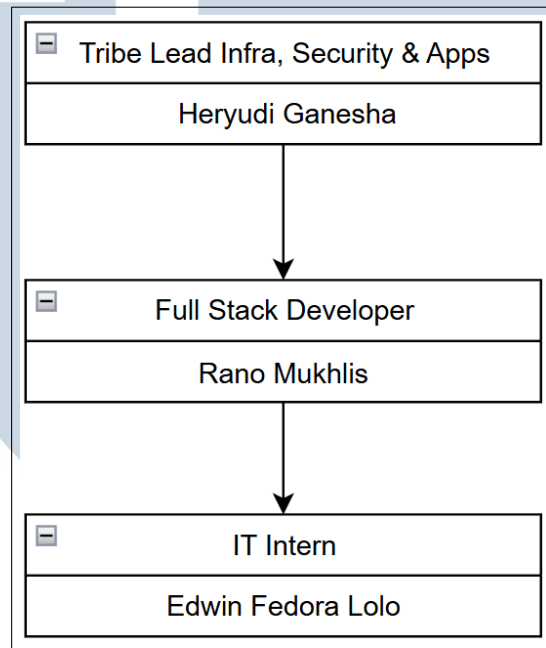


BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi



Gambar 3.1. Kedudukan dalam organisasi

Pelaksanaan kerja magang pada PT Devoteam Cloud Services sebagai *Fullstack App Developer Intern* pada departemen Professional Service Department dibawah pimpinan bapak Heryudi Ganesha selaku supervisi *Tribe Lead Infra, Security & Apps*. Selama masa kerja magang mendapatkan bimbingan dari bapak Rano Mukhlis selaku mentor selama menjadi karyawan magang di PT Devoteam Cloud Services. Koordinasi pelaksanaan kerja magang dilakukan melalui google chat atau bertatap muka secara langsung.

3.2 Tugas yang Dilakukan

Selama pelaksanaan program kerja magang sebagai Fullstack App Developer Intern di PT Devoteam Cloud Services, terdapat tiga proyek utama, yaitu HRIS Devoteam, BNI LDAP, dan Retina Telkomsel. Ketiga proyek ini dikerjakan secara paralel dan memberikan pengalaman kerja nyata di bidang pengembangan perangkat lunak, baik pada sisi frontend maupun backend.

Proyek HRIS Devoteam berfokus pada sistem manajemen internal perusahaan, sedangkan proyek BNI LDAP berkaitan dengan implementasi solusi berbasis AI untuk mendukung kebutuhan perusahaan. Sementara itu, proyek Retina Telkomsel merupakan aplikasi berbasis PWA yang dikembangkan untuk mendukung proses dokumentasi dan identifikasi produk di lapangan. Proyek Retina menjadi fokus utama karena kompleksitas teknis dan kontribusi yang lebih mendalam dibandingkan proyek lainnya. Berikut ini adalah penjelasan untuk setiap project:

3.2.1 HRIS Devoteam

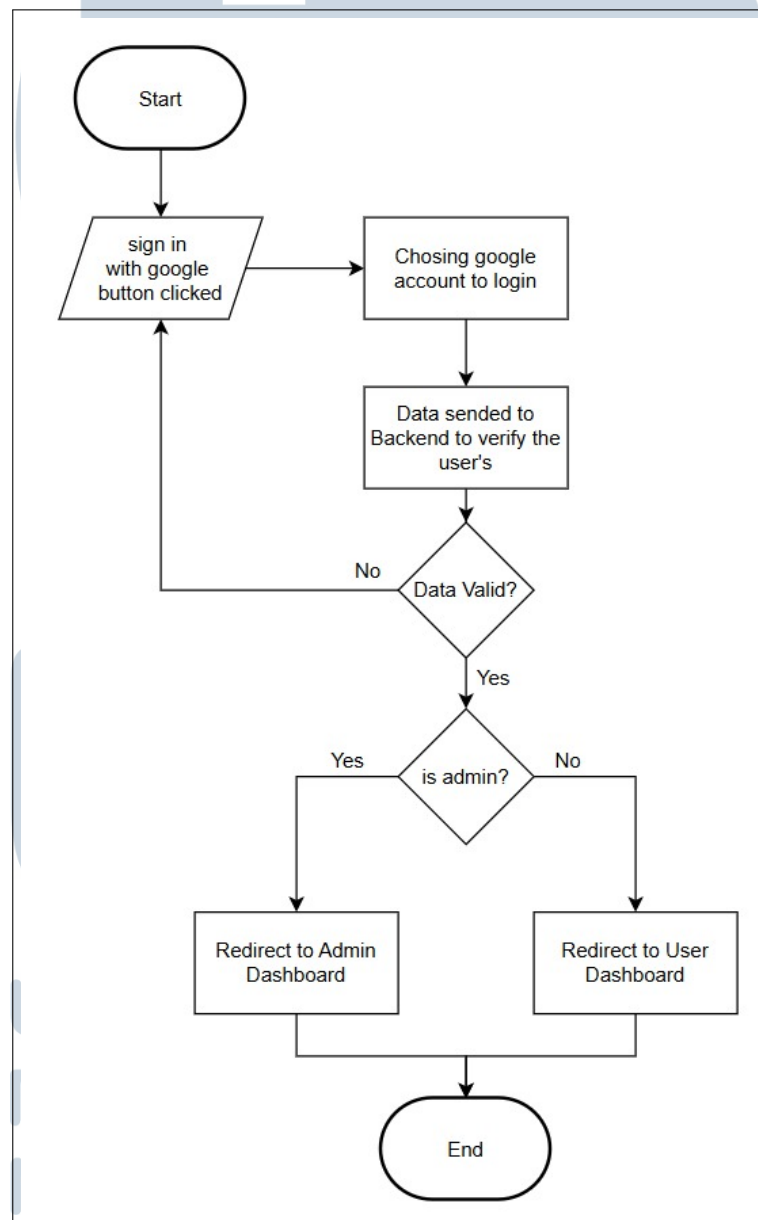
Proyek HRIS (*Human Resource Information System*) Devoteam merupakan salah satu pengembangan sistem internal perusahaan yang bertujuan untuk meningkatkan efisiensi dalam pengelolaan administrasi sumber daya manusia. Sistem ini dirancang sebagai aplikasi berbasis web yang memfasilitasi proses pengajuan cuti serta klaim reimbursement untuk kebutuhan kesehatan seperti perawatan mata dan gigi.

Tabel 3.1. Deskripsi pekerjaan yang dilakukan yang dilakukan dalam pengembangan project HRIS Devoteam

No	Deskripsi
1	Mengembangkan fitur autentikasi menggunakan layanan Google Authentication serta mengimplementasikan <i>middleware</i> pada sisi frontend untuk pengelolaan hak akses (<i>role management</i>) pengguna.
2	Melakukan integrasi antara antarmuka pengguna (frontend) dengan endpoint layanan backend yang telah tersedia.
3	Melakukan <i>testing</i> dan perbaikan terhadap kesalahan (<i>bug</i>) baik pada sisi frontend maupun backend.

A. Mengembangkan Fitur Autentikasi

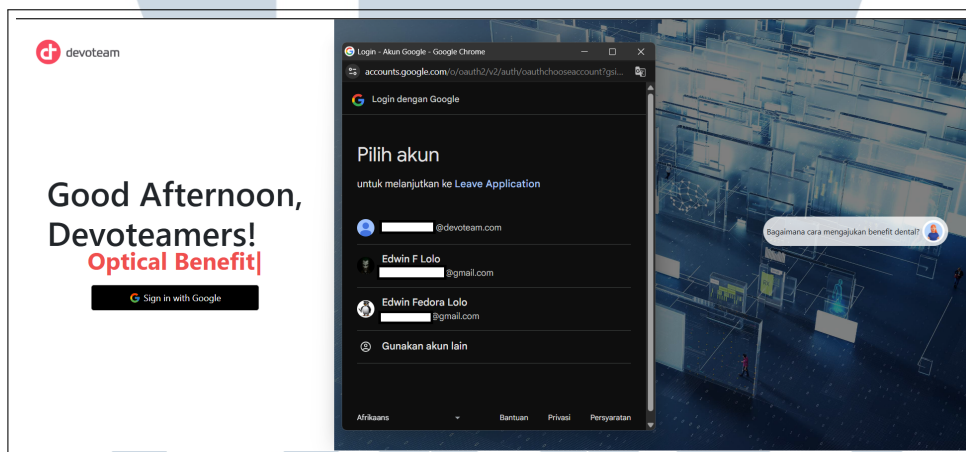
Fitur autentikasi merupakan komponen krusial dalam sistem informasi untuk menjamin keamanan dan validitas akses pengguna terhadap sistem. Pengembangan autentikasi dilakukan guna memastikan bahwa hanya pengguna yang memiliki hak akses yang sesuai yang dapat mengakses informasi dan fitur tertentu.



Gambar 3.2. Alur login dari HRIS

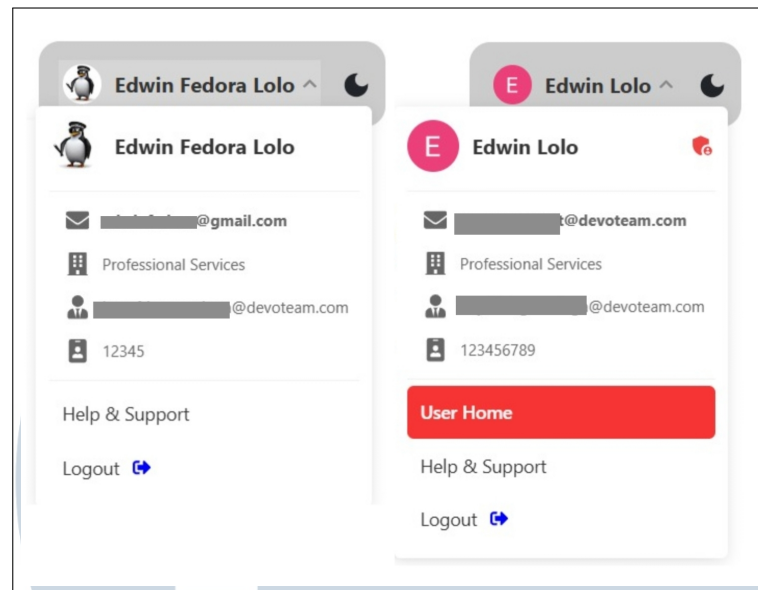
Pada saat pertama kali aplikasi diakses, sistem secara otomatis akan

mengarahkan pengguna menuju halaman login, di mana pengguna diminta untuk melakukan proses autentikasi menggunakan akun Google yang dimiliki. Setelah pengguna memilih akun yang akan digunakan, data autentikasi tersebut akan dikirimkan ke sisi backend untuk dilakukan proses verifikasi ulang terhadap data yang telah tersimpan di dalam basis data internal sistem. Proses verifikasi ini bertujuan untuk memastikan identitas pengguna sekaligus menjamin keamanan akses terhadap sistem. Apabila data pengguna dinyatakan valid, sistem akan melanjutkan dengan proses (*role checking*) untuk menentukan hak akses pengguna. Pengguna dengan *role* sebagai admin akan langsung diarahkan menuju admin dashboard guna mengelola fitur-fitur administratif, sedangkan pengguna dengan *role* non-admin akan diarahkan menuju halaman utama pengguna yang telah disesuaikan dengan otoritas akses yang dimiliki.



Gambar 3.3. Tampilan login dengan menggunakan Google auth

Gambar 3.3 menampilkan proses autentikasi diimplementasikan dengan memanfaatkan layanan pihak ketiga, yaitu Google Authentication, yang memungkinkan pengguna melakukan login secara aman melalui akun Google masing-masing.

