

OPTIMALISASI LAYANAN FREELANCE MELALUI PENGEMBANGAN APLIKASI MARKETPLACE BERBASIS MOBILE

Joy Nasten Sinaga¹, Ricky Paian Limbong²

^{1,2}Universitas Mikroskil

(Program Studi Teknik Informatika Fakultas Ilmu Informatika Universitas Mikroskil)

Jl MH Thamrin No 140, (061) 4573767/ (061) 4567789

e-mail: ¹joy.sinaga@mikroskil.ac.id, ²ricky.limbong@mikroskil.ac.id

Abstrak

Marketplace Freelance Service adalah platform yang menghubungkan freelancer sebagai penyedia jasa dengan klien sebagai pencari jasa, memungkinkan transaksi layanan secara online. Namun, terdapat tantangan dalam proses pembayaran honor yang kurang efisien, di mana freelancer sering kali tidak menerima honor tepat waktu, sementara klien merasa tidak nyaman membayar penuh di awal kontrak. Selain itu, freelancer tidak memiliki bukti otentik berupa Laporan Penghasilan yang merinci pendapatan mereka, yang penting sebagai referensi melamar pekerjaan atau untuk negosiasi gaji.

Penelitian ini bertujuan mengembangkan Aplikasi Mobile Marketplace Freelance Service sebagai solusi atas permasalahan tersebut. Aplikasi ini dikembangkan menggunakan metode Rapid Application Development (RAD) dengan pemodelan berbasis React Native dan Node.js untuk aplikasi mobile, serta React.js dan Node.js untuk aplikasi back-end. Database dikelola menggunakan MySQL. Aplikasi yang dihasilkan mendukung tiga jenis pengguna: freelancer, klien, dan admin, sehingga menciptakan ekosistem layanan yang lebih efisien dan transparan.

Kata kunci: Marketplace, Aplikasi Mobile, Layanan Freelance, RAD, React

Abstract

The Freelance Service Marketplace is a platform that connects freelancers as service providers with clients as service seekers, enabling online service transactions. However, challenges arise in the payment process, where freelancers often face delays in receiving compensation, while clients are uncomfortable paying the full amount upfront. Additionally, freelancers lack authentic Income Reports that detail their earnings, which are essential for job applications and salary negotiations.

This study aims to develop a Mobile Marketplace Freelance Service Application to address these issues. The application is developed using the Rapid Application Development (RAD) method, with React Native and Node.js for mobile development, React.js and Node.js for the back-end, and MySQL as the database management system. The resulting application supports three types of users: freelancers, clients, and admins, creating a more efficient and transparent service ecosystem.

Keywords: Marketplace, Mobile Application, Freelance Service, RAD, React

1. PENDAHULUAN

Perkembangan teknologi saat ini semakin pesat, yang dapat terlihat dari meningkatnya jumlah pengguna teknologi di berbagai kalangan. Hal ini dapat dilihat dari semakin banyaknya aplikasi yang dipasang pada *gadget* atau *smartphone* yang memiliki berbagai fungsi dan manfaat, salah satunya adalah untuk mencari, mendapatkan, dan membagikan informasi dengan mudah[1]. Aplikasi-aplikasi ini memberikan kemudahan bagi penggunaannya untuk mengakses berbagai jenis informasi sesuai dengan kebutuhan, dan mendukung aktivitas mereka, baik dalam aspek pribadi, sosial, maupun profesional.

Freelancer adalah salah satu profesi yang banyak diminati oleh masyarakat saat ini. Pekerjaan *freelance* merujuk pada individu yang bekerja berdasarkan hubungan kerja harian, di mana upahnya dihitung berdasarkan jumlah hari kerja, jam kerja, atau volume pekerjaan yang diselesaikan[2]. Dengan kata lain, pekerja harian lepas tidak memiliki kewajiban untuk bekerja secara rutin atau tetap dan tidak memperoleh hak yang setara dengan pekerja tetap. Orang yang menjalani pekerjaan ini disebut

freelancer [3]. Salah satu ciri khas dari pekerjaan *freelance* adalah fleksibilitas waktu dan tempat kerjanya, yang memungkinkan seorang *freelancer* untuk bekerja kapan saja dan di mana saja. Selain itu, *freelancer* juga memiliki kebebasan untuk memilih proyek dan klien yang ingin diajak bekerja sama, serta dapat bergabung dengan berbagai pihak dalam menyelesaikan proyek-proyek tertentu. Dengan demikian, pekerjaan *freelance* tidak terikat oleh rutinitas kantor yang kaku. Menurut survei BPS (Badan Pusat Statistik) Republik Indonesia per Februari 2024, sekitar 7 juta (10,74%) dari 65.582.464 total pekerja di Indonesia saat ini bekerja di sektor Pekerja Bebas di Nonpertanian, yang mana banyak di antaranya berprofesi sebagai *freelancer* [4]. Profesi sebagai *freelancer* semakin diminati karena memberikan kebebasan dalam mengatur waktu dan tempat kerja, serta memungkinkan untuk mendapatkan penghasilan dari hobi yang menjadi pekerjaan.

Perkembangan teknologi yang pesat saat ini telah membuka berbagai peluang bagi para *freelancer* untuk bergabung dengan berbagai platform yang dapat digunakan untuk mencari pekerjaan. Berbagai platform digital ini menyediakan berbagai tawaran pekerjaan yang sesuai dengan keahlian dan minat *freelancer*, serta memudahkan para *freelancer* untuk memperoleh penghasilan dari berbagai jenis pekerjaan yang ditawarkan secara *online* [5], [6]. Namun, meskipun teknologi telah memudahkan akses ke berbagai peluang pekerjaan, dunia *freelance* juga menghadapi sejumlah permasalahan yang perlu diselesaikan agar lebih efisien dan transparan.

Salah satu masalah utama yang muncul adalah terkait dengan sistem pembayaran honor *freelancer* yang kurang efektif dan efisien. Proses pembayaran yang kurang tepat dapat menghambat hubungan antara *freelancer* dan klien. Banyak *freelancer* yang mengalami kendala dalam mendapatkan honor mereka tepat waktu, terutama apabila pembayaran dilakukan di akhir penyelesaian proyek, yang terkadang membuat *freelancer* tidak menerima pembayaran untuk pekerjaan yang telah mereka selesaikan [7]. Di sisi lain, beberapa klien juga merasa tidak nyaman apabila mereka diminta untuk membayar honor *freelancer* di awal pekerjaan, karena mereka khawatir pekerjaan yang dihasilkan tidak sesuai dengan harapan atau kesepakatan. Kekhawatiran ini dapat menyebabkan ketidakpastian bagi kedua belah pihak, baik *freelancer* maupun klien. Hal ini berujung pada pandangan masyarakat yang menganggap bahwa pekerjaan *freelance* bukanlah pilihan yang stabil atau menjanjikan. *Freelancer* pun sering merasa cemas karena keraguan apakah mereka akan dibayar dengan layak setelah pekerjaan selesai, sementara klien merasa tidak aman membayar di awal karena takut pekerjaan tidak akan sesuai dengan harapan.

