BAB II
KAJIAN LITERATUR

2.1 Pencatatan Keuangan

Pencatatan Keuangan merupakan proses sistematis untuk mendokumentasi dan mencatat setiap transaksi yang terjadi dalam suatu kegiatan ekonomi sebuah entitas, baik dalam bentuk pemasukan (*income*) maupun pengeluaran (*expense*). Salah satu kendala utama perkembangan UMKM adalah minimnya pengetahuan dalam pencatatan dan pengaturan keuangan yang baik [11]. Tujuan utama dari pencatatan keuangan adalah untuk memberikan catatan yang berisi informasi yang akurat, relevan, dan terstruktur mengenai kondisi keuangan suatu entitas, yang memudahkan untuk melakukan pemantauan (*monitoring*) keuangan[12] sehingga dapat digunakan sebagai dasar dalam pengambilan keputusan ekonomi.

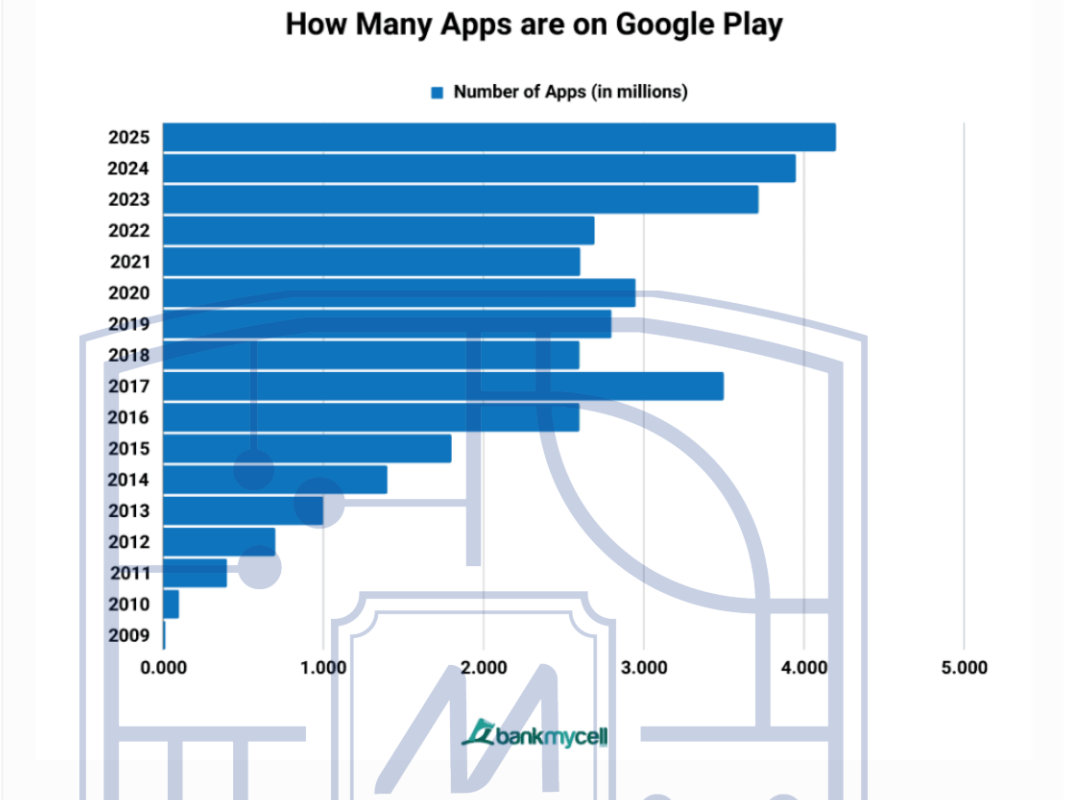
Bagi usaha skala kecil seperti apotik, pencatatan keuangan menjadi sangat krusial karena usaha skala kecil biasanya memiliki sumber daya terbatas dan rentan terhadap fluktuasi operasional. Pencatatan yang rapi membantu menjaga keberlanjutan usaha dan memfasilitasi pertanggungjawaban serta akses ke pembiayaan jika diperlukan. Seperti penelitian yang telah menunjukkan bahwa adopsi sistem pencatatan keuangan yang baik pada UMKM memperkuat transparansi dan meningkatkan kemampuan usaha untuk berkembang [13].

Pencatatan keuangan pada Apotik Sinar Bintang 2 memiliki peran penting untuk memantau pendapatan, mengontrol pengeluaran, dam menilai kinerja usaha yang terjadi secara keseluruhan. Tanpa pencatatan yang baik, pemilik usaha sering mengalami kesulitan untuk mengetahui keuntungan atau kerugian yang sebenarnya diperoleh. Pencatatan keuangan juga membantu pemilik usaha dalam membuat keputusan seperti menentukan harga jual berdasarkan modal serta kondisi keuangan, serta mengevaluasi efektivitas strategi penjualan.

2.2 Aplikasi *Mobile*

Aplikasi *mobile* adalah perangkat lunak yang dirancang khusus untuk digunakan pada perangkat *mobile* seperti *smartphone* dan *tablet* dengan tujuan memeberikan kemudahan akses terhadap informasi dan layanan secara cepat dan fleksibel. Aplikasi *mobile* juga didefenisikan sebagai perangkat lunak yang dikembangkan untuk memenuhi kebutuhan

pengguna dalam melakukan berbagai aktifitas sehari-hari seperti komunikasi, manajemen data maupun transaksi keuangan secara efisien dari perangkat [14].



Gambar 2. 1 Data Statistik Aplikasi Mobile di Google Play Store

Berdasarkan data statistik yang diperoleh dari BankMyCell, jumlah aplikasi yang tersedia di Google Play Store menunjukkan peningkatan yang signifikan dalam kurun waktu lebih dari sepuluh tahun terakhir. Sejak tahun 2010 hingga 2025, pertumbuhan jumlah aplikasi *mobile* terus meningkat seiring dengan berkembangnya teknologi smartphone dan meningkatnya kebutuhan masyarakat terhadap layanan digital. Meskipun pada beberapa periode terjadi penurunan akibat kebijakan penertiban aplikasi oleh Google, secara keseluruhan tren pertumbuhan tetap menunjukkan arah yang positif. Hal ini menegaskan bahwa aplikasi *mobile* memiliki peran yang semakin penting dalam berbagai sektor kehidupan, sehingga pengembangan aplikasi *mobile* menjadi kebutuhan yang relevan dan strategis di era digital saat ini.

