JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 14 No. 3, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v14i3.10578

IMPLEMENTASI OPTICAL CHARACTER RECOGNITION (OCR) PADA STRUK BELANJA MENGGUNAKAN TESSERACT UNTUK OTOMATISASI PENCATATAN PENGELUARAN

Najibah Aisyah Muhaa[1*], Nono Heryana[2], Billy Ibrahim Hasbi[3]

[1,2, 3]Universitas Singaperbangsa Karawang; Jl. HS. Ronggowaluyo, Telukjambe Timur, Karawang, 41363; Telp +62 812 1866 9229

Keywords:
OCR;
Tesseract;
Blackbox Testing;
Usability Testing;
Otomatisasi.

Corespondent Email:
2210631250068@student.unsika.ac.id

Abstrak. Penelitian ini bertujuan mengimplementasikan teknologi Optical Character Recognition (OCR) Tesseract untuk otomatisasi pencatatan pengeluaran dari struk belanja. Sistem dikembangkan dengan metode Waterfall untuk meminimalkan proses input manual yang tidak efisien. Pengujian sistem dilakukan melalui tiga tahap yang terdiri dari: Black Box Testing untuk validasi fungsi, Uji Akurasi untuk mengukur keberhasilan ekstraksi teks Tesseract, dan Usability Testing untuk menilai kemudahan bagi pengguna umum. Hasil pengujian Black Box Testing menunjukkan seluruh fitur fungsional berjalan baik. Pada pengujian OCR sistem mampu mengekstraksi data struk secara otomatis dengan tingkat akurasi paling optimal pada kondisi terang-jelas dengan nilai 98,46%, serta mendapatkan nilai "sangat baik" pada pengujian User Testing. Implementasi ini membuktikan OCR Tesseract efektif dalam mempercepat manajemen keuangan digital.

Copyright © 2026 The Author(s), Published by Jurnal Informatika dan Teknik Elektro Terapan. This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Abstract. *This research aims to implement Optical Character Recognition (OCR) using Tesseract to automate expense recording from shopping receipts. The system was developed using the Waterfall methodology to minimize the inefficient manual data entry process. System evaluation was conducted through three testing phases: Black Box Testing to validate system functionality, Accuracy Testing to measure the success of Tesseract's text extraction, and Usability Testing to assess ease of use for general users. The Black Box Testing results showed that all functional features operated correctly. In the OCR accuracy test, the system successfully extracted receipt data automatically, achieving its highest accuracy of 98.46% under clear and well-lit conditions. Furthermore, the system received an "excellent" rating in the Usability Testing. These results demonstrate that the implementation of Tesseract OCR is effective in improving the efficiency of digital financial management.*

1. PENDAHULUAN

Pencatatan pengeluaran merupakan bagian penting dalam pengelolaan keuangan pribadi maupun rumah tangga. Namun, aktivitas ini sering kali diabaikan karena dianggap merepotkan, terutama ketika harus mencatat setiap transaksi secara manual dari struk belanja. Akibatnya, individu kesulitan memantau arus kas pribadi, yang dapat berdampak pada ketidakterkendalinya pengeluaran dan kurangnya kesadaran finansial. Ketika individu kurang memiliki pemahaman yang cukup tentang bagaimana merencanakan dan mengelola keuangan,

mereka seringkali kesulitan mencapai tujuan kesejahteraan finansial yang diharapkan [1].

Seiring perkembangan teknologi, pendekatan manual mulai digantikan dengan solusi otomatis berbasis digital. Karena Teknologi informasi memegang peranan krusial dalam mendukung operasional organisasi dan sektor bisnis, baik di ranah publik maupun privat, dengan mengandalkan internet sebagai infrastruktur utamanya [2]. Salah satu teknologi yang potensial untuk mengatasi masalah pencatatan manual adalah Optical Character Recognition (OCR). OCR memungkinkan sistem untuk mengenali dan mengekstraksi teks dari gambar atau dokumen yang dipindai, termasuk struk belanja. Dengan mengimplementasikan OCR, informasi seperti nama toko, items, dan total harga, dapat diambil secara otomatis tanpa perlu input manual dari pengguna.

Optical Character Recognition (OCR) merupakan teknologi yang mampu mengonversi gambar teks, seperti struk belanja, menjadi format teks digital yang dapat dibaca dan diolah oleh komputer [3]. Dengan kemampuan ini, OCR dapat mengekstraksi informasi penting dari struk belanja secara otomatis, seperti nama toko, dan total pengeluaran. Hal ini memungkinkan proses pencatatan pengeluaran menjadi lebih cepat, akurat, dan efisien tanpa perlu entri data secara manual.

