BAB I

PENDAHULUAN

A. Latar Belakang

Bawang merah (*Allium cepa L. var. aggregatum*) merupakan tanaman hortikultura strategis yang memberikan kontribusi signifikan bagi ekonomi nasional. Selain perannya yang penting sebagai bahan pokok dalam berbagai hidangan khas Indonesia, komoditas ini bernilai tinggi dan berpotensi untuk diekspor (Lubis et al., 2022). Di wilayah Sulawesi Selatan, Kabupaten Enrekang dikenal sebagai sentra budidaya bawang merah dan memiliki potensi pertanian yang menjanjikan di sektor ini. Menjaga kestabilan pasokan dan harga bawang merah, memerlukan sistem pengelolaan yang efektif, khususnya dalam aspek monitoring pasokan.

Kurangnya sistem monitoring pasokan bawang merah yang terintegrasi di Kabupaten Enrekang menyebabkan berbagai permasalahan yang berdampak pada sektor pertanian. Proses pencatatan dan pengawasan yang masih dilakukan secara manual seringkali menghasilkan data yang tidak akurat, sehingga menyulitkan pemangku kepentingan untuk memprediksi ketersediaan stok bawang merah di pasaran. Ketidakpastian data ini dapat memicu fluktuasi harga yang tidak terkendali, kerugian petani akibat *over*produksi, atau kesulitan distribusi akibat minimnya informasi stok di daerah lain (Marina et al., 2024). Koordinasi antara petani, distributor, dan pemerintah daerah sering kali terhambat akibat kurangnya transparansi dan keterlambatan dalam pengelolaan informasi pasokan. Situasi ini

menunjukkan perlunya sistem yang dapat memperbaiki efisiensi pengelolaan pasokan bawang merah secara menyeluruh.

Dampak dari ketidakoptimalan monitoring pasokan ini juga berimbas pada ketahanan pangan daerah. Ketika distribusi bawang merah tidak berjalan lancar, pasokan di beberapa wilayah dapat menipis, mengakibatkan lonjakan harga yang memberatkan konsumen, khususnya masyarakat berpenghasilan rendah. Petani, yang seharusnya mendapat manfaat dari hasil panen mereka, sering kali merugi akibat hasil produksi yang tidak terserap dengan baik oleh pasar (Ayuni, 2024). Kurangnya akses terhadap data yang akurat juga menghambat pemerintah dalam merancang kebijakan yang mendukung sektor pertanian bawang merah. Dengan meningkatnya kebutuhan bawang merah di tingkat nasional, permasalahan ini menjadi semakin mendesak untuk segera ditangani melalui inovasi berbasis teknologi.

Aplikasi ini dirancang untuk menyediakan *platform* digital yang mempermudah proses pencatatan, pemantauan, dan analisis data pasokan secara lebih efektif dan efisien. Informasi yang tersedia secara *real-time* di dalam aplikasi diharapkan mampu membantu petani, distributor, maupun pemerintah daerah dalam mengambil keputusan strategis yang berbasis data akurat. Dengan adanya aplikasi ini, potensi kendala yang sering terjadi dalam pengelolaan pasokan dapat diminimalisir.

Aplikasi ini memiliki beberapa keunggulan yang dapat berdapak positif bagi mengkoordinasi pasokan bawang merah di Kabupaten Enrekang. Dengan menyediakan fitur-fitur seperti pelacakan ketersediaan waktu nyata, perekaman

informasi secara elektronik, dan memanfaatkan analisis prediktif, aplikasi ini tidak hanya meningkatkan efisiensi manajemen tetapi juga mempercepat proses pengambilan keputusan berdasarkan data yang akurat. Harapannya, solusi ini dapat meningkatkan kesejahteraan petani bawang merah, menjaga kestabilan harga, serta memperkuat posisi Kabupaten Enrekang sebagai sentra bawang merah yang kompetitif di tingkat nasional maupun internasional (Harahap et al., 2023).

B. Rumusan Masalah

Berdasarkan latar belakang sebelumnya, disimpulkan permasalahan utama dalam penelitian ini adalah bagaimana membuat aplikasi berbasis android yang dapat memantau secara efektif distribusi dan ketersediaan bawang merah di Kabupaten Enrekang?

C. Tujuan

Studi ini bertujuan untuk mengembangkan aplikasi Android untuk memantau pasokan bawang merah di Kabupaten Enrekang guna memudahkan pengelolaan dan distribusi bawang merah di wilayah tersebut.

D. Batasan Masalah

Untuk memastikan bahwa penelitian ini bergerak tidak melampaui kerangka kerja yang telah ditetapkan, beberapa batasan diperlukan meliputi:

1. Hanya mencakup fitur utama seperti pencatatan pasokan dan pemantauan stok, tanpa melibatkan fitur yang lebih kompleks seperti analisis prediktif atau integrasi dengan sistem lain.

2. Aplikasi hanya ditujukan untuk pengguna terbatas di Kabupaten Enrekang, tidak untuk penggunaan skala besar.
3. Pengujian hanya dilakukan pada pengguna awal yang terbatas, tanpa melibatkan uji skala besar atau dalam kondisi yang sangat variatif.

E. Manfaat Penelitian

Manfaat dalam penelitian ini diuraikan sebagai berikut:

1. Petani

   Mempermudah pemantauan pasokan, meningkatkan pendapatan, dan mengurangi kerugian akibat overproduksi.
2. Distributor

   Mempercepat distribusi, meningkatkan efisiensi stok, dan mendukung transparansi data pasokan.
3. Pemerintah daerah

   Membantu perencanaan kebijakan, memantau pasokan secara terintegrasi, dan menjaga stabilitas harga serta ketahanan pangan.
4. Konsumen

   Menjamin ketersediaan bawang merah dengan harga stabil.
5. Bidang Pendidikan

   Menjadi referensi penelitian dan inovasi dalam penerapan teknologi di sektor pertanian.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Aplikasi

Aplikasi adalah perangkat lunak yang menjalankan aktivitas atau layanan tertentu pada perangkat digital seperti telepon pintar, komputer, atau tablet. Secara umum, aplikasi berfungsi sebagai alat yang membantu pengguna dalam menjalankan tugas atau aktivitas tertentu melalui sistem komputer (Zalukhu et al., 2023). Teknologi aplikasi telah berkembang pesat seiring dengan meningkatnya kebutuhan pengguna akan kemudahan dan efisiensi dalam berbagai aspek kehidupan. Dalam ekosistem Android, pengembangan aplikasi biasanya dilakukan menggunakan bahasa pemrograman seperti *Kotlin* atau *Java*, dengan *Android Studio* berfungsi sebagai lingkungan kerja utama untuk merancang, menulis, dan menguji kode aplikasi.

Peran aplikasi dalam kehidupan sehari-hari menjadi semakin penting, terutama untuk memecahkan masalah spesifik di berbagai bidang, termasuk pertanian. Pengguna dapat mengakses aplikasi melalui antarmuka yang dirancang ramah pengguna, memungkinkan interaksi yang mudah dan efisien. Sistem ini juga dapat dilengkapi dengan berbagai fitur, seperti pemrosesan data, penyimpanan berbasis cloud, dan integrasi dengan perangkat keras lainnya. Aplikasi monitoring pasokan bawang merah, misalnya, dapat memberikan solusi praktis untuk memantau data

pasokan secara *real-time* sekaligus menyediakan laporan analisis yang bermanfaat bagi petani, distributor, dan pemangku kepentingan lainnya.

2. Sistem Monitoring

Sistem monitoring adalah tahapan pemantauan yang dirancang untuk mengontrol suatu kegiatan dilaksanakan selaras dengan spesifikasi manajemen organisasi. Pemantauan juga berfungsi untuk menilai sejauh mana kegiatan yang dilaksanakan konsisten dengan rencana awal (Husna et al., 2021). Melalui monitoring, manajemen dapat mengidentifikasi sejauh mana program atau kegiatan berjalan sesuai target, sehingga meminimalisir penyimpangan dan memastikan efisiensi dalam penggunaan sumber daya. Proses ini umumnya melibatkan pengumpulan data, pemantauan perkembangan, serta analisis informasi untuk mengukur kinerja suatu kegiatan atau proyek.

Monitoring yang efektif memerlukan pendekatan sistematis dan berkelanjutan. Langkah-langkah dalam proses ini meliputi:

1. Perencanaan indikator kinerja
2. Pengumpulan data secara berkala
3. Evaluasi hasil berdasarkan parameter yang telah ditetapkan sebelumnya.

Teknologi informasi dan perangkat lunak berbasis data kini sering digunakan untuk mendukung kegiatan monitoring, karena mampu memberikan laporan yang tepat akurat. Dengan adanya teknologi, organisasi dapat mengakses data *real-time* dan mengefisienkan pengambilan keputusan berdasarkan fakta yang ada.

Peran sistem monitoring sangat penting dalam menjaga kualitas, transparansi, dan akuntabilitas dalam sebuah organisasi. Monitoring berperan lebih dari

mengukur berhasilnya program, tetapi juga dapat menemukan hambatan serta memberikan solusi dalam proses perbaikan. Evaluasi yang berkelanjutan akan membantu organisasi merumuskan strategi yang lebih strategis untuk mencapai tujuan dalam waktu dekat dan jangka panjang. Dengan demikian, sistem monitoring bukan hanya alat pengawasan, tetapi juga fondasi untuk peningkatan kualitas kinerja organisasi secara keseluruhan.

Selain itu, sistem monitoring juga dapat diterapkan di berbagai sektor, seperti pendidikan, kesehatan, lingkungan, dan bisnis. Dalam sektor pendidikan, monitoring digunakan untuk mengevaluasi kemajuan siswa, efektivitas kurikulum, serta kinerja tenaga pengajar. Di sektor kesehatan, sistem ini membantu dalam memantau distribusi obat, pelaksanaan program kesehatan masyarakat, hingga pengendalian penyakit menular. Pada sektor lingkungan, monitoring sering digunakan untuk mengawasi kualitas udara, air, serta dampak aktivitas manusia terhadap ekosistem**.**

Kecanggihan teknologi *Internet of Things* (IoT) dan kecerdasan buatan (AI) telah membantu pengembangan sistem monitoring yang lebih canggih. Dengan IoT, sensor dan perangkat pintar dapat digunakan untuk mengumpulkan data secara otomatis dan *real-time*, sementara AI membantu dalam analisis data yang kompleks serta memberikan rekomendasi berbasis pola dan prediksi**.** Hal ini memungkinkan organisasi untuk lebih responsif dalam menghadapi tantangan, serta meningkatkan efisiensi dan efektivitas proses monitoring.

Dengan integrasi teknologi ini, sistem monitoring semakin menjadi bagian penting dalam pengelolaan organisasi modern. Penerapan teknologi canggih dalam

monitoring diperkirakan akan semakin luas kedepannya, mengoptimalkan proses pengambilan keputusan dan meningkatkan keunggulan kompetitif organisasi dalam menghadapi dinamika global.

3. Android



Gambar 2. 1 Ikon *Android*

Android adalah sistem operasi berbasis *Linux* yang dirancang khusus untuk perangkat seluler seperti telepon pintar. Platform ini menawarkan kebebasan bagi pengembang untuk membuat berbagai macam aplikasi. Awalnya dikembangkan oleh *Android Inc*., sistem ini diakuisisi oleh *Google* pada tahun 2005. Karena Android berbasis pada kernel *Linux*, sistem ini bersifat sumber terbuka dan memungkinkan pengembang dan produsen perangkat melakukan kustomisasi dan adaptasi yang luas (R. Gunawan et al., 2021). Dengan sifatnya yang *open-source*, Android berkembang sebagai sistem operasi *mobile* yang mendunia, dengan berbagai perangkat menggunakan sistem ini di berbagai merek dan model.

Android menawarkan berbagai fitur unggulan, termasuk kemampuan multitasking, integrasi aplikasi yang luas, serta akses ke jutaan aplikasi melalui *Google Play Store*. Sistem ini mendukung pengembangan aplikasi dengan bahasa pemrograman seperti *Java*, *Kotlin*, dan *C++*. Selain itu, Android juga mendukung integrasi dengan layanan Google, seperti Google Maps, Gmail, dan Google Drive, yang memperluas fungsionalitasnya.

Salah satu alasan utama popularitas Android adalah fleksibilitasnya untuk diadaptasi oleh berbagai produsen perangkat keras, seperti Samsung, Xiaomi, Oppo, dan banyak lainnya. Selain itu, antarmuka pengguna yang intuitif dan kemampuan untuk menyesuaikan fitur sesuai preferensi pengguna menjadikannya pilihan utama di pasar global. Android juga mendukung berbagai teknologi modern, seperti konektivitas 5G, *Internet of Things* (IoT), dan aplikasi berbasis *Artificial Intellegent* (AI), sehingga terus relevan dengan kebutuhan teknologi yang terus berkembang.

Sebagai sistem operasi, Android juga telah diimplementasikan dalam perangkat lain seperti televisi pintar (Android TV), perangkat otomotif *(Android Auto*), dan perangkat *wearable* (Wear OS). Dengan ekosistem yang terus berkembang, Android menjadi platform yang mendominasi pasar perangkat bergerak, menawarkan solusi komprehensif bagi pengguna dan pengembang untuk memenuhi kebutuhan sehari-hari maupun keperluan teknologi tingkat lanjut.

Android juga memiliki sistem keamanan yang terus ditingkatkan, dengan pembaruan rutin untuk menangkal ancaman siber dan menjaga privasi pengguna. Google Play Protect, sebagai contoh, berfungsi untuk memindai aplikasi dari potensi malware sebelum diunduh oleh pengguna, serta memantau aktivitas aplikasi yang telah terpasang di perangkat. Selain itu, Android menyediakan opsi untuk enkripsi perangkat, kontrol izin aplikasi yang lebih baik, dan fitur keamanan biometrik seperti sidik jari dan pengenalan wajah, sehingga memberikan rasa aman bagi pengguna.

Dengan komunitas pengembang yang besar dan dukungan aktif dari Google, Android terus berinovasi untuk memenuhi kebutuhan zaman. Contohnya adalah pengembangan *Android Open Source Project* (AOSP), yang memungkinkan pengembang memanfaatkan kode sumber untuk membuat sistem operasi khusus yang sesuai dengan perangkat tertentu atau kebutuhan spesifik**.** Inisiatif ini mempercepat perkembangan teknologi dan menjadikan Android sebagai platform yang adaptif pada beragam sektor, mulai dari kesehatan bisnis, dan pendidikan.

4. *Visual Studio Code*



Gambar 2. 2 *Visual Studio Code*

Visual Studio Code (VS Code) berupa editor teks sumber terbuka gratis dari *Microsoft*. Sejak diluncurkan pada tahun 2015, *VS Code* telah menikmati popularitas di seluruh dunia dan termasuk salah satu alat penyuntingan kode yang paling ramai digunakan. Editor ini didasarkan pada kerangka kerja *Electron* dan berjalan di berbagai sistem operasi, termasuk *Windows*, *macOS*, dan *Linux*, dengan tetap mempertahankan tampilan yang konsisten. Meskipun memiliki nama yang mirip dengan *Visual Studio IDE*, *VS Code* adalah aplikasi yang terpisah dan dirancang khusus sebagai editor kode yang ringan namun *powerful* (A. Gunawan, 2024).

Salah satu keunggulan utama *VS Code* adalah fitur *IntelliSense* yang cerdas. Fitur ini menyediakan fitur-fitur seperti pelengkapan otomatis, petunjuk parameter,

keterangan alat, dan daftar anggota, semuanya disesuaikan berdasarkan jenis variabel, definisi fungsi, dan modul yang digunakan dalam kode. *IntelliSense* bekerja dengan sangat baik untuk berbagai bahasa pemrograman dan dapat diperluas melalui ekstensi. Sistem ini menganalisis kode secara *real-time* dan memberikan saran yang relevan, membantu developer menulis kode dengan lebih cepat dan mengurangi kesalahan sintaks. Fitur ini juga mendukung snippet kode yang dapat disesuaikan untuk mempercepat penulisan kode yang berulang.

Debugging merupakan aspek penting dalam pengembangan *software*, dan *VS Code* menyediakan debugger terintegrasi yang powerful. Editor ini mendukung debugging untuk berbagai bahasa pemrograman dengan fitur-fitur seperti *breakpoint*, *variable inspection*, *call stack navigation*, dan *interactive console*. Developer dapat mengatur breakpoint dengan mudah, menjalankan kode step-by-step, dan memeriksa nilai variabel pada titik tertentu dalam eksekusi program. Debugger *VS Code* juga mendukung debugging aplikasi yang berjalan di container Docker dan dapat terhubung ke aplikasi yang berjalan di *server remote*.

Integrasi Git yang seamless menjadi salah satu fitur yang paling dihargai oleh developer. *VS Code* menyediakan antarmuka grafis yang intuitif untuk operasi Git seperti commit, push, pull, merge, dan branch management. Panel *Source Control* menampilkan perubahan file secara visual, memungkinkan developer untuk melihat diff, stage changes, dan menulis commit message tanpa meninggalkan editor. Fitur ini juga mendukung resolusi konflik merge dengan antarmuka visual yang memudahkan developer untuk mengatasi konflik kode. Timeline view

menampilkan riwayat perubahan file dan memungkinkan developer untuk melihat evolusi kode dari waktu ke waktu.

5. *PhpMyAdmin*



Gambar 2. 3 *phpMyAdmin*

phpMyAdmin adalah aplikasi web gratis dan *open-source* yang ditulis dalam bahasa PHP, berfungsi sebagai antarmuka administrasi untuk database MySQL dan MariaDB. Aplikasi ini dikembangkan di tahun 1998, yang kini menjadi tools ter populer untuk mengelola database MySQL melalui web browser. *phpMyAdmin* menyediakan interface grafis yang user-friendly, memungkinkan pengguna untuk melakukan berbagai operasi database tanpa perlu mengetik perintah SQL secara manual (Satria et al., 2023). Tool ini sangat populer di kalangan web developer, database administrator, dan siapa saja yang bekerja dengan MySQL database karena kemudahan penggunaan dan aksesibilitasnya.