Aplikasi *mobile* memiliki karakteristik portabilitas dan kemudahan akses karena dapat digunakan kapan saja dan dimana saja. Terdapat 3 jenis aplikasi *mobile*, yaitu [15], [16]:

1. Aplikasi *native*

Aplikasi *native*/asli adalah aplikasi yang dikembangkan khusus pada perangkat dengan sistem operasi tertentu, seperti *Android* maupun iOS. Umumnya, terdapat beberapa aplikasi *native* yang sudah terpasang secara default namun dapat juga di-download oleh pengguna melalui *Play Store* atau *App Store*. Aplikasi ini dapat mengakses seluruh fitur perangkat seperti kamera, GPS, sensor, kontak, dan notifikasi secara optimal. Selain itu, aplikasi ini dapat beroperasi tanpa koneksi internet (*offline*) untuk beberapa fungsi.

2. Aplikasi *hybrid*

   Aaplikasi *hybrid* merupakan perpaduan antara aplikasi *native* dan aplikasi web, yang mana antarmuka dan logika aplikasi dikembangkan menggunakan teknologi web, tetapi dikemas dalam wadah (*wrapper*) *native* sehingga dapat diinstal seperti aplikasi biasa.

3. Aplikasi *Cross-Platform*, yaitu aplikasi yang dikembangkan satu kali dan dapat berjalan diberbagai sistem operasi menggunakan framework seperti *flutter* atau *react native*. Pendekatan cross-platform semakin banyak digunakan karena mampu menjaga keseimbangan antara kinerja, stabilitas, dan efisiensi biaya pengembangan. Hal ini menjadikannya pilihan ideal untuk usaha kecil dan menengah (UKM) seperti apotik, yang membutuhkan solusi aplikasi efisien namun tetap fungsional.

Dalam konteks pengembangan perangkat lunak modern, salah satu teknologi yang banyak digunakkan adalah *flutter*. *Flutter* merupakan *Software Development Kit* (SDK) yang menggunakan bahasa pemograman *dart* yang memungkinkan pengembang membuat aplikasi yang dapat dijalankan pada sistem operasi android maupun iOS dengan satu basis kode. Hal ini tentu memberikan memberikan keuntungan dari segi efisisensi waktu pengembangan dan perawatan aplikasi. Selain itu, *flutter* memiliki fitur *hot reload* yang memungkinkan pengembangan melihat perubahan kode secara langung tanpa melakukan kompilasi ulang secara penuh, sehingga proses pengembangan menjadi lebih cepat dan interaktif [17].

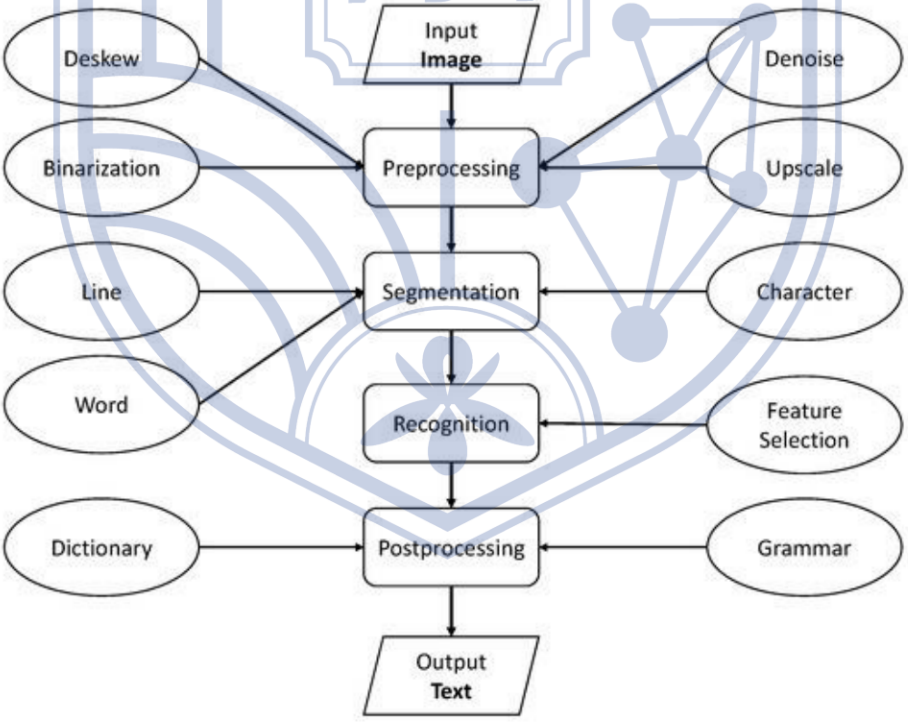
Dalam pengembangan aplikasi pencatatan keuangan, penggunaan aplikasi *mobile* berbasis *flutter* dinilai tepat dikarenakan mampu menyajikan antarmuka yang responsif, mudah digunakan, serta mampu berintegrasi dengan pihak ketiga yaitu OCR untuk pemindaian teks[18]. Dengan demikian, aplikasi *mobile* tidak hanya berfungsi sebagai media pencatatan data, tetapi juga sebagai alat otomatis proses input yang sebelumnya dilakukan secara manual.

2.3 *Optical Character Recognition* (OCR)

Optical Character Recognition (OCR) adalah proses yang mengonversi gambar teks menjadi format teks yang dapat dibaca oleh mesin. OCR memiliki kemampuan membaca seperti manusia, walaupun hasil pembacaan belum sepenuhnya akurat dibandingkan dengan manusia. OCR banyak digunakan secara luas di berbagai industri seperti pemerintahan, keuangan, dan layanan kesehatan [19]. Sebagai bagian dari otomatisasi dokumen, OCR telah berkembang dari sekadar pengenalan huruf ke sistem berlapis yang menyertakan pembelajaran mesin (*machine learning*) dan jaringan saraf dalam (*deep learning*) untuk meningkatkan akurasi dan fleksibilitas pengenalan karakter [20].

Pada penelitian ini, teknologi OCR digunakan untuk membantu proses pembacaan teks pada nota pengeluaran apotik. Fitur OCR bertujuan untuk mempermudah pengguna dalam melakukan pencatatan transaksi pengeluaran dengan cara mengekstraksi informasi teks dari gambar nota transaksi. Penggunaan OCR pada dokumen transaksi dinilai mampu membantu proses digitalisasi data serta mengurangi kesalahan input manual pada sistem pencatatan keuangan [21].

Proses kerja OCR meliputi beberapa tahapan berikut:



Gambar 2. 2 Proses Kerja Optical Character Recognition
Sumber [22].

