna pada aplikasi mencakup beberapa tiga skenario utama untuk memastikan fungsionalitas dan pengalaman pengguna yang baik.

1. Pengujian dengan Formulir Kosong

Pada skenario ini, pengguna mengirimkan permintaan tanpa mengisi data *username* dan data *password*. Sistem akan merespon dengan kode status *400 Bad Request*, menampilkan pesan error "*Username can't be empty*" menunjukkan bahwa data yang dikirim tidak valid. Gambar 20. menunjukkan hasil pengujian pada API registrasi akun penatu.

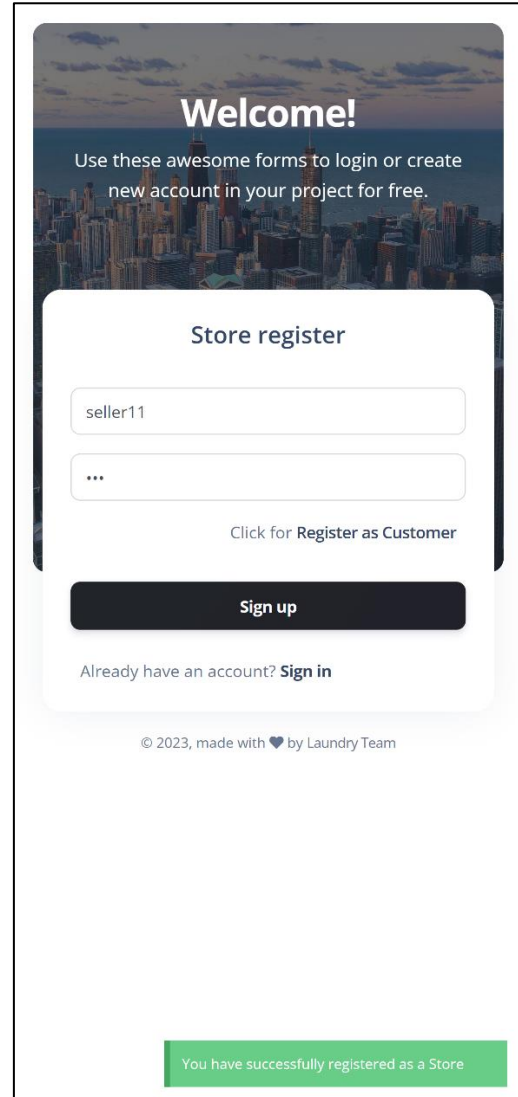


Gambar 20. Pengujian dengan Formulir Kosong

2. Pengujian dengan Data Sesuai Spesifikasi

Pengguna mengirimkan permintaan dengan data yang sesuai dengan spesifikasi yaitu *username* "seller11" dan *password* yg telah ditentukan. Sistem kemudian

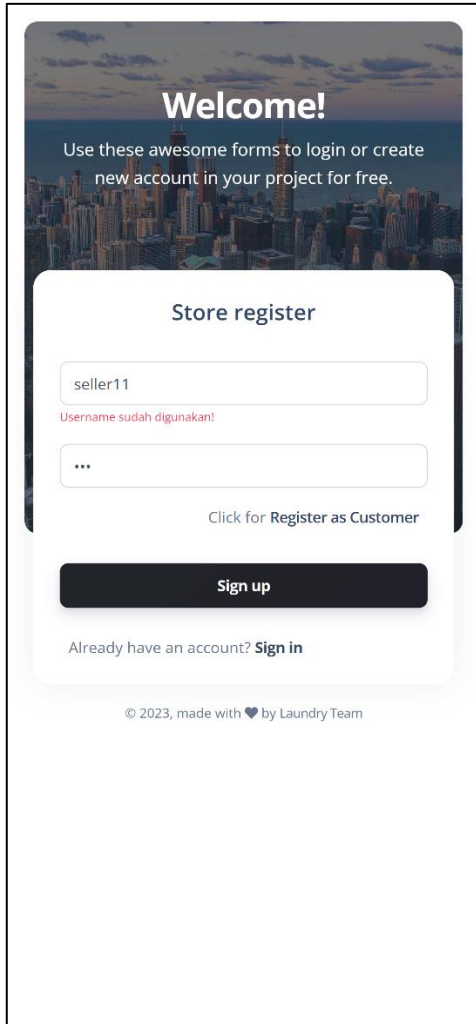
memberikan respons berhasil dengan kode status 201 *Created*, menandakan bahwa akun penatu berhasil didaftarkan. Gambar 21. menunjukkan hasil pengujian pada API registrasi akun penatu.



