

Rancang Bangun Aplikasi Jasa Tukang Berbasis Location-Based Service (LBS) Menggunakan Framework Next.js dan Express.js

Proyek Akhir

Disusun Oleh:

Ahmad Candra Ramadhan

3312301064

Disusun Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Ahli Madya
Teknik Informatika



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

2026

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena atas berkat, rahmat, dan karunia-Nya, penulis menyelesaikan laporan Tugas Akhir ini dengan baik dan tepat waktu. Adapun penulisan laporan Tugas Akhir yang berjudul **"Rancang Bangun Aplikasi Jasa Tukang Berbasis Location Based-Service (LBS) Menggunakan Framework Next.js dan Express.js"** ini disusun sebagai salah satu syarat utama untuk menyelesaikan program studi D3 Teknik Informatika dan meraih gelar A.Md Kom. di Politeknik Negeri Batam.

Penyusunan laporan ini tidak akan terwujud tanpa adanya bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang tulus kepada:

1. Bapak Dr. Ari Wibowo, S.T., M.T., selaku Dosen Pembimbing, yang telah dengan sabar memberikan banyak waktu, bimbingan, arahan, dan motivasi sejak awal hingga selesainya penyusunan laporan ini.
2. Semua pihak lain yang tidak dapat disebutkan satu per satu, yang telah turut membantu kelancaran penyusunan laporan Tugas Akhir ini.

Penulis menyadari bahwa masih terdapat kekurangan dalam laporan Tugas Akhir ini. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan di masa mendatang.

Batam, 2026

Penulis

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	4
DAFTAR GAMBAR.....	5
ABSTRAK.....	6
BAB I PENDAHULUAN.....	7
1.1. Latar Belakang.....	7
1.2. Rumusan Masalah.....	8
1.3. Tujuan.....	9
1.4. Batasan Masalah.....	10
1.5. Manfaat.....	10
1.6. Pembagian Peran dan Tanggung Jawab Tim.....	12
BAB II TINJAUAN PUSTAKA.....	13
2.1. Penelitian Terkait.....	13
2.2. Landasan Teori.....	15
2.3. Metode Pengembangan Produk.....	21
BAB III ANALISIS DAN PERANCANGAN.....	25
3.1. Analisis Kebutuhan.....	25
3.2. Perancangan.....	27
BAB IV ANALISIS DAN PERANCANGAN.....	40
4.1. Hasil Implementasi.....	40
4.2. Pengujian Sistem.....	53
BAB V KESIMPULAN DAN SARAN.....	62
5.1 Kesimpulan.....	62
5.2 Saran.....	63
DAFTAR PUSTAKA.....	64

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Sebelumnya	14
Tabel 3.1 Kebutuhan Fungsional Sistem	27
Tabel 3.2 Kebutuhan Non-Fungsional Sistem	29
Tabel 3.3 Skenario Register	31
Tabel 3.4 Skenario Login	31
Tabel 3.5 Verifikasi tukang	32
Tabel 3.6 Skenario Pencarian Tukang Berdasarkan Lokasi,	32
Tabel 3.7 Skenario Membuat pesanan	33
Tabel 3.8 Skenario Mengelola Pesanan	33
Tabel 3.9 Skenario Menyelesaikan Pesanan	34
Tabel 3.10 Skenario Memberikan Rating	34
Tabel 4.1 Implementasi API	42
Tabel 4.2 Test Case Fungsional	53
Tabel 4.3 Test Case Non Fungsional	60

DAFTAR GAMBAR

Gambar 2.1 Metode Agile.....	22
Gambar 3.1 Gambaran Umum Sistem.....	26
Gambar 3.2 Use Case Diagram.....	30
Gambar 3.3 Activity Diagram.....	38
Gambar 3.4 ER Diagram.....	39
Gambar 4.1 Implementasi ERD.....	41

ABSTRAK

Perkembangan teknologi informasi telah membawa perubahan besar dalam cara masyarakat mengakses layanan, termasuk sektor jasa. Namun, sebagian besar layanan tukang seperti tukang bangunan, tukang servis AC, dan tukang sedot WC masih menggunakan metode konvensional. Hal ini menyebabkan proses pencarian tukang menjadi kurang efisien dan sulit dijangkau oleh masyarakat umum. Laporan akhir ini bertujuan untuk merancang dan membangun aplikasi jasa tukang berbasis website menggunakan geolocation service agar pengguna dapat menemukan tukang terdekat. Metode pengembangan yang digunakan adalah model Waterfall dengan tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Aplikasi akan dibangun menggunakan *framework* Next.js dengan integrasi API geolocation untuk menampilkan posisi tukang terdekat. Hasil yang diharapkan adalah sebuah prototipe aplikasi web yang memudahkan pengguna mencari, memesan, dan menilai jasa tukang secara transparan. Sistem ini juga diharapkan dapat meningkatkan aksesibilitas dan profesionalitas penyedia jasa di Indonesia.

Kata Kunci: Jasa Tukang, Website, Geolocation, Sistem Informasi

BAB I PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah membawa perubahan signifikan pada berbagai sektor kehidupan, termasuk sektor jasa dan ketenagakerjaan. Digitalisasi layanan tidak hanya mempermudah akses masyarakat terhadap berbagai kebutuhan, tetapi juga membuka peluang baru bagi para pekerja informal untuk memperluas jangkauan pasar mereka. Di Indonesia, sektor jasa informal seperti tukang bangunan, tukang listrik, tukang ledeng, dan berbagai jenis pekerja terampil lainnya memiliki peran yang sangat vital dalam memenuhi kebutuhan masyarakat sehari-hari.

Berdasarkan data Badan Pusat Statistik (BPS) tahun 2023, jumlah pekerja informal di Indonesia mencapai lebih dari 80 juta orang, dengan sebagian besar berada di sektor jasa dan konstruksi. Namun demikian, mayoritas pekerja informal ini masih mengandalkan cara tradisional dalam mendapatkan pekerjaan, seperti informasi dari mulut ke mulut, melalui kenalan, atau hanya menunggu di lokasi tertentu. Cara konvensional ini memiliki berbagai keterbatasan, antara lain jangkauan pasar yang sempit, ketidakpastian pendapatan, serta minimnya kepercayaan dari calon pelanggan karena tidak adanya sistem verifikasi dan penilaian yang terstandarisasi.

Di sisi lain, masyarakat sebagai pengguna jasa juga menghadapi tantangan tersendiri ketika membutuhkan tenaga kerja terampil. Proses pencarian tukang yang kompeten, terpercaya, dan berlokasi dekat seringkali membutuhkan waktu yang cukup lama dan tidak efisien. Tidak adanya platform terpusat yang menghubungkan antara pengguna jasa dengan para pekerja terampil di sekitar lokasi mereka menjadi hambatan utama yang perlu diselesaikan. Kebutuhan akan solusi digital yang mampu menjembatani kesenjangan ini semakin mendesak, terutama di era transformasi digital seperti saat ini.

Teknologi Location-Based Service (LBS) hadir sebagai solusi yang memungkinkan pengguna untuk menemukan layanan berdasarkan kedekatan

lokasi secara efisien. Konsep ini telah berhasil diterapkan pada berbagai platform sukses di dunia, seperti Gojek dan Grab untuk layanan transportasi, serta Airbnb dan Booking.com untuk layanan penginapan. Dengan memanfaatkan teknologi geolokasi, sebuah aplikasi dapat menyajikan daftar pekerja terampil yang tersedia dalam radius tertentu dari lokasi pengguna, sehingga mempermudah proses pencarian dan pemesanan jasa.

Berdasarkan permasalahan yang telah diuraikan di atas, maka diusulkan sebuah pengembangan dengan judul "Rancang Bangun Aplikasi Jasa Tukang Berbasis Location Based-Service menggunakan framework Next.js dan Express.js".

Aplikasi ini dirancang sebagai platform digital yang mempertemukan antara pengguna jasa dengan para pekerja terampil (tukang) berdasarkan kedekatan lokasi. Dengan memanfaatkan teknologi PostgreSQL dengan ekstensi PostGIS untuk pengelolaan data geospasial, serta framework Next.js untuk antarmuka pengguna, diharapkan aplikasi ini dapat menjadi solusi yang efektif dan efisien dalam memenuhi kebutuhan masyarakat akan jasa pekerja terampil di sekitar mereka.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka rumusan masalah dalam pengembangan sistem ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem pencarian pekerja terampil (tukang) berbasis lokasi yang dapat menampilkan pekerja dalam radius tertentu dari posisi pengguna?
2. Bagaimana mengimplementasikan fitur registrasi dan manajemen profil pekerja terampil yang mencakup data kompetensi, ketersediaan, dan koordinat lokasi?

3. Bagaimana merancang sistem pemesanan dan penilaian layanan yang dapat meningkatkan kepercayaan antara pengguna jasa dan pekerja terampil?
4. Bagaimana mengimplementasikan teknologi geospasial menggunakan PostgreSQL dan PostGIS untuk mendukung pencarian berbasis radius secara akurat dan efisien?

1.3. Tujuan

Berdasarkan rumusan masalah yang telah ditetapkan, maka tujuan dari pengembangan ini adalah sebagai berikut:

1. Merancang dan membangun sistem pencarian pekerja terampil berbasis lokasi yang dapat menampilkan pekerja tersedia dalam radius tertentu dari posisi pengguna menggunakan ekstensi geospasial PostGIS pada sistem manajemen basis data .
2. Mengimplementasikan fitur registrasi dan manajemen profil pekerja terampil yang mencakup pengelolaan data kompetensi, ketersediaan, dan penyimpanan koordinat lokasi secara permanen.
3. Merancang sistem pemesanan dan penilaian layanan yang transparan dan terverifikasi untuk meningkatkan kepercayaan antara pengguna jasa dan pekerja terampil.
4. Mengimplementasikan teknologi geospasial menggunakan PostgreSQL dengan ekstensi PostGIS sebagai basis data utama untuk mendukung pencarian berbasis radius yang akurat dan efisien.

1.4. Batasan Masalah

Agar pengembangan ini dapat dilaksanakan dengan terarah dan terfokus, maka ditetapkan batasan masalah sebagai berikut:

1. Aplikasi ini dikembangkan sebagai aplikasi berbasis web menggunakan framework Next.js untuk antarmuka pengguna dan Node.js (Express.js) untuk backend, serta tidak mencakup pengembangan aplikasi mobile native.
2. Sistem geolokasi yang digunakan adalah geolokasi statis, di mana koordinat lokasi pekerja terampil disimpan pada saat registrasi dan tidak berubah secara real-time selama sesi aktif.
3. Fitur pencarian mencakup radius jarak yang bisa dipilih dari lokasi pengguna, dengan tampilan peta menggunakan library Leaflet.js yang terintegrasi dengan OpenStreetMap.
4. Sistem pembayaran tidak diimplementasikan dalam aplikasi ini; transaksi keuangan antara pengguna dan pekerja dilakukan secara langsung di luar platform.
5. Kategori pekerja terampil yang tersedia dapat ditambahkan berdasarkan kebutuhan tukang melalui kontak saran atau *suggestion*.
6. Sistem verifikasi identitas pekerja hanya mencakup validasi nomor identitas (KTP) secara manual oleh administrator, tanpa integrasi dengan layanan verifikasi pihak ketiga.
7. Notifikasi pada aplikasi ini terbatas pada notifikasi dalam aplikasi (in-app notification)

1.5. Manfaat

TULEN dirancang sebagai marketplace dua sisi (two-sided marketplace) yang menjembatani dua masalah yang saling berkaitan: pekerja terampil (tukang)

yang kesulitan memperluas jangkauan pasar dan promosi diri, serta pengguna jasa yang kesulitan menemukan tukang kompeten dan terpercaya di sekitar lokasi mereka secara cepat. Sebagaimana model marketplace berbasis lokasi lain (Gojek, Grab, Airbnb), kedua sisi ini saling menguatkan — semakin banyak tukang terverifikasi yang bergabung, semakin mudah pengguna jasa menemukan yang sesuai kebutuhan, dan sebaliknya. Manfaat pengembangan ini bagi masing-masing pihak dapat diuraikan sebagai berikut:

1. Bagi Pengguna Jasa:

- Mempermudah proses pencarian pekerja terampil yang berada di sekitar lokasi pengguna secara cepat dan efisien.
- Memberikan akses terhadap informasi profil, kompetensi, dan penilaian pekerja secara transparan sehingga meningkatkan kepercayaan dalam proses pemesanan jasa.
- Menghemat waktu dan biaya yang sebelumnya digunakan untuk mencari pekerja terampil melalui cara konvensional.

2. Bagi Pekerja Terampil (Tukang):

- Memperluas jangkauan pasar dan meningkatkan peluang mendapatkan pekerjaan melalui platform digital yang dapat diakses oleh banyak pengguna.
- Memberikan media promosi diri yang terstandarisasi dan profesional melalui fitur profil dan portofolio digital.
- Meningkatkan kepercayaan calon pelanggan melalui sistem penilaian dan ulasan yang transparan.

3. Bagi Akademisi dan Peneliti:

- Memberikan referensi dan contoh implementasi teknologi Location-Based Service dan geospasial dalam pengembangan aplikasi marketplace berbasis lokasi.

- Menambah khasanah pengetahuan dalam bidang rekayasa perangkat lunak, khususnya terkait implementasi query geospasial menggunakan PostgreSQL/PostGIS.

