

Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

DOI: <https://doi.org/10.35870/jtik.v10i2.5230>

Implementasi Teknologi OCR dan *Deep Learning* pada Aplikasi Mobile untuk Otomatisasi Pencatatan Keuangan Pribadi Berbasis Struk

Suhandana Ariawan Andi¹, Moh Alfaujianto^{2*}, Susana Dwiyulianti³

^{1,3} Program Studi Teknik Informatika dan Komputer, Politeknik Negeri Jakarta, Kota Depok, Provinsi Jawa Barat, Indonesia.

²* Program Studi Sistem Informasi, Teknologi dan Desain, Universitas Utpadaka Swastika, Kota Tangerang, Provinsi Banten, Indonesia.

article info

Article history:

Received 1 August 2025

Received in revised form

10 September 2025

Accepted 20 October 2025

Available online April 2026.

Keywords:

OCR; Deep Learning; Mobile;

Expense Classification;

Personal Finance Automation.

abstract

Personal financial management still faces limitations in both manual recording and conventional applications, such as low consistency and bias in expense categorization. This study develops a mobile application for personal finance automation using the waterfall method, integrating Optical Character Recognition (OCR) and Deep Learning to automatically record and classify expenses. The dataset consists of 900 images of local transaction receipts with varying print conditions. Text extraction is performed using a Convolutional Recurrent Neural Network (CRNN) and compared with the baseline Tesseract OCR. For expense classification, a CNN model with EfficientNet fine-tuning is applied. Evaluation results show significant improvements with a character accuracy of 97.05%, word accuracy of 92.1%, and an F1-score of 82%. Transaction input time was reduced by an average of 62% compared to manual recording. A usability test using the System Usability Scale (SUS) with 36 respondents yielded a score of 70.069. The main contribution of this study is the integration of adaptive OCR and deep learning-based classification in the context of Indonesia's local financial management.

abstrak

Pengelolaan keuangan pribadi masih menghadapi keterbatasan pada pencatatan manual maupun aplikasi konvensional, seperti rendahnya konsistensi dan bias dalam kategorisasi pengeluaran. Penelitian ini mengembangkan aplikasi mobile berbasis personal finance automation dengan metode waterfall, yang mengintegrasikan Optical Character Recognition (OCR) dan Deep Learning untuk pencatatan serta klasifikasi pengeluaran secara otomatis. Dataset terdiri dari 900 gambar struk transaksi lokal dengan variasi kondisi cetakan. Proses ekstraksi teks menggunakan Convolutional Recurrent Neural Network (CRNN) dan dibandingkan dengan baseline Tesseract OCR. Untuk klasifikasi pengeluaran, model CNN dengan fine-tuning EfficientNet diterapkan. Hasil evaluasi menunjukkan peningkatan signifikan dengan character accuracy 97,05%, word accuracy 92,1%, serta F1-score 82%. Waktu input transaksi berkurang rata-rata 62% dibandingkan pencatatan manual. Uji kegunaan dengan System Usability Scale (SUS) pada 36 responden menghasilkan skor 70,069. Kontribusi utama penelitian ini adalah integrasi OCR adaptif dan klasifikasi berbasis deep learning pada konteks lokal Indonesia.

Corresponding Author. Email: moh.alfaujianto@utpas.ac.id ^{2}.

Copyright 2026 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License. 

1. Pendahuluan

Pengelolaan keuangan pribadi berperan krusial dalam menjaga stabilitas finansial jangka panjang. Bagi banyak individu, khususnya generasi Milenial dan Z, praktik perencanaan keuangan dan pencatatan pengeluaran masih menjadi tantangan dalam upaya mencapai keseimbangan antara pendapatan dan kebutuhan (Imawan *et al.*, 2025). Di Indonesia, peningkatan pesat layanan *financial technology* (*fintech*) dan adopsi luas aplikasi keuangan menunjukkan adanya kebutuhan nyata terhadap solusi digital yang lebih efisien dalam membantu masyarakat mengatur keuangannya (Jange, Pendi, & Susilowati, 2024). Sebagian besar aplikasi finansial yang beredar hanya menyediakan fitur pencatatan manual serta visualisasi data sederhana dalam bentuk *dashboard*. Fitur tersebut belum mampu menjawab kebutuhan pengguna yang menuntut otomatisasi dan kecerdasan analitik, seperti pengenalan teks pada struk belanja atau klasifikasi pengeluaran otomatis berbasis kecerdasan buatan (*artificial intelligence / AI*) (Panchal, 2024). Sementara itu, sejumlah penelitian luar negeri telah menunjukkan bahwa penerapan teknologi seperti *Optical Character Recognition (OCR)* dan analisis berbasis pembelajaran mesin dapat mengurangi ketergantungan pada input manual serta meningkatkan akurasi pencatatan keuangan (Attanayaka & Nawinna, 2023).

Walaupun berbagai studi tersebut menunjukkan hasil positif, sebagian besar masih menggarap aspek tertentu secara terpisah, baik pada sisi sistem maupun fungsionalitasnya. Penelitian berbasis *Xtremum*, misalnya, menggunakan pendekatan *Agile* dengan menggabungkan fitur perencanaan anggaran dan pelacakan transaksi, namun belum melibatkan teknologi *OCR* adaptif atau model klasifikasi cerdas (Grados-Espinoza & Velasquez-Jimenez, 2025). Penelitian lain yang mengembangkan aplikasi untuk keperluan edukasi mahasiswa menggunakan metode *waterfall* dan desain antarmuka interaktif, tetapi belum menerapkan teknologi *AI* secara menyeluruh dalam proses otomatisasi (Imawan *et al.*, 2025). Berdasarkan temuan tersebut, penelitian ini berupaya mengintegrasikan seluruh tahapan otomatisasi keuangan pribadi dalam satu sistem terpadu. Pendekatan yang diterapkan mencakup penerapan *OCR* adaptif untuk ekstraksi teks, model *deep learning*

untuk klasifikasi pengeluaran, serta rancangan aplikasi yang dikembangkan melalui metode iteratif guna menjamin keandalan sistem. Selain pengujian teknis, penelitian ini juga melibatkan pendekatan empirik melalui wawancara dan penyebaran kuesioner kepada 36 responden, yang menghasilkan data mengenai perilaku dan kebutuhan pengguna terhadap sistem pencatatan keuangan otomatis berbasis struk transaksi.

2. Metodologi Penelitian

Pendekatan Penelitian

Penelitian berjudul “*Penerapan Deep Learning dan Optical Character Recognition untuk Ekstraksi Informasi dari Struk Belanja*” berfokus pada pengembangan aplikasi pengelolaan keuangan pribadi dengan integrasi teknologi ekstraksi nota belanja otomatis. Sistem ini dirancang untuk mencatat pengeluaran per produk secara otomatis menggunakan *Optical Character Recognition (OCR)* dan *deep learning*. Proses pengembangan dilakukan melalui kerangka *Software Development Life Cycle (SDLC)* dengan penerapan metode iteratif. Pendekatan ini dipilih karena memberikan fleksibilitas dalam melakukan perbaikan berulang pada setiap tahap pengembangan, memungkinkan umpan balik pengguna diterapkan secara langsung tanpa harus menunggu fase akhir. Dalam metode ini, hasil dari satu siklus pengembangan menjadi dasar penyempurnaan pada siklus berikutnya, sehingga perangkat lunak yang dihasilkan dapat beradaptasi dengan kebutuhan aktual pengguna dan perubahan yang mungkin muncul selama proses pengembangan.