Secara umum, sistem OCR bekerja melalui beberapa tahapan seperti *preprocessing*, *segmentasi karakter*, proses pengenalan karakter, dan *post-processing*. Tahapan tersebut bertujuan untuk meningkatkan kualitas citra dan menghasilkan teks digital yang lebih akurat. Namun pada penelitian ini, tahapan-tahapan tersebut tidak diimplementasikan secara manual karena seluruh proses pengenalan teks ditangani secara internal oleh *Google Vision OCR* melalui layanan *Google Cloud Vision API*.

Implementasi OCR pada aplikasi ini menggunakan *Google Cloud Vision API* sebagai library OCR pada platform *Flutter*. *Google Cloud Vision API* merupakan layanan *cloud-based* dari *Google* yang menyediakan fitur pendeteksian teks dan ekstraksi karakter dari gambar secara otomatis [23]. Library ini mampu melakukan proses pengenalan teks secara internal sehingga pengembang tidak perlu membangun algoritma OCR secara manual dari awal. Fitur yang digunakan dalam penelitian ini adalah *Document Text Detection*, yaitu fitur OCR yang dirancang untuk mengenali teks pada dokumen dengan struktur yang kompleks, seperti nota transaksi, faktur, dan dokumen administrasi lainnya. Fitur tersebut mampu menghasilkan teks hasil ekstraksi beserta informasi posisi setiap kata pada gambar (*bounding box*), sehingga dapat membantu proses identifikasi informasi yang terdapat pada dokumen. Penggunaan *Google Cloud Vision API* dinilai sesuai untuk implementasi aplikasi *mobile* karena mampu memberikan hasil pengenalan teks yang cepat dan akurat pada berbagai jenis gambar dokumen maupun nota transaksi [24].

Adapun alur proses OCR pada aplikasi ini dimulai dari pengguna mengambil gambar nota transaksi menggunakan kamera perangkat atau memilih gambar dari galeri. Gambar tersebut kemudian dikirim ke *Google Vision API* menggunakan fitur *Document Text Detection*. Setelah proses OCR selesai, sistem menerima hasil ekstraksi berupa teks mentah (*raw text*) dan informasi posisi kata pada gambar.

Teks mentah yang dihasilkan oleh OCR tidak langsung disimpan ke basis data. Hasil OCR terlebih dahulu diproses menggunakan mekanisme *vendor detection*, *table extraction*, dan *parser* berbasis aturan (*rule-based parser*) untuk mengubah data yang tidak terstruktur menjadi informasi transaksi yang terstruktur. Data yang berhasil diekstraksi meliputi vendor, grand total transaksi, nama barang, kuantitas, satuan, harga satuan, dan total harga barang. Setelah proses ekstraksi selesai, hasil akan ditampilkan kepada pengguna untuk diverifikasi dan diperbaiki apabila diperlukan sebelum disimpan ke basis data Firebase.

1. Vendor Detection

Vendor Detection merupakan proses identifikasi vendor berdasarkan hasil ekstraksi teks yang diperoleh dari OCR. Proses ini bertujuan untuk mengenali sumber nota transaksi sehingga sistem dapat menentukan metode pemrosesan yang sesuai dengan format nota yang digunakan. Setiap vendor umumnya memiliki tata letak, susunan informasi, dan format penulisan yang berbeda sehingga diperlukan mekanisme identifikasi sebelum proses ekstraksi data dilakukan.

Pada penelitian ini, *Vendor Detection* dilakukan dengan mencocokkan kata kunci tertentu yang terdapat pada hasil OCR dengan daftar vendor yang telah ditentukan sebelumnya. Hasil identifikasi vendor digunakan untuk memilih parser yang sesuai dengan format nota. Vendor yang menjadi fokus penelitian adalah vendor yang paling sering digunakan oleh Apotik Sinar Bintang 2 yaitu: Ochi Farma, Asia Susu, dan Matrix. Penggunaan Vendor Detection membantu meningkatkan akurasi ekstraksi data karena setiap format nota dapat diproses menggunakan aturan yang sesuai dengan karakteristik masing-masing vendor.

2. Table Extraction

Table Extraction merupakan proses pengambilan informasi data transaksi yang tersusun dalam bentuk tabel pada dokumen. Pada nota transaksi, informasi seperti nama barang, jumlah barang, satuan, harga, dan total harga umumnya disusun dalam format baris dan kolom sehingga diperlukan mekanisme untuk mengidentifikasi hubungan antar data tersebut.

Pada penelitian ini, proses *Table Extraction* dilakukan dengan memanfaatkan informasi posisi kata (*bounding box*) yang diperoleh dari *Google Vision OCR*. Informasi posisi tersebut digunakan untuk mengelompokkan kata-kata yang berada pada baris yang sama sehingga membentuk satu data transaksi. Hasil dari proses ini berupa kumpulan baris transaksi yang selanjutnya diproses oleh parser untuk memperoleh data yang lebih terstruktur. Penggunaan *Table Extraction* memungkinkan sistem mengenali struktur data transaksi dengan lebih baik dibandingkan hanya menggunakan teks mentah hasil OCR.

3. Rule-Based Parser

Rule-Based Parser merupakan metode ekstraksi data yang menggunakan seperangkat aturan tertentu untuk mengidentifikasi dan mengubah data tidak terstruktur menjadi data yang terstruktur. Pendekatan ini banyak digunakan pada

dokumen yang memiliki pola penulisan yang relatif konsisten sehingga informasi penting dapat dikenali berdasarkan aturan yang telah ditentukan sebelumnya.

Pada penelitian ini, *Rule-Based Parser* digunakan untuk mengubah hasil OCR yang masih berupa teks mentah menjadi data transaksi yang terstruktur. Parser bertugas mengekstraksi informasi seperti vendor, grand total transaksi, nama barang, kuantitas, satuan, harga satuan, dan total harga barang. Karena setiap vendor memiliki format nota yang berbeda, penelitian ini menggunakan beberapa parser yang disesuaikan dengan karakteristik masing-masing vendor. Pendekatan ini dipilih untuk meningkatkan akurasi ekstraksi data transaksi sehingga informasi yang dihasilkan lebih sesuai dengan isi nota asli.

4. Fuzzy Matching

Fuzzy Matching merupakan teknik pencocokan data yang digunakan untuk membandingkan dua buah teks yang memiliki kemiripan meskipun tidak identik secara karakter. Teknik ini sering digunakan untuk menangani kesalahan penulisan, perbedaan format, maupun kesalahan hasil pengenalan karakter yang dihasilkan oleh sistem OCR.