Selain itu, masalah lain yang dihadapi adalah ketidaktersediaan laporan penghasilan yang dapat digunakan oleh *freelancer* sebagai bukti otentik dari pekerjaan yang telah mereka selesaikan. Laporan penghasilan yang rinci dan transparan sangat penting untuk *freelancer*, karena laporan ini dapat digunakan sebagai referensi untuk melamar pekerjaan baru atau untuk negosiasi gaji di masa depan [6], [8]. Tanpa adanya laporan yang jelas, *freelancer* kesulitan untuk menunjukkan besaran penghasilannya atau untuk memperkirakan pendapatan mereka dalam jangka waktu tertentu. Selain itu, kurangnya perhatian terhadap kategori pekerjaan non-digital dan low-skilled juga menjadi salah satu kendala yang membuat *freelancer* terbatas pada pekerjaan yang membutuhkan keahlian digital tertentu. Hal ini menyebabkan mereka yang tidak memiliki keahlian khusus di bidang teknologi mengalami kesulitan untuk menemukan pekerjaan yang sesuai dengan kemampuan mereka.

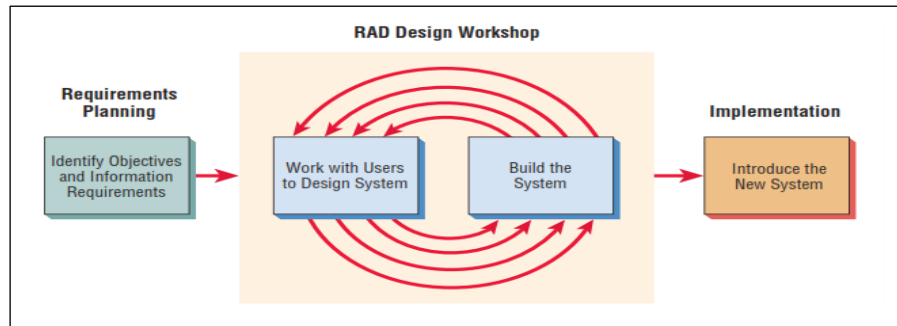
Oleh karena itu, untuk mengatasi permasalahan tersebut, dibutuhkan sistem yang dapat mengoptimalkan proses transaksi dan memberikan kenyamanan bagi *freelancer* dan klien. Sistem yang tepat dapat memastikan bahwa *freelancer* mendapatkan honor yang pantas sesuai dengan pekerjaan yang telah mereka lakukan, memberikan laporan penghasilan yang jelas, dan mempermudah klien untuk memantau perkembangan pekerjaan yang sedang dilakukan oleh *freelancer*. Dengan sistem yang lebih efisien dan transparan, diharapkan dapat meningkatkan kepercayaan dan kenyamanan bagi kedua belah pihak, serta memperluas peluang bagi mereka yang ingin bergabung dalam dunia *freelance*, baik yang memiliki keahlian digital maupun non-digital.

2. METODE PENELITIAN

Metodologi yang digunakan dalam pengembangan aplikasi ini mengacu pada pendekatan *Rapid Application Development* (RAD) yang ditampilkan pada Gambar 1 dengan tahapan-tahapan sebagai berikut [9]:

2.1. Perencanaan Kebutuhan Sistem (Requirement Planning)

Tahap ini merupakan tahap pertama dalam siklus pendekatan *Rapid Application Development* (RAD). Pada tahap ini, dilakukan identifikasi tujuan dan syarat-syarat informasi yang dibutuhkan untuk pengembangan aplikasi. Langkah pertama adalah menganalisis masalah yang ada untuk pengembangan aplikasi ini, dengan tujuan untuk memahami dengan jelas apa yang ingin dicapai melalui aplikasi. Selanjutnya, dilakukan identifikasi terhadap syarat-syarat yang diperlukan dalam pembuatan aplikasi mobile berbasis sistem marketplace. Untuk itu, dilakukan observasi terhadap aplikasi mobile sejenis, seperti *freelancer.com* dan *Fastwork.id*, guna memahami fitur dan fungsi yang dibutuhkan dalam aplikasi marketplace. Hasil dari observasi ini digunakan untuk menyusun Use Case Diagram yang menggambarkan kebutuhan fungsional dari sistem yang akan dibangun.



Gambar 1. Fase Rapid Application Development (RAD)

2.2. Rancangan Pengguna (User Design)

Pada tahap ini, dilakukan perancangan alur proses yang akan terjadi dalam sistem serta perancangan antarmuka (*interface*). Penulis membuat desain awal berdasarkan kebutuhan sistem yang telah ditentukan sebelumnya, menggunakan aplikasi Adobe XD CC. Desain ini akan mencakup tiga jenis pengguna utama dalam sistem, yaitu:

1. Desain untuk *admin*,
2. Desain untuk *freelancer*,
3. Desain untuk *client*.

Setelah desain antarmuka selesai, penulis juga merancang basis data yang akan digunakan untuk menyimpan data dalam aplikasi. Desain basis data ini dibuat untuk memastikan bahwa sistem dapat menyimpan dan mengelola data dengan efisien, serta mendukung operasional aplikasi yang direncanakan.

2.3. Implementasi (Implementation)

Pada tahap ini, pengembangan aplikasi mobile dilakukan berdasarkan desain yang telah dibuat pada tahap sebelumnya (*User Design*). Aplikasi *mobile* dikembangkan menggunakan bahasa pemrograman JavaScript dan *framework* React Native, yang memungkinkan aplikasi berjalan pada berbagai perangkat, khususnya perangkat Android[10]. Khusus untuk fitur admin dalam mengelola data pengguna *freelancer* dan *client* dikembangkan dalam bentuk aplikasi web dengan bantuan *framework* javascript React JS[11]. Untuk pengelolaan dan penyimpanan data, digunakan sistem manajemen basis data (DBMS) MySQL.

Pengujian sistem dilakukan pada berbagai komponen. Pengujian database dilakukan dengan menggunakan browser web Google Chrome untuk memverifikasi koneksi dan interaksi dengan database.

Pengujian komunikasi aplikasi dengan database dilakukan menggunakan Representational State Transfer Application Programming Interface (REST API) melalui aplikasi Postman[12], [13]. Sedangkan, pengujian aplikasi secara keseluruhan dilakukan pada perangkat smartphone dengan sistem operasi Android untuk memastikan aplikasi berfungsi dengan baik di lingkungan pengguna yang sesungguhnya.

