BAB I

PENDAHULUAN

A. Latar Belakang

Kesadaran sosial, politik, maupun lingkungan pada kalangan konsumen semakin berkembang dengan pesat. Selain kualitas dan harga, dampak etis produk juga menjadi pertimbangan utama. Adanya perbedaan sudut pandang konsumen terhadap isu-isu sosial atau politik yang bertentangan dengan sudut pandang suatu merek, menjadi penyebab masyarakat tidak tertarik dengan merek perusahaan tersebut (Wang et al., 2021), sehingga tidak jarang mereka melakukan berbagai tindakan sebagai wujud protes. Salah satu bentuk protes yang muncul adalah boikot berbagai produk yang terafiliasi atau memiliki hubungan bisnis terhadap pihak yang dianggap melakukan pelanggaran (Mairiza et al., 2024). Melalui boikot, konsumen ingin menunjukkan ketidaksetujuan terhadap praktik tidak etis perusahaan dan mendorong perubahan positif. Boikot ini terbukti efektif dalam mengubah kebijakan perusahaan dan mendukung produk yang lebih berkelanjutan dan etis.

Fenomena boikot yang semakin meluas menghadirkan tantangan bagi konsumen, yakni kesulitan dalam memperoleh informasi akurat dan terkini mengenai status boikot suatu produk. Meskipun ada inisiatif dan kampanye yang mencatat daftar produk yang harus diboikot, informasi ini sering tersebar dan tidak terorganisir dengan baik (Li et al., 2019). Konsumen harus mencari informasi tersebut secara manual melalui media sosial, *website*, atau kampanye khusus, yang memakan waktu dan tidak praktis. Selain itu, banyak produk yang memiliki label

dengan informasi terbatas atau tidak mencantumkan status boikot, membuat identifikasi produk yang diboikot semakin sulit (Wang et al., 2021). Kondisi ini membuat konsumen kesulitan mengambil keputusan berbasis pertimbangan etis, meskipun mereka berniat mendukung produk yang sesuai dengan nilai-nilai mereka.

Solusi dalam mengatasi masalah ini adalah pengembangan aplikasi berbasis teknologi *Optical Character Recognition* (OCR), yang memungkinkan pemindaian dan pengenalan teks dari label produk. Aplikasi ini memungkinkan konsumen memindai label produk dengan perangkat *mobile* untuk mendapatkan informasi terkini apakah produk tersebut termasuk dalam daftar boikot . Dengan integrasi *database* yang selalu diperbarui, aplikasi ini dapat memberikan informasi cepat, akurat, dan *real-time*, memudahkan konsumen dalam membuat keputusan pembelian yang lebih etis (Yaser et al., 2023).

Penerapan OCR sudah luas, namun untuk deteksi boikot produk masih terbatas. Tantangan utama adalah variasi label produk yang beragam, yang dapat mempengaruhi akurasi OCR, terutama jika label buram atau rusak (Guimarães et al., 2023). Aplikasi juga harus terhubung dengan *database* yang memperbarui informasi boikot secara dinamis. Penelitian ini bertujuan mengembangkan aplikasi yang mengatasi masalah teknis OCR dan mengintegrasikan sistem basis data yang efisien dalam memperbarui informasi boikot (Yaser et al., 2023).

Aplikasi pendeteksi boikot berbasis OCR ini diharapkan menjadi solusi efektif bagi konsumen yang ingin mendukung produk dengan nilai etis tanpa harus menghabiskan waktu untuk mencari informasi secara manual. Dengan aplikasi ini,

konsumen dapat dengan mudah mengetahui status boikot suatu produk hanya dengan memindai labelnya (Lechterman et al., 2023). Keberhasilan pada aplikasi ini memberi dampak positif bukan hanya bagi konsumen, namun juga bagi perusahaan yang diharapkan akan semakin meningkatkan transparansi dan tanggung jawab sosial mereka (Wang et al., 2021).

B. Rumusan Masalah

Berdsarakan uraian latar belakang sebelumnya, penulis dapat menentukan rumusan masalah pada penelitian ini, yaitu bagaimana merancang sebuah Aplikasi Pendeteksi Boikot pada Produk Konsumen Berbasis *Optical Character Recognition* (OCR) ?

C. Tujuan

Penelitian ini bertujuan untuk merancang Aplikasi Pendeteksi Boikot Pada Produk Konsumen Berbasis *Optical Character Recognition* (OCR).

D. Batasan Masalah

Penelitian ini memiliki sejumlah batasan agar fokus tetap terjaga sesuai dengan ruang lingkup yang tetapkan. Berikut beberapa batasan pada penelitian ini:

1. Aplikasi ini hanya akan memindai kemasan produk yang menggunakan teks jelas, bahasa latin dan font standar.
2. Aplikasi ini tidak dapat memindai produk yang memiliki kemasan rusak, kotor ataupun terlipat.

3. Pemindaian tidak dapat dilakukan pada produk yang mengalami perubahan kemasan produk atau kemasan edisi khusus yang tidak sesuai dengan data yang ada.
4. Penelitian dibatasi hanya untuk produk kebutuhan konsumsi sehari-hari, tidak termasuk makanan atau minuman siap saji.

E. Manfaat Penelitian

Harapan penulis pada studi ini, yaitu:

1. Bidang Pendidikan

   Menjadikan referensi terhadap pengembangan penelitian kedepannya dan menambah wawasan mengenai penerapan algoritma deteksi.

2. Perusahaan yang menjadi sasaran boikot

   Memungkinkan perusahaan untuk merespons kampanye boikot dengan cepat dan efektif, misalnya dengan mengambil langkah-langkah perbaikan.

3. Konsumen

   Membantu pengambilan keputusan berbelanja guna mendukung gerakan boikot produk yang terafiliasi dengan tindakan merugikan

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Aplikasi

Application dalam Bahasa Indonesia yang merujuk pada penggunaan atau pengimplementasian. Executive Computer Dictionary mengutip bahwa aplikasi merupakan rancangan system yang memungkinkan seseorang dapat menyelesaikan tugas tertentu atau mengatasi masalah tertentu. Aplikasi adalah system operasi yang berjalan pada suatu komponen dan dibuat agar dapat melaksanakan kinerja atau fungsi individu, mulai dari pekerjaan sehari-hari hingga kebutuhan khusus yang lebih rumit (Huda & Priyatna, 2019). Aplikasi juga dapat berfungsi sebagai peningkatan efisiensi dan produktivitas dalam berbagai bidang, seperti Kesehatan, pendidikan, hiburan, dan bisnis.

Aplikasi merupakan perangkat lunak yang dirancang untuk memungkinkan pengguna menjalankan berbagai proses pada sistem komputer. Aplikasi terdiri dari berbagai file tertentu dengan kode di dalamnya, guna sebagai penghubung antara *user* dengan komputer guna mendukung interaksi serta eksekusi tugas tertentu. Program ini memungkinkan pengguna untuk berinteraksi dengan sistem dan perangkat keras untuk menjalankan tugas tertentu, seperti pengolahan data, komunikasi, atau hiburan (Wulandari et al., 2021). Aplikasi juga dapat disesuaikan dengan kebutuhan spesifik pengguna, memberikan kemudahan dalam melakukan berbagai aktivitas, mulai dari pekerjaan profesional hingga aktivitas sehari-hari.

Dengan berkembangnya teknologi, aplikasi kini memungkinkan untuk dijalankan di *platform-platform*, komputer, *smartphone*, maupun perangkat lainnya.

2. Android



Gambar 2. 1 Ikon *Android*

Sistem perangkat lunak yang berbasis Linux, dikenal dengan sebutan Android. Sistem ini dibentuk sebagai komponen perangkat portabel seperti tablet, ponsel, dan variasi lainnya. Awal dari system ini, dikelola oleh *Android Inc*. hingga di tahun 2005, Android diakuisisi oleh Google. Android saat ini, menjadi *platform* yang paling dikenal khalayak ramai (Permana, 2022). Android manghadirkan *platform* terbuka yang memfasilitasi pengembang untuk menciptakan aplikasi *mobile* menggunakan bahasa pemrograman *Java* dan *Kotlin*, yang dapat diunduh dan dipasang melalui Google Play Store.

Keunggulan utama Android adalah sifatnya yang *open-source*, yang memungkinkan siapa saja untuk mengakses dan memodifikasi kode sumbernya, memberikan fleksibilitas lebih dalam pengembangan aplikasi (Amrute et al., 2023). Dengan ekosistem yang besar dan berbagai perangkat yang mendukungnya, Android telah memungkinkan pengembang untuk menciptakan aplikasi dengan fungsionalitas yang sangat beragam, dari aplikasi produktivitas hingga hiburan, edukasi, hingga utilitas.

Android juga menunjang berbagai fitur canggih seperti GPS, sensor, kamera, maupun konektivitas jaringan, agar mempermudah pengembangan aplikasi yang memanfaatkan perangkat keras untuk berbagai keperluan (Zailani et al., 2021). Sebagai sistem operasi yang terus berkembang, Android tidak hanya fokus pada pengalaman pengguna yang lebih baik, tetapi juga pada keamanan, efisiensi energi, dan pengelolaan aplikasi yang lebih baik, dengan kemampuan yang serupa Android Studio, yang digunakan *Integrated Development Environment* (IDE) resmi untuk mengoptimalkan perangkat lunak pada Android. Sistem operasi ini juga menawarkan dukungan untuk berbagai arsitektur perangkat keras dan mendukung integrasi dengan layanan Google lainnya, seperti *Firebase* dan *Google Maps*, yang memperkaya fungsionalitas aplikasi yang dikembangkan.

Android terus berinovasi melalui pembaruan versi yang memperkenalkan fitur-fitur baru, seperti peningkatan antarmuka pengguna, manajemen daya yang lebih baik, dan keamanan aplikasi. Misalnya, fitur *Scoped Storage* pada versi Android terbaru membantu meningkatkan privasi pengguna dengan membatasi akses aplikasi ke data yang disimpan di perangkat (Akash Rawat et al., 2024). Selain itu, Android mendukung *Machine Learning* melalui pustaka seperti TensorFlow Lite, yang memungkinkan pengembang untuk menciptakan aplikasi berbasis kecerdasan buatan yang efisien dan berjalan langsung di perangkat pengguna.

Dalam dunia pengembangan aplikasi, Android menawarkan ekosistem alat bantu yang lengkap, seperti emulator bawaan di Android Studio untuk menguji aplikasi pada berbagai jenis perangkat. Selain itu, fitur *Live Updates* memungkinkan pengembang untuk melihat perubahan kode secara langsung di

perangkat selama proses pengembangan (Akash Rawat et al., 2024). Android juga mendukung *Jetpack Libraries*, yang menyediakan komponen siap pakai untuk pengelolaan data, antarmuka pengguna, dan navigasi, sehingga mempercepat pengembangan aplikasi.

Keunggulan lainnya, Android memiliki dukungan kuat untuk integrasi teknologi modern, seperti Augmented Reality (AR) melalui ARCore, yang memungkinkan pengembang menciptakan pengalaman imersif pada perangkat pengguna. Selain itu, dukungan untuk *Internet of Things* (IoT) melalui platform Android Things memungkinkan pengembang menciptakan solusi perangkat pintar yang terhubung dengan ekosistem Android. Komunitas pengembang yang besar, dokumentasi yang lengkap, serta dukungan dari Google menjadikan Android sebagai pilihan utama para *developer* dalam pembuatan perangkat lunak yang efisien, inovatif, dan kemudahan akses bagi pengguna di seluruh dunia.

3. Pendeteksi Boikot

Pendeteksi adalah proses pemeriksaan atau pengidentifikasian sesuatu menggunakan teknik serta metode tertentu. Proses ini bertujuan untuk menemukan atau mengenali objek, pola, atau informasi yang spesifik, baik secara manual maupun otomatis (Pambudi et al., 2020). Dalam konteks teknologi, pendeteksi sering melibatkan penggunaan perangkat atau sistem yang dapat menganalisis data atau kondisi secara *real-time*, seperti perangkat lunak untuk mendeteksi kesalahan, ancaman, atau bahkan pola perilaku pengguna.

Pendeteksi boikot adalah sistem yang digunakan untuk mengidentifikasi produk, merek, atau perusahaan yang termasuk dalam daftar boikot, yang biasanya

dipicu oleh masalah sosial, politik, atau etika tertentu. Proses pendeteksian ini mencakup analisis serta pengumpulan data di berbagai sumber seperti berita, media sosial, dan kampanye konsumen, guna memantau perkembangan boikot yang dapat berlangsung secara dinamis. Dengan memanfaatkan teknologi seperti machine learning, OCR, dan analisis big data, pendeteksi boikot dapat mengenali pola-pola atau sentimen negatif terhadap suatu produk atau merek, serta mengonfirmasi status boikot secara *real-time* (Yaser et al., 2023). Pendeteksi ini berfungsi membantu konsumen dalam penentuan keputusan pembelian yang lebih etis, sesuai dengan nilai-nilai sosial yang mereka yakini, dan memberikan transparansi lebih terhadap kebijakan perusahaan, sekaligus mendorong perusahaan untuk lebih bertanggung jawab dalam menjalankan praktik bisnis yang berkelanjutan dan etis (Wang et al., 2021).

4. Optical Character Recognition (OCR)

Pengenalan Karakter Optik yang sering di sebut OCR merupakan alat transformasi berbagai bentuk dokumen cetak atau tulisan tangan yang terdeteksi dalam gambar menjadi teks yang dapat diedit, dicari, dan diproses secara digital. OCR berfungsi dengan menganalisis gambar atau dokumen yang dipindai menggunakan algoritma pengenalan pola untuk mengidentifikasi karakter-karakter yang ada dalam gambar tersebut, kemudian mengonversinya menjadi teks yang bisa digunakan lebih lanjut dalam berbagai aplikasi.

Pada dasarnya, proses OCR melibatkan beberapa tahap, yaitu praproses gambar, pemisahan karakter, pengenalan karakter, dan pasca-pemrosesan teks untuk meningkatkan akurasi (Faizullah et al., 2023). Teknologi ini sangat berguna

dalam bidang otomasi, terutama untuk menggantikan proses penginputan secara manual yang pastinya memakan waktu dan tak jarang rawan kesalahan. OCR banyak digunakan dalam berbagai aplikasi, seperti pemindaian dokumen, pengenalan nomor plat kendaraan, digitalisasi arsip, serta aplikasi yang mengharuskan pemindai teks, seperti pembacaan label produk atau nomor seri pada barang.

Dalam konteks aplikasi berbasis *mobile*, OCR memungkinkan pengguna untuk memindai teks dalam bentuk gambar menggunakan perangkat ponsel mereka, yang kemudian diproses dan diterjemahkan menjadi informasi yang berguna, misalnya dalam aplikasi untuk mengidentifikasi produk yang diboikot. Seiring dengan perkembangan teknologi, OCR kini telah semakin akurat dan dapat mengenali berbagai jenis tulisan, *font*, serta simbol dalam berbagai bahasa.

Beberapa solusi OCR yang terkenal, seperti *Tesseract* dan *Google Vision* OCR, kini banyak digunakan dalam pengembangan aplikasi karena kemampuan mereka yang terbukti dalam mengenali teks dengan tingkat akurasi yang tinggi (Kusnantoro et al., 2022). Meskipun demikian, tantangan utama dalam pengembangan aplikasi berbasis OCR adalah kemampuan sistem untuk menangani teks yang terdistorsi, kabur, atau dengan kualitas rendah, yang sering kali mempengaruhi akurasi hasil pemrosesan OCR.

Perkembangan lebih lanjut dalam pembelajaran mesin (*machine learning*) dan *Artificial Intelligence* (AI), baik itu kualitas maupun kemampuan OCR semakin meningkat, membuka lebih banyak potensi penggunaan teknologi ini dalam

berbagai sektor, termasuk dalam pengembangan aplikasi mobile yang membutuhkan pemrosesan teks secara otomatis dan *real-time* (Yaser et al., 2023).

5. ML Kit Text Recognition

ML Kit merupakan sebuah mobile SDK yang dikembangkan oleh Google untuk menyediakan berbagai kemampuan kecerdasan buatan (AI) secara langsung pada perangkat (on-device) maupun melalui layanan cloud. Salah satu fitur yang tersedia dalam ML Kit adalah Text Recognition, yaitu kemampuan untuk mengenali dan mengekstrak teks dari gambar atau hasil tangkapan kamera secara otomatis. Fitur ini mendukung pengenalan teks dalam berbagai bahasa dan bentuk, termasuk teks cetak dan tulisan tangan (handwriting recognition) pada versi tertentu.

Text Recognition pada ML Kit bekerja dengan memproses citra sebagai input, kemudian mengidentifikasi blok teks, baris, dan kata menggunakan machine learning model yang telah dilatih sebelumnya. Proses ini dapat berjalan secara real-time dan offline tanpa memerlukan koneksi internet, selama menggunakan model on-device. Keunggulan utama dari ML Kit Text Recognition antara lain adalah kemudahan integrasi dalam aplikasi Android, kecepatan deteksi, serta fleksibilitas dalam berbagai skenario penggunaan, seperti pemindaian dokumen, deteksi teks pada label produk, dan kebutuhan OCR (Optical Character Recognition) lainnya.

Dalam penelitian ini, ML Kit Text Recognition digunakan untuk mendeteksi teks pada kemasan produk, khususnya nama produk, yang selanjutnya akan dicocokkan dengan database guna mengidentifikasi status boikot produk tersebut. Pemilihan ML Kit didasarkan pada kemampuannya yang efisien, ringan, serta

kompatibel dengan berbagai perangkat Android tanpa perlu ketergantungan terhadap layanan pihak ketiga seperti Firebase.

6. Application Programming Interface (API)

Interoperabilitas antar system pada perangkat lunak modern menjadi kebutuhan yang sangat penting. Banyak aplikasi dan layanan saat ini tidak berdiri sendiri, melainkan saling terhubung dan bertukar data guna memberi pengalaman yang lebih baik untuk pengguna. Untuk mewujudkan hal ini, dibutuhkan mekanisme yang memungkinkan komunikasi antar sistem secara efisien dan terstandarisasi.

Application Programming Interface (API), merupakan ide dalam pengembangan aplikasi yang memberi kesempatan kepada pihak luar untuk mengakses serta memanfaatkan aplikasi tanpa perlu merubah skema kode inti atau data sistem. API juga mendukung interaksi antara berbagai sistem, meski beroperasi di atas platform yang tidak sama. (Kusnantoro et al., 2022). API bertindak sebagai jembatan antara dua sistem yang berbeda, memungkinkan pertukaran data dan perintah dengan cara yang aman dan terstruktur.