Pada penelitian ini, *Fuzzy Matching* digunakan pada tahap pasca pemrosesan (*post-processing*) untuk mencocokkan nama barang hasil OCR dengan data master barang yang terdapat pada sistem. Proses ini dilakukan karena hasil OCR terkadang menghasilkan kesalahan karakter, seperti huruf yang terbaca berbeda atau adanya karakter yang hilang. Melalui *Fuzzy Matching*, sistem dapat memilih nama barang yang paling mendekati hasil OCR sehingga membantu meningkatkan kualitas data transaksi yang akan disimpan ke basis data. Penggunaan metode ini juga membantu mengurangi kesalahan pencatatan yang disebabkan oleh ketidakakuratan hasil pengenalan teks.

Metode *Fuzzy Matching* yang digunakan pada penelitian ini adalah *Jaccard Similarity*. *Jaccard Similarity* merupakan metode pengukuran kemiripan dua himpunan data yang dihitung berdasarkan perbandingan jumlah elemen yang sama terhadap jumlah seluruh elemen unik dari kedua himpunan [25].

$$J(A,B) = \frac{|\, A \cap B \,|}{|\, A \cup B \,|}$$

Nilai Jaccard Similarity berada pada rentang 0 hingga 1. Semakin mendekati nilai 1 maka semakin tinggi tingkat kemiripan antara kedua teks yang dibandingkan.

Pada penelitian ini, nilai similarity digunakan untuk menentukan nama barang pada data master yang memiliki tingkat kemiripan tertinggi dengan hasil OCR. Apabila nilai similarity yang diperoleh lebih besar atau sama dengan 0,70 maka nama barang hasil OCR akan diganti dengan nama barang dari data master yang memiliki nilai similarity tertinggi.

Dengan kemampuan tersebut, *Google Vision OCR* dapat digunakan sebagai komponen utama dalam proses digitalisasi pencatatan pengeluaran pada Apotik Sinar Bintang 2. Penggunaan OCR pada aplikasi ini diharapkan dapat membantu mengurangi kesalahan input data transaksi serta meningkatkan efisiensi proses pencatatan keuangan pada apotik.

2.4 Basis Data (*Database*)

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis dalam suatu sistem manajemen basis data (DBMS) agar dapat diakses, dikelola, dan diperbarui dengan efisien [26]. Dalam konteks pengembangan aplikasi mobile seperti aplikasi pencatatan keuangan, basis data berperan sebagai fondasi untuk menyimpan seluruh data yang diperlukan, mulai dari data transaksi, data pengguna, hingga hasil ekstraksi teks dari proses OCR.

2.4.1 Fungsi Basis Data

Fungsi utama dari basis data dalam sistem informasi, termasuk aplikasi mobile yaitu [26]:

1. Penyimpanan Data Terpusat

   Basis data menyediakan repositori tunggal untuk semua data aplikasi, mengurangi redundansi dan memudahkan manajemen data. Hal ini memungkinkan penyimpanan yang terorganisir dari berbagai jenis data seperti transaksi keuangan, data pengguna, dan *log-system*.
2. Manipulasi Data

   Basis data mendukung operasi dasar yang dikenal sebagai CRUD (*Create, Read, Update, Delete*) pada data transaksi keuangan. Operasi ini memungkinkan aplikasi untuk menambah transaksi baru, membaca riwayat transaksi, memperbarui data yang sudah ada, dan menghapus data yang tidak diperlukan.
3. Menjaga Integritas dan Konsistensi Data

Melalui penerapan aturan, constraint, dan transaksi ACID (*Atomicity, Consistency, Isolation, Durability*), basis data memastikan data yang disimpan tetap akurat dan andal [27]. Integritas data sangat kritis dalam konteks keuangan untuk menghindari kesalahan pencatatan.

4. Keamanan Data

   Basis data mengontrol akses data melalui mekanisme otentikasi dan otorisasi, melindungi data sensitif keuangan dari akses yang tidak sah. Implementasi *security layer* ini *essential* untuk aplikasi yang menangani data finansial.

5. Fasilitas Pencarian dan Pelaporan

   Basis data memungkinkan pengambilan data yang cepat dan spesifik untuk kebutuhan laporan keuangan dan analisis. *Query* yang efisien memungkinkan pembuatan laporan keuangan periodik dengan akurasi tinggi.

2.4.2 Jenis-jenis Basis Data

Basis data dibedakan menjadi dua jenis utama, yaitu basis data relasional dan basis data non-relasional. Keduanya memiliki karakteristik dan keunggulan yang berbeda-beda sesuai dengan kebutuhan aplikasi. Berikut penjabaran dari kedua basis data tersebut [26]:

a. Basis Data Relasional

   Basis data relasional (*Relational Database*) merupakan sistem penyimpanan data yang menggunakan struktur tabel dan hubungan (relasi) antar tabel untuk mengorganisir data yang disusun dalam bentuk baris dan kolom dengan *primary key* dan *foreign key* sebagai penghubung antar tabel. Basis data relasional banyak digunakan karena kemudahan dalam pengelolaan data serta dukungan terhadap Structured Query Language (SQL) untuk melakukan operasi CRUD (*Create, Read, Update, Delete*).

b. Basis Data Non-Relasional

   Basis data non-relasional atau sering disebut NoSQL (*Not Only SQL*) adalah sistem penyimpanan data yang dirancang untuk menangani data dalam skala besar, tidak terstruktur, atau semi-terstruktur. Tidak seperti sistem relasional yang berbasis tabel, basis data NoSQL menggunakan berbagai model penyimpanan seperti dokumen (*document-based*), *key-value*, kolom (*column-based*), atau graf (*graph-based*). Model NoSQL banyak digunakan pada aplikasi modern yang memerlukan skalabilitas tinggi, fleksibilitas struktur data, dan kinerja cepat, misalnya pada aplikasi berbasis *cloud*, media sosial, maupun sistem *big-data*.

Dalam penelitian ini, jenis basis data yang digunakan adalah basis data non-relasional berupa Firebase Firestore karena mampu mendukung sinkronisasi data secara real-time dan integrasi yang baik dengan framework Flutter.

2.4.3 Firebase Firestore

Firebase Firestore merupakan layanan basis data NoSQL berbasis cloud yang dikembangkan oleh Google untuk mendukung aplikasi web maupun mobile secara real-time. Firestore menyimpan data dalam bentuk collection dan document sehingga lebih fleksibel dibandingkan basis data relasional yang menggunakan tabel. Firestore dipilih karena memiliki struktur data yang fleksibel, mendukung query secara real-time, serta mudah diintegrasikan dengan aplikasi mobile berbasis Flutter [28].Firebase Firestore mendukung sinkronisasi data secara real-time, akses cloud, autentikasi pengguna, serta integrasi yang baik dengan framework Flutter. Selain itu, Firestore memungkinkan aplikasi untuk menyimpan dan mengambil data secara efisien tanpa memerlukan pengelolaan server secara manual. Penggunaan Firebase pada aplikasi mobile juga dinilai mampu meningkatkan efisiensi pengelolaan data dan mempermudah proses pengembangan aplikasi [28], [29].