3. HASIL DAN PEMBAHASAN

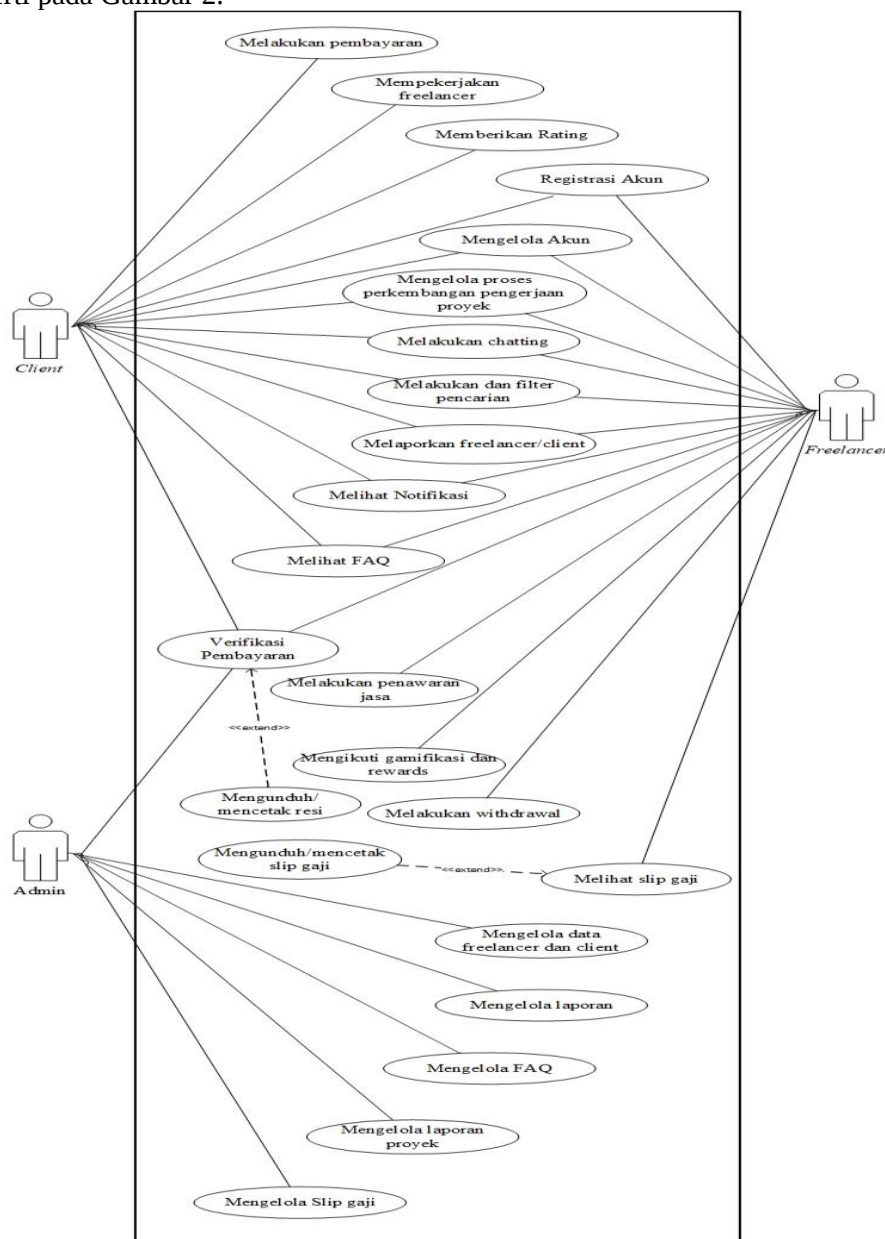
Bagian ini membahas hasil pengembangan aplikasi berdasarkan metode *Rapid Application Development* (RAD). Setiap tahapan dijelaskan secara terperinci, mulai dari perencanaan kebutuhan sistem hingga implementasi dan pengujian.

3.1. Perencanaan Kebutuhan Sistem (Requirement Planning)

Pada tahap ini, dilakukan analisis kebutuhan yang mendalam untuk memastikan bahwa sistem dapat memenuhi kebutuhan pengguna, baik dari sisi *freelancer*, *client*, maupun *admin*. Perencanaan ini mencakup identifikasi proses bisnis, fitur utama, serta batasan-batasan teknis yang harus dipenuhi oleh sistem.

3.1.1. Use Case Diagram

Use Case Diagram digunakan untuk memvisualisasikan interaksi antara pengguna dan sistem dalam aplikasi[14]. Diagram ini menggambarkan berbagai peran yang terlibat, seperti admin, freelancer, dan client, serta bagaimana masing-masing peran tersebut berinteraksi dengan fungsi-fungsi utama dalam sistem. Berikut adalah kebutuhan fungsional sistem usulan yang digambarkan dalam bentuk use case diagram seperti pada Gambar 2.



Gambar 2. Usecase Diagram

3.1.2. Defenisi Aktor

Berikut ini merupakan deskripsi aktor berdasarkan Diagram *Usecase* yang sudah ditampilkan pada Gambar 2:

Tabel 1. Defenisi Aktor Use Case Diagram

No.	Aktor	Deskripsi
1	Admin	Admin adalah orang – orang yang memiliki hak akses dalam aplikasi mobile dan memiliki hak tertinggi dalam pengelolaan sistem dan data.
2	Freelancer	<i>Freelancer</i> adalah pengguna aplikasi yang berperan sebagai penjual dan yang menawarkan jasa kepada <i>client</i> . <i>Freelancer</i> akan dapat melakukan Pendaftaran akun, <i>Login</i> , Mengelola Profil, Perpesanan, Pemberitahuan, Melihat FAQ, Pencarian dan Filter, Memberikan <i>Feedback</i> , Penilaian dan Peringkat, Laporkan Pengguna, Melihat Status Pembayaran, Pelacakan waktu dan kemajuan Proyek, Melakukan <i>Bidding</i> , Melakukan Penawaran Jasa, Melakukan Penarikan Dana, dan Melihat dan Mengunduh Laporan Penghasilan.
3	Client	<i>Client</i> adalah pengguna aplikasi yang berperan sebagai pembeli jasa dan menawarkan pekerjaan kepada <i>freelancer</i> . <i>Client</i> akan dapat melakukan Pendaftaran akun, <i>Login</i> , Mengelola Profil, Perpesanan, Pemberitahuan, Melihat FAQ, Pencarian dan Filter, Memberikan <i>Feedback</i> , Penilaian dan Peringkat, Laporkan Pengguna, Melakukan Pembayaran, Melihat Status Pembayaran, Pelacakan Waktu dan Kemajuan Proyek, Melakukan <i>Open Bidding</i> , Menetapkan <i>Bidding</i> Terpilih, dan Merekrut <i>Freelancer</i> .