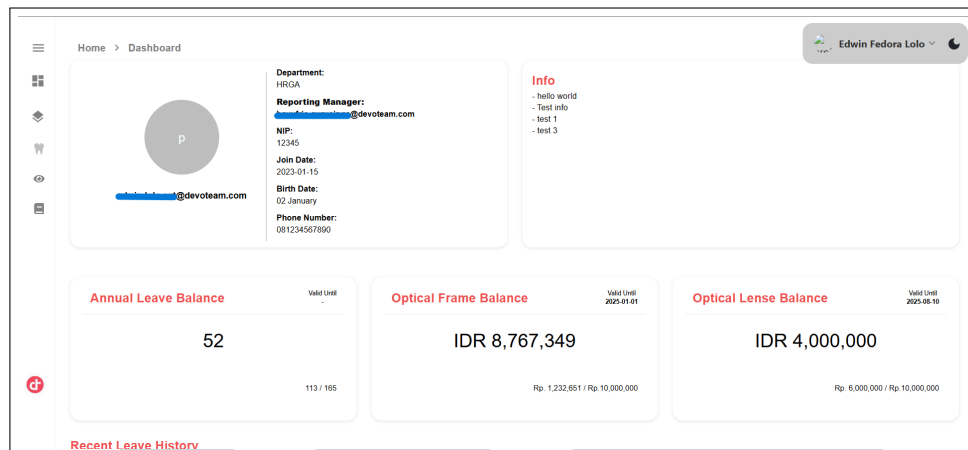


Gambar 3.4. Tampilan profile dengan role user

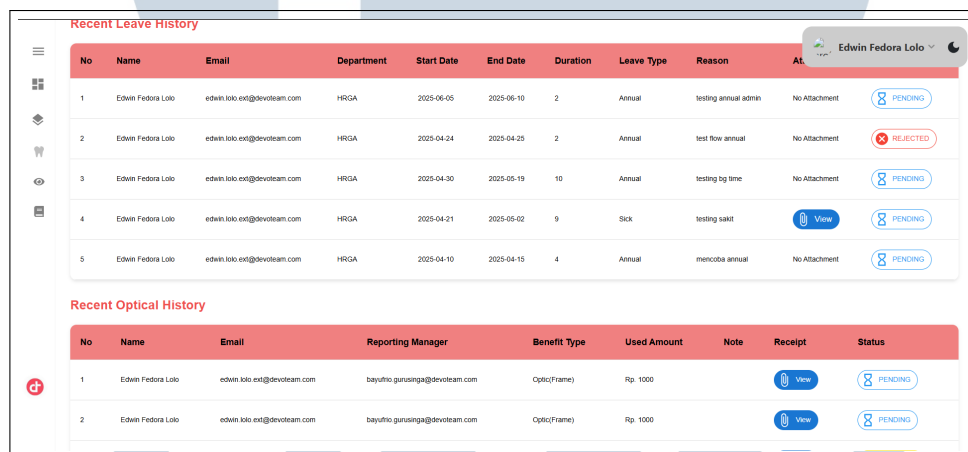
Gambar 3.4 menggambarkan mekanisme pengelolaan hak akses (*role management*) yang diterapkan untuk membatasi dan mengatur tingkat akses pengguna terhadap sistem, berdasarkan *role* yang telah ditentukan. Mekanisme ini bertujuan untuk memastikan bahwa setiap pengguna hanya dapat mengakses fitur dan data yang sesuai dengan otoritasnya.

B. Melakukan Integrasi Antara Frontend dengan Endpoint Backend

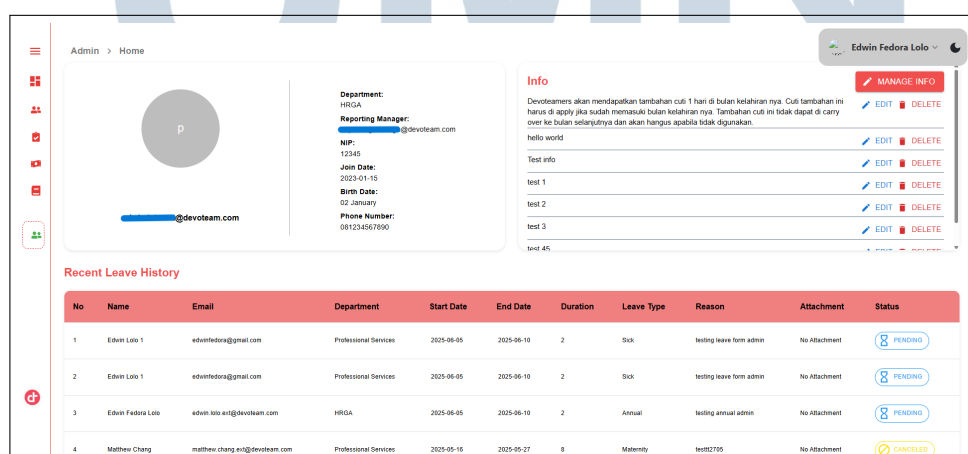
Integrasi antara frontend dan backend merupakan salah satu tahapan penting dalam pengembangan aplikasi berbasis web. Tahapan ini bertujuan untuk menghubungkan antarmuka pengguna dengan logika dan basis data yang dikelola oleh layanan backend, sehingga sistem dapat berjalan secara dinamis dan sesuai dengan kebutuhan pengguna. Dalam pelaksanaan tugas ini, bertanggung jawab untuk memastikan bahwa data yang dikirim dan diterima melalui *Application Programming Interface (API)* dapat ditampilkan dengan benar pada sisi antarmuka pengguna.



Gambar 3.5. Tampilan *user* dashboard dengan data yang diambil dari endpoint backend



Gambar 3.6. Tampilan *user* dashboard dengan data yang diambil dari endpoint backend



Gambar 3.7. Tampilan admin dashboard dengan data yang diambil dari endpoint backend

The screenshot shows an admin dashboard with two main tables. The top table, 'Recent Leave History', lists leave requests with columns: No, Name, Email, Department, Start Date, End Date, Duration, Leave Type, Reason, Attachment, and Status. The bottom table, 'Recent Optical History', lists optical services with columns: No, Name, Email, Reporting Manager, Benefit Type, Used Amount, Note, Receipt, and Status. Both tables show data for Edwin Fedora Lolo and Matthew Chang.

No	Name	Email	Department	Start Date	End Date	Duration	Leave Type	Reason	Attachment	Status
1	Edwin Lolo 1	edwinfedoral@gmail.com	Professional Services	2025-06-05	2025-06-10	2	Sick	testing leave form admin	No Attachment	PENDING
2	Edwin Lolo 1	edwinfedoral@gmail.com	Professional Services	2025-06-05	2025-06-10	2	Sick	testing leave form admin	No Attachment	PENDING
3	Edwin Fedora Lolo	edwin.lolo.ext@devoteam.com	HRGA	2025-06-05	2025-06-10	2	Annual	testing annual admin	No Attachment	PENDING
4	Matthew Chang	matthew.chang.ext@devoteam.com	Professional Services	2025-05-16	2025-05-27	8	Maternity	test12345	No Attachment	CANCELLED
5	Matthew Chang	matthew.chang.ext@devoteam.com	Professional Services	2025-05-14	2025-05-23	8	Maternity	test1	No Attachment	APPROVE

No	Name	Email	Reporting Manager	Benefit Type	Used Amount	Note	Receipt	Status
1	Edwin Fedora Lolo	edwin.lolo.ext@devoteam.com	bayuho.gunustnga@devoteam.com	Optic(Frame)	Rp. 1000		View	PENDING
2	Edwin Fedora Lolo	edwin.lolo.ext@devoteam.com	bayuho.gunustnga@devoteam.com	Optic(Frame)	Rp. 1000		View	PENDING

Gambar 3.8. Tampilan admin dashboard dengan data yang diambil dari endpoint backend

The screenshot shows a user's annual balance page. It features a 'Your current balance is:' section with a list of leave types and their respective counts. An 'Info' section provides additional details about the balance and how to use it.

Leave Type	Count
Annual:	52
Compassionate:	4
Marriage:	0
Childcare:	8
Sick:	49
Maternity:	46
Paternity:	48
Unpaid:	0

Info

- Devoteamers akan mendapatkan tambahan cuti 1 hari di bulan kelahiran nya. Cuti tambahan ini harus di apply jika sudah memasuki bulan kelahiran nya. Tambahan cuti ini tidak dapat di carry over ke bulan selanjutnya dan akan hangus apabila tidak digunakan.
- test 2
- test 45

Gambar 3.9. Tampilan *user* annual balance dengan data yang diambil dari endpoint backend

The screenshot shows a 'Leave Form' for a user to apply for leave. The form includes fields for Name, Email, Department, Start Date, End Date, and Duration (days). A note indicates that the user can only apply for a maximum of 10 working days.

Leave Form

Name: Edwin Fedora Lolo

Email: edwin.lolo.ext@devoteam.com

Department: HRGA

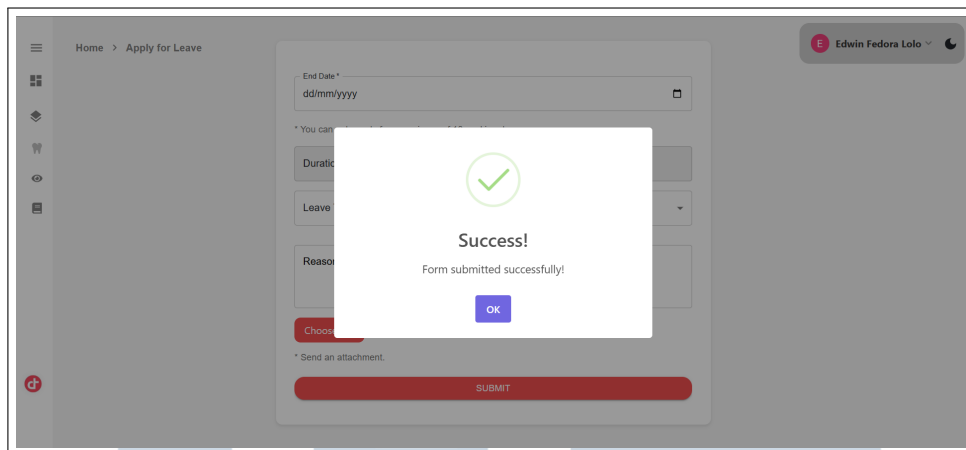
Start Date: dd/mm/yyyy

End Date: dd/mm/yyyy

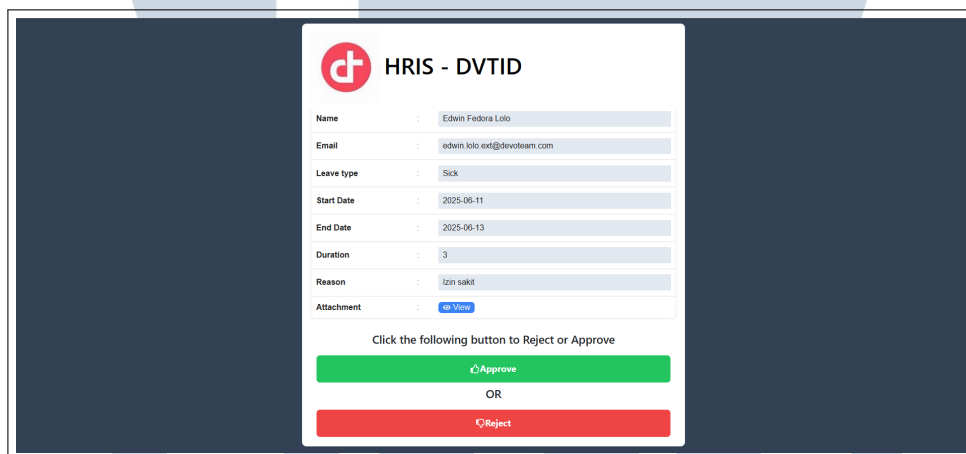
* You can only apply for a maximum of 10 working days.

Duration (days):

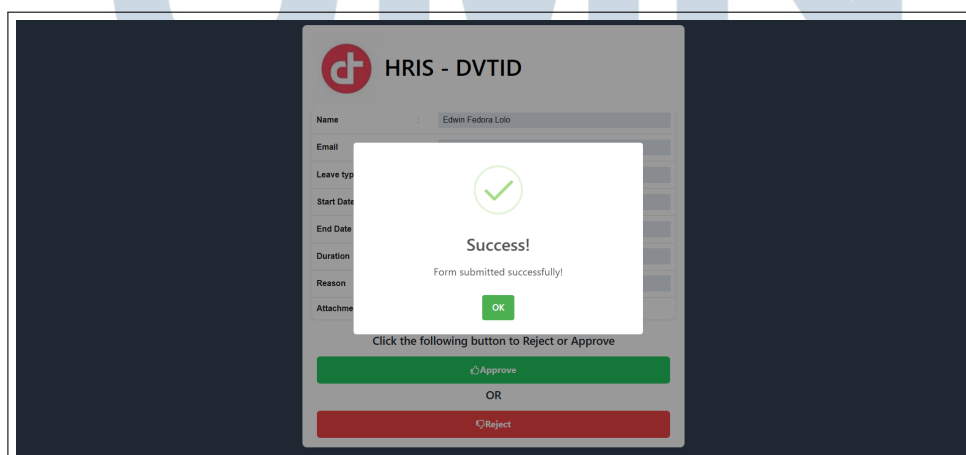
Gambar 3.10. Tampilan *user* annual form



Gambar 3.11. Tampilan *user* annual form



Gambar 3.12. Tampilan accept and reject form



Gambar 3.13. Tampilan accept and reject form

Gambar-gambar di atas menunjukkan tampilan hasil dari proses pengambilan data dari sisi backend yang kemudian ditampilkan pada antarmuka pengguna. Selain itu, ditunjukkan juga *form* untuk melakukan pengiriman data (*POST request*) ke sisi backend untuk keperluan penyimpanan ke dalam basis data, yang sebelumnya telah melalui proses validasi dan pemeriksaan logika yang diterapkan pada sistem.