Pada penelitian ini, Firebase Firestore digunakan sebagai basis data utama untuk menyimpan data pengguna, data barang, transaksi pemasukan, transaksi pengeluaran, serta hasil ekstraksi OCR dari nota transaksi.

2.4.4 Arsitektur dan Komponen Basis Data

Arsitektur basis data menggambarkan bagaimana data disimpan, dikelola, dan diakses oleh pengguna atau aplikasi melalui sistem manajemen basis data (DBMS). Secara umum, arsitektur basis data terdiri atas beberapa lapisan yang saling berhubungan untuk memastikan proses penyimpanan dan pengambilan data berjalan efisien, aman, dan konsisten.

Arsitektur basis data modern biasanya dibagi menjadi tiga tingkat utama, yaitu tingkat eksternal (*external level*), tingkat konseptual (*conceptual level*), dan tingkat internal (*internal level).* Pembagian ini dikenal dengan istilah *three-schema architecture*, yang berfungsi untuk memisahkan cara data disimpan secara fisik dengan cara data ditampilkan kepada pengguna [30].

1. Tingkat Eksternal (External Level)

Tingkat eksternal merupakan lapisan paling luar yang berhubungan langsung dengan pengguna atau aplikasi. Pada tingkat ini, data ditampilkan dalam bentuk yang mudah dipahami sesuai kebutuhan masing-masing pengguna. Setiap pengguna dapat memiliki pandangan (*view*) yang berbeda terhadap data yang sama.

Dalam konteks aplikasi mobile pencatatan keuangan, tingkat eksternal diwakili oleh antarmuka pengguna (*user interface*) yang menampilkan data transaksi, laporan keuangan, dan hasil ekstraksi teks dari OCR.

2. Tingkat Konseptual (Conceptual Level)

   Lapisan ini menggambarkan struktur logis dari seluruh basis data, tanpa memperhatikan bagaimana data disimpan secara fisik. Tingkat konseptual menentukan bagaimana entitas, atribut, relasi, dan batasan (*constraint*) saling berhubungan dalam satu sistem.

   Pada aplikasi ini, tingkat konseptual mencakup entitas utama seperti pengguna, barang, pemasukan, dan pengeluaran beserta relasi antar data. Struktur ini biasanya digambarkan dalam *Entity Relationship Diagram* (ERD) sebagai acuan perancangan basis data aplikasi.

3. Tingkat Internal (Internal Level)

   Tingkat internal merupakan lapisan yang berhubungan langsung dengan penyimpanan fisik data. Pada Firebase Firestore, data disimpan secara *cloud-based* dalam bentuk collection dan document sehingga memungkinkan sinkronisasi data secara real-time dan akses data dari berbagai perangkat secara fleksibel.

2.4.4 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah model konseptual yang digunakan untuk menggambarkan struktur logis basis data melalui entitas, atribut, serta relasi antar entitas [31]. ERD berfungsi sebgai acuan utama dalam perancangan struktur data sebelum basis data di implementasikan pada *Firebase Firestore*. ERD membantu proses perancangan data menjadi lebih terstruktur, mengurangi redudansi data keuangan aplikasi [31], [32]. Pada ERD terdapat tiga komponen utama yaitu [33]:

1. Entitas

   Entitas adalah objek atau komponen utama yang di representasikan dalam sistem basis data dan memiliki peran penting dalam proses pengolahan data. Entitas menggambarkan objek nyata ataupun konsep yang relevan dengan kebutuhan sistem.

2. Atribut

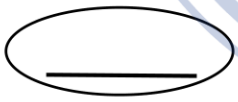
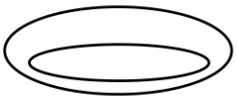
Atribut merupakan karakteristik atau properti yang dimiliki oleh suatu entitas. Atribut digunakan untuk menyimpan informasi spesifik terkait entitas. Atribut dapat ditetapkan sebagai *Primary Key* (PK) untuk menandai keunikan data, atau *Foreign Key* (FK) yang digunakan untuk menghubungkan suatu entitas dengan entitas lainnya. Dengan adanya atribut, basis data mampu menyimpan informasi dengan struktur yang jelas.

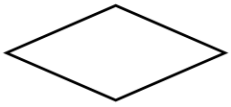
3. Relasi

Relasi adalah hubungan yang menggambarkan keterkaitan antar entitas dalam suatu sistem. Relasi menentukan bagaimana satu entitas dapat berhubungan atau terhubung dengan entitas lain nya, seperti *one-to-many*, atau *many-to-one*.

Penjelasan komponen-komonen pembentuk ERD dapat dilihat pada tabel di bawah ini [34]:

Tabel 2. 1 Komponen Penyusun ERD

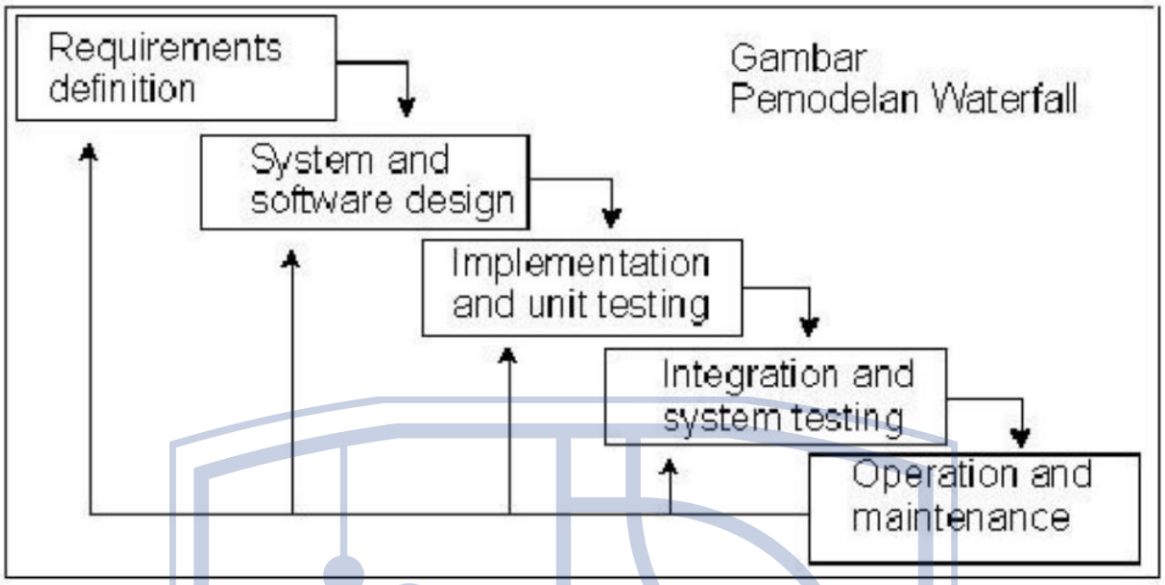
Notasi	Komponen	Keterangan
	Entitas	Entitas merupakan objek yang memiliki informasi dan perlu disimpan datanya agar dapat diakses serta digunakan oleh sistem atau aplikasi.
	Atribut	Atribut merupakan *field* atau kolom informasi yang perlu disimpan sebagai bagian dari entitas.
	Atribut Kunci Primer	Atribut kunci primer merupakan *field* atau kolom dalam entitas yang harus disimpan dan berfungsi sebagai pengenal untuk mengakses *record* tertentu. Kunci primer dapat terdiri lebih dari satu kolom, selama kombinasi kolom tersebut menghasilkan nilai yang unik.
	Atribut Multi Nilai	Atribut multi nilai adalah *field* ata kolom pada sebuah entitas yang memungkinkan penyimpanan beberapa nilai dalam satu atribut.