Fitur utama *phpMyAdmin* mencakup manajemen database yang komprehensif melalui *interface web*. Pengguna dapat membuat, menghapus, dan memodifikasi database serta tabel dengan mudah menggunakan form dan wizard yang tersedia. *phpMyAdmin* menyediakan editor SQL yang memungkinkan eksekusi query langsung, serta *query builder* visual untuk membuat query tanpa menulis SQL. Fitur import dan export memungkinkan transfer data dalam berbagai format seperti SQL,

CSV, XML, dan JSON. Tool ini juga menyediakan manajemen user dan privileges, memungkinkan administrator untuk mengatur akses pengguna ke database dengan kontrol yang granular.

Kemampuan manajemen data dalam *phpMyAdmin* sangat lengkap dan user-friendly. Pengguna dapat menambah, mengedit, dan menghapus records melalui form yang otomatis dihasilkan berdasarkan struktur tabel. Fitur pencarian advanced memungkinkan filtering data dengan multiple criteria dan operator. Sorting dan pagination membantu navigasi data dalam tabel besar. *phpMyAdmin* juga menyediakan fitur copy dan move data antar tabel atau database. Untuk operasi bulk, tersedia fitur untuk select multiple records dan melakukan operasi batch seperti delete atau export.

6. *Java* Script



Gambar 2. 4 *JavaScript*

JavaScript adalah bahasa pemrograman tingkat tinggi, interpretatif, dan dinamis yang pertama kali dikembangkan oleh Brendan Eich pada tahun 1995 saat bekerja di Netscape. Dirancang untuk meningkatkan interaktivitas pada halaman web, bahasa ini menjadi bahasa pemrograman yang paling ramai diaplikasikan di berbagai platform. Meskipun namanya menyiratkan hubungan dengan *Java*, *Java*Script sebenarnya adalah bahasa yang berbeda dengan sintaks yang terinspirasi dari C dan *Java*. *Java*Script dirancang sebagai bahasa scripting untuk web browser,

memungkinkan interaktivitas dan manipulasi dinamis pada halaman web. Seiring perkembangannya, *Java*Script telah berkembang menjadi bahasa pemrograman full-stack yang dapat digunakan tidak hanya di browser (client-side) tetapi juga di server (server-side) melalui platform seperti Node.js. Standarisasi *Java*Script dilakukan melalui ECMAScript (ES), dengan versi terbaru seperti ES2015 (ES6), ES2017, ES2018, dan seterusnya yang terus menambahkan fitur-fitur baru dan modern.

Karakteristik fundamental *Java*Script mencakup beberapa paradigma pemrograman yang membuatnya unik dan fleksibel. *Java*Script adalah bahasa yang weakly typed dan dynamic, artinya variabel tidak perlu dideklarasikan dengan tipe data spesifik dan dapat berubah tipe selama runtime. Bahasa ini mendukung first-class functions, dimana function dapat diperlakukan sebagai nilai, disimpan dalam variabel, diteruskan sebagai argumen, dan dikembalikan dari function lain. *Java*Script juga mendukung object-oriented programming melalui prototype-based inheritance, berbeda dari class-based inheritance yang umum di bahasa lain (Supardi, 2021). Closures merupakan fitur powerful yang memungkinkan function untuk mengakses variabel dari scope luar bahkan setelah function tersebut selesai dieksekusi. Event-driven programming model memungkinkan *Java*Script untuk merespons user interactions dan sistem events secara asynchronous.

Sintaks *Java*Script relatif mudah dipelajari dengan struktur yang familiar bagi programmer dari bahasa C-family. Dalam *Java*Script, variabel dapat dibuat menggunakan kata kunci var, let, atau const. Setiap kata kunci ini memiliki properti yang berbeda terkait cakupan dan mutabilitas. Bahasa ini juga mendukung berbagai

tipe data, mulai dari tipe primitif seperti angka, teks, Boolean, undefined, null, simbol, dan bigint hingga struktur data kompleks seperti objek dan array. Control structures seperti if-else, for loops, while loops, dan switch statements bekerja secara familiar. Function dapat dideklarasikan menggunakan function declaration, function expression, atau arrow function syntax yang diperkenalkan di ES6. Template literals dengan backticks memungkinkan string interpolation dan multi-line strings. Destructuring assignment memudahkan ekstraksi nilai dari array dan object, sementara spread operator dan rest parameters memberikan fleksibilitas dalam menangani collections dan function parameters.

7. Extensible Markup Language (XML)



Gambar 2. 5 *Extensible Markup Language*

Extensible Markup Language (XML) merupakan format markup serbaguna yang mirip dengan HTML, namun dirancang khusus untuk menyusun, menyimpan, dan mentransfer data secara terstruktur antar sistem atau aplikasi (Septiani et al., 2020). Berbeda dengan HTML yang fokus pada penampilan, XML lebih fokus pada struktur dan representasi data, memungkinkan pengguna untuk mendefinisikan *tag* kustom sesuai kebutuhan. XML digunakan untuk mentransfer data antara sistem yang berbeda dengan cara yang terstandarisasi, sehingga memudahkan integrasi antar aplikasi (Rosyidi et al., n.d.). XML menyimpan informasi dalam bentuk teks

yang tidak hanya mudah dipahami oleh manusia, tetapi juga dapat diolah secara efisien oleh komputer. Karena sifatnya yang fleksibel, XML banyak dimanfaatkan dalam berbagai konteks, seperti pengaturan konfigurasi aplikasi, pertukaran data antar sistem, serta integrasi layanan berbasis web.

XML memiliki keunggulan utama dalam hal fleksibilitas, karena pengguna dapat mendefinisikan elemen dan atribut sesuai kebutuhan aplikasi. Struktur XML yang hierarkis memungkinkan data disusun dalam format pohon, yang membuatnya mudah untuk diproses, dicari, dan dikelola. Selain itu, XML bersifat *platform-independent*, artinya data yang disimpan dalam format XML dapat dipindahkan antar sistem operasi atau perangkat tanpa mengalami perubahan, sehingga mempermudah pertukaran data antar aplikasi yang berjalan di *platform* yang berbeda.

Dalam konteks web, XML sering digunakan dalam berbagai standar, seperti RSS untuk umpan berita atau SOAP untuk komunikasi layanan web. XML juga sering digunakan dalam file konfigurasi, penyimpanan data, dan pengolahan informasi dalam aplikasi berbasis web atau perangkat lunak. Dengan kemampuannya untuk mendukung data terstruktur dan kompatibilitas yang tinggi antar sistem, XML tetap menjadi sebuah format utama dalam pengoptimalan perangkat lunak serta pertukaran fakta antar *platform*.

8. Xampp



Gambar 2. 6 Xampp

XAMPP adalah aplikasi tanpa biaya serta *open-source* yang dioptimalkan oleh *Apache Friends*, dirancang untuk memudahkan developer dalam mengatur lingkungan pengembangan web lokal. Dalam *Java*Script, variabel dapat dibuat menggunakan kata kunci var, let, atau const. Setiap kata kunci ini memiliki properti yang berbeda terkait cakupan dan mutabilitas. Bahasa ini juga mendukung berbagai tipe data, mulai dari tipe primitif seperti angka, teks, Boolean, undefined, null, simbol, dan bigint hingga struktur data kompleks seperti objek dan array (Cross-border & 2022, n.d.). XAMPP hadir di sejumlah sistem operasi mencakup Windows, Linux, dan macOS, memungkinkan developer untuk memiliki lingkungan pengembangan yang konsisten di berbagai platform.

Apache HTTP Server merupakan komponen inti dari XAMPP yang berfungsi sebagai web server. Apache salah satu web server paling stabil, dipakai oleh jutaan website. Dalam konteks XAMPP, Apache dikonfigurasi untuk berjalan di port 80 (HTTP) dan 443 (HTTPS) secara default, memungkinkan developer untuk mengakses aplikasi web mereka melalui localhost atau 127.0.0.1. Server ini mendukung berbagai modul dan fitur seperti URL rewriting, virtual hosts, SSL/TLS encryption, dan kompresi content. Apache di XAMPP sudah dikonfigurasi untuk bekerja optimal dengan PHP dan menyediakan direktori htdocs sebagai document root tempat file-file web disimpan. Developer juga dapat mengonfigurasi virtual hosts untuk menjalankan multiple website dalam satu instalasi XAMPP.

9. *Unified Modeling Language* (UML)

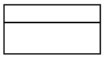
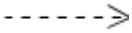
Unified Modeling Language (UML) merupakan standar visual yang dimanfaatkan dalam merepresentasikan struktur dan perilaku sistem, terutama

dalam pengembangan aplikasi. Dioptimalkan oleh *Object Management Group* (OMG) pada tahun 1997, UML telah diadopsi secara luas sebagai alat bantu untuk merancang, menganalisis, dan mendokumentasikan sistem, baik dalam konteks rekayasa perangkat lunak maupun pemodelan proses bisnis dan sistem lainnya yang memerlukan pendekatan grafis (Margaretha et al., 2023).

UML menawarkan beragam jenis diagram yang berfungsi untuk mem*visualisasi*kan berbagai dimensi dari sebuah sistem, mulai dari aspek strukturnya, perilakunya, hingga cara elemen-elemen di dalamnya saling berinteraksi. Diagram-diagram ini dikelompokkan ke dalam dua kelompok besar: diagram struktur dan diagram perilaku. Diagram struktur berfokus pada bagian-bagian tetap dari sistem, seperti hubungan antara kelas, objek, atau komponen, yang menunjukkan keterkaitan antar bagian dalam arsitektur sistem. Diagram perilaku, di sisi lain, menggambarkan bagaimana elemen-elemen dalam sistem berinteraksi dan bergerak seiring waktu, seperti diagram urutan, diagram aktivitas, dan diagram keadaan.

Salah satu keuntungan utama UML adalah kemampuannya untuk menyederhanakan komunikasi antar anggota tim pengembangan. Dengan menggunakan representasi visual, pengembang, analis, *desainer*, dan pemangku kepentingan lainnya dapat lebih mudah memahami struktur dan fungsionalitas sistem yang sedang dikembangkan. Selain itu, UML memungkinkan dokumentasi yang lebih jelas dan terstruktur mengenai keputusan desain dan implementasi sistem.

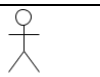
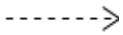
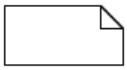
Tabel 2. 1 *Simbol Class Diagram*

No	Gambar	Nama	Keterangan
1		*Association*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek
5		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*

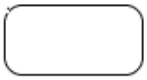
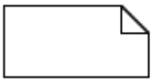
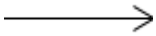
Tabel 2. 2 *Simbol Sequence Diagram*

No	Gambar	Nama	Keterangan
1		*Life Line*	Objek entitas: Antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi komunikasi di antara objek yang mencakup informasi tentang aktivitas yang sedang berlangsung.
3		*Message*	Spesifikasi komunikasi di antara objek yang mencakup informasi tentang aktivitas yang sedang berlangsung.

Tabel 2. 3 Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		*Actor*	Menetapkan fungsi yang dijalankan pengguna ketika berinteraksi dengan skenario penggunaan.
2		*Use Case*	Menjelaskan rangkaian langkah yang dilaksanakan oleh sistem yang menghasilkan hasil yang terukur untuk pihak terkait
3		*Dependency*	Hubungan di kondisi modifikasi pada elemen yang mandiri berdampak pada elemen yang bergantung dan yang tidak bergantung.
4		*Include*	Menetapkan bahwa skenario penggunaan tujuan memperluas perilaku skenario penggunaan asal pada beberapa titik tertentu.
5		*Generalization*	Menetapkan bahwa skenario penggunaan asal bersifat jelas.
6		*Extend*	Apa yang menghubungkan satu objek dengan objek lainnya.
7		*Association*	Menetapkan paket yang menggambarkan sistem dengan cara terbatas.
8		*Note*	Elemen fisik yang ada saat aplikasi berfungsi yang mewakili sumber daya komputasi
9		*Collaboration*	Hubungan antara aturan dan elemen lain yang bekerja sama untuk menciptakan hasil yang lebih besar daripada total elemen ter (sinergi).

Tabel 2. 4 Simbol *State Chart Diagram*

No	Gambar	Nama	Keterangan
1		*State*	Nilai atribut dan nilai *Link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
3		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali.
4		*Final State*	Bagaimana objek dibentuk dan dihancurkan.
5		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

B. Penelitian Terdahulu

Studi terdahulu mengarah pada hasil studi yang telah dilakukan oleh pihak lain sebelumnya dan memiliki relevansi atau hubungan dengan tema maupun tantangan yang sedang diungkit pada penelitian saat ini. Studi ini digunakan sebagai referensi untuk memahami landasan teori, metode, serta temuan yang relevan, sekaligus untuk mengidentifikasi celah penelitian yang dapat diisi melalui penelitian yang baru. Penelitian terdahulu membantu memberikan gambaran tentang apa yang telah diketahui di bidang tersebut dan bagaimana penelitian baru dapat memberikan kontribusi yang signifikan. Beberapa penelitian terdahulu yang dilakukan:

1. Mohamad Anas Sobarnas, Suherwin, Ashari Imamuddin (2021) Sekolah Tinggi Teknologi Muhammadiyah Cileungsi dalam Jurnal INFOTECH yang berjudul

“Aplikasi *Software* untuk Monitoring Bahan Baku pada Perusahaan Pengolah Rempah-Rempah”. Penelitian ini bertujuan merancang aplikasi monitoring bahan baku untuk perusahaan pengolahan rempah-rempah guna mengatasi masalah penumpukan dan kerusakan bahan akibat pencatatan manual. Perancangan aplikasi dilakukan dengan pendekatan *Unified Modeling Language* (UML) dan mencakup fitur-fitur inti seperti pemeriksaan ketersediaan bahan baku, pengajuan permintaan barang, penyusunan purchase order (PO), proses penerimaan dan distribusi barang, serta pelaporan hasil stok opname. Berdasarkan hasil evaluasi, desain sistem ini dinilai mampu menjawab kebutuhan operasional perusahaan dan layak untuk diimplementasikan guna meningkatkan efisiensi dalam proses bisnis..

2. Migel Simanjorang (2024) Universitas Atma Jaya Yogyakarta dalam Jurnal yang berjudul **“Pengembangan Aplikasi Web Monitoring Rantai Pasokan Jeruk untuk Meningkatkan Kualitas Produk dan Kepuasan Pelanggan”.** Penelitian ini bertujuan mengembangkan aplikasi web monitoring rantai pasokan jeruk untuk meningkatkan kualitas produk dan kepuasan pelanggan. Aplikasi menggunakan Laravel sebagai backend, Vue.js untuk frontend, dan blockchain untuk transparansi serta keamanan data. Hasilnya, aplikasi ini meningkatkan efisiensi pemantauan pengiriman, menjaga kualitas jeruk hingga konsumen, dan meningkatkan kepuasan pelanggan melalui fitur interaktif, estimasi waktu pengiriman, serta umpan balik responsif.
3. Ninda Lutfiani, Fitra Putri Oganda, Farida Agustin, Qurotul AIni, Untung Rahardja (2020) Universitas Raharja dalam Jurnal InfoTekJar yang berjudul

“Desain dan Metodologi Teknologi Blockchain untuk Monitoring Manajemen Rantai Pasokan Makanan yang Terdesentralisasi”. Penelitian ini merancang sistem monitoring rantai pasokan makanan terdesentralisasi berbasis blockchain untuk meningkatkan keterlacakan, transparansi, dan kepercayaan, khususnya selama pandemi COVID-19. Hasilnya menunjukkan *blockchain* mampu menyediakan pemantauan *real-time* kondisi dan lokasi produk, menjaga keamanan data, meminimalkan penipuan, meningkatkan efisiensi manajemen, serta mendukung stabilitas pasokan makanan selama krisis.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Studi ini mengadopsi pendekatan strategi penelitian pengembangan (*Research and Development*), dengan pendekatan peningkatan *prototipe* untuk membangun dan menyempurnakan aplikasi monitoring pasokan bawang merah berbasis Android. Pendekatan kuantitatif diterapkan melalui evaluasi kinerja sistem menggunakan metrik seperti keakuratan data, waktu pemrosesan informasi, dan kepuasan pengguna, sehingga memastikan aplikasi dapat memenuhi kebutuhan pemantauan pasokan secara efektif. Data dalam penelitian ini diperoleh melalui dua pendekatan utama, yaitu teknik wawancara dengan pihak terkait dan penelusuran referensi dari berbagai sumber pustaka yang relevan.

B. Lokasi dan Waktu

Riset ini dilaksanakan di wilayah Kabupaten Enrekang, sebuah daerah yang dikenal sebagai sentra utama budidaya bawang merah di Indonesia. Penelitian difokuskan pada wilayah penghasil bawang merah utama di Enrekang, dengan melibatkan petani, pedagang, dan dinas terkait untuk mengembangkan aplikasi yang dapat memantau pasokan bawang merah secara lebih efektif. Waktu pelaksanaan penelitian akan berlangsung selama ± 3 bulan.

C. Alat dan Bahan

Pengembangan aplikasi pada studi ini membutuhkan dorongan oleh perangkat lunak serta perangkat keras yang memadai. Oleh karena itu, berikut disajikan rincian mengenai tools dan perangkat yang digunakan dalam proses perancangan dan implementasi:

1. *Software*/Perangkat Lunak

Daftar perangkat lunak yang dimanfaatkan dalam proses perancangan aplikasi ini disajikan secara rinci pada Tabel 3.1.