Gambar 21. Pengujian dengan Data Sesuai Spesifikasi

3. Pengujian dengan Data Berulang atau Tidak Valid

Pengguna mengirimkan permintaan dengan *username* "seller11" yang sebelumnya sudah ada dalam sistem. Akibatnya, sistem memberikan respons dengan kode status 409 *Conflict*, menampilkan pesan kesalahan bahwa *username* telah digunakan. Gambar 22. menunjukkan hasil pengujian pada API registrasi akun penatu.



Gambar 22. Pengujian dengan Data Berulang

Pengujian *black box* dilakukan untuk memastikan bahwa setiap fungsionalitas dalam aplikasi bekerja sesuai dengan spesifikasi yang diharapkan. Pada Tabel 3. menunjukkan bahwa aplikasi berfungsi sesuai harapan, dengan semua fitur utama seperti registrasi, login, dan pengelolaan data berhasil memenuhi target output.

Tabel 3. Hasil Uji Fungsionalitas

No.	Deskripsi Pengujian	Hasil
1	Admin menambahkan serta mengisi data <i>user</i> baru dengan <i>input</i> sesuai spesifikasi.	Fungsionalitas bekerja dengan baik.
2	Admin menambahkan serta mengisi data <i>user</i> baru dengan <i>input</i> kosong.	Sistem menampilkan pesan kesalahan yang sesuai.

3	Admin menambahkan serta mengisi data <i>user</i> baru dengan <i>input username</i> yang berulang.	Sistem menampilkan pesan kesalahan yang sesuai.
4	Admin melakukan <i>delete</i> kepada <i>user</i> yang diinginkan.	<i>User</i> yang dihapus tidak lagi muncul di <i>dashboard</i> admin.
5	Admin masuk ke kolom <i>search</i> serta melakukan <i>input</i> nama <i>user</i> yang diinginkan.	<i>User</i> yang dicari muncul sesuai dengan <i>keyword</i> .
6	Admin mengklik tombol <i>logout</i> pada <i>navigation bar</i> .	Admin diarahkan ke halaman <i>login</i> .
7	Penatu melakukan <i>input</i> data yang diperlukan untuk registrasi akun dengan <i>input</i> sesuai spesifikasi.	Registrasi akun berhasil dan Penatu dapat <i>login</i> .
8	Penatu melakukan <i>input</i> data yang diperlukan untuk registrasi akun dengan <i>input username</i> yang berulang.	Sistem menampilkan pesan kesalahan yang sesuai.
9	Penatu melakukan <i>input username</i> dan <i>password</i> .	Penatu berhasil <i>login</i> ke halaman <i>dashboard</i> .
10	Penatu memasukkan data yang salah.	Sistem menampilkan pesan kesalahan yang sesuai.
11	Penatu mengisi data <i>store</i> dengan <i>input</i> sesuai spesifikasi.	Penatu dapat mengakses halaman <i>store</i> .
12	Penatu mengisi data <i>store</i> dengan <i>input</i> kosong.	Sistem menampilkan pesan kesalahan yang sesuai.
13	Penatu mengisi data <i>store</i> dengan <i>input username</i> yang berulang.	Sistem menampilkan pesan kesalahan yang sesuai.
14	Penatu menekan tombol <i>add service item</i> dan mengisi <i>item</i> yang diinginkan.	<i>Item</i> baru muncul di <i>dashboard</i> penatu.
15	Penatu menekan tombol <i>delete</i> pada <i>item</i> yang ingin dihapus.	<i>Item</i> yang dipilih hilang dari <i>dashboard</i> penatu.