C. Melakukan *Testing* dan Perbaikan Terkait *bug*

Kegiatan *testing* dan perbaikan terhadap *bug* merupakan bagian penting dalam proses pengembangan perangkat lunak untuk memastikan bahwa sistem berjalan sesuai dengan fungsionalitas yang diharapkan. Pengujian dilakukan guna mengidentifikasi kesalahan logika, ketidaksesuaian tampilan, maupun kegagalan fungsi yang dapat mengganggu pengalaman pengguna atau menyebabkan ketidakakuratan data. Setelah *bug* ditemukan, dilakukan proses analisis akar masalah serta perbaikan pada kode program baik di sisi frontend maupun backend, diikuti dengan pengujian ulang untuk memastikan stabilitas sistem setelah perubahan dilakukan.

The screenshot displays the 'Admin's Leave Form' within a web application. On the left, there is a sidebar with navigation icons. The main content area features a search bar at the top. Below it is a calendar for June 2025, with the 10th of June selected. To the right of the calendar are several empty input fields. Below these fields are two date pickers labeled 'dd/mm/yyyy' for start and end dates. A note states: '* You can only apply for a maximum of 10 working days.' At the bottom, there is a 'Simulation Leave Quota' section with a 'Duration (days)' input field. The top right corner shows the user 'Edwin Fedora Lolo' is logged in.

Gambar 3.14. Tampilan admin leave form

Gambar 3.15. Tampilan admin leave form

Gambar 3.14 dan Gambar 3.15 menunjukkan tampilan formulir yang digunakan oleh admin untuk menginput data cuti bagi pengguna lain. Pada Gambar 3.14, pemilihan tanggal cuti dibatasi hingga tanggal saat ini, sehingga tidak memungkinkan untuk memilih tanggal di masa mendatang. Setelah dilakukan perbaikan, sebagaimana ditampilkan pada Gambar 3.15, sistem telah disesuaikan sehingga admin dapat memilih tanggal cuti yang melebihi tanggal hari ini sesuai kebutuhan operasional, serta penambahan logika untuk durasi cuti.

Gambar 3.16. Tampilan admin untuk menambahkan user

The screenshot shows a web application interface for adding a user. On the left is a sidebar with navigation icons. The main content area is titled 'Admin > Admin Form'. It contains a form with the following fields:

- Reporting Manager (dropdown)
- Sex (dropdown)
- Role (dropdown)
- Birthday (text input)
- Join Date (text input with format dd/mm/yyyy)
- Annual (text input with value 0)
- Marriage (text input with value 0)
- Sick (text input with value 0)
- Compassionate (text input with value 0)
- Childcare (text input with value 0)
- Maternity (text input with value 0)
- Paternity (text input with value 0)

Below these fields is a section titled 'User Masters' containing:

- Annual Master (text input with value 0)
- Marriage Master (text input with value 0)
- Sick Master (text input with value 0)
- Compassionate Master (text input with value 0)
- Childcare Master (text input with value 0)

A red button labeled 'CREATE' is positioned at the bottom of the form.

Gambar 3.17. Tampilan admin untuk menambahkan user

Gambar 3.16 dan Gambar 3.17 menunjukkan tampilan formulir untuk penambahan pengguna baru. Pada Gambar 3.16, belum tersedia kolom untuk user master, yaitu data yang digunakan untuk menentukan jumlah hak cuti yang dapat diperoleh oleh masing-masing pengguna. Setelah dilakukan perbaikan, sebagaimana ditampilkan pada Gambar 3.17, formulir tersebut telah dilengkapi dengan bagian untuk menginput data user master. Selain penyesuaian pada sisi antarmuka, dilakukan juga pembenahan pada kode di sisi backend untuk mendukung penambahan data user master, termasuk pembuatan endpoint baru yang berfungsi untuk melakukan *rollback* secara otomatis apabila salah satu proses *POST* mengalami kegagalan.

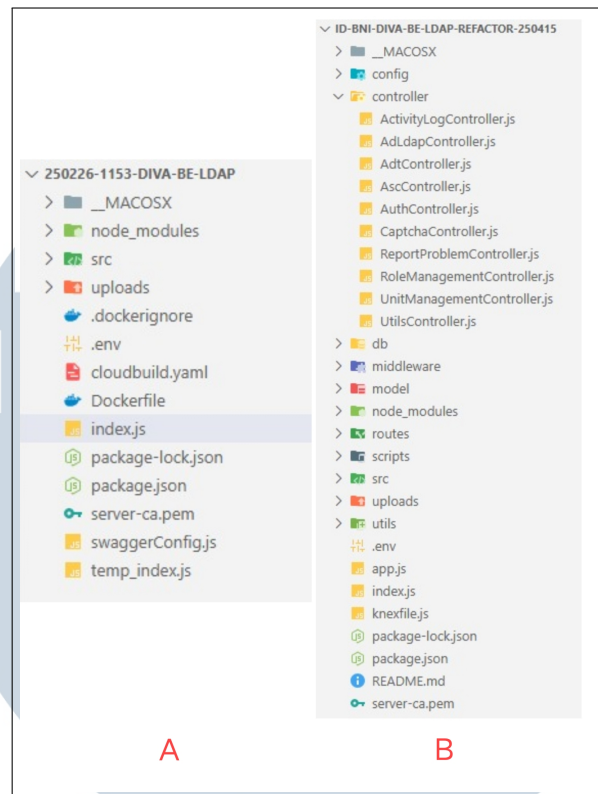
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.2 BNI LDAP

BNI LDAP merupakan proyek pengembangan sistem internal berbasis web yang dirancang untuk memungkinkan pengguna melakukan interaksi dengan kecerdasan buatan yang telah dikustomisasi secara khusus sesuai dengan pedoman dan kebijakan internal perusahaan. Sistem ini bertujuan untuk memberikan layanan tanya jawab yang efisien dan sesuai konteks organisasi. Mengingat sifat proyek yang bersifat rahasia, dokumentasi visual maupun teknis secara mendetail tidak dapat disertakan dalam laporan ini. Namun, pengembangan aplikasi dilakukan dengan tetap mengacu pada praktik terbaik dalam rekayasa perangkat lunak, baik dari sisi fungsionalitas maupun dari segi arsitektur sistem.

Ruang lingkup pekerjaan yang dilakukan dalam proyek ini mencakup beberapa aspek penting, di antaranya adalah perbaikan tampilan antarmuka pengguna (frontend), penambahan fitur pelaporan masalah yang mencakup pembuatan tampilan antarmuka pengguna (frontend) hingga integrasi dengan endpoint backend, serta kontribusi utama dalam melakukan refactoring kode pada sisi backend. Sebelumnya, struktur kode backend masih terpusat dalam satu *file* tunggal dengan logika yang tidak terpisah, sehingga menyulitkan proses pemeliharaan dan pengembangan lebih lanjut. Untuk mengatasi hal tersebut, dilakukan pemisahan antara logika dengan bagian lainnya seperti *routing*, pengendalian alur aplikasi, dan logika lainnya.





Gambar 3.18. Struktur file sebelum dan setelah refactor

Gambar 3.18 memperlihatkan perbandingan struktur data sebelum dan sesudah dilakukannya proses refactoring kode. Gambar A menunjukkan kondisi struktur data sebelum refactoring, sedangkan Gambar B menampilkan struktur yang telah diperbarui. Hasil refactoring memperlihatkan penyusunan data yang lebih terorganisasi dan sistematis, yang berdampak pada peningkatan keterbacaan kode serta mempermudah proses pemeliharaan dan pengembangan sistem di masa yang akan datang.

Teknologi yang digunakan dalam proyek ini meliputi Express.js sebagai kerangka kerja utama pada sisi backend, serta React sebagai kerangka untuk sisi frontend. Dengan mengadopsi struktur kode yang modular dan terorganisasi, proses pengembangan menjadi lebih terstruktur, serta mempermudah integrasi fitur baru dan pengujian unit secara terpisah. Pendekatan ini tidak hanya meningkatkan efisiensi kerja tim pengembang, tetapi juga mendukung keberlanjutan sistem dalam jangka panjang.

3.2.3 Retina Telkomsel

Project Retina Telkomsel adalah sebuah *Progressive Web App* (PWA) yang dikembangkan untuk membantu Telkomsel dalam melakukan identifikasi dan dokumentasi produk yang beredar di pasaran. Produk-produk yang didokumentasikan meliputi kartu perdana, poster promosi, etalase toko, tampak depan toko dan berbagai bentuk tampilan fisik dari kartu perdana yang berada di pasaran.

Aplikasi Retina Telkomsel ditujukan terutama untuk penggunaan pada perangkat mobile, sehingga desain dan implementasinya mengedepankan efisiensi performa, responsivitas tampilan, serta kompatibilitas dengan berbagai perangkat. Pengembangan aplikasi ini dilakukan dengan bimbingan dari mentor proyek, yang turut berperan dalam memastikan kesesuaian fitur dengan kebutuhan fungsional dan teknis yang ditetapkan oleh perusahaan.

A. Deskripsi Pekerjaan

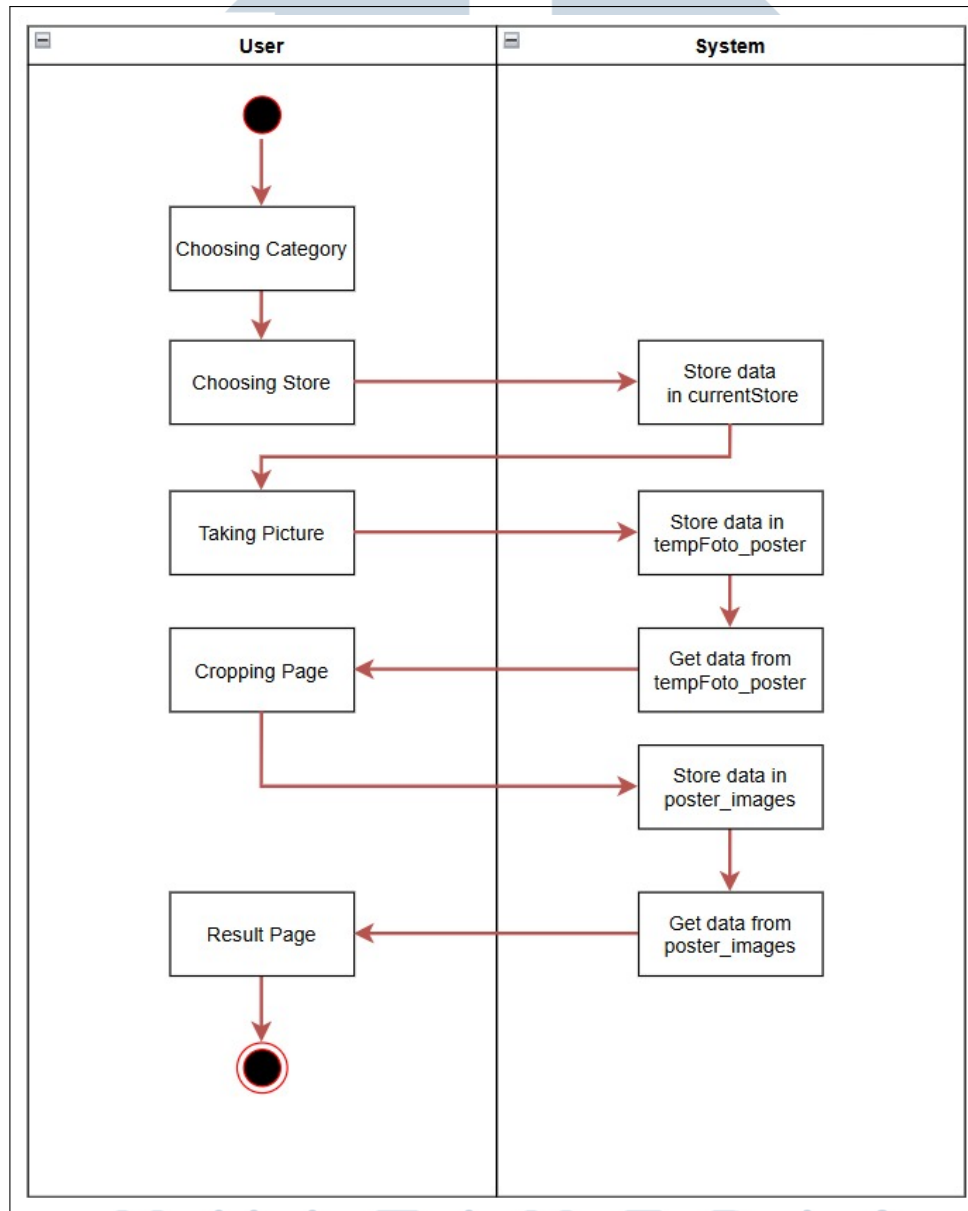
Deskripsi pekerjaan yang dilakukan selama pengembangan project Retina telkomsel, antara lain sebagai berikut:

Tabel 3.2. Deskripsi pekerjaan yang dilakukan yang dilakukan dalam pengembangan project Retina telkomsel

No	Deskripsi
1	Mengembangkan fitur kamera dengan kualitas tangkapan gambar yang lebih baik dibandingkan versi sebelumnya.
2	Menyediakan fitur pemotongan (crop) gambar tanpa menurunkan kualitas gambar secara signifikan.
3	Menambahkan fitur pembesaran (zoom) gambar guna meningkatkan fleksibilitas dalam pengambilan gambar.
4	Membuat menu “Kartu Perdana” yang memungkinkan pengguna untuk mengambil beberapa gambar kartu perdana secara berurutan dalam satu sesi pengambilan.
5	Menerapkan IndexedDB sebagai solusi penyimpanan lokal untuk menggantikan penggunaan localStorage dalam menyimpan data gambar berukuran besar.