4. Bagi Institusi Pendidikan:

- Merupakan bukti nyata kemampuan mahasiswa dalam mengaplikasikan ilmu yang diperoleh selama masa studi ke dalam sebuah produk teknologi yang bernilai guna bagi masyarakat.

1.6. Pembagian Peran dan Tanggung Jawab Tim

Proyek Akhir ini dilaksanakan secara mandiri (*individual project*), di mana seluruh tahapan siklus pengembangan sistem dikerjakan secara penuh oleh penulis tanpa adanya pembagian anggota tim. Meskipun dilakukan secara perorangan, manajemen pengerjaan tetap mengacu pada pembagian fungsi peran dalam rekayasa perangkat lunak secara akuntabel. Dalam hal ini, penulis bertanggung jawab penuh sebagai *Project Manager* untuk mengelola lini masa proyek, sekaligus bertindak sebagai *System Analyst* yang menganalisis kebutuhan pengguna serta merancang arsitektur diagram sistem. Selain itu, penulis juga berperan sebagai *UI/UX Designer* dalam merancang visualisasi antarmuka aplikasi, sebagai *Software Developer* yang melakukan pengodean basis data dan implementasi *query* geospasial menggunakan PostGIS, serta bertindak sebagai *Quality Assurance* untuk menyusun skenario pengujian guna memastikan fungsionalitas aplikasi bebas dari kendala teknis.

BAB II TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian terkait merupakan bagian yang menguraikan penelitian-penelitian terdahulu yang memiliki relevansi dengan topik Proyek Akhir ini. Tujuannya adalah untuk mengetahui posisi penelitian ini dibandingkan dengan karya-karya sebelumnya, serta untuk menunjukkan kebaruan (novelty) dari penelitian yang dilakukan. Berikut adalah beberapa penelitian terdahulu yang berkaitan dengan pengembangan aplikasi berbasis lokasi dan marketplace jasa.

Tabel 2.1 Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan /Celah
Fajar & Ramadan (2021)	Aplikasi Pencarian Jasa Servis Elektronik Berbasis Android Menggunakan GPS	Android Native, Google Maps API, MySQL	Antarmuka mobile yang familiar, akurasi lokasi tinggi menggunakan GPS real-time	Hanya tersedia di platform Android, tidak ada sistem penilaian pekerja
Hidayat et al. (2022)	Rancang Bangun Marketplace Jasa Berbasis Web dengan Fitur Geolokasi	PHP CodeIgniter, Leaflet.js, MySQL, OpenStreetMap	Platform web multiplatform, integrasi peta open-source tanpa biaya API	Tidak menggunakan database geospasial, query radius kurang optimal untuk data besar
Pratiwi & Nugroho (2022)	Sistem Rekomendasi Tukang Bangunan Berbasis Lokasi Menggunakan Algoritma Haversine	Laravel, Google Maps, PostgreSQL, Algoritma Haversine	Perhitungan jarak akurat, sistem rekomendasi berdasarkan	Tidak menggunakan ekstensi PostGIS, antarmuka kurang

Peneliti/Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan /Celah
			rating dan jarak	modern dan responsif
Santoso et al. (2023)	Pengembangan Aplikasi Jasa Pekerja Rumah Tangga Berbasis Mobile dengan Fitur Real-Time Tracking	React Native, Node.js, Firebase, Google Maps	Tracking real-time pekerja, notifikasi push, UI modern multiplatform	Tracking real-time boros baterai, biaya Firebase meningkat seiring data pengguna
Kurniawan & Susanti (2023)	Implementasi Location-Based Service pada Platform Freelancer Lokal Berbasis Web	Vue.js, Express.js, MongoDB, Mapbox API	Pencarian berdasarkan radius, filter kategori, manajemen portofolio pekerja	Menggunakan Mapbox berbayar, MongoDB kurang optimal untuk relasi data kompleks
Penelitian Ini	Rancang Bangun Aplikasi Marketplace Jasa Tukang Berbasis Website Menggunakan Location-Based Service (LBS)	Next.js, Express.js, PostgreSQL + PostGIS, Leaflet.js, Waterfall	Database geospasial PostGIS untuk query radius optimal, peta gratis Leaflet.js, sistem penilaian & portofolio, platform web responsif	-

Berdasarkan perbandingan pada Tabel 2.1, dapat diidentifikasi beberapa celah penelitian yang belum diselesaikan oleh penelitian-penelitian sebelumnya.

Penelitian Fajar & Ramadan (2021) hanya terbatas pada platform Android dan belum memiliki sistem penilaian. Hidayat et al. (2022) tidak memanfaatkan database geospasial sehingga query radius kurang optimal. Pratiwi & Nugroho (2022) menggunakan PostgreSQL namun belum mengimplementasikan ekstensi PostGIS yang lebih powerful. Santoso et al. (2023) mengandalkan real-time tracking yang boros sumber daya, sementara Kurniawan & Susanti (2023) menggunakan layanan peta berbayar.

Penelitian ini hadir untuk menjawab celah-celah tersebut dengan mengimplementasikan database geospasial PostgreSQL + PostGIS untuk query radius yang optimal, menggunakan Leaflet.js dengan OpenStreetMap sebagai alternatif peta gratis dan open-source, menerapkan geolokasi statis yang lebih hemat sumber daya, serta membangun platform berbasis web yang dapat diakses dari berbagai perangkat. Kombinasi teknologi ini belum pernah diimplementasikan secara bersamaan pada konteks marketplace jasa tukang di penelitian-penelitian sebelumnya.

2.2. Landasan Teori

2.2.1 Marketplace Digital

Marketplace digital adalah platform elektronik yang mempertemukan antara penjual dan pembeli atau penyedia jasa dan pengguna jasa dalam satu ekosistem digital yang terintegrasi. Menurut Turban et al. (2018), marketplace digital berfungsi sebagai perantara yang memfasilitasi transaksi antara dua pihak atau lebih dengan menyediakan infrastruktur teknologi, sistem pembayaran, dan mekanisme kepercayaan seperti rating dan ulasan. Contoh marketplace digital yang sukses di Indonesia antara lain Tokopedia, Bukalapak, dan GoJek yang telah mengubah cara masyarakat mengakses berbagai layanan.

Dalam konteks pengembangan ini, aplikasi Tukang Online dikembangkan sebagai marketplace jasa yang menghubungkan antara pekerja terampil (tukang) sebagai penyedia jasa dengan masyarakat sebagai pengguna jasa. Perbedaan utama marketplace jasa dengan marketplace produk adalah bahwa layanan yang

ditawarkan bersifat tidak berwujud (intangible) dan penyampaianannya memerlukan kehadiran fisik, sehingga faktor kedekatan lokasi menjadi sangat krusial dalam proses pencocokan antara penyedia dan pengguna jasa.

2.2.2 Location-Based Service (LBS)

Location-Based Service (LBS) adalah layanan berbasis teknologi informasi yang memanfaatkan data geografis atau lokasi untuk menyajikan informasi atau layanan yang relevan berdasarkan posisi pengguna. Menurut Schiller & Voisard (2004), LBS didefinisikan sebagai layanan yang memanfaatkan posisi geografis perangkat mobile untuk memberikan informasi, hiburan, atau keamanan kepada pengguna. LBS memiliki dua komponen utama, yaitu komponen posisi yang menentukan lokasi pengguna, dan komponen layanan yang menyajikan informasi berdasarkan lokasi tersebut.

Dalam aplikasi Tukang Online, teknologi LBS diimplementasikan untuk memungkinkan pengguna menemukan pekerja terampil yang berlokasi dalam radius tertentu dari posisi mereka. Koordinat lokasi pekerja disimpan secara permanen dalam database pada saat registrasi, sementara koordinat pengguna diperoleh melalui browser geolocation API pada saat pencarian dilakukan. Pendekatan ini mengadopsi model geolokasi statis yang efisien dan tidak memerlukan pembaruan lokasi secara terus-menerus.

2.2.3 Geographic Information System (GIS) dan PostGIS

Geographic Information System (GIS) adalah sistem informasi yang dirancang untuk menangkap, menyimpan, memanipulasi, menganalisis, mengelola, dan menyajikan semua jenis data geografis. Menurut Longley et al. (2015), GIS memungkinkan pengguna untuk mengintegrasikan data spasial dengan data atribut sehingga dapat dilakukan analisis geografis yang kompleks. Dalam pengembangan aplikasi berbasis lokasi, konsep GIS diterapkan untuk mengelola dan memproses data koordinat pengguna dan pekerja.

PostGIS adalah ekstensi untuk database PostgreSQL yang menambahkan dukungan objek geografis, memungkinkan lokasi dijalankan dalam SQL. PostGIS

mengimplementasikan spesifikasi Simple Features for SQL (SFSQL) dari Open Geospatial Consortium (OGC) dan menyediakan fungsi-fungsi geospasial seperti ST_Distance, ST_DWithin, dan ST_GeogFromText yang memungkinkan query berbasis jarak dan radius secara efisien. Penggunaan PostGIS dalam pengembangan ini memungkinkan pencarian pekerja dalam radius tertentu dilakukan langsung pada level database tanpa memerlukan pemrosesan di sisi aplikasi.

2.2.4 Sistem Informasi

Sistem informasi adalah kombinasi teratur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. O'Brien & Marakas (2011) mendefinisikan sistem informasi sebagai kombinasi terorganisir dari manusia, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber data yang mengumpulkan, mengolah, dan mendistribusikan informasi. Sistem informasi berperan penting dalam mendukung pengambilan keputusan, koordinasi, pengendalian, analisis, dan visualisasi dalam suatu organisasi.

Aplikasi Tukang Online merupakan bentuk implementasi sistem informasi berbasis web yang berfungsi untuk mengelola data pekerja terampil, permintaan jasa, transaksi pemesanan, dan penilaian layanan. Sistem ini dirancang untuk memenuhi kebutuhan dua jenis pengguna utama, yaitu pekerja terampil yang membutuhkan platform untuk mempromosikan jasa mereka, dan masyarakat yang membutuhkan kemudahan dalam menemukan dan memesan jasa pekerja terampil di sekitar lokasi mereka.

2.2.5 Aplikasi Berbasis Web

Aplikasi berbasis web adalah perangkat lunak yang diakses melalui browser internet dan berjalan di atas infrastruktur web server. Berbeda dengan aplikasi desktop yang diinstal pada komputer pengguna, aplikasi web dapat diakses dari berbagai perangkat dan sistem operasi selama memiliki koneksi internet dan browser yang kompatibel. Menurut Reese (2000), aplikasi web

memberikan keunggulan dalam hal kemudahan akses, pembaruan terpusat, dan tidak memerlukan proses instalasi pada sisi klien.

Pemilihan platform web dalam pengembangan aplikasi Tukang Online didasarkan pada pertimbangan aksesibilitas yang lebih luas, kemudahan pemeliharaan, dan tidak adanya keterbatasan pada sistem operasi tertentu. Pengguna dapat mengakses aplikasi melalui smartphone maupun komputer desktop menggunakan browser tanpa perlu menginstal aplikasi tambahan, sehingga memperluas jangkauan pengguna potensial dari aplikasi ini.

2.2.6 Next.js

Next.js adalah framework berbasis React yang dikembangkan oleh Vercel untuk membangun aplikasi web modern yang bersifat server-side rendering (SSR) maupun client-side rendering (CSR). Next.js menyediakan berbagai fitur bawaan seperti routing berbasis file, optimasi performa, dan integrasi API yang memudahkan pengembangan aplikasi fullstack.

Menurut beberapa literatur, Next.js merupakan pengembangan dari React yang bertujuan untuk menyederhanakan proses pembuatan aplikasi web dengan arsitektur yang lebih terstruktur dan efisien. Dengan dukungan fitur seperti automatic routing dan hybrid rendering, Next.js mampu meningkatkan performa dan SEO dibandingkan aplikasi React konvensional.

Dalam pengembangan ini, Next.js digunakan sebagai frontend untuk membangun antarmuka pengguna aplikasi Tukang Online. Penggunaan Next.js memungkinkan pengelolaan halaman yang lebih terstruktur, integrasi komponen yang efisien, serta kemudahan dalam pengembangan fitur seperti pencarian berbasis lokasi dan tampilan peta interaktif.

2.2.7 Node.js dan Express.js

Node.js adalah runtime environment JavaScript yang berjalan di sisi server, dibangun di atas mesin JavaScript V8 milik Google Chrome. Node.js memungkinkan pengembang untuk menggunakan JavaScript tidak hanya di sisi

klien (browser) tetapi juga di sisi server, sehingga memungkinkan pengembangan full-stack menggunakan satu bahasa pemrograman. Menurut Tilkov & Vinoski (2010), Node.js menggunakan model event-driven dan non-blocking I/O yang membuatnya sangat efisien untuk menangani banyak koneksi secara bersamaan.

Express.js adalah framework web minimalis untuk Node.js yang menyediakan fitur-fitur esensial untuk membangun aplikasi web dan RESTful API. Dalam pengembangan ini, Node.js bersama Express.js digunakan sebagai backend server yang bertanggung jawab untuk menangani request dari frontend, melakukan validasi data, memproses logika bisnis, dan berkomunikasi dengan database PostgreSQL. Arsitektur RESTful API yang dibangun memungkinkan pemisahan yang jelas antara frontend dan backend, sehingga meningkatkan skalabilitas dan kemudahan pemeliharaan aplikasi.