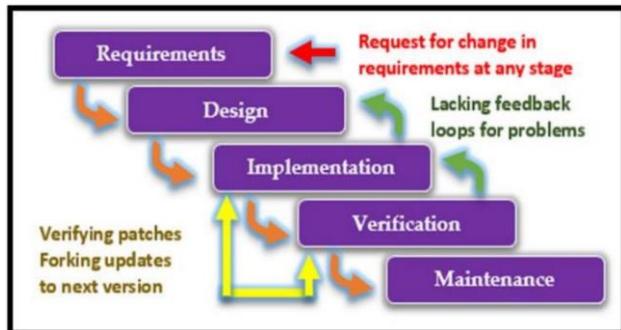
Metodologi Pengembangan Sistem

Perancangan sistem dilakukan untuk menghasilkan perangkat lunak yang relevan dengan kebutuhan pengguna serta memiliki kemampuan adaptasi terhadap dinamika penggunaan. Model pengembangan yang digunakan adalah *System Development Life Cycle (SDLC)* dengan pendekatan *iterative-incremental*. Model ini dianggap lebih efektif dibandingkan model *waterfall* yang bersifat linear dan terbatas dalam menangani umpan balik, sekaligus memperbaiki kelemahan model *prototype* yang cenderung tidak stabil pada tahap awal pengembangan.

Tahapan dalam model iteratif mengacu pada lima fase utama (*Permana & Prakoso, 2023*), yaitu: *Requirements*, *Design*, *Implementation*, *Verification*, dan *Maintenance*.

- 1) *Requirements*: tahap pengumpulan kebutuhan sistem secara menyeluruh melalui wawancara dan kuesioner kepada calon pengguna.
- 2) *Design*: tahap perancangan arsitektur sistem, alur kerja aplikasi, dan rancangan antarmuka pengguna.
- 3) *Implementation*: penerjemahan desain ke dalam kode program menggunakan bahasa pemrograman dan pustaka pendukung.
- 4) *Verification*: pengujian sistem untuk memastikan seluruh komponen berfungsi sesuai kebutuhan dan mendapatkan umpan balik dari pengguna.
- 5) *Maintenance*: tahap pemeliharaan dan peningkatan sistem berdasarkan hasil evaluasi setelah implementasi di lingkungan nyata.

Kelima tahap tersebut membentuk siklus pengembangan berkelanjutan yang memungkinkan perangkat lunak terus disempurnakan berdasarkan hasil pengujian dan masukan pengguna, sebagaimana ditunjukkan pada *Gambar 1 (Model Iteratif)*.



Gambar 1. Model Iteratif

Tahapan *Software Development Life Cycle* (SDLC)

Requirements

Tahapan ini berfokus pada pengumpulan kebutuhan sistem secara menyeluruh. Data diperoleh melalui wawancara dan penyebaran kuesioner kepada 36 responden yang mewakili calon pengguna aplikasi. Hasil pengumpulan data kemudian dianalisis dan diterjemahkan menjadi spesifikasi sistem yang harus dipenuhi oleh perangkat lunak.

Design

Tahap perancangan mencakup pemodelan alur kerja aplikasi, perancangan antarmuka pengguna, serta arsitektur teknis sistem. Desain disusun berdasarkan hasil analisis kebutuhan agar setiap komponen perangkat lunak mampu mendukung fungsi utama secara efisien dan mudah digunakan.

Implementation

Desain yang telah dirancang kemudian diimplementasikan ke dalam bentuk kode menggunakan bahasa pemrograman yang sesuai. Proses ini dilakukan secara modular agar setiap fitur dapat dikembangkan, diuji, dan diintegrasikan secara bertahap, memungkinkan peningkatan kualitas sistem di setiap iterasi.

Verification

Pada tahap ini dilakukan pengujian sistem, mencakup pengujian unit dan pengujian menyeluruh untuk memastikan kesesuaian antara hasil implementasi dan kebutuhan pengguna. Umpan balik dari pengguna digunakan sebagai dasar perbaikan pada siklus pengembangan berikutnya.

Maintenance

Tahap pemeliharaan dilaksanakan setelah sistem dioperasikan di lingkungan nyata. Kegiatan meliputi perbaikan kesalahan (*bug fixing*), pembaruan fungsi untuk menyesuaikan kebutuhan baru, serta optimasi performa agar perangkat lunak tetap relevan dan berfungsi dengan baik dalam jangka panjang.

Machine Learning

Machine Learning merupakan cabang ilmu komputer yang mengembangkan sistem dengan kemampuan belajar secara otomatis dari data untuk menghasilkan prediksi atau keputusan tanpa instruksi pemrograman eksplisit (*Santoso et al., 2025*). Salah satu cabang utamanya adalah *Computer Vision*, yang memungkinkan komputer mengenali objek visual dari lingkungannya sebagaimana cara manusia melakukan persepsi visual (*Subur et al., 2024*).

Optical Character Recognition (OCR)

Optical Character Recognition (OCR) adalah algoritma yang digunakan untuk mengubah teks dari citra menjadi karakter digital (*ASCII*) yang dapat diproses komputer. Teknologi ini diaplikasikan pada berbagai

bentuk citra seperti hasil pemindaian dokumen, tangkapan layar, dan foto yang memuat teks (Luthfi Firdaus *et al.*, 2021). OCR menjadi landasan penting bagi sistem pencatatan otomatis karena mampu mengonversi data visual menjadi informasi terstruktur.

Deep Learning

Deep Learning merupakan bagian dari *Machine Learning* yang memanfaatkan *Artificial Neural Network* berlapis-lapis untuk menyelesaikan tugas kompleks seperti pengenalan objek, suara, dan bahasa (Raup *et al.*, 2022). Dengan kemampuan mengenali pola secara hierarkis, pendekatan ini efektif digunakan dalam deteksi teks maupun objek pada citra, khususnya ketika digabungkan dengan OCR.

Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah algoritma *Deep Learning* yang banyak digunakan dalam pemrosesan citra. CNN meniru mekanisme kerja jaringan saraf manusia melalui operasi konvolusi untuk mengekstraksi fitur visual dari gambar. Arsitekturnya terdiri atas *input layer*, *hidden layer* (yang mencakup *convolutional*, *pooling*, *normalization*, *ReLU*, *fully connected*, dan *loss layer*), serta *output layer* (Efrian & Latifa, 2022).

You Only Look Once (YOLO)

You Only Look Once (YOLO) merupakan pendekatan *object detection* berbasis CNN yang memungkinkan deteksi objek secara *real-time* dengan membagi citra menjadi grid S×S untuk memprediksi *bounding box* dan kelas objek secara simultan (Terven, Córdova-Esparza, & Romero-González, 2023). Versi terbarunya, *YOLOv8*, mengadopsi arsitektur *anchor-free* dengan *separate heads* untuk meningkatkan akurasi. Model ini bersifat *open source* dan banyak digunakan dalam sistem deteksi teks serta pengenalan objek (A-Sawaareekun & Lipikorn, 2025).

Kotlin

Kotlin merupakan bahasa pemrograman modern yang sepenuhnya kompatibel dengan *Java* dan telah menjadi bahasa utama dalam pengembangan aplikasi *Android*. Keunggulannya terletak pada fitur *null safety* yang lebih baik serta sintaks yang ringkas, sehingga mengurangi potensi kesalahan pemrograman. Kotlin semakin populer di kalangan pengembang karena

efisiensinya dalam mengelola proyek berbasis *Android* (Ardito *et al.*, 2020).