Implementasi OCR dalam pencatatan pengeluaran telah terbukti dapat mengurangi beban kerja manual dan meningkatkan akurasi data. Selain itu, kemajuan teknologi kecerdasan buatan (AI) yang terintegrasi dengan OCR semakin meningkatkan akurasi dan kecepatan ekstraksi data dari berbagai jenis dokumen, termasuk struk belanja yang memiliki format beragam. Dengan demikian, otomatisasi pencatatan pengeluaran berbasis OCR dapat menjadi solusi inovatif untuk memudahkan individu maupun organisasi dalam mengelola keuangan secara digital [4].

Penelitian ini berfokus pada pemanfaatan teknologi *Optical Character Recognition* (OCR) dalam proses digitalisasi data struk belanja untuk otomatisasi pencatatan pengeluaran. Sistem yang dikembangkan dirancang untuk mengekstraksi informasi penting dari gambar struk seperti total belanja, dan rincian item secara otomatis, sehingga pengguna tidak perlu melakukan pencatatan secara manual. Teknologi OCR dipilih karena kemampuannya dalam membaca teks dari gambar dengan tingkat akurasi yang semakin meningkat. Sistem ini diharapkan dapat membantu meningkatkan efisiensi pencatatan, mengurangi human error, dan mendorong kebiasaan pengelolaan keuangan yang lebih teratur.

2. TINJAUAN PUSTAKA

2.1 Optical Character Recognition (OCR)



Gambar 2. 1 Alur OCR
Sumber: [5]

Pada gambar 2.1 merupakan proses pengolahan *Optical Character Recognition* (OCR), OCR merupakan sekumpulan teknik dan proses yang digunakan untuk mengekstraksi informasi karakter dari citra digital. Pada jurnal yang berjudul ”Aplikasi Ekstraksi Data Kartu Vaksi Berbasis Web Menggunakan Metode OCR” dijelaskan bahwa OCR bekerja dengan mengubah data visual hasil pemindaian atau gambar digital menjadi data teks, di mana sinyal analog dari dokumen terlebih dahulu ditransformasikan oleh scanner menjadi data digital yang mengandung karakter untuk kemudian diinterpretasikan oleh sistem OCR menjadi string teks yang dapat diolah lebih lanjut oleh aplikasi [6]. OCR digunakan untuk mengotomatisasi proses entri data, sehingga teks yang terdapat dalam gambar seperti struk belanja dapat diubah menjadi format digital yang dapat diproses lebih lanjut. Teknologi ini penting dalam digitalisasi dokumen dan sangat relevan dalam konteks pengolahan struk belanja secara otomatis.

2.2 *Tesseract*

Tesseract merupakan salah satu mesin OCR *open-source* yang dikembangkan oleh HP dan kemudian dilanjutkan oleh Google [7]. Tesseract mendukung banyak bahasa, termasuk Bahasa Indonesia, dan dikenal karena kemampuannya yang fleksibel dan akurat dalam mengenali karakter.

2.3 Google Gemini AI

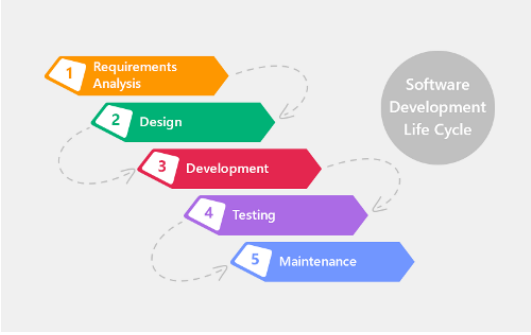
Gemini AI merupakan model multimodal yang dikembangkan oleh Google dengan kemampuan native vision untuk memahami dan mengekstrak informasi dari gambar dan dokumen. Gemini mampu mengenali teks, tabel, diagram, dan tulisan tangan dari berbagai jenis dokumen dan gambar, serta mengubah informasi tersebut menjadi format terstruktur seperti JSON atau tabel [8].

2.4 Pengolahan Citra Digital

Pengolahan citra merupakan teknik untuk meningkatkan kualitas gambar melalui proses komputasi. Proses ini bekerja dengan menerima citra sebagai input, kemudian memodifikasinya sehingga menghasilkan citra baru sesuai kebutuhan. Secara umum, pengolahan citra dibedakan menjadi tiga jenis, yaitu citra warna, citra biner, dan citra *grayscale* atau hitam putih [9] Pengolahan citra terbagi menjadi 2 proses yakni :

- Pengolahan citra analog biasanya diterapkan pada kegiatan seperti penggandaan atau pencetakan gambar, sedangkan
- Pengolahan citra digital lebih banyak digunakan karena prosesnya dilakukan dengan bantuan komputer [10].

2.5 *Waterfall*



Gambar 2. 2 Waterfall Model
Sumber: [11]

Pada gambar 2.2 merupakan alur pada Model Waterfall, Waterfall adalah salah satu metode pengembangan perangkat lunak tradisional yang bersifat sekuensial dan sistematis. Model ini memiliki beberapa tahapan utama, yaitu:

- *Requirements Analysis* : Mengidentifikasi kebutuhan pengguna dan masalah yang harus diselesaikan.
- *Design* : Menyusun rancangan sistem, seperti struktur database, alur data, serta desain antarmuka.
- *Development* : Mengubah rancangan menjadi kode program sehingga sistem dapat dijalankan sesuai fungsi yang diinginkan.
- *Testing* : Memeriksa apakah sistem berjalan dengan benar dan sesuai kebutuhan.
- *Maintenance* : Melakukan perbaikan dan pembaruan setelah sistem digunakan, agar tetap stabil dan dapat memenuhi kebutuhan pengguna di masa depan [12].