	Relasi	Relasi merupakan simbol yang menunjukkan adanya hubungan antara satu entitas dengan entitas lainnya, dan umumnya dirumuskan menggunakan kata kerja.
	Asosiasi	Asosiasi adalah hubungan yang mengaitkan relasi dengan entitas, dimana masing-masing ujungnya menunjukkan banyaknya kemungkinanan keterlibatan atau perulangannya.

2.5 Metode Waterfall

Metode Waterfall merupakan salah satu metode pengembangan perangkat lunak yang dilakukan secara sistematis dan berurutan, dimana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya [35]. Metode ini banyak digunakan dalam pengembangan sistem informasi karena memiliki alur kerja yang terstruktur dan mudah dipahami. Setiap tahapan pada metode waterfall menghasilkan keluaran yang menjadi dasar bagi tahapan selanjutnya sehingga proses pengembangan sistem dapat berjalan secara lebih terorganisir.

Tahapan metode waterfall secara umum terdiri dari requirement analysis, system design, implementation, testing, dan maintenance [36]. Tahapan tersebut dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Model Pengembangan Waterfall

Sumber [36]

Berdasarkan Gambar 2.3, tahapan metode waterfall terdiri dari:

1. Requirement Analysis

   Tahap ini merupakan proses pengumpulan dan analisis kebutuhan sistem berdasarkan permasalahan yang ada. Pada tahap ini dilakukan identifikasi kebutuhan pengguna, kebutuhan sistem, serta fitur-fitur yang akan dikembangkan dalam aplikasi.
2. System Design

   Tahap desain dilakukan untuk merancang struktur sistem, basis data, antarmuka pengguna, serta alur kerja aplikasi sebelum proses implementasi dimulai.
3. Implementation

   Tahap implementasi merupakan proses penerjemahan rancangan sistem ke dalam bentuk kode program menggunakan bahasa pemrograman dan framework yang telah ditentukan.
4. Testing

   Tahap pengujian dilakukan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan dan fungsi yang telah dirancang sebelumnya. Pengujian dilakukan untuk menemukan kesalahan atau bug pada sistem.
5. Maintenance

Tahap maintenance dilakukan setelah sistem digunakan. Pada tahap ini dilakukan perbaikan, pembaruan, maupun pengembangan fitur tambahan agar sistem tetap berjalan dengan baik.

Metode waterfall digunakan dalam penelitian ini karena memiliki tahapan pengembangan yang terstruktur dan sistematis sehingga memudahkan proses pengembangan aplikasi pencatatan keuangan berbasis mobile. Selain itu, metode ini sesuai digunakan pada penelitian yang memiliki kebutuhan sistem yang sudah jelas sejak awal proses pengembangan.

2.6 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk merancang, menggambarkan, dan mendokumentasikan sistem perangkat lunak berbasis objek. UML membantu pengembang sistem dalam memvisualisasikan struktur dan alur kerja sistem sebelum dilakukan proses implementasi program [37]. UML juga digunakan sebagai alat komunikasi antara pengembang dan pengguna agar kebutuhan sistem dapat dipahami dengan lebih jelas.

UML memiliki berbagai jenis diagram yang digunakan sesuai kebutuhan perancangan sistem, seperti Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram [38]. Setiap diagram memiliki fungsi dan peran masing-masing dalam memodelkan sistem. Penggunaan UML dinilai mampu membantu proses analisis dan perancangan sistem menjadi lebih terstruktur serta mempermudah proses pengembangan perangkat lunak.

Dalam penelitian ini, UML digunakan untuk memodelkan proses bisnis dan alur sistem aplikasi pencatatan keuangan berbasis mobile pada Apotik Sinar Bintang 2. Diagram UML yang digunakan meliputi Use Case Diagram dan Activity Diagram.

2.6.1 Use Case Diagram

Use Case Diagram merupakan salah satu jenis diagram UML yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem. Use Case Diagram membantu dalam mengidentifikasi fungsi-fungsi yang tersedia pada sistem serta pihak yang dapat

menggunakan fungsi tersebut [39]. Diagram ini digunakan untuk menggambarkan kebutuhan fungsional sistem dari sudut pandang pengguna.

Pada Use Case Diagram terdapat dua komponen utama yaitu aktor dan use case. Aktor merupakan pihak yang berinteraksi langsung dengan sistem, sedangkan use case menggambarkan aktivitas atau layanan yang disediakan oleh sistem [40]. Dengan menggunakan Use Case Diagram, pengembang dapat memahami hubungan antara pengguna dengan sistem secara lebih jelas sehingga proses perancangan sistem menjadi lebih mudah dilakukan.

Dalam penelitian ini, Use Case Diagram digunakan untuk menggambarkan interaksi pengguna dengan fitur-fitur pada aplikasi pencatatan keuangan, seperti pengelolaan transaksi pemasukan, pengeluaran, pengelolaan transaksi, pengelolaan karyawan, laporan keuangan, dan proses OCR pada nota transaksi.

2.6.2 Activity Diagram

Activity Diagram merupakan salah satu jenis diagram UML yang digunakan untuk menggambarkan alur aktivitas atau proses kerja suatu sistem. Diagram ini menunjukkan urutan aktivitas, proses pengambilan keputusan, serta aliran proses dari awal hingga akhir sistem [41].