2.2.8 PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional (RDBMS) open-source yang dikenal dengan keandalan, fleksibilitas, dan dukungannya terhadap standar SQL yang lengkap. PostgreSQL mendukung berbagai tipe data lanjutan, termasuk tipe data JSON, array, dan — dengan ekstensi PostGIS — tipe data geometri dan geografi. Menurut Stonebraker & Rowe (1986), PostgreSQL (awalnya dikenal sebagai POSTGRES) dirancang dengan arsitektur yang extensible, memungkinkan penambahan tipe data, operator, dan fungsi baru melalui sistem ekstensi.

Dalam penelitian ini, PostgreSQL dipilih sebagai database utama karena kemampuannya yang unggul dalam menangani data relasional yang kompleks dan dukungan penuh terhadap ekstensi PostGIS untuk keperluan geospasial. Skema database mencakup tabel-tabel untuk data pengguna, profil pekerja, kategori jasa, pemesanan, dan penilaian, yang semuanya saling berelasi dengan menggunakan foreign key untuk menjaga integritas data.

2.2.9 Leaflet.js

Leaflet.js adalah library JavaScript open-source yang ringan dan mobile-friendly untuk menampilkan peta interaktif pada aplikasi web. Dikembangkan oleh Vladimir Agafonkin, Leaflet.js menyediakan API yang sederhana namun powerful untuk menampilkan tile map, menambahkan marker, menggambar radius lingkaran, dan menangani interaksi pengguna pada peta. Keunggulan utama Leaflet.js dibandingkan alternatif berbayar seperti Google Maps API adalah tidak adanya biaya penggunaan dan kemudahan integrasinya dengan berbagai tile provider termasuk OpenStreetMap.

Dalam aplikasi Tukang Online, Leaflet.js diintegrasikan melalui library react-leaflet untuk menampilkan peta lokasi pekerja terampil sebagai marker, radius pencarian sebagai lingkaran, serta posisi pengguna. Penggunaan OpenStreetMap sebagai sumber data tile peta memastikan bahwa aplikasi dapat beroperasi tanpa biaya API dan tidak memiliki keterbatasan kuota request seperti pada layanan peta berbayar.

2.2.10 RESTful API

REST (Representational State Transfer) adalah gaya arsitektur perangkat lunak untuk sistem hypermedia terdistribusi seperti World Wide Web. RESTful API adalah antarmuka pemrograman aplikasi yang mengikuti prinsip-prinsip REST, menggunakan protokol HTTP dengan metode GET, POST, PUT, dan DELETE untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada sumber daya. Menurut Fielding (2000), REST menekankan antarmuka yang seragam, komunikasi tanpa state (stateless), dan pemisahan antara klien dan server.

Dalam penelitian ini, RESTful API dibangun menggunakan Node.js dan Express.js sebagai jembatan komunikasi antara frontend Next.js dan database

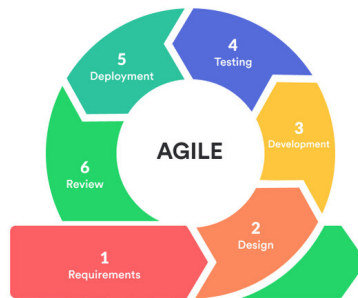
PostgreSQL. Setiap endpoint API dirancang untuk menangani operasi spesifik, seperti endpoint pencarian tukang yang menerima parameter koordinat dan radius untuk kemudian mengembalikan daftar pekerja dalam jangkauan menggunakan query geospasial PostGIS.

2.3. Metode Pengembangan Produk

Metode pengembangan produk yang digunakan dalam pembuatan sistem ini adalah **Agile Development**. Metode Agile merupakan pendekatan pengembangan perangkat lunak yang bersifat iteratif dan fleksibel, di mana proses pengembangan dilakukan secara bertahap melalui beberapa siklus pengembangan (iteration). Setiap iterasi menghasilkan peningkatan atau penambahan fitur yang dapat dievaluasi dan disesuaikan berdasarkan kebutuhan pengguna maupun hasil pengujian yang dilakukan.

Metode Agile dipilih karena mampu memberikan fleksibilitas terhadap perubahan kebutuhan selama proses pengembangan berlangsung. Selain itu, Agile memungkinkan pengembang untuk melakukan evaluasi dan perbaikan secara berkelanjutan sehingga kualitas sistem dapat terus ditingkatkan. Pendekatan ini juga mendukung komunikasi yang lebih baik antara pengembang dan pengguna melalui proses umpan balik (feedback) yang dilakukan secara berkala.

Tahapan dalam metode Agile yang digunakan pada penelitian ini meliputi **Perencanaan (Planning)**, **Perancangan (Design)**, **Pengembangan (Development)**, **Pengujian (Testing)**, **Evaluasi (Review)**, dan **Rilis (Release)**. Proses tersebut dilakukan secara berulang hingga sistem yang dikembangkan memenuhi kebutuhan pengguna. Alur metode Agile dapat dilihat pada Gambar 1.



Gambar 2.1 Metode Agile

a. Requirements Analysis (Analisis Kebutuhan)

Tahap perencanaan merupakan tahap awal dalam metode Agile yang bertujuan untuk mengidentifikasi kebutuhan sistem dan menentukan prioritas pengembangan fitur. Pada tahap ini dilakukan pengumpulan kebutuhan dari pengguna, analisis kebutuhan fungsional dan non-fungsional, serta penyusunan backlog pengembangan.

Adapun kegiatan yang dilakukan meliputi analisis kebutuhan aplikasi Tukang Online, seperti fitur pencarian tukang berbasis lokasi, manajemen profil tukang, sistem pemesanan layanan, dan sistem penilaian. Hasil dari tahap ini berupa daftar kebutuhan sistem (product backlog) yang akan menjadi acuan dalam proses pengembangan.

b. System Design (Perancangan Sistem)

Tahap perancangan dilakukan untuk menghasilkan gambaran sistem yang akan dikembangkan berdasarkan kebutuhan yang telah diidentifikasi. Perancangan dilakukan secara bertahap dan dapat disesuaikan pada setiap iterasi pengembangan.

Pada tahap ini dilakukan perancangan Use Case Diagram, Activity Diagram, Entity Relationship Diagram (ERD), desain basis data PostgreSQL dengan ekstensi PostGIS, serta perancangan antarmuka pengguna menggunakan Figma. Selain itu, dirancang pula arsitektur sistem yang mencakup frontend

menggunakan Next.js, backend menggunakan Node.js/Express, dan basis data PostgreSQL.

c. Development (Pengembangan)

Tahap pengembangan merupakan proses implementasi rancangan sistem ke dalam bentuk kode program. Pengembangan dilakukan secara bertahap berdasarkan prioritas fitur yang telah ditentukan pada product backlog.

Implementasi dilakukan dengan membangun frontend menggunakan Next.js dan Leaflet.js untuk menampilkan peta interaktif, backend RESTful API menggunakan Node.js dan Express.js, serta pengelolaan data geospasial menggunakan PostgreSQL dengan ekstensi PostGIS. Pengembangan dilakukan secara iteratif mulai dari fitur autentikasi, manajemen profil, pencarian tukang berbasis lokasi, pemesanan layanan, hingga sistem penilaian.

d. Testing (Pengujian)

Tahap pengujian dilakukan pada setiap iterasi pengembangan untuk memastikan fitur yang dibangun berjalan sesuai kebutuhan dan bebas dari kesalahan yang mengganggu fungsi sistem.

Pengujian dilakukan menggunakan metode Black Box Testing untuk memverifikasi fungsi-fungsi aplikasi, User Acceptance Testing (UAT) untuk memperoleh umpan balik dari pengguna, serta pengujian performa query geospasial pada PostgreSQL/PostGIS. Hasil pengujian digunakan sebagai dasar perbaikan dan penyempurnaan fitur pada iterasi berikutnya.

e. Review dan Release

Tahap review dilakukan setelah pengembangan dan pengujian pada setiap iterasi selesai. Pada tahap ini dilakukan evaluasi terhadap fitur yang telah dikembangkan berdasarkan hasil pengujian dan umpan balik pengguna.

Setelah fitur dinyatakan sesuai dengan kebutuhan, sistem atau pembaruan fitur akan dirilis untuk digunakan. Hasil evaluasi dari tahap review digunakan

sebagai masukan untuk perencanaan iterasi berikutnya sehingga pengembangan dapat dilakukan secara berkelanjutan hingga seluruh kebutuhan sistem terpenuhi.

BAB III ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan

Analisis kebutuhan merupakan tahap awal dalam proses pengembangan perangkat lunak yang bertujuan untuk mengidentifikasi dan mendokumentasikan seluruh kebutuhan sistem yang akan dibangun. Pada tahap ini dilakukan pengumpulan informasi mengenai proses bisnis yang berjalan, identifikasi permasalahan yang ada, serta penetapan kebutuhan fungsional dan non-fungsional dari aplikasi Tukang Online. Hasil dari analisis kebutuhan ini menjadi dasar acuan dalam tahap perancangan sistem selanjutnya.

3.1.1 Proses Bisnis Berjalan

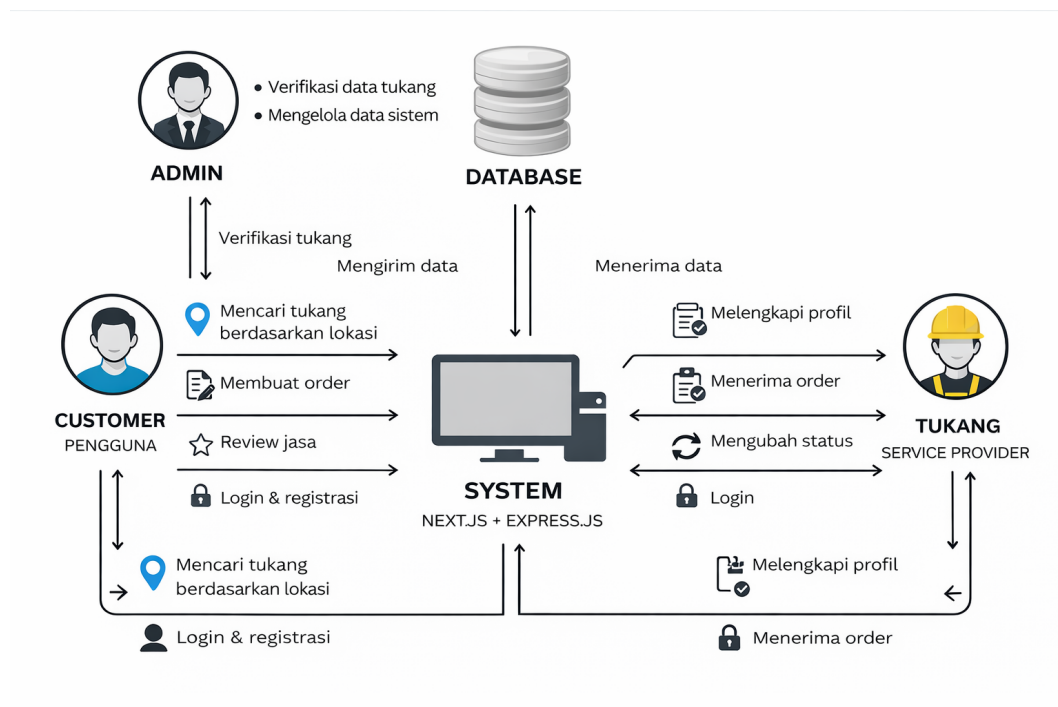
Proses pencarian dan pemesanan jasa tukang yang saat ini berjalan di masyarakat masih dilakukan secara konvensional dan belum terdigitalisasi. Ketika seseorang membutuhkan jasa pekerja terampil, proses yang umumnya dilakukan adalah dengan bertanya kepada tetangga, kerabat, atau kenalan untuk mendapatkan rekomendasi. Proses ini memakan waktu yang tidak pasti dan hasilnya sangat bergantung pada jaringan sosial yang dimiliki oleh individu tersebut.

Apabila rekomendasi berhasil diperoleh, pengguna kemudian menghubungi tukang melalui nomor telepon pribadi untuk menanyakan ketersediaan dan kesepakatan harga. Tidak jarang tukang yang direkomendasikan sudah tidak tersedia atau berlokasi terlalu jauh dari lokasi pengguna, sehingga proses pencarian harus diulang dari awal. Tidak adanya sistem penilaian yang terstandarisasi juga membuat pengguna kesulitan dalam menilai kualitas dan kredibilitas seorang tukang sebelum menggunakan jasanya.

Di sisi tukang, keterbatasan dalam mendapatkan pelanggan baru menjadi tantangan utama. Mayoritas tukang mengandalkan rekomendasi dari pelanggan

sebelumnya sebagai sumber pekerjaan utama, sehingga jangkauan pasar mereka sangat terbatas. Tidak adanya platform digital yang dapat mempromosikan keahlian dan ketersediaan mereka secara luas menyebabkan banyak tukang yang kompeten namun kesulitan mendapatkan pelanggan baru secara konsisten. Permasalahan-permasalahan inilah yang menjadi dasar diperlukannya sebuah sistem digital yang dapat menjembatani kebutuhan kedua pihak secara efisien.

3.1.2 Gambaran Umum Sistem



Gambar 3.1 Gambaran Umum Sistem

Aplikasi Tukang Online dirancang sebagai platform marketplace berbasis web yang menghubungkan tiga aktor utama, yaitu Pengguna (pelanggan), Tukang (pekerja terampil), dan Admin (pengelola platform). Sistem ini memanfaatkan teknologi Location-Based Service (LBS) dengan menyimpan koordinat lokasi tukang secara permanen pada saat registrasi, kemudian mencocokkannya dengan posisi pengguna pada saat pencarian dilakukan menggunakan query geospasial PostgreSQL + PostGIS.