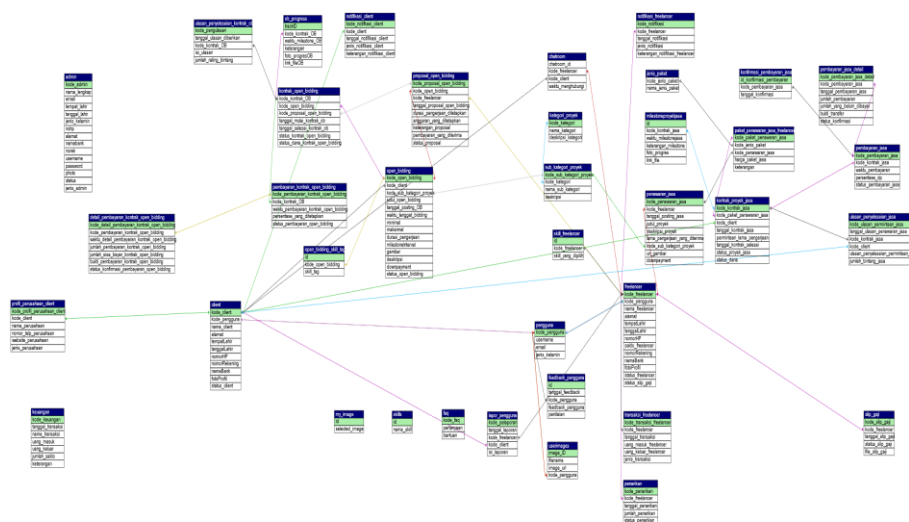
Tabel diatas menjelaskan peran dan hak akses dari masing-masing aktor dalam aplikasi berdasarkan use case diagram.

3.2. Rancangan Pengguna (User Design)

Tahap rancangan pengguna (*User Design*) berfokus pada perancangan antarmuka pengguna (user interface) serta proses interaksi yang terjadi dalam aplikasi. Desain ini bertujuan menciptakan pengalaman pengguna yang intuitif, responsif, dan mudah digunakan oleh tiga jenis pengguna utama: admin, freelancer, dan client.

3.2.1. Rancangan Database

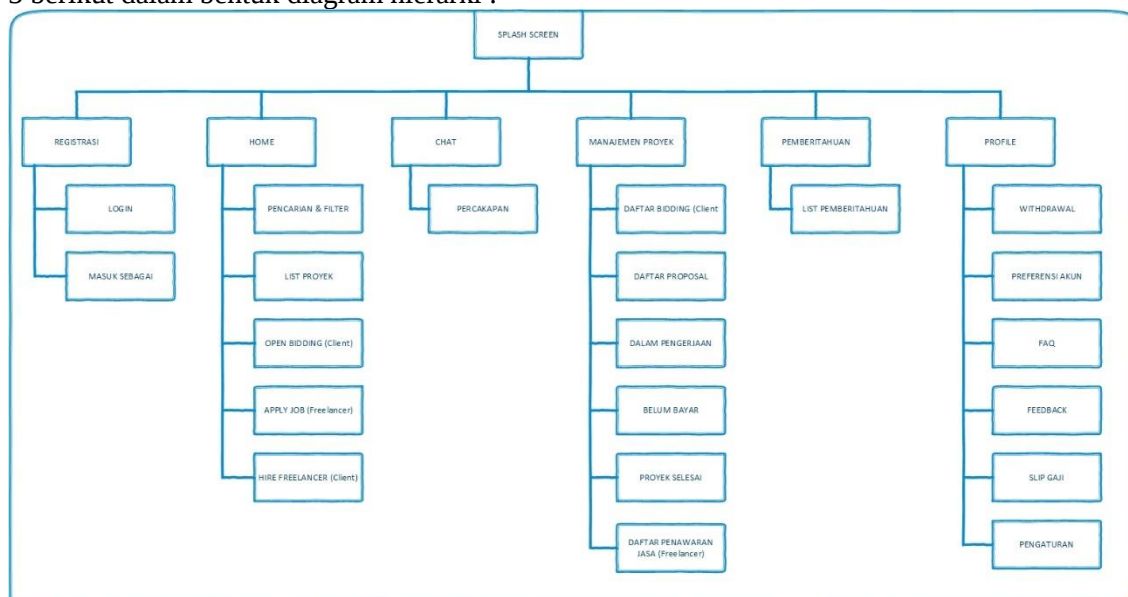
Perancangan database pada aplikasi di rancang dengan menggunakan DBMS MySQL dan terdapat struktur tabel dan view. Adapun gambaran relasi antar tabel database pada aplikasi adalah sebagai berikut:



Gambar 3. Diagram Basis Data ([Link Pratinjau Gambar](#))

3.2.2. Rancangan Struktur Aplikasi

Beberapa fitur dan sekaligus juga sebagai *layout* untuk aplikasi *mobile*-nya ditampilkan pada Gambar 3 berikut dalam bentuk diagram hierarki :



Gambar 4. Rancangan Struktur Aplikasi

Diagram di atas merupakan representasi visual dari alur dan struktur aplikasi yang dirancang untuk mempermudah pengguna dalam memahami navigasi dan fitur yang tersedia. Diagram ini menggambarkan hubungan antara fitur utama (level atas) dan sub-fitur (level bawah) yang saling terhubung secara hierarkis.

Untuk mempermudah pemahaman, setiap elemen yang terdapat dalam diagram akan dijelaskan lebih lanjut yang sudah dibuatkan dalam bentuk tabel, mulai dari Tabel 2 hingga Tabel 7 (Mengikuti diagram pada Gambar 2 dari kiri ke kanan).

Tabel 2. Fungsi Fitur *Registrasi*

Fitur	Fungsi
<i>Registrasi</i>	Fitur untuk melakukan pendaftaran akun.
<i>Login</i>	Fitur untuk masuk menggunakan aplikasi dengan akun yang sudah didaftarkan.
Masuk sebagai	Fitur untuk memilih pengguna akan menggunakan aplikasi sebagai <i>client</i> atau <i>freelancer</i> .

Tabel diatas menjelaskan fungsi dari fitur-fitur yang terdapat dalam proses registrasi dan autentikasi pengguna pada aplikasi.

Tabel 3. Fungsi Fitur *Home*

Fitur	Fungsi
Pencarian dan Filter	Fitur untuk melakukan pencarian dan di dukung fitur filter untuk pencarian yang lebih spesifik.
List Proyek	Fitur daftar proyek yang di tawarkan oleh <i>freelancer</i> .
<i>Open Bidding</i>	Penawaran yang dilakukan oleh <i>client</i> dengan sistem open bidding kepada <i>freelancer</i> .

<i>Apply Job/Proyek</i>	Fitur untuk <i>freelancer</i> menawarkan jasanya dengan mengajukan <i>bid</i> dan proposalnya.
<i>Hire Freelancer</i>	Fitur untuk memilih <i>freelancer</i> yang akan mengerjakan proyek yang ditawarkan oleh <i>client</i> .