Implementasi API kini sangat luas, mulai dari penggunaan dalam aplikasi mobile, web, hingga sistem cloud. Contohnya, aplikasi cuaca dapat mengambil data dari layanan pihak ketiga melalui API tanpa harus mengelola data cuaca itu sendiri. Selain itu, API juga memungkinkan integrasi layanan pembayaran, media sosial, dan lainnya ke dalam aplikasi, sehingga pengembang dapat lebih fokus pada logika bisnis utama tanpa harus membangun seluruh fungsionalitas dari awal.

7. Android Studio



Gambar 2. 2 Ikon *Android Studio*

Perangkat lunak pengembangan resmi dari Google di kenal dengan sebutan Android Studio, bertujuan untuk mempermudah proses optimisasi aplikasi Android. IDE ini dirancang untuk menunjang *developer* dalam pembuatan perangkat lunak Android dengan menyediakan berbagai alat dan fitur canggih. Android Studio berbasis *IntelliJ* IDEA, memberikan pengalaman pengembangan yang kaya dan efisien (Kurbatova et al., 2021).

Adanya dukungan penuh untuk bahasa pemrograman Java dan Kotlin, Android Studio menyediakan berbagai fitur seperti visual layout editor, emulator Android, dan debugging tools untuk membantu pengembang membangun, menguji, dan mengoptimalkan aplikasi Android secara lebih mudah dan cepat. Android Studio juga secara teratur diperbarui untuk mendukung API terbaru dan perangkat Android terkini, sehingga menjadi pilihan utama bagi *developer* aplikasi Android.

Banyaknya fitur-fitur canggih yang ditawarkan Android Studio untuk memberikan kemudahan serta efisiensi dalam seluruh proses pengembangan aplikasi Android, mulai dari perancangan antarmuka pengguna hingga pengujian dan optimasi performa aplikasi. Keunggulannya yang terus diperbarui menjadikannya alat yang sangat diperlukan untuk menciptakan aplikasi yang

berkualitas tinggi dan kompatibel dengan berbagai perangkat Android. Beberapa kelebihan utama yang dimiliki Android Studio antara lain:

- *Visual Layout Editor*: Membantu pengembang dalam mendesain UI aplikasi dnegan *drag-and-drop* dengan tampilan visual yang intuitif.
- *Emulator Android*: Menyediakan emulator sebagai penguji aplikasi di berbagai perangkat dan versi Android tanpa memerlukan perangkat fisik.
- Integrasi dengan Git: Memudahkan pengelolaan kode sumber dan kolaborasi tim dengan fitur integrasi Git yang kuat.
- Kotlin *Support*: Mendukung bahasa pemrograman Kotlin secara penuh, yang semakin populer di kalangan pengembang Android.

Android SDK *Manager*: Memudahkan pengelolaan berbagai versi SDK dan dependensi Android yang diperlukan dalam pengembangan aplikasi.

8. phpMyAdmin



Gambar 2. 3 Ikon phpMyAdmin

phpMyAdmin merupakan suatu perangkat atau aplikasi gratis berbasis *opensource* untuk melakukan pemrograman ataupun administrasi pada database MySQL (Ramadhan & Mukhaiyar, 2020). Aplikasi ini menyediakan antarmuka berbasis web yang memudahkan pengguna dalam mengelola database, seperti membuat, mengedit, menghapus tabel dan data, serta menjalankan perintah SQL

tanpa harus menggunakan command line. Selain itu, *phpMyAdmin* juga mendukung berbagai fitur lainnya seperti impor dan ekspor data, manajemen hak akses pengguna, serta pemantauan performa database, sehingga sangat membantu dalam pengelolaan basis data secara efisien dan praktis.

PhpMyAdmin pertama kali dikembangkan pada tahun 1998 oleh Tobias Ratschiller, seorang pengembang asal Austria. Awalnya, proyek ini dibuat sebagai solusi pribadi untuk memudahkan administrasi MySQL melalui antarmuka web. Adanya popularitas dan kebutuhan yang tinggi, membuat proyek ini menjadi open source lalu untuk lebih lanjutnya dikembangkan oleh komunitas. PhpMyAdmin mendukung berbagai sistem operasi misalnya macOS, Linux, Windows, serta kompatibel dengan berbagai versi MariaDB dan MySQL. Karena kemudahan instalasi dan penggunaannya, PhpMyAdmin banyak digunakan oleh pengembang web, administrator server, maupun pengguna umum yang ingin mengelola database secara lebih praktis.

Sampai saat ini, PhpMyAdmin terus diperbarui dan digunakan sebagai salah satu alat manajemen database MySQL yang populer secara luas di seluruh dunia. Selain itu, dokumentasi yang lengkap serta kekuatan oleh dukungandari berbagai komunitas membuat PhpMyAdmin menjadi pilihan utama dalam berbagai proyek pengembangan web, baik berskala kecil maupun besar.

9. Visual Studio Code (VS Code)



Gambar 2. 4 Ikon *Visual Studio Code*

Aplikasi desktop ringan namun Tangguh yang berfungsi sebagai kode editor dikenal dengan sebutan *Visual Studio Code*. Aplikasi ini mendukungan bawaan bahasa pemrograman seperti *Node.js, JavaScript*, dan *TypeScript*, serta bermacam-macam ekstensi untuk bahasa lainnya, termasuk *PHP, C#*, *C++*, dan *Python* (Hartati, 2020).

Pengguna aplikasi ini, dapat menambahkan berbagai ekstensi dari marketplace untuk memperluas fungsionalitasnya, termasuk dukungan untuk *PHP, C#*, *C++*, dan *Python*. Banyaknya keunggulan *Visual Studio Code* menjadikan aplikasi ini sebagai editor serbaguna yang sesuai dengan kebutuhan para *developer*.

Visual Studio Code juga memiliki berbagai fitur unggulan seperti integrasi dengan sistem kontrol versi seperti *Git*, *debugger* terintegrasi, *terminal* bawaan, serta kemampuan untuk bekerja secara kolaboratif melalui ekstensi seperti *Live Share*. Semua fitur ini hadir dalam sebuah aplikasi yang ringan, menjadikannya ideal digunakan pada sistem operasi seperti Windows, macOS, dan Linux.

10. Xampp



Gambar 2. 5 Ikon Xampp

XAMPP merupakan perangkat lunak gratis yang dimanfaatkan pada berbagai platform sistem operasi. Perangkat ini adalah paket lengkap yang berisi beberapa program penting dan berfungsi sebagai server lokal (*localhost*) yang bisa dijalankan secara mandiri. Di dalamnya terdapat komponen utama seperti database MySQL, *Apache HTTP Server*, serta interpreter untuk bahasa pemrograman *Perl* dan PHP. (Hartiwati, 2022).

XAMPP bertujuan untuk memudahkan proses pengugjian dan pengembangan aplikasi web lokal sebelum diupload ke server sebenarnya. Dengan menyediakan lingkungan yang lengkap, XAMPP memungkinkan pengembang untuk menjalankan situs web dinamis dan sistem manajemen basis data tanpa memerlukan koneksi internet atau konfigurasi server yang kompleks.

Dari berbagai *open source* yang ada, XAMPP memiliki antarmuka pengguna yang sangat sederhana serta mudah digunakan untuk pemula. Instalasi dan pengelolaan layanan seperti *Apache*, *MySQL*, dan *FileZilla* dapat dilakukan hanya dengan beberapa klik. Hal ini menjadikan XAMPP sebagai salah satu solusi favorit bagi pelajar, mahasiswa, maupun pengembang profesional yang membutuhkan lingkungan server lokal yang cepat dan praktis.

Tidak hanya untuk pengembangan, XAMPP juga sering digunakan sebagai media pembelajaran dalam memahami konsep dasar server, bahasa pemrograman

web, dan pengelolaan basis data. Karena sifatnya yang fleksibel dan universal, XAMPP dapat dioperasikan pada macOS, Windows, dan Linux tanpa banyak penyesuaian.

11. Kotlin



Gambar 2. 6 Ikon Kotlin

Kotlin merupakan bahasa pemrograman modern yang dioptimalkan oleh JetBrains dan pertama kali diumumkan pada tahun 2011. Seiring dengan pengumuman bahwa Kotlin sebagai bahasa resmi untuk perancangan aplikasi Android oleh Google pada tahun 2017, Kotlin semakin populer di kalangan pengembang *Android*. Bahasa ini dirancang untuk sepenuhnya *interoperabel* dengan Java, yang menjadikannya pilihan yang sangat menarik bagi pengembang Java yang ingin memanfaatkan fitur-fitur modern tanpa harus meninggalkan ekosistem Java yang sudah mapan (Courtney & Neilsen, 2019).

Kotlin menawarkan berbagai fitur canggih, seperti tipe data yang lebih kuat, pengelolaan *null* yang lebih aman, sintaksis yang lebih ringkas, dan kemampuan untuk menghindari banyak *boilerplate code* yang umumnya ditemukan pada Java. Salah satu keunggulan utama Kotlin adalah kemampuannya untuk mengurangi kemungkinan kesalahan runtime melalui penggunaan tipe yang lebih ketat dan pengecekan *null* yang lebih ketat (Kozub & Kozub, 2023).

Selain itu, Kotlin mendukung pemrograman fungsional dan pemrograman berorientasi objek, memberikan fleksibilitas yang lebih besar bagi pengembang

dalam menulis kode yang bersih dan efisien. Dengan dukungan dari Google dan ekosistem Android yang luas, Kotlin kini menjadi pilihan utama untuk pengembangan aplikasi Android dan semakin banyak digunakan dalam proyek-proyek *software* yang berskala besar.

Keunggulan lainnya, Kotlin memiliki dukungan untuk *coroutines*, yang mempermudah pengelolaan operasi *asinkron*, seperti pemanggilan API atau pengolahan data secara *parallel*. Dengan fitur ini, pengembang dapat menulis kode asinkron yang lebih sederhana dan mudah dibaca dibandingkan dengan menggunakan *callbacks* atau *threads* secara manual .

Kotlin juga mendukung *multiplatform development*, sehingga pengembang dapat mengimplementasikan logika program di berbagai platform, seperti iOS, Android, backend, dan web (Kozub & Kozub, 2023). Dengan fitur ini, Kotlin membantu mengurangi usaha pengembangan dan meningkatkan konsistensi antara berbagai aplikasi pada ekosistem yang sama.

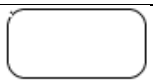
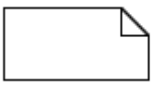
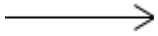
Dengan alat bantu seperti *Kotlin Extensions*, pengembang dapat menambahkan fungsionalitas baru pada kelas yang ada tanpa perlu mengubah kode aslinya. Selain itu, Kotlin juga memiliki pustaka yang terus berkembang, seperti *Ktor* untuk pengembangan aplikasi web dan server, serta *Anko* yang mempermudah pengembangan antarmuka pengguna di Android. Berkat dokumentasi yang lengkap, komunitas yang terus berkembang, dan kompatibilitas penuh dengan Java, Kotlin kini menjadi bahasa yang ideal untuk pengembang modern yang ingin menciptakan aplikasi yang cepat, aman, dan mudah dipelihara.

12. Unified Modeling Language (UML)

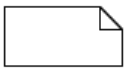
Bahasa sederhana yang diterapkan secara luas di berbagai sektor, bertujuan merumuskan kecukupan sistem, menyusun desain dan analisis, serta pemodelan arsitektur pada konteks pengembangan berbasis objek di kenal dengan sebutan UML (Putra & Andriani, 2019). UML menyajikan berbagai diagram dalam penggambaran struktur dan perilaku sistem, misalnya diagram use case, diagram *sequence*, dan diagram *class*, yang memudahkan pengembang memahami dan merancang mekanisme secara visual.

Adanya UML, pengembang dapat merencanakan aplikasi secara lebih terstruktur dan efisien. UML banyak digunakan diberbagai jenis pengembangan perangkat lunak, mulai dari aplikasi desktop hingga *mobile*, serta membantu dalam mengelola kompleksitas sistem dan memastikan rekayasa perangkat lunak yang baik.

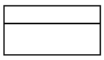
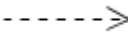
Tabel 2. 1 Simbol *State Chart Diagram*

No	Gambar	Nama	Keterangan
1		*State*	Atribut data dan atribut tautan suatu objek pada titik tertentu.
2		*Node*	Komponen fisik yang ada selama eksekusi aplikasi dan mewakili sumber daya komputasi.
3		*Initial Pseudo State*	Bagaimana objek dibuat atau diinisiasi.
4		*Final State*	Bagaimana objek dibuat dan diakhiri.
5		*Transition*	Peristiwa pmemicu status objek dengan memperbarui satu atau beberapa nilai atributnya.
6		*Association*	Apa yang menghubungkan satu objek dengan objek lainnya.

Tabel 2. 2 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. 3 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		*Association*	Hubungan kondisi objek turunan memiliki kesamaan perilaku dan struktur data dengan objek induk mereka.
2		*Nary Association*	Hindari mengaitkan lebih dari dua objek.
3		*Class*	Sekumpulan objek yang memiliki atribut dan metode yang serupa.
4		*Realization*	Aktivitas konkret yang dilakukan oleh suatu objek.
5		*Dependency*	Hubungan di mana perubahan pada elemen yang berdiri sendiri mempengaruhi elemen yang tergantung (elemen yang tidak berdiri sendiri).
6		*Association*	Apa yang menghubungkan satu objek dengan objek lainnya.
7		*Collaboration*	Penjelasan tentang rangkaian tindakan yang dilakukan oleh sistem, yang menghasilkan hasil terukur bagi pengguna

Tabel 2. 4 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.

B. Penelitian Terdahulu

Studi terdahulu berupa hasil penelitian terlaksana oleh pihak lain dan memiliki keterkaitan dengan topik yang sedang dikaji. Referensi terhadap penelitian-penelitian ini berperan penting dalam membentuk dasar teoritis yang kokoh bagi studi baru, meningkatkan kualitas analisis, serta mendorong perkembangan ilmu pengetahuan secara berkelanjutan (Putra & Andriani, 2019). Beberapa penelitian terdahulu yang dilakukan:

1. Shierly Mayco Angela, Ade Eviyanti, Metatia Intan Mauliana (2024) Universitas Muhammadiyah Sidoarjo dalam Jurnal TEKNIKOM yang berjudul **"Pengembangan Teknologi *Optical Character Recognition* di *Flutter* Berupa Deteksi Teks Pada Gambar"**. Penelitian ini bertujuan mengembangkan teknologi OCR berbasis *Flutter* untuk mendeteksi teks pada gambar dengan cepat dan akurat menggunakan *Tesseract* OCR. Dengan pendekatan Rapid Application Development (RAD), aplikasi diuji dnegan *Black Box* serta *User Acceptance Testing* (UAT), menghasilkan tingkat kepuasan pengguna sebesar 88%. Aplikasi ini fleksibel, mudah digunakan, dan bermanfaat untuk berbagai keperluan seperti pencarian teks dalam gambar dan identifikasi plat nomor kendaraan.
2. Anisya Sonita, Khairunnisyah (2018) Universitas Muhammadiyah Bengkulu dalam Jurnal Informatika UPGRIS yang berjudul **"Aplikasi Pendeteksi Obat dan Makanan Menggunakan OCR (Optical Character Recognition)"**. Penelitian ini bertujuan untuk meningkatkan kinerja perangkat lunak berbasis Android dengan menerapkan teknologi OCR dan metode pencocokan string

untuk menilai kecocokan produk farmasi dan makanan berdasarkan informasi dari BPOM. Temuan dari penelitian ini menunjukkan bahwa aplikasi mampu mencocokkan data dengan tingkat akurasi yang tinggi, sehingga mempermudah proses pengenalan produk yang telah terdaftar secara resmi.

3. Anggi Sanjaya Kusnanto (2014) Universitas Komputer Indonesia dalam Jurnal Ilmiah Komputer dan Informatika (KOMPUTA) yang berjudul "**Implementasi OCR (*Optical Character Recognition*) Menggunakan Metode *String Matching* untuk Mendeteksi Obat Dan Makanan Berbasis Android"**. Tujuan dari studi ini adalah untuk menerapkan teknologi pembacaan karakter optik (OCR) yang berbasis Android dengan metode pencocokan string guna mengenali informasi terkait obat dan makanan, sehingga mempermudah masyarakat dalam mengakses data dari BPOM. Hasilnya menunjukkan aplikasi ini efektif mengenali karakter pada gambar produk, menghasilkan tingkat keberhasilan hingga 96% dalam mendeteksi kesesuaian informasi.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Pendekatan studi ini menerapkan metode *Research and Development* (RnD) dengan model prototipe untuk merancang, menciptakan, dan menguji aplikasi yang mampu mendeteksi boikot dengan Optical Character Recognition (OCR). Proses pengembangan dilakukan melalui tahapan perancangan antarmuka, integrasi teknologi OCR untuk mendeteksi teks nama produk, serta koneksi ke basis data yang berisi daftar produk boikot. Evaluasi dilakukan secara kuantitatif untuk menilai kinerja sistem, yang difokuskan pada dua aspek utama, yaitu akurasi hasil pendeteksian teks oleh OCR hingga menghasilkan output yang relevan (Li et al., 2019). Pengujian ini bertujuan untuk memvalidasi perkembangan aplikasi dengan kemampuan fungsi secara maksimal dan efektif dalam mendeteksi produk yang termasuk dalam daftar boikot.

B. Lokasi dan Waktu

Kajian studi ini dilaksanakan di Marauleng, Kecamatan Suppa, Kabupaten Pinrang. Selain itu, penelitian juga dilaksanakan secara *online* untuk mengakses artikel dan sumber literatur terpercaya terkait produk boikot. Waktu pelaksanaan penelitian akan berlangsung selama ± 2 bulan.

C. Alat dan Bahan

Penelitian untuk perancangan aplikasi ini hanya dapat berjalan dengan tersedianya *hardware* dan *software* untuk mendukung penelitian. Berikut pejelasan mengenai *hardware* dan *software* pada table 3.1 dan 3.2 yang akan digunakan :

1. Hardware/Perangkat Keras

Hardware dalam perancangan aplikasi terlihat pada Tabel 3.1.

Tabel 3. 1 Spesifikasi *Hardware*

Spesifikasi	
Merk Laptop	HP
Processor Laptop	Intel Core i5
RAM	8GB

2. Software/Perangkat Lunak

Penggunaan *software* dalam perancangan aplikasi terlihat pada Tabel 3.2.