Tabel 3. 1 Spesifikasi Perangkat Lunak

Spesifikasi	
Sistem Operasi	Windows 11
Tool Pemrograman	*Android Studio*
Bahasa Pemrograman	*Kotlin*

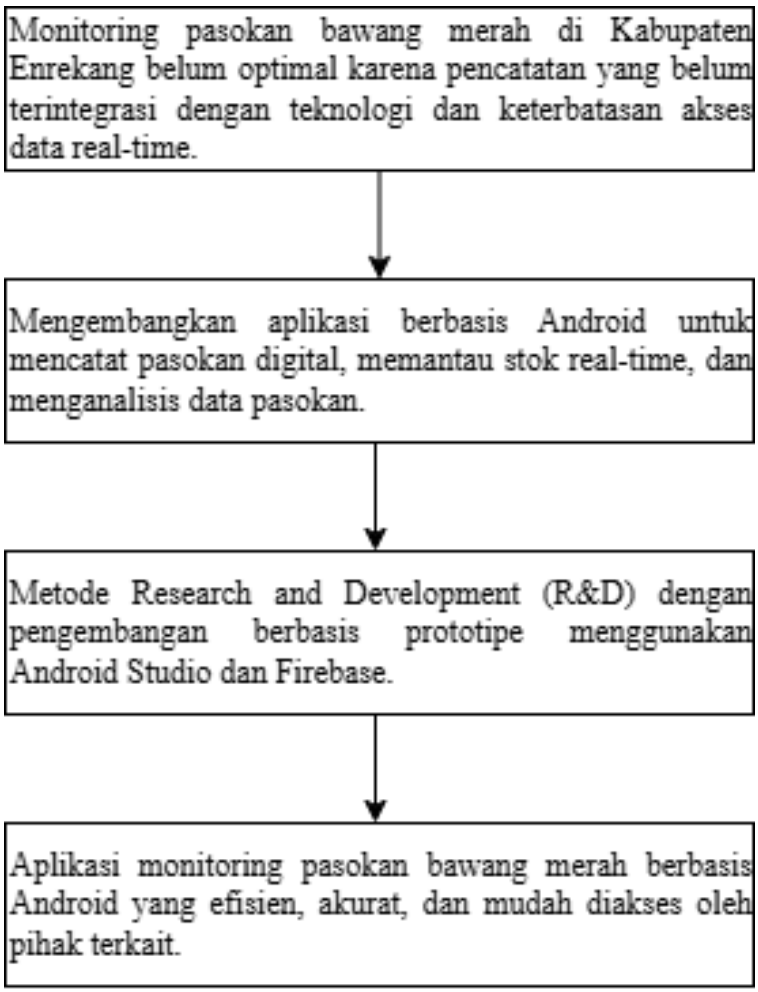
2. Hardware/Perangkat Keras

Daftar perangkat keras yang dimanfaatkan dalam proses perancangan aplikasi ini disajikan secara rinci pada Tabel 3.1.

Tabel 3. 2 Spesifikasi Perangkat Keras

Spesifikasi	
Merk Laptop	MSI GF63 Thin 10SC
Processor Laptop	Intel Core i5
RAM	8GB

D. Kerangka Berfikir



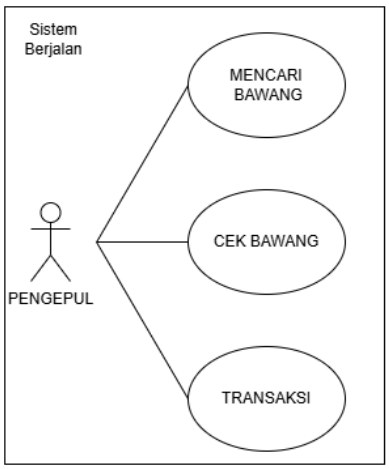
Gambar 3. 1 Kerangka Berfikir

E. Perancangan Sistem

Tahap perancangan sistem sebagai proses dalam perumusan desain dari sistem yang ingin dikembangkan. Pada studi ini, rancangan sistem disajikan melalui Use Case Diagram serta diagram alur kerja sistem yang menggambarkan bagaimana sistem tersebut akan beroperasi. Penjelasan selengkapnya dapat dilihat pada bagian berikut

1) Sistem yang Berjalan

Pengepul bawang merah memulai proses dengan mencari bawang merah yang tersedia di pasaran. Setelah menemukan bawang merah yang diinginkan, pengepul akan melakukan pengecekan untuk memastikan kualitas, kuantitas, dan kesesuaian dengan kebutuhan, seperti ukuran, tingkat kesegaran, dan harga. Proses pengecekan ini sangat penting untuk menghindari kerugian akibat barang yang tidak sesuai standar. Jika semua sudah sesuai, pengepul akan melanjutkan ke tahap transaksi, yaitu pembayaran kepada penjual atau pemasok. Transaksi ini bisa dilakukan secara tunai, melalui transfer bank, atau sistem pembayaran lainnya, tergantung kesepakatan.

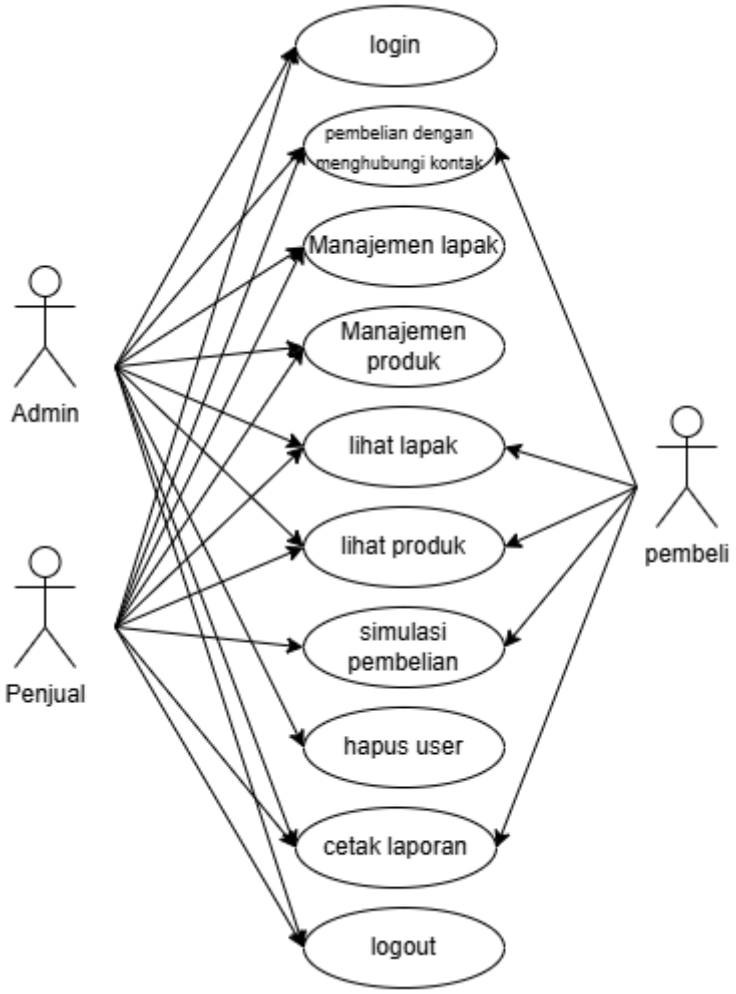


Gambar 3. 2 *Use Case* Diagram

2) Sistem Yang Di Usulkan

Desain sistem yang diusulkan berupa aplikasi Android yang dirancang untuk memfasilitasi interaksi antara petani sebagai admin dan pedagang sebagai pengguna dalam transaksi produk pertanian seperti bawang merah. Petani, sebagai admin,

memiliki kontrol penuh untuk mengelola produk dengan mengunggah informasi seperti nama produk, harga, foto, deskripsi, dan stok yang tersedia.



Gambar 3. 3 *Use Case Diagram* yang diusulkan

F. Metode Pengumpulan Data

Dalam studi ini, data dihimpun dengan dua metode utama: wawancara dan studi kepustakaan. Wawancara dijalankan dengan sejumlah pihak yang memiliki peran langsung dalam rantai pasok, termasuk petani, distributor, serta instansi

pemerintah setempat di Kabupaten Enrekang. Melalui wawancara ini, diperoleh gambaran mengenai mekanisme pencatatan pasokan yang sedang berjalan, hambatan yang sering ditemui di lapangan, serta kebutuhan fungsional yang diharapkan dapat dipenuhi oleh aplikasi yang dikembangkan. Studi literatur digunakan untuk mengkaji penelitian terdahulu, teknologi, serta metode yang relevan untuk mendukung pengembangan aplikasi. Data yang dikumpulkan melalui kedua teknik ini diolah dan dianalisis sebagai dasar untuk merancang sistem yang sesuai dengan kebutuhan *user*.

G. Metode Pengujian

Studi ini menerapkan dua jenis metode pengujian, yaitu pengujian *Blackbox* dan pengujian *Whitebox*, guna mengevaluasi fungsi dan kestabilan aplikasi. Penggunaan kedua pendekatan ini berguna untuk memverifikasi bahwa aplikasi beroperasi sesuai dengan keperluan fungsional serta mengikuti alur logika kode yang telah disusun dengan benar.

1. Black Box Testing

Pengujian *blackbox* adalah tahap uji aplikasi yang mengevaluasi cara kerja suatu sistem dari sudut pandang pengguna, terlepas dari struktur internal atau penulisan kodenya. Fokusnya adalah memvalidasi hasil terhadap saran yang dihadirkan untuk memvalidasi jika sistem berfungsi sesuai ketentuan spesifikasi (Shadiq et al., 2021).

2. White Box Testing

Pengujian *whitebox* adalah metode pengujian yang memeriksa logika internal, alur program, dan struktur kode suatu aplikasi. Tujuannya adalah untuk memastikan

bahwa semua jalur logis dalam kode berjalan sesuai dengan desain, termasuk kondisi if-else, loop, dan alur data (Teknologi & 2022, n.d.).

BAB IV

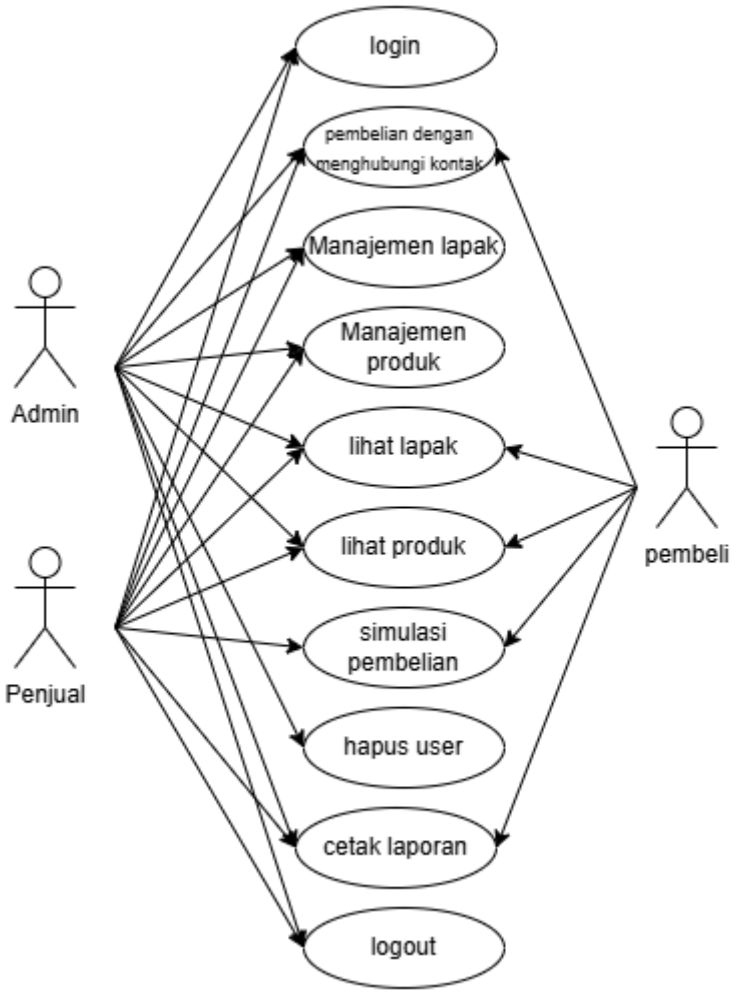
HASIL DAN PEMBAHASAN

A. Analisis Aliran Data UML

Pemanfaatan UML (*Unified Modeling Language*) dalam menganalisis aliran data memiliki peran krusial dalam membantu perancangan sistem informasi yang tepat guna dan efisien. UML sendiri adalah bahasa pemodelan standar yang memungkinkan *visualisasi* elemen-elemen struktur dan dinamika perilaku dalam sebuah sistem perangkat lunak. Dalam konteks analisis aliran data, UML membantu mengemukakan data yang bergerak dari berbagai proses, mulai dari masukan (input), pemrosesan, hingga keluaran (output). Dalam proses analisis aliran data, terdapat sejumlah jenis diagram UML yang sering dimanfaatkan, seperti *Use Case Diagram* yang merepresentasikan relasi pada aktor dan sistem, *Activity Diagram* yang mengimplementasikan rangkaian aktivitas serta alur proses kerja dalam sistem, serta *Sequence Diagram* yang memperlihatkan urutan komunikasi atau pertukaran pesan antar objek dalam sebuah skenario tertentu. Dengan memanfaatkan diagram-diagram ini, pengembang atau peneliti dapat mem*visualisasi*kan proses bisnis dan aliran informasi secara sistematis, mengidentifikasi potensi masalah, serta mendokumentasikan sistem secara terstruktur. Analisis ini menjadi bagian penting dalam tahapan perancangan sistem karena dapat memperjelas kebutuhan fungsional dan teknis, serta menjadi dasar dalam implementasi sistem yang optimal.

1. Use Case Diagram

Use case diagram merupakan bentuk *visualisasi* UML yang berguna untuk memetakan hubungan antara sistem yang dirancang dengan para aktor, baik itu pengguna maupun entitas eksternal lainnya. Diagram ini menampilkan fungsi-fungsi utama yang tersedia dalam sistem, dilihat dari perspektif pengguna, serta menunjukkan bagaimana aktor berperan dalam menjalankan masing-masing fungsi tersebut.



Gambar 4. 1 *Use Case Diagram*

Tabel 4. 1 *Use Case* Skenario

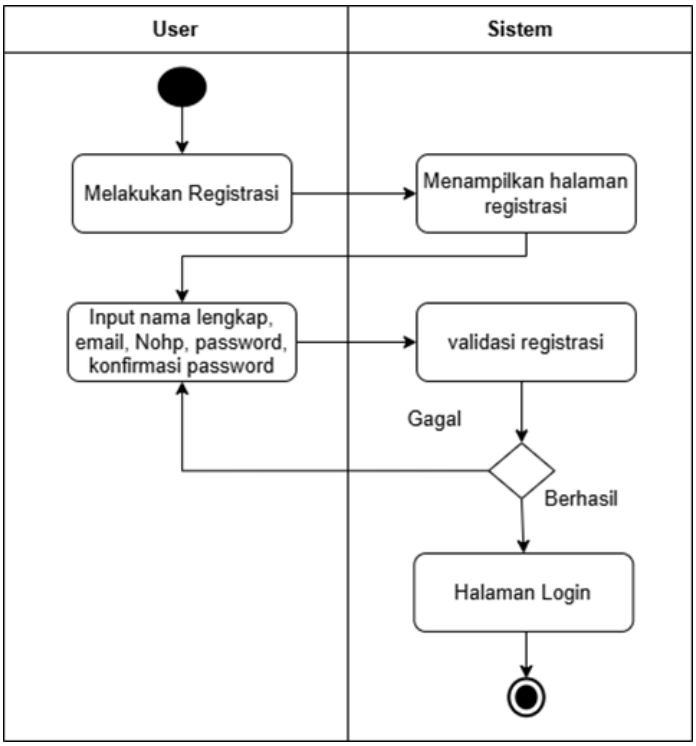
Nama *Use Case*	Deskripsi	Aktor	Alur Kerja
Login	Petani masuk ke system menggunakan akun yang telah dibuat.	Petani	- Petani mengakses aplikasi android yang telah di download. - Sistem menampilkan halaman *login*
Manajemen Lapak	Setelah melakukan *login*, petani dapat menambahkan atau menghapus lapak yang telah dibuat.	Petani	- Petani dapat melakukan pembuatan lapak, mengedit atau menghapus lapak. - Sistem menyimpan database lapak.
Manajemen Produk	Petani dapat menambahkan beberapa produk setelah membuat lapak. Petani juga dapat mengedit, menghapus atau menambahkan produk.	Petani	- Petani menambahkan produk, mengedit atau menghapus produk yang telah habis. - Sistem melakukan penyimpanan database.
Lihat Lapak	Petani dan pembeli dapat melihat lapak yang tersedia di sistem.	Petani dan Pembeli	- Petani dan pembeli dapat melihat lapak yang tersedia. - Sistem menapilkan halaman lapak.
Lihat Produk	Petani dan pembeli dapat melihat produk dan beberapa lapak yang telah dibuat.	Petani dan Pembeli	- Petani dan pembeli dapat melihat produk yang tersedia. - Sistem menampilkan halaman produk.
Lihat List Produk	Petani dan pembeli dapat melihat list dari beberapa produk yang telah dibuat oleh petani lain.	Petani dan Pembeli	- Petani dan pembeli dapat melihat list produk dari lapak lain. - Sistem menampilkan halaman produk.
Simulasi Pembelian Bawang	Petani dan pembeli dapat melakukan simulasi pembelian dalam aplikasi. Petani dan pembeli juga dapat melihat total harga dari pesanan tertentu.	Petani dan Pembeli	- Petani dan pembeli melakukan simulasi pembelian. - Sistem menampilkan halaman simulasi pembelian.

Nama *Use Case*	Deskripsi	Aktor	Alur Kerja
Pembelian dengan menghubingi kontak yang tercantum	Pembeli dapat langsung menghubungi penjual dari nomor kontak yang telah didaftarkan sebelumnya.	Pembelian	- Pembeli menghubungi nomor kontak yang tersedia. - Sistem menyediakan nomor kontak dari pemilik produk.

2. *Activity Diagram*

Activity Diagram berfungi untuk menggambarkan secara visual alur kerja penggunaan sebuah sistem. Diagram ini memberi *visualisasi* aktivitas yang berlangsung dalam sistem dan tahapan pelaksanaannya.