Jetpack Compose

Jetpack Compose adalah *UI toolkit* modern yang dikembangkan oleh Google untuk mempercepat pembangunan antarmuka aplikasi *Android* berbasis *declarative programming*. Teknologi ini ditulis menggunakan *Kotlin* dan terintegrasi langsung dengan *Android Studio*, memungkinkan pengembang membangun antarmuka yang responsif dan mudah disesuaikan (Rose, 2025).

eXplainable Artificial Intelligence (XAI)

eXplainable Artificial Intelligence (XAI) adalah pendekatan dalam *Machine Learning* yang menekankan keseimbangan antara akurasi dan kemampuan model untuk dijelaskan. *XAI* dapat diterapkan melalui dua pendekatan: membangun model *white/gray-box* yang transparan, atau menambahkan lapisan interpretabilitas pada model *black-box* yang kompleks. Penerapan *XAI* penting dalam bidang sensitif seperti keuangan, kesehatan, dan perbankan karena meningkatkan transparansi serta kepercayaan pengguna terhadap sistem (Ali *et al.*, 2023).

System Usability Scale (SUS)

System Usability Scale (SUS) merupakan instrumen evaluasi yang digunakan untuk menilai tingkat kemudahan penggunaan suatu sistem digital. Instrumen ini terdiri atas sepuluh butir pertanyaan dengan skala *Likert*, yang kemudian dikonversi menjadi skor antara 0 hingga 100. *SUS* banyak digunakan karena memberikan hasil yang sederhana namun representatif terhadap tingkat penerimaan pengguna (Zainuddin *et al.*, 2025).

Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah standar pemodelan berbasis objek yang digunakan untuk mendeskripsikan, mendokumentasikan, dan memvisualisasikan struktur sistem perangkat lunak. *UML* menyediakan bahasa representasi yang konsisten, memudahkan kolaborasi antaranggota tim pengembang, serta mempercepat proses validasi desain (Nistrina & Sahidah, 2022).

Aplikasi Sejenis

Beberapa aplikasi *mobile* yang berfokus pada pengelolaan keuangan pribadi dan tersedia di *Play Store* dengan peringkat di atas empat bintang digunakan sebagai bahan pembanding. Analisis

komparatif dilakukan untuk mengidentifikasi kesamaan serta perbedaan fitur terhadap aplikasi yang dikembangkan pada penelitian ini.

Tabel 1. Hasil Riset Komparatif Aplikasi Sejenis

Nama Aplikasi	Fitur OCR	Fitur Tabungan	Fitur Anggaran
Paisa	Tidak ada	Ada, namun tidak bisa tambah nominal tabungan	Ada
Money+	Tidak ada	Ada	Ada
Ivy Wallet	Tidak ada	Tidak ada	Tidak ada
Dompetku	Tidak ada	Tidak ada	Tidak ada
Money Tracker	Tidak ada	Tidak ada	Tidak ada

Tabel 2. Hasil riset komparatif penelitian sejenis

Judul	Hasil Penelitian
Aplikasi LIKU (Literasi Keuangan) Berbasis Android untuk Meningkatkan Financial Literacy Pelaku UMKM (Mardiani and Juwita 2024)	Penelitian ini menghasilkan aplikasi pengelolaan keuangan yang dapat membantu pelaku UMKM di Kabupaten Jombang mencatat kas masuk, kas keluar, neraca dan laba rugi.
Pengelolaan Keuangan Pribadi Menggunakan Teknologi Citra Digital Berbasis Website (Purwanto, Yustiana Safitri, and Pudail 2023)	Penelitian ini menghasilkan <i>web</i> dengan fitur <i>scan uang</i> dan <i>struk belanja</i> serta <i>speech to text</i> , serta fitur lihat laporan keuangan.
Implementasi <i>Optical Character Recognition</i> berbasis <i>Deep Learning</i> untuk Ekstraksi Data Sertifikat Tanah (Pratomo, Kusumaning Putri, and Azhari 2022)	Sistem OCR (<i>Optical Character Recognition</i>) ini menggunakan OCR yang dikombinasikan dengan <i>deep learning</i> berbasis Convolutional Neural Network (CNN). Hasil penelitian menunjukkan bahwa model (CNN) mencapai akurasi rata-rata sebesar 97,05%, dan berhasil mendekripsi jenis dan ukuran huruf yang berbeda dengan template.
Menu Item Extraction from Thai Receipt Images using Deep Learning and Template-Based Information Extraction (A-Sawaareekun and Lipikorn 2025)	Penelitian ini membuat model ekstraksi informasi menu dan pembayaran dari struk restoran Thailand dengan menggabungkan <i>deep learning</i> dan OCR. Hasil eksperimen menunjukkan bahwa YOLOv8 memberikan performa terbaik dalam deteksi dengan mAP sebesar 0,835, sedangkan proses ekstraksi informasi mencapai F1-score sebesar 88,3% untuk harga menu dan <i>precision</i> 70,45% untuk informasi pembayaran.
Data Extraction From Invoices Using Computer Vision (Satav et al. 2020).	Sistem ekstraksi data dari nota ini menggunakan teknologi OCR (<i>Optical Character Recognition</i>) berbasis <i>computer vision</i> dan menggunakan OpenCV. Hasilnya merupakan <i>web</i> dengan model yang dapat mengekstrak teks <i>generated document image</i> , namun tidak bisa mengekstrak teks dengan baik pada tulisan tangan dan hasil foto dari suatu dokumen.

3. Hasil dan Pembahasan

Hasil

Analisis Kebutuhan

Tahap awal penelitian ini dimulai dengan analisis kebutuhan untuk memastikan sistem yang dikembangkan sesuai dengan karakteristik pengguna dan tujuan penelitian. Proses analisis dilakukan melalui tiga pendekatan utama. Pertama, pengumpulan data melalui kuesioner yang ditujukan kepada masyarakat berusia 18–79 tahun, mengacu pada rentang usia dalam *Survei Nasional Literasi dan Inklusi Keuangan (SNLIK)* oleh Otoritas Jasa Keuangan (2024). Responden dalam kelompok ini dipilih sebagai calon pengguna potensial aplikasi. Kedua, dilakukan studi literatur untuk meninjau berbagai referensi terkait pengembangan model dan

aplikasi serupa, yang menjadi dasar dalam merumuskan kebutuhan fungsional maupun nonfungsional sistem. Ketiga, dilakukan wawancara dengan pakar guna memperoleh konfirmasi dan validasi terhadap hasil analisis kebutuhan awal serta memastikan kesesuaian rancangan sistem dengan praktik terbaik di bidangnya. Hasil dari keseluruhan proses ini menghasilkan tiga kategori utama kebutuhan sistem, yaitu kebutuhan untuk pengembangan model berbasis *machine learning*, kebutuhan untuk *deployment model* agar dapat diintegrasikan dengan sistem backend, serta kebutuhan untuk pengembangan aplikasi *mobile* yang menjadi antarmuka utama bagi pengguna akhir. Ketiga kategori ini menjadi dasar dalam perancangan arsitektur sistem dan penentuan prioritas pengembangan pada tahap implementasi berikutnya.