Activity Diagram membantu pengembang dalam memodelkan proses bisnis dan logika sistem secara lebih detail. Selain itu, Activity Diagram dapat digunakan untuk menggambarkan proses yang berjalan secara berurutan maupun paralel dalam suatu sistem [42]. Dengan adanya Activity Diagram, alur kerja sistem dapat dipahami dengan lebih mudah sebelum sistem diimplementasikan ke dalam bentuk program.

Dalam penelitian ini, Activity Diagram digunakan untuk menggambarkan alur aktivitas pengguna pada aplikasi pencatatan keuangan, seperti proses login, pengelolaan transaksi, pengelolaan karyawan, proses OCR nota, serta pembuatan laporan keuangan.

2.7 Kebutuhan Non-Fungsional dan ISO/IEC 25010

Kebutuhan non-fungsional (*non-functional requirements*) merupakan kebutuhan yang menggambarkan bagaimana suatu sistem harus bekerja, mencakup aspek kualitas seperti kinerja, keamanan, kemudahan penggunaan, dan keandalan, yang berbeda dengan

kebutuhan fungsional yang berfokus pada fitur-fitur yang harus dimiliki sistem [43]. Meskipun sering dianggap sebagai aspek pendukung, kebutuhan non-fungsional memiliki peran penting dalam menentukan keberhasilan penerapan suatu perangkat lunak, karena kualitas sistem tidak hanya dinilai dari kelengkapan fiturnya, tetapi juga dari seberapa baik sistem tersebut berjalan dalam praktiknya [44].

Dalam menentukan kategori kebutuhan non-fungsional, penelitian ini mengacu pada standar **ISO/IEC 25010**, yaitu model kualitas produk perangkat lunak (*Software Product Quality Model*) yang merupakan pembaruan dari standar ISO/IEC 9126 sebelumnya [45]. ISO/IEC 25010 mendefinisikan delapan karakteristik kualitas perangkat lunak, yaitu *functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability,* dan *portability* [46]. Kedelapan karakteristik tersebut dapat dijelaskan sebagai berikut [46]:

1. Functional Suitability (Kesesuaian Fungsional)
   Mengukur sejauh mana suatu perangkat lunak menyediakan fungsi-fungsi yang memenuhi kebutuhan pengguna, baik dari segi kelengkapan, ketepatan, maupun kesesuaian fungsi tersebut dengan tugas dan tujuan penggunaan yang telah ditetapkan.
2. Performance Efficiency (Efisiensi Kinerja)
   Mengukur kinerja suatu sistem relatif terhadap sumber daya yang digunakan dalam kondisi tertentu, mencakup aspek waktu respons (*time behavior*), pemanfaatan sumber daya (*resource utilization*), dan kapasitas sistem dalam menangani beban kerja.
3. Compatibility (Kompatibilitas)
   Mengukur kemampuan suatu perangkat lunak untuk bertukar informasi dengan sistem atau perangkat lunak lain, serta dapat berjalan berdampingan dengan sistem lain dalam lingkungan atau perangkat keras yang sama tanpa saling mengganggu fungsinya.
4. Usability (Kegunaan)
   Mengukur sejauh mana suatu perangkat lunak dapat dipahami, dipelajari, dioperasikan, dan menarik bagi pengguna ketika digunakan dalam kondisi tertentu, sehingga pengguna dapat mencapai tujuan penggunaannya secara efektif dan efisien.

5. Reliability (Keandalan)

   Mengukur sejauh mana suatu sistem, produk, atau komponen dapat menjalankan fungsi yang telah ditentukan dalam kondisi tertentu selama periode waktu tertentu, termasuk kemampuan sistem untuk tetap tersedia (*availability*) dan pulih dari kegagalan (*recoverability*).

6. Security (Keamanan)

   Mengukur sejauh mana suatu perangkat lunak dapat melindungi informasi dan data agar hanya dapat diakses sesuai dengan tingkat otorisasi yang telah ditetapkan, mencakup aspek kerahasiaan (*confidentiality*), integritas data, dan akuntabilitas.

7. Maintainability (Kemudahan Pemeliharaan)

   Mengukur tingkat efektivitas dan efisiensi suatu perangkat lunak untuk dapat dimodifikasi, diperbaiki, atau dikembangkan lebih lanjut oleh pengembang, baik untuk memperbaiki kesalahan (*bug*) maupun untuk menyesuaikan dengan kebutuhan baru di lingkungan yang berubah.

8. Portability (Portabilitas)

   Mengukur sejauh mana suatu perangkat lunak dapat dipindahkan dan tetap berfungsi dengan baik dari satu lingkungan perangkat keras, perangkat lunak, atau lingkungan operasional lain ke lingkungan yang berbeda.

Penerapan ISO/IEC 25010 sebagai acuan kebutuhan non-fungsional juga telah digunakan pada berbagai penelitian sistem informasi sejenis, termasuk pada sistem informasi berbasis keuangan, sebagai landasan untuk memastikan kualitas fungsional dan *usability* sistem yang dibangun [47].

Pada penelitian ini terdapat 6 karakteristik yang dinilai relevan dengan konteks aplikasi, yaitu *Performance, Security, Usability, Reliability, Portability, Maintainability,* dan *Compatibility* yang selanjutnya menjadi dasar penyusunan kebutuhan non-fungsional. Karakteristik *Compatibility* menjadi relevan karena aplikasi ini terintegrasi dengan layanan pihak ketiga, yaitu Google Vision OCR, untuk mendukung fitur ekstraksi teks dari gambar nota transaksi. Karakteristik *Functional Suitability* tidak dibahas secara terpisah pada bagian ini karena telah diuraikan pada sub-bab Kebutuhan Fungsional.