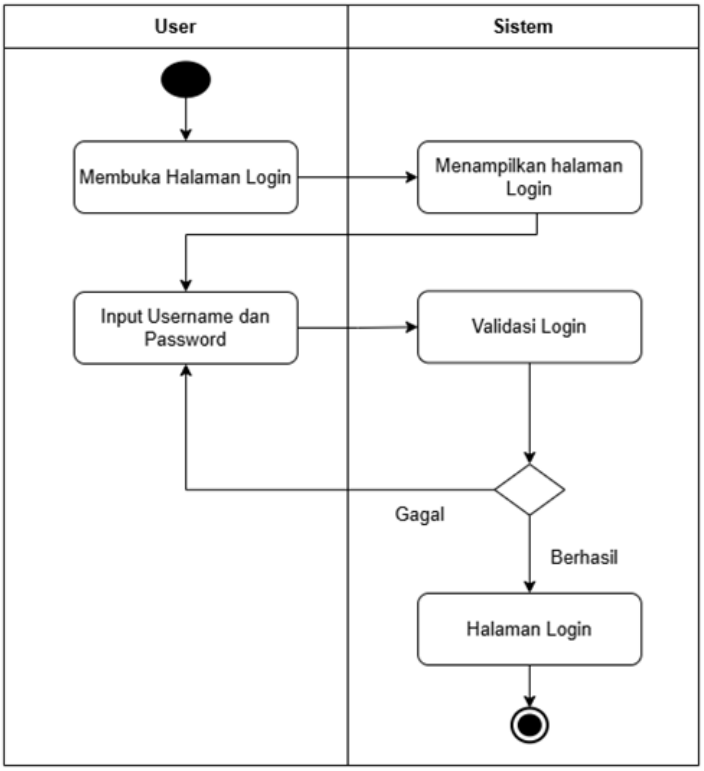
a. *Activity Diagram Registrasi*



Gambar 4. 2 *Activity Diagram Registrasi*

Pada Gambar 4.2 menyajikan *activity diagram* untuk proses pendaftaran akun. Prosedur diawali oleh pengguna yang mengakses tampilan awal aplikasi, yang kemudian direspons oleh sistem dengan menampilkan halaman *login*. Pada fase ini, pengguna disajikan formulir yang berisi kolom untuk mengisi username dan password sebagai langkah awal proses autentikasi. Setelah halaman tampil, pengguna mengisi form tersebut dengan data yang dimiliki, kemudian mengirimkan data tersebut ke sistem. Sistem kemudian memproses dan memvalidasi informasi *login* yang diberikan.

b. *Activity Diagram Login*

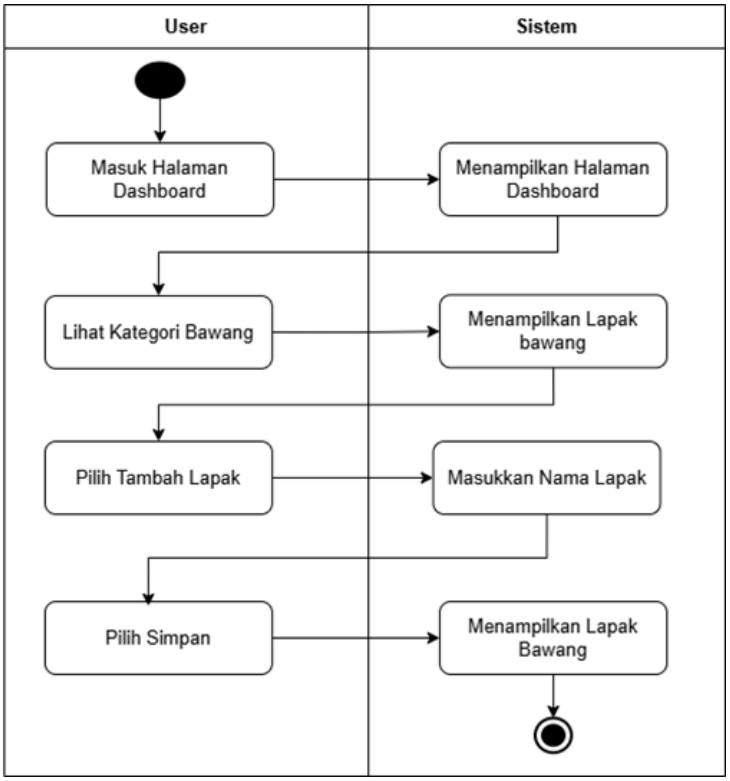


Gambar 4. 3 *Activity Diagram Login*

Gambar 4.3 menyajikan diagram aktivitas proses *login*. Proses awalnya, pengguna yang masuk ke halaman *login*. Sistem menampilkan halaman *login*

dengan formulir untuk memasukkan *username* dan *password*. Setelah halaman *login* muncul, pengguna memasukkan *username* dan *password* lalu mengirimkannya ke sistem. Sistem memproses data yang dikirimkan dan memvalidasi *login*. Data yang dimasukkan pengguna dibandingkan dengan informasi yang tersimpan pada sistem, misalnya, *database*.

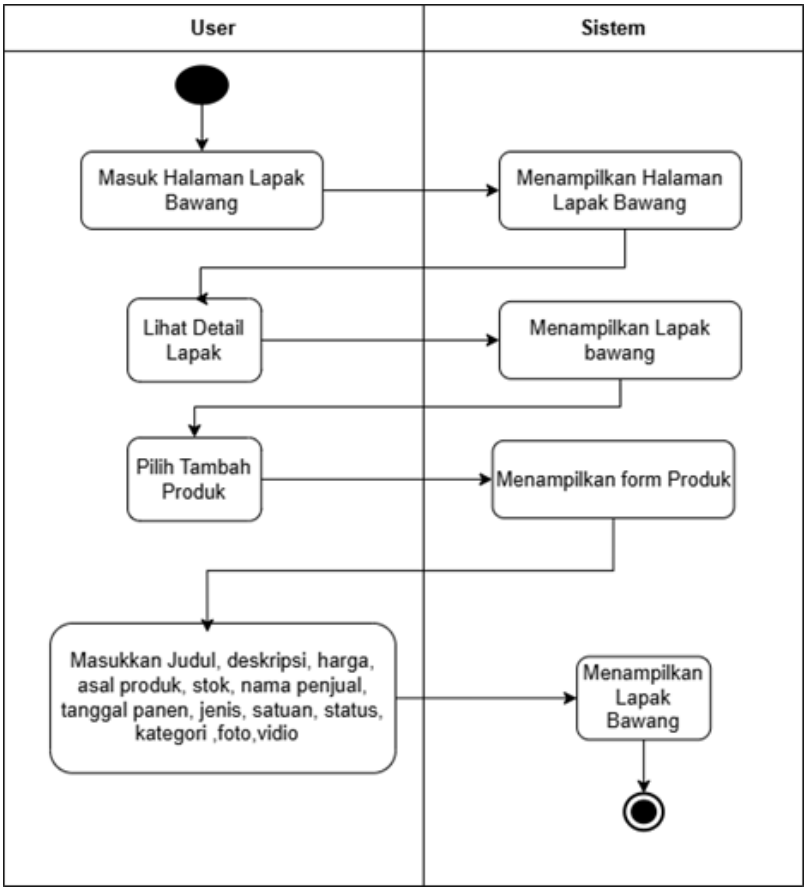
c. *Activity Diagram* Tambah Lapak



Gambar 4. 4 *Activity Diagram* Tambah Lapak

Gambar 4.4 menampilkan *Activity Diagram* proses Tambah Lapak. Proses dimulai saat *user* masuk ke halaman dashboard aplikasi. Sistem kemudian akan menampilkan tampilan dashboard yang berisi berbagai menu atau fitur yang tersedia. Dari sana, pengguna memilih untuk melihat kategori bawang, yang kemudian akan direspons oleh sistem dengan menampilkan daftar lapak bawang yang sudah ada. Setelah itu, pengguna memilih opsi "Tambah Lapak", yang berarti ingin menambahkan entri baru terkait lapak bawang.

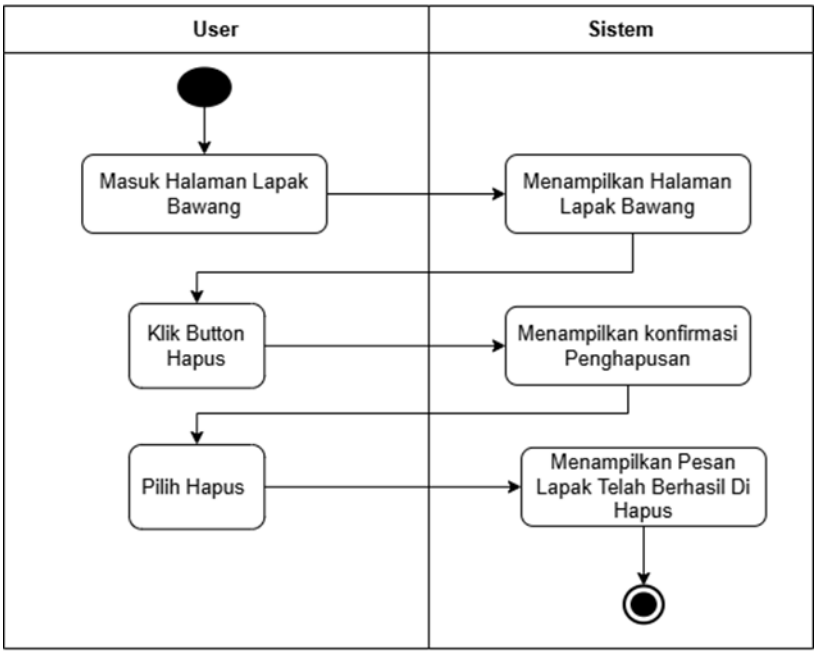
d. *Activity Diagram* Tambah Produk



Gambar 4. 5 *Activity Diagram* Tambah Produk

Proses dimulai saat pengguna masuk ke halaman Lapak Bawang. Sistem akan menampilkan halaman tersebut yang berisi berbagai lapak yang sudah ada. Pengguna kemudian memilih untuk melihat detail dari salah satu lapak yang tersedia. Sistem merespons dengan menampilkan informasi lengkap mengenai lapak tersebut. Setelah melihat detail lapak, pengguna memilih opsi Tambah Produk, yang menandakan bahwa pengguna ingin menambahkan informasi produk baru terkait lapak tersebut. Sistem kemudian menampilkan form input produk yang berisi kolom-kolom isian yang harus dilengkapi

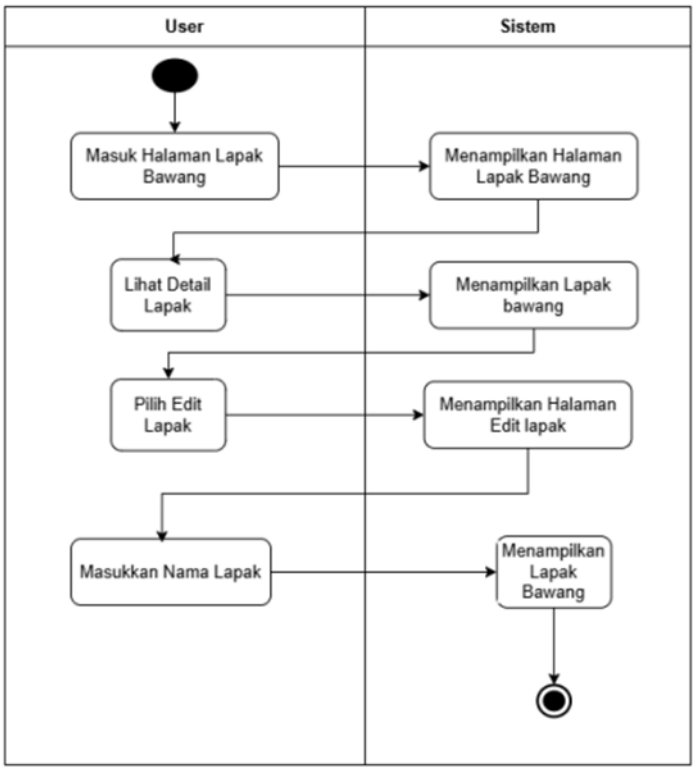
e. *Activity Diagram* Hapus Lapak



Gambar 4. 6 *Activity Diagram* Hapus Lapak

Gambar 4.6 Menampilkan *Activity Diagram* Hapus Lapak. Proses dimulai ketika pengguna masuk ke halaman Lapak Bawang. Sistem akan menampilkan halaman tersebut yang berisi daftar lapak yang telah tersedia. Dari sana, pengguna memilih untuk menghapus salah satu lapak dengan mengklik tombol Hapus.Setelah tombol hapus diklik, sistem tidak langsung menghapus lapak tersebut. Sebagai langkah pengamanan, sistem menampilkan konfirmasi penghapusan, yaitu sebuah peringatan atau pertanyaan kepada pengguna untuk memastikan bahwa tindakan penghapusan memang diinginkan.

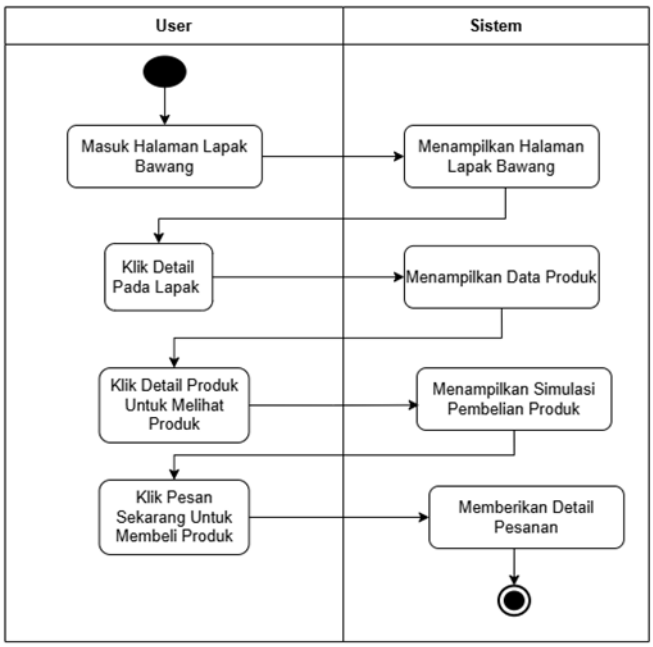
f. *Activity Diagram* Edit Produk



Gambar 4. 7 *Activity Diagram* Edit Produk

Gambar 4.7 Menampilkan *Activity Diagram* Edit Lapak. Proses dimulai saat pengguna masuk ke halaman Lapak Bawang. Sistem kemudian menampilkan halaman tersebut yang berisi daftar lapak yang tersedia. Pengguna kemudian memilih untuk melihat detail dari salah satu lapak. Sistem merespons dengan menampilkan informasi detail lapak bawang yang dipilih.Setelah melihat detail lapak, pengguna memilih opsi Edit Lapak untuk memperbarui informasi yang ada

g. *Activity Diagram* Pembelian



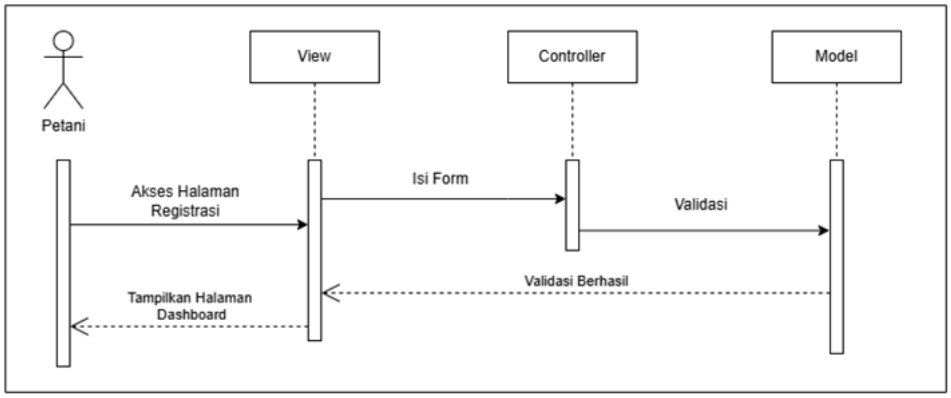
Gambar 4. 8 *Activity Diagram* Pembelian

Gambar 4.8 *Activity Diagram* Pembelian menunjukkan alur pembelian produk bawang oleh pengguna melalui aplikasi monitoring pasokan bawang merah di Kabupaten Enrekang. Proses ini memperlihatkan interaksi antara pengguna dan sistem mulai dari mengakses lapak hingga menerima detail pesanan. Pertama, pengguna masuk ke halaman Lapak Bawang, di mana sistem akan menggambarkan halaman yang memuat daftar lapak yang ada. Setelah itu, pengguna mengklik detail pada salah satu lapak untuk melihat produk-produk yang ditawarkan. Sistem akan merespons dengan menampilkan data produk dari lapak tersebut. Selanjutnya, pengguna memilih salah satu produk dengan mengklik detail produk untuk melihat informasi lebih lanjut mengenai produk tersebut.