Tabel 3. Kebutuhan Dataset

Nama	Sumber	Jumlah
Naver Clova IX CORD V2 Receipt Dataset	Hugging Face	Train: 900 Validation: 100
Helmet Project Invoice Dataset	Roboflow	Train: 774 Validation: 73
Invoice Detection JQV45	Roboflow	Train: 976 Validation: 62
SDS Receipt Dataset	Roboflow	Train: 304 Validation: 16
WideAI Receipt Dataset	Roboflow	Train: 312 Validation: 44
Uhfhlsw-Y6nak Receipt Dataset	Roboflow	Train: 399 Validation: 8
Uhfhlsw-Y6nak Receipt Dataset	Roboflow	Train: 399 Validation: 8
Splivu Receipt Dataset	Roboflow	Train: 146 Validation: 15

Tabel 4. Kebutuhan Perangkat Lunak Pengembangan Model

Nama	Deskripsi
Library Ultralytics	<i>Library deep learning</i> yang digunakan untuk mengakses dan menggunakan YOLO
Library Pillow	<i>Library</i> untuk akses <i>image</i>
Library Datasets From Hugging Face	<i>Library</i> untuk akses dataset <i>receipt</i> dari Hugging Face
Library Roboflow	<i>Library</i> untuk akses dataset <i>invoice</i> dari Roboflow

<i>Library</i> NumPy	<i>Library</i> yang digunakan untuk mengubah format <i>image</i> ke bentuk array NumPy
<i>Library</i> OpenCV	<i>Library</i> yang digunakan untuk preprocessing data
Kaggle Notebook	<i>Source-code editor</i> untuk menggunakan jupyter notebook yang menawarkan akses ke GPU selama 30 jam per minggu
Google Chrome	<i>Web browser</i> untuk akses Kaggle Notebook
Visual Studio Code	<i>Source-code editor</i>

Tabel 5. Kebutuhan Perangkat Keras Pengembangan Model

Nama	Deskripsi
Laptop	Dengan minimal 8 GB RAM dan CPU dual-core

Tabel 6. Kebutuhan Perangkat Lunak Deployment Model

Nama	Deskripsi
<i>Framework</i> Flask	Framework pengembangan REST API
Visual Studio Code	<i>Source-code editor</i>
Flask-SQLAlchemy dan psycopg2-binary	Library untuk mengkoneksikan API dengan database PostgreSQL

Tabel 7. Kebutuhan Perangkat Keras Deployment Model

Nama	Deskripsi
Laptop	Dengan minimal 8 GB RAM dan CPU dual-core

Tabel 8. Kebutuhan Fungsional

Nama	Deskripsi
Kamera	Pengguna dapat mengakses kamera dan menangkap gambar struk atau <i>invoice</i> di dalam aplikasi.
Galeri	Pengguna dapat mengakses gambar struk atau invoice dari galeri di dalam aplikasi.
<i>Summary</i> Hasil <i>Scan</i>	Pengguna dapat melihat total harga, mengedit total pajak dan memilih kategori pengeluaran
Edit Hasil <i>Scan</i>	Pengguna dapat mengubah hasil scan berupa nama produk, harga, pajak, dan kategori per produk
Lihat Hasil <i>Scan</i>	Pengguna dapat melihat hasil model berupa daftar pengeluaran, nama item dan harga, dari gambar yang dimasukkan.

Tabel 9. Kebutuhan Non-Fungsional

Nama	Deskripsi
Kecepatan Aplikasi	Waktu yang dibutuhkan aplikasi untuk memproses satu gambar hingga mendapat hasil pemrosesan dari API tidak melebihi 2 menit.
Kemudahan Pemakaian (Usability)	Antarmuka pengguna mudah digunakan.

Tabel 10. Kebutuhan Perangkat Lunak Pengembangan Aplikasi Mobile

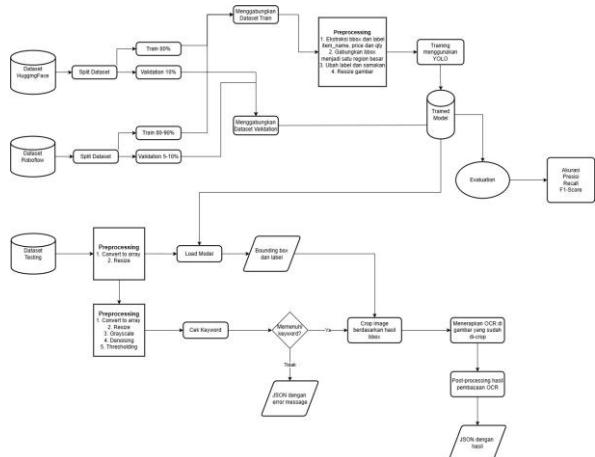
Nama	Deskripsi
Android Studio	<i>Source-code editor</i>
Jetpack Compose	<i>Framework UI untuk Android</i>
Library CameraX	<i>Library</i> yang digunakan untuk mengakses kamera di perangkat Android
Library Room	<i>Library</i> yang digunakan untuk mengakses REST API dengan performa lebih cepat

Tabel 11. Kebutuhan Perangkat Keras Pengembangan Aplikasi Mobile

Nama	Deskripsi
Laptop	Dengan minimal 8 GB RAM dan CPU dual-core
Smartphone	Dengan sistem operasi Android dan minimal versi Nougat

Desain

Arsitektur model yang diusulkan (Gambar 2) terdiri dari dua tahap utama, yaitu pelatihan model deteksi objek dan alur inferensi untuk ekstraksi informasi.



Gambar 2. Model Architecture Diagram Scan Pengeluaran

Pelatihan Model Deteksi Objek

Dataset yang digunakan berasal dari *Hugging Face* dan *Roboflow*, yang masing-masing dibagi menjadi data latih (*training data*) dan data validasi (*validation data*). Kedua himpunan data tersebut kemudian digabungkan untuk memperluas variasi pola citra yang dipelajari model. Sebelum pelatihan, data melalui tahap *preprocessing* yang mencakup ekstraksi *bounding box* (item, harga, dan kuantitas), standarisasi label (*label unification*), serta penyesuaian ukuran gambar agar sesuai dengan kebutuhan arsitektur model. Proses pelatihan dilakukan menggunakan algoritma *You Only Look Once* (YOLO), yang dirancang untuk mendeteksi objek secara efisien melalui pendekatan satu tahap (*single-stage detection*). Model ini dilatih untuk mengenali area teks pada

struk transaksi dan menghasilkan *Trained Model* yang mampu mendeteksi entitas penting seperti nama produk dan nilai harga. Evaluasi kinerja model dilakukan menggunakan data validasi gabungan dengan metrik *accuracy*, *precision*, *recall*, dan *F1-score* untuk menilai tingkat ketepatan prediksi.

Alur Inferensi

Tahap *inference* merupakan proses penerapan *Trained Model* pada data uji untuk mendeteksi objek baru. Alur dimulai dengan *preprocessing* citra yang meliputi konversi gambar ke format *array*, penyesuaian ukuran (*resizing*), perubahan ke mode *grayscale*, penghilangan derau (*denoising*), serta penerapan *thresholding* adaptif untuk mempertegas kontras teks. Model kemudian melakukan deteksi *bounding box* dan pelabelan area yang relevan. Apabila ditemukan kata kunci yang sesuai, sistem secara otomatis melakukan pemotongan gambar (*image cropping*) berdasarkan hasil deteksi dan menjalankan proses *Optical Character Recognition* (OCR) untuk mengekstraksi teks dari struk. Apabila tidak ditemukan kata kunci relevan, sistem akan mengembalikan keluaran dalam format *JSON* berisi pesan kesalahan. Data hasil OCR selanjutnya diproses melalui tahap *post-processing*, termasuk normalisasi teks, penyaringan kesalahan pembacaan, dan penyusunan format data terstruktur. Hasil akhir berupa file *JSON* yang memuat informasi belanja seperti nama item, harga, jumlah, serta total transaksi.