Alur utama sistem berjalan sebagai berikut: tukang mendaftar dan mengisi profil lengkap termasuk koordinat lokasi, kemudian menunggu verifikasi dari Admin. Setelah diverifikasi, profil tukang akan muncul dalam hasil pencarian pengguna. Pengguna membuka aplikasi, mengizinkan akses lokasi browser, memilih kategori jasa dan radius pencarian, lalu sistem menampilkan daftar dan marker peta tukang yang tersedia dalam radius tersebut. Pengguna dapat melihat detail profil tukang, melakukan pemesanan, dan setelah pekerjaan selesai memberikan rating serta ulasan.

3.2. Perancangan

Perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan yang telah diuraikan pada subbab sebelumnya. Bagian ini memuat seluruh rancangan sistem aplikasi Tukang Online.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fitur-fitur dan fungsi-fungsi yang harus dimiliki oleh sistem untuk memenuhi kebutuhan seluruh aktor. Berikut adalah daftar kebutuhan fungsional aplikasi Tukang Online yang dikelompokkan berdasarkan aktor.

Tabel 3.1 Kebutuhan Fungsional Sistem

ID	Deskripsi Kebutuhan
FR-01	Pengguna dapat melakukan registrasi akun dan login ke sistem dengan data yang sudah didaftarkan.
FR-02	Pengguna dapat melihat dan mengubah data profil pribadi (nama, foto, nomor telepon, alamat).
FR-03	Pengguna dapat melihat riwayat pemesanan/pekerjaan
FR-04	pelanggan dapat melakukan pemesanan dengan cara memilih tukang berdasarkan kategori jasa dan radius jarak dari lokasi pengguna menggunakan peta interaktif.

ID	Deskripsi Kebutuhan
FR-05	pelanggan dapat melihat detail profil tukang mencakup nama, kategori jasa, foto profil, unggahan, deskripsi, rating, dan ulasan.
FR-06	pelanggan dapat memberikan rating dan ulasan terhadap tukang setelah pekerjaan selesai dilaksanakan.
FR-07	pelanggan dapat membatalkan pemesanan yang masih berstatus menunggu konfirmasi dari tukang.
FR-08	pelanggan dapat melihat terkait status pemesanan (menunggu, diterima, ditolak, selesai).
FR-09	Tukang dapat melakukan verifikasi akun dengan mengisi data pribadi, kategori jasa, deskripsi keahlian, foto KTP, file pendukung dan koordinat lokasi.
FR-10	Tukang dapat melihat daftar pemesanan masuk yang ditujukan kepadanya beserta detail pekerjaan yang diminta.
FR-11	Tukang dapat menerima atau menolak pemesanan jasa yang masuk disertai alasan penolakan jika diperlukan.
FR-12	Tukang dapat mengubah status pemesanan menjadi 'sedang dikerjakan' dan 'selesai' sesuai perkembangan pekerjaan.
FR-13	Tukang dapat melihat terkait pemesanan baru yang masuk ke akunnya.
FR-14	Tukang dapat mengunggah hasil kerja ke halaman profil pribadi
FR-15	Admin dapat melakukan login ke sistem
FR-16	Admin dapat memverifikasi atau menolak pendaftaran tukang berdasarkan kelengkapan dan keaslian data yang diunggah.
FR-17	Admin dapat melihat daftar seluruh tukang yang mendaftar beserta status verifikasi akun mereka.
FR-18	Admin dapat mengelola data pengguna dan tukang
FR-19	Admin dapat mengelola kategori jasa dan melihat ringkasan platform

3.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mendefinisikan atribut kualitas sistem yang harus dipenuhi, meliputi aspek performa, keamanan, kegunaan, ketersediaan, skalabilitas, dan kompatibilitas.

Tabel 3.2 Kebutuhan Non-Fungsional Sistem

ID	Kategori	Deskripsi Kebutuhan
NFR-01	Performa	Sistem dapat menampilkan hasil pencarian tukang dalam radius tertentu dalam waktu tidak lebih dari 5 detik pada kondisi jaringan normal.
NFR-02	Keamanan	Sistem menggunakan mekanisme autentikasi untuk memastikan hanya pengguna terdaftar dan terverifikasi yang dapat mengakses fitur tertentu.
NFR-03	Keamanan	Data sensitif seperti password pengguna disimpan dalam bentuk terenkripsi untuk menjaga keamanan data pengguna.
NFR-04	Kegunaan	Sistem memiliki antarmuka yang responsif dan mudah digunakan pada berbagai perangkat.
NFR-05	Skalabilitas	Arsitektur sistem menggunakan pemisahan frontend dan backend sehingga memudahkan pengembangan dan penambahan fitur di masa mendatang.
NFR-06	Kompatibilitas	Aplikasi dapat berjalan dengan baik pada browser modern seperti Google Chrome, Mozilla Firefox, Microsoft Edge, dan Safari versi terbaru.

3.2.3 Diagram Use Case

Diagram use case menggambarkan interaksi antara aktor dengan sistem secara keseluruhan. Aplikasi Tukang Online melibatkan tiga aktor utama, yaitu Pelanggan, Tukang, dan Admin



Gambar 3.2 Use Case Diagram

3.2.4 Skenario *Use Case*

Berikut adalah skenario dari beberapa *Use Case* yang telah di sajikan pada diagram *Use Case*.

a. Skenario *Register*

Tabel 3.3 Skenario Register

Kode <i>Use Case</i>	UC001
Nama <i>Use Case</i>	Login
Aktor Utama	Pelanggan, Tukang
Deskripsi	Pada Skenario ini, Aktor membuat akun baru pada sistem.
Kondisi Awal	Aktor berada pada halaman registrasi..
Kondisi Akhir	Data akun berhasil tersimpan dalam sistem.
Alur kejadian Normal	
Aktor	Sistem
• Membuka halaman registrasi.	• Menampilkan form registrasi.
• Mengisi data diri.	• Memvalidasi data yang dimasukkan.
• Menekan tombol Daftar.	• Menyimpan data pengguna.
	• Menampilkan pesan registrasi berhasil.

b. Skenarion *Login*

Tabel 3.4 Skenario Login

Kode <i>Use Case</i>	UC002
Nama <i>Use Case</i>	Login
Aktor Utama	Admin, Pelanggan, Tukang
Deskripsi	Pada Skenario ini, Aktor akan mencoba untuk masuk ke dalam sistem menggunakan akun terdaftar.

Kondisi Awal	Aktor sedang berada di halaman login.
Kondisi Akhir	Aktor berhasil login dan diarahkan ke halaman <i>Dashboard</i> sesuai dengan hak aksesnya.
Alur kejadian Normal	
Aktor	Sistem
Aktor memasukkan <i>email</i> dan <i>password</i> ke dalam input <i>field</i> yang telah di sediakan.	
	Sistem menampilkan <i>password</i> yang telah di <i>masking</i>.
Aktor menekan tombol <i>login</i>	
	Sistem memvalidasi <i>email</i> dan <i>password</i> dengan data dari tabel <i>users</i>.
	Sistem berhasil memvalidasi, membuat sesi, dan menampilkan halaman <i>dashboard</i> yang sesuai dengan Aktor.

c. Skenario Verifikasi Tukang

Tabel 3.5 Verifikasi tukang

Kode <i>Use Case</i>	UC003
Nama <i>Use Case</i>	Verifikasi tukang
Aktor Utama	Admin
Deskripsi	Admin melakukan verifikasi data pendaftaran tukang sebelum akun dapat digunakan untuk menerima pesanan dari pelanggan.

Kondisi Awal	Tukang telah melakukan registrasi dan mengunggah dokumen persyaratan.
Kondisi Akhir	Status akun tukang berubah menjadi Terverifikasi atau Ditolak.
Alur kejadian Normal	
Aktor	Sistem
• Membuka halaman verifikasi tukang.	• Menampilkan daftar tukang yang menunggu verifikasi.
• Memilih data tukang yang akan diverifikasi.	• Menampilkan detail data tukang.
• Memeriksa data yang diajukan.	• Menampilkan informasi profil dan dokumen pendukung.
• Menekan tombol Verifikasi.	• Mengubah status akun menjadi Terverifikasi.
	• Mengirim notifikasi kepada tukang bahwa akun telah diverifikasi.

d. Skenario Pencarian Tukang Berdasarkan Lokasi

Tabel 3.6 Skenario Pencarian Tukang Berdasarkan Lokasi

Kode <i>Use Case</i>	UC004
Nama <i>Use Case</i>	Pencarian Tukang Berdasarkan Lokasi
Aktor Utama	Pelanggan
Deskripsi	Pelanggan mencari tukang terdekat berdasarkan lokasi saat ini.
Kondisi Awal	Pelanggan telah login, dan berada di halaman utama.
Kondisi Akhir	Sistem menampilkan data para tukang yang berada di dalam radius yang ditentukan

Alur kejadian Normal	
Aktor	Sistem
• Membuka halaman pencarian tukang.	• Mengambil lokasi pelanggan.
• Memilih kategori jasa.	• Memproses pencarian berdasarkan lokasi.
	• Menghitung jarak menggunakan PostGIS.
	• Menampilkan daftar tukang terdekat pada peta dan daftar hasil pencarian.

e. Skenario Membuat Pesanan

Tabel 3.7 Skenario Membuat pesanan

Kode <i>Use Case</i>	UC005
Nama <i>Use Case</i>	Membuat pesanan
Aktor Utama	Pelanggan
Deskripsi	Pelanggan melakukan pemesanan jasa kepada tukang yang dipilih.
Kondisi Awal	Pelanggan telah login dan memilih tukang.
Kondisi Akhir	Pesanan berhasil dibuat dan menunggu konfirmasi tukang.
Alur kejadian Normal	
Aktor	Sistem
• Pelanggan memilih tukang	• Menampilkan data tukang
• Mengisi detail pekerjaan.	• Menampilkan formulir pemesanan.
• Menekan tombol Pesan.	• Menyimpan data pesanan.

	• Menampilkan daftar tukang terdekat pada peta dan daftar hasil pencarian.
--	--

f. Skenario Mengelola Pesanan

Tabel 3.8 Skenario Mengelola Pesanan

Kode <i>Use Case</i>	UC006
Nama <i>Use Case</i>	Mengelola Pesanan
Aktor Utama	Tukang
Deskripsi	Tukang menerima atau menolak pesanan yang masuk.
Kondisi Awal	Tukang telah login dan memiliki pesanan masuk.
Kondisi Akhir	Status pesanan diperbarui.
Alur kejadian Normal	
Aktor	Sistem
• Membuka daftar pesanan.	• Menampilkan daftar pesanan masuk.
• Memilih salah satu pesanan.	• Menampilkan detail pesanan.
• Menekan tombol Terima atau Tolak.	• Memperbarui status pesanan.
	• Mengirim informasi status kepada pelanggan.

g. Skenario Menyelesaikan Pesanan

Tabel 3.9 Skenario Menyelesaikan Pesanan

Kode <i>Use Case</i>	UC007
Nama <i>Use Case</i>	Menyelesaikan Pesanan
Aktor Utama	Tukang
Deskripsi	Tukang menandai pekerjaan sebagai selesai.
Kondisi Awal	Status pesanan sedang dikerjakan.
Kondisi Akhir	Status pesanan berubah menjadi selesai.
Alur kejadian Normal	
Aktor	Sistem
• Membuka detail pesanan.	• Menampilkan detail pesanan.
• Menekan tombol Selesaikan Pesanan.	• Memvalidasi status pesanan.
	• Mengubah status menjadi Selesai.
	• Memberikan akses <i>rating</i> kepada pelanggan.

h. Skenario Memberikan *Rating*

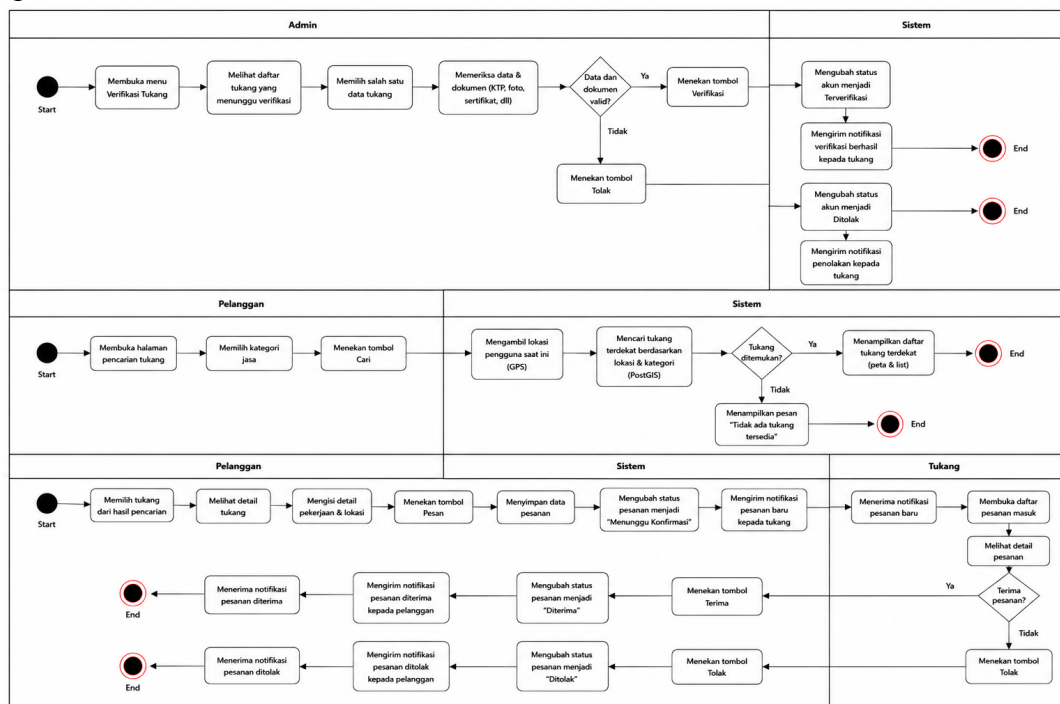
Tabel 3.10 Skenario Memberikan *Rating*

Kode <i>Use Case</i>	UC008
Nama <i>Use Case</i>	Memberikan <i>Rating</i>
Aktor Utama	Pelanggan
Deskripsi	Pelanggan memberikan penilaian terhadap layanan tukang yang telah selesai.
Kondisi Awal	Status pesanan telah selesai.