3. *Sequence Diagram*

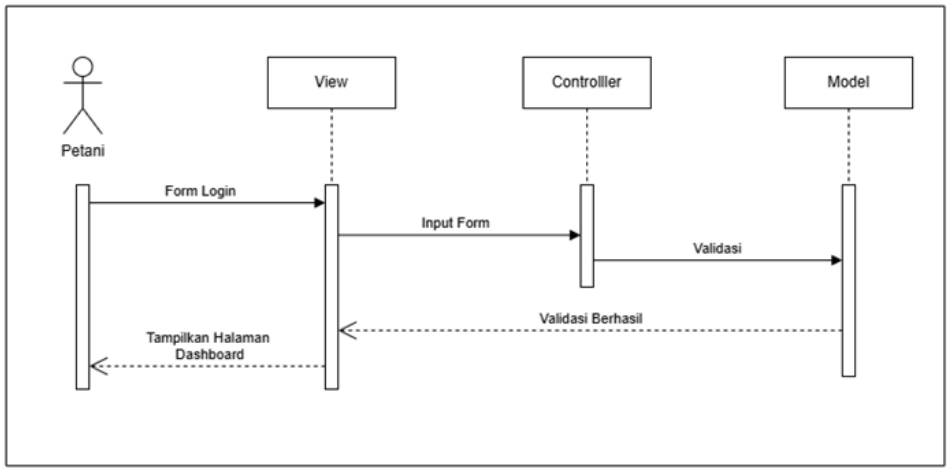
Sequence Diagram mem*visualisasi*kan interaksi pada objek dalam sistem secara berurutan dan temporal. Diagram ini menggambarkan bagaimana objek bertukar informasi atau data untuk menyelesaikan suatu tugas atau kasus penggunaan.

a. *Sequence Diagram* Register (Petani)



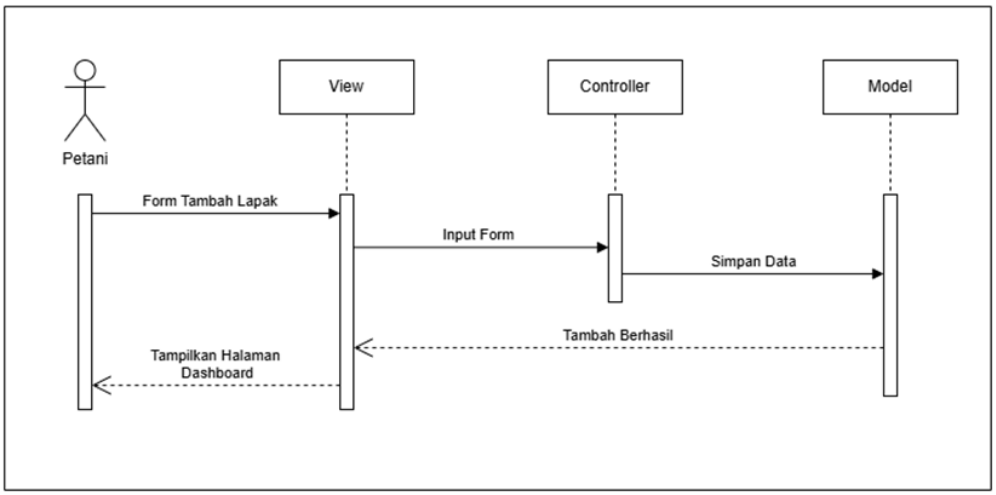
Gambar 4. 9 *Sequence Diagram* Register (Petani)

b. *Sequence Diagram Login* (Petani)



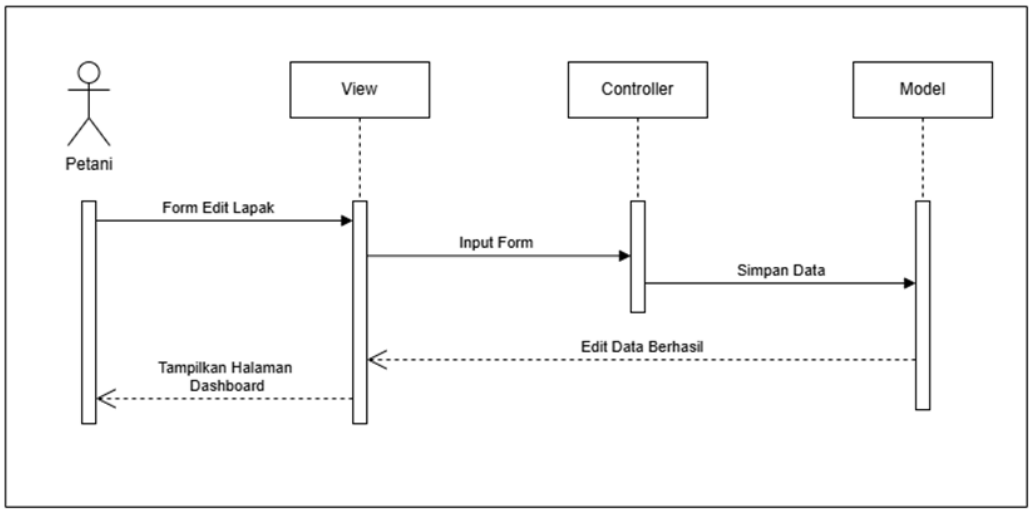
Gambar 4. 10 *Sequence Diagram Login* (Petani)

c. *Sequence Diagram* Tambah Produk (Petani)



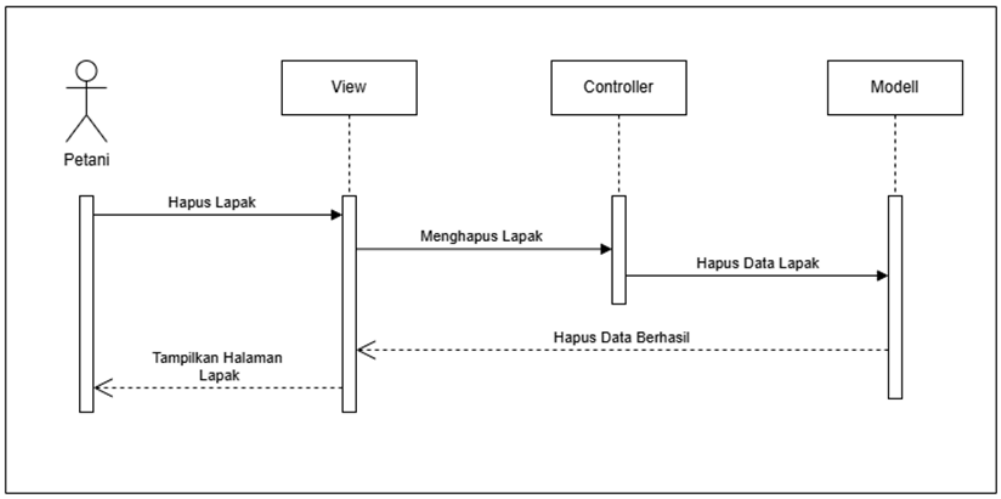
Gambar 4. 11 *Sequence Diagram* Tambah Produk (Petani)

d. *Sequence Diagram* Edit Lapak (Petani)



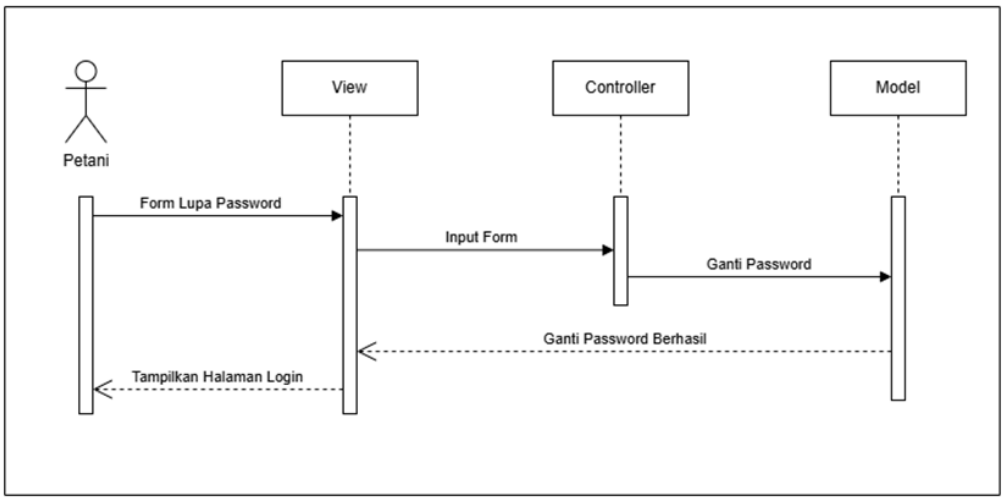
Gambar 4. 12 *Sequence Diagram* Edit Lapak (Petani)

e. *Sequence Diagram* Hapus Lapak (Petani)



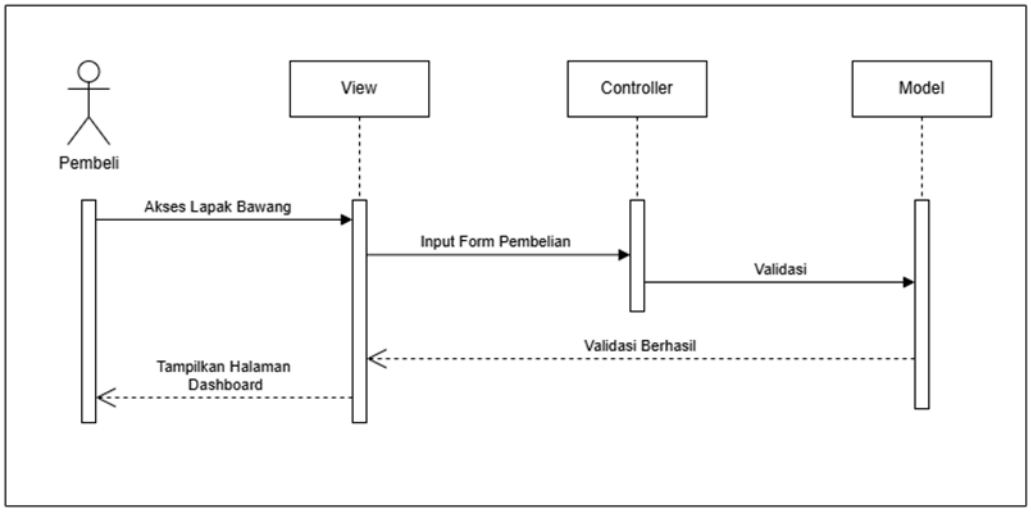
Gambar 4. 13 *Sequence Diagram* Hapus Lapak (Petani)

f. *Sequence Diagram* Lupa *Password* (Petani)



Gambar 4. 14 *Sequence Diagram* Lupa *Password* (Petani)

g. *Sequence Diagram* Pembeli



Gambar 4. 15 Sequence Diagram Pembeli

B. Desain Sistem

1. Halaman Awal

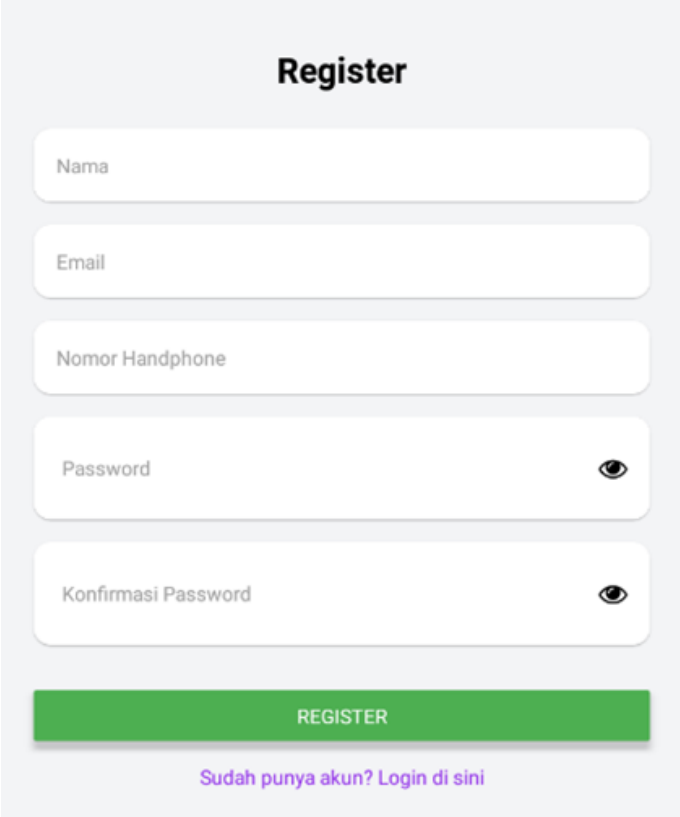
Halaman awal aplikasi pantau bawang merupakan tampilan utama yang menampilkan beberapa menu utama yang memudahkan pengguna mengakses aplikasi.



Gambar 4. 16 Tampilan Awal Aplikasi

2. Halaman *Registrasi* Petani

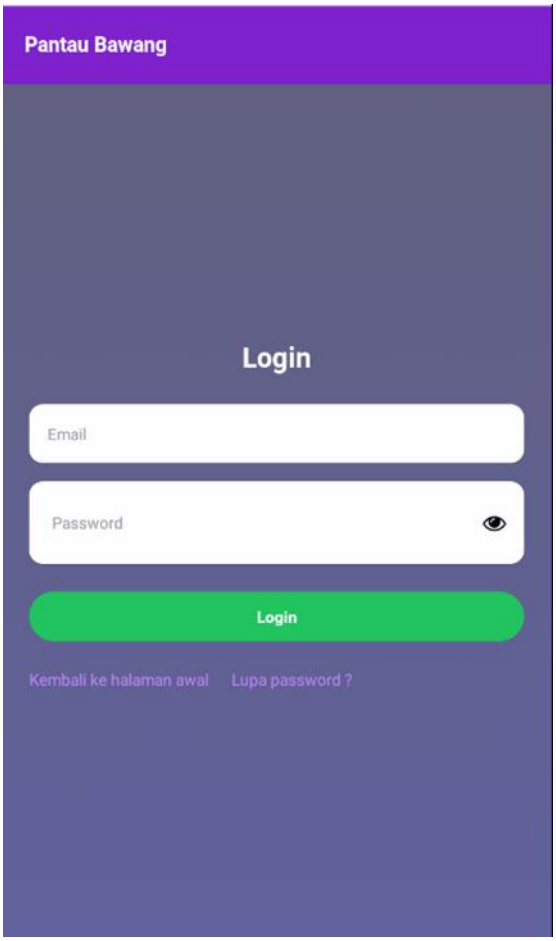
Halaman *Registrasi* Aplikasi Pantau bawang adalah antarmuka untuk pengguna baru mendaftarkan akun. Pengguna diminta mengisi form data yang telah disediakan.



Gambar 4. 17 Halaman *Registrasi* Petani

3. Halaman *Login* Petani

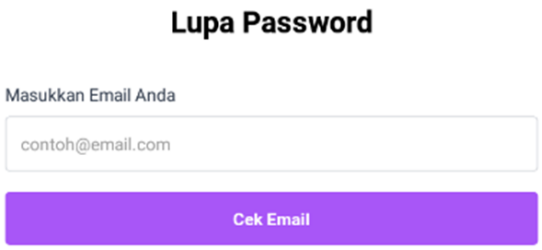
Halaman *login* aplikasi Pantau Bawang yang merupakan antarmuka pengguna untuk melakukan autentikasi sebelum mengakses sistem. Pada halaman ini, *user* memasukkan alamat *email* serta kata sandi sebagai kredensial *login*.



Gambar 4. 18 Halaman *Login* Petani

4. Halaman Lupa *Password*

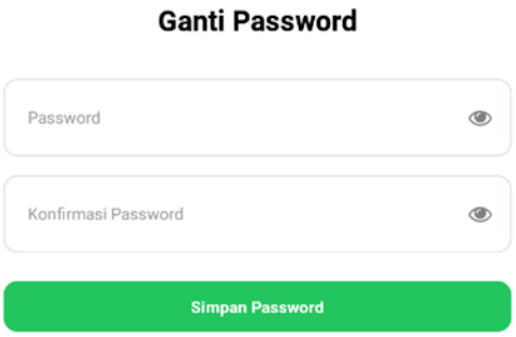
Halaman ini dirancang untuk membantu pengguna yang tidak dapat mengakses akun mereka karena lupa kata sandi. Pengguna diminta untuk memasukkan alamat email yang terdaftar pada sistem ke dalam kolom yang tersedia



Gambar 4. 19 Halaman Lupa *Password*

5. Halaman Ganti *Password*

Halaman "Ganti *Password*" pada aplikasi Pantau Bawang memungkinkan pengguna untuk menetapkan kata sandi baru setelah melalui proses verifikasi akun. Pada halaman ini, *user* memasukkan kata sandi baru dan mengonfirmasinya dengan cara mengetik ulang pada kolom yang tersedia.



Gambar 4. 20 Halaman Ganti *Password*

6. Halaman Lihat Lapak Pembeli

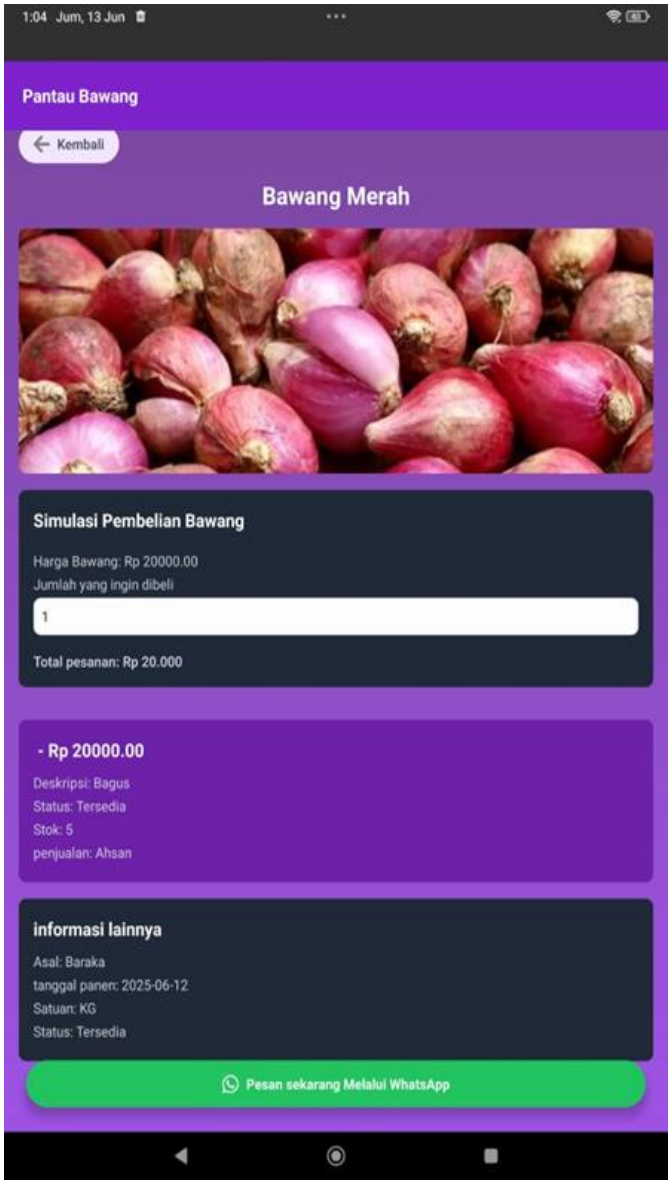
Halaman Lihat Lapak Pembeli Aplikasi Pantau Bawang memungkinkan para pembeli dapat melihat lapak para penjual dan juga dapat melihat stok yang ada pada lapak yang tersedia.