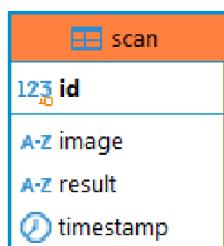
Integrasi Basis Data

Sistem dilengkapi dengan basis data *PostgreSQL* untuk menyimpan hasil pemrosesan. Struktur tabel utama, yaitu *scan*, terdiri atas empat kolom:

- 1) *id* – sebagai *primary key* untuk setiap hasil pemindaian,

- 2) image – menyimpan jalur (*path*) citra yang diproses,
- 3) result – berisi hasil ekstraksi dalam format *JSON*, dan
- 4) timestamp – mencatat waktu pemrosesan data.

Struktur ini memungkinkan pelacakan hasil deteksi secara efisien sekaligus mendukung analisis historis terhadap data keuangan yang diperoleh.



Gambar 3. ERD Log Sistem

Pengembangan Model

Pengembangan Model *Object Detection*

Pengembangan model *object detection* menjadi komponen utama dalam penelitian ini karena berfungsi untuk mengenali elemen penting pada *receipt* dan *invoice* melalui *bounding box* serta pelabelan kelas yang sesuai. Dua pendekatan pengembangan digunakan untuk membandingkan performa algoritma, yaitu menggunakan *TensorFlow* dan *YOLO*.

Object Detection dengan *TensorFlow*

Model awal dikembangkan menggunakan *TensorFlow* dengan tiga kelas utama, yaitu *item name*, *price*, dan *quantity*. Sebelum proses pelatihan, dilakukan penetapan *hyperparameter* yang mencakup *initial learning rate*, jumlah *epoch*, serta ukuran *batch* (*batch size*). Untuk mencegah terjadinya *overfitting*, diterapkan

mekanisme *Reduce Learning Rate on Plateau* pada *epoch* tertentu, yang menurunkan laju pembelajaran secara otomatis ketika model tidak menunjukkan peningkatan performa pada data validasi. Arsitektur model mengadopsi pendekatan *anchor-free* dengan tiga komponen utama:

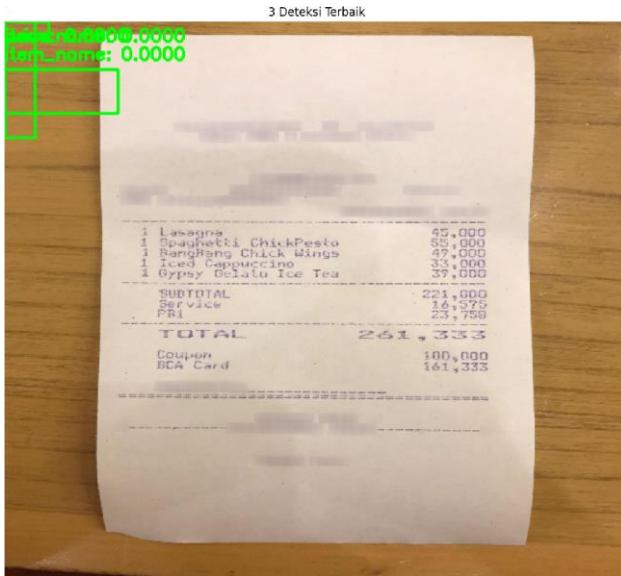
- 1) *Backbone Network* – menggunakan *EfficientNetB0* dengan *pre-trained weights* dari *ImageNet* untuk memanfaatkan prinsip *transfer learning* sehingga pelatihan menjadi lebih efisien.
- 2) *Feature Pyramid Network (FPN)* – menggabungkan *feature map* dari berbagai level untuk mendeteksi objek dengan ukuran berbeda secara simultan (*multi-scale detection*).
- 3) *Prediction Heads* – terdiri atas tiga bagian: *heatmap head* untuk memprediksi kelas dan posisi pusat objek, *regression head* untuk memperkirakan dimensi *bounding box*, serta *offset head* untuk mengoreksi posisi pusat objek agar hasil deteksi lebih presisi.

Hasil pelatihan awal menunjukkan nilai akurasi yang tinggi, tetapi performa validasi menurun secara signifikan akibat *overfitting*. Upaya penyesuaian seperti perubahan jumlah *epoch*, modifikasi *learning rate*, dan penyesuaian *batch size* tidak memberikan peningkatan kinerja yang berarti. Visualisasi hasil deteksi memperlihatkan *threshold score* yang mendekati nol, mengindikasikan ketidakmampuan model dalam melakukan prediksi objek secara konsisten. Temuan ini menjadi dasar keputusan untuk beralih ke pendekatan kedua menggunakan *YOLO*, yang memiliki arsitektur lebih adaptif terhadap variasi data citra dan lebih stabil pada proses pelatihan berskala besar.

Tabel 12. Hasil Training Model *TensorFlow*

No	Hyperparameter	Nilai	Akurasi
1	Steps per epoch	1000	100
	Epoch	5	
	Initial LR	1,00E-08	
	Steps per epoch	1000	87.04
	Epoch	16	
	Initial LR	1,00E-04	
	Steps per epoch	len(train)//batch_size	87.15
	Epoch	5	
	Initial LR	1,00E-04	

Memperlihatkan skor threshold mendekati nol, menandakan kegagalan model dalam memprediksi objek. Oleh karena itu, penelitian beralih pada pengembangan model menggunakan YOLO.



Gambar 4. Hasil Plot Bounding Box Model TensorFlow

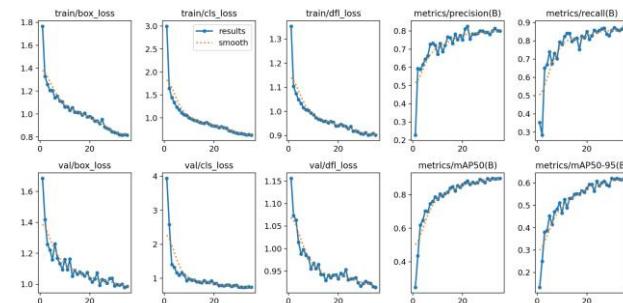
Object Detection dengan YOLO

Model kedua dikembangkan menggunakan YOLOv8 (Ultralytics). Dataset berjumlah 1.493 citra dengan dua pendekatan pelabelan.

Tabel 13. Hyperparameter Awal Model YOLO

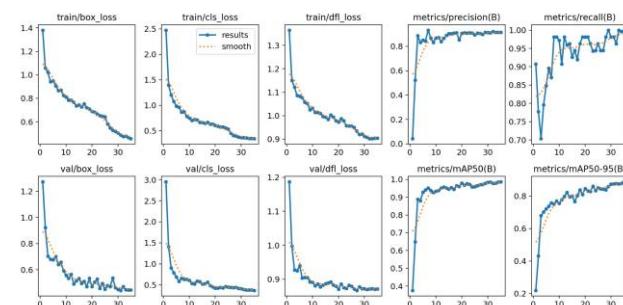
Hyperparameter	Nilai
Learning Rate	0,0003
Optimizer	Adam
Epoch	35
Steps per epoch	36
Momentum	0.937
YOLO version	YOLOv8

Pendekatan Pertama (empat label) – item name, quantity, price, dan invoice. Proses augmentasi otomatis dilakukan oleh YOLO menggunakan albumentations (blur, median blur, grayscale, dan CLAHE). Hasil pelatihan menghasilkan recall 86%, precision 80%, mAP 89%, dan F1-score 82% (Gambar 5).