16	Penatu menekan tombol <i>add type</i> dan mengisi tipe layanan yang diinginkan.	Tipe layanan yang baru muncul di <i>dashboard</i> penatu.
17	Penatu menekan tombol <i>delete type</i> yang diinginkan.	Tipe layanan yang dipilih hilang dari <i>dashboard</i> penatu.
18	Penatu melakukan <i>input</i> jenis layanan berdasarkan durasi yang diinginkan.	Jenis layanan berdasarkan durasi muncul di <i>dashboard</i> penatu.
19	Penatu menekan tombol <i>delete duration</i> yang diinginkan.	Jenis layanan yang dipilih hilang dari <i>dashboard</i> penatu.
20	Penatu menambahkan layanan utama beserta dengan harga.	Layanan utama muncul di <i>dashboard</i> penatu.
21	Penatu melakukan <i>update</i> pada status transaksi pelanggan.	Status transaksi pelanggan diperbarui dengan benar.
22	Penatu masuk ke halaman riwayat transaksi kemudian menekan tombol rekap transaksi.	Rekap penjualan ditampilkan dalam bentuk file PDF.
23	Penatu menekan tombol <i>logout</i> pada <i>navigation bar</i> .	Penatu diarahkan ke halaman <i>login</i> .
24	Pelanggan melakukan <i>input</i> data yang diperlukan untuk registrasi akun.	Pelanggan berhasil <i>login</i> dengan data yang di- <i>input</i> saat registrasi.
25	Pelanggan melakukan <i>input username</i> yang sudah ada.	Sistem menampilkan pesan kesalahan yang sesuai.
26	Pelanggan melakukan <i>input username</i> dan <i>password</i> .	Pelanggan berhasil masuk ke <i>dashboard</i> pelanggan.
27	Pelanggan memasukkan data yang salah.	Sistem menampilkan pesan kesalahan yang sesuai.
28	Pelanggan melakukan <i>input</i> kata kunci nama penatu pada <i>search bar</i> .	Penatu yang sesuai dengan pencarian muncul.
29	Pelanggan melakukan <i>filtering</i>	Daftar penatu diurutkan

	berdasarkan lokasi terdekat.	berdasarkan jarak terdekat.
30	Pelanggan melakukan <i>filtering</i> berdasarkan biaya terendah.	Daftar penatu diurutkan berdasarkan biaya terendah.
31	Pelanggan menekan tombol penatu pada daftar penatu.	Detail penatu ditampilkan.
32	Pelanggan melakukan <i>input</i> jumlah pakaian dan melakukan <i>checkout</i> .	Pemesanan berhasil dan nota elektronik dicetak.
33	Pelanggan mengganti data pribadi yang sesuai.	Data pribadi pelanggan tersimpan dengan benar.
34	Pelanggan memasukkan <i>username</i> dan kata sandi baru yang diinginkan.	<i>Username</i> dan kata sandi diperbarui dengan benar.
35	Pelanggan masuk ke halaman riwayat pemesanan.	Halaman riwayat transaksi pelanggan ditampilkan dengan benar.
36	Pelanggan menekan tombol <i>logout</i> pada <i>navigation bar</i> .	Pelanggan diarahkan ke halaman <i>login</i> .

4.5 Usability Test

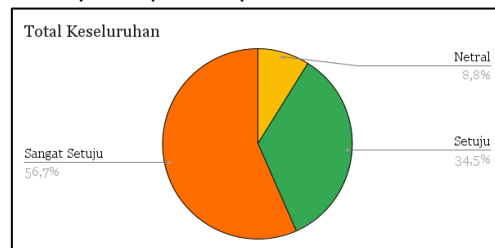
Pengujian *usability test* dilakukan untuk melihat respon pengguna terhadap kelayakan aplikasi *marketplace* penatu. *Usability test* ini melibatkan 30 responden dengan syarat minimal mahasiswa Teknik Komputer Semester 6. Pengujian ini menggunakan metode pengisian kuesioner yang terdiri dari 20 pertanyaan dan terdapat pilihan jawaban dari sangat tidak setuju hingga sangat setuju yang masing-masing memiliki bobot nilai 1-5 untuk setiap jawabannya. Daftar pertanyaannya dapat dilihat pada Tabel 4.