Gambar 4. 21 Halaman Lihat Lapak Pembeli

7. Halaman Simulasi Pembelian

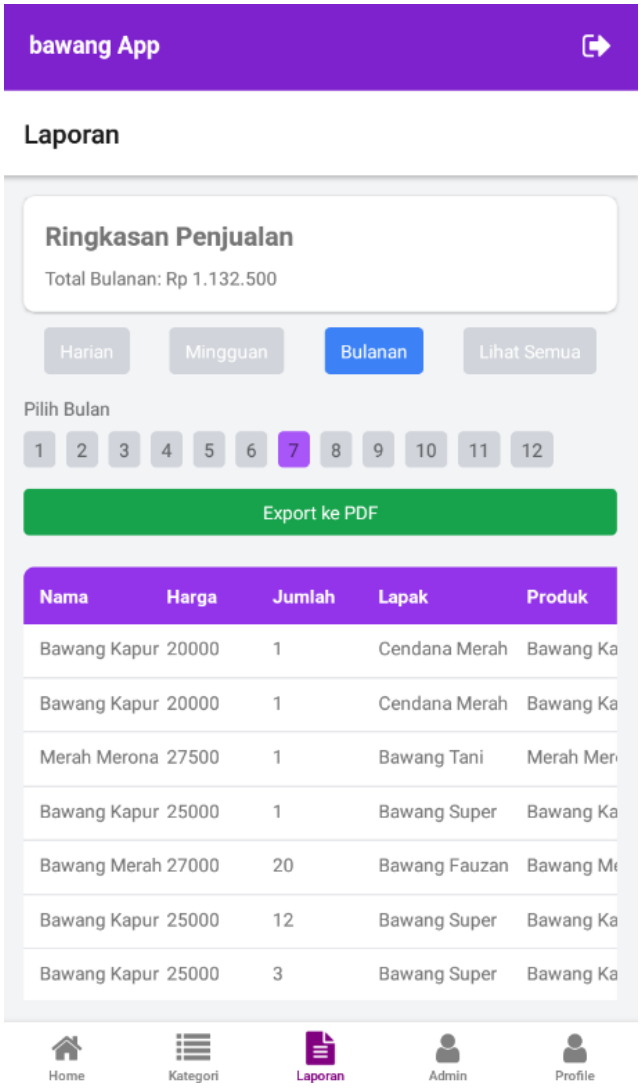
Halaman Simulasi Pembalian Aplikasi Pantau Bawang Menampilkan antarmuka simulasi pembelian bawang merah dan pengguna dan pembeli dapat melakukan simulasi pengecekan harga.



Gambar 4. 22 Halaman Simulasi Pembelian

8. Halaman Cetak Laporan Pembelian

Halaman Cetak Laporan Pada Aplikasi Pantau bawang, Menampilkan Halaman dimana struk atau laporan pembelian dapat dicetak kemudian dapat menjadi bukti pembelian.



Gambar 4. 23 Halaman Laporan

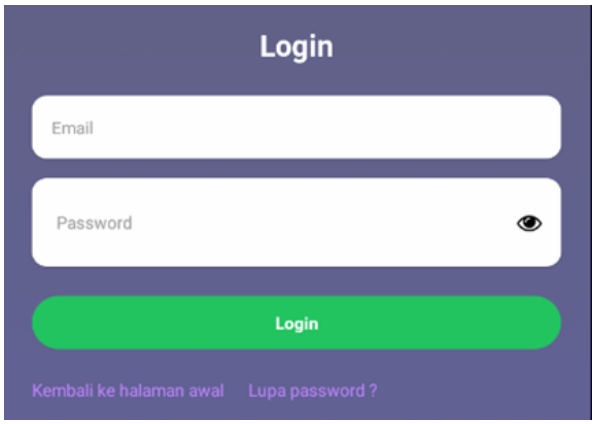
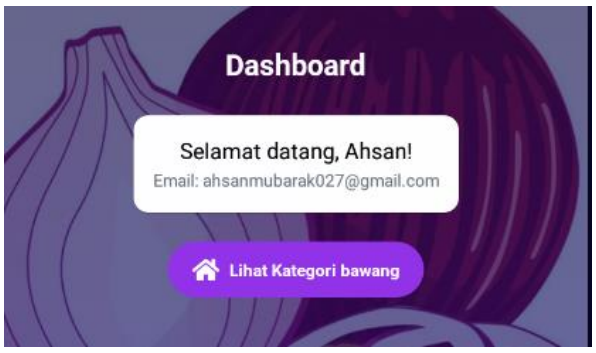
C. Pengujian Sistem

1. *Blackbox Testing*

Blackbox Testing berisikan pendekatan pengujian aplikasi yang mengarah pada kapabilitas aplikasi tanpa mengetahui konfigurasi internal atau kode programnya.

a. Pengujian *Blackbox* pada halaman *login*

Tabel 4. 2 *Blackbox Testing* halaman *login*

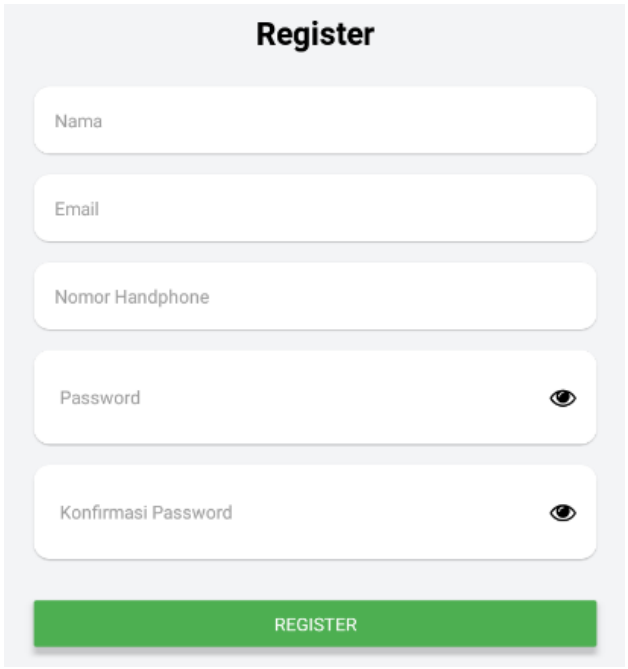
Test Faktor	Hasil	Keterangan
Memasukkan *Email dan Password y*ang sesuai	✔	Sistem berhasil menampilkan halaman Dashboard
Screenshot		
 		

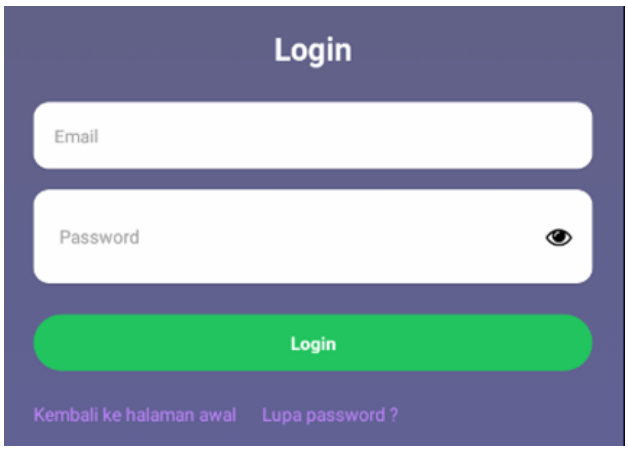
b. *Blackbox Testing* halaman *Registrasi*

Tabel 4. 3 *Blackbox Testing* halaman *Registrasi*

Test Faktor	Hasil	Keterangan

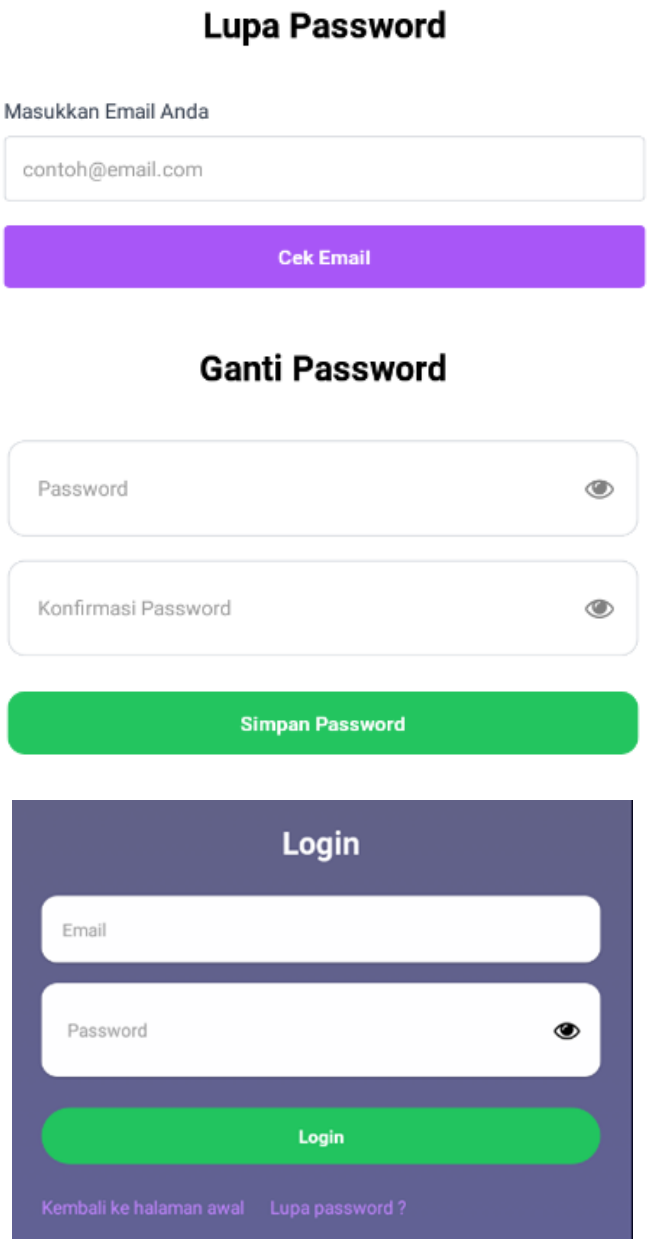
Mengisi *Form input* yang sesuai	✔	Sistem berhasil menampilkan halaman *Login*
Screenshot		





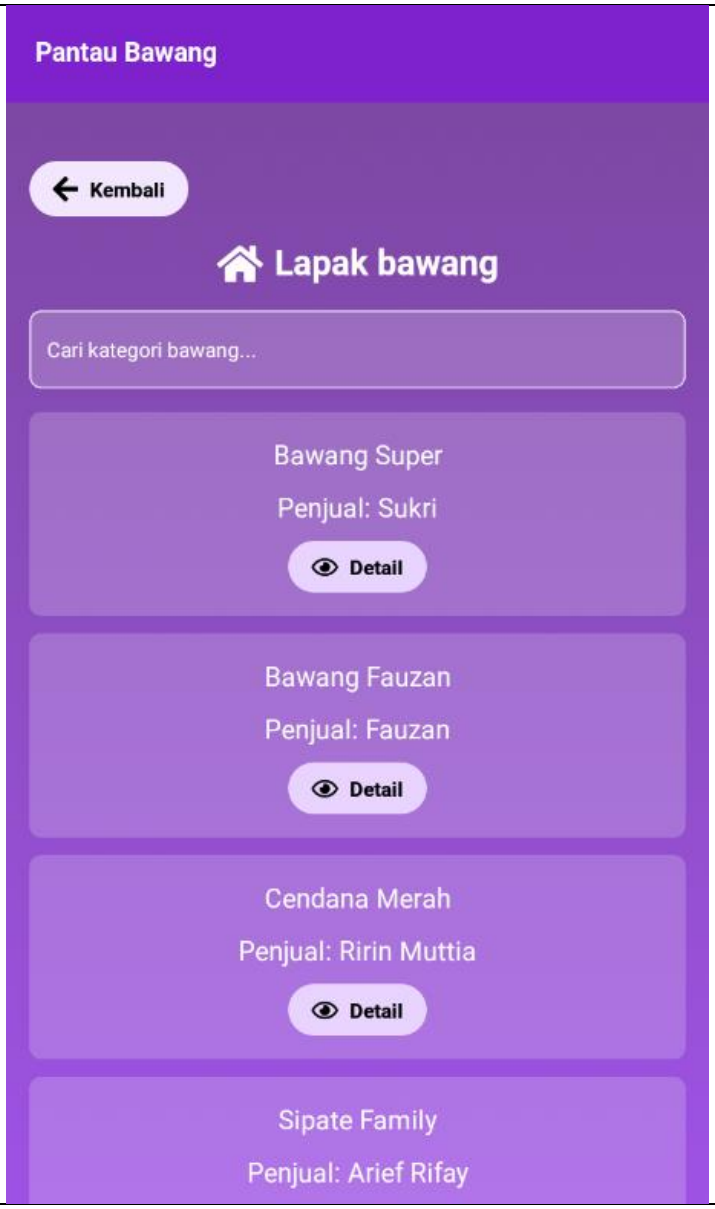
c. *Blackbox Testing* halaman lupa *Password*

Tabel 4. 4 Halaman lupa *Password*

Test Faktor	Hasil	Keterangan
Mengisi *Form input Email dan Password* baru	✔	Sistem berhasil menampilkan halaman *Login*
Screenshot		
		

d. *Blackbox Testing* tampilan halaman lapak bawang

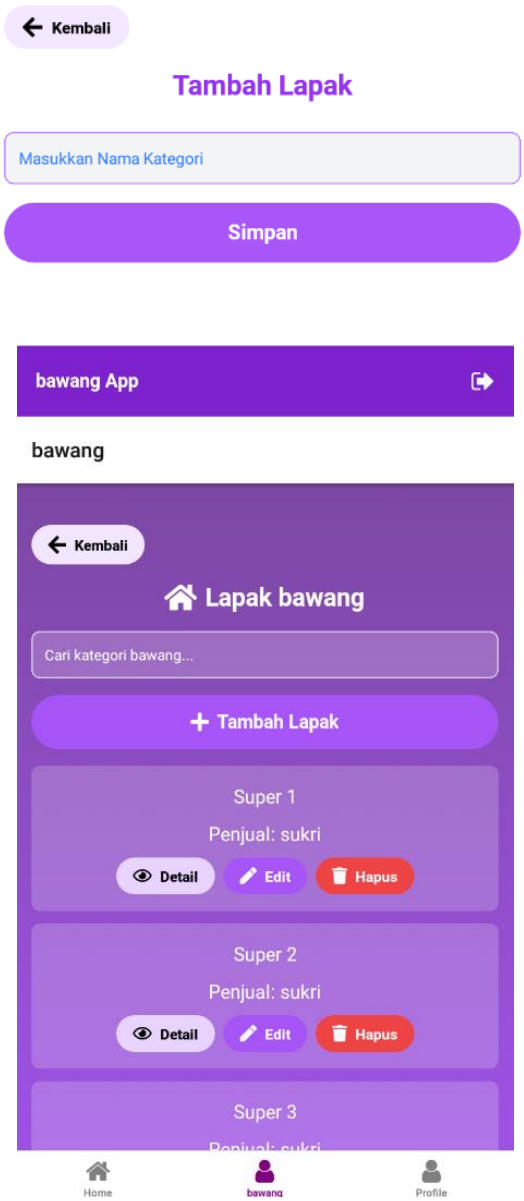
Tabel 4. 5 *Blackbox Testing* Tampilan Halaman lapak bawang

Test Faktor	Hasil	Keterangan
Masuk ke halaman lapak bawang	✔	Sistem akan menampilkan halaman lapak bawang
Screenshot		
		

e. *Blackbox Testing* tambah lapak

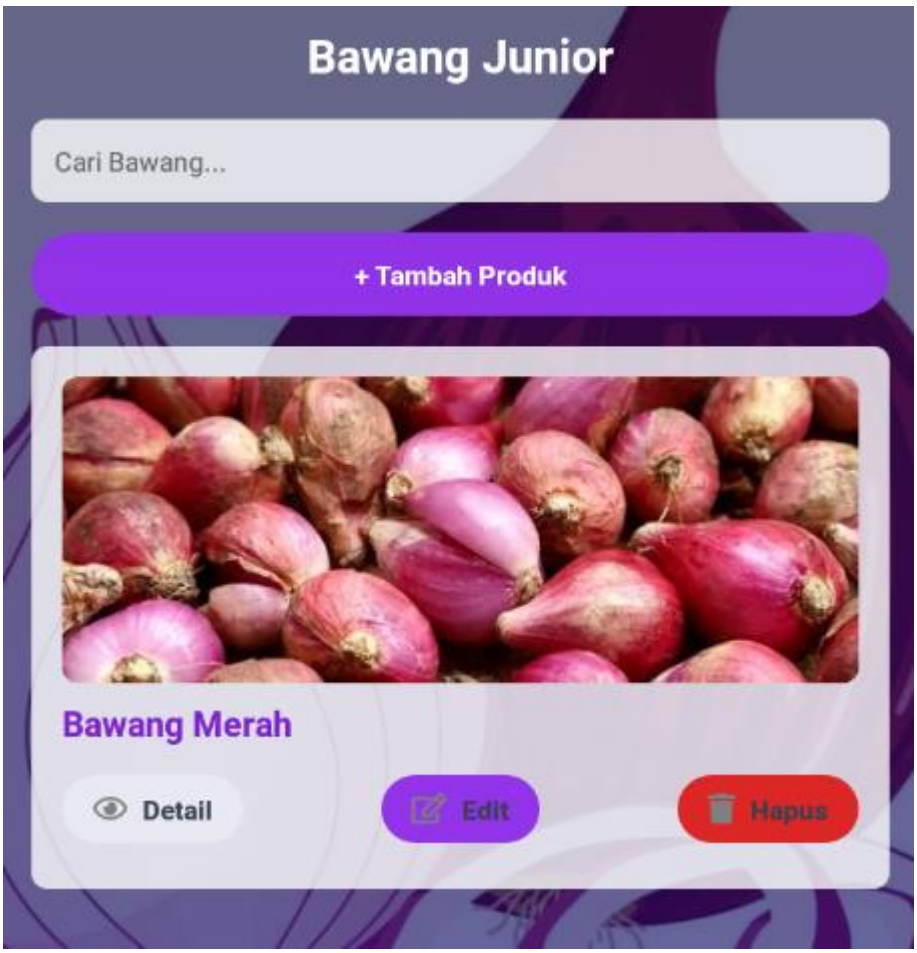
Tabel 4. 6 *Blackbox Testing* Tambah lapak