2.6 *Unified Modelling Language* (UML)

UML (*Unified Modeling Language*) adalah bahasa pemodelan standar yang digunakan untuk menggambarkan dan merancang sistem perangkat lunak melalui berbagai diagram visual, seperti *use case, activity, class*. UML membantu menyusun struktur dan alur kerja sistem secara jelas sehingga memudahkan pengembang dalam memahami, mengkomunikasikan, dan mendokumentasikan rancangan sistem selama proses pengembangan [13].

2.7 *Framework*

Framework adalah sekumpulan fungsi, kelas, seperangkat aturan yang tersusun secara sistematis dan berperan menetapkan pola pengembangan aplikasi [14]. *Framework* menyediakan kerangka kerja lengkap yang mengatur alur pembangunan perangkat lunak yang digunakan dengan tujuan tertentu. Dengan demikian, proses pengembangan menjadi lebih cepat, efisien, dan terstandarisasi, serta mendukung pemeliharaan dan skalabilitas sistem dalam jangka panjang.

2.8 *Rest Api*

REST (Representational State Transfer) adalah arsitektur komunikasi berbasis web yang menggunakan protokol HTTP untuk mengakses sumber daya melalui metode seperti GET, POST, dan DELETE, serta memanfaatkan format JSON agar pertukaran data lebih ringan dan efisien. Sementara itu, Application

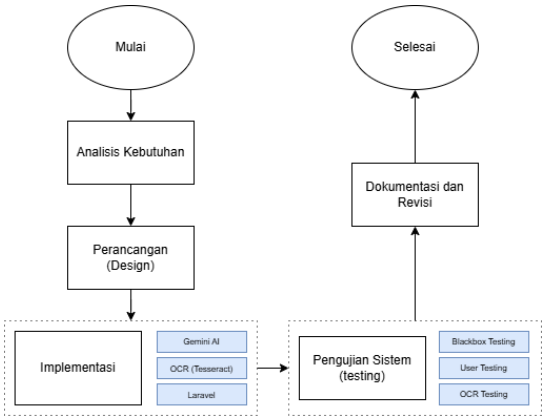
Programming Interface (API) memungkinkan komunikasi antara aplikasi klien dan server. Dalam penelitian ini, RESTful API digunakan sebagai penghubung antara aplikasi mobile dan backend untuk mengirim dan menerima data secara aman, terstruktur, dan efisien dengan dukungan autentikasi menggunakan Laravel Sanctum dan JWT [15].

2.9 *Pengujian Perangkat Lunak*

Pengujian perangkat lunak merupakan proses untuk memastikan bahwa sistem berfungsi dengan benar serta memenuhi kebutuhan pengguna. Selain itu, pengujian juga dipahami sebagai prosedur untuk menjalankan program dengan tujuan menemukan kesalahan, sehingga membantu pengembang mengidentifikasi dan memperbaiki cacat sistem sebelum digunakan oleh pengguna akhir. Dalam konteks yang lebih luas, pengujian perangkat lunak berfungsi sebagai kegiatan verifikasi dan validasi kualitas perangkat lunak, baik dari aspek fungsional maupun nonfungsional, sehingga perangkat lunak dapat memberikan performa yang optimal, antarmuka yang layak, serta keandalan yang sesuai dengan kebutuhan operasional pengguna [16].

3. METODE PENELITIAN

Dalam penelitian ini terdapat kerangka kerja sebagai rancangan penelitian yang akan dilakukan sebagai gambaran. Rancangan penelitian ini memiliki penjelasan tahapan sebagai berikut.



Gambar 3. 1 Alur Penelitian

3.1 Analisis Kebutuhan

Berdasarkan tahapan alur penelitian pada Gambar 3. 1 Alur Penelitian, analisis kebutuhan dilakukan dengan mengidentifikasi masalah secara langsung kepada calon pengguna melalui wawancara, dan didukung oleh studi pustaka yang sudah dilakukan sebelumnya. Lalu diidentifikasi kebutuhan dalam bentuk kebutuhan fungsional dan non fungsional.

3.2 Perancangan (*Design*)

Tahap perancangan dilakukan dengan menyusun struktur dan alur kerja sistem berdasarkan hasil analisis kebutuhan. Perancangan dimulai dengan pembuatan diagram Unified Modeling Language (UML), yaitu Use Case Diagram untuk menggambarkan interaksi antara pengguna dan sistem, Activity Diagram untuk mendeskripsikan alur proses kerja, serta Class Diagram untuk menunjukkan struktur data dan hubungan antar entitas yang digunakan.