Tabel diatas menjelaskan fungsi dari fitur-fitur utama yang terdapat pada halaman Home dalam aplikasi.

Tabel 4. Fungsi Fitur *Chat*

Fitur	Fungsi
Percakapan	Fitur percakapan antara sesama pengguna untuk melakukan negosiasi lebih lanjut atau memberikan informasi yang lebih detail.

Tabel diatas menjelaskan fungsi fitur percakapan yang tersedia dalam aplikasi.

Tabel 5. Fungsi Fitur Manajemen Proyek

Fitur	Fungsi
Daftar <i>Bidding</i>	Fitur untuk menampilkan daftar proyek yang ditawarkan dengan sisten <i>bidding</i> .
Daftar Proposal	Fitur untuk menampilkan daftar proposal dan <i>bid</i> yang diajukan oleh <i>freelancer</i> terhadap proyek yang ditawarkan oleh <i>client</i> .
Dalam Pengerjaan	Fitur untuk menampilkan daftar proyek yang sedang berjalan yang sedang dikerjakan oleh <i>freelancer</i> .
Belum Bayar	Fitur untuk menampilkan daftar proyek yang belum terbayarkan namun <i>client</i> dan <i>freelancer</i> sudah melakukan kesepakatan.
Proyek Selesai	Fitur untuk menampilkan daftar proyek yang sudah selesai.

Tabel diatas menjelaskan fungsi dari berbagai fitur yang digunakan untuk mengelola proyek dalam aplikasi.

Tabel 6. Fungsi Fitur Pemberitahuan

Fitur	Fungsi
<i>List Pemberitahuan</i>	Fitur untuk menampilkan daftar pemberitahuan yang ada pada aplikasi.

Tabel ini menjelaskan fungsi fitur pemberitahuan dalam aplikasi.

Tabel 7. Fungsi Fitur Profil

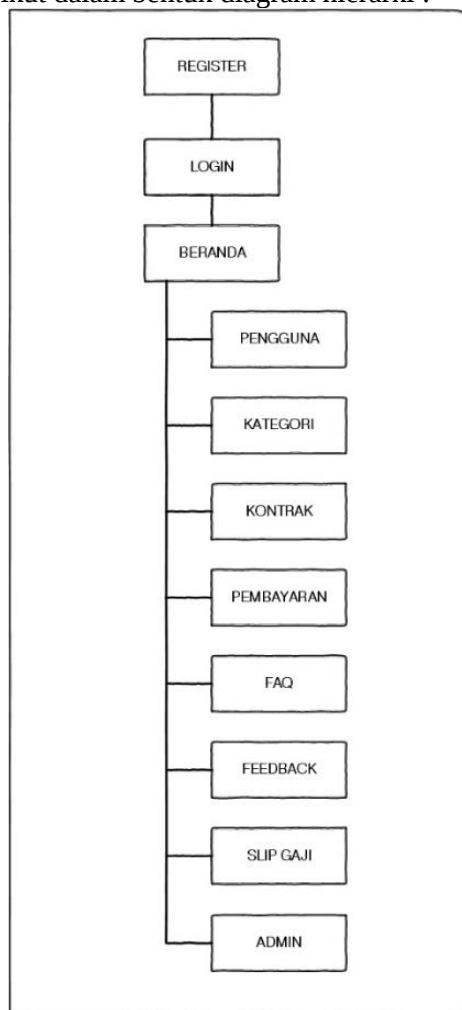
Fitur	Fungsi
Penarikan Dana	Fitur untuk melakukan penarikan dana dari aplikasi dari hasil pendapatan pembayaran proyek.
Preferensi Akun	Fitur untuk melakukan pengolahan akun seperti mengubah dan memperbaharui profil.
FAQ	Fitur yang dapat membantu pengguna dalam pemakaian aplikasi.
<i>Feedback</i>	Fitur yang memungkinkan pengguna untuk memberikan masukan berupa kritik dan saran.

Pengaturan	Fitur yang memungkinkan pengguna untuk mengubah pengaturan aplikasi dan untuk beralih jenis pengguna atau <i>logout</i> .
------------	---

Tabel diatas menjelaskan fungsi dari berbagai fitur yang tersedia dalam menu profil pengguna pada aplikasi.

3.2.3. Rancangan Struktur Admin

Beberapa fitur dan sekaligus juga sebagai halaman untuk aplikasi web yang akan diunakan oleh admin ditampilkan pada Gambar 4 berikut dalam bentuk diagram hierarki :



Gambar 5. Rancangan Struktur Admin

Penjelasan fitur terhadap diagram hirarki pada Gambar 4 diatas adalah sebagai berikut, Pada tabel 8 dibawah ini menjelaskan fungsi dari setiap fitur yang ada.

Tabel 8. Fungsi Fitur Pada Admin

Fitur	Fungsi
<i>Login</i>	Tampilan pertama pada admin yang memungkinkan admin untuk mengakses seluruh informasi pada aplikasi.
<i>Dashboard</i>	Tampilan utama tempat dimana administrator dapat mengatur dan mengelola konten pada situs aplikasi
Pengguna	Tempat dimana <i>administrator</i> dapat mengelola data keseluruhan pengguna

Kategori	Tempat dimana <i>administrator</i> dapat mengelola keseluruhan data kategori dalam aplikasi
Proyek	Tempat dimana <i>administrator</i> dapat mengelola keseluruhan data proyek
Pembayaran	Tempat dimana <i>administrator</i> dapat mengelola keseluruhan data pembayaran.
FAQ	Tempat dimana <i>administrator</i> dapat mengelola keseluruhan data FAQ.
Feedback	Tempat dimana <i>administrator</i> dapat mengatur dan mengelola data masukan dari pengguna.
Laporan Penghasilan	Tempat dimana <i>administrator</i> dapat mengelola laporan setiap transaksi pada aplikasi dan mengelola laporan penghasilan <i>freelancer</i> .
Admin	Tempat dimana <i>administrator</i> dapat mengelola keseluruhan data admin.

Tabel diatas menjelaskan berbagai fitur yang tersedia bagi administrator untuk mengelola aplikasi.

3.3. Implementasi (Implementation)

Tahap implementasi merupakan langkah nyata dalam merealisasikan rancangan yang telah dibuat sebelumnya. Pada tahap ini, aplikasi dikembangkan berdasarkan desain antarmuka dan struktur database yang telah dirancang. Proses ini mencakup pengkodean menggunakan bahasa pemrograman yang dipilih, integrasi antarmuka dengan basis data, serta pengujian fungsi untuk memastikan sistem berjalan sesuai dengan kebutuhan pengguna.

3.3.1. Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna (UI) bertujuan menciptakan tampilan aplikasi yang sederhana, fungsional, dan mudah digunakan. Aplikasi ini dirancang menggunakan React Native untuk platform Android, dengan tujuan melayani tiga jenis pengguna: admin, freelancer, dan client.