B. Flowchart

B.1. Alur Menu Poster, Etalase, dan Storefront



Gambar 3.19. Activity Diagram Menu Poster

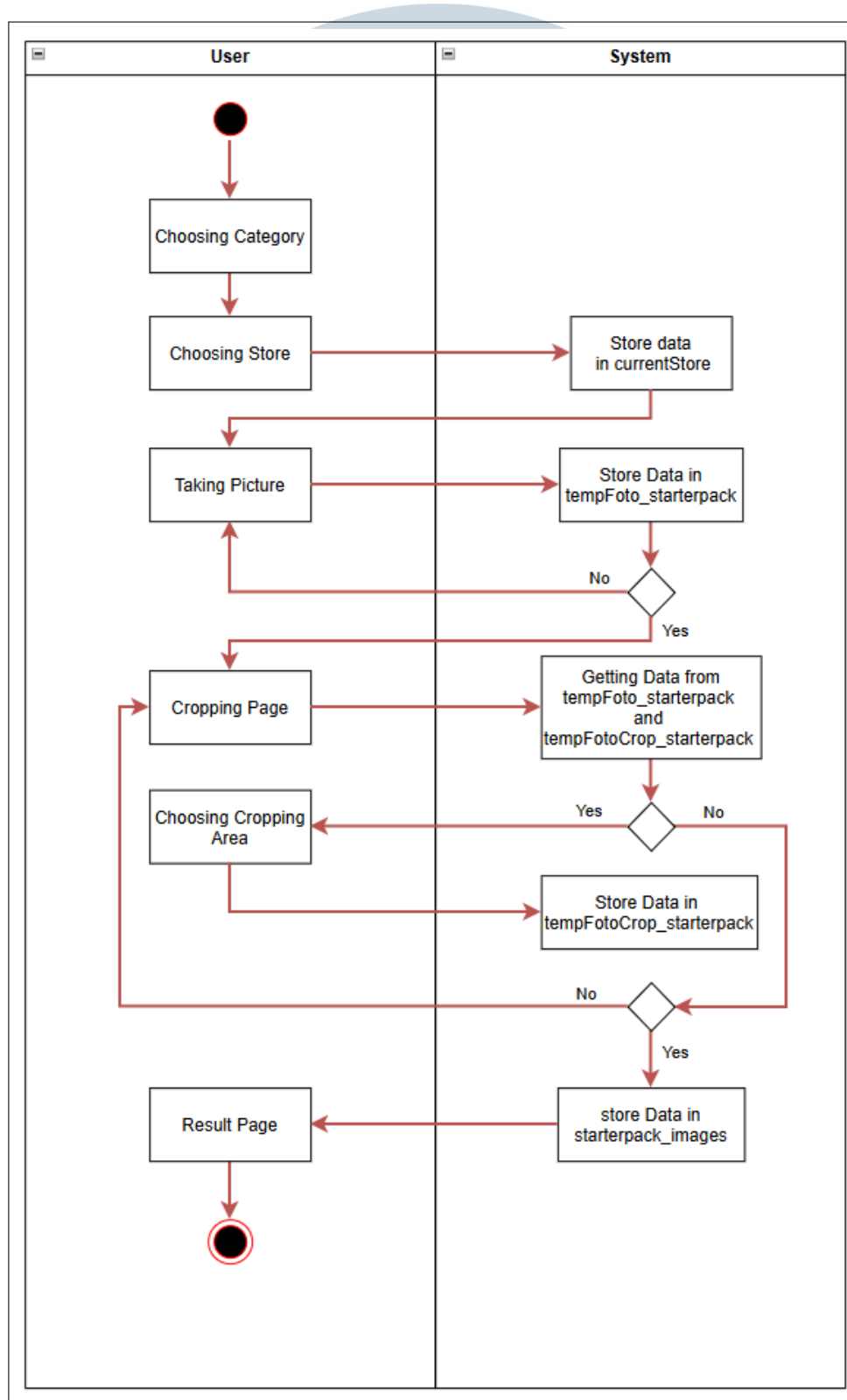
Pada bagian ini, sistem aplikasi dirancang untuk memulai proses dengan meminta pengguna untuk memilih salah satu kategori dan store yang tersedia. Informasi hasil pilihan store oleh pengguna kemudian akan disimpan ke dalam CurrentStore yang dikelola melalui IndexedDB. Setelah proses pemilihan selesai,

pengguna akan diarahkan ke fitur pengambilan foto, di mana gambar yang diambil akan langsung disimpan ke dalam tempFoto_poster sebagai penyimpanan sementara. Penyimpanan sementara ini berfungsi untuk menampung gambar mentah sebelum dilakukan tahap pengolahan lanjutan. Gambar yang tersimpan selanjutnya akan diproses menggunakan fitur pemotongan (*cropping*) untuk menyesuaikan fokus gambar sesuai dengan kebutuhan.

Setelah proses *cropping* selesai, gambar hasil pemotongan akan disimpan ke dalam penyimpanan poster_images bersamaan dengan metadata yang telah dihasilkan sebelumnya, seperti waktu pengambilan, ID store, dan informasi lainnya. Penyimpanan ini menjadi bagian penting dari proses dokumentasi data visual yang dibutuhkan oleh sistem. Tahapan akhir dari alur ini akan membawa pengguna menuju result page, yaitu halaman hasil yang menampilkan citra yang telah diproses serta informasi tambahan yang berkaitan dengan gambar tersebut. Seluruh proses ini menggunakan satu skema alur yang sama untuk dua jenis menu lainnya, yaitu menu etalase dan storefront, mengingat kesamaan logika proses yang diterapkan pada keduanya.



B.2. Alur Menu Kartu Perdana



Gambar 3.20. Activity Diagram Kartu Perdana

Pada tahap awal alur sistem ini, pengguna akan diminta untuk melakukan pemilihan terhadap jenis kategori serta store yang tersedia. Data hasil pilihan store tersebut kemudian akan disimpan ke dalam objek `CurrentStore` menggunakan penyimpanan lokal berbasis `IndexedDB`. Setelah proses pemilihan selesai, sistem akan mengarahkan pengguna ke tahap pengambilan foto. Foto-foto yang diambil pada tahap ini akan disimpan secara sementara di dalam `tempFoto.starterpack`, yang dirancang sebagai penampung gambar mentah sebelum dilakukan tahap pemrosesan lebih lanjut.

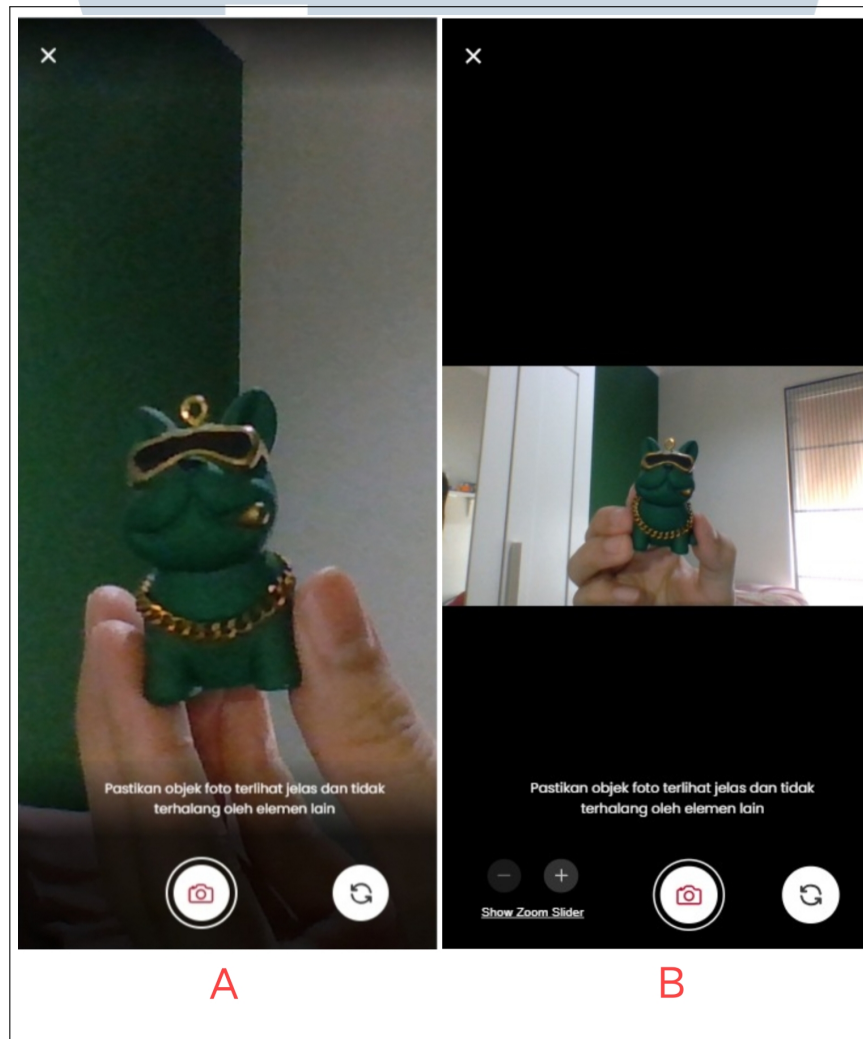
Setelah pengguna menyelesaikan pengambilan foto, sistem akan berpindah ke halaman *cropping*. Pada tahap ini, pengguna diberikan kemampuan untuk memotong bagian gambar yang dianggap relevan, guna memastikan bahwa hasil akhir sesuai dengan kebutuhan identifikasi. Gambar yang melewati proses pemotongan kemudian disimpan ke dalam direktori lokal `tempFotoCrop.starterpack`. Sistem dirancang untuk dapat menangani baik gambar yang telah melalui proses *cropping* maupun gambar yang tidak dipotong, dan seluruh data visual tersebut akan disimpan secara permanen ke dalam `starterpack.images` bersama dengan informasi metadata yang relevan, seperti identitas store, timestamp, dan atribut lainnya. Setelah proses penyimpanan selesai, pengguna diarahkan ke *result page*, yaitu halaman yang menyajikan hasil akhir gambar-gambar tersebut beserta data pelengkapannya.



C. Implementasi

C.1. Mengembangkan fitur kamera

Salah satu tantangan utama adalah meningkatkan kualitas modul kamera bawaan aplikasi. Implementasi awal menggunakan *library* react-webcam, namun hasil tangkapan gambar cenderung buram dengan ukuran file rendah (sekitar 400KB). Untuk mengatasi hal tersebut, dilakukan pengembangan ulang menggunakan `getUserMedia()` API yang dikombinasikan dengan elemen HTML video dan canvas, yang menghasilkan gambar dengan kualitas lebih tinggi dan ukuran file bisa mencapai sekitar 3MB, sesuai dengan kebutuhan tim identifikasi berbasis visual yang membutuhkan gambar yang memiliki kualitas tinggi.



Gambar 3.21. Perbandingan Kamera sebelum dan setelah pengembangan

Gambar 3.21 menampilkan dua hasil tangkapan kamera, yakni Gambar A yang diambil sebelum proses pengembangan dan Gambar B yang diperoleh setelah pengembangan dilakukan. Secara umum, Gambar B menunjukkan kualitas visual yang lebih tinggi, ditandai dengan tingkat ketajaman dan kejernihan yang lebih baik dibandingkan dengan Gambar A. Perbedaan ini tidak hanya terlihat dari peningkatan resolusi, tetapi juga dari penyesuaian aspek teknis lainnya yang mendukung optimalisasi hasil gambar. Selain itu, Gambar B menggunakan rasio aspek 16:9, yang memungkinkan gambar tetap mengikuti rasio asli perangkat, sehingga tampilan lebih proporsional. Sebagai tambahan informasi, jika gambar ditampilkan pada perangkat seluler dengan rasio 9:16, maka gambar akan memenuhi layar secara vertikal, seperti yang terlihat pada Gambar A. Penyesuaian ini turut mendukung kualitas tampilan, meskipun peningkatan yang terjadi bukan semata-mata disebabkan oleh perubahan rasio gambar. Perubahan ini juga dilakukan karena fokus utama dari aplikasi ini adalah pengguna *handphone* sehingga hasil tampilan akan sesuai dengan desain awal.