Kondisi Akhir	Rating dan ulasan tersimpan pada sistem.
Alur kejadian Normal	
Aktor	Sistem
• Membuka riwayat pesanan.	• Menampilkan daftar pesanan selesai.
• Memilih pesanan.	• Menampilkan form rating. .
• Memberikan rating dan ulasan.	• Menyimpan rating dan ulasan.
• Menekan tombol Kirim.	• Menampilkan pesan berhasil.

3.2.5 Activity Diagram

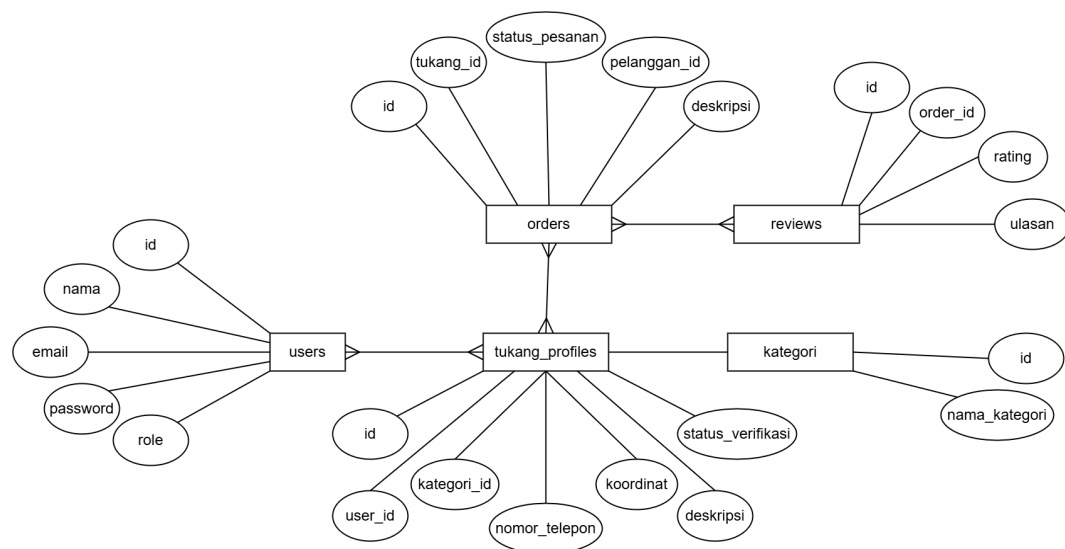
Activity diagram ini menunjukkan proses pemesanan layanan oleh pelanggan dan pengelolaan pesanan oleh tukang. Pelanggan memilih tukang dan mengirimkan pesanan, kemudian tukang menerima atau menolak pesanan tersebut. Jika pesanan diterima, proses pekerjaan dapat dilanjutkan hingga status pesanan selesai.



Gambar 3.3 Activity Diagram

3.2.6 ER Diagram

Entity Relationship Diagram (ERD) menggambarkan struktur basis data aplikasi Tukang Online beserta relasi antar entitas yang terlibat. Database dirancang menggunakan PostgreSQL dengan ekstensi PostGIS untuk mendukung penyimpanan dan pemrosesan data geospasial. Berikut adalah penjelasan entitas-entitas utama yang terdapat dalam sistem.



Gambar 3.4 ER Diagram

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi

Tahap implementasi merupakan proses penerapan hasil perancangan sistem ke dalam bentuk aplikasi yang dapat digunakan oleh pengguna. Aplikasi Tukang Online ini dikembangkan menggunakan framework Next.js pada sisi frontend dan Node.js dengan Express.js pada sisi backend. Basis data yang digunakan adalah PostgreSQL yang dilengkapi dengan ekstensi PostGIS untuk mendukung pengolahan data geospasial dan pencarian lokasi berbasis jarak.

Sistem yang dikembangkan memiliki tiga jenis pengguna, yaitu Admin, Pelanggan, dan Tukang. Setiap pengguna memiliki hak akses dan fungsi yang berbeda sesuai dengan kebutuhan sistem. Admin bertugas untuk mengelola data pengguna serta melakukan verifikasi terhadap akun tukang yang mendaftar ke dalam sistem. Proses verifikasi dilakukan untuk memastikan bahwa tukang yang terdaftar merupakan penyedia jasa yang valid sehingga dapat meningkatkan kepercayaan pelanggan terhadap layanan yang tersedia.

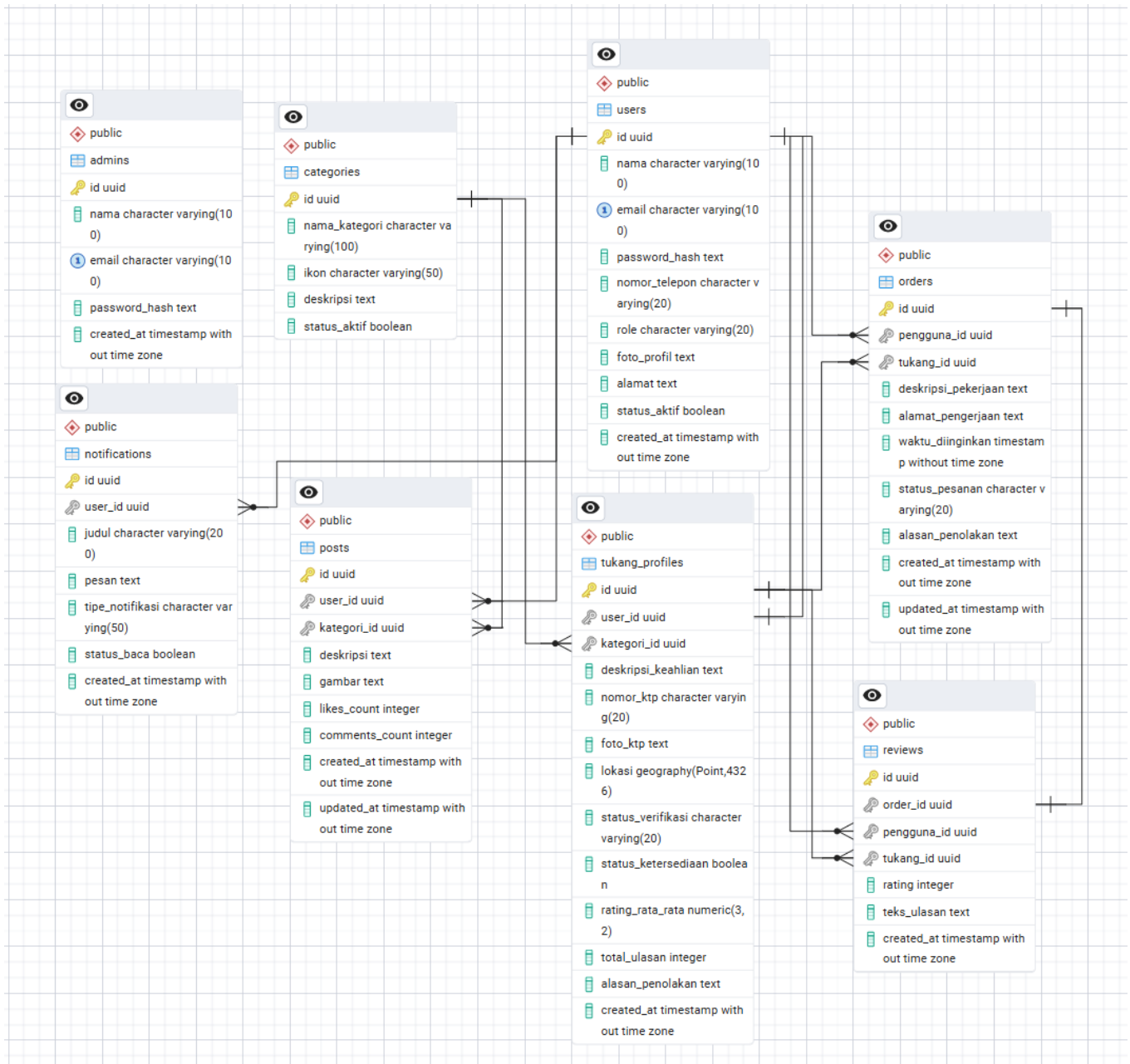
Pada sisi Pelanggan, sistem menyediakan fitur registrasi dan login, pencarian tukang berdasarkan lokasi, melihat informasi detail tukang, melakukan pemesanan jasa, serta memberikan penilaian terhadap layanan yang telah diterima. Fitur pencarian tukang memanfaatkan teknologi location Based Service (LBS) yang memungkinkan sistem menampilkan daftar tukang terdekat berdasarkan lokasi pengguna.

Sementara itu, pada sisi Tukang, sistem menyediakan fitur pengelolaan profil, penerimaan dan penolakan pesanan, serta pembaruan status pekerjaan. Melalui fitur tersebut, tukang dapat mengelola layanan yang diberikan kepada pelanggan secara lebih terstruktur dan efisien.

Implementasi sistem ini juga dilengkapi dengan antarmuka yang dirancang agar mudah digunakan (*user-friendly*) sehingga pengguna dapat mengoperasikan

aplikasi dengan lebih mudah. Selain itu, integrasi antara React, Node.js, PostgreSQL, dan PostGIS memungkinkan sistem berjalan secara responsif serta mampu mendukung proses pencarian tukang berdasarkan lokasi dengan baik.

4.1.1 Implementasi Basis Data



Gambar 4.1 implementasi ERD

4.1.2 Implementasi API

tabel 4.1 Implementasi API

No	Endpoint	Method	Fungsi / Kegunaan
1. AUTH ROUTES — Autentikasi User (/api/auth)			
1	POST /api/auth/register	POST	Mendaftarkan user baru (pelanggan atau tukang), menyimpan foto KTP untuk verifikasi identitas
2	POST /api/auth/login	POST	Login user dengan email & password, menghasilkan JWT token untuk session
3	GET /api/auth/profile	GET	Mendapatkan profil lengkap user yang sedang login (terautentikasi)
4	PUT /api/auth/profile	PUT	Mengupdate data profil user, termasuk foto profil (opsional)
2. TUKANG ROUTES — Manajemen Profil Tukang (/api/tukang)			
5	GET /api/tukang/search	GET	Mencari tukang berdasarkan kategori, nama, atau lokasi (publik, tanpa autentikasi)
6	GET /api/tukang/profile/me	GET	Mendapatkan profil lengkap tukang yang sedang login
7	PUT /api/tukang/profile	PUT	Mengupdate informasi profil tukang sendiri, termasuk foto profil
8	GET /api/tukang/:id	GET	Menampilkan detail profil tukang berdasarkan ID (publik)
9	GET /api/tukang/:id/reviews	GET	Mendapatkan semua ulasan & rating untuk tukang tertentu (publik)
10	PUT /api/tukang/:id	PUT	Mengupdate profil tukang berdasarkan ID (admin atau tukang sendiri)
3. POSTS ROUTES — Manajemen Posting / Portfolio (/api/posts)			
11	POST /api/posts	POST	Tukang membuat posting portfolio/showcase pekerjaan dengan upload gambar
12	GET /api/posts	GET	Menampilkan feed/galeri semua posting secara publik
13	GET /api/posts/my-posts	GET	Mendapatkan daftar posting milik tukang yang sedang login
14	GET /api/posts/:id	GET	Melihat detail satu posting berdasarkan ID (publik)
15	DELETE /api/posts/:id	DELETE	Menghapus posting dari portfolio (hanya owner/pemilik)
4. ORDER ROUTES — Manajemen Pesanan (/api/orders)			
16	POST /api/orders	POST	Pelanggan membuat pesanan baru ke tukang (berisi deskripsi, lokasi, anggaran)
17	GET /api/orders/my	GET	Pelanggan/tukang melihat daftar pesanan dan riwayat transaksi mereka
18	PUT /api/orders/:id/status	PUT	Update status pesanan: pending → accepted → in-progress → completed
19	PUT /api/orders/:id/cancel	PUT	Membatalkan pesanan oleh pelanggan atau tukang