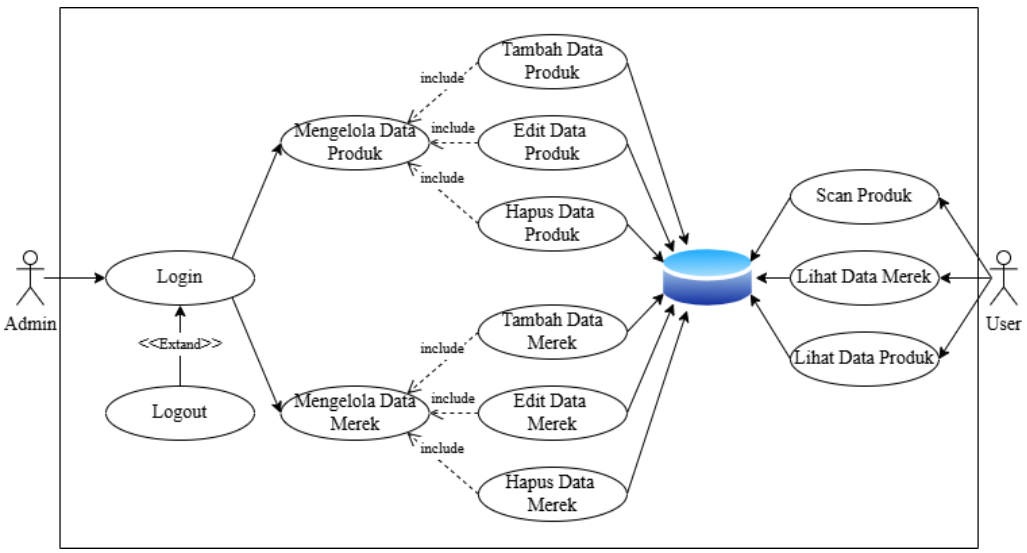
Tabel 3. 2 Spesifikasi *Software*

Spesifikasi	
Sistem Operasi	Windows 11
Software Basis Data	phpMyAdmin
Tool Pemrograman	Android Studio
Bahasa Pemrograman	Kotlin

D. Perancangan Sistem

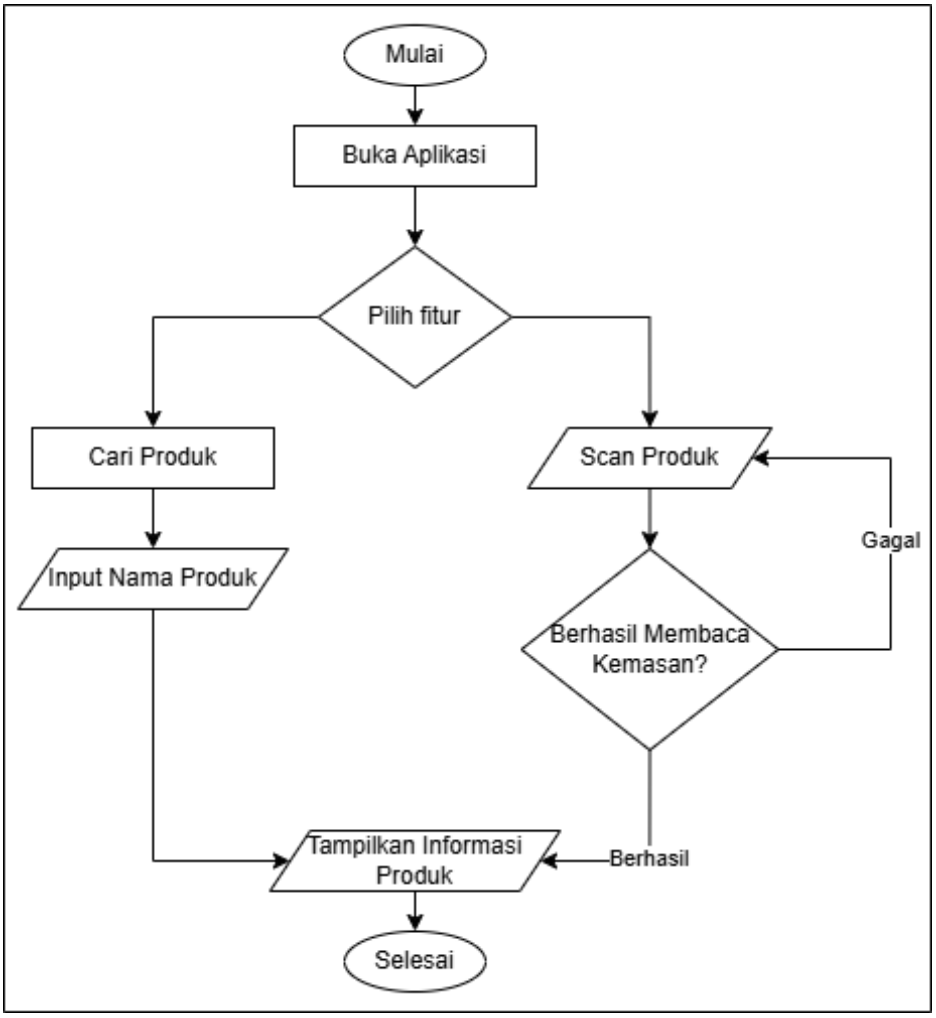
Desain sistem menggambarkan alur kerja dan fungsi utama aplikasi yang akan dibuat. Salah satu langkah pertama dalam proses ini adalah membuat diagram *use case* untuk memvisualisasikan interaksi antara aktor (pengguna) dan sistem. Diagram ini menggambarkan berbagai fungsi atau layanan yang dapat diakses pengguna, seperti login, memindai teks pada produk, dan melihat daftar produk yang diboikot. Selain itu, *flowchart* dibuat untuk memperjelas urutan logis setiap proses dalam sistem dari input hingga output. *Flowchart* menunjukkan urutan aktivitas langkah demi langkah, termasuk pengambilan keputusan, pemindaian teks, rekonsiliasi data dengan basis data, dan penyajian informasi kepada pengguna.

1) *Use Case Diagram*



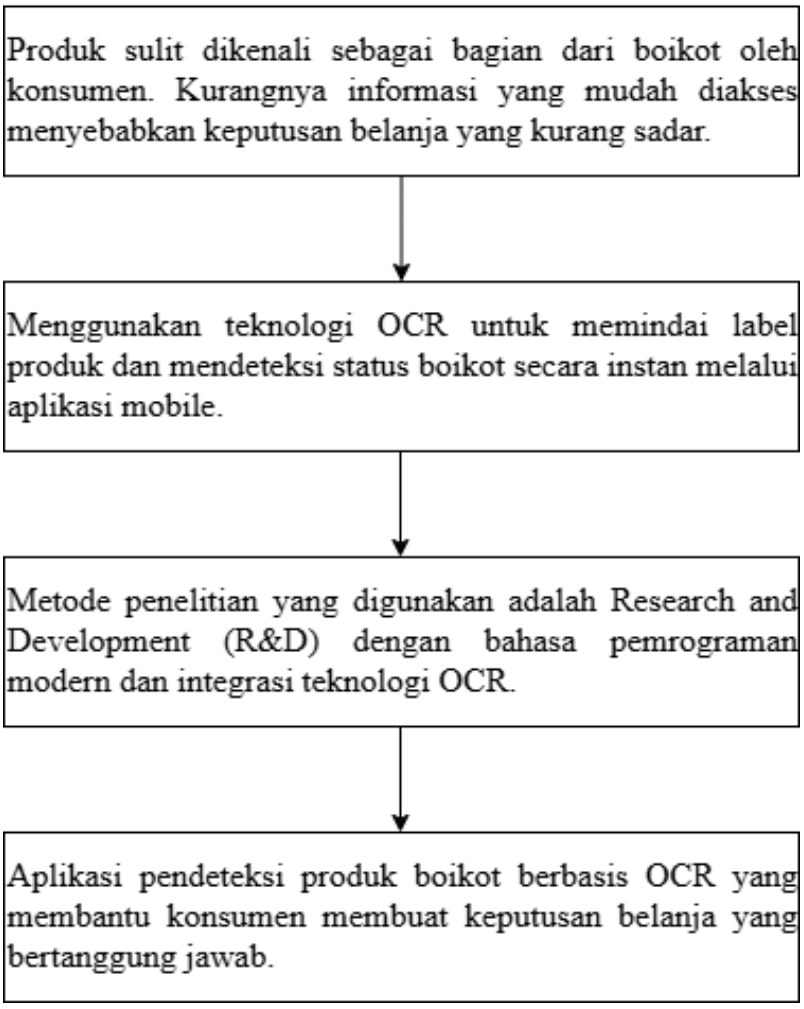
Gambar 3. 1 *Use Case* Diagram

2) *Flowchart* Aplikasi



Gambar 3. 2 *Flowchart* Aplikasi

E. Kerangka Berfikir



Gambar 3. 3 Kerangka Berfikir

F. Metode Pengumpulan Data

Tahapan ini menggunakan teknik pengambilan data melalui studi literatur yang bersifat mendalam dan terarah. Penulis mengkaji berbagai sumber tertulis seperti artikel ilmiah, laporan, serta publikasi yang berasal dari komunitas nasional maupun internasional yang secara aktif menyuarakan seruan pemboikotan terhadap produk-produk tertentu. Beberapa komunitas yang dijadikan acuan antara lain

Boycott, Divestment, Sanctions (BDS) Movement, *Consumers International*, *asar humanity, Aqsa Working Group*, *No Thanks App* dan berbagai forum media sosial yang memiliki perhatian terhadap isu kemanusiaan dan etika konsumsi.

Melalui kegiatan ini, peneliti memperoleh data sekunder yang relevan, mencakup daftar produk yang diboikot, latar belakang seruan boikot, serta dinamika dan perkembangan isu yang berkaitan. Teknik ini dipilih guna memperoleh pemahaman yang komprehensif sebagai dasar dalam perancangan sistem yang mampu mendeteksi produk berdasarkan daftar boikot yang berlaku.

G. Metode Pengujian

Pengujian sistem dilakukan dengan mengevaluasi kode program dan mengimplementasikan hasilnya melalui serangkaian pengujian kasus pada lingkungan penelitian yang telah disiapkan. Setelah struktur program tersusun secara sistematis, tahap pengujian unit menjadi sangat penting guna memastikan bahwa setiap modul atau bagian dari sistem berfungsi sebagaimana mestinya. Dalam pelaksanaannya, terdapat dua pendekatan utama yang diterapkan pada evaluasi unit, yaitu penggunaan uji *White Box* dan *Black Box.*

1. Black Box Testing

Pengujian *black box* berupa teknik pengujian suatu sistem engan menekankan pengujian fungsi sistem dari perspektif pengguna, tanpa melihat pada rincian kode yang mendasarinya. Teknik ini meliputi pemberian sejumlah input tertentu dan langkah berikutnya adalah melakukan verifikasi apakah output yang diperoleh telah selaras dengan kriteria atau standar yang telah ditetapkan sebelumnya (Sholeh et al., 2021). Penguji tidak perlu mengetahui bagaimana proses di dalam sistem

berlangsung, melainkan hanya menilai apakah sistem memenuhi spesifikasi yang telah ditentukan.

2. White Box Testing

Evaluasi pada *white box* menitikberatkan perhatian ke susunan dalam suatu program. Dalam pengujian ini, penguji harus memahami alur logika, struktur kontrol, dan kode sumber dari sistem yang diuji. Tujuan utamanya adalah memastikan bahwa setiap jalur logika dalam program telah diuji, serta untuk mendeteksi kesalahan tersembunyi di dalam kode. Metode ini sangat berguna untuk mengidentifikasi potensi kesalahan pada tingkat algoritma atau implementasi (Syaikhuddin et al., 2018).

BAB IV

HASIL DAN PEMBAHASAN

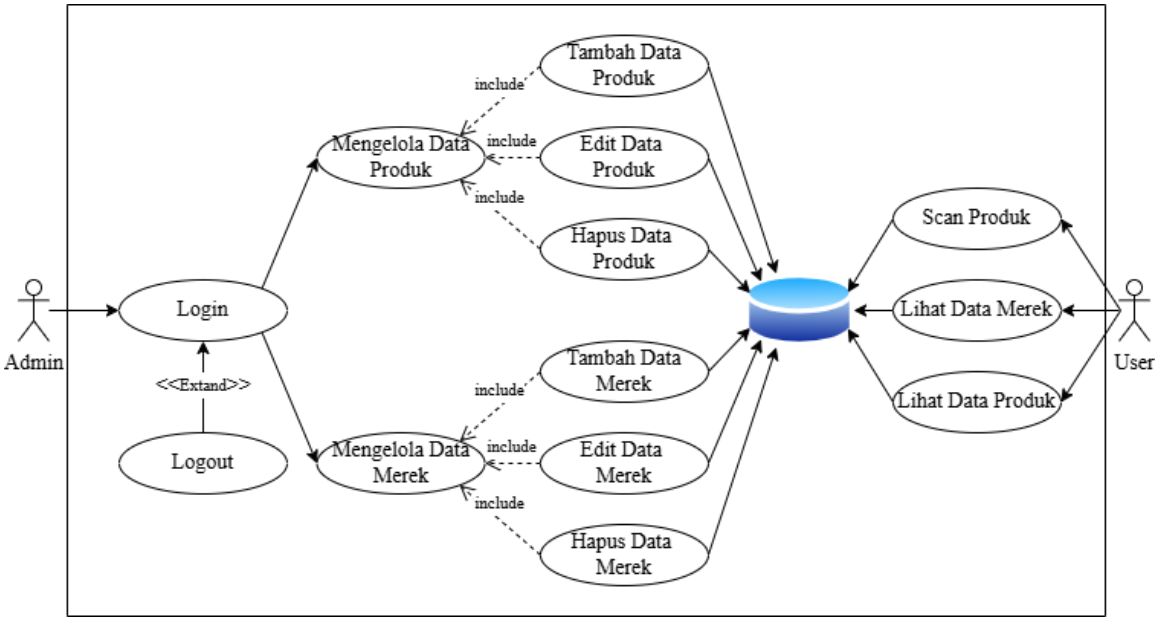
A. Analisis Aliran Data UML

Data *flow* UML merujuk pada proses perpindahan informasi atau data antar elemen sistem, seperti komponen, objek, atau aktivitas, meskipun istilah "aliran data" tidak secara eksplisit digunakan sebagaimana pada *Data Flow Diagram* (DFD). UML lebih menekankan pada pemodelan struktur dan perilaku sistem melalui berbagai jenis diagram, namun konsep aliran data tetap tergambarkan secara implisit melalui interaksi, hubungan, dan pesan antar elemen sistem.

Beberapa diagram UML yang merepresentasikan aliran data secara tidak langsung meliputi representasi visual seperti diagram urutan, diagram aktivitas, diagram kasus penggunaan, serta diagram komunikasi.. Dalam Diagram Sequence, aliran data terwujud melalui pesan yang dikirim antara objek seiring waktu, menunjukkan urutan interaksi antar komponen. Activity Diagram menggambarkan aliran kontrol dan data dalam bentuk aktivitas dan object flow, yang mengilustrasikan bagaimana data berpindah di berbagai aktivitas.

1. Use Case Diagram

Diagram *Use Case* merupakan tipe diagram dalam UML berguna untuk menggambarkan relasi antar aktor dan sistem berdasarkan peran atau fasilitan (kasus penggunaan) tawaran sistem.



Gambar 4. 1 *Use Case Diagram*

Tabel 4. 1 *Use Case* Skenario Admin

Nama Use Case	Deskripsi
Login	Untuk melakukan manajemen data produk dan merek, admin harus melakukan login terlebih dahulu.
Kelola Data Merek	Ini adalah use case yang mencakup pengelolaan data merek. Di dalamnya terdapat tiga use case tambahan yang di-include: • Tambah Data Merek Admin dapat menambahkan entri merek baru ke dalam database. • Ubah Data Merek Admin dapat mengedit data merek yang sudah ada. • Hapus Data Merek

Nama Use Case	Deskripsi
	Admin dapat menghapus data merek yang tidak diperlukan lagi. Semua aksi ini berinteraksi dengan Database, yang artinya: • Setiap perubahan akan disimpan permanen • Sistem harus melakukan validasi dan update data
Kelola Data Produk	Ini adalah use case yang mencakup pengelolaan data produk. Di dalamnya terdapat tiga use case tambahan yang di-include: • Tambah Data Produk Admin menambahkan entri produk baru ke dalam database. • Ubah Data Produk Admin mengubah informasi dari produk. • Hapus Data Produk Admin menghapus produk yang sudah tidak relevan atau salah input. Semua aksi ini berinteraksi dengan Database, yang artinya: • Setiap perubahan akan disimpan permanen • Sistem harus melakukan validasi dan update data

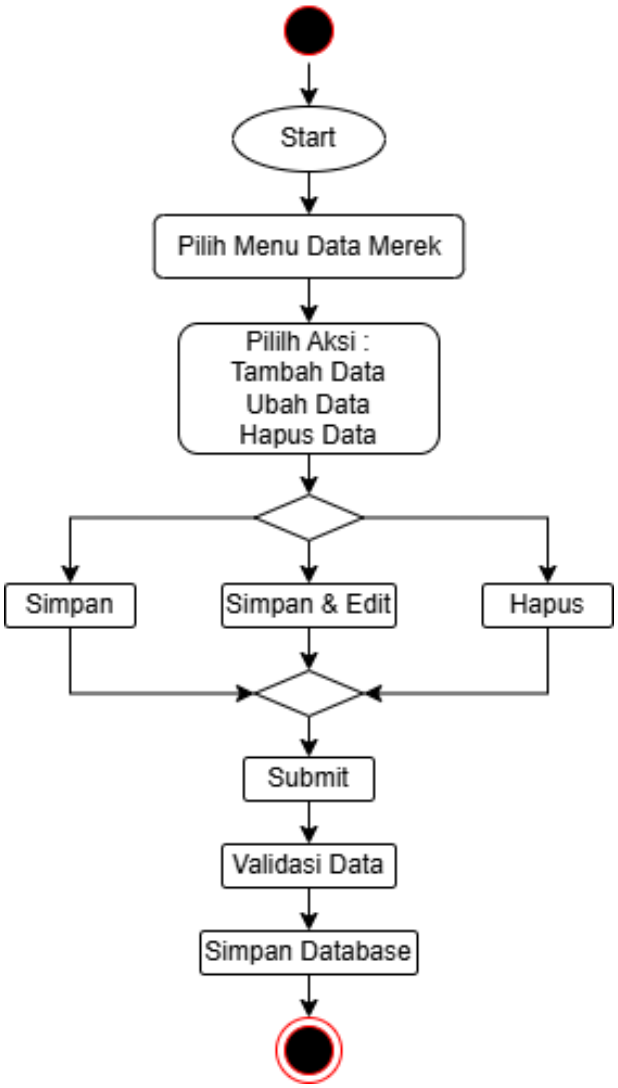
Tabel 4. 2 *Use Case* Skenario User

Nama Use Case	Deskripsi
Scan Produk	User melakukan pemindaian produk menggunakan OCR dan mendapatkan informasi terkait.
Lihat Daftar Merek	User mengamati daftar merek yang tersedia dalam sistem.
Lihat Daftar Produk	User melihat daftar produk yang tersedia.