Setiap elemen UI dirancang agar responsif dan navigasi yang intuitif, dengan pengujian menyeluruh untuk memastikan pengalaman pengguna yang optimal. Berikut adalah tampilan dan penjelasan antarmuka pengguna aplikasi.

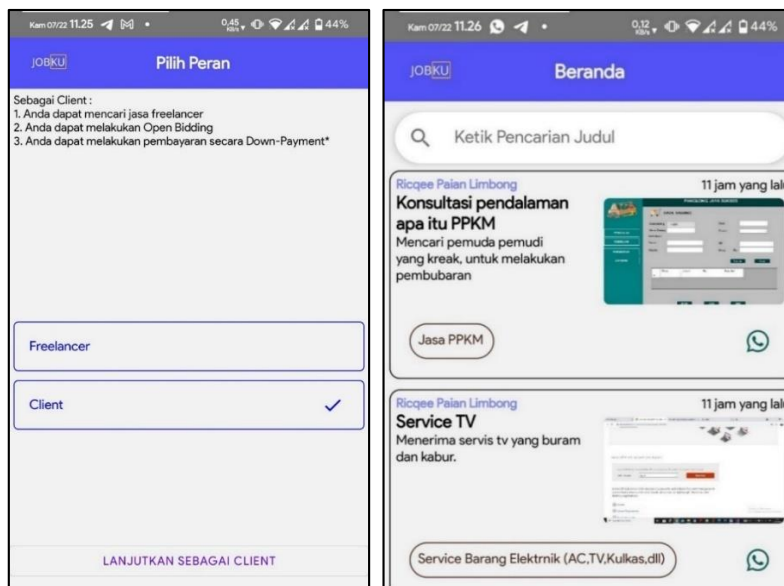
a. Antarmuka Pengguna untuk Client

Antarmuka pengguna untuk client didesain untuk mempermudah pengelolaan proyek, dengan fitur utama seperti:

1. **Pembuatan Proyek Baru:** Client dapat membuat proyek baru dengan detail kebutuhan mereka.
2. **Pemantauan Perkembangan Proyek:** Proyek dapat dipantau secara *real-time* melalui laporan yang tersedia.
3. **Pengelolaan Pembayaran:** Client dapat melakukan pembayaran dengan opsi seperti down payment.

Desain ini menekankan pada kemudahan navigasi, responsivitas, dan tampilan intuitif agar client dapat dengan mudah memahami alur aplikasi.

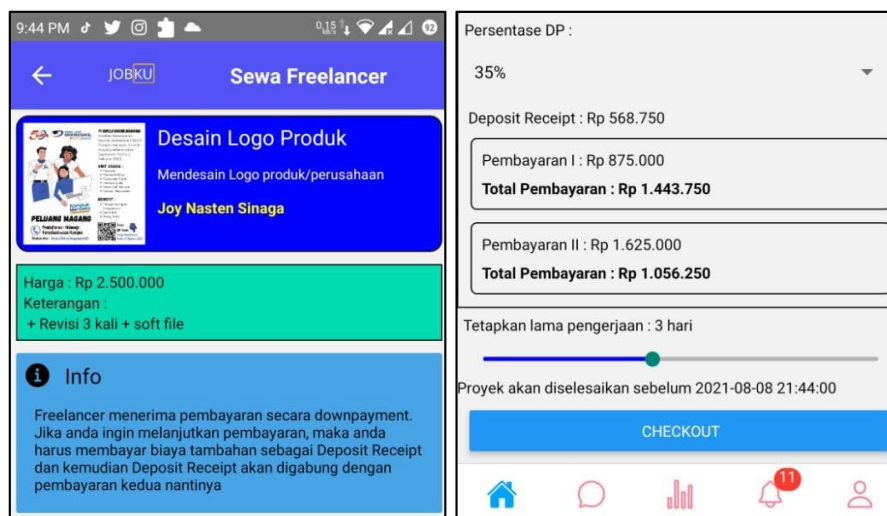
Saat pengguna sudah melakukan pendaftaran akun dan masuk ke dalam aplikasi, pengguna akan diminta untuk memilih opsi peran yang akan menentukan fungsional aplikasi yang dapat dilihat pada Gambar 6 (gambar pertama dari sebelah kiri).



Gambar 6. Antarmuka Pengguna Client

Sebelum pengguna memilih peran, aplikasi menampilkan penjelasan terhadap masing-masing peran yang akan dipilih. Kemudian setelah pengguna menekan tombol “Lanjutkan sebagai Client”, aplikasi akan menavigasikan pengguna ke tampilan beranda, yang merupakan tempat pengguna sebagai *Client* dapat mencari beberapa penawaran jasa sesuai dengan kebutuhan pengguna.

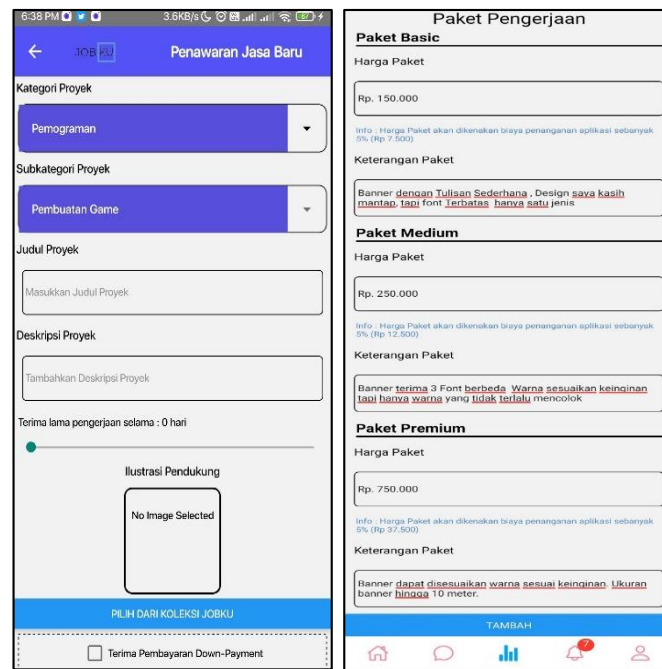
Ketika pengguna memilih salah satu penawaran dari beranda, maka akan dilanjutkan pada proses konfirmasi terhadap penawaran yang dipilih. Pada Gambar 7 menampilkan informasi penawaran beserta informasi kebijakan *down-payment* terhadap penawaran yang dipilih.