3.3 Implementasi

Tahap implementasi dilakukan dengan menerjemahkan desain sistem ke dalam bentuk kode program sesuai diagram UML yang telah disusun pada tahap perancangan. Implementasi dimulai dengan membangun struktur aplikasi menggunakan framework Laravel yang mengatur routing, pengelolaan request, autentikasi sederhana, struktur MVC, dan komunikasi dengan basis data MySQL. Pada sisi frontend juga menggunakan templating engine dari laravel. Pada proses pengunggahan gambar struk akan diintegrasikan dengan Tesseract.js, yaitu library OCR berbasis JavaScript yang dipilih karena dapat melakukan ekstraksi teks langsung di sisi klien dan mengurangi beban pada server.

Setelah Tesseract.js menghasilkan teks mentah dari gambar, hasil tersebut diproses dan disempurnakan menggunakan Gemini AI, yang berperan dalam menormalkan, memvalidasi, dan menginterpretasikan teks agar sesuai dengan struktur data transaksi. Hasil yang telah divalidasi kemudian dikirim ke backend Laravel untuk diolah lebih lanjut dan disimpan dalam database MySQL.

Alur kerja implementasi sistem terdiri dari beberapa tahap utama dimulai dari pengguna mengunggah gambar struk melalui antarmuka web, lalu sistem akan menjalankan proses OCR menggunakan Tesseract.js, hasil ekstraksi selanjutnya akan dianalisis dan

diperbaiki oleh Gemini AI, kemudian Laravel menerima data akhir untuk disimpan dan ditampilkan kembali kepada pengguna dalam bentuk pencatatan pengeluaran. Dengan pendekatan ini, rancangan sistem yang sebelumnya dibuat dalam bentuk Use Case Diagram, Activity Diagram, dan Class Diagram dapat direalisasikan secara terstruktur dalam kode program sehingga seluruh alur unggah, ekstraksi, validasi, dan penyimpanan data berjalan otomatis, cepat, dan responsif.

3.4 Pengujian Sistem (Testing)

Pada tahap pengujian, sistem diuji untuk memastikan seluruh fungsi berjalan sesuai kebutuhan. Pengujian dilakukan menggunakan :

1. Black Box Testing untuk menguji fitur dari sisi pengguna, seperti unggah gambar, ekstraksi teks, penyimpanan data, dan tampilan hasil pencatatan.
2. Pengujian Akurasi Pembacaan OCR dengan membandingkan hasil OCR terhadap isi struk asli pada pengujian turut mencakup berbagai kondisi gambar struk, seperti pencahayaan rendah atau teks yang pudar, untuk menilai keandalan OCR.
3. Usability Testing dilakukan dengan melibatkan pengguna untuk mencoba langsung fitur utama sistem dan memberikan umpan balik terkait kenyamanan penggunaan. Evaluasi ini mencakup persepsi pengguna terhadap kemudahan navigasi, kemudahan memahami antarmuka, serta penilaian subjektif mengenai kecepatan sistem apakah dirasa cepat, lambat, atau sesuai harapan.

Hasil seluruh pengujian ini digunakan untuk menilai performa sistem dan menentukan bagian yang perlu ditingkatkan.

3.5 Dokumentasi dan Revisi

Pada tahap ini dilakukan dokumentasi dan revisi terhadap sistem yang dikembangkan untuk memastikan fitur sudah sesuai kebutuhan dan bebas dari error atau bug. Dokumentasi mencakup struktur database, diagram sistem, serta penjelasan teknis terkait integrasi Tesseract.js, Gemini AI, dan Laravel. Apabila ditemukan kesalahan, dilakukan revisi berdasarkan hasil pengujian maupun temuan selama implementasi. Tahapan revisi meliputi identifikasi bug, penentuan prioritas perbaikan, pembaruan modul terkait OCR, parser, dan antarmuka pengguna. Evaluasi akurasi dijalankan kembali setelah revisi untuk menilai peningkatan kinerja sistem, dengan baseline minimal 50% sebagai prototipe awal dan target peningkatan minimal 70% pada field utama sebelum sistem dapat digunakan secara lebih luas.

4. HASIL DAN PEMBAHASAN

Data hasil analisis kebutuhan berupa wawancara secara langsung dibuat analisis kebutuhan dalam bentuk kebutuhan functional dan nonfunctional, kemudian kebutuhan tersebut disusun sesuai rancangan penelitian.