2. Activity Diagram

Diagram aktivitas berupa tampilan grafis dalam UML yang digunakan dalam penggambaran rangkaian langkah maupun tahapan sebuah sistem atau bisnis. Diagram ini mengilustrasikan keterkaitan antara aktivitas dan Proses perpindahan kendali antar aktivitas selama jalannya suatu rangkaian kegiatan.

a. *Activity Diagram Admin*

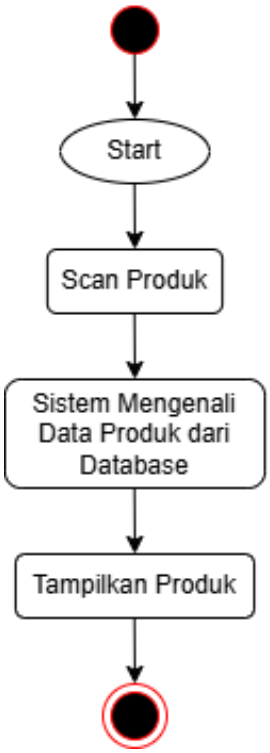


Gambar 4. 2 *Activity Diagram Admin*

Diagram aktivitas pada Gambar 4.2, menunjukkan alur pengelolaan data merek oleh Admin dalam aplikasi.Proses dimulai dari titik awal (start), dilanjutkan dengan Admin memilih menu “Data Merek”. Setelah itu, Admin diberikan pilihan untuk melakukan: Tambah, Ubah Data, maupun Hapus Data. Jika Admin memilih “Tambah Data” atau “Ubah Data”, maka sistem memberikan form, lalu Admin dapat mengisi dan men-submit data. Sistem kemudian memvalidasi data yang

dikirimkan. Jika data tersebut valid, maka akan tersimpan ke dalam database. Jika Admin memilih “Hapus Data”, maka data langsung dihapus tanpa melalui proses submit dan validasi. Proses diakhiri ketika sistem selesai menyimpan (atau menghapus) data ke *database*.

b. *Activity Diagram User*



Gambar 4. 3 *Activity Diagram User*

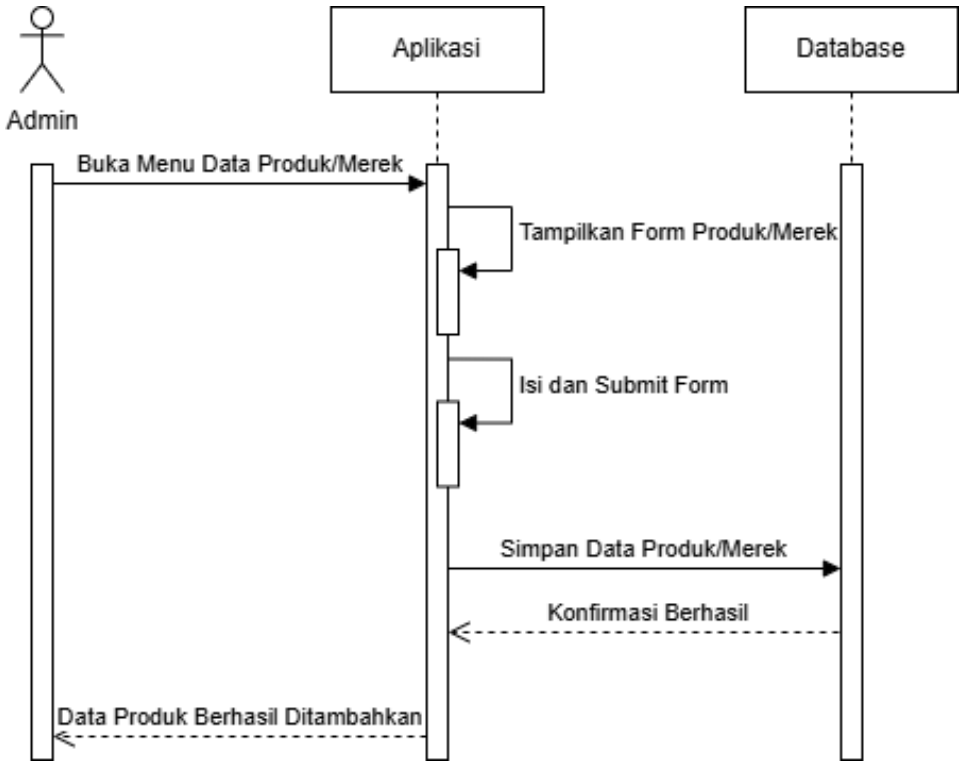
Diagram aktivitas yang tertera pada Gambar 4.3, menunjukkan alur proses pemindaian produk oleh User dalam aplikasi.Proses dimulai dari titik awal (start), dilanjutkan dengan User melakukan aksi Scan Produk menggunakan kamera melalui aplikasi. Setelah gambar atau data produk di-scan, sistem secara otomatis akan mengenali data produk yang sesuai dari database menggunakan pencocokan data (menggunakan OCR). Setelah produk berhasil diidentifikasi, sistem akan

menampilkan informasi produk kepada User, seperti nama, status boikot, atau detail lainnya. Proses kemudian berakhir di titik akhir (end).

3. Sequence Diagram

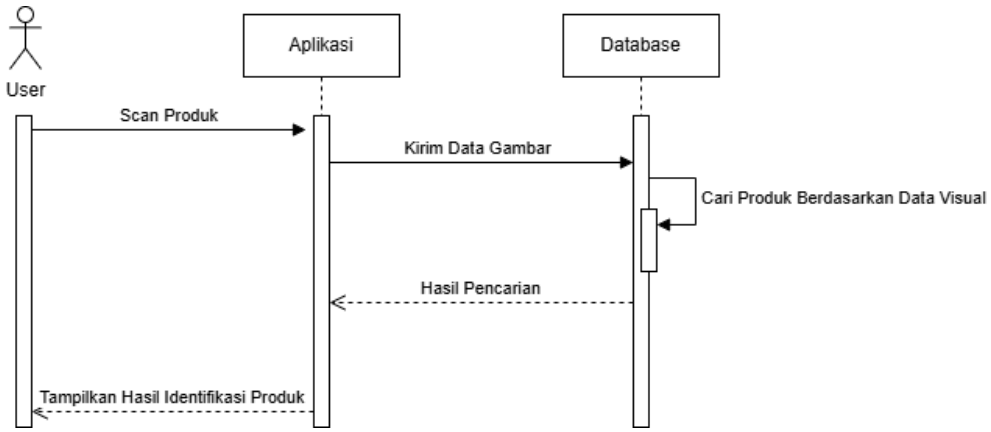
Sequence Diagram merupakan penggambaran UML yang memperlihatkan interaksi antara objek-objek dalam sistem yang disusun berdasarkan kronologi kejadian.. Diagram ini menggambarkan cara objek saling berinteraksi dengan mengirimkan pesan secara berurutan untuk menjalankan suatu skenario atau menyelesaikan pekerjaan dalam sistem.

a. *Sequence Diagram Admin*



Gambar 4. 4 *Sequence Diagram Admin*

b. *Sequence Diagram User*



Gambar 4. 5 *Sequence Diagram User*

B. Desain Sistem

1. *User*

a. Tampilan *Splash (Splash Screen)*

Antarmuka di awal atau *splash screen* berupa kemunculan sejenak di pembukaan aplikasi mobile atau perangkat lunak sebelum tampilan utama ditampilkan. Tampilan ini seringkali menampilkan logo aplikasi, nama perusahaan, atau elemen visual lain yang memberikan kesan pertama kepada pengguna.



Gambar 4. 6 Tampilan *Splash*

b. Tampilan Layar Beranda

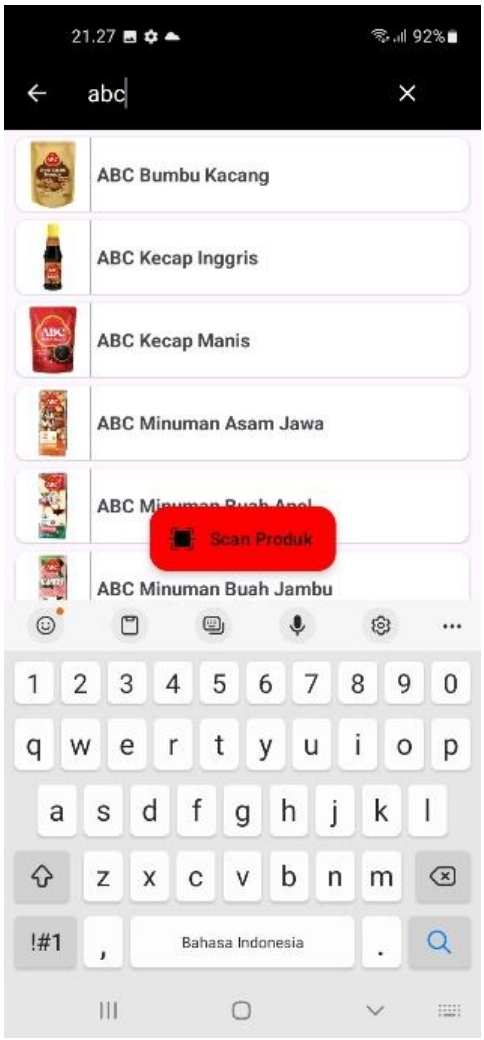
Tampilan layar utama aplikasi, yang sering juga disebut "home screen" atau "layar beranda", adalah titik awal yang digunakan pengguna untuk mengakses berbagai fitur dan aplikasi di perangkat mereka. Layar ini biasanya menampilkan ikon aplikasi, widget, dan informasi penting lainnya, seperti waktu, notifikasi, dan status baterai.



Gambar 4. 7 Tampilan Layar Beranda

c. Tampilan Halaman Pencarian

Halaman pencarian sangat membantu untuk menemukan produk yang ingin diketahui status boikotnya secara manual.



Gambar 4. 8 Tampilan Halaman Pencarian

d. Fitur *Scan*

Fitur scan pada aplikasi deteksi boikot, berfungsi untuk memindai nama produk untuk memeriksa apakah produk tersebut masuk dalam daftar boikot. Aplikasi ini akan memberikan informasi tentang asal produk, merek, dan apakah produk tersebut mendukung atau terafiliasi dengan pihak yang diboikot.



Gambar 4. 9 Perintah Mengarahkan Kamera



Gambar 4. 10 Deteksi Kemasan Produk

e. Tampilan Informasi

Setelah proses scan berhasil, maka akan muncul informasi mengenai status produk.

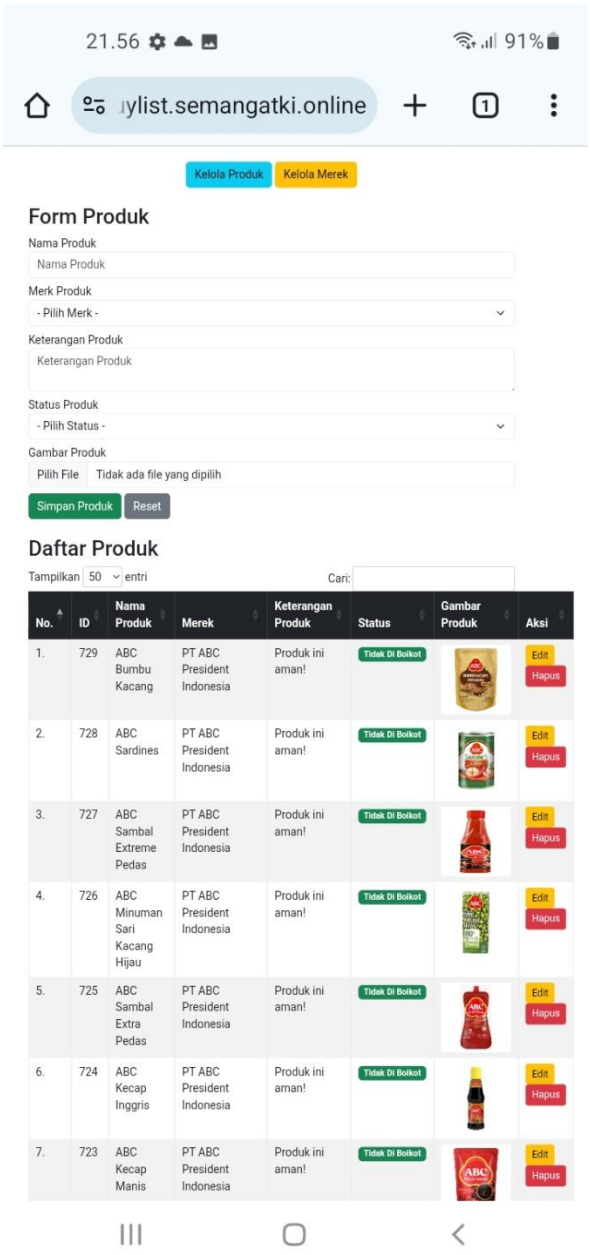


Gambar 4. 11 Tampilan Informasi

2. *Admin*

a. Tampilan Halaman Kelola Produk

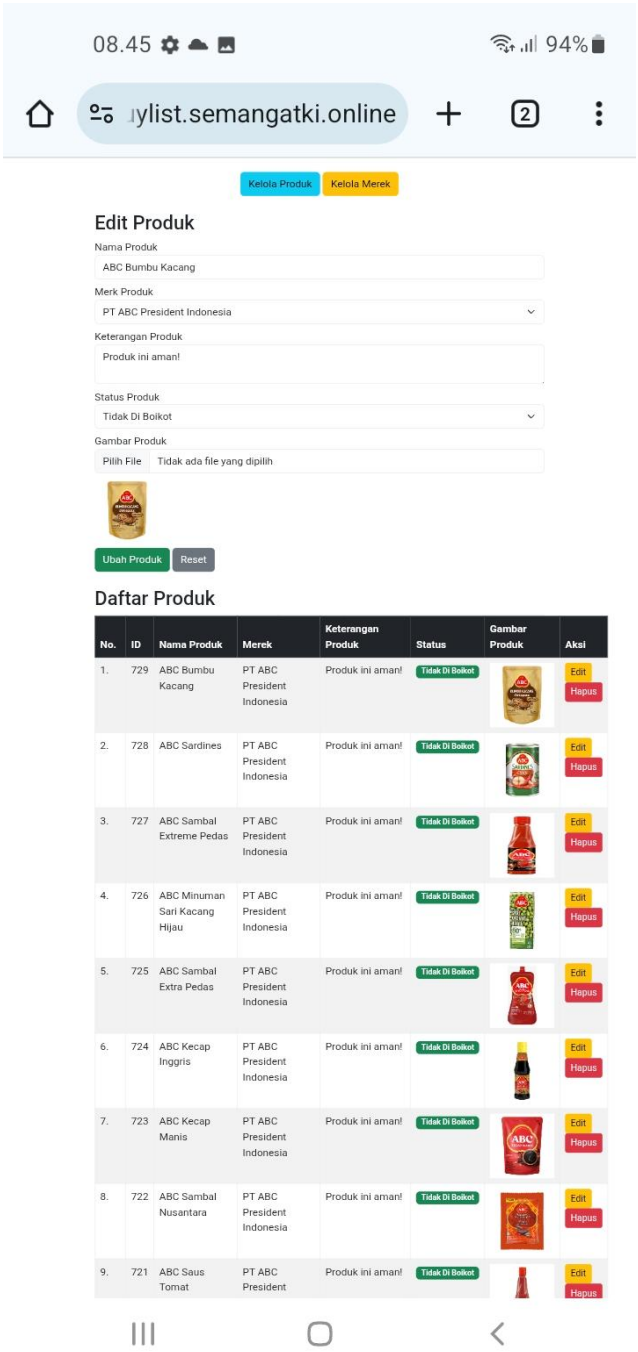
Halaman ini berguna untuk admin melakukan manajemen data produk yang tersedia pada sistem.



Gambar 4. 12 Halaman Kelola Produk

b. Tampilan Edit Produk

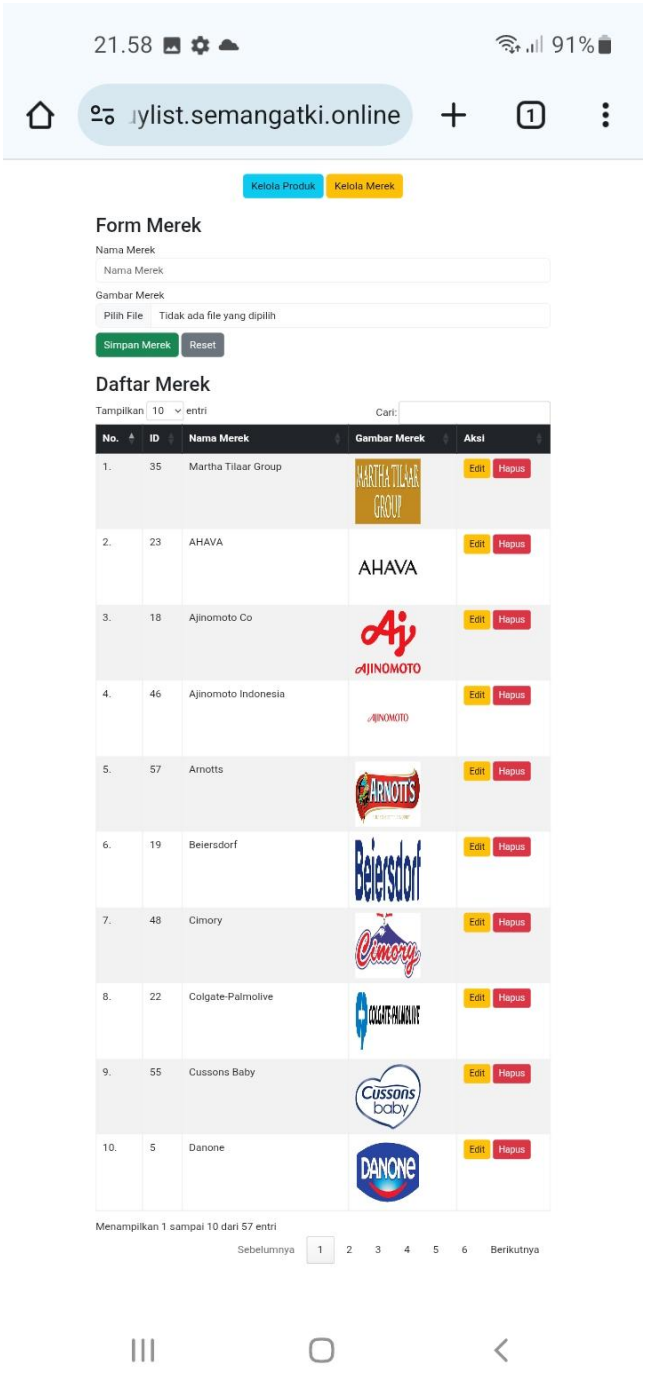
Halaman edit produk digunakan untuk mengubah atau memperbarui data yang sudah ada dalam sistem.