Tabel 4. Daftar Pertanyaan

No.	Pertanyaan
Aspek Kegunaan (Usefulness)	
1	Apakah aplikasi <i>marketplace</i> penatu membantu saya menjadi lebih efektif dalam mencari dan memesan layanan penatu?
2	Apakah aplikasi <i>marketplace</i> penatu membantu saya menjadi lebih produktif dalam mengelola pesanan penatu?

3	Aplikasi marketplace penatu sangat berguna dalam mengatur layanan penatu.
4	Aplikasi marketplace penatu membuat saya lebih fleksibel dalam mengatur waktu untuk layanan penatu.
5	Aplikasi marketplace penatu memudahkan proses pemesanan layanan penatu.
6	Aplikasi marketplace penatu memenuhi kebutuhan saya dalam mencari dan memesan layanan penatu.
Aspek Kemudahan Penggunaan (Ease of Use)	
7	Aplikasi marketplace penatu mudah digunakan.
8	Aplikasi marketplace penatu praktis digunakan.
9	Aplikasi marketplace penatu ramah pengguna.
10	Menggunakan aplikasi marketplace penatu tidak memerlukan banyak usaha.
11	Saya bisa menggunakan aplikasi marketplace penatu tanpa memerlukan instruksi tertulis atau panduan.
12	Saya tidak menemukan adanya inkonsistensi dalam aplikasi marketplace penatu.
13	Saya berhasil menggunakan aplikasi marketplace penatu setiap kali mencobanya.
Aspek Kemudahan Pembelajaran (Ease of Learning)	
14	Saya dapat dengan cepat belajar cara menggunakan aplikasi marketplace penatu.
15	Saya dapat dengan mudah mengingat cara menggunakan aplikasi marketplace penatu.
16	Aplikasi marketplace penatu mudah dipelajari.
17	Saya dengan cepat menjadi mahir menggunakan aplikasi marketplace penatu.
Aspek Kepuasan (Satisfaction)	
18	Saya puas dengan aplikasi marketplace penatu.
19	Saya akan merekomendasikan aplikasi marketplace penatu kepada teman dan keluarga.
20	Menggunakan aplikasi marketplace penatu sangat menyenangkan.

Pengujian 4 aspek *usability* yaitu kegunaan (*usefulness*), kemudahan penggunaan (*ease of use*), kemudahan pembelajaran (*ease of learning*), dan kepuasan (*satisfaction*), dilakukan dengan melibatkan 30 partisipan. Hasil pengujian menunjukkan bahwa 56,7% partisipan memberikan jawaban "sangat setuju", 34,5% memberikan jawaban "setuju", dan 8,8% memberikan jawaban "netral". Tidak ada responden yang memilih jawaban "tidak setuju" atau "sangat tidak setuju". Berikut adalah diagram hasil *usability test* aplikasi *marketplace* penatu pada Gambar 23.



Gambar 23. Diagram Hasil *Usability Test*

Tabel 5. menunjukkan hasil pengujian *usability* dari 30 responden yang mengisi 20 pertanyaan kuesioner. Setiap jawaban pada kuesioner dinilai dengan bobot dari 1 (Sangat Setuju) hingga 5 (Sangat Tidak Setuju), dan skor dihitung dengan mengalikan jumlah jawaban dengan bobot masing-masing kategori. Total skor diperoleh dengan menjumlahkan semua skor kategori.

Tabel 5. Hasil *Usability Test*

Pernyataan	Jawaban	Bobot	Skor
Sangat Tidak Setuju	340	5	1700
Tidak Setuju	207	4	828
Netral	53	3	159
Setuju	0	2	0
Sangat Setuju	0	1	0
Total			2687

Untuk menghitung persentase hasil, total skor dibagi dengan skor maksimum. Skor maksimum dihitung berdasarkan jumlah responden (30) dan jumlah pertanyaan (20), dengan bobot tertinggi 5 untuk setiap jawaban. Maka skor maksimumnya adalah 3000.

$$\text{Persentase} = \left(\frac{\text{Total Skor}}{\text{Skor Maksimum}} \right) \times 100\% \quad (1)$$

Berdasarkan rumus (1) didapatkan hasil persentase keseluruhan sebesar 89,57%, yang mana menunjukkan bahwa aplikasi *marketplace* penatu menerima tanggapan yang sangat positif dari para responden.