Gambar 7. Antarmuka Pengguna Client.

b. Antamuka Pengguna untuk Freelancer

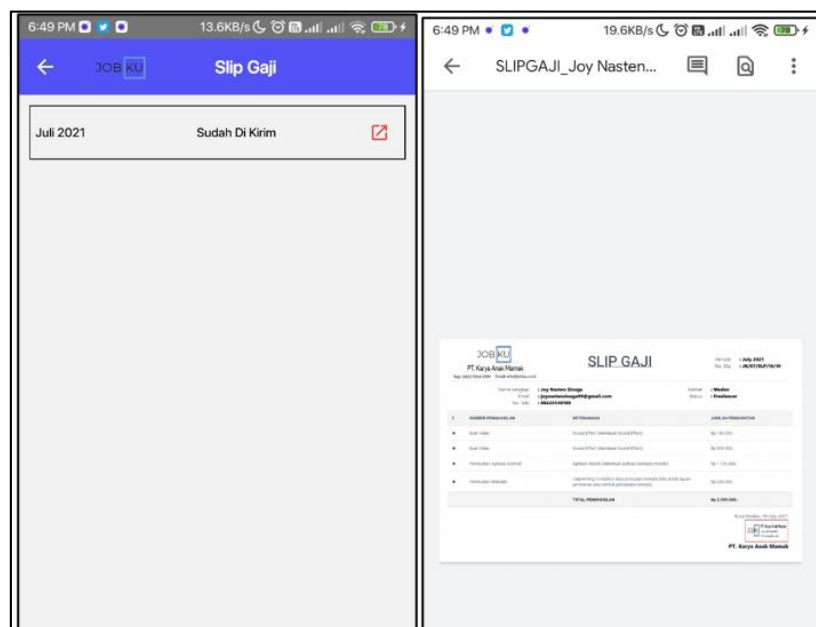
Freelancer memiliki fitur fungsionalitas yang hampir sama dengan Client, pengguna perlu memilih peran sebagai Freelancer untuk mengerjakan proyek, salah satunya membuat penawaran jasa. Pada Gambar 8 merupakan form untuk menambah penawaran jasa sebagai Freelancer.



Gambar 8. Menambah penawaran jasa sebagai Freelancer

Freelancer dimungkinkan untuk menambahkan detail mengenai penawaran jasa yang akan ditambahkan ke beranda, mulai dari info penawaran jasa, paket level pengerjaan, durasi pengerjaan, harga untuk setiap paket, dan menambahkan media pendukung.

Setiap Freelancer telah menyelesaikan proyek dengan baik dan dikonfirmasi oleh Client, maka Freelancer berhak mendapatkan komisi[15]. Pada Gambar 9 merupakan fitur yang memungkinkan Freelancer memiliki dokumen resmi mengenai informasi proyek yang telah dikerjakan dan pendapatan yang sah diterbitkan oleh pihak aplikasi setiap bulannya.

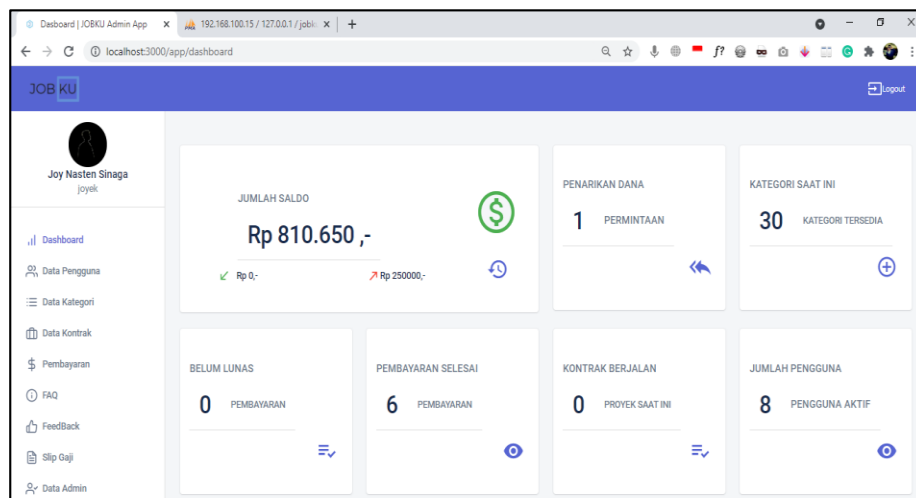


Gambar 9. Slip Gaji Freelancer

c. Antarmuka Pengguna untuk Admin

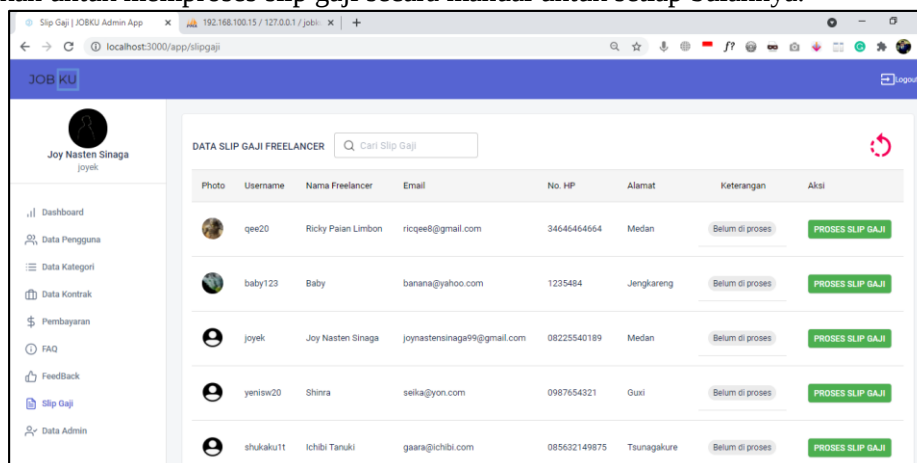
Antarmuka admin dirancang untuk mengelola data pengguna, proyek, serta transaksi pembayaran. Admin dapat memantau seluruh aktivitas dalam sistem, memastikan keamanan transaksi, dan mengelola laporan keuangan. Gambar 9 menunjukkan fitur utama dari halaman web sebagai admin,

yaitu dashboard yang menampilkan ringkasan informasi mengenai saldo keseluruhan pengguna hingga jumlah pengguna aktif.



Gambar 1. Antarmuka Pengguna Admin

Admin juga berperan untuk mengelola transaksi dan juga memantau kontrak yang sedang berlangsung. Admin juga bertugas untuk menerbitkan dokumen slip gaji yang siap untuk di pratinjau oleh Freelancer. Gambar 10 merupakan tampilan halaman web untuk mengelola slip gaji, admin dimungkinkan untuk memproses slip gaji secara manual untuk setiap bulannya.