Gambar 4. 13 Tampilan Edit Produk

c. Tampilan Halaman Kelola Merek

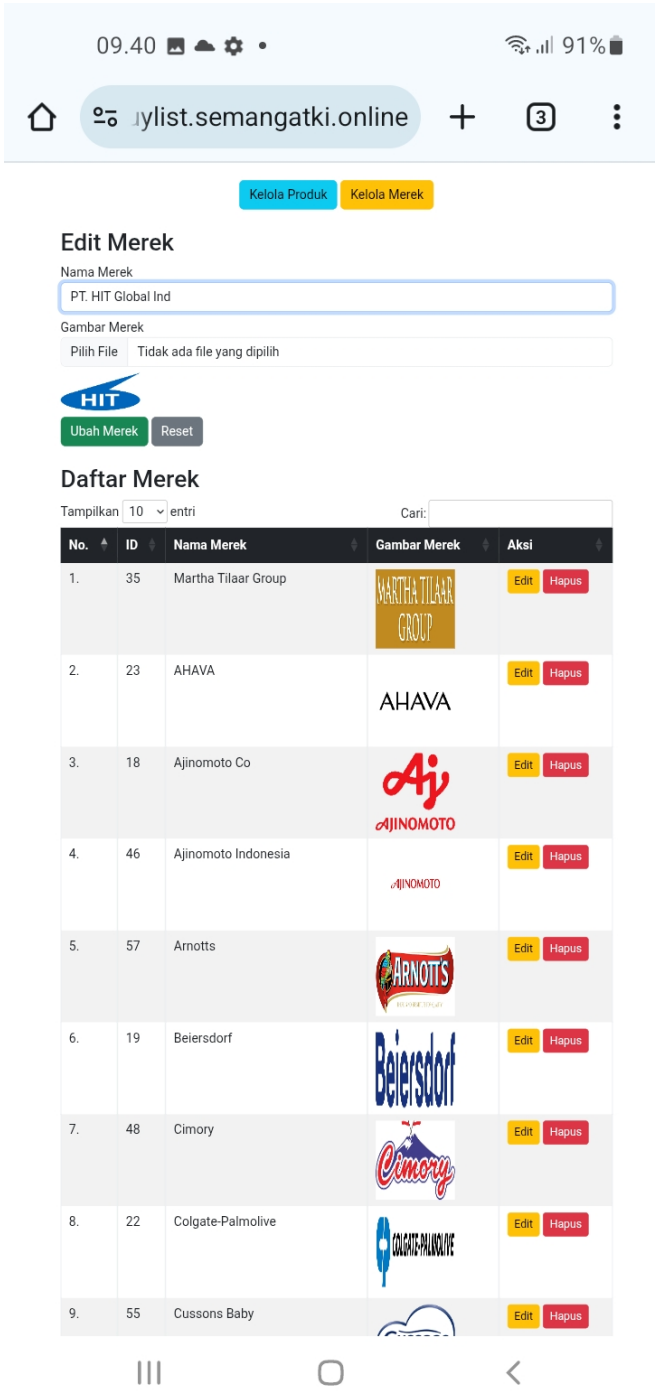
Pada halaman ini, admin dapat mengelola informasi merek yang ada dalam sistem.



Gambar 4. 14 Halaman Kelola Merek

d. Tampilan Edit Merek

Halaman ini berguna bagi administrator dalam mengelola informasi merek yang tersimpan dalam sistem.



Gambar 4. 15 Tampilan Edit Merek

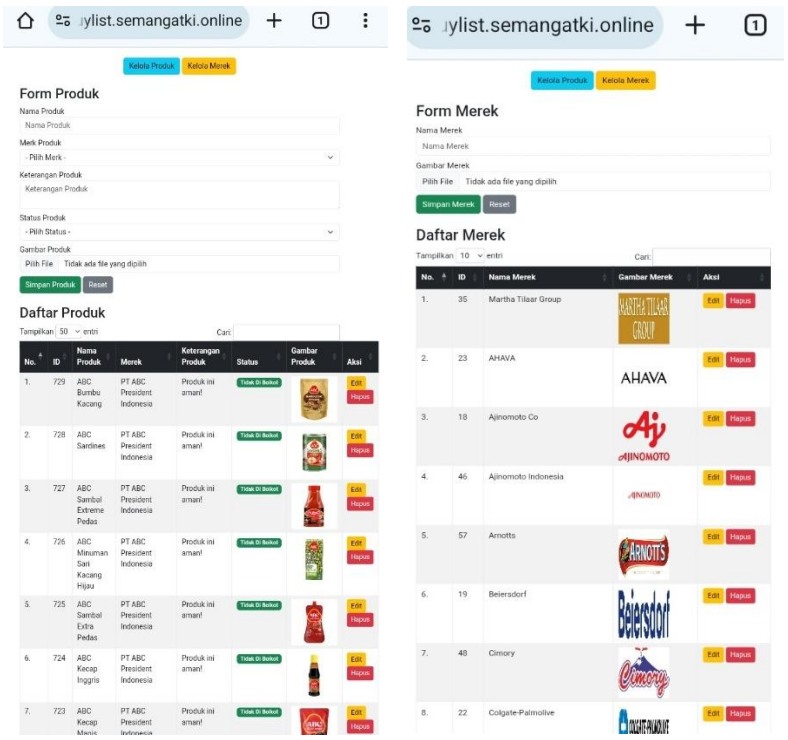
C. Pengujian Sistem

1. Blackbox Testing

Evaluasi *Blackbox* berupa Teknik pengujian aplikasi dengan mengevaluasi performa sebuah aplikasi dari tampak luar, tanpa melakukan analisis terhadap kode sumber atau sistem internalnya.

a. Pengujian *Blackbox* pada halaman kelola data produk (Admin)

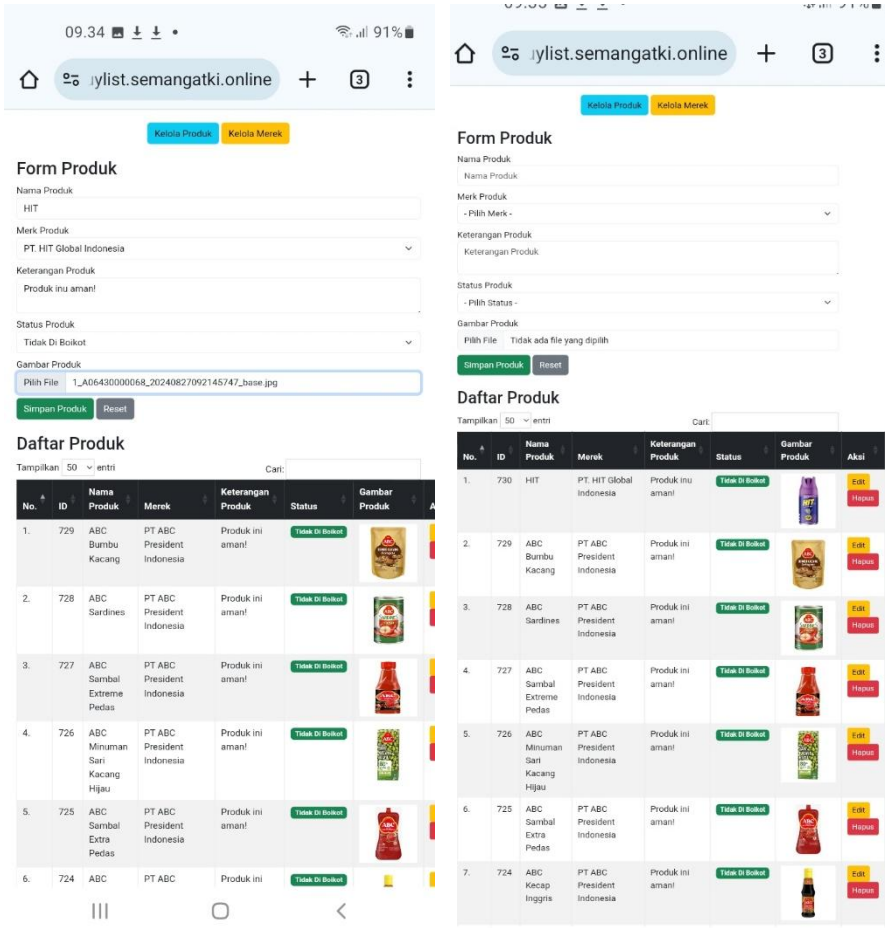
Tabel 4. 3 *Blackbox testing* halaman kelola data produk dan merek

Test Faktor	Hasil	Keterangan
Masuk ke halaman Manajemen produk dan merek	✔	Sistem menunjukkan halaman manajemen produk dan merek
Screenshot		
		

b. *Blackbox testing* tambah data produk (Admin)

Tabel 4. 4 *Blackbox testing* tambah data produk

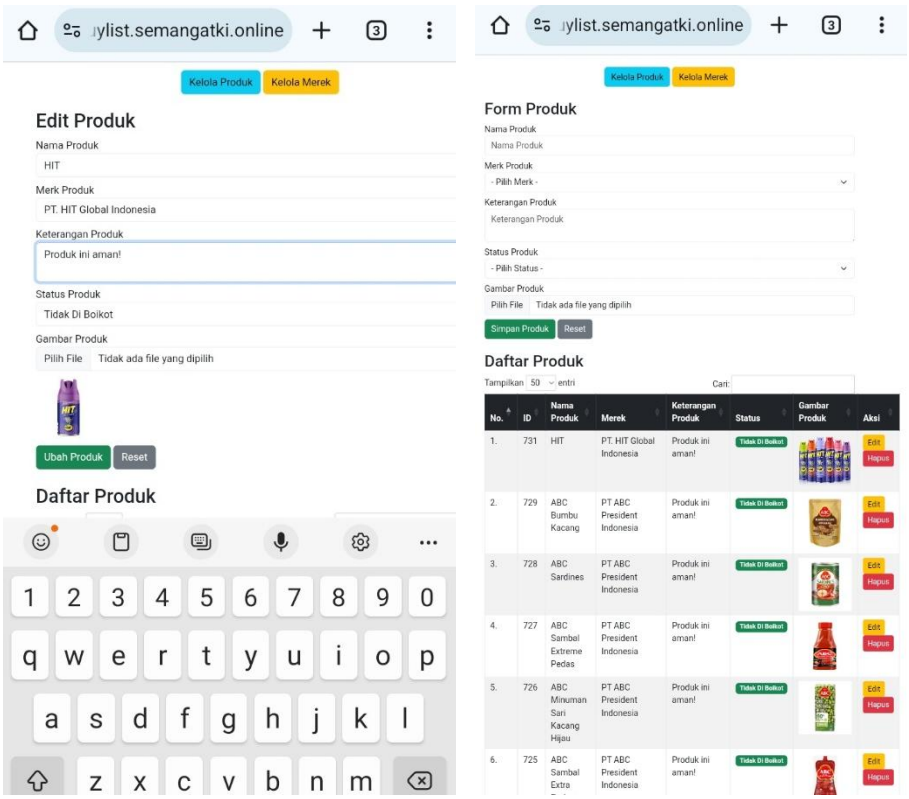
Test Faktor	Hasil	Keterangan
Mengisi form produk yang ingin di input	✔	Sistem akan menyimpan produk yang baru diinput
Screenshot		



c. *Blackbox testing* edit data produk (Admin)

Tabel 4. 5 Edit data produk

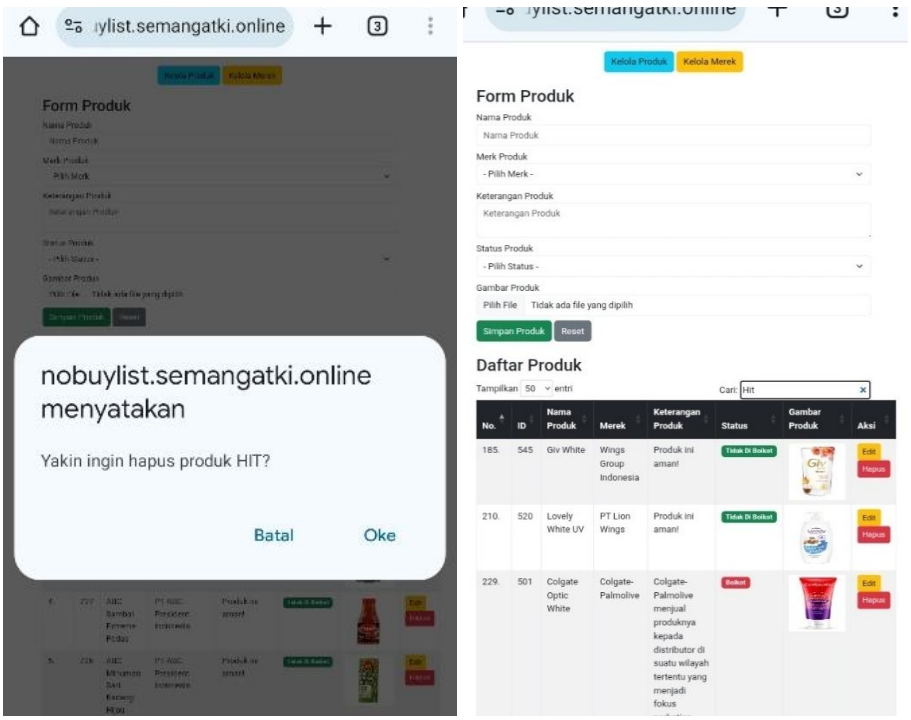
Test Faktor	Hasil	Keterangan
Mengubah gambar produk	✔	Sistem akan menyimpan produk yang telah di edit
Screenshot		



d. *Blackbox testing* hapus data produk (Admin)

Tabel 4. 6 Hapus data produk

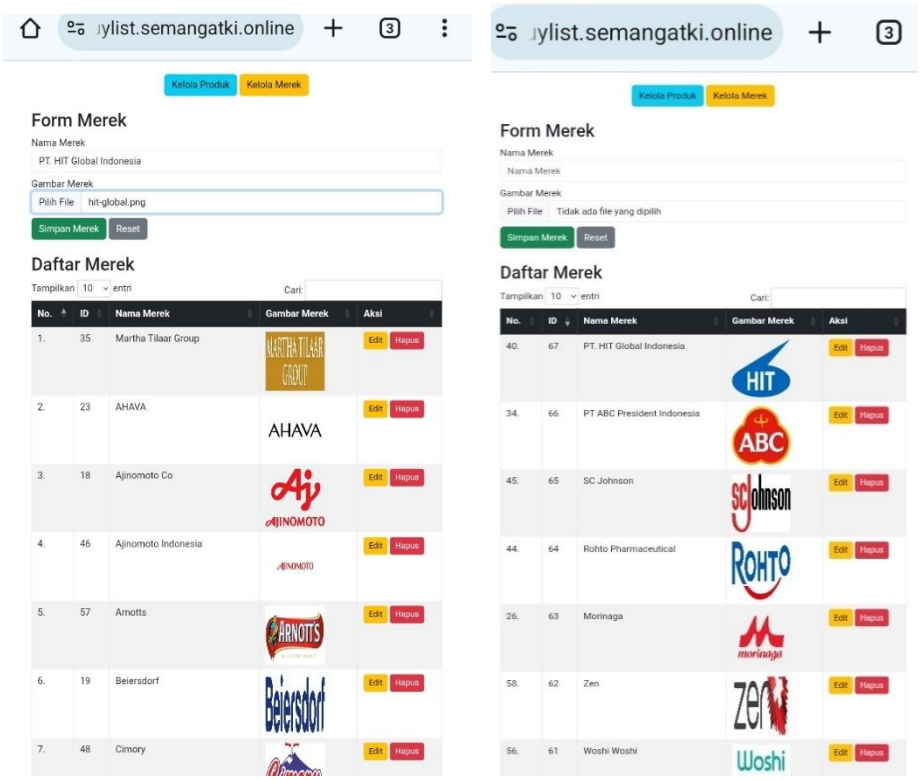
Test Faktor	Hasil	Keterangan
Menghapus entri produk	✔	Sistem akan menghapus entri produk yang dipilih oleh admin
Screenshot		



e. *Blackbox testing* tambah data merek (Admin)

Tabel 4. 7 Tambah Data Merek

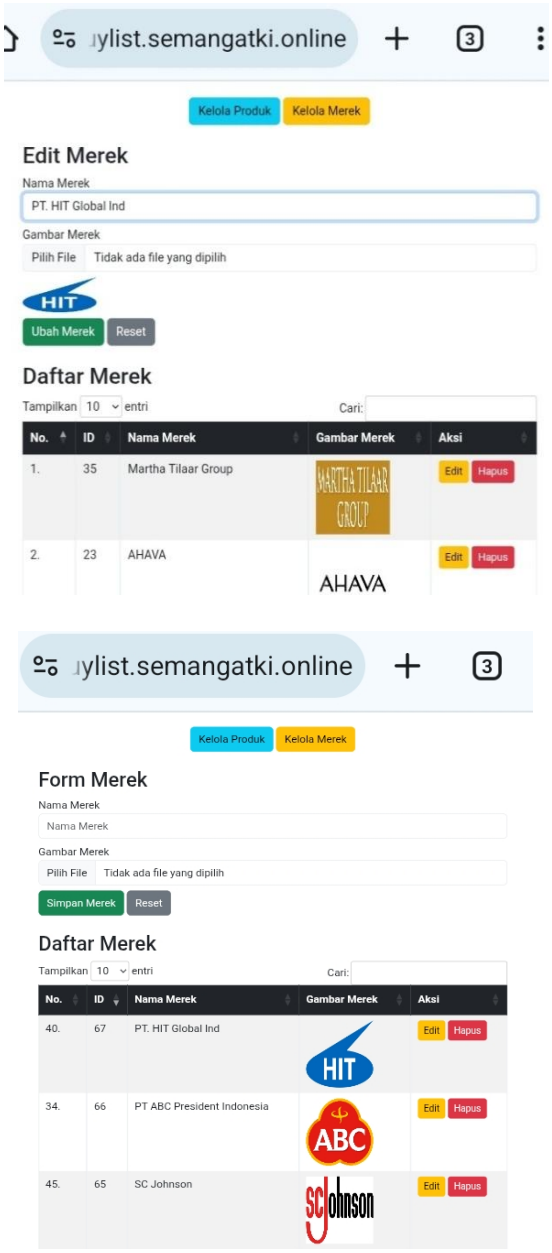
Test Faktor	Hasil	Keterangan
Mengisi form merek yang ingin di input	✔	Sistem akan menyimpan merek yang baru diinput
Screenshot		



f. *Blackbox testing* edit data merek (Admin)