5. KESIMPULAN

Setelah melewati semua tahapan penelitian seperti tercantum pada poin 3.1, maka dapat disimpulkan bahwa:

1. Aplikasi *mobile web* berhasil dirancang dengan tampilan responsif yang menyesuaikan otomatis di berbagai perangkat, termasuk desktop, tablet, dan *smartphone*. Penggunaan flexbox menyederhanakan pengaturan tata letak, meningkatkan pengalaman pengguna dengan tingkat kepuasan 89,57% dari 30 responden. Fitur pencarian dan filter memudahkan pengguna dalam menemukan penatu yang sesuai, sekaligus membantu penatu dalam mendapatkan pelanggan.
2. *Framework* Express.js berhasil mengelola data dan permintaan dengan efisien. Fitur seperti *middleware* untuk *parsing* data, CORS, dan struktur *routing* modular membuat kode terorganisir dan aman. Penggunaan *async/await* mengurangi risiko *blocking*, sementara autentikasi JWT memastikan keamanan akses. Integrasi dengan basis data MySQL juga berjalan lancar.
3. Implementasi *Geocoding API* dan *Distance Matrix API* pada aplikasi *mobile web* untuk menentukan jarak antara lokasi pelanggan dan beberapa lokasi penatu terdaftar berhasil dilakukan. *Geocoding API* digunakan untuk mengonversi alamat fisik menjadi koordinat geografis, sementara *Distance Matrix API* menghitung jarak tempuh antara lokasi-lokasi tersebut. Hasil uji kegunaan menunjukkan tingkat kepuasan tinggi sebesar 89,22% dari 30 responden, yang menilai penggunaan kedua API ini efektif,

nyaman, dan bermanfaat. Penelitian ini membuktikan bahwa integrasi kedua API tersebut menyediakan solusi yang andal dan efisien, menghemat waktu dan tenaga pelanggan serta meningkatkan kenyamanan layanan aplikasi.

6. DAFTAR PUSTAKA

- [1] Budiman, M. J., & Yudianthi, N. (2022). Perancangan Sistem Informasi Pelayanan Jasa Usaha *Laundry* Sepatu Berbasis *Progressive Web App* (PWA).
- [2] Saragih, F., Dalimunthe, Y. A., & Lubis, H. (2021). Rancang Bangun Sistem *Tracking* Jasa *Laundry* Sepatu Di *Clinix Shoes Care* Berbasis *Website*. METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi, 5(1), 73–76.
- [3] Regmi, S. (2022). *Building A Responsive Website*.
- [4] Andriansyah, D. (2018). Penerapan Model Waterfall Pada Sistem Informasi Layanan Jasa *Laundry* Berbasis Web. IJSE-Indonesian Journal on Software Engineering, 4(1).
- [5] Syarifudin, G. (2018). Perancangan E-*Marketplace* Dalam Menghasilkan Sistem *Laundry* di Pontianak (Vol. 12).
- [6] Indriasari, T. D., & Sidhi, T. A. P. (2011). Sistem Pencarian Orang Hilang Berbasis *Mobile Web* Dengan *Social Network Analysis*. In Seminar Nasional Informatika.
- [7] StrongLoop, IBM, & other expressjs.com contributors. (2017). Express.js Documentation.
- [8] Yani, A., Setiawan, D., Egi, N., Subagja, R., & Desyani, T. (2020). Pengujian Aplikasi Reservasi Hotel di LeGreen Hotel & Suite dengan Metode *Black Box Testing Boundary Value Analysis*.
- [9] Lund, A. M. (2001). *Measuring Usability with the USE Questionnaire*. Usability Professional Association Conference.