Gambar 3.22. Kamera setelah pengembangan menggunakan perangkat handphone

Gambar 3.22 menunjukkan hasil tangkapan kamera setelah pengembangan, yang diambil menggunakan kamera HP. Penggunaan kamera HP pada tahap ini dilakukan untuk menguji fungsi kamera setelah pengembangan, sekaligus menyesuaikan dengan kebutuhan implementasi nyata, di mana pengguna akhir umumnya akan menggunakan perangkat dengan kamera berkualitas lebih tinggi

seperti pada *handphone*.

Berdasarkan kedua gambar 3.21 dan 3.22, dapat diamati bahwa resolusi dan kualitas visual yang dihasilkan mengalami peningkatan signifikan dibandingkan dengan versi sebelumnya. Tingkat kejernihan dan ketajaman detail pada objek maupun tulisan dalam gambar terlihat lebih optimal, sehingga mendukung kebutuhan identifikasi visual secara lebih akurat.



C.2. Menyediakan fitur pemotongan (crop)

Implementasi fitur pemotongan gambar yang memungkinkan pengguna untuk memilih dan memotong area tertentu dari gambar secara manual. Fitur ini dihadirkan untuk memberikan fleksibilitas dalam mengelola hasil tangkapan kamera, khususnya dalam situasi di mana hanya sebagian objek pada gambar yang perlu digunakan untuk proses identifikasi atau dokumentasi. Proses pemotongan ini juga dilakukan tanpa menyebabkan penurunan kualitas gambar yang signifikan, sehingga hasil akhir tetap memenuhi standar visual yang dibutuhkan.

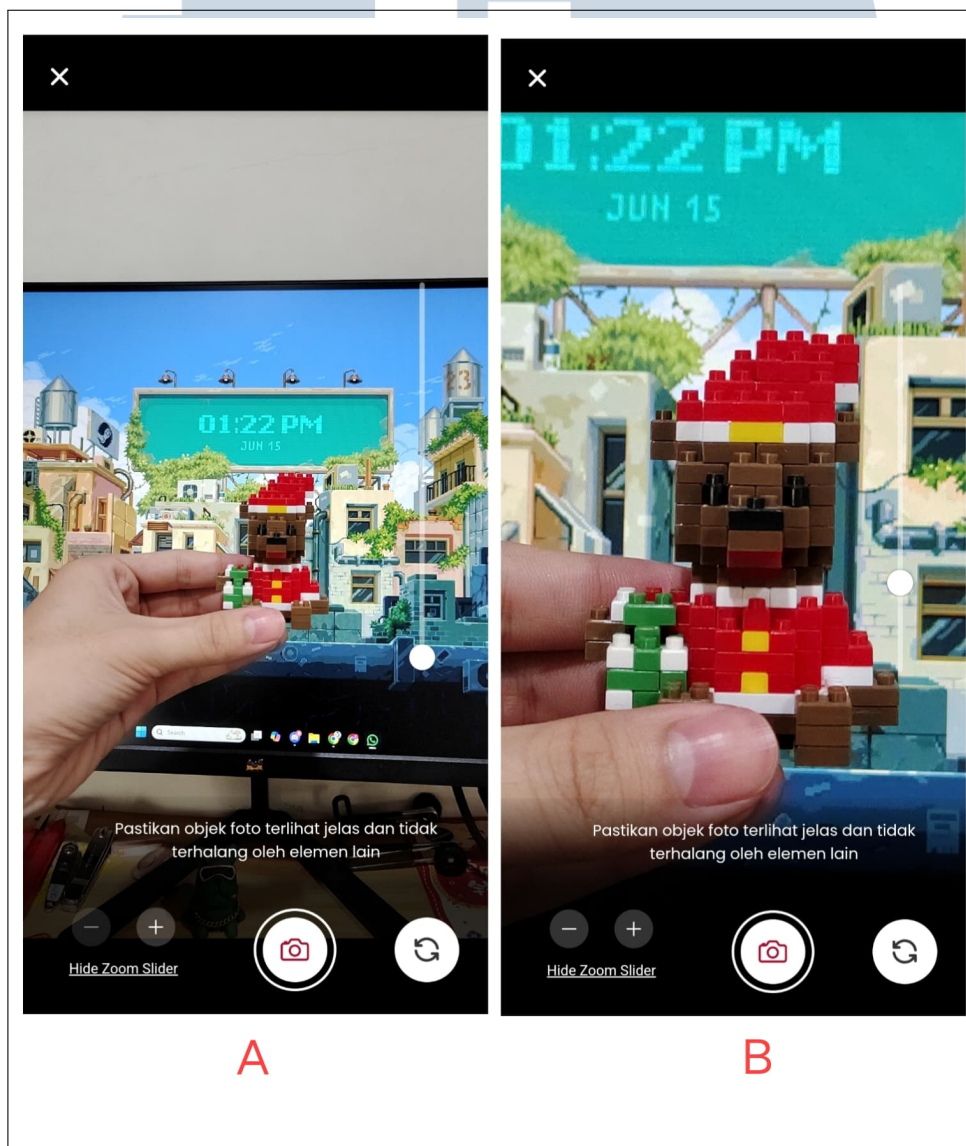


Gambar 3.23. Fitur pemotongan (crop) gambar

Gambar 3.23 merupakan hasil dari pengembangan fitur crop yang memungkinkan pengguna untuk menentukan area yang diinginkan dengan penyesuaian secara langsung, seperti menyeret dan memperbesar batas pemotongan sesuai kebutuhan.

C.3. Menambahkan Fitur Zoom

Fitur *zoom* merupakan salah satu peningkatan yang diterapkan pada sistem kamera dalam proyek Retina, yang bertujuan untuk meningkatkan fleksibilitas pengguna dalam proses pengambilan gambar. Melalui fitur ini, pengguna diberikan kemampuan untuk memperbesar tampilan objek secara digital sebelum melakukan pemotretan, sehingga objek yang berjarak relatif jauh tetap dapat ditangkap dengan jelas dan fokus.

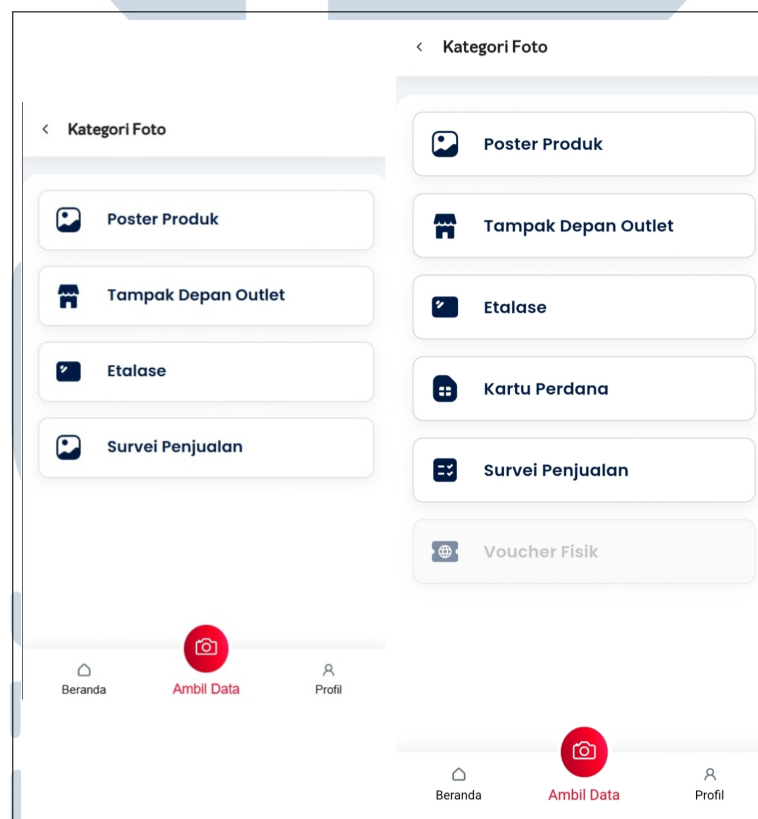


Gambar 3.24. Hasil kamera sebelum dan setelah *zoom*

Gambar 3.24 menampilkan implementasi fitur pembesaran (*zoom*) pada antarmuka pengguna. Pada ilustrasi tersebut, Gambar A menunjukkan tampilan sebelum proses pembesaran dilakukan, sedangkan Gambar B menampilkan hasil setelah fitur *zoom* diaktifkan. Berdasarkan perbandingan visual, tidak terlihat adanya penurunan kualitas resolusi yang signifikan setelah melakukan *zoom*. Kehadiran fitur ini memberikan kontribusi yang signifikan terhadap peningkatan akurasi dalam proses dokumentasi, terutama ketika objek yang diamati memiliki dimensi kecil atau berada pada jarak yang menyulitkan pengambilan gambar.

C.4. Membuat menu “Kartu Perdana”

Menu Kartu Perdana merupakan salah satu inovasi yang dikembangkan pada proyek Retina. Fitur ini dirancang agar pengguna dapat mengambil beberapa gambar kartu perdana secara berurutan dalam satu sesi pengambilan, tanpa perlu melakukan proses pemotretan secara terpisah.



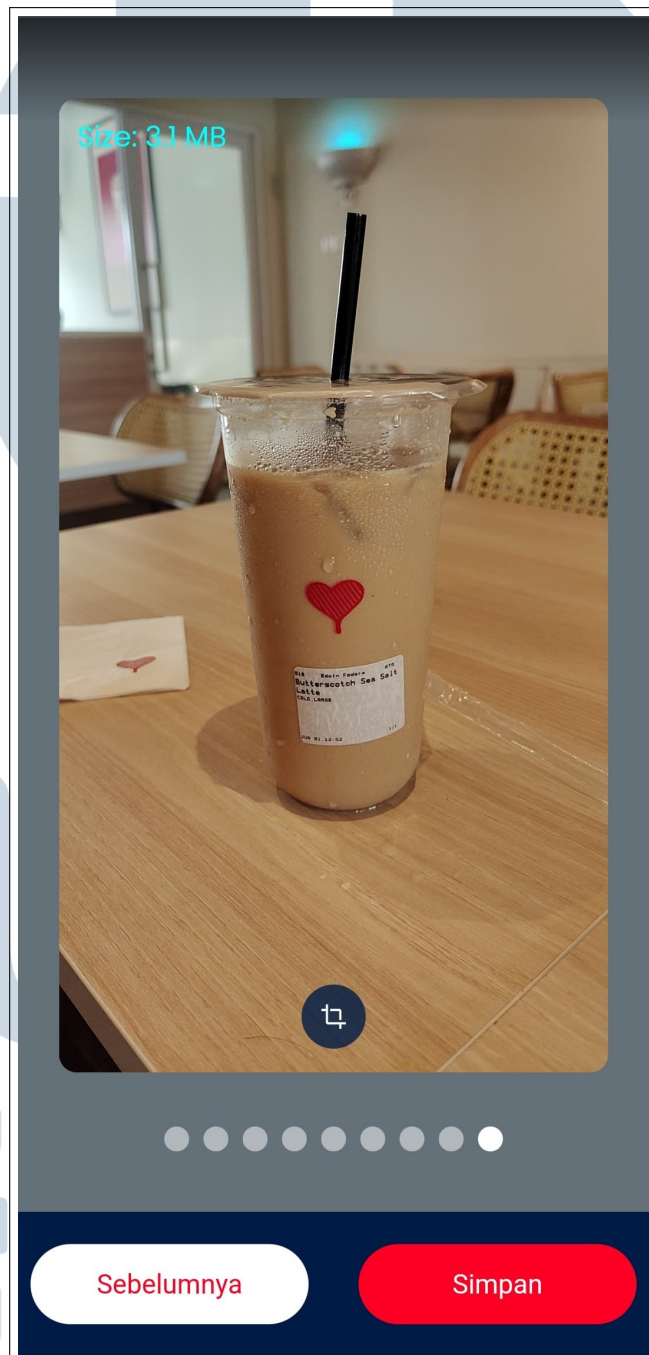
Gambar 3.25. Penambahan menu kartu perdana

Gambar 3.25 merupakan perbandingan tampilan antarmuka aplikasi sebelum dan sesudah penambahan menu Kartu Perdana. Tampilan sebelum pengembangan ditunjukkan pada bagian kiri gambar, sementara hasil setelah penambahan menu Kartu Perdana ditampilkan pada bagian kanan.