Tabel 4. 8 Edit Data Merek

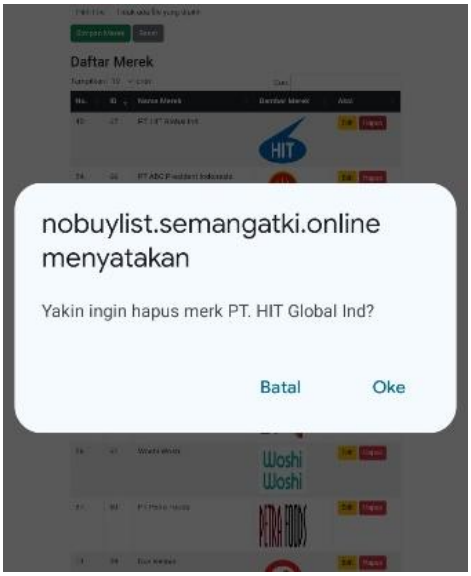
Test Faktor	Hasil	Keterangan
Mengubah nama merek	✔	Sistem akan menyimpan nama merek yang baru edit
Screenshot		

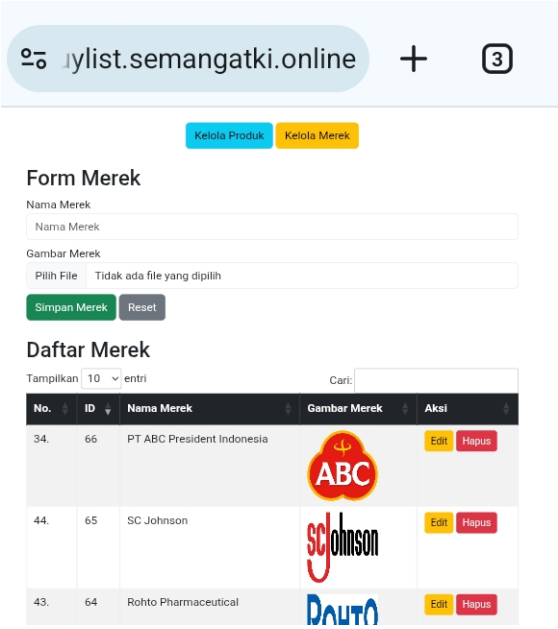


g. *Blackbox testing* hapus data merek (Admin)

Tabel 4. 9 Hapus data merek

Test Faktor	Hasil	Keterangan
Menghapus data merek	✔	Sistem akan menghapus data merek yang dipilih
Screenshot		

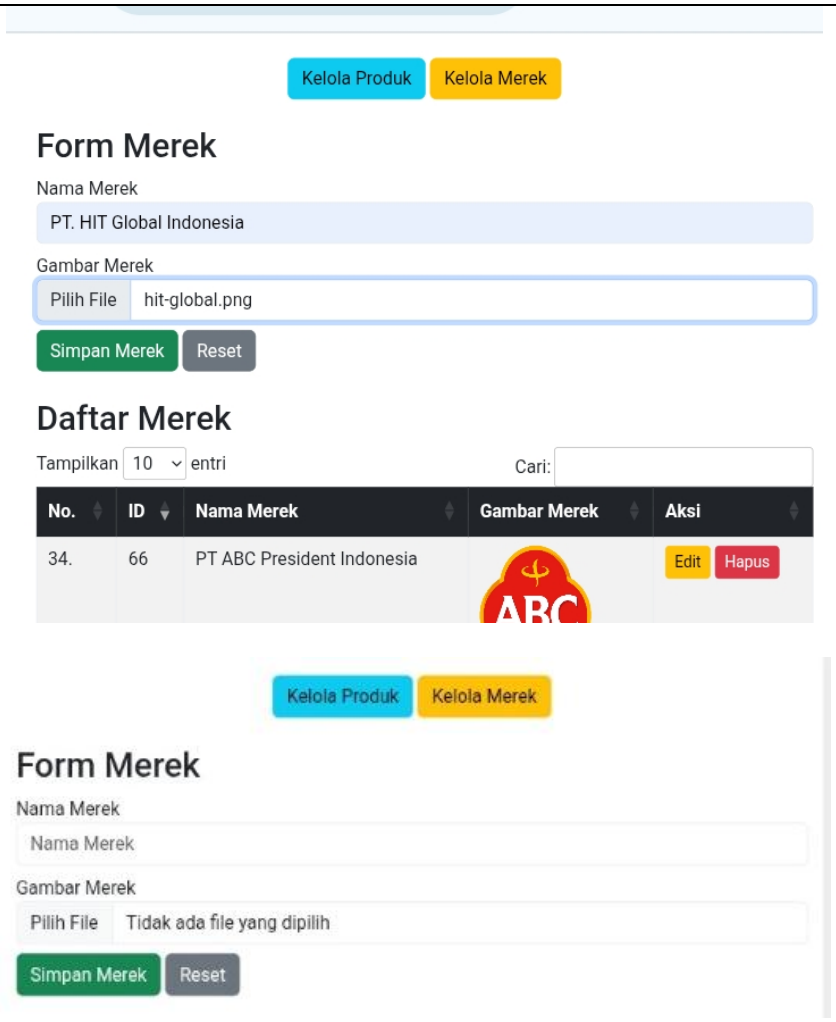




h. *Blackbox testing* reset data produk (Admin)

Tabel 4. 10 Reset Data Produk

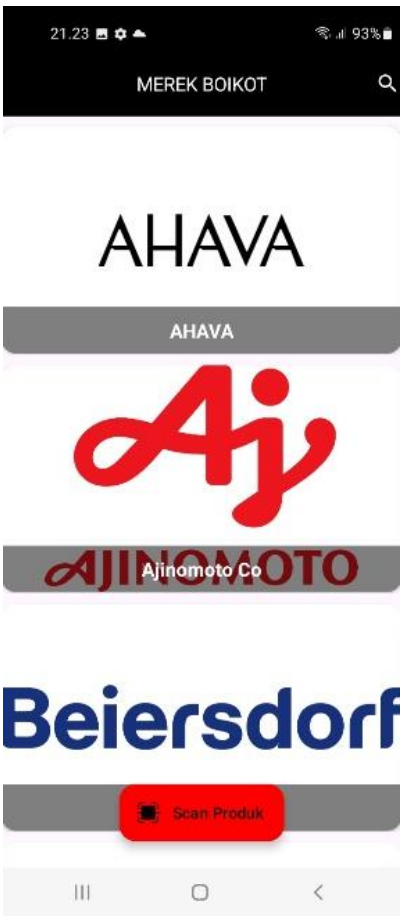
Test Faktor	Hasil	Keterangan
Mereset form	✔	Sistem akan menghapus semua isian kolom pada form
Screenshot		



i. *Blackbox testing* tampilan halaman utama (*User*)

Tabel 4. 11 Tampilan Halaman Utama/Beranda *User*

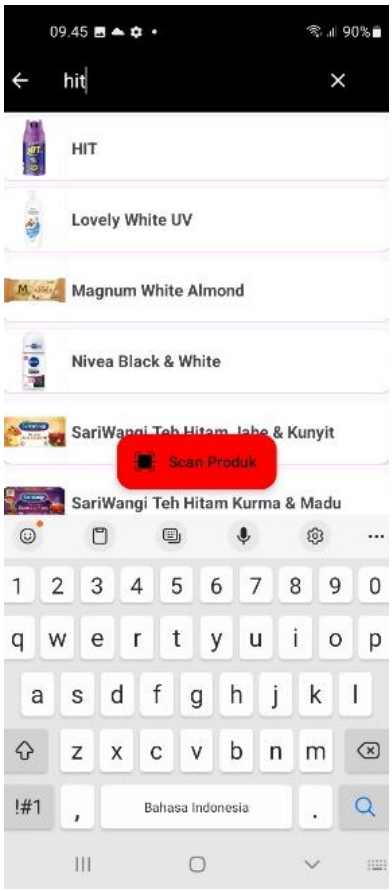
Test Faktor	Hasil	Keterangan
Masuk ke halaman utama	✔	Sistem akan menampilkan halaman utama
Screenshot		



j. *Blackbox testing* pencarian produk secara manual (*User*)

Tabel 4. 12 Tampilan Halaman Pencarian Manual

Test Faktor	Hasil	Keterangan
Pencarian produk	✔	Sistem akan menampilkan produk yang dicari
Screenshot		



k. *Blackbox testing* halaman daftar produk (*User*)

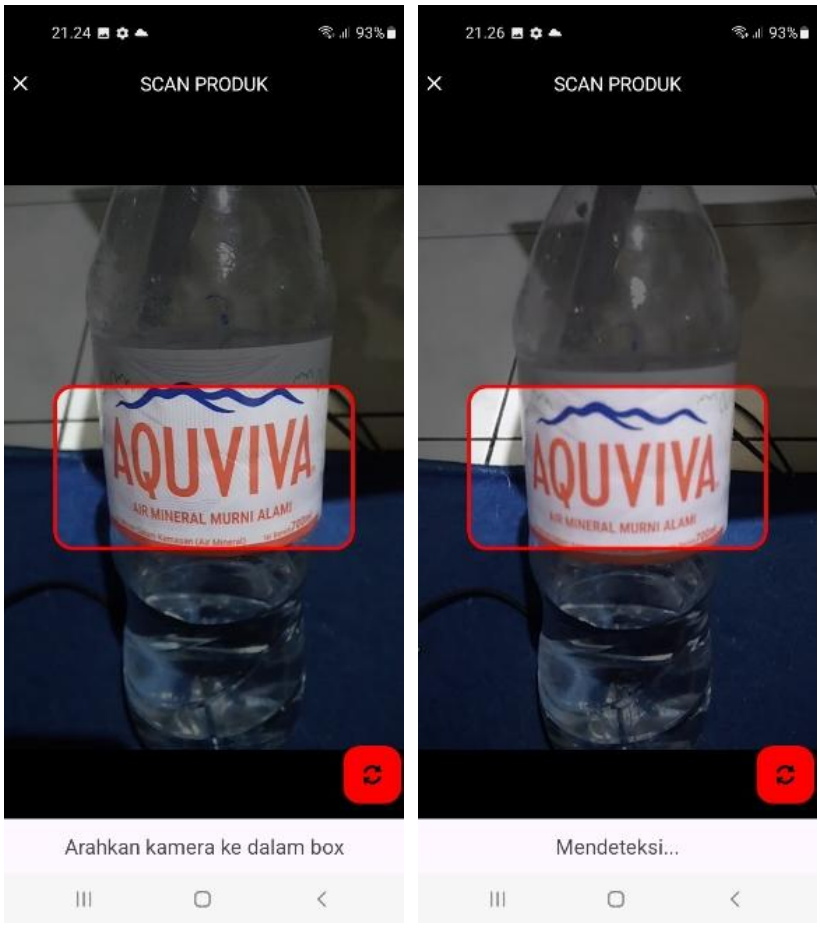
Tabel 4. 13 Tampilan Daftar Produk dalam Satu Merek

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



1. *Blackbox testing* scan produk (*User*)

Tabel 4. 14 Tampilan Scan Produk

Test Faktor	Hasil	Keterangan
Melakukan pemindaian pada kemasan produk	✔	Sistem akan memberi arahan posisi untuk melakukan scan dan mendeteksi produk (mencocokkan produk yang di scan dengan database)
Screenshot		
		

m. *Blackbox testing* menampilkan informasi status produk (*User*)

Tabel 4. 15 Tampilan Informasi Status Produk

Test Faktor	Hasil	Keterangan
Setelah pendeteksian berhasil maka muncul informasi status boikot	✔	Sistem akan menampilkan informasi status produk
Screenshot		



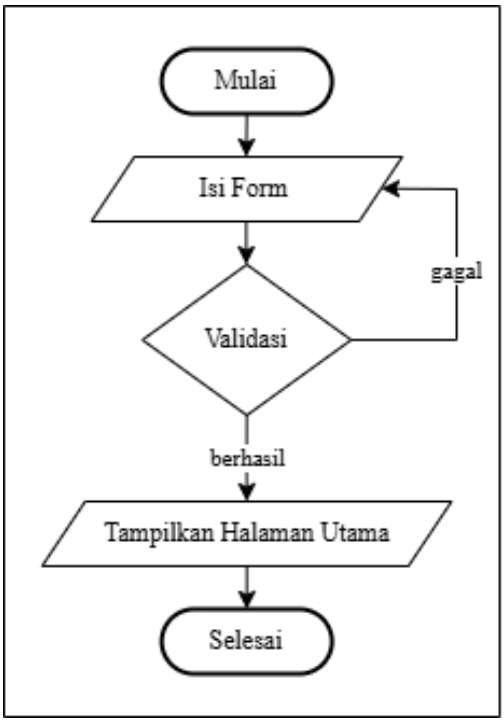


2. Whitebox Testing

Pengujian *Whitebox* berupa teknik evaluasi pada aplikasi untuk memahami struktur agian dalam sistem atau instruksi pemrograman yang mendasarinya. pada aplikasi yang dievaluasi. Tujuannya adalah untuk menguji logika internal, jalur kode, dan struktur kontrol program.

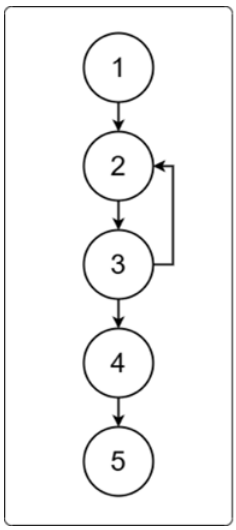
a) *Whitebox Testing* Halaman Login Admin

1) Flowchart



Gambar 4. 16 *Flowchart* Halaman Login Admin

2) Flowgraph



Gambar 4. 17 Flowgraph Halaman Login Admin

Berdasarkan flowgraph pada gambar 4. 17, 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

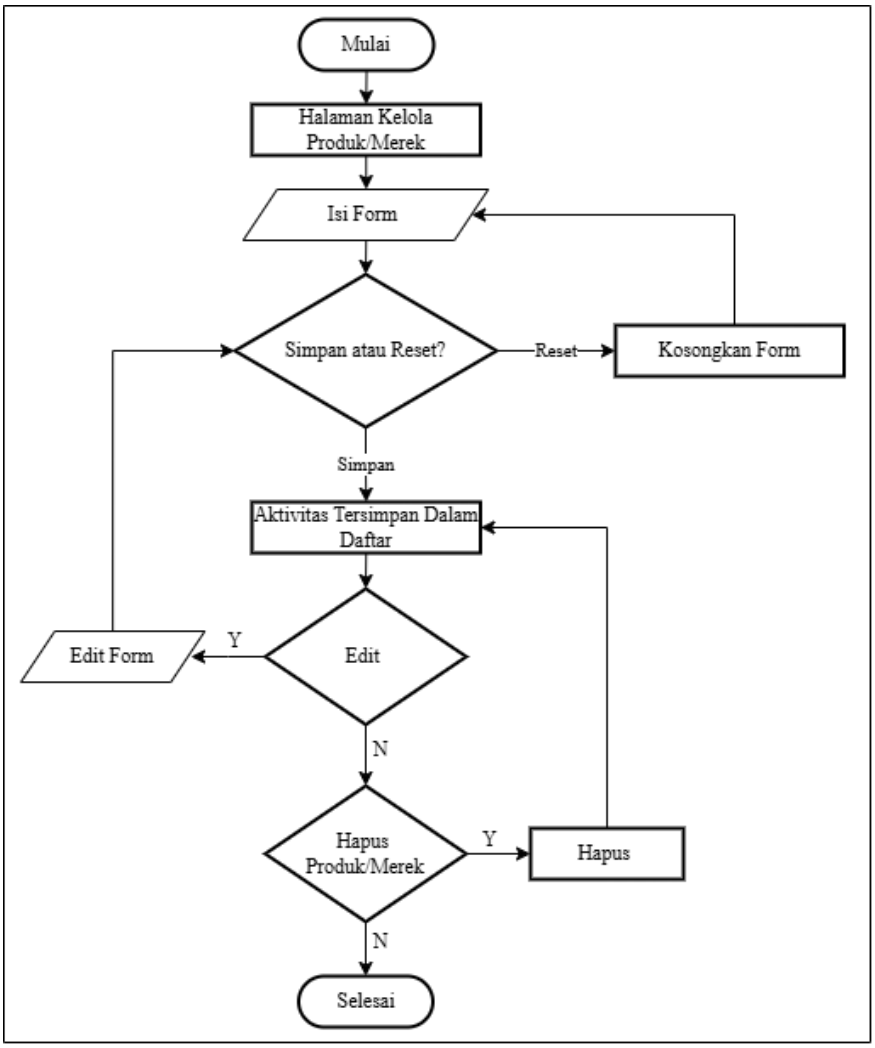
d. Grafik Matriks

Tabel 4. 16 Grafik Matriks Login Admin

	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

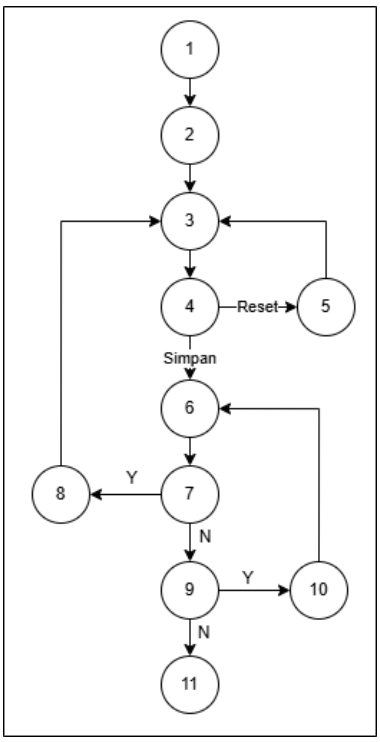
b) *Whitebox Testing* Halaman Kelola Produk/Merek Admin