2.8 Review Penelitan Terdahulu

Tabel 2. 2 Penelitian Terdahulu

No	Peneliti & Tahun	Judul	Metode	Fokus penelitian	Hasil Utama
1	Munawir Sajali Harahap, Richi Andrianto, Dermilan, Dewi Elok, Citra Khoiriah, Babang Andika (2023) [48]	Pencatat Keuangan: Pengembangan Aplikasi Futter untuk Mencatat dan Mengelola Transaksi Keuangan	Flutter, MySQL	Pengembangan aplikasi pencatatan keuangan berbasis Flutter untuk memudahkan pengguna dalam mencatat, mengelola, dan memantau transaksi keuangan secara efisien dan interaktif.	Aplikasi pencatatan keuangan berbasis Flutter berhasil dikembangkan dengan fitur pengelompokkan kategori utama pemasukan dan pengeluaran serta laporan keuangan.
2	Asri Mulyani, Yosep Septiana, Rizky Helmi (2022) [49]	Rancang Bangun Aplikasi Penjualan dan Persediaan Obat pada Apotek Berbasis Android	Extreme Programming (XP), UML	Pengembangan aplikasi penjualan dan persediaan obat berbasis Android untuk mempermudah pengelolaan transaksi dan monitoring stok secara efisien.	Aplikasi berbasis Android yang dapat membantu proses penjualan, pembuatan nota, serta monitoring persediaan obat sehingga meningkatkan efisiensi dan mengurangi human error.
3	Agnes Saputri, Alauddin	Aplikasi Manajemen Inventori	Waterfall, Flutter, Firebase	Pengembangan aplikasi manajemen	Aplikasi inventori berbasis mobile yang mampu meningkatkan

	Maulana Hirzan (2024) [50]	Berbasis Mobile Menggunakan Flutter dan Firebase Realtime Database	Realtime Database	inventori berbasis mobile menggunakan Flutter dan Firebase untuk meningkatkan efisiensi pengelolaan stok dan akses informasi secara real-time.	efisiensi pengelolaan stok, mempercepat pengecekan stok dari 15 menit menjadi kurang dari 1 menit, serta meningkatkan kualitas pelayanan.
4	Amelia Ananda Setiawan, Rangga Gelar Guntara, Btari Mariska Purwaamijaya (2025) [51]	Ekstraksi Informasi Struk Belanja Melalui Pemanfaatan Tesseract dan Regular Expressions	SDLC (Waterfall), YOLO, Tesseract (OCR), Regular Expressions	pengembangan sistem ekstraksi informasi dari struk belanja menggunakan teknologi OCR dan pengolahan data untuk mendukung otomatisasi pencatatan keuangan.	Sistem mampu mengekstrak informasi dari struk belanja menjadi data terstruktur dan membantu proses pencatatan keuangan dengan nilai rata-rata kepuasan pengguna sebesar 4,47 dari skala 5.
5	Alvin Rosidi & Afriyudi (2023) [6]	Aplikasi Pencatatan Keuangan Pribadi Berbasis Web Mobile	Object-Oriented Analysis and Design (OOAD)	Pengembangan aplikasi pencatatan keuangan berbasis web mobile untuk membantu pengguna dalam mengelola pemasukan dan	Penelitian ini menghasilkan aplikasi pencatatan keuangan berbasis web mobile yang memudahkan pengguna mencatat pemasukan dan pengeluaran secara terstruktur, serta menyajikan informasi

				pengeluaran secara terstruktur.	keuangan secara terorganisir dan akurat.
6	Ahmad Amru Fakhri & Theodoran Maria Putri Komul (2025) [9]	Implementasi Aplikasi Laundry Berbasis Machine Learning untuk Otomatisasi Pengelolaan Nota Transaksi	Metode Waterfall dengan penerapan Machine Learning dan OCR	Pengembangan aplikasi laundry dengan pemanfaatan machine learning dan OCR untuk otomatisasi pengelolaan nota serta pemantauan status transaksi secara real-time.	Penelitian ini menghasilkan aplikasi laundry berbasis mobile yang mengotomatisasi pengelolaan transaksi dan nota menggunakan OCR, serta menyediakan fitur status real-time, pencatatan digital, manajemen stok, dan laporan statistik untuk meningkatkan efisiensi operasional.
7	Shierly Mayco Angela & Ade Eviyanti (2024) [18]	Development Of Optical Character Recogition Technology In Flutter For Text Detection In Images [Pengembangan Teknologi Optical	Rapid Application Development (RAD)	Pengembangan teknologi OCR berbasis Flutter untuk mendeteksi dan mengubah teks pada gambar menjadi teks digital secara otomatis.	Penelitian ini menghasilkan aplikasi mobile berbasis Flutter yang mampu mendeteksi dan mengekstrak teks dari gambar dengan akurasi 88%, sehingga mempermudah proses konversi teks menjadi digital secara cepat dan efisien.

		Character Recognition Di Flutter Berupa Deteksi Teks Pada Gambar]			

Berdasarkan hasil kajian penelitian terdahulu, diketahui bahwa berbagai penelitian telah mengembangkan aplikasi pencatatan keuangan, manajemen inventori, dan pengelolaan transaksi berbasis mobile yang mampu meningkatkan efisiensi dan akurasi pengolahan data.

Namun, sebagian besar penelitian masih memiliki keterbatasan, seperti pencatatan yang dilakukan secara manual atau belum mengintegrasikan OCR secara optimal dalam sistem pencatatan keuangan, khususnya pada konteks apotik. Oleh karena itu, penelitian ini mengembangkan aplikasi mobile pencatatan keuangan dengan integrasi OCR pada Apotik Sinar Bintang 2 untuk mengotomatisasi input data dari nota, sehingga dapat meningkatkan efisiensi, akurasi, dan kemudahan dalam pengelolaan keuangan.

2.9 Apotik Sinar Bintang 2

Apotik merupakan sarana pelayanan kesehatan yang berfungsi sebagai tempat peracikan, penyimpanan, serta penyaluran obat kepada masyarakat berdasarkan resep dokter maupun secara swamedikasi (tanpa resep). Menurut peraturan Menteri Kesehatan Republik Indonesia Nomor 73 Tahun 2016 tentang Standar Pelayanan Kefarmasian, apotik memiliki peran strategis dalam menjamin ketersediaan dan distribusi obat yang aman, bermutu, serta bermanfaat bagi masyarakat. Selain menjalankan fungsi pelayanan kesehatan, apotik juga merupakan unit usaha yang memerlukan pengelolaan administratif dan keuangan yang tertib agar operasionalnya dapat berjalan secara berkelanjutan [52]. Oleh karena itu, pencatatan keuangan di apotik memiliki peran penting dalam memantau transaksi, mengendalikan pengeluaran, dan mengevaluasi kinerja keuangan secara periodik.

Apotik Sinar Bintang 2 merupakan salah satu unit usaha farmasi yang berlokasi di Kota Medan dan bergerak dalam bidang penjualan obat-obatan, alat kesehatan, serta produk kebutuhan medis lainnya. Sebagai apotik yang melayani masyarakat umum, Apotik Sinar Bintang 2 berkomitmen untuk memberikan pelayanan kesehatan yang cepat, tepat, dan

terjangkau. Dalam menjalankan kegiatan operasionalnya, apotik ini melakukan berbagai aktivitas utama seperti pengadaan obat dari distributor, pelayanan resep pelanggan, serta pencatatan transaksi harian yang terkait dengan keuangan dan inventori.

Pada praktiknya, sistem pencatatan keuangan di Apotik Sinar