No	Endpoint	Method	Fungsi / Kegunaan
5. REVIEW ROUTES — Manajemen Ulasan (/api/reviews)			
20	POST /api/reviews	POST	Pelanggan memberikan rating (1–5) dan komentar ulasan untuk tukang
21	GET /api/reviews/tukang/:id	GET	Mendapatkan semua ulasan untuk tukang tertentu (publik, social proof)
22	PUT /api/reviews/:id	PUT	Mengedit ulasan yang sudah diberikan (rating atau komentar)
6. NOTIFICATION ROUTES — Manajemen Notifikasi (/api/notifications)			
23	GET /api/notifications	GET	Menampilkan semua notifikasi milik user yang login (order, review, dll)
24	PUT /api/notifications/:id/read	PUT	Menandai notifikasi sebagai sudah dibaca (ubah status read/unread)
25	POST /api/notifications	POST	Sistem membuat dan mengirim notifikasi otomatis kepada user tertentu
7. CATEGORY ROUTES — Manajemen Kategori (/api/categories)			
26	GET /api/categories	GET	Menampilkan semua kategori keahlian tukang (listrik, plumbing, dll) — publik
27	POST /api/categories	POST	Admin menambahkan kategori keahlian baru ke master data
8. ADMIN ROUTES — Manajemen Administrasi (/api/admin)			
28	POST /api/admin/login	POST	Login sebagai admin untuk mengakses dashboard manajemen sistem
29	GET /api/admin/tukang	GET	Admin melihat daftar tukang yang pending verifikasi (KYC/onboarding)
30	PUT /api/admin/tukang/:id/verify	PUT	Admin melakukan approve atau reject verifikasi identitas tukang baru
31	GET /api/admin/stats	GET	Dashboard admin: statistik total user, order, revenue, dan analytics
32	DELETE /api/admin/categories/:id	DELETE	Admin menghapus kategori dari master data sistem
33	GET /api/admin/categories	GET	Ambil daftar kategori
34	GET /api/admin/users	GET	Ambil semua user
35	PUT /api/admin/users/:id/status	PUT	Toggle aktif/nonaktif user

4.1.3 Implementasi Antar Muka

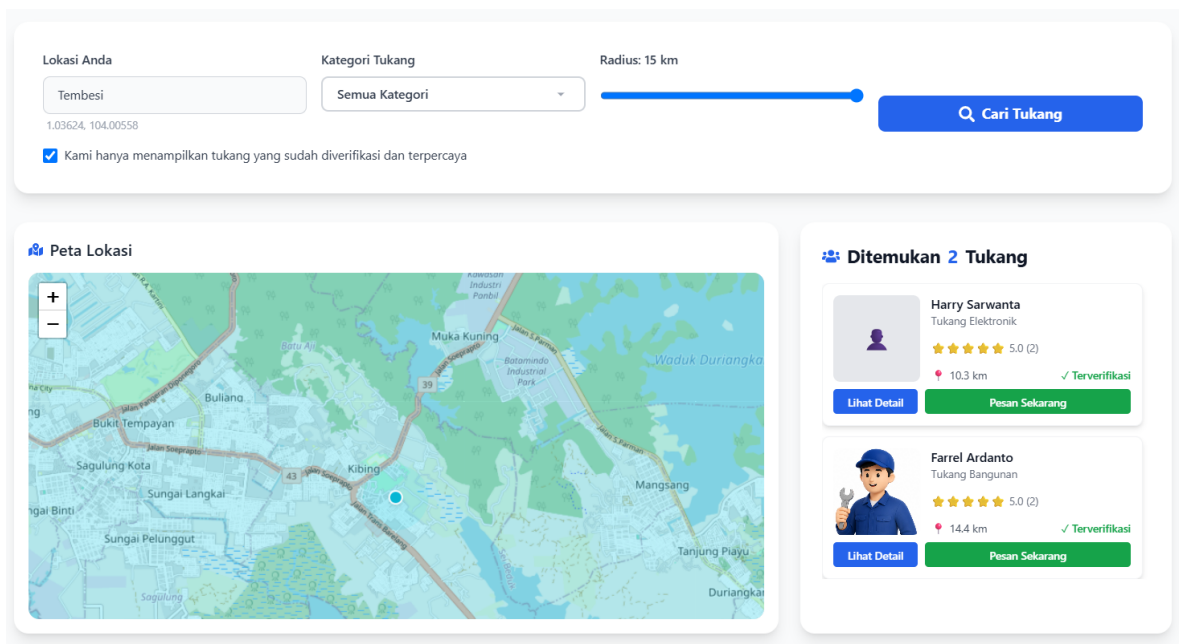
a. landing page



Gambar 4.2 Landing page

Gambar 4.2 menampilkan rancangan antarmuka halaman utama (*Landing Page*) yang menjadi tampilan awal saat pengguna pertama kali mengakses aplikasi. Halaman ini dirancang agar dapat diakses secara terbuka tanpa mewajibkan proses masuk (*login*) terlebih dahulu, sehingga pengguna umum dapat menjelajahi informasi awal sebelum memanfaatkan fitur utama sistem. Antarmuka ini mengimplementasikan elemen visual berupa gambar latar belakang bertema konstruksi yang dipadukan dengan teks pemikat (*hook*) "*Cari Tukang Terdekat*" untuk menegaskan fungsi utama aplikasi, serta dilengkapi dengan tombol akses masuk dan pendaftaran akun di sudut kanan atas untuk mengarahkan pengguna ke tahapan autentikasi.

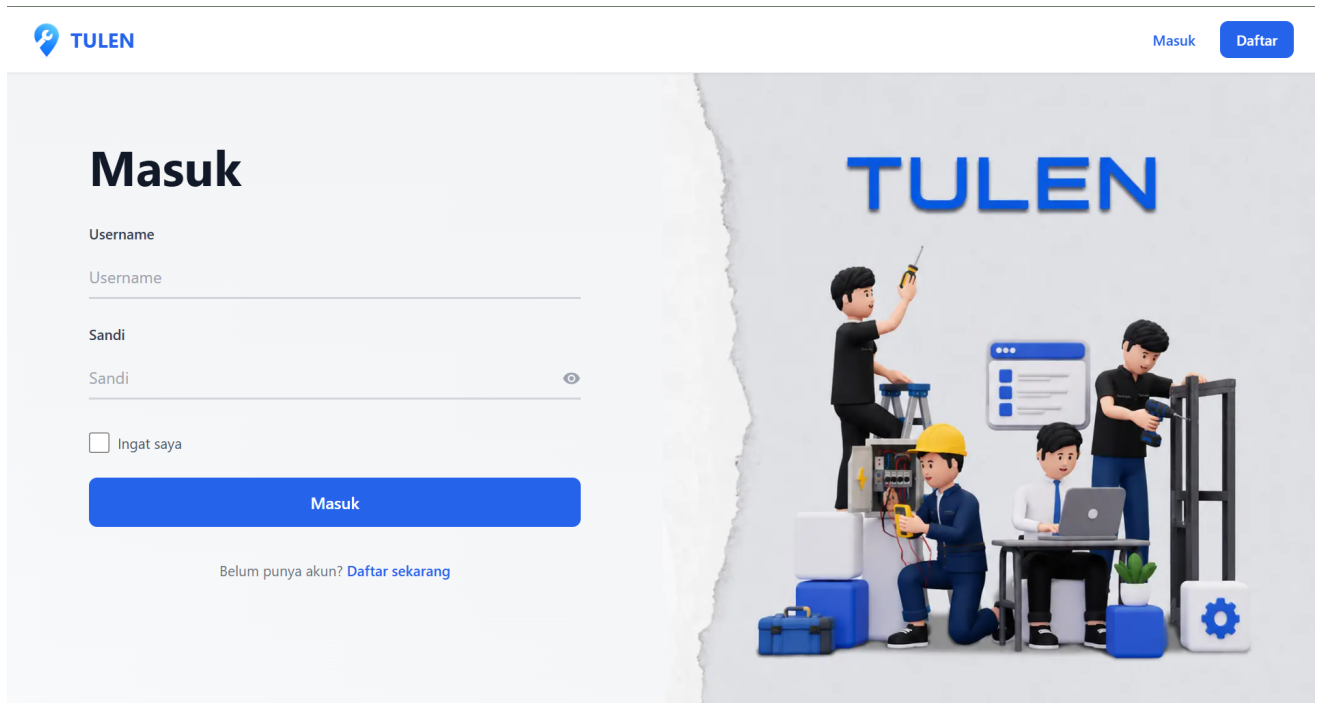
b. Fitur Utama



Gambar 4.3 Fitur Utama

Gambar 4.3 menampilkan rancangan antarmuka halaman utama fitur unggulan yang menjadi pusat interaksi pengguna saat menjelajahi aplikasi. Pada halaman ini, pengguna dapat melakukan pemindaian atau pencarian mitra tukang secara langsung berdasarkan radius lokasi geografis mereka secara *real-time*. Antarmuka ini dirancang secara dinamis sehingga pengguna cukup menggulir layar ke bawah untuk memuat daftar pekerja yang tersedia di sekitar posisi mereka, lengkap dengan informasi pemetaan lokasi serta tombol akses cepat untuk melihat detail profil dari masing-masing mitra tukang tersebut.

c. Halaman login



Gambar 4.4 Halaman *Login*

Gambar 4.4 menampilkan rancangan antarmuka halaman masuk (*Login Page*) yang berfungsi sebagai gerbang keamanan awal bagi pengguna dan mitra tukang sebelum dapat mengakses fitur utama di dalam aplikasi. Pada antarmuka ini, pengguna diwajibkan mengisi form input kredensial yang terdiri dari kolom *email* dan kata sandi (*password*) untuk divalidasi ke dalam basis data sistem. Selain tombol aksi utama untuk mengeksekusi proses masuk, halaman ini juga dilengkapi dengan tautan navigasi tambahan yang mengarahkan pengguna baru menuju halaman pendaftaran akun apabila mereka belum terdaftar di dalam sistem.

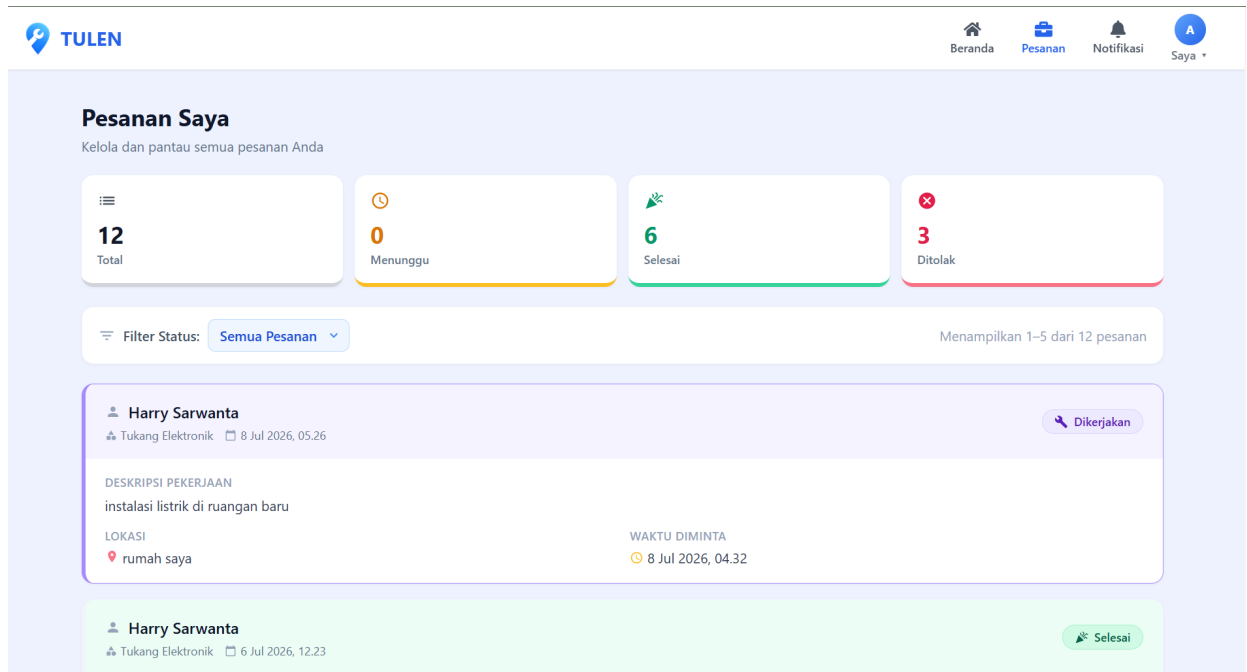
d. Halaman *Register*



Gambar 4.5. Halaman *Register*

Gambar 4.5 menampilkan rancangan antarmuka halaman pendaftaran akun (*Register*) yang berfungsi sebagai tahap awal klasifikasi peran pengguna di dalam sistem. Pada halaman ini, calon pengguna diberikan dua pilihan peran utama, yaitu sebagai konsumen yang membutuhkan jasa atau sebagai mitra tukang yang menyediakan keahlian konstruksi. Sistem dirancang untuk meminta pengisian data pribadi dasar pada kedua jenis peran tersebut, namun khusus untuk pengguna yang mendaftar sebagai mitra tukang, sistem menerapkan parameter keamanan tambahan berupa kewajiban mengunggah foto Kartu Tanda Penduduk (KTP) sebagai instrumen verifikasi dan validasi identitas resmi demi menjaga aspek keamanan dan kepercayaan di dalam platform.