1) Flowchart



Gambar 4. 18 *Flowchart* Kelola Produk/Merek (*Admin*)

2) Flowgraph



Gambar 4. 19 *Flowgraph* Kelola Produk/Merek (*Admin*)

Berdasarkan flowgraph pada gambar 4. 19, maka dilakukan perhitungan sebagai berikut:

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

V(G) = E – N + 2

N (*node*) = 11

E (*edge*) = 13

P (*predikat node*) = 3

Penyelesaian :

V(G) = E – N + 2

= 13 – 11 + 2

= 4

Predikat (P) = P + 1

= 3 + 1

= 4

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 4

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

Path 1 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 11

Path 2 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 3 – 4 – 5 – 6 – 7 – 9 – 11

Path 3 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 6 – 7 – 9 – 11

Path 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 11

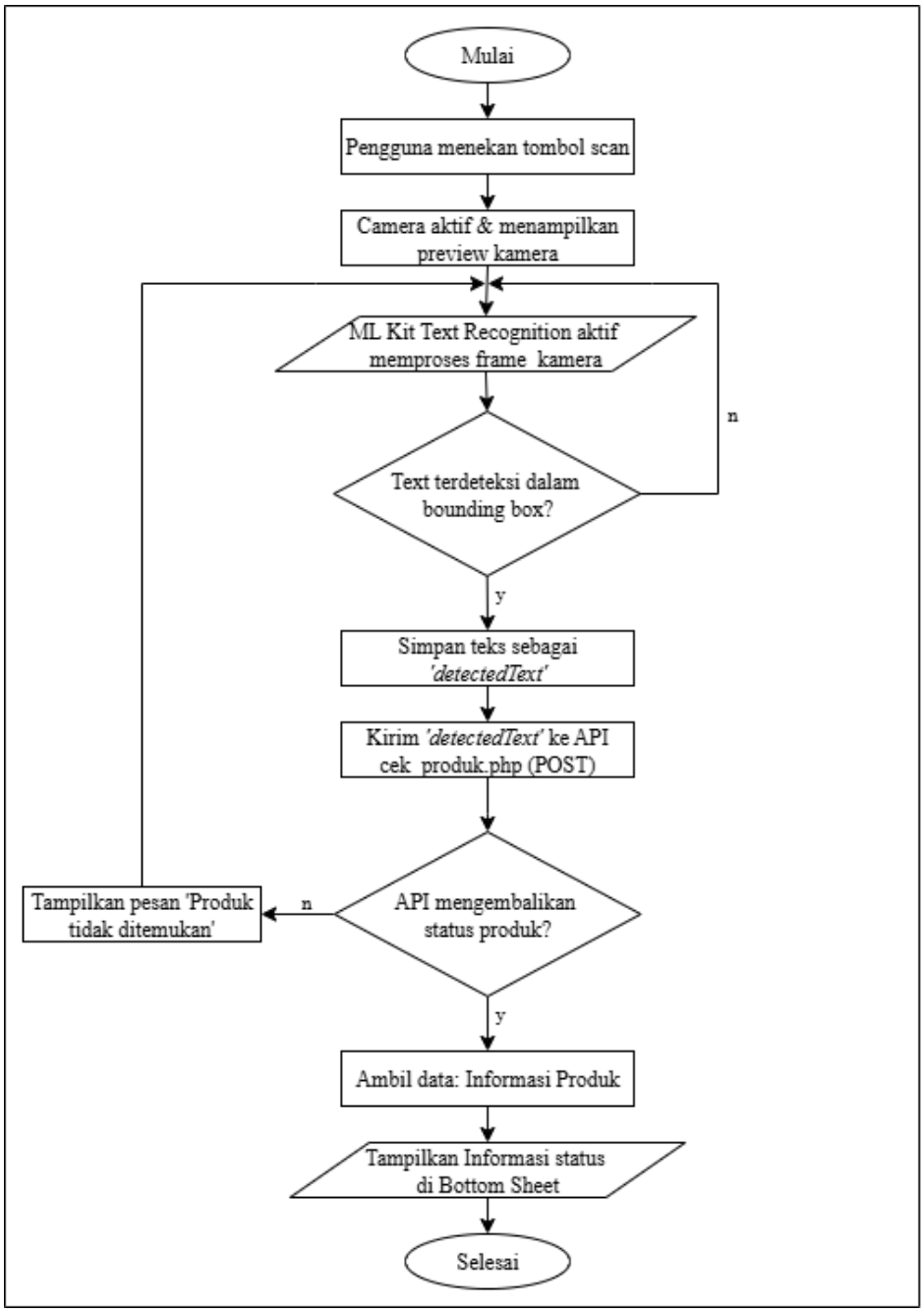
d. Grafik Matriks

Tabel 4. 17 Grafik Matriks Kelola Produk/Merek (*Admin*)

	1	2	3	4	5	6	7	8	9	10	11	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
6							1					1 – 1 = 0
7								1	1			2 – 1 = 1
8			1									1 – 1 = 0
9										1	1	2 – 1 = 1
10						1						1 – 1 = 0
Sum (E + 1)												3 + 1 = 4

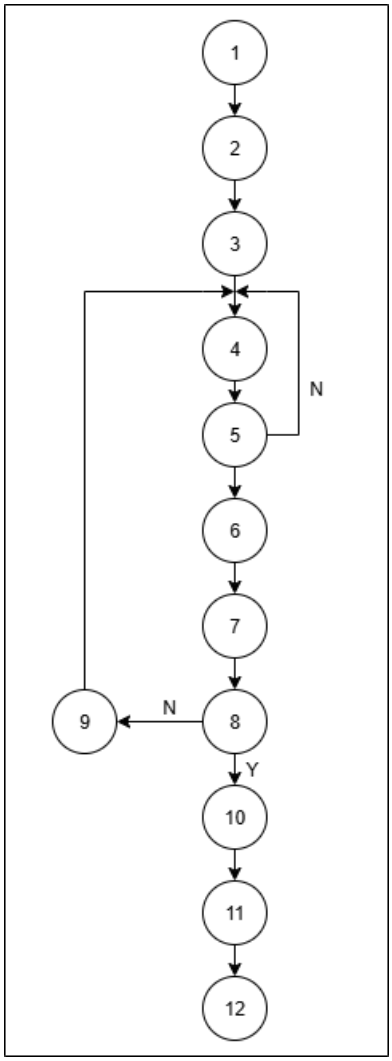
c) *Whitebox Testing Scan* Produk

1) Flowchart



Gambar 4. 20 *Flowchart Scan* Produk

2) Flowgraph



Gambar 4. 21 *Flowgraph Scan* Produk

Berdasarkan flowgraph pada gambar 4. 21, maka dilakukan perhitungan sebagai berikut:

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

V(G) = E – N + 2

N (*node*) = 13

E (*edge*) = 12

P (*predikat node*) = 2

Penyelesaian :

V(G) = E – N + 2

= 13 – 12 + 2

= 3

Predikat (P) = P + 1

= 3

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 3

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

Path 1 = 1 – 2 – 3 – 4 – 5 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 - 12

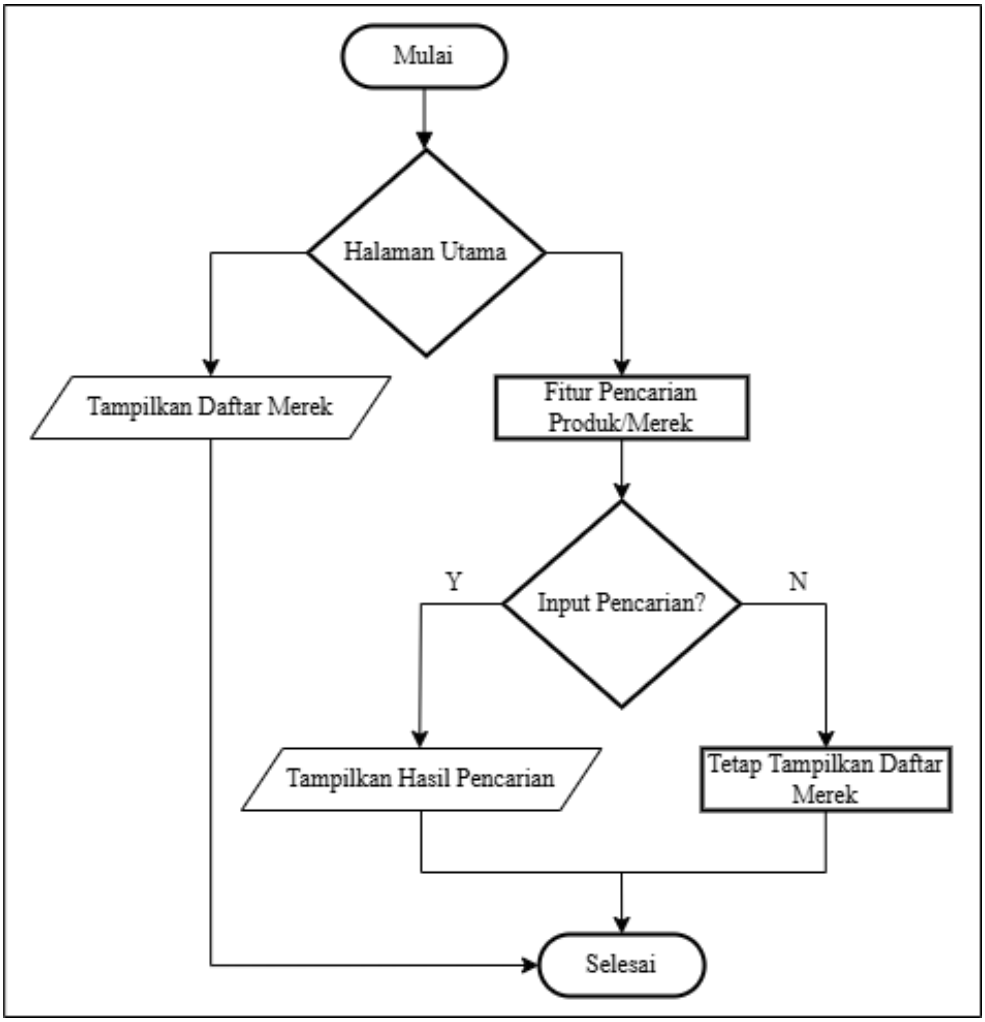
d. Grafik Matriks

Tabel 4. 18 Grafik Matriks *Scan* Produk

	1	2	3	4	5	6	7	8	9	10	11	12	E - 1
1		1											1 – 1 = 0
2			1										1 – 1 = 0
3				1									1 – 1 = 0
4					1								1 – 1 = 0
5				1		1							2 – 1 = 1
6							1						1 – 1 = 0
7								1					1 – 1 = 0
8									1	1			2 – 1 = 1
9				1									1 – 1 = 0
10											1		1 – 1 = 0
11												1	1 – 1 = 0
Sum (E + 1)													2 + 1 = 3

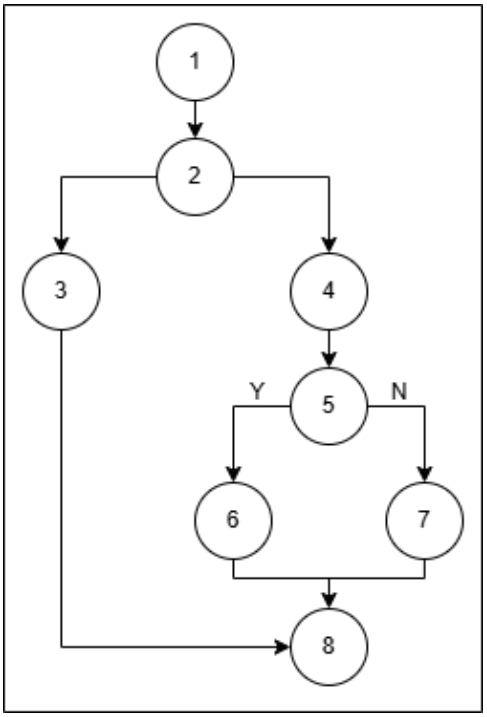
d) *Whitebox Testing* Pencarian Manual *(User)*

1) Flowchart



Gambar 4. 22 *Flowchart* Pencarian Manual (*User*)

2) Flowgraph



Gambar 4. 23 *Flowgraph* Pencarian Manual (*User*)

Berdasarkan flowgraph pada gambar 4. 23, maka dilakukan perhitungan sebagai berikut:

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

V(G) = E – N + 2

N (*node*) = 9

E (*edge*) = 8

P (*predikat node*) = 2

Penyelesaian :

V(G) = E – N + 2

= 9 – 8 + 2

= 3

Predikat (P) = P + 1

= 3

b. Dari perhitungan *Cyclomatic Complecity* dari *Flowgraph* diatas memiliki *Region* = 3

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

Path 1 = 1 – 2 – 3 – 8

Path 2 = 1 – 2 – 4 – 5 – 6 – 8

Path 3 = 1 – 2 – 4 – 5 – 7 – 8

d. Grafik Matriks

Tabel 4. 19 Grafik Matriks Pencarian Manual (*User*)

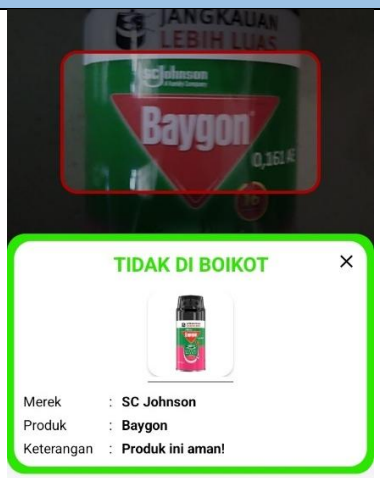
	1	2	3	4	5	6	7	8	E - 1
1		1							1 – 1 = 0
2			1	1					2 – 1 = 1
3								1	1 – 1 = 0
4					1				1 – 1 = 0
5						1	1		2 – 1 = 1
6								1	1 – 1 = 0
7								1	1 – 1 = 0
Sum (E + 1)									2 + 1 = 3

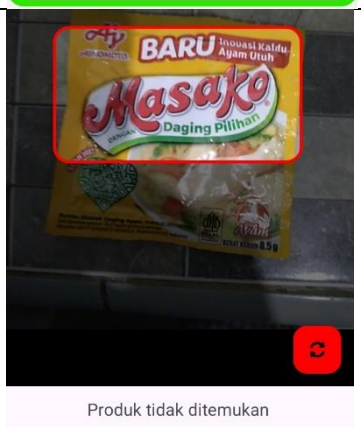
e) Pengujian Fungsionalitas OCR

Pengujian menunjukkan bahwa aplikasi berhasil mengenali teks pada kemasan produk dengan font standar atau yang terbaca. Sebaliknya, pengenalan kurang optimal pada kemasan dengan font yang tidak biasa, bentuk huruf yang rumit, atau struktur font yang tidak teratur. Tidak seperti dokumen formal dengan format teks

linier dan seragam, nama produk sering kali menggunakan desain rumit seperti, melengkung, bersilangan, bertumpuk, bahkan bergambar, yang mempersulit proses pemindaian. Hasil pengujian pada 15 produk disajikan dalam tabel 4.20.

Tabel 4. 20 Tabel Pengujian Fungsionalitas OCR.

No	Nama Produk	Gambar	Hasil	Keterangan
1	Baygon		✔	Desain nama produk “Baygon”, menggunakan font standar dengan struktur penulisan nama produk yang teratur sehingga mudah dibaca.
2	Pantene		✔	Desain nama produk “Pantene”, menggunakan font standar dengan struktur penulisan nama produk yang teratur sehingga mudah dibaca.
3	Aquviva		✔	Desain nama produk “Aquviva”, menggunakan font standar dengan struktur penulisan nama produk yang teratur sehingga mudah dibaca.

No	Nama Produk	Gambar	Hasil	Keterangan
4	My Baby		✔	Desain nama produk “My Baby”, menggunakan font standar meskipun struktur penulisan nama produk yang bertumpuk namun tetap dapat terbaca.
5	Ultra Milk		✔	Desain nama produk “Ultra Milk”, menggunakan font standar meskipun struktur penulisan nama produk yang bertumpuk namun tetap dapat terbaca.
6	Masako		✘	Desain nama produk “Masako”, menggunakan font tidak biasa dengan struktur penulisan nama produk yang tidak seragam sehingga sulit/tidak dapat terbaca.

No	Nama Produk	Gambar	Hasil	Keterangan
7	Johnson's	 Produk tidak ditemukan	✗	Desain nama produk "Johnson's", menggunakan font tidak biasa sehingga sulit/tidak dapat terbaca.
8	Pepsodent		✓	Desain nama produk "Pepsodent", menggunakan font standar dengan struktur penulisan nama produk yang teratur sehingga mudah dibaca.
9	Giv		✓	Desain nama produk "Giv", menggunakan font standar dengan struktur penulisan nama produk yang teratur sehingga mudah dibaca.

No	Nama Produk	Gambar	Hasil	Keterangan
10	Sajiku	 Produk tidak ditemukan	✗	Desain nama produk “Sajiku”, menggunakan font tidak biasa sehingga sulit/tidak dapat terbaca.
11	SariWangi		✓	Desain nama produk “SariWangi”, menggunakan font standar dengan struktur penulisan nama produk yang teratur sehingga mudah dibaca.
13	Downy		✓	Desain nama produk “Downy”, menggunakan font standar dengan struktur penulisan nama produk yang sedikit rapat namun masih dapat dibaca.

No	Nama Produk	Gambar	Hasil	Keterangan
14	Mama Lemon		✗	Desain nama produk “Mama Lemon”, menggunakan font standar namun struktur penulisan nama produk yang bertumpuk sehingga sulit/tidak dapat terbaca.
15	Vicks		✓	Desain nama produk “Vicks”, menggunakan font standar dengan struktur penulisan nama produk yang teratur sehingga mudah dibaca.

f) Pengujian Fungsi OCR

Berikut logika pengambilan teks dari gambar kemasan menggunakan modul OCR.

```
private fun processImageProxy(imageProxy: ImageProxy) {
    val mediaImage = imageProxy.image
    if (mediaImage != null) {
        val image = InputImage.fromMediaImage(mediaImage,
imageProxy.imageInfo.rotationDegrees)
        tvResult.text = "Mendeteksi..."
        textRecognizer.process(image)
            .addOnSuccessListener { visionText ->
                var foundInBox = false
                for (block in visionText.textBlocks) {
                    val box = block.boundingBox
                    if (box != null) {
                        val rectF = RectF(box)
                        val mappedRect =
mapRectToViewCoordinates(rectF, imageProxy)
                        if
(scannerOverlay.scannerRect.contains(
                                mappedRect.centerX(),
mappedRect.centerY()) ) {
                            val detectedText =
block.text.trim()
                            if (detectedText.isNotBlank() &&
detectedText != lastDetectedText) {
                                lastDetectedText =
detectedTextcheckWithServer(detectedText)
                                foundInBox = true
                                break}  }  }  }
                if (!foundInBox) {
                    isProcessing = false // siap memproses
frame berikutnya        }}
            .addOnFailureListener {
                isProcessing = false // gagal proses
```

1) Path yang diuji:

- Gambar berhasil diproses → Teks diekstrak dengan benar.
- Gagal memproses gambar → Teks gagal dibaca (error handler).