Gambar 2. Antarmuka Pengguna Admin

3.4 Pembahasan

Pengembangan aplikasi ini mengikuti metode Rapid Application Development (RAD) yang memungkinkan iterasi cepat dalam perancangan dan implementasi. Setiap tahap dalam proses ini dijelaskan secara rinci, mulai dari perencanaan kebutuhan sistem hingga pengujian. Analisis kebutuhan dilakukan untuk memastikan sistem memenuhi kebutuhan tiga peran utama: admin, freelancer, dan client. Diagram use case digunakan untuk memvisualisasikan interaksi antar pengguna, sementara rancangan database dan struktur aplikasi dibuat untuk memastikan keterhubungan data yang efisien. Desain antarmuka difokuskan pada pengalaman pengguna yang intuitif, dengan fitur-fitur utama yang mendukung kemudahan navigasi dan penggunaan aplikasi.

Tahap implementasi mencakup pengembangan aplikasi menggunakan React Native untuk platform Android, dengan integrasi database berbasis MySQL. Fitur-fitur utama, seperti pengelolaan proyek, sistem bidding, perpesanan, dan pembayaran, dikembangkan sesuai dengan rancangan yang telah dibuat. Antarmuka untuk masing-masing peran dirancang agar responsif dan mudah digunakan, memastikan pengalaman pengguna yang optimal.

4. KESIMPULAN

Kesimpulan dari pengembangan aplikasi mobile untuk layanan freelance menunjukkan bahwa sistem yang diterapkan berhasil meningkatkan kualitas dan kredibilitas profesi freelance. Dengan adanya pembayaran menggunakan down payment (DP), transaksi menjadi lebih aman dan dapat meningkatkan kepercayaan antara freelancer dan client, sehingga mengubah pandangan negatif terhadap pekerjaan freelance.

Selain itu, freelancer mendapatkan laporan penghasilan bulanan yang otentik, memperkuat status profesional mereka dan memudahkan negosiasi gaji atau melamar pekerjaan baru. Client juga dapat memantau perkembangan proyek secara real-time, berkat kewajiban freelancer untuk memberikan laporan berkala, yang memastikan transparansi dan kelancaran proyek.

Terakhir, biaya penanganan aplikasi sebesar 5% dari nilai proyek menjadi sumber pendapatan untuk pengembang aplikasi, yang mendukung keberlanjutan pengembangan dan pemeliharaan aplikasi. Semua fitur ini mendukung tujuan utama penelitian untuk meningkatkan kualitas dan efisiensi layanan freelance melalui teknologi aplikasi mobile.

Daftar Pustaka

- [1] D. Wiryany, S. Natasha, and R. Kurniawan, "Perkembangan Teknologi Informasi dan Komunikasi terhadap Perubahan Sistem Komunikasi Indonesia," *Jurnal Nomosleca*, vol. 8, no. 2, pp. 242–252, 2022, doi: 10.26905/nomosleca.v8i2.8821.
- [2] E. Kurniasih and A. D. Milandry, "Jurnal Kajian Ilmu Hukum : IMPLIKASI PEMBERLAKUAN UNDANG-UNDANG CIPTA KERJA TERHADAP Info Artikel Intisari Jurnal kajian Ilmu Hukum : Pembangunan ketenagakerjaan sebagai bagian integral pembangunan nasional berdasarkan Pancasila dan Undang-Undang Dasar Ne," vol. 1, no. 2, pp. 176–193, 2022.
- [3] R. L. Pramana and Sugiyanto, "Tantangan Pemerintah Indonesia dalam menghadapi Gig Economy," *JEBI: Jurnal Ekonomi dan Bisnis*, vol. 2, no. 6, pp. 857–874, 2024.
- [4] A. Hartono, "Keadaan Pekerja di Indonesia Februari 2024," Badan Pusat Statistik, Jun. 2024. Accessed: Dec. 03, 2024. [Online]. Available: <https://www.bps.go.id/id/publication/2024/06/07/bd40ab2fa4fac6c726ab4ad4/keadaan-pekerja-di-indonesia-februari-2024.html>
- [5] N. D. Pramanik, "Pengaruh komunikasi organisasi dan motivasi terhadap kinerja freelance di Jagoketik.com," *Jurnal Akuntansi Keuangan dan Manajemen*, vol. 1, no. 3, pp. 213–226, 2020, doi: 10.35912/jakman.v1i3.40.
- [6] M. E. Hernantyo, V. Gunawan, and E. Adriana, "Peran Fitur Fiverr dan Manfaatnya bagi Digital Freelancer," vol. 4, pp. 250–263, 2024.
- [7] A. Anisah and R. Damayanti, "Perlindungan Hukum Bagi Pekerja Freelance : Analisis Regulasi , Tantangan , dan Akses Jaminan Sosial di Indonesia," vol. 2, no. 4, pp. 566–571, 2024.
- [8] B. Dharma, Y. Ramadhani, and R. Reitandi, "Pentingnya Laporan Keuangan untuk Menilai Kinerja Suatu Perusahaan," *El-Mujtama: Jurnal Pengabdian Masyarakat*, vol. 4, no. 1, pp. 137–143, 2023, doi: 10.47467/elmujtama.v4i1.3209.
- [9] A. Fauzi, G. Ginabila, and M. A. Azis, "Pengembangan Aplikasi E-learning dengan Metode Rapid Application Development," *Infotek : Jurnal Informatika dan Teknologi*, vol. 6, no. 1, pp. 82–91, Jan. 2023, doi: 10.29408/jit.v6i1.7414.
- [10] S. bin Uzayr, *Mastering React Native*. Boca Raton: CRC Press, 2022. doi: 10.1201/9781003310440.
- [11] P. Rawat and A. N. Mahajan, "ReactJS: A Modern Web Development Framework," *Int J Innov Sci Res Technol*, vol. 5, no. 11, pp. 698–702, 2020.
- [12] P. P. Kore, M. J. Lohar, M. T. Surve, and S. Jadhav, "API Testing Using Postman Tool," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 841–843, 2022, doi: 10.22214/ijraset.2022.48030.
- [13] I. R. D. Muhammad and I. V. Paputungan, "Development of Backend Server Based on REST API Architecture in E-Wallet Transfer System," *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 3, no. 2, pp. 79–87, Jan. 2024, doi: 10.20885/snati.v3.i2.35.
- [14] A. R. Abdurrahman, M. Bayu Rizki, R. Bagus Pradana, and A. Saka Fitri, "PERANCANGAN SISTEM INFORMASI GEOGRAFIS WILAYAH RAWAN PEMBEGALAN

MENGGUNAKAN METODE AAOD,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 1, pp. 1643–1650, Jan. 2025, doi: 10.36040/jati.v9i1.12710.

- [15] L. Khoirunnisa and I. G. K. C. B. A. Kusuma, “Tinjauan Atas Pajak Penghasilan Pekerja Lepas (Freelancer),” *JURNAL PAJAK INDONESIA (Indonesian Tax Review)*, vol. 8, no. 1, pp. 145–170, Jun. 2024, doi: 10.31092/jpi.v8i1.2685.



ZONAsi: Jurnal Sistem Informasi

Is licensed under a [Creative Commons Attribution International \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)