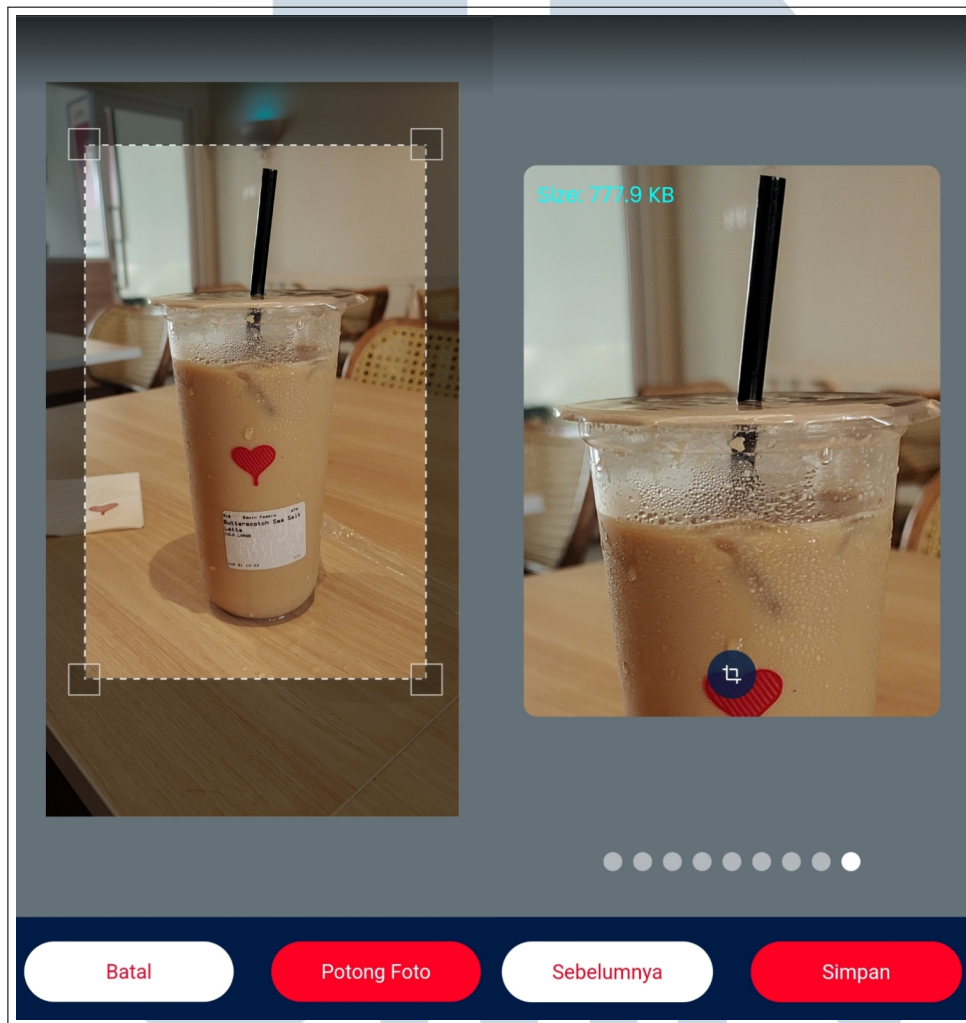
Gambar 3.26. Fitur foto pada menu kartu perdana

Gambar 3.26 menampilkan tampilan antarmuka fitur pengambilan foto pada menu kartu perdana. Pada fitur ini, pengguna diberikan fleksibilitas untuk mengambil foto secara berulang hingga mendapatkan hasil yang sesuai dengan kebutuhan.



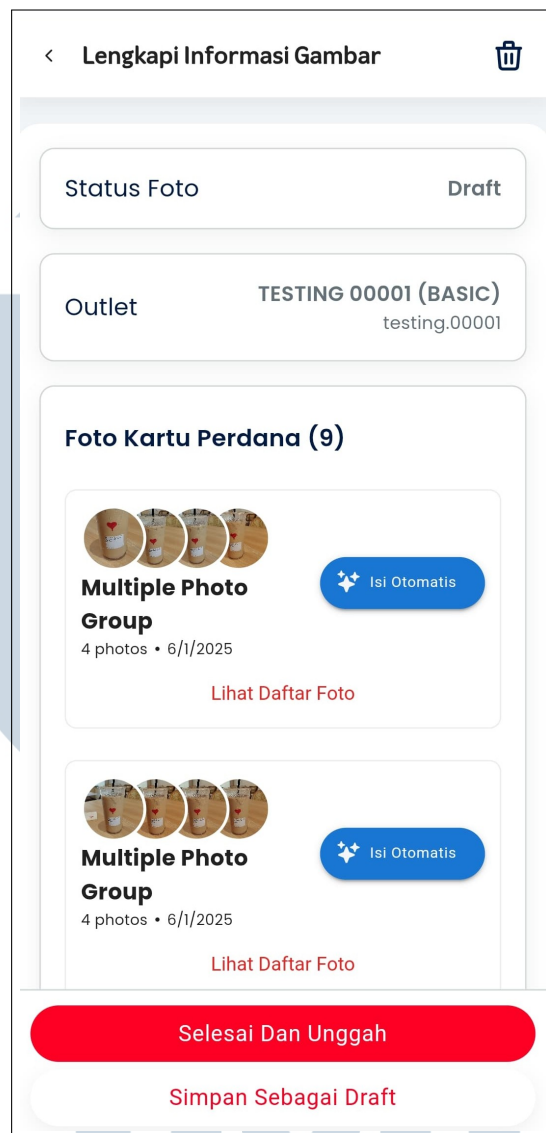
Gambar 3.27. Hasil foto pada menu kartu perdana

Setelah proses pengambilan gambar selesai, pengguna akan diarahkan ke tahap pemotongan gambar (*crop*) seperti pada gambar 3.27, di mana hasil tangkapan yang telah diambil ditampilkan untuk dilakukan penyesuaian area secara manual sebelum disimpan atau digunakan lebih lanjut.



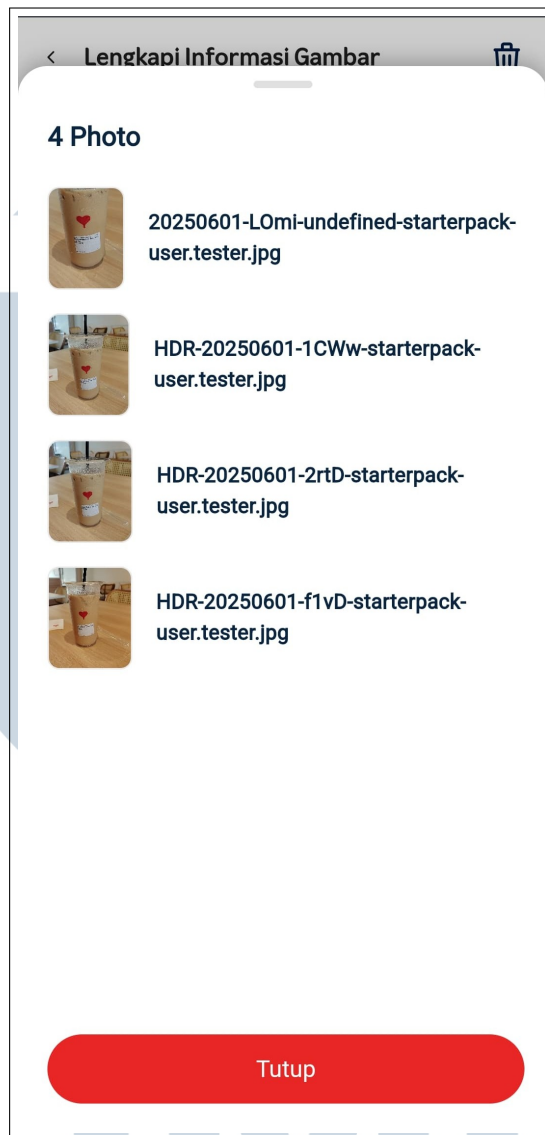
Gambar 3.28. Fitur crop pada menu kartu perdana

Gambar 3.28 menunjukkan hasil pemotongan (*crop*) gambar pada fitur Kartu Perdana. Proses pemotongan ini menghasilkan gambar dengan ukuran file yang lebih kecil dibandingkan gambar asli, namun tanpa menyebabkan penurunan kualitas visual yang signifikan. Resolusi gambar tetap terjaga dengan baik, sehingga detail pada objek tetap terlihat jelas dan tidak mengalami efek *blur*.



Gambar 3.29. Tampilan akhir pada menu kartu perdana

Gambar 3.29 memperlihatkan tampilan hasil akhir dari menu Kartu Perdana, di mana seluruh gambar yang telah diambil dikelompokkan secara sistematis ke dalam beberapa kelompok. Setiap kelompok terdiri atas empat gambar, yang disusun berdasarkan urutan pengambilan. Pengelompokan ini dirancang untuk memudahkan pengguna dalam melakukan proses pengisian data secara otomatis.



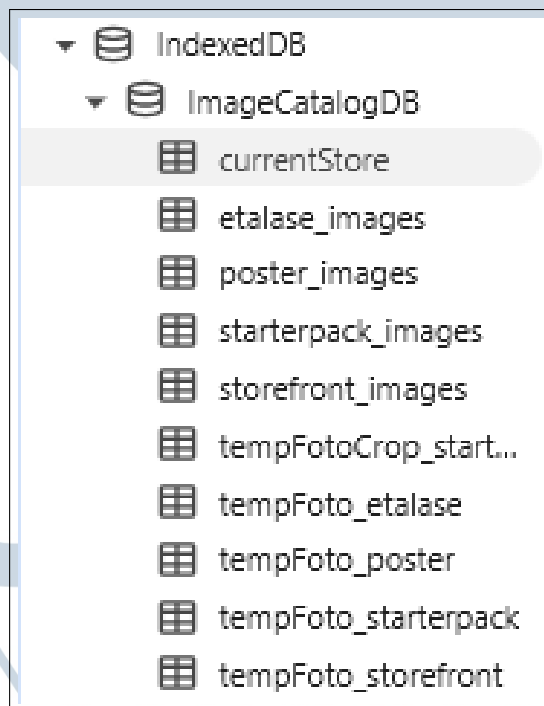
Gambar 3.30. Tampilan akhir pada menu kartu perdana

Gambar 3.30 memperlihatkan tampilan akhir dari data foto yang telah dikelompokkan. Tampilan tersebut disajikan menggunakan *library* react-modal-sheet, yang dirancang untuk meniru pengalaman interaktif layaknya aplikasi native.

C.5. Menerapkan IndexedDB

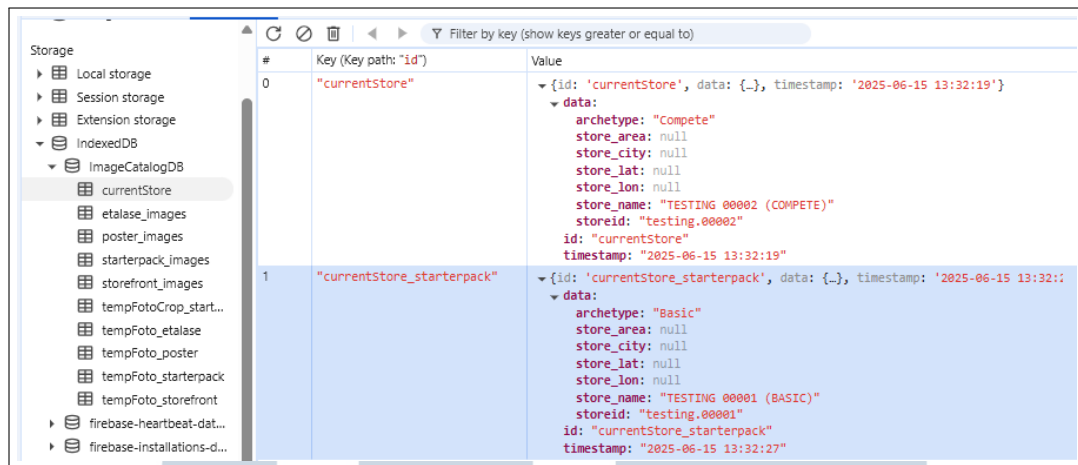
Pengembangan fitur kamera dalam proyek Retina menghasilkan gambar dengan kualitas dan ukuran yang lebih tinggi dibandingkan versi sebelumnya. Peningkatan ini berdampak pada kebutuhan ruang penyimpanan di sisi klien yang semakin besar, sehingga penggunaan localStorage sebagai media penyimpanan utama menjadi kurang optimal. Hal ini disebabkan oleh batas kapasitas localStorage yang relatif kecil, serta tidak didesain untuk menyimpan objek berukuran besar seperti gambar beresolusi tinggi.

Untuk mengatasi keterbatasan tersebut, digunakan IndexedDB sebagai solusi penyimpanan berbasis browser yang lebih efisien dan mampu menangani data dalam jumlah besar. Melalui penerapan IndexedDB, setiap hasil tangkapan kamera disimpan langsung dalam basis data lokal pengguna tanpa perlu dikirim ke server secara langsung.