4.1 Analisis Kebutuhan

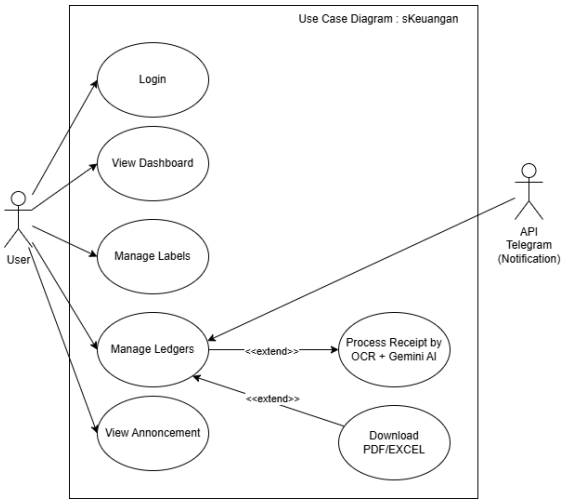
Hasil dari wawancara yang telah diajukan bahwa pengembangan sistem keuangan atau yang Bernama “sKeuangan” ini didasarkan pada analisis kebutuhan fungsional dan non-fungsional untuk memastikan efektivitas serta keandalan aplikasi saat diimplementasikan. Dari aspek fungsional, sistem menyediakan kapabilitas visualisasi data melalui antarmuka dashboard interaktif yang dilengkapi fitur penyaringan (filtering) berbasis periode bulan dan tahun, serta modul announcement untuk merepresentasikan kondisi finansial secara real-time. Manajemen data transaksi diakomodasi melalui fitur CRUD (Create, Read, Update, Delete) pada menu labels dan ledgers yang memfasilitasi pencatatan pos pemasukan serta pengeluaran.

Untuk mendukung aksesibilitas data, modul ini juga diintegrasikan dengan Application Programming Interface (API) Telegram sebagai media notifikasi otomatis serta menyediakan fitur retrival laporan dalam format Excel dan PDF. Selain entri data manual, fitur utama sistem terletak pada otomatisasi input melalui pengunggahan citra struk belanja (format .jpg, .jpeg, dan .png). Citra tersebut kemudian diproses menggunakan mesin Optical Character Recognition (OCR) Tesseract untuk mengekstraksi entitas penting seperti nama gerai, nominal pengeluaran, serta rincian item belanja sebelum akhirnya disimpan ke dalam basis data relasional.

Sementara itu, pemenuhan kebutuhan non-fungsional ditujukan untuk menjaga kualitas layanan (Quality of Service) dari sistem yang dibangun. Aspek usability diimplementasikan melalui perancangan

antarmuka yang intuitif guna memudahkan navigasi pengguna non-teknis, dengan jaminan compatibility pada peramban web modern seperti Google Chrome dan Microsoft Edge. Dari segi performa dan reliabilitas, sistem dikonfigurasi untuk menangani beban komputasi proses OCR dalam batas waktu respons maksimal lima menit pada kondisi jaringan normal, di mana akurasi hasil ekstraksi teks sangat dipengaruhi oleh kualitas resolusi dan intensitas pencahayaan pada citra struk yang diunggah. Terakhir, faktor keamanan (security) diterapkan secara berlapis melalui mekanisme autentikasi pengguna (login session) serta validasi input berkas yang ketat guna meminimalisasi risiko eksploitasi kode berbahaya melalui unggahan dokumen.

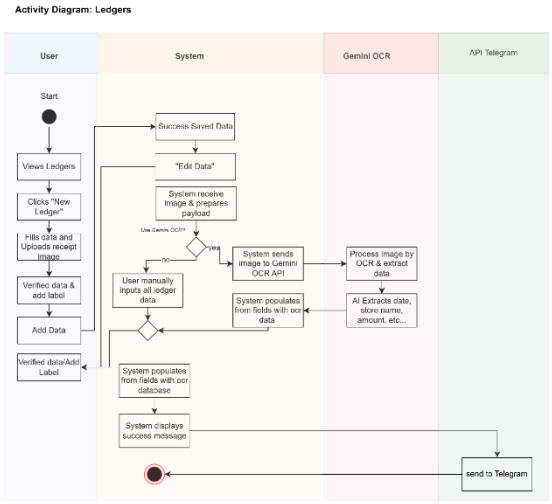
4.2 Perancangan (Design)



Gambar 4. 1 Use Case Diagram

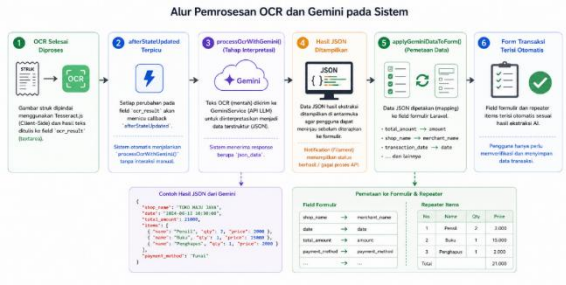
Berdasarkan Use Case Diagram Gambar 4.1, pengguna sebagai aktor utama dengan hak akses penuh ke seluruh fitur sistem melalui mekanisme single account login. Fungsionalitas utama mencakup pemantauan ringkasan keuangan pada modul Dashboard dan Announcement, serta manajemen data transaksi melalui fitur Manage Labels dan Manage Ledgers. Untuk meningkatkan efisiensi dan keamanan, sistem ini mengintegrasikan teknologi OCR dan Gemini AI guna otomatisasi pencatatan keuangan dari struk belanja, serta API Telegram untuk pengiriman notifikasi pembaruan data secara real-time.