2) Hasil:

- Teks dapat diambil dari kemasan jika pencahayaan dan teks jelas.
- Teks gagal dibaca pada kemasan buram/rusak → ditangani oleh blok onFailure

g) Pengujian Fungsi Pencocokan Teks dengan Database

Berikut logika yang memastikan teks hasil OCR dibandingkan dengan database boikot.

```
private fun processImageProxy(imageProxy: ImageProxy) {
    val mediaImage = imageProxy.image
    if (mediaImage != null) {
        val image = InputImage.fromMediaImage(mediaImage,
imageProxy.imageInfo.rotationDegrees)
        tvResult.text = "Mendeteksi..."
        textRecognizer.process(image)
            .addOnSuccessListener { visionText ->
                var foundInBox = false
                for (block in visionText.textBlocks) {
                    val box = block.boundingBox
                    if (box != null) {
                        val rectF = RectF(box)
                        val mappedRect =
mapRectToViewCoordinates(rectF, imageProxy) if
(scannerOverlay.scannerRect.contains(
                                    mappedRect.centerX(),
mappedRect.centerY()) ) {
                      val detectedText = block.text.trim()
                      if (detectedText.isNotBlank() &&
detectedText != lastDetectedText) {lastDetectedText =
detectedTextcheckWithServer(detectedText)
                                    foundInBox = true
                                    break}  }  }  }
                if (!foundInBox) {
                    isProcessing = false // siap memproses
frame berikutnya        }}
            .addOnFailureListener {
                isProcessing = false // gagal proses
```

1) Path yang diuji:

- Teks cocok dengan nama produk → return true.
- Teks tidak cocok dengan produk apapun → return false.

2) Hasil:

- Fungsi berhasil mengidentifikasi produk yang sesuai dengan database.
- Produk yang tidak ada dalam daftar tidak ditandai sebagai boikot.

h) Pengujian Kegunaan

Pengujian kegunaan bertujuan untuk mengevaluasi kegunaan dan pemahaman suatu aplikasi serta memastikan pengalaman pengguna yang memuaskan. Untuk tujuan ini, kuesioner dibagikan kepada peserta yang telah menguji aplikasi pengenalan teks secara langsung. Pengujian ini dilakukan dengan memberikan kuesioner berskala Likert kepada responden yang telah menggunakan sistem.

Tabel 4. 21 Tabel Responden

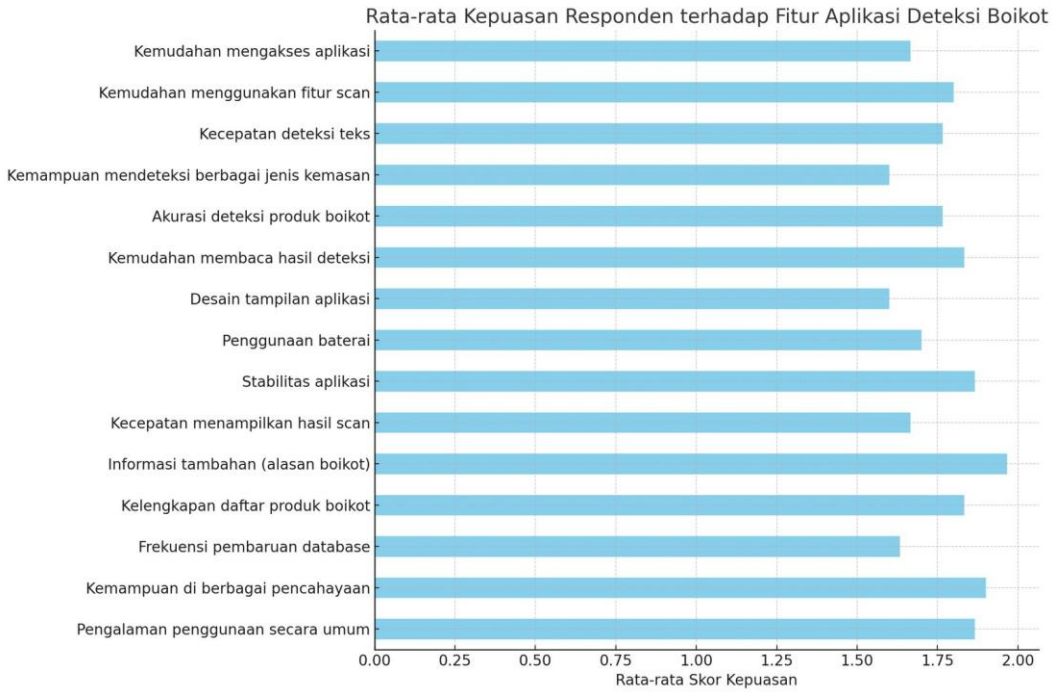
Responden	**Pertanyaan**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
R1	1	3	1	1	3	2	1	1	1	1	2	1	1	1	2
R2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R4	2	1	1	1	1	1	1	2	2	2	2	3	2	3	2
R5	1	1	1	1	2	2	1	2	1	2	1	2	1	2	1
R6	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
R7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R8	1	1	1	1	1	2	2	1	2	2	2	1	1	1	1
R9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R10	3	1	2	2	2	2	2	2	2	2	3	3	3	3	3
R11	1	2	2	2	2	1	1	1	1	2	2	2	1	1	1
R12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
R13	2	3	2	3	3	2	1	1	2	1	3	1	2	1	3
R14	1	2	3	1	1	2	1	1	2	1	2	2	1	3	1
R15	2	2	1	1	2	2	1	1	2	1	2	2	2	1	2
R16	1	2	1	1	1	2	2	3	3	1	2	2	1	1	1
R17	1	2	2	1	3	3	1	2	2	1	3	1	1	1	2
R18	1	2	2	1	1	1	2	2	2	3	3	2	2	3	2
R19	2	1	2	1	2	1	2	2	1	2	2	2	2	2	2

Responden	Pertanyaan														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R20	3	3	3	3	3	3	2	2	3	2	3	3	3	3	3
R21	1	1	1	1	1	1	1	1	1	1	1	1	1	3	2
R22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R23	2	2	2	1	1	2	2	2	2	2	2	2	2	2	2
R24	2	2	2	1	2	2	2	2	2	2	2	2	1	2	2
R25	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
R26	2	2	2	3	1	2	2	2	2	2	1	1	2	2	1
R27	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
R28	1	1	2	2	1	2	1	1	2	1	2	2	1	2	2
R29	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
R30	3	3	3	3	3	3	3	3	3	3	2	3	2	3	4

Setiap pertanyaan dalam kuesioner menggunakan skala likert dengan rentang nilai 1 sampai 5, yang memiliki makna sebagai berikut:

- 1 (4.00 – 5.00) = Sangat Puas (nilai tertinggi, menunjukkan kepuasan maksimal terhadap aspek yang dinilai)
- 2 (3.00 – 3.99) = Puas
- 3 (2.00 – 2.99) = Cukup Puas
- 4 (1.00 – 1.99) = Tidak Puas
- 5 = Sangat Tidak Puas (nilai terendah, menunjukkan ketidakpuasan maksimal).

Pada hasil nilai diatas, maka ditetapkan jika semakin rendah nilai rata-rata, semakin baik kualitas sistem menurut responden. Berdasarkan kuesioner yang diisi oleh 30 responden, dapat dilakukan penilaian kepuasan pengguna dan efektivitas aplikasi deteksi boikot berbasis OCR pada kemasan produk. Hasil analisis dalam bentuk diagram disajikan pada Gambar 4.24.



Gambar 4. 24 Grafik Rata-rata Kepuasan Responden

Grafik pada Gambar 4. 24. menunjukkan kepuasan rata-rata responden terhadap berbagai fitur aplikasi pendeteksi boikot. Secara keseluruhan, semua fitur mendapat penilaian cukup baik. Nilai mendekati 2 menunjukkan tingkat kepuasan yang tinggi. Informasi tambahan tentang alasan boikot dan kemampuan menggunakan berbagai jenis pencahayaan sangat dihargai. Namun, aspek seperti desain tampilan aplikasi dan kemampuan mendeteksi berbagai jenis kemasan mendapat penilaian agak rendah dan menunjukkan ruang untuk perbaikan.

BAB V

PENUTUP

A. Kesimpulan

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

1. Aplikasi pendeteksi boikot berbasis OCR berhasil dirancang dan diimplementasikan dengan memanfaatkan teknologi Optical Character Recognition untuk mengenali teks dari label produk dan mencocokkannya dengan database berisi daftar produk yang diboikot.
2. Proses identifikasi produk berjalan secara real-time dan dapat membantu konsumen memperoleh informasi status boikot suatu produk dengan mudah melalui pemindaian kemasan.
3. Software telah berhasil diuji dengan pengujian Black Box dan White Box, yang menunjukkan bahwa aplikasi dapat berfungsi seperti yang diinginkan pada fitur-fitur utama, seperti pemindaian, pencarian, dan pengelolaan informasi produk serta merek.
4. Keberhasilan proses pemindaian teks sangat dipengaruhi oleh jenis font, struktur penulisan, latar belakang kemasan, serta jarak pemindaian. OCR bekerja optimal pada font standar dengan susunan teks yang rapi dan terstruktur. Sebaliknya, teks dengan font dekoratif atau artistik, susunan yang tidak sejajar, bertumpuk, atau melengkung sering kali sulit dikenali. Latar belakang kemasan yang berpola atau terdiri dari banyak warna turut

memengaruhi akurasi pembacaan. Selain itu, pemindaian dari jarak yang terlalu jauh dapat menurunkan ketajaman gambar yang diproses.

5. Dengan adanya aplikasi ini, pengambilan keputusan etis dalam berbelanja dapat ditingkatkan, serta diharapkan mampu mendukung gerakan sosial dalam rangka meningkatkan kesadaran konsumen terhadap isu-isu kemanusiaan dan etika.

B. Saran

1. Peningkatan akurasi OCR: Aplikasi sebaiknya dikembangkan lebih lanjut untuk mengenali teks dari berbagai jenis kemasan, termasuk yang memiliki variasi font, warna, dan latar belakang yang kompleks.
2. Integrasi data yang lebih luas dan otomatis: Disarankan agar sistem database diperluas dengan dukungan API dari berbagai komunitas boikot atau organisasi pemantau etika perusahaan agar data selalu terbarui secara otomatis.
3. Pengembangan fitur pelaporan pengguna: Tambahkan fitur bagi pengguna untuk memberikan masukan atau menambahkan produk yang belum terdaftar, sehingga aplikasi lebih partisipatif dan responsif terhadap dinamika gerakan boikot.
4. Ekspansi ke produk non-konsumsi: Aplikasi dapat dikembangkan agar bukan hanya sebatas produk konsumsi sehari-hari, namun juga mencakup layanan atau perusahaan lain yang terafiliasi dengan isu sensitif.
5. Peningkatan keamanan dan privasi: Pengelolaan data pengguna dan hasil pemindaian perlu memperhatikan aspek keamanan informasi agar tidak disalahgunakan.

DAFTAR PUSTAKA

Akash Rawat, Anish Kumar, Aman Kumar Singh, & Dr Kavita Arora. (2024). Exploring Android Security Landscape: Threats, Vulnerabilities, and Best Practices. *Deleted Journal*, *2*(06), 1831–1839. https://doi.org/10.47392/IRJAEM.2024.0271

Amrute, P. S., Mohare, R. N., Yevatkar, S. D., Bagade, S. K., & Deshmukh, R. V. (2023). "JE Connect" an Educational Purpose Android App for Tracking Students Attendance, Performance with Parents Feedback System. *International Journal of Information Technology and Computer Engineering*, *33*, 15–22. https://doi.org/10.55529/IJITC.33.15.22

Courtney, C., & Neilsen, M. (2019). Vetting Anti-patterns in Java to Kotlin Translation. *Proceedings of 34th International Conference on Computers and Their Applications, CATA 2019*, *58*, 191–202. https://doi.org/10.29007/CGWR

Faizullah, S., Ayub, M. S., Hussain, S., & Khan, M. A. (2023). A Survey of OCR in Arabic Language: Applications, Techniques, and Challenges. *Applied Sciences*, *13*(7), 4584–4584. https://doi.org/10.3390/APP13074584

Guimarães, V., Nascimento, J., Viana, P., & Carvalho, P. (2023). A Review of Recent Advances and Challenges in Grocery Label Detection and Recognition. *Applied Sciences 2023, Vol. 13, Page 2871*, *13*(5), 2871. https://doi.org/10.3390/APP13052871

Hartati, S. (2020). PERANCANGAN SISTEM INFORMASI INVENTARIS BARANG PADA KANTOR NOTARIS DAN PPAT R.A LIA KHOLILA, S.H MENGGUNAKAN VISUAL STUDIO CODE. *SISKOMTI: Jurnal Sistem Informasi Komputer Dan Teknologi Informasi*, *2*(2), 37–48. https://doi.org/10.54342/8JMK7K05

Hartiwati, E. N. (2022). APLIKASI INVENTORI BARANG MENGGUNAKAN JAVA DENGAN PHPMYADMIN. *Cross-Border*, *5*(1), 601–610. https://journal.iaisambas.ac.id/index.php/Cross-Border/article/view/1113

Huda, B. (Baenil), & Priyatna, B. (Bayu). (2019). Penggunaan Aplikasi Content Management System (CMS) untuk Pengembangan Bisnis Berbasis E-commerce. *Systematics*, *1*(2), 81–88. https://doi.org/10.35706/SYS.V1I2.2076

KOZUB, Y., & KOZUB, H. (2023). Features of multiplatform application development on kotlin. *Vìsnik Hmel'nic'kogo Nacìonal'nogo Unìversitetu*, *317*(1), 224–229. https://doi.org/10.31891/2307-5732-2023-317-1-224-229

Kurbatova, Z., Golubev, Y., Kovalenko, V., & Bryksin, T. (2021). The IntelliJ Platform: a Framework for Building Plugins and Mining Software Data.

Proceedings - 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2021, 14–17. https://doi.org/10.1109/ASEW52652.2021.00016

Kusnantoro, K., Rohana, T., & Kusumaningrum, D. (2022). Implementasi Metode Tesseract OCR(Optical Character Recognition) untuk Deteksi Plat Nomor Kendaraan Pada Sistem Parkir. *Scientific Student Journal for Information, Technology and Science*, *3*(1), 59–67. https://journal.ubpkarawang.ac.id/mahasiswa/index.php/ssj/article/view/422

Lechterman, T. M., Jenkins, R., & Strawser, B. J. (2023). #StopHateForProfit and the Ethics of Boycotting by Corporations. *Journal of Business Ethics*, *191*(1), 77–91. https://doi.org/10.1007/S10551-023-05415-Y

Li, H., Alarcon, B., Espinosa, S. M., B, & Hecht. (2019). Out of Site: Empowering a New Approach to Online Boycotts. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW). https://doi.org/10.1145/3274375

Mairiza, D., Hasda, M., Winario, M., Assyifa, Z., & Irmawanti, I. (2024). Dampak Boikot Produk Afiliasi Pro Israel Bagi Pertumbuhan Perekonomian di Indonesia. *Journal of Economic, Management, Business, Accounting Sustainability*, *1*(3), 37–46. https://doi.org/10.69693/JOEMBAS.V1I3.53

Pambudi, A. R., Garno, & Purwantoro. (2020). DETEKSI KEASLIAN UANG KERTAS BERDASARKAN WATERMARK DENGAN PENGOLAHAN CITRA DIGITAL. *Jurnal Informatika Polinema*, *6*(4), 69–74. https://doi.org/10.33795/JIP.V6I4.407

Permana, B. A. (2022). APLIKASI PRESENSI ONLINE MENGGUNAKAN VALIDASI JARAK LOKASI PENGGUNA BERBASIS ANDROID. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, *3*(1), 86–92. https://doi.org/10.33365/JATIKA.V3I1.1865

Putra, D. W. T., & Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, *7*(1), 32–39. https://doi.org/10.21063/JTIF.2019.V7.1.32-39

Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. *JTEIN: Jurnal Teknik Elektro Indonesia*, *1*(2).

Sholeh, M., Gisfas, I., Cahiman, & Fauzi, M. A. (2021). Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods. *Journal of Physics: Conference Series*, *1823*(1). https://doi.org/10.1088/1742-6596/1823/1/012029

Syaikhuddin, M. M., Anam, C., Rinaldi, A. R., & Conoras, M. E. B. (2018). Conventional Software Testing Using White Box Method. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics,*

and Control, 65–72. https://doi.org/10.22219/KINETIK.V3I1.231

Wang, C. C., Chang, S. C., & Chen, P. Y. (2021). The Brand Sustainability Obstacle: Viewpoint Incompatibility and Consumer Boycott. *Sustainability 2021, Vol. 13, Page 5174*, *13*(9), 5174. https://doi.org/10.3390/SU13095174

Wulandari, S., Jupriyadi, J., & Fadly, M. (2021). RANCANG BANGUN APLIKASI PEMASARAN PENGGALANGAN INFAQ BERAS (STUDI KASUS: GERAKAN INFAQ). *TELEFORTECH : Journal of Telematics and Information Technology*, *2*(1), 11–16. https://doi.org/10.33365/TFT.V2I1.1991

Yaser, M., Moflih, Y., Benabdelouahed, R., & El Abdellaoui, M. (2023). Impact de la Communication de Crise Sur la Réputation de l'Entreprise - Cas des Réseaux Sociaux Boycott au Maroc. *European Scientific Journal*, *19*(16), 254–254. https://doi.org/10.19044/ESJ.2023.V19N16P254

Zailani, A. U., Octaviano, A., & Sholihin, S. (2021). Design of computer repair services application android-based. *TEKNOKOM*, *4*(1), 7–15. https://doi.org/10.31943/TEKNOKOM.V4I1.58