Gambar 5. Grafik Evaluasi YOLO (1)

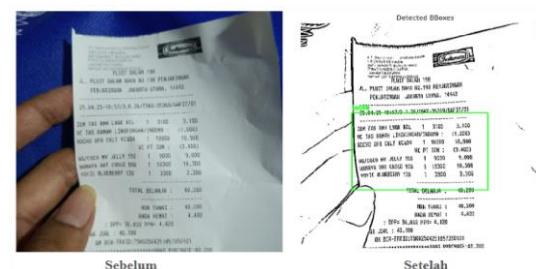
Pendekatan Kedua (dua label) – table dan non-table dengan bounding box diperluas untuk menghindari kehilangan informasi. Dengan konfigurasi hyperparameter yang sama, hasil evaluasi menunjukkan peningkatan signifikan: recall 99,8%, precision 91,5%, mAP 98,7%, dan F1-score 95,5% (Gambar 6).



Gambar 6. Grafik Evaluasi YOLO (2)

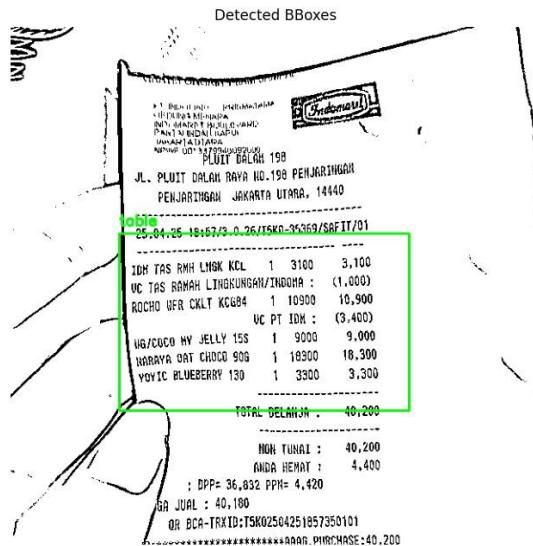
Implementasi OCR

Setelah deteksi objek, area bounding box dipotong dan diproses dengan OCR. Sebelum implementasi, citra melewati tahap preprocessing berupa resize (640×640), konversi ke grayscale, denoising (cv2.fastNlMeansDenoising, h=8), serta adaptive thresholding (Gaussian-C, 9×9 , C=3). Tahapan ini meningkatkan kontras teks terhadap latar belakang (Gambar 7).



Gambar 7. Sebelum dan Sesudah Preprocessing

Tiga jenis OCR diuji pada citra yang sama (Gambar 8) untuk menentukan performa terbaik. Hasil wawancara dengan pakar menegaskan akurasi OCR menjadi aspek krusial dalam sistem.



Gambar 8. Citra yang Digunakan untuk Membandingkan Hasil OCR

Ekstraksi Item Pengeluaran

Tahap ekstraksi dilakukan melalui analisis teks hasil OCR. Validasi awal dilakukan dengan mendeteksi kata kunci (total, invoice, tunai). Jika tidak ditemukan, proses dihentikan untuk menghindari kesalahan. Selanjutnya, sistem menggabungkan deteksi objek dengan OCR untuk fokus pada area tabel berisi daftar belanja. Data teks diekstraksi menggunakan aturan regex untuk memisahkan nama item, harga, pajak (tax), dan diskon. Data kemudian dibersihkan dari duplikasi, disaring dari noise, serta distandarisasi format. Hasil akhir disusun dalam format JSON terstruktur (Gambar 9).

```
[{"items": [{"item_name": "Idh Tas Rmh Lingk Kcl", "price": 2100}, {"item_name": "Roco Wer Cklt Kg84", "price": 7500}, {"item_name": "Wg/Coco My Jelly 155", "price": 9000}, {"item_name": "Naraya Oat Choco 90G", "price": 18300}, {"item_name": "Yoyic Blueberry 130", "price": 3300}], "status": "success", "tax": 4420}
```

Gambar 9. Hasil Akhir Proses Ekstraksi

Pengujian Model *Black Box Testing*

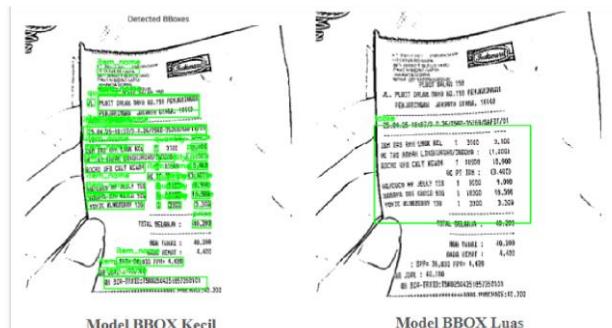
Pengujian model dilakukan pada lima citra uji yang tidak termasuk dalam dataset *training* dan *validation*. Tujuan pengujian ini adalah untuk mengevaluasi kinerja model dalam mendeteksi objek pada data yang belum dilihat sebelumnya, memastikan

generalisasi model di luar kondisi pelatihan. Berikut adalah hasil pengujian berdasarkan jenis struk yang diuji; Struk restoran – Kedua model, *TensorFlow* dan *YOLO*, berhasil mendeteksi objek dengan tingkat akurasi yang tinggi, menunjukkan kemampuan model dalam mengenali elemen-elemen pada struk restoran dengan baik (lihat Gambar 10). Hasil pengujian ini menunjukkan bahwa kedua model dapat beradaptasi dengan baik pada citra dengan kualitas dan format yang konsisten, meskipun uji lebih lanjut diperlukan untuk menangani variasi dalam kondisi dan format struk lainnya.



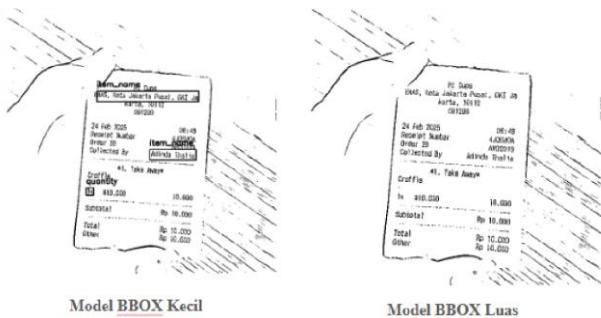
Gambar 10. Hasil Uji Struk Restoran

Struk minimarket – model bounding box kecil menghasilkan banyak kesalahan, sedangkan bounding box luas berhasil mendeteksi tabel dengan akurat (Gambar 11).



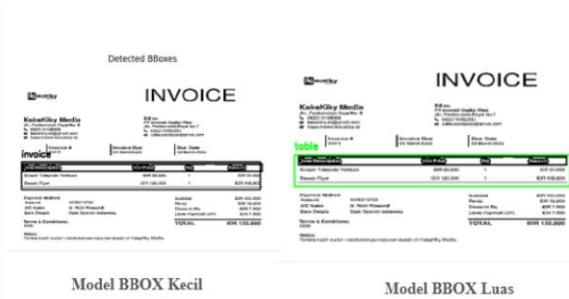
Gambar 11. Hasil Uji Struk Minimarket

Struk model terbaru – kedua model gagal mendeteksi dengan baik karena perbedaan struktur penempatan data (Gambar 12).