Menu utama pada web sKeuangan ada pada menu ledgers yang terdapat fitur ekstraksi dengan gemini AI dan Tesseract.js. pada gambar 4.2 Activity Diagram Ledgers, proses dimulai saat pengguna membuka menu view Ledgers dan menekan tombol Ledger. Pengguna kemudian menambahkan data yang harus diisi dan menguploads gambar lalu disimpan. Setelah data disimpan akan masuk ke menu edit dan pengguna dapat memilih melanjutkan process menggunakan OCR + Gemini AI atau menambahkan label data (store name, store address, entry detail yang berisikan item name, jumlah, dan price) secara manual. Setelah disimpan data akan masuk kedalam database dan mengirimkan success message dan mengirimkan informasi perubahan tersebut ke telegram.



Gambar 4. 2 Activity Diagram Ledgers

4.3 Implementasi



Gambar 4. 3 Pemrosesan OCR dan Gemini AI

Implementasi tersebut diterapkan pada halaman Edit Transaction, di mana hasil OCR yang tersimpan pada field ocr_result secara otomatis diproses oleh Gemini AI melalui

mekanisme afterStateUpdated. Data JSON yang dihasilkan kemudian dipetakan ke dalam field formulir dan komponen repeater sehingga informasi transaksi beserta daftar barang dapat terisi secara otomatis. Dengan demikian, pengguna hanya perlu melakukan verifikasi terhadap hasil identifikasi sebelum data disimpan ke dalam database. Alur proses mulai dari hasil OCR, lalu diinterpretasi oleh Gemini AI, hingga pemetaan data ke formulir ditunjukkan pada Gambar 4.3 Alur Pemrosesan OCR dan Gemini AI.

4.4 Pengujian Sistem (Testing)

Pengujian Sistem dilakukan dengan Blackbox Testing, User Testing, dan OCR Testing. Hasil blackbox testing secara umum menunjukkan performa yang baik, meskipun terdapat indikasi warning pada modul Gemini AI (gemini-2.5-flash), akibat kendala timeout serta limitasi kuota API. Sementara itu, user testing yang melalui sesi demonstrasi langsung dan pengisian kuesioner menghasilkan predikat "Sangat Baik" dengan skor 86.29%, dengan catatan evaluasi pada peningkatan aspek UI/UX agar lebih intuitif bagi pengguna.

Pengujian dilakukan menggunakan skala biner, yaitu 0 dan 1. Skor 0 menunjukkan bahwa hasil identifikasi OCR yang diproses menggunakan Gemini AI tidak dapat dikenali atau terbaca dengan jelas, sedangkan skor 1 menunjukkan bahwa informasi pada struk berhasil dikenali dan diidentifikasi dengan baik.

No.	Foto	Fixed		Dynamic				%
		SN	SA	No	Item	Jumlah	Price	
1.	Struk1	0	0	1.	1	1	1	60%
2.	Struk2	1	0	1.	0	1	1	45,45%
				2.	0	1	0	
				3.	0	1	0	
3.	Struk3	0	0	1.	1	1	0	50%
				2.	0	1	1	
4.	Struk4	0	0	1.	1	0	0	20%
5.	Struk5	0	0	1.	1	1	1	75%
				2.	1	1	1	

Tabel 4. 1 Struk Rusak-Gelap

Digunakan rumus rata-rata untuk menhitung akurasi dari struk yang diuji, seperti rumus dibawah ini:

$$\textbf{Rata-Rata} = \frac{Jumlah\ Persentase}{Jumlah\ Sample}$$

$$\textbf{Rata-Rata} = \frac{250{,}45\%}{5} = 50{,}09\%$$

Pada Tabel 4. 1 kondisi struk rusak dan pencahayaan yang gelap, memiliki akurasi paling buruk yakni 50,09%. Dimana sistem mengalami kesulitan dalam mengenali karakter. Rendahnya akurasi ini disebabkan oleh dua kendala utama yaitu fisik struk yang tidak utuh atau cacat menyebabkan bentuk karakter terdistorsi dan minimnya cahaya menghilangkan kontras warna dan memicu noise pada gambar.

No.	Foto	Fixed		Dynamic				%
		SN	SA	No	Item	Jumlah	Price	
1.	Struk1	1	1	1.	1	1	1	100%
2.	Struk2	1	0	1.	1	1	0	63,63%
				2.	1	1	1	
				3.	0	1	0	
3.	Struk3	0	1	1.	1	1	0	62,5%
				2.	0	1	1	
4.	Struk4	1	0	1.	1	0	0	40%
5.	Struk5	0	0	1.	1	1	1	75%
				2.	1	1	1	

Tabel 4. 2 Struk Rusak-Terang

Pada Tabel 4. 2 kondisi struk rusak dan pencahayaan yang terang, memiliki akurasi yang lebih baik dengan nilai 72,22% dibandingkan kondisi rusak dan gelap 50,09%. Intensitas cahaya yang optimal mampu meningkatkan contrast ratio antara karakter teks dengan latar belakang kertas struk. Selain itu, pencahayaan yang cukup juga meminimalisir kemunculan digital noise dan efek motion blur pada sensor kamera saat proses akuisisi data dilakukan.