Gambar 3.31. Penyimpanan pada IndexedDB

Gambar 3.31 menunjukkan struktur IndexedDB yang diinisialisasi secara otomatis ketika pengguna berhasil masuk ke dalam aplikasi PWA. Pada struktur tersebut, terdapat beberapa object store yang masing-masing memiliki fungsi spesifik, antara lain untuk menyimpan foto hasil tangkapan serta metadata yang terkait dengan setiap menu yang tersedia pada aplikasi.



#	Key (Key path: "id")	Value
0	"currentStore"	<pre>{id: 'currentStore', data: {}, timestamp: '2025-06-15 13:32:19'} data: archetype: "Compete" store_area: null store_city: null store_lat: null store_lon: null store_name: "TESTING 00002 (COMPETE)" storeid: "testing.00002" id: "currentStore" timestamp: "2025-06-15 13:32:19"</pre>
1	"currentStore_starterpack"	<pre>{id: 'currentStore_starterpack', data: {}, timestamp: '2025-06-15 13:32:27'} data: archetype: "Basic" store_area: null store_city: null store_lat: null store_lon: null store_name: "TESTING 00001 (BASIC)" storeid: "testing.00001" id: "currentStore_starterpack" timestamp: "2025-06-15 13:32:27"</pre>

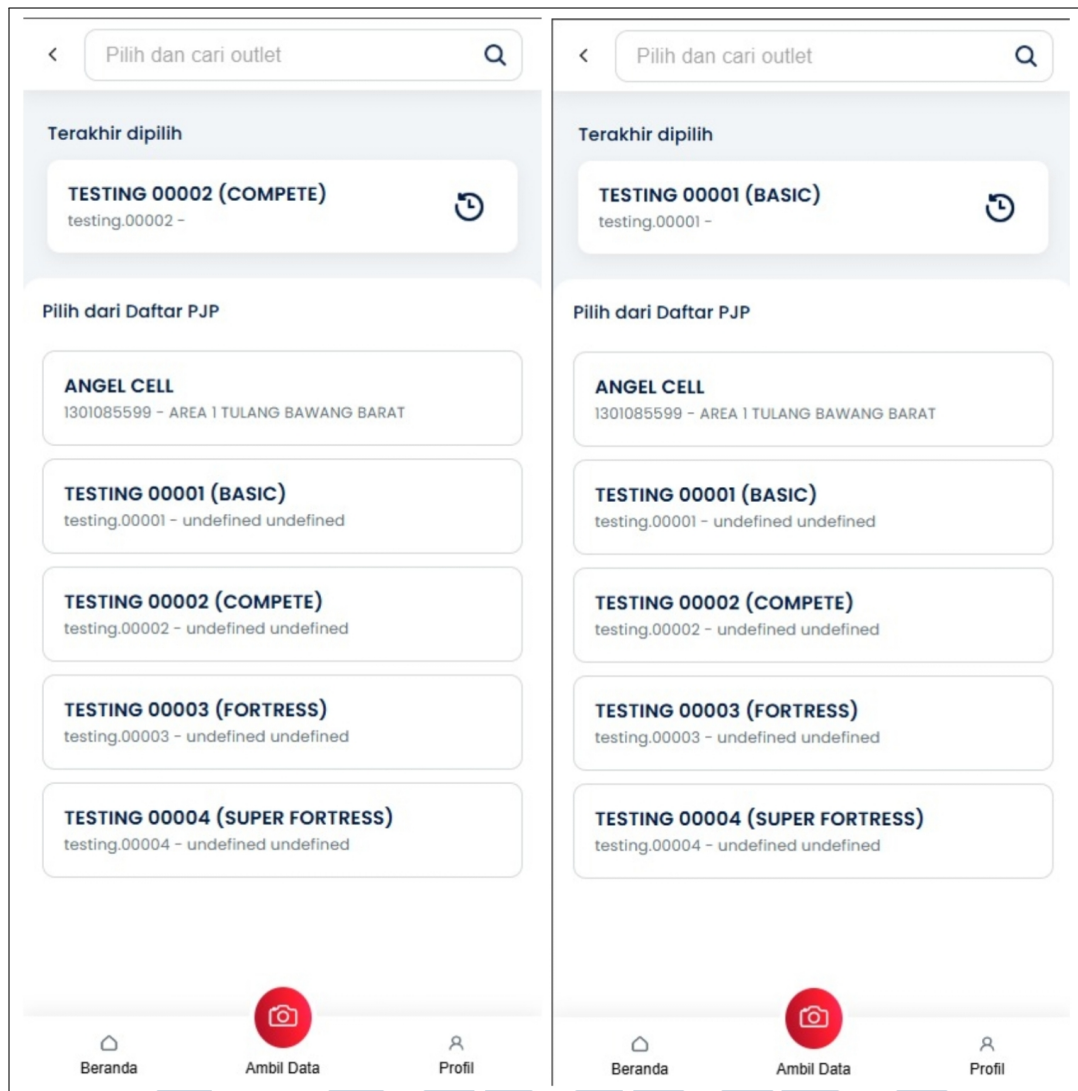
Gambar 3.32. Penyimpanan pada currentStore di IndexedDB

Gambar 3.32 menggambarkan struktur variabel `currentStore` dan `currentStore_starterpack`. Kedua variabel ini dipisahkan untuk menangani perbedaan logika antara jenis store umum dan store khusus untuk fitur starterpack.

Variabel `currentStore_starterpack` berfungsi untuk memeriksa apakah store yang sedang aktif sama dengan store baru yang dipilih oleh pengguna. Apabila terjadi perbedaan antara keduanya, maka seluruh data sementara yang tersimpan di `tempFoto_starterpack` dan `tempFotoCrop_starterpack` akan dihapus secara otomatis. Mekanisme ini dirancang untuk mengoptimalkan penggunaan ruang penyimpanan (*storage*).

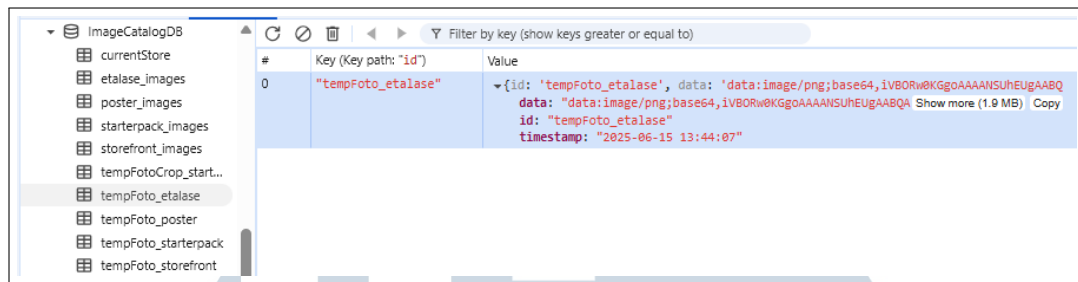
Selain itu, baik `currentStore` maupun `currentStore_starterpack` juga berperan dalam mencatat informasi mengenai store terakhir yang diakses. Informasi ini dapat digunakan untuk mendukung pelacakan aktivitas pengguna, atau penyelarasan data ketika pengguna kembali ke sesi sebelumnya. Pendekatan ini berkontribusi pada efisiensi sistem sekaligus meningkatkan keandalan dalam pengelolaan data gambar dan metadata yang terkait.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



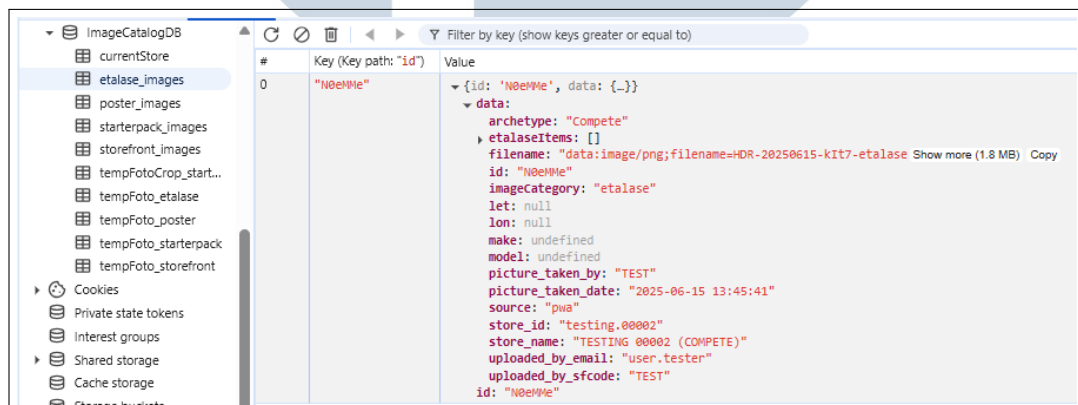
Gambar 3.33. Store terakhir yang dipilih pada menu poster dan menu kartu perdana

Gambar 3.33 memperlihatkan data store terakhir yang diakses oleh pengguna pada dua menu yang berbeda, yaitu menu Poster (ditunjukkan pada gambar sebelah kiri) dan menu Kartu Perdana (ditunjukkan pada gambar sebelah kanan).



Gambar 3.34. Penyimpanan pada tempFoto_etalase di IndexedDB

Gambar 3.34 menunjukkan tempFoto_etalase berfungsi sebagai media penyimpanan sementara untuk hasil tangkapan kamera. Data yang disimpan pada tempFoto berupa foto asli (*original*) yang telah diubah menjadi bentuk base64 yang belum mengalami pemrosesan lebih lanjut, sehingga memungkinkan pengguna untuk melakukan pemotongan (*crop*) ulang apabila diperlukan. Pendekatan ini bertujuan untuk menjaga fleksibilitas dan kualitas data gambar selama proses pengolahan berlangsung.



Gambar 3.35. Penyimpanan pada etalase di IndexedDB

Gambar 3.35 menunjukkan data metadata yang telah disiapkan dan disimpan setelah proses penyimpanan gambar dilakukan. Metadata ini dirancang agar siap menerima hasil identifikasi dari modul kecerdasan buatan (AI) terhadap gambar yang telah diunggah. Selain itu, struktur metadata tersebut juga berfungsi sebagai skema penyimpanan yang memastikan setiap data dapat tersimpan secara terorganisasi dan konsisten, sesuai dengan kebutuhan yang telah terintegrasi.

#	Key (Key path: "id")	Value
0	1749970612489	{id: 1749970612489, data: 'data:image/png;filename=HDR-20250615-VZCC-starterp_zjpj37', data: 'data:image/png;filename=HDR-20250615-VZCC-starterp_zjpj37', id: 1749970612489, timestamp: '2025-06-15 13:56:52'}
1	1749970612805	{id: 1749970612805, data: 'data:image/png;filename=HDR-20250615-9cEr-starterp_/_HwDA2g'}
2	1749970613096	{id: 1749970613096, data: 'data:image/png;filename=HDR-20250615-W1NR-starterp_vxy94X3'}
3	1749970613387	{id: 1749970613387, data: 'data:image/png;filename=HDR-20250615-dly2-starterp_xPwe6Fn'}
4	1749970613695	{id: 1749970613695, data: 'data:image/png;filename=HDR-20250615-FSRJ-starterp_1ldke91'}
5	1749970614194	{id: 1749970614194, data: 'data:image/png;filename=HDR-20250615-dzIZ-starterp_84TUA+P'}
6	1749970614662	{id: 1749970614662, data: 'data:image/png;filename=HDR-20250615-Sctj-starterp_xfgD8ob'}
7	1749970615223	{id: 1749970615223, data: 'data:image/png;filename=HDR-20250615-3a9E-starterp_mo7F+5r'}
8	1749970615801	{id: 1749970615801, data: 'data:image/png;filename=HDR-20250615-8fpt-starterp_2ucIfHK'}