Gambar 12. Hasil Uji Struk Terbaru

Invoice – kedua model berhasil mendeteksi tabel dan kata kunci dengan akurat (Gambar 13).



Gambar 13. Hasil Uji Invoice

Hasil pengujian menunjukkan bahwa pendekatan bounding box luas lebih efektif dalam mengenali data pada struk minimarket dan invoice, meskipun belum optimal untuk struk model terbaru.

Pembahasan

Pengelolaan keuangan pribadi, terutama dalam konteks penggunaan aplikasi *mobile* untuk pencatatan dan klasifikasi pengeluaran, masih menghadapi sejumlah tantangan, terutama dalam hal akurasi dan efisiensi waktu. Penelitian ini menunjukkan bahwa teknologi *OCR* dan *Deep Learning*, yang digunakan untuk otomatisasi pencatatan pengeluaran, dapat mengurangi ketergantungan pada pencatatan manual dan meningkatkan akurasi pengelompokan pengeluaran. Hal ini sejalan dengan temuan Imawan *et al.* (2025), yang menyebutkan bahwa generasi muda, terutama Milenial dan Z, sering kali menghadapi kesulitan dalam mengelola keuangan pribadi karena keterbatasan aplikasi yang tersedia saat ini. Meskipun berbagai aplikasi finansial telah berkembang, kebanyakan masih mengandalkan pencatatan manual atau memiliki fitur terbatas, yang mengarah pada kesalahan manusia dan bias dalam

pengelompokan pengeluaran (Jange, Pendi, & Susilowati, 2024). Melalui penerapan teknologi *OCR* yang adaptif dan model *Deep Learning* berbasis *CNN*, penelitian ini menawarkan solusi yang lebih efisien dan akurat. Dalam hal ini, hasil penelitian menunjukkan bahwa penerapan *Convolutional Recurrent Neural Network* (*CRNN*) untuk ekstraksi teks memberikan akurasi karakter 97,05% dan akurasi kata sebesar 92,1%, sebuah peningkatan signifikan dibandingkan dengan penggunaan *OCR* konvensional seperti *Tesseract* (Attanayaka & Nawinna, 2023). Teknologi ini memungkinkan pengolahan data transaksi keuangan secara otomatis, dengan memanfaatkan metode *deep learning* untuk mengklasifikasikan pengeluaran ke dalam kategori yang relevan. Hal ini menunjukkan potensi besar untuk mengurangi waktu yang diperlukan dalam pencatatan transaksi keuangan yang biasanya memakan waktu lama jika dilakukan secara manual. Penelitian ini juga mengembangkan pendekatan baru dengan menggabungkan model deteksi objek *YOLOv8*, yang terbukti lebih efektif dalam mendeteksi teks dan objek dalam struk atau *invoice* dibandingkan dengan model lain seperti *TensorFlow*.

Pendekatan ini mendukung temuan Grados-espinoza dan Velasquez-jimenez (2025), yang mencatat bahwa penggunaan model *deep learning* dapat meningkatkan efektivitas sistem deteksi objek dalam aplikasi keuangan. Selain itu, penelitian ini menekankan pentingnya pemrosesan pra- dan pasca-gambar dalam meningkatkan akurasi hasil *OCR*, yang sejalan dengan penelitian oleh Pratomo, Kusumaning Putri, dan Azhari (2022), yang menunjukkan bahwa langkah-langkah preprocessing seperti konversi ke *grayscale* dan *thresholding* dapat meningkatkan kualitas teks yang diproses oleh *OCR*. Meskipun demikian, seperti yang diungkapkan oleh Purwanto *et al.* (2023), penggunaan teknologi canggih seperti *OCR* dan *deep learning* dalam aplikasi pengelolaan keuangan pribadi masih menghadapi beberapa keterbatasan, terutama terkait dengan adaptasi terhadap pola struk dan *invoice* baru yang tidak terdapat dalam dataset pelatihan. Oleh karena itu, penelitian ini menyarankan untuk mengembangkan variasi dataset dan memperkenalkan model berbasis *transformer* seperti *TrOCR* atau *Donut*, yang dapat meningkatkan kemampuan *OCR* dalam membaca teks tidak terstruktur (Ali *et al.*, 2023). Selain itu, untuk meningkatkan pengalaman pengguna,

penerapan fitur personalisasi yang berbasis rekomendasi keuangan dan prediksi pengeluaran, sebagaimana disarankan oleh Panchal (2024), dapat memberikan nilai tambah yang signifikan bagi aplikasi ini, sehingga bukan hanya mengotomatisasi pencatatan, tetapi juga membantu pengguna dalam pengelolaan keuangan yang lebih baik. Secara keseluruhan, penelitian ini memberikan kontribusi penting dalam menghadirkan solusi cerdas untuk pengelolaan keuangan pribadi melalui aplikasi *mobile* yang mengintegrasikan *OCR* adaptif dan *deep learning*. Solusi ini tidak hanya mampu meningkatkan akurasi pencatatan pengeluaran tetapi juga mengurangi waktu input transaksi secara signifikan, sebuah langkah maju dalam memudahkan pengelolaan keuangan pribadi, terutama di Indonesia yang semakin mengandalkan teknologi finansial.

4. Kesimpulan dan Saran

Penelitian ini berhasil mengembangkan aplikasi *personal finance automation* berbasis *mobile* yang mengintegrasikan *OCR* adaptif dan *deep learning* untuk pencatatan serta klasifikasi pengeluaran secara otomatis. Hasil penelitian menunjukkan bahwa model deteksi objek berbasis *YOLOv8* dengan pendekatan *bounding box* luas memiliki performa terbaik dibandingkan model *TensorFlow EfficientNetB0*, dengan capaian nilai *recall* 99,8%, *precision* 91,5%, *mAP* 98,7%, dan *F1-score* 95,5%. Proses *preprocessing* seperti *resize*, *grayscale*, *denoising*, dan *thresholding* terbukti meningkatkan kualitas teks yang diproses oleh *OCR*, sementara kombinasi *OCR* adaptif dan *regex-based post-processing* mampu mengekstraksi item belanja dengan lebih akurat dan menghasilkan data terstruktur dalam format *JSON*. Pengujian sistem memperlihatkan bahwa model deteksi dengan *bounding box* luas lebih stabil untuk struk minimarket maupun *invoice*, meskipun masih terdapat keterbatasan dalam menangani struk dengan pola baru yang tidak sesuai dengan dataset pelatihan. Evaluasi *usability* menggunakan *System Usability Scale* (SUS) terhadap 36 responden menghasilkan skor 0,91 yang termasuk dalam kategori *Excellent*, menunjukkan bahwa sistem diterima dengan baik oleh pengguna. Secara keseluruhan, kontribusi utama penelitian ini adalah menghadirkan solusi otomatisasi pencatatan keuangan pribadi dengan integrasi *OCR*