No.	Foto	Fixed		Dynamic				%
		SN	SA	No	Item	Jumlah	Price	
1.	Struk1	1	1	1.	0	1	1	80%
2.	Struk2	1	1	1.	1	1	1	100%
				2.	1	1	1	
3.	Struk3	0	1	1.	1	1	1	80%
4.	Struk4	0	1	1.	1	1	1	90%
				2.	1	1	0	
				3.	1	1	1	
				4.	1	1	1	
				5.	1	1	1	
				6.	1	1	1	
5.	Struk5	0	0	1.	1	1	1	92,30%
				2.	1	1	1	
				3.	1	1	1	
				4.	1	1	1	
				5.	1	1	1	
				6.	1	1	1	
				7.	1	1	1	
				8.	1	1	1	

Tabel 4. 3 Struk Jelas-Gelap

Pada tabel 4. 3 kondisi struk jelas dan pencahayaan yang gelap, memiliki akurasi

88,46% yang menunjukkan tingkat akurasi cukup tinggi. Pada tabel nilai 0 kemungkinan terjadi karena keadaan gelap yang menimbulkan banyak noise pada gambar.

No.	Foto	Fixed		Dynamic				%
		SN	SA	No	Item	Jumlah	Price	
1.	Struk1	1	1	1.	1	1	1	100%
2.	Struk2	1	1	1.	1	1	1	100%
				2.	1	1	1	
3.	Struk3	1	1	1.	1	1	1	100%
4.	Struk4	1	1	1.	1	1	1	100%
				2.	1	1	1	
				3.	1	1	1	
				4.	1	1	1	
				5.	1	1	1	
				6.	1	1	1	
5.	Struk5	0	1	1.	1	1	1	92,30%
				2.	1	1	1	
				3.	1	1	1	
				4.	1	1	1	
				5.	1	1	1	
				6.	1	1	1	
				7.	1	1	1	
				8.	1	1	0	

Tabel 4. 4 Struk Jelas-Terang

Pada tabel 4. 4 kondisi struk jelas dan pencahayaan yang terang, memiliki akurasi 98,46%. Paling tinggi dibanding ke tiga kondisi sebelumnya (rusak-gelap, rusak-terang, dan jelas-gelap). Berdasarkan perbandingan seluruh data tersebut, dapat disimpulkan bahwa kualitas fisik dari dokumen struk belanja dan kondisi pencahayaan memegang peranan yang jauh lebih dominan dalam menentukan keberhasilan. Oleh karena itu, sistem OCR ini dapat dinyatakan andal untuk digunakan pada kondisi sehari-hari, dengan syarat struk yang diproses dalam kondisi fisik yang bersih, utuh, tidak cacat, dan kondisi cahaya yang baik agar mendapatkan hasil yang optimal.

4.5 Dokumentasi dan Revisi

Berdasarkan hasil pengujian, implementasi Optical Character Recognition (OCR) dan Gemini AI telah berjalan optimal serta memenuhi target akurasi baseline minimal pada kondisi struk jelas-terang. Sebagai bentuk penyempurnaan sistem pada tahap akhir, dilakukan dokumentasi dan revisi arsitektur melalui penambahan modul antarmuka pendukung menggunakan n8n dan bot Telegram untuk memperluas fungsionalitas pencatatan keuangan. Melalui integrasi ini, pengguna dapat mengunggah citra struk, melakukan entri data transaksi via pesan teks (store, items, total), mengakses laporan (report) keseluruhan, serta membuka panduan fitur (help), di mana seluruh aktivitas data tersebut otomatis terekam ke dalam Google Sheets. Sementara pada platform web, pembaruan dilakukan dengan mengintegrasikan modul alur kerja n8n langsung pada menu edit.