Test Faktor	Hasil	Keterangan
Masuk Ke Halaman Dashboard kemudian tambah lapak	✔	Sistem berhasil menambahkan lapak
Screenshot		



f. *Blackbox Testing* halaman detail lapak

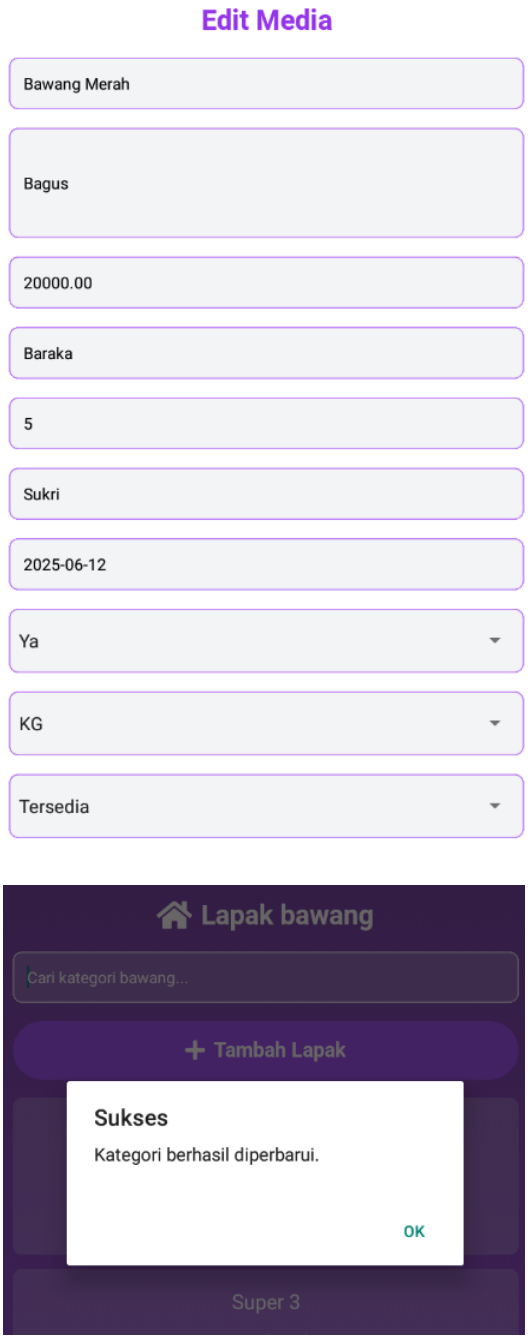
Tabel 4. 7 *Blackbox Testing* halaman detail lapak

Test Faktor	Hasil	Keterangan
User mengakses Halaman Detail Lapak	✔	Sistem berhasil menampilkan halaman detail produk
Screenshot		
		

g. *Blackbox Testing* halaman edit produk

Tabel 4. 8 *Blackbox Testing* halaman edit produk

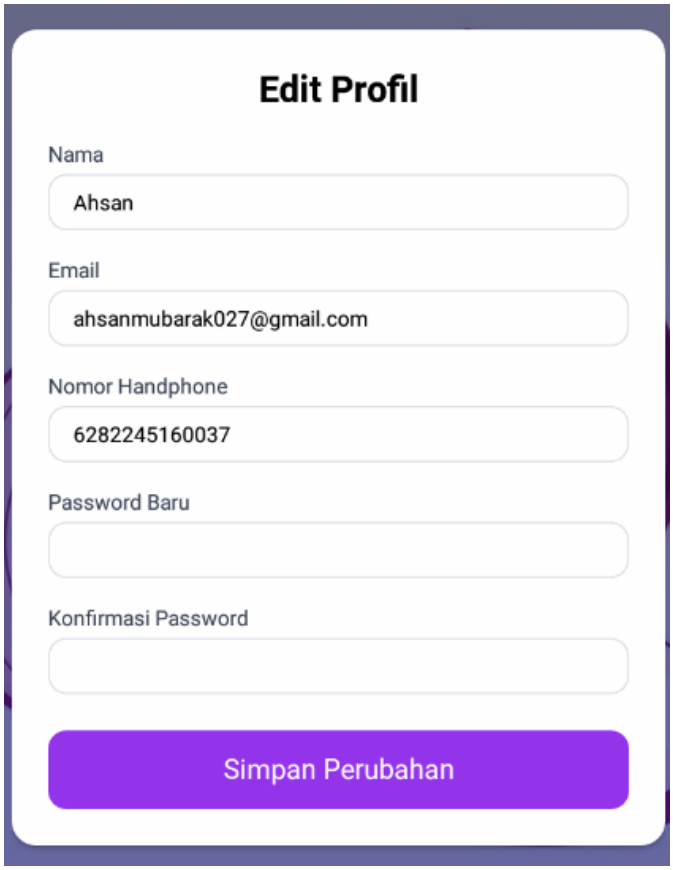
Test Faktor	Hasil	Keterangan
User mengakses Halaman Edit Produk	✔	Sistem berhasil menampilkan halaman detail produk
Screenshot		



h. *Blackbox Testing* halaman edit profil

Tabel 4. 9 *Blackbox Testing* halaman edit profil

Test Faktor	Hasil	Keterangan
User mengakses Halaman Edit Profil	✔	Sistem berhasil menampilkan halaman edit profile
Screenshot		

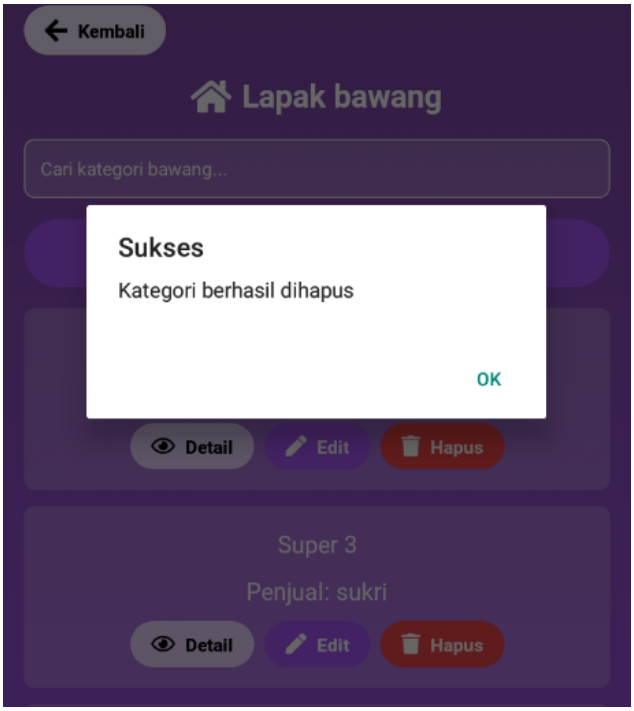


i. *Blackbox Testing* hapus kategori

Tabel 4. 10 *Blackbox Testing* Hapus kategori

Test Faktor	Hasil	Keterangan
User menghapus kategori	✔	Sistem berhasil menghapus kategori
Screenshot		





j. *Blackbox Testing* pencarian lapak

Tabel 4. 11 *Blackbox Testing* Pencarian Lapak

Test Faktor	Hasil	Keterangan
User mecari lapak	✔	Sistem akan menampilkan lapak yang dicari
Screenshot		



k. *Blackbox Testing* halaman daftar produk dalam lapak

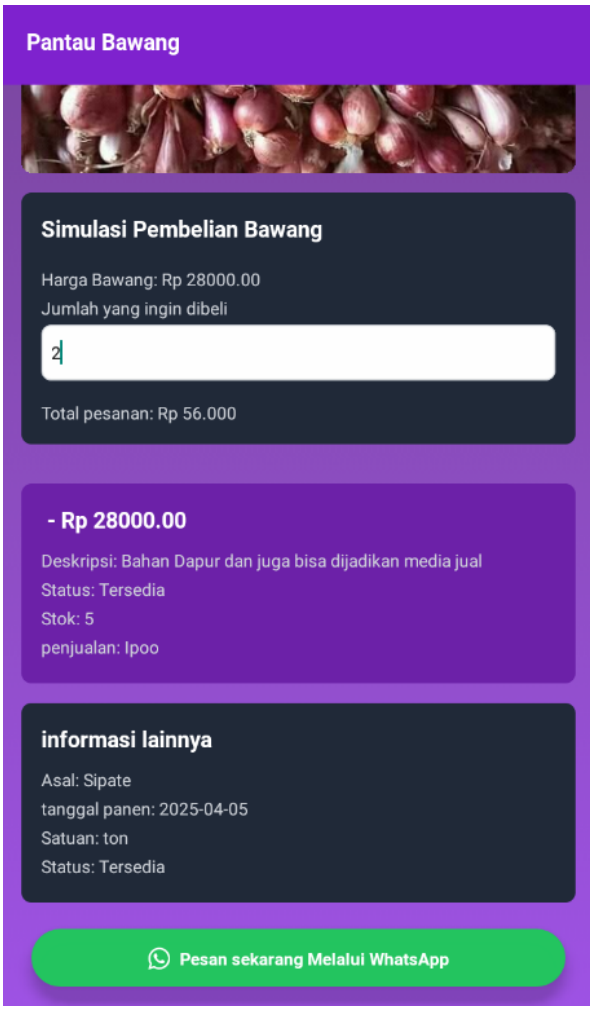
Tabel 4. 12 *Blackbox Testing* Halaman Daftar Produk Dalam Lapak

Test Faktor	Hasil	Keterangan
Melihat daftar produk pada satu lapak	✔	Sistem akan menampilkan produk dari merek yang di pilih
Screenshot		
		

1. *Blackbox Testing* halaman pembelian produk

Tabel 4. 13 *Blackbox Testing* halaman pembelian produk

Test Faktor	Hasil	Keterangan
Pembeli mengakses halaman pembelian produk	✔	Sistem berhasil menampilkan halaman pembelian produk
Screenshot		

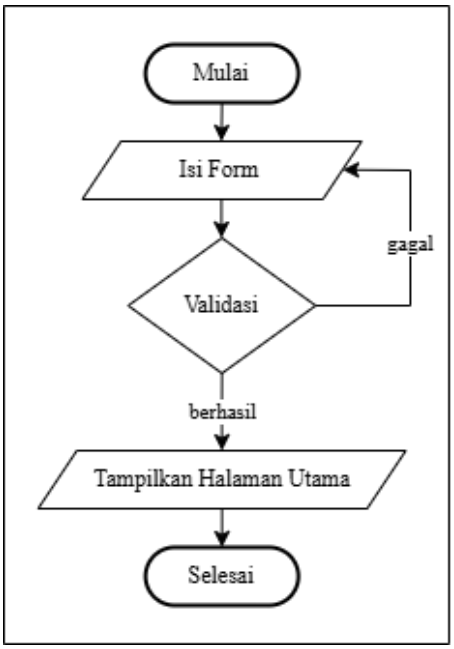


2. Whitebox Testing

Whitebox testing merupakan teknik pengujian aplikasi yang focus utamanya pada konfigurasi internal dan logika kode program. Penguji harus memahami cara kerja program di balik layar, termasuk alur logika, kondisi, dan loop, untuk memastikan bahwa setiap bagian kode berfungsi sebagaimana mestinya.

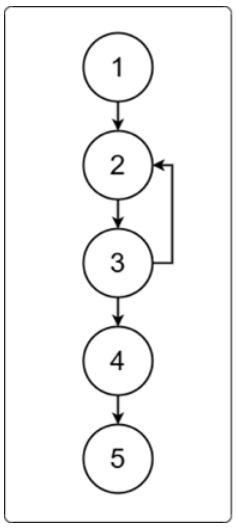
a) *Whitebox Testing* Halaman *Login*

1) *Flowchart*



Gambar 4. 23 *Flowchart* Halaman *Login*

2) *Flowgraph*



Gambar 4. 24 *Flowgraph* Halaman *Login*

Berdasarkan *Flowgraph* pada gambar 4. 24, maka dilakukan perhitungan sebagai berikut:

a. Menghitung Cyclomatic Complecity V(G) dari Edge dan Node :

V(G) = E – N + 2

N (*node*) = 5

E (*edge*) = 5

P (*predikat node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 2 – 3 – 4 – 5

Path 2 = 1 – 2 – 3 – 4 – 5

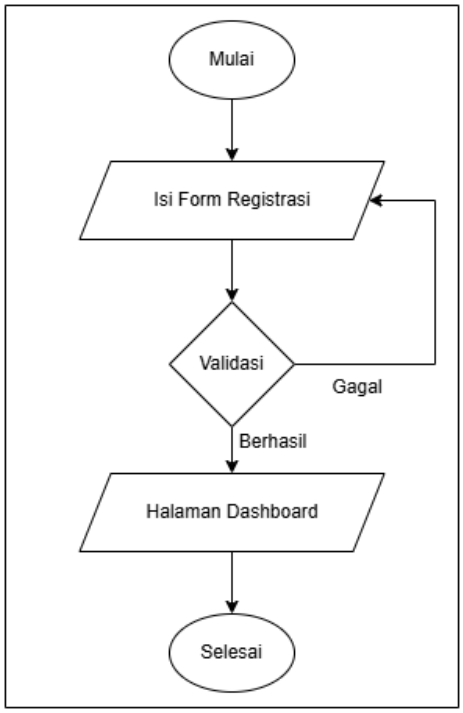
d. Grafik Matriks

Tabel 4. 14 Grafik Matriks *Login*

	1	2	3	4	5	E - 1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 1
5						1 – 1 = 0
Sum (E + 1)						1 + 1 = 2

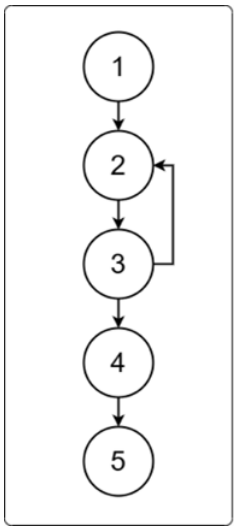
b) *Whitebox Testing* Halaman *Registrasi*

1) *Flowchart*



Gambar 4. 25 *Flowchart* Halaman *Registrasi*

2) *Flowgraph*



Gambar 4. 26 *Flowgraph* Halaman *Registrasi*

Berdasarkan *Flowgraph* pada gambar 4. 26, maka dilakukan perhitungan sebagai berikut:

a. Menghitung Cyclomatic Complecity V(G) dari Edge dan Node :

   V(G) = E – N + 2

   N (*node*) = 5

   E (*edge*) = 5

   P (*predikat node*) = 1

   Penyelesaian :

   V(G) = E – N + 2

   = 5 – 5 + 2

   = 2

   Predikat (P) = P + 1

   = 1 + 1

   = 2

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

   Path 1 = 1 – 2 – 3 – 2 – 3 – 4 – 5

   Path 2 = 1 – 2 – 3 – 4 – 5

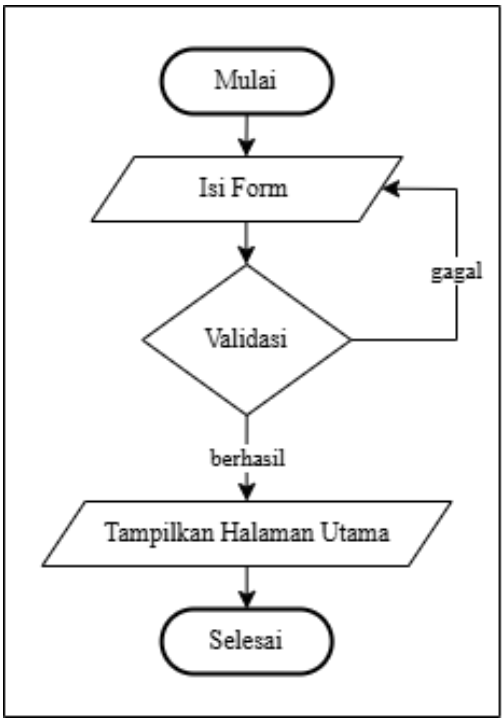
d. Grafik Matriks

Tabel 4. 15 Grafik Matriks Resgistrasi

	1	2	3	4	5	E - 1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 1
5						1 – 1 = 0
Sum (E + 1)						1 + 1 = 2

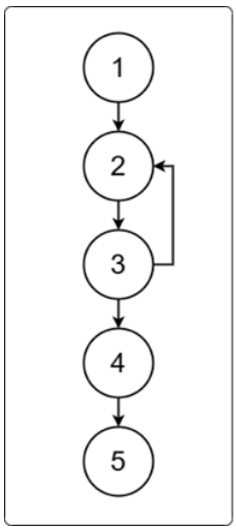
c) *Whitebox Testing* Halaman Tambah Lapak

1) *Flowchart*



Gambar 4. 27 *Flowchart* Tambah Lapak

2) *Flowgraph*



Gambar 4. 28 *Flowgraph* Tambah Lapak

Berdasarkan *Flowgraph* pada gambar 4. 28, maka dilakukan perhitungan sebagai berikut:

a. Menghitung Cyclomatic Complecity V(G) dari Edge dan Node :

V(G) = E – N + 2

N (*node*) = 5

E (*edge*) = 5

P (*predikat node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 2 – 3 – 4 – 5