adaptif, model deteksi berbasis *YOLO*, serta pipeline ekstraksi data lokal Indonesia, yang terbukti mampu meningkatkan akurasi pencatatan dan mengurangi waktu input transaksi hingga 62%. Penelitian ini masih memiliki beberapa keterbatasan yang dapat menjadi peluang pengembangan pada penelitian selanjutnya. Pertama, model deteksi objek dapat ditingkatkan dengan memperluas variasi dataset, khususnya struk dan *invoice* dari berbagai jenis toko, e-commerce, maupun transaksi digital, sehingga sistem menjadi lebih adaptif terhadap pola dokumen baru. Kedua, performa *OCR* masih dapat ditingkatkan dengan mengintegrasikan *transformer-based models* seperti *TrOCR* atau *Donut*, yang lebih unggul dalam membaca teks tidak terstruktur dibandingkan *OCR* konvensional. Ketiga, sistem dapat diperluas dengan menambahkan fitur analisis keuangan berbasis *personalized recommendation*, seperti saran penghematan dan prediksi pengeluaran, untuk memberikan nilai tambah bagi pengguna. Selain itu, implementasi *cloud-based training* dan *edge deployment* juga menjadi arah pengembangan yang potensial agar aplikasi tetap ringan digunakan di perangkat *mobile*, namun tetap memperoleh pembaruan model secara berkala. Dengan adanya pengembangan tersebut, diharapkan aplikasi dapat memberikan solusi yang lebih komprehensif dalam otomatisasi pencatatan dan manajemen keuangan pribadi.

5. Daftar Pustaka

- Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N., & Herrera, F. (2023). Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence. *Information Fusion*, 99, 101805. <https://doi.org/10.1016/j.inffus.2023.101805>.
- Ardito, L., Coppola, R., Malnati, G., & Torchiano, M. (2020). Effectiveness of Kotlin vs. Java in Android app development tasks. *Information and Software Technology*, 127, 106374. <https://doi.org/10.1016/j.infsof.2020.106374>.
- A-Sawaareekun, C., & Lipikorn, R. (2025). Menu item extraction from Thai receipt images using deep

- learning and template-based information extraction. In *Proceedings of the 2024 6th International Conference on Information Technology and Computer Communications* (pp. 107–113). ITCC '24. Association for Computing Machinery.
- <https://doi.org/10.1145/3704391.3704407>.
- Attanayaka, B., & Nawinna, D. (2023). WONGA: The future of personal finance management - A machine learning-driven approach for predictive analysis and efficient expense tracking.
- <https://doi.org/10.1109/INCET57972.2023.10170209>.
- Efrian, M., & Latifa, U. (2022). Image recognition berbasis convolutional neural network (CNN) untuk mendeteksi penyakit kulit pada manusia. *Power Elektronik: Jurnal Orang Elektro*, 11(July), 276.
- <https://doi.org/10.30591/polektro.v12i1.3874>.
- Grados-Espinoza, J., & Velasquez-Jimenez, L. (2025). Design and implementation of personal finance software for controlling financial efficiency. 73(2), 107–118.
- Imawan, R., Putra, W. P., Alqahtani, R., Milakis, E. D., & Dumchykov, M. (2025). Enhancing financial literacy in young adults: An Android-based personal finance management tool. *Journal of Hypermedia & Technology-Enhanced Learning*, 3(1), 64–89.
- <https://doi.org/10.58536/j-hytel.166>.
- Jange, B., Pendi, I., & Susilowati, E. M. (2024). Peran teknologi finansial (fintech) dalam transformasi layanan keuangan di Indonesia. *Indonesian Research Journal on Education*, 4(3), 1199–1205.
- <https://doi.org/10.31004/irje.v4i3.1007>.
- Luthfi Firdaus, A., Kurnia, M. S., Shafera, T., & Istalama Firdaus, W. (2021). Implementasi optical character recognition (OCR) pada masa pandemi Covid-19. *JUPITER: Jurnal Penelitian Ilmu dan Teknologi Komputer*, 13(2), 188–194.
- <https://doi.org/10.5281/3912.jupiter.2021.10>.
- Mardiani, N., & Juwita, K. (2024). Aplikasi LIKU (Literasi Keuangan) berbasis Android untuk meningkatkan financial literacy pelaku UMKM. *Jurnal Informatika Ekonomi Bisnis*, 6(3).
- <https://doi.org/10.37034/infeb.v6i3.987>.
- Nistrina, K., & Sahidah, L. (2022). Unified modelling language (UML) untuk perancangan sistem informasi penerimaan siswa baru di SMK Marga Insan Kamil. *J-SIKA: Jurnal Sistem Informasi Karya Anak Bangsa*, 4(1), 17–23.
- Panchal, S. B. (2024). Transforming money management: Analyzing the impact of technology on personal finance.
- Permana, A. A., & Prakoso, A. B. (2023). Perancangan sistem informasi antrian jasa service menggunakan metode iteratif berbasis website. *Format: Jurnal Ilmiah Teknik Informatika*, 11(2), 100.
- <https://doi.org/10.22441/format.2022.v11.i2.001>.
- Pratomo, D. N., Kusumaning Putri, D. U., & Azhari, A. (2022). Implementasi optical character recognition berbasis deep learning untuk ekstraksi data sertifikat tanah. *Jurnal Informatika: Jurnal Pengembangan IT*, 7(3), 131–134.
- <https://doi.org/10.30591/jpit.v7i3.3657>.
- Purwanto, P., Safitri, D. Y., & Pudail, M. (2023). Edukasi pencatatan laporan keuangan sederhana bagi pelaku usaha mikro, kecil dan menengah (UMKM). *As-Sidanah: Jurnal Pengabdian Masyarakat*, 5(1), 1–14.
- <https://doi.org/10.35316/assidanah.v5i1.1-14>.
- Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, S., & Zaqiah, Q. Y. (2022). Deep learning dan penerapannya dalam pembelajaran. *JIIP - Jurnal Ilmiah Ilmu Pendidikan*, 5(9), 3258–3267.
- <https://doi.org/10.54371/jiip.v5i9.805>.
- ROSE, O. (2025). *Android Jetpack Compose UI components handbook*. Onyx Rose.

- Santoso, I. B., Ajii, I. P., Franskusuma, S., Putri, K. A., Ardharani, Y., Mujiastuti, R., Ambo, S. N., Meilina, P., Rosanti, N., & Amri, N. (2025). Educating on the application of TensorFlow in artificial intelligence, machine learning, and deep learning. *Society: Jurnal Pengabdian Masyarakat*, 4(2), 318–325. <https://doi.org/10.55824/jpm.v4i2.547>.
- Satav, M. S., Varade, T., Kothavale, D., Thombare, S., & Lokhande, P. (2020). Data extraction from invoices using computer vision. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, 316–320. <https://doi.org/10.1109/ICIIS51140.2020.9342722>.
- Subur, J., Suryadhi, M. T., Al Hafizh, N. R., & Reza, M. (2024). Pemanfaatan teknologi computer vision untuk deteksi ukuran ikan bandeng dalam membantu proses sortir ikan. *CYCLOTRON*, 7(01), 52–60. <https://doi.org/10.30651/cl.v7i01.21239>.
- Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*. <https://doi.org/10.3390/make5040083>.
- Zainuddin, F. F., P, F. C. E., Dhyaksa, P. S., Ardiansyah, M. B., Buana, P. A., & Penulis Korespondensi. (2025). System usability scale (SUS): Analisis pengalaman pengguna pada portal penerimaan mahasiswa baru Universitas Semarang. *Jurnal Komputer dan Teknologi Sains (KOMTEKS)*, 4(1), 23–28.