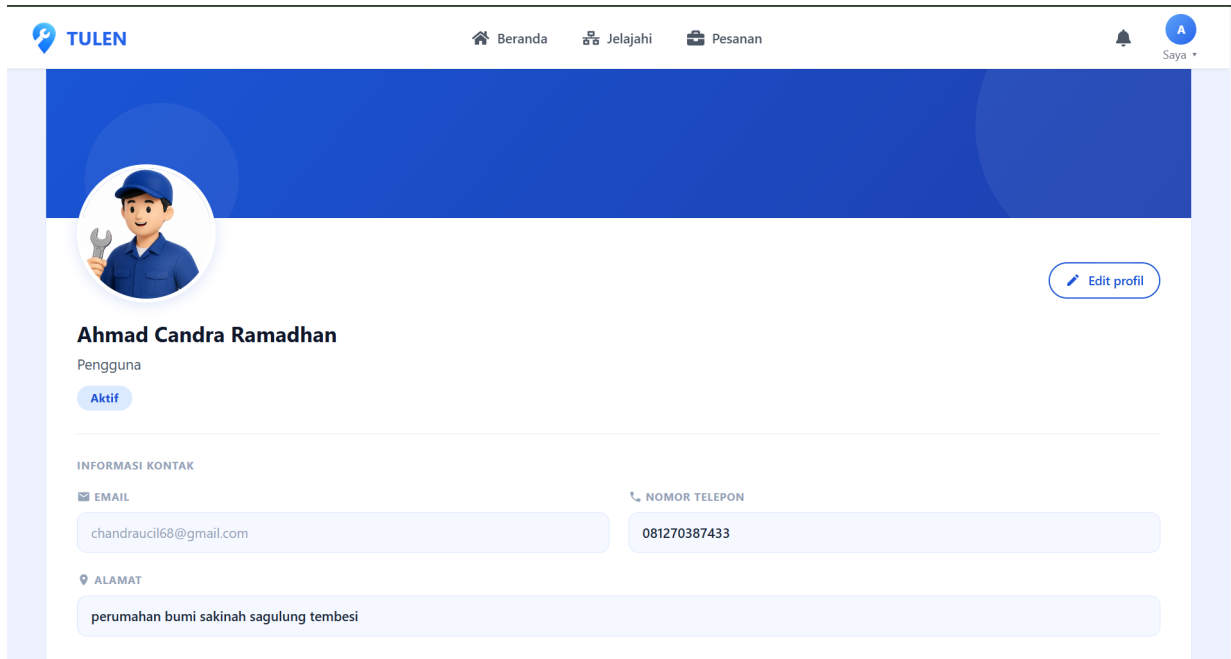
e. Halaman Pesanan



Gambar 4.6 Halaman Pesanan

Gambar 4.6 menampilkan rancangan antarmuka halaman pesanan (*Order Page*) yang berfungsi sebagai pusat pelacakan dan manajemen transaksi jasa bagi kedua belah pihak. Pada halaman ini, sistem menyajikan riwayat pemesanan secara terstruktur yang memuat informasi mendetail seperti nama mitra tukang yang dipilih, jenis keahlian atau kategori pekerjaan, serta status perkembangan pengerjaan yang sedang berjalan. Antarmuka ini dirancang untuk mempermudah pengguna dalam memantau alur transaksi secara berkala, memastikan transparansi proses kerja, serta menyediakan akses cepat menuju detail komunikasi atau pembayaran terkait pesanan tersebut.

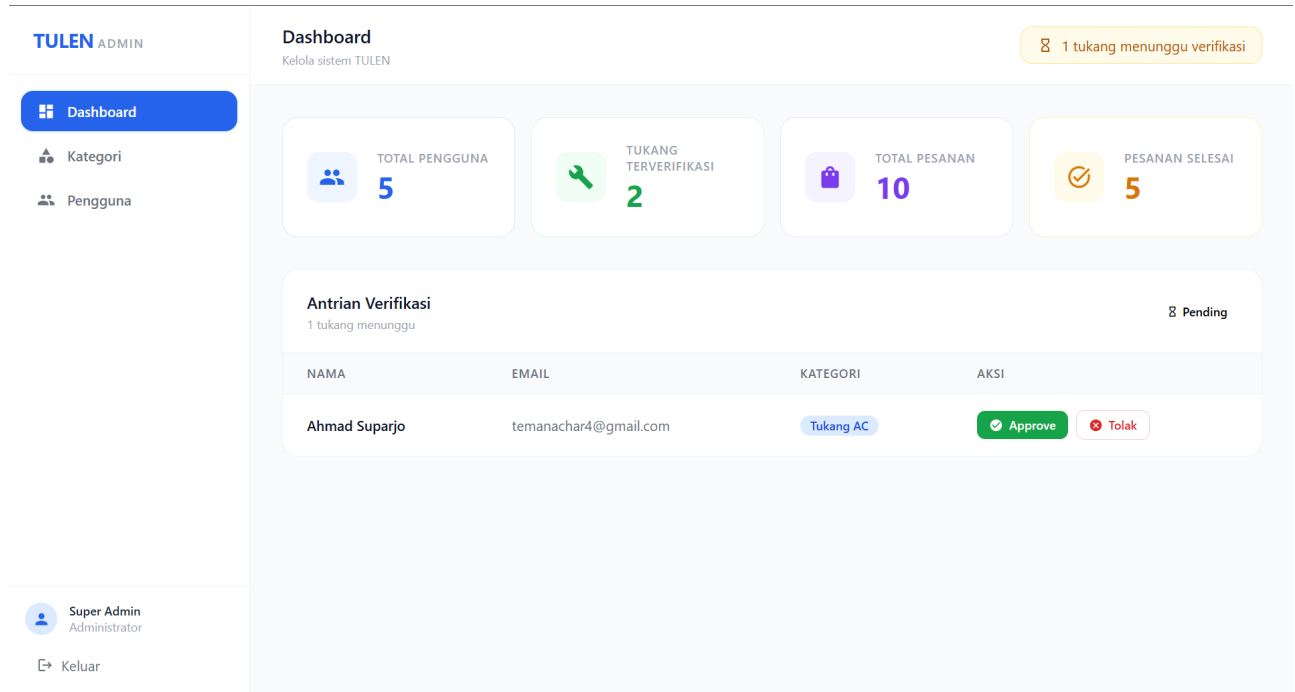
f. Halaman Profil



Gambar 4.7 Halaman Profil

Gambar 4.7 menampilkan rancangan antarmuka halaman profil (*Profile Page*) yang berfungsi untuk menyajikan seluruh data pribadi dan informasi kontak pengguna yang tersimpan di dalam sistem. Pada halaman ini, pengguna dapat melihat ringkasan identitas terdaftar seperti nama lengkap, status keaktifan akun, alamat *email*, nomor telepon, hingga alamat domisili secara terstruktur. Antarmuka ini juga dilengkapi dengan fitur interaktif berupa tombol ubah profil (*Edit Profile*) yang memberikan hak akses kepada pengguna untuk memperbarui atau menyelaraskan kembali data pribadi mereka apabila terjadi perubahan informasi di kemudian hari.

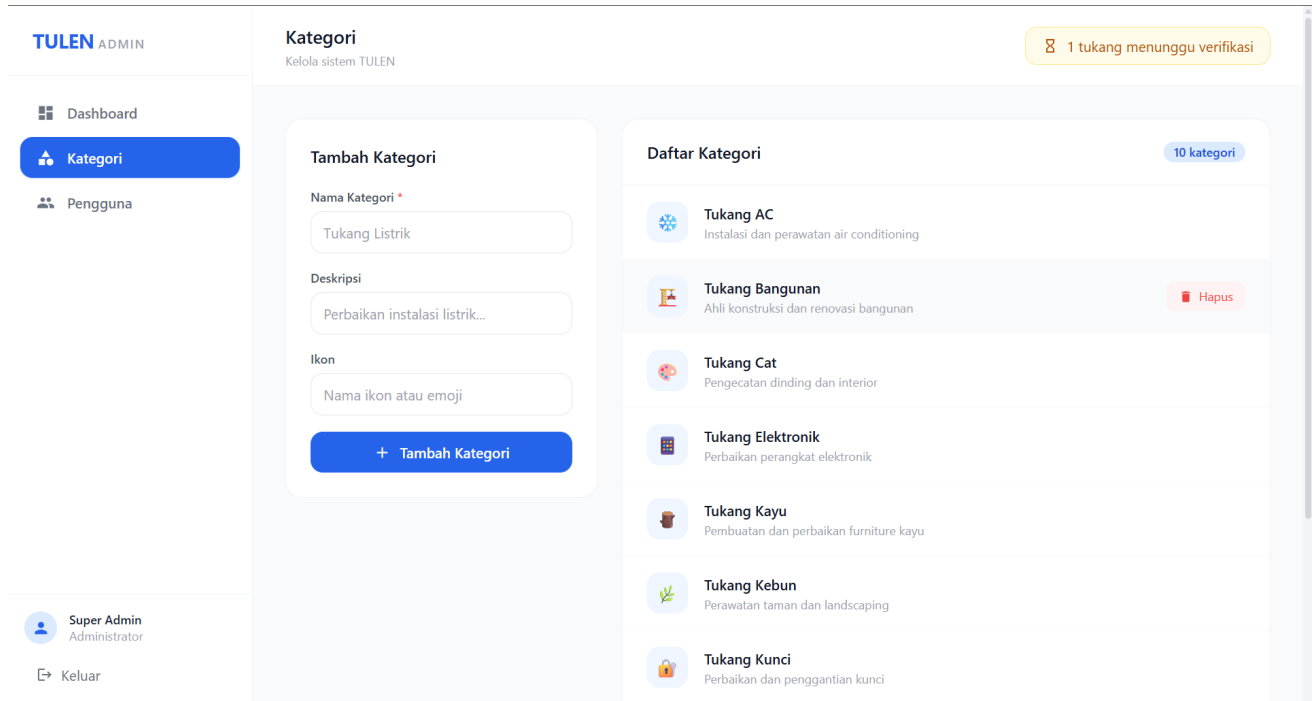
g. Halaman *Dashboard* Admin



Gambar 4.8 Halaman *Dashboard* Admin

Gambar 4.8 menampilkan rancangan antarmuka halaman panel kendali admin (*Dashboard Admin*) yang berfungsi sebagai pusat pengawasan dan pengelolaan seluruh aktivitas di dalam sistem. Pada halaman utama ini, sistem menyajikan ringkasan data statistik berupa jumlah total pengguna, mitra tukang yang terverifikasi, total pesanan, serta jumlah pesanan yang telah selesai. Selain itu, antarmuka ini mengimplementasikan tabel khusus bernama Antrian Verifikasi yang menampilkan daftar mitra tukang baru yang sedang menunggu proses validasi identitas oleh admin, lengkap dengan opsi aksi untuk menyetujui (*Approve*) atau menolak (*Tolak*) pendaftaran tersebut. Halaman ini juga dilengkapi dengan menu navigasi di sisi kiri yang mencakup menu Kategori untuk mengatur jenis keahlian jasa serta menu Pengguna untuk mengelola data akun yang terdaftar di dalam aplikasi.

h. Halaman Kategori



Gambar 4.9 Halaman Kategori

Gambar 4.9 menampilkan rancangan antarmuka halaman manajemen kategori yang berfungsi sebagai wadah bagi admin untuk mengelola klasifikasi jenis keahlian mitra tukang di dalam sistem. Antarmuka ini terbagi menjadi dua bagian utama, yaitu formulir "*Tambah Kategori*" di sisi kiri yang dilengkapi kolom input nama kategori, deskripsi singkat, serta ikon pendukung untuk mendaftarkan kategori baru ke dalam basis data. Sementara itu, pada sisi kanan terdapat komponen "*Daftar Kategori*" yang menyajikan seluruh data kategori aktif seperti Tukang AC, Tukang Bangunan, dan Tukang Cat, lengkap dengan tombol aksi "*Hapus*" untuk mengeliminasi kategori yang sudah tidak diperlukan lagi oleh sistem.

i. Halaman Kelola Pengguna

TULEN ADMIN

Pengguna
Kelola sistem TULEN

1 tukang menunggu verifikasi

Daftar Pengguna
5 pengguna terdaftar

Cari nama atau email...

NAMA	EMAIL	ROLE	STATUS	AKSI
Ahmad Suparjo	temanachar4@gmail.com	Tukang	Aktif	Nonaktifkan
Harry Sarwanta	chandraucil29@gmail.com	Tukang	Aktif	Nonaktifkan
ahmad candra ramadhan	chandraucil68@gmail.com	Pengguna	Aktif	Nonaktifkan
cancan	cancan@gmail.com	Pengguna	Aktif	Nonaktifkan
Farrel Ardanto	tukang2@gmail.com	Tukang	Aktif	Nonaktifkan

Super Admin
Administrator

Keluar

Gambar 4.10 Kelola Pengguna

Gambar 4.10 menampilkan rancangan antarmuka halaman manajemen pengguna (*User Management*) yang berfungsi sebagai pusat kendali admin dalam mengawasi seluruh akun yang terdaftar di dalam sistem, baik akun konsumen maupun akun mitra tukang. Halaman ini mengimplementasikan komponen tabel terstruktur yang menyajikan informasi detail meliputi nama pengguna, alamat *email*, peran (*role*), serta status keaktifan akun secara *real-time*. Selain sebagai media pemantauan, antarmuka ini juga dilengkapi dengan fitur keamanan interaktif berupa tombol aksi "*Nonaktifkan*" di sisi kanan setiap baris data, yang memberikan hak otoritas penuh kepada admin untuk membekukan atau menonaktifkan akun pengguna secara sepihak apabila terbukti melakukan pelanggaran terhadap ketentuan platform.