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan pada ”Implementasi Optical Character Recognition (OCR) Pada Struk Belanja Menggunakan Tesseract Untuk Otomatisasi Pencatatan Pengeluaran”, maka kesimpulan yang dapat diperoleh sebagai berikut:

1. Teknologi Optical Character Recognition (OCR) berhasil digunakan untuk mengekstraksi data dari struk belanja secara otomatis, meliputi informasi nama toko, nama barang, dan harga, dengan menggabungkan Tesseract OCR dan Gemini AI. Hasil Blackbox Testing menunjukkan bahwa hampir seluruh fungsi sistem berjalan dengan valid, meskipun terdapat kondisi warning yang disebabkan oleh server sibuk, timeout, atau batasan token API.
2. Otomatisasi OCR terbukti efektif mempermudah proses pencatatan dibanding metode manual. Berdasarkan uji User Testing, sistem ini mendapatkan penilaian kepuasan yang sangat tinggi sebesar 86,29%.
3. Sistem berbasis OCR berhasil mampu mempermudah dan meningkatkan efisiensi pengelolaan keuangan sehari-hari bagi individu maupun rumah tangga melalui proses pencatatan transaksi yang lebih cepat dan otomatis. Berdasarkan performa ekstraksi OCR pada kondisi Jelas-Terang (98,46%) dinilai sangat optimal, meskipun mengalami penurunan pada kondisi seperti Jelas-Gelap (88,46%), Rusak-Terang (72,22%), dan Rusak-Gelap (50,09%).

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] H. A. Putra, H. Halpiah, and R. Rosita, "Transparansi arus kas rumah tangga keluarga menggunakan aplikasi expense track," *Jurnal Inovasi Hasil Pengabdian Masyarakat (JIPEMAS)*, vol. 6, no. 1, pp. 125–138, Mar. 2023, doi: 10.33474/jipemas.v6i1.19079.

[2] D. MARDIATI and Y. SAPUTRA, "IMPLEMENTASI SISTEM INFORMASI MANAJEMEN KLINIK MENGGUNAKAN METODE BLACK BOX TESTING," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 1, Jan. 2025, doi: 10.23960/jitet.v13i1.6015.

[3] D. Piter and F. S. Lee, "RANCANG BANGUN APLIKASI PELACAK KEUANGAN BERBASIS OCR UNTUK MANAJEMEN PENGELUARAN DEPARTEMEN KEUANGAN," *JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang*, vol. 9, 2025.

[4] R. Sari, I. Musa Adi, and A. Hidayati, "Personal Track Your Cash: Prototipe Aplikasi Pembacaan Setruk 0020 Belanja Menggunakan OCR dan Google Vision," 2023.

[5] Klippa, "Tesseract OCR: What It Is and Why Choose It." Accessed: Jun. 15, 2025. [Online]. Available: https://www.klippa.com/en/blog/information/tesseract-ocr/

[6] Wahyuddin, "APLIKASI EKSTRAKSI DATA KARTU VAKSIN BERBASIS WEB MENGGUNAKAN METODE OCR," May 2023. [Online]. Available: https://jurnal.umpar.ac.id/index.php/sylog

[7] A. Aprilino, "Implementasi Algoritma Yolo dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis," *Jurnal TeknoInfo*, vol. 16, 2022.

[8] A. Baddepudi, "7 examples of Gemini's multimodal capabilities in action," Gemini AI.

[9] A. Fadjeri, A. M. Asroriyah, and A. Rahmawati, "Analisis Teks Bahasa Indonesia Dan Inggris Dari Sebuah Citra Menggunakan Pengolahan Citra Digital," *Jurnal Teknologi Informasi dan Komunikasi (TIKomSiN)*, vol. 10, no. 2, p. 42, Nov. 2022, doi: 10.30646/tikomsin.v10i2.650.

[10] A. Fadjeri, B. A. Saputra, D. K. Adri Ariyanto, and L. Kurniatin, "Karakteristik Morfologi Tanaman Selada Menggunakan Pengolahan Citra Digital," *Jurnal Ilmiah SINUS*, vol. 20, no. 2, p. 1, Jul. 2022, doi: 10.30646/sinus.v20i2.601.

[11] Guntoro, "Metode Waterfall : Pengertian, Tahapan, Contoh, Kelebihan dan Kekurangan," BadoyStudio.

[12] D. Febri Kuncoro, U. Juniarti, J. Syahputra, R. Bagus, B. Sumantri, and R. Suryani, "Rancang Bangun Sistem Pengaduan Masyarakat Berbasis Web Dengan Metode Waterfall," 2022. [Online]. Available: www.journal.peradaban.ac.id

[13] M. Sumiati, R. Abdillah, and A. Cahyo, "Pemodelan UML untuk Sistem Informasi Persewaan Alat Pesta," *Jurnal Fasilkom*, vol. 11, Aug. 2021.

[14] W. Muthia Kansha, "Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application," *Jurnal Teknik Informatika STMIK Antar Bangsa*, Feb. 2023.

[15] O. M. A. AL-atraqchi, "A Proposed Model for Build a Secure Restful API to Connect between Server Side and Mobile Application Using Laravel Framework with Flutter Toolkits," *Cihan University-Erbil Scientific Journal*, vol. 6, no. 2, pp. 28–35, Aug. 2022, doi: 10.24086/cuesj.v6n2y2022.pp28-35.

[16] M. Nur Ichsanudin, M. Yusuf, S. Jurusan Rekayasa Sistem Komputer, J. Teknik Industri, I. AKPRIND Yogyakarta, and R. Artikel, "PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA INFO ARTIKEL ABSTRAK," vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123.