Path 2 = 1 – 2 – 3 – 4 – 5

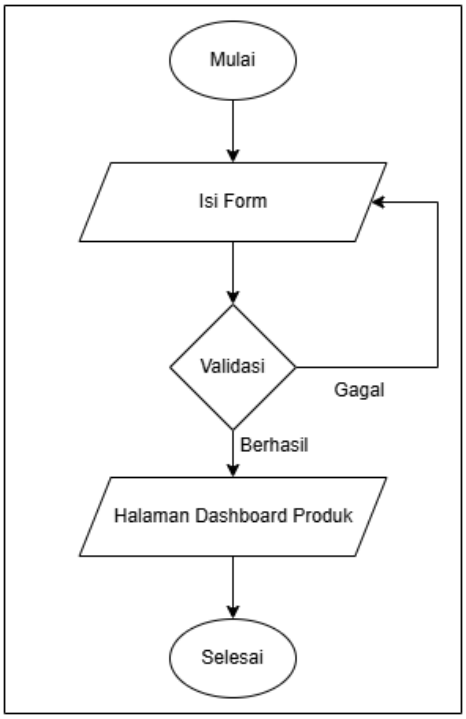
d. Grafik Matriks

Tabel 4. 16 Grafik Matriks Tambah Lapak

	1	2	3	4	5	E - 1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 1
5						1 – 1 = 0
Sum (E + 1)						1 + 1 = 2

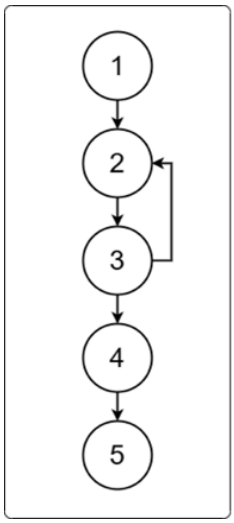
d) *Whitebox Testing* Tambah Produk

1) *Flowchart*



Gambar 4. 29 *Flowchart* Tambah Produk

2) *Flowgraph*



Gambar 4. 30 *Flowgraph* Tambah Produk

Berdasarkan *Flowgraph* pada gambar 4. 30, maka dilakukan perhitungan sebagai berikut:

a. Menghitung Cyclomatic Complecity V(G) dari Edge dan Node :

   V(G) = E – N + 2

   N (*node*) = 5

   E (*edge*) = 5

   P (*predikat node*) = 1

   Penyelesaian :

   V(G) = E – N + 2

   = 5 – 5 + 2

   = 2

   Predikat (P) = P + 1

   = 1 + 1

   = 2

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

   Path 1 = 1 – 2 – 3 – 2 – 3 – 4 – 5

   Path 2 = 1 – 2 – 3 – 4 – 5

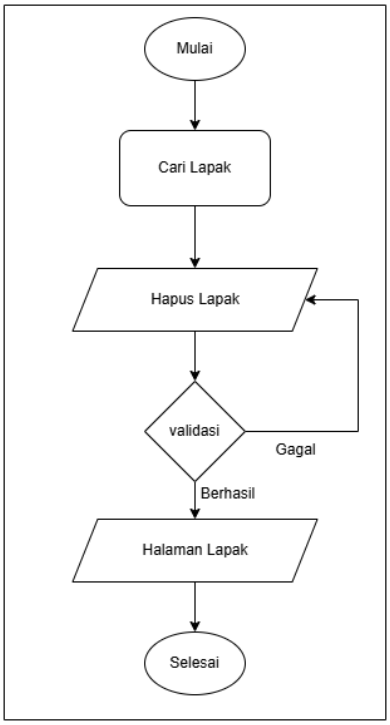
d. Grafik Matriks

Tabel 4. 17 Grafik Matriks Tambah Produk

	1	2	3	4	5	E - 1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 1
5						1 – 1 = 0
Sum (E + 1)						1 + 1 = 2

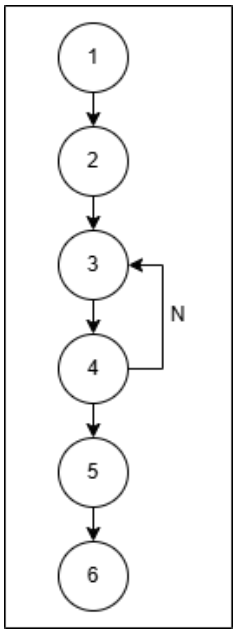
e) *Whitebox Testing* Hapus Lapak

1) *Flowchart*



Gambar 4. 31 *Flowchart* Hapus Lapak

2) *Flowgraph*



Gambar 4. 32 *Flowgraph* Hapus Lapak

Berdasarkan *Flowgraph* pada gambar 4. 32, maka dilakukan perhitungan sebagai berikut:

a. Menghitung Cyclomatic Complecity V(G) dari Edge dan Node :

   V(G) = E – N + 2

   N (*node*) = 6

   E (*edge*) = 6

   P (*predikat node*) = 1

   Penyelesaian :

   V(G) = E – N + 2

   = 6 – 6 + 2

   = 2

Predikat (P) = P + 1

= 1 + 1

= 2

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 4 – 5 – 6

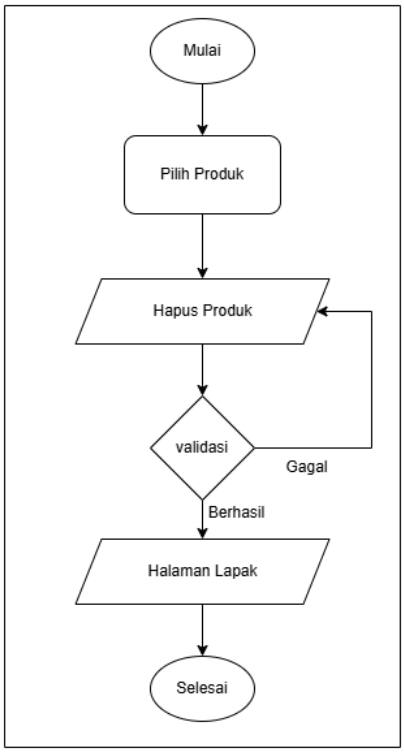
d. Grafik Matriks

Tabel 4. 18 Grafik Matriks Hapus Lapak

	1	2	3	4	5	6	E - 1
1		1					1 – 1 = 0
2			1				1 – 1 = 0
3				1			1 – 1 = 0
4			1		1		2 – 1 = 1
5						1	1 – 1 = 0
Sum (E + 1)							1 + 1 = 2

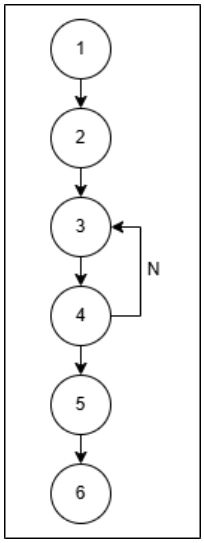
f) *Whitebox Testing* Hapus Produk

1) *Flowchart*



Gambar 4. 33 *Flowchart* Hapus Produk

2) *Flowgraph*



Gambar 4. 34 *Flowgraph* Hapus Produk

Berdasarkan *Flowgraph* pada gambar 4. 34, maka dilakukan perhitungan sebagai berikut:

a. Menghitung Cyclomatic Complecity V(G) dari Edge dan Node :

   V(G) = E – N + 2

   N (*node*) = 6

   E (*edge*) = 6

   P (*predikat node*) = 1

   Penyelesaian :

   V(G) = E – N + 2

   = 6 – 6 + 2

   = 2

   Predikat (P) = P + 1

   = 1 + 1

   = 2

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

   Path 1 = 1 – 2 – 3 – 2 – 3 – 4 – 5 – 6

   Path 2 = 1 – 2 – 3 – 4 – 5 – 6

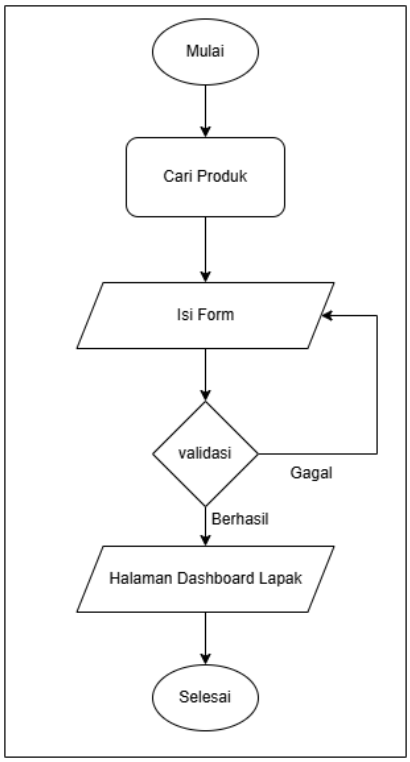
d. Grafik Matriks

Tabel 4. 19 Grafik Matriks Hapus Lapak

	1	2	3	4	5	6	E - 1
1		1					1 – 1 = 0
2			1				1 – 1 = 0
3				1			1 – 1 = 0
4			1		1		2 – 1 = 1
5						1	1 – 1 = 0
Sum (E + 1)							1 + 1 = 2

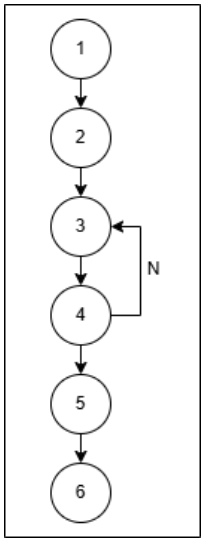
g) *Whitebox Testing* Cari Produk

1) *Flowchart*



Gambar 4. 35 *Flowchart* Cari Produk

2) *Flowgraph*



Gambar 4. 36 *Flowgraph* Cari Produk

Berdasarkan *Flowgraph* pada gambar 4. 36, maka dilakukan perhitungan sebagai berikut:

a. Menghitung Cyclomatic Complecity V(G) dari Edge dan Node :

V(G) = E – N + 2

N (*node*) = 6

E (*edge*) = 6

P (*predikat node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 2

c. *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 4 – 5 – 6

d. Grafik Matriks

Tabel 4. 20 Grafik Matriks Cari Produk

	1	2	3	4	5	6	E - 1
1		1					1 – 1 = 0
2			1				1 – 1 = 0
3				1			1 – 1 = 0
4			1		1		2 – 1 = 1
5						1	1 – 1 = 0
Sum (E + 1)							1 + 1 = 2

BAB V

PENUTUP

A. Kesimpulan

Berbagai temuan dalam studi dan evaluasi aplikasi, maka penulis menyimpulkan bahwa:

1. Aplikasi ini dikembangkan menggunakan *Android Studio* dengan bahasa *Java*Script, serta didukung basis data untuk mencatat dan mengelola data distribusi pasokan bawang merah.
2. Aplikasi mampu mencatat, memantau, dan mengelola data pasokan secara digital dan *real-time*. Sistem yang dibangun menyediakan fitur utama seperti *Registrasi* pengguna, *login*, tambah lapak, tambah produk, edit lapak, hapus produk, dan simulasi pembelian.
3. Pengujian *blackbox* dan *whitebox* menunjukkan bahwa aplikasi berjalan sesuai dengan yang dirancang. Black-box testing membuktikan bahwa fitur seperti *login*, *registrasi*, dan manajemen produk berfungsi dengan baik sesuai input dan output yang diharapkan.
4. Aplikasi ini memberikan manfaat nyata bagi seluruh pihak dalam rantai pasok: petani terbantu memantau dan memasarkan hasil panen, distributor lebih mudah mengakses data stok, dan pemerintah daerah dapat memanfaatkan data untuk kebijakan distribusi, stabilisasi harga, dan ketahanan pangan.

B. Saran

1. Mode offline dan sinkronisasi data otomatis :Mengingat sebagian pengguna mungkin berada di daerah dengan akses internet terbatas, disarankan aplikasi mendukung penggunaan secara offline dengan fitur sinkronisasi data otomatis saat koneksi tersedia..
2. Ekspansi ke komoditas pertanian lainnya: Ke depan, aplikasi ini dapat dikembangkan untuk mencakup komoditas pertanian lain seperti cabai, tomat, atau kentang, sehingga mampu menjadi solusi monitoring pasokan yang lebih menyeluruh dan berdampak luas..
3. Fitur pelaporan dan umpan balik dari pengguna: Perlu ditambahkan fitur yang memungkinkan petani, pedagang, maupun pengguna lainnya untuk memberikan laporan kondisi lapangan atau umpan balik terhadap aplikasi, agar pengembang bisa terus melakukan perbaikan sesuai kebutuhan nyata
4. Integrasi dengan sistem informasi pertanian daerah: Aplikasi sebaiknya dikembangkan agar dapat terhubung dengan sistem atau database dari dinas pertanian atau instansi terkait, guna memperkuat koordinasi dan validasi data pasokan antar wilayah.
5. Peningkatan akurasi data stok dan distribusi: Disarankan agar sistem pencatatan pasokan bawang merah ditingkatkan dengan dukungan teknologi sensor atau integrasi dengan perangkat IoT untuk mendapatkan data yang lebih akurat dan *real-time*.

DAFTAR PUSTAKA

Ayuni, I. K. (2024). Karakteristik Ekonomi Petani di Desa Sempu Kabupaten Kediri. *Indonesian Journal of Social Development*, *1*(3), 7–7. https://doi.org/10.47134/JSD.V1I3.2334

Cross-border, E. H.-, & 2022, undefined. (n.d.). Aplikasi Inventori Barang Menggunakan *Java* Dengan *PhpMyAdmin*. *Journal.Iaisambas.Ac.Id*. Retrieved June 28, 2025, from https://journal.iaisambas.ac.id/index.php/Cross-Border/article/view/1113

Gunawan, A. (2024). *Mobile Programming Menggunakan Flutter dan Visual Studio Code Untuk Pemula*. http://repository.unas.ac.id/id/eprint/11255

Gunawan, R., … A. Y.-E. J. E., & 2021, undefined. (n.d.). Rancang Bangun Sistem Presensi Mahasiswa Dengan Menggunakan Qr Code Berbasis Android. *Journal.Stekom.Ac.Id*. Retrieved June 28, 2025, from https://journal.stekom.ac.id/index.php/elkom/article/view/369

Harahap, N., Lestari, Y. M., & Siregar, A. Z. (2023). Pemberdayaan dalam Meningkatkan Kesejahteraan Petani Bawang Merah di Kecamatan Medan Marelan. *Jurnal Penyuluhan*, *19*(01), 170–180. https://doi.org/10.25015/19202340349

Husna, M., Komputer), P. R.-J. (Jurnal R., & 2021, undefined. (n.d.). Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Zabbix yang Terintegrasi dengan Grafana dan Telegram. *Academia.Edu*. Retrieved June 28, 2025, from https://www.academia.edu/download/113144641/2416.pdf

Lubis, N., Wasito, M., Marlina, L., … R. G.-A. J. I., & 2022, undefined. (2022). Respon pemberian ekoenzim dan pupuk organik cair terhadap pertumbuhan dan produksi bawang merah (Allium ascalonicum L). *Jurnal.Umsu.Ac.Id*, *25*(2). https://doi.org/10.30596/agrium.v25i2.10354

Margaretha, J., Syst, A. V.-J. (Journal I., & 2023, undefined. (n.d.). Perancangan Sistem Informasi Pemesanan Tiket Konser Musik Berbasis Web Menggunakan *Unified Modeling Language* (UML). *Pdfs.Semanticscholar.Org*. Retrieved June 28, 2025, from https://pdfs.semanticscholar.org/3c5e/c37a5726cdc19eb0972973d1a28212e7ed84.pdf

Marina, I., Sukmawati, D., Juliana, E., & Safa, Z. N. (2024). Dinamika Pasar Komoditas Pangan Strategis: Analisis Fluktuasi Harga Dan Produksi. *Paspalum: Jurnal Ilmiah Pertanian*, *12*(1), 160–168. https://doi.org/10.35138/PASPALUM.V12I1.700

Rosyidi, R., Informasi, S., Ppkia, S., & Paramita, P. (n.d.). IMPLEMENTASI WEBSCRAPING UNTUK PENCARIAN PRODUK DENGAN METODE

EXTENSIBLE PATH LANGUAGE (XPATH) PADA HALAMAN WEBSITE E. *Repo.Stimata.Ac.Id*. Retrieved June 28, 2025, from http://repo.stimata.ac.id/102/8/Jurnal_Artikel_16510010.pdf

Satria, A., Ramadhani, F., Jurnal, I. S.-W., & 2023, undefined. (n.d.). Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Sekolah Menengah Kejuruan Telkom 2 Medan Menggunakan Codeigniter. *Jurnal.Ilmubersama.Com*. https://doi.org/10.56211/wahana.v2i1.285

Septiani, A. P., Junardi, W., Amaliah, A., Bachtiar, A., Mahendra, J. I., & Muttaqin, M. I. (2020). Sistem Informasi Desa Siaga Pangan Menghadapi Covid19 berbasis Web Service. *Jurnal Sistem Cerdas*, *3*(3), 231–240. https://doi.org/10.37396/JSC.V3I3.63

Shadiq, J., Safei, A., For, R. L.-I. M., & 2021, undefined. (n.d.). Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan *Blackbox Testing*. *Academia.Edu*. Retrieved June 28, 2025, from https://www.academia.edu/download/101770609/1295.pdf

Supardi, I. (2021). *Semua bisa menjadi programmer JavaScript & Node. js*. https://books.google.com/books?hl=id&lr=&id=CG0qEAAAQBAJ&oi=fnd&pg=PP1&dq=*Java*script+adalah&ots=5zmart57n-&sig=ZJQbX6ccjNi9LwxzbyrdtVYINBM

Teknologi, M. S.-J. S. S. S. dan, & 2022, undefined. (n.d.). Implementasi White Box Testing Berbasis Path Pada Form *Login* Aplikasi Berbasis Web. *Jurnal.Unsil.Ac.Id*, *8*(1), 2022. Retrieved June 28, 2025, from https://jurnal.unsil.ac.id/index.php/jssainstek/article/view/4093

Zalukhu, A., Purba, S., Darma, D., Zalukhu[1], A., Purba[2], S., Darma[3], D., Teknik Informatika, M., & Industri, F. T. (2023). PERANGKAT LUNAK APLIKASI PEMBELAJARAN *FLOWCHART*. *Jurnal Teknologi, Informasi Dan Industri*, *4*(1), 61–70. http://ejurnal.istp.ac.id/index.php/jtii/article/view/351