4.2. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi dan fitur yang telah diimplementasikan pada aplikasi ini dapat berjalan sesuai dengan kebutuhan fungsional yang telah dirancang sebelumnya. Metode pengujian yang digunakan adalah *Black-Box Testing*, yang berfokus pada pengujian antarmuka, validasi input, serta kesesuaian alur logika sistem tanpa memeriksa struktur kode internal perangkat lunak. Skenario pengujian (*test case*) dirancang secara terstruktur berdasarkan *Use Case Scenario* yang mencakup kondisi sukses (*positive testing*) maupun kondisi gagal (*negative testing*).

Tabel 4.2 Test Case Fungsional

Kode Test Case	Nama Test Case	Skenario	Langkah	Hasil yang Diharapkan
TC01	Register	Sukses (Konsumen)	Membuka halaman registrasi, memilih peran sebagai pelanggan, mengisi seluruh data diri dengan lengkap dan valid, lalu menekan tombol "Daftar".	Sistem berhasil memvalidasi data, menyimpan data akun ke basis data, dan menampilkan notifikasi sukses.
TC01	Register	Sukses (Mitra Tukang)	Membuka halaman registrasi, memilih peran sebagai tukang, mengisi data diri, mengunggah foto KTP yang valid, lalu	Sistem berhasil memvalidasi seluruh data beserta berkas KTP, menyimpan data ke basis data dengan status awal "Menunggu

			menekan tombol "Daftar".	Verifikasi", dan menampilkan notifikasi sukses.
TC01	Register	Gagal (Mitra Tukang Tanpa KTP)	Membuka halaman registrasi, memilih peran sebagai tukang, mengisi data diri tetapi mengosongkan unggahan foto KTP, lalu menekan tombol "Daftar".	Sistem menolak proses registrasi dan menampilkan pesan kesalahan bahwa dokumen KTP wajib diunggah.
TC02	Login	Sukses	Memasukkan email dan kata sandi yang sudah terdaftar dengan benar, lalu menekan tombol "Login".	Sistem berhasil memvalidasi kredensial, membuat sesi baru, dan mengarahkan pengguna ke halaman dashboard utama sesuai hak aksesnya.
TC02	Login	Gagal (Kredensial Salah)	Memasukkan email yang belum terdaftar atau memasukkan kata sandi yang salah, lalu menekan tombol "Login".	Sistem menampilkan pesan peringatan bahwa email atau kata sandi tidak cocok dan menolak akses masuk.

TC03	Verifikasi Tukang oleh Admin	Sukses (Persetujuan)	Admin masuk ke menu antrian verifikasi, memilih salah satu profil mitra tukang yang berstatus menunggu, memeriksa dokumen KTP, lalu menekan tombol "Approve".	Sistem mengubah status mitra tukang menjadi "Terverifikasi" di basis data dan mengirimkan notifikasi bahwa akun telah aktif.
TC03	Verifikasi Tukang oleh Admin	Sukses (Penolakan)	Admin memilih profil mitra tukang yang data KTP-nya tidak jelas/palsu, lalu menekan tombol "Tolak".	Sistem mengubah status akun menjadi "Ditolak" dan membatalkan hak akses akun tersebut untuk menerima pesanan.
TC04	Pencarian Tukang Berdasarkan Lokasi	Sukses	Konsumen membuka halaman utama, mengizinkan akses lokasi perangkat (geolocation), memilih salah satu kategori jasa tukang, lalu melakukan pencarian.	Sistem menangkap koordinat titik konsumen, mengeksekusi kueri PostGIS (ST_DWithin), lalu menampilkan daftar mitra tukang terdekat yang berada di dalam radius geografis pada peta secara real-time.

TC05	Membuat Pesanan	Sukses	Pelanggan masuk ke halaman profil tukang yang dipilih, mengisi detail formulir pemesanan pekerjaan dengan lengkap, lalu menekan tombol "Pesan".	Sistem berhasil memvalidasi input, menyimpan data transaksi baru ke basis data dengan status awal "Menunggu Konfirmasi", dan mengirimkan notifikasi pesanan masuk ke pihak tukang.
TC05	Membuat Pesanan	Gagal (Form Kosong)	Pelanggan langsung menekan tombol "Pesan" tanpa mengisi detail deskripsi pekerjaan pada formulir.	Sistem menolak proses pemesanan dan memunculkan pesan peringatan bahwa formulir detail pekerjaan wajib diisi.
TC06	Mengelola Pesanan oleh Tukang	Sukses (Menerima Pesanan)	Tukang membuka daftar pesanan masuk, memilih salah satu pesanan, lalu menekan tombol "Terima".	Sistem memperbarui status pesanan menjadi "Sedang Dikerjakan" di basis data dan mengirimkan notifikasi perubahan status kepada pelanggan.
TC06	Mengelola Pesanan oleh Tukang	Sukses (Menolak Pesanan)	Tukang memilih salah satu pesanan masuk yang tidak	Sistem mengubah status pesanan menjadi

			sesuai kapasitasnya, lalu menekan tombol "Tolak".	"Ditolak" dan memperbarui riwayat pesanan di sisi pelanggan.
TC07	Menyelesaikan Pesanan	Sukses	Tukang membuka detail pesanan yang statusnya sedang dikerjakan, kemudian menekan tombol "Selesaikan Pesanan".	Sistem mengubah status transaksi menjadi "Selesai" di basis data, serta secara otomatis membuka/mengaktifkan akses pengisian rating pada halaman pelanggan.
TC08	Memberikan Rating dan Ulasan	Sukses	Pelanggan membuka riwayat pesanan yang sudah selesai, memilih bintang rating, mengetik ulasan performa tukang, lalu menekan tombol "Kirim".	Sistem berhasil menyimpan data nilai rating dan teks ulasan ke dalam basis data, mengakumulasi ulang akumulasi rating si tukang, dan menampilkan pesan sukses.
TC09	Kelola Profil	Sukses	Pengguna membuka halaman profil, mengubah data (nama/foto/nomor telepon/alamat), lalu menekan tombol "Simpan".	Sistem memperbarui data profil di basis data dan menampilkan pesan sukses.

Berdasarkan hasil eksekusi dari seluruh skenario pengujian yang telah dipetakan di atas, sistem menunjukkan tingkat keberhasilan yang optimal di mana setiap fungsi utama mulai dari proses autentikasi, manajemen pengguna, hingga kueri spasial menggunakan PostGIS dapat merespon input pengguna sesuai dengan hasil yang diharapkan. Dengan demikian, aplikasi ini dinyatakan memenuhi standar fungsionalitas dan siap untuk diimplementasikan ke tahap berikutnya.

Merujuk pada tujuan pengembangan yang telah ditetapkan pada Bab I, hasil pengujian di atas menunjukkan bahwa tujuan 1 (pencarian tukang berbasis radius menggunakan PostGIS), tujuan 2 (registrasi dan manajemen profil tukang), dan tujuan 4 (implementasi teknologi geospasial PostgreSQL/PostGIS) telah tercapai sepenuhnya, sebagaimana ditunjukkan oleh TC01, TC03, dan TC04. Tujuan 3 (sistem pemesanan dan penilaian layanan yang transparan dan terverifikasi) tercapai sebagian — fitur pemesanan, pengelolaan status, dan rating/ulasan telah berjalan (TC05–TC08), namun mekanisme transparansi harga dan negosiasi antara pengguna jasa dan tukang belum tersedia pada versi prototipe ini dan menjadi catatan pengembangan lanjutan.

Tabel 4.3 Test Case Non Fungsional

Kode Test Case	Nama Test Case	Skenario	Hasil yang Diharapkan
TC-NF R-01	Performa	Melakukan pencarian tukang berbasis radius dan mengukur waktu respons sistem	Hasil pencarian tampil dalam waktu wajar (< 5 detik)
TC-NF R-02	Keamanan (autentikasi)	Mendaftar menggunakan alamat email yang tidak valid/dummy	Sistem menolak pendaftaran dan menampilkan pesan validasi email
TC-NF R-03	Keamanan (data sensitif)	Memeriksa data password pengguna di basis data	Password tersimpan dalam bentuk terenkripsi (hash), bukan teks polos
TC-NF R-04	Kegunaan	Mengakses aplikasi melalui perangkat dengan ukuran layar berbeda (desktop, tablet, mobile)	Tampilan menyesuaikan (responsif) tanpa elemen yang terpotong
TC-NF R-05	Skalabilitas	Memeriksa pemisahan proses request antara frontend dan backend (API terpisah)	Frontend dan backend dapat berjalan dan diskalakan secara independen
TC-NF R-06	Kompatibilitas	Mengakses aplikasi melalui beberapa browser berbeda (Chrome, Firefox)	Aplikasi tampil dan berfungsi dengan baik di seluruh browser yang diuji

Selain pengujian fungsional, pengujian terhadap kebutuhan non-fungsional (Tabel 4.x) menunjukkan bahwa seluruh NFR telah terpenuhi, termasuk NFR-02 setelah penambahan mekanisme verifikasi email pada proses registrasi.

BAB V KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil menghasilkan aplikasi Tukang Online berbasis web yang dibangun menggunakan Next.js pada sisi frontend, Node.js dan Express.js pada sisi backend, serta PostgreSQL dengan ekstensi PostGIS sebagai basis data geospasial.
2. Fitur Geolocation Based Service (GBS) berhasil diimplementasikan pada aplikasi untuk membantu pelanggan menemukan tukang berdasarkan lokasi terdekat. Sistem mampu menampilkan daftar tukang sesuai posisi pengguna sehingga proses pencarian jasa menjadi lebih efektif dan efisien.
3. Aplikasi menyediakan fitur utama yang meliputi registrasi pengguna, verifikasi akun tukang, pencarian tukang berdasarkan lokasi, pemesanan jasa, pengelolaan pesanan, serta pemberian rating terhadap layanan yang telah diberikan.
4. Pengujian sistem menggunakan metode Black Box Testing menunjukkan bahwa seluruh fungsi utama aplikasi berjalan sesuai dengan kebutuhan yang telah ditentukan dan mampu memberikan hasil yang sesuai dengan skenario pengujian.
5. TULEN berhasil menjawab dua kebutuhan yang saling berkaitan sebagai marketplace dua sisi: memperluas jangkauan promosi dan peluang kerja bagi tukang, serta mempermudah dan mempercepat proses pencarian tukang kompeten dan terpercaya bagi pengguna jasa, menggantikan cara konvensional (rekomendasi mulut ke mulut atau papan iklan di tiang) yang selama ini kurang efisien.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan untuk pengembangan sistem pada penelitian selanjutnya, yaitu:

1. Menambahkan fitur komunikasi langsung (chat) antara pelanggan dan tukang untuk memudahkan proses koordinasi pekerjaan.
2. Menambahkan fitur pelacakan lokasi tukang secara real-time sehingga pelanggan dapat mengetahui posisi tukang saat menuju lokasi pekerjaan.
3. Mengembangkan aplikasi dalam bentuk mobile application berbasis Android maupun iOS agar lebih mudah diakses oleh pengguna.
4. Menambahkan sistem pembayaran digital yang terintegrasi dengan payment gateway untuk mendukung proses transaksi secara online.
5. Meningkatkan sistem keamanan dan verifikasi identitas pengguna agar tingkat kepercayaan terhadap layanan yang tersedia menjadi lebih baik.

DAFTAR PUSTAKA

- Agafonkin, V. (2022). Leaflet: An open-source JavaScript library for mobile-friendly interactive maps. Leaflet Contributors. <https://leafletjs.com>
- Fajar, R., & Ramadan, D. (2021). Aplikasi pencarian jasa servis elektronik berbasis Android menggunakan GPS. *Jurnal Informatika dan Rekayasa Elektronik*, 4(1), 45–54. <https://doi.org/10.36595/jire.v4i1.234>
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Doctoral dissertation). University of California, Irvine.
- Hidayat, A., Nurrahman, M., & Pratama, R. (2022). Pengembangan aplikasi e-commerce berbasis microservices untuk peningkatan skalabilitas sistem. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 9(4), 789–798.
- Kurniawan, D., & Susanti, E. (2023). Evaluasi performa framework JavaScript server-side pada lingkungan cloud computing. *Jurnal Sistem Informasi dan Teknologi*, 11(1), 45–54.
- Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic information science and systems* (4th ed.). John Wiley & Sons.
- O'Brien, J. A., & Marakas, G. M. (2011). *Management information systems* (10th ed.). McGraw-Hill/Irwin.
- Pratiwi, E., & Nugroho, A. (2022). Implementasi sistem informasi geografis berbasis web untuk pemetaan fasilitas publik. *Jurnal Edukasi dan Penelitian Informatika*, 8(3), 301–309.
- Reese, G. (2000). *Database programming with JDBC and Java* (2nd ed.). O'Reilly Media.

- Santoso, B., Wijaya, K., & Handoko, T. (2023). Optimasi integrasi data relasional dan non-relasional dalam arsitektur web modern. *Jurnal Informatika dan Komputer*, 15(2), 134–143.
- Schiller, J., & Voisard, A. (2004). *Location-based services*. Morgan Kaufmann Publishers..
- Stonebraker, M., & Rowe, L. A. (1986). The design of POSTGRES. *ACM SIGMOD Record*, 15(2), 340–355. <https://doi.org/10.1145/16894.16888>
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80–83. <https://doi.org/10.1109/MIC.2010.145>
- Turban, E., Outland, J., King, D., Lee, J. K., Liang, T. P., & Turban, D. C. (2018). *Electronic commerce 2018: A managerial and social networks perspective* (9th ed.). Springer International Publishing. <https://doi.org/10.1007/978-3-319-58715-8>