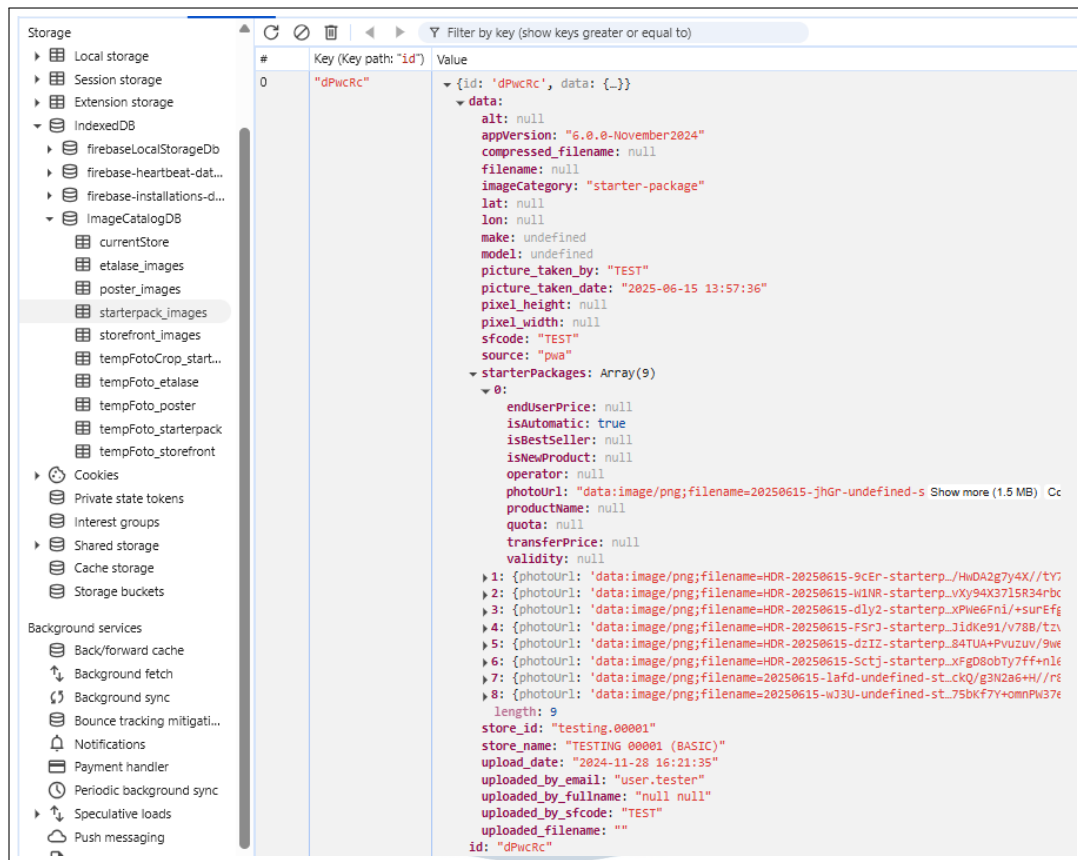
Gambar 3.36. Penyimpanan pada tempFoto_starterpack

Gambar 3.36 memperlihatkan struktur penyimpanan sementara yang direpresentasikan oleh objek tempFoto_starterpack. Objek ini berfungsi sebagai media penyimpanan lokal sementara yang menampung hasil tangkapan gambar yang telah dikonversi ke dalam format base64. Konversi ini dilakukan untuk mempermudah proses penyimpanan data secara lokal pada browser, khususnya dalam konteks penggunaan IndexedDB. Selain itu, penyimpanan *original image* dalam format ini juga ditujukan untuk memberikan fleksibilitas bagi pengguna, apabila diperlukan proses pemotongan (*crop*) ulang pada gambar yang telah diambil, tanpa harus melakukan pengambilan ulang gambar secara fisik.

#	Key (Key path: "id")	Value
0	1749970612489	{id: 1749970612489, data: 'data:image/png;filename=20250615-jhgr-undefined-st_0a5z', data: 'data:image/png;filename=20250615-jhgr-undefined-s', id: 1749970612489, timestamp: '2025-06-15 13:57:17'}
1	1749970615223	{id: 1749970615223, data: 'data:image/png;filename=20250615-lafd-undefined-st_ckQ/', data: 'data:image/png;filename=20250615-lafd-undefined-s', id: 1749970615223, timestamp: '2025-06-15 13:57:33'}
2	1749970615801	{id: 1749970615801, data: 'data:image/png;filename=20250615-w33U-undefined-st_75bk', data: 'data:image/png;filename=20250615-w33U-undefined-s', id: 1749970615801, timestamp: '2025-06-15 13:57:24'}

Gambar 3.37. Penyimpanan pada tempFotoCrop_starterpack di IndexedDB

Gambar 3.37 menunjukkan data hasil pemotongan (*crop*) gambar yang telah disimpan secara terstruktur. Setiap hasil *crop* diidentifikasi dan dikaitkan langsung dengan ID dari foto utama yang diambil sebelumnya. Dengan pendekatan ini, sistem mampu menyimpan versi hasil *crop* tanpa menimpa atau mengubah data gambar asli (*original image*) yang telah tersimpan sebelumnya.



Gambar 3.38. Penyimpanan pada starterpack_images di IndexedDB

Gambar 3.38 menunjukkan data yang telah disusun dalam bentuk metadata, di mana data gambar untuk fitur starterpack disimpan dalam struktur array. Hal ini disesuaikan dengan karakteristik fitur tersebut yang memungkinkan pengambilan beberapa gambar dalam satu sesi. Setiap elemen dalam array merepresentasikan satu gambar hasil tangkapan, dan masing-masing gambar dilengkapi dengan metadata tersendiri. Pendekatan ini digunakan untuk memastikan bahwa setiap gambar dapat diidentifikasi, dikelola, dan diproses secara individual sesuai dengan kebutuhan sistem.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

D. Pengujian Sistem

Berikut adalah hasil testing yang dilakukan oleh mentor selama melakukan pelaksanaan kerja magang, sebagai berikut:

Tabel 3.3. Tabel pengujian Menu Poster, Etalase dan Tampak Depan

Skenario Pengujian	Perkiraan Hasil	Hasil	Kesimpulan
Pemilihan Menu dan Store	Sistem menyimpan pilihan ke dalam CurrentStore di IndexedDB	sesuai	sukses
Pengambilan Gambar	Gambar disimpan ke tempFoto_poster (penyimpanan sementara)	sesuai	sukses
Kamera gagal diakses	Sistem menampilkan pesan error	sesuai	sukses
Proses Cropping Gambar	Gambar ter-crop dengan benar dan siap disimpan	sesuai	sukses
Navigasi ke Halaman Hasil	Halaman hasil menampilkan gambar beserta informasi terkait	sesuai	sukses

Tabel 3.4. Tabel pengujian Menu Kartu Perdana

Skenario Pengujian	Perkiraan Hasil	Hasil	Kesimpulan
Pemilihan Menu dan Store	Sistem menyimpan pilihan ke dalam CurrentStore di IndexedDB	sesuai	sukses
Pengambilan Gambar	Gambar disimpan ke tempFoto_starterpack (penyimpanan sementara)	sesuai	sukses

Bersambung ke halaman berikutnya

Lanjutan dari Tabel 3.4

Skenario Pengujian	Perkiraan Hasil	Hasil	Kesimpulan
Kamera gagal diakses	Sistem menampilkan pesan error	sesuai	sukses
Proses Cropping Gambar	Gambar hasil crop disimpan ke tempFotoCrop_starterpack	sesuai	sukses
Navigasi ke Halaman Hasil	Halaman hasil menampilkan gambar beserta informasi terkait	sesuai	sukses

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang yang dilakukan setiap minggu diuraikan pada tabel 3.5.

Tabel 3.5. Uraian pekerjaan tiap minggu selama pelaksanaan kerja magang

Minggu Ke-	Uraian Pekerjaan
1	Belajar secara mandiri melalui google partner skillboost.
2	Mempelajari kerangka kerja <i>FastAPI</i> melalui kode yang telah disiapkan oleh pengembang senior, serta menerima bimbingan secara langsung dari mentor di PT Devoteam Cloud Services.
3	Mengerjakan proyek HRIS Devoteam, mencakup integrasi antara sisi frontend dan backend melalui pemanfaatan endpoint yang tersedia, serta implementasi fitur seperti <i>dashboard</i> pengguna, formulir cuti tahunan, saldo cuti, dan sistem autentikasi menggunakan Google Auth.
4-5	Melanjutkan pengembangan HRIS pada bagian <i>middleware</i> autentikasi login dan fitur persetujuan/penolakan cuti oleh admin. Selain itu, melakukan perbaikan bug pada sisi frontend BNI Diva, menyusun dokumentasi Swagger untuk seluruh endpoint, dan mengembangkan fitur pelaporan masalah baik dari sisi logika backend, tampilan pengguna, maupun antarmuka admin.

Bersambung ke halaman berikutnya

Lanjutan dari Tabel 3.5

Minggu Ke-	Uraian Pekerjaan
6	Mengembangkan <i>dashboard</i> admin untuk HRIS serta merancang formulir penyampaian informasi dari admin ke pengguna. Pada minggu yang sama, memulai pengembangan fitur kamera, termasuk fungsi pembesaran <i>zoom</i> dan pemotongan <i>crop</i> gambar pada proyek Retina Telkomsel.
7	Melakukan refaktor kode backend proyek BNI Diva agar sesuai dengan struktur kode yang lebih efisien, mudah dipelihara, dan terstruktur dengan baik.
8–9	Mengembangkan fitur “Kartu Perdana” pada proyek Retina, yaitu fitur yang memungkinkan pengguna untuk mengambil beberapa gambar dalam satu sesi pengambilan, serta pengelolaan hasil tangkapan tersebut dalam struktur data yang sesuai.
10	Mengembangkan antarmuka pengguna (UI) untuk proses verifikasi data dan survei pada aplikasi Retina. Selain itu, melakukan instalasi Flutter dan emulator, serta merancang antarmuka login dan registrasi menggunakan Flutter.
11	Menyusun dokumentasi Swagger untuk autentikasi dan skema data. Melakukan perbaikan bug pada sistem HRIS.
12	Melanjutkan pembelajaran mandiri melalui Google Partner Skillboost serta mempelajari Express.js dengan PostgreSQL. Melakukan perbaikan bug pada modul <i>dashboard</i> , formulir, dan proses persetujuan admin di HRIS. Menyusun dokumen pengujian regresi (regression test) untuk sistem BNI LDAP.
13	Melanjutkan penyusunan dokumen pengujian regresi untuk BNI LDAP. Melakukan perbaikan bug pada fitur <i>Audit Trail</i> , komponen <i>sidebar</i> , serta penerapan batasan pencarian pada sisi frontend dan backend proyek BNI Diva.
14	Melakukan pembaruan terhadap <i>test case</i> pada proyek BNI Diva serta perbaikan bug berdasarkan temuan dari pengujian pengguna. Juga dilakukan pengujian internal terhadap sistem HRIS Devoteam.

Bersambung ke halaman berikutnya

Lanjutan dari Tabel 3.5

Minggu Ke-	Uraian Pekerjaan
15	Melakukan pembaruan terhadap fitur kamera pada proyek Retina, serta perbaikan bug pada BNI Diva sesuai dengan hasil uji coba pengguna. Di samping itu, dilakukan perbaikan peran admin pada HRIS Devoteam dan pembaruan data saldo karyawan pada basis data PostgreSQL.
15	Melakukan migrasi penyimpanan dari <i>localStorage</i> ke IndexedDB pada proyek Retina untuk mendukung penyimpanan gambar beresolusi tinggi secara lebih efisien.
16	Melanjutkan proses migrasi dari <i>localStorage</i> ke IndexedDB pada proyek Retina. Di samping itu, dilakukan perbaikan bug terkait durasi cuti tahunan, serta pengembangan formulir penambahan pengguna dengan kemampuan untuk mengirim data ke dua endpoint secara bersamaan. Perbaikan bug juga dilakukan pada fitur kamera proyek Retina.

3.4 Spesifikasi Sistem

Berikut adalah perangkat lunak dan perangkat keras yang digunakan sebagai penunjang dalam pelaksanaan magang dan pengerjaan proyek magang. Berikut perangkat lunak yang gunakan berupa:

1. Google Chrome
2. Visual Studio Code
3. Postman
4. PostgreSQL

Adapun perangkat keras yang digunakan, yaitu:

1. Operating System: Windows
2. Processor: Intel Core i7-12700H
3. GPU: RTX 3050
4. RAM: 16 GB DDR 5
5. Storage: 1 TB SSD

3.5 Kendala dan Solusi yang Ditemukan

3.5.1 Kendala

Berikut merupakan beberapa kendala yang dihadapi selama proses pengembangan fitur-fitur pada project Retina Telkomsel:

1. Sulit menemukan pendekatan yang efektif untuk meningkatkan performa kamera berbasis web agar mampu menghasilkan gambar dengan resolusi tinggi dan tajam.
2. Proses crop dan zoom menyebabkan penurunan kualitas gambar secara signifikan, sehingga gambar menjadi kurang optimal untuk proses identifikasi.
3. localStorage hanya mendukung penyimpanan data dalam ukuran terbatas (sekitar 5 MB per domain), sehingga tidak dapat mengakomodasi file gambar dengan resolusi tinggi dalam jumlah banyak.

3.5.2 Solusi

1. Pengambilan gambar dilakukan langsung melalui elemen HTML menggunakan `<canvas>` dan `<video>` tanpa menggunakan pustaka react-webcam, sehingga menghasilkan gambar dengan ukuran rata-rata yang lebih optimal dan tingkat kejernihan yang lebih baik.
2. Pengolahan gambar menggunakan *library* react-image-crop dan canvas memungkinkan pengguna memotong gambar dan juga menggunakan canvas untuk memperbesar gambar tanpa kehilangan kualitas secara drastis, sekaligus menjaga proporsi asli gambar.
3. IndexedDB diimplementasikan sebagai solusi penyimpanan gambar dan metadata karena mendukung penyimpanan data dalam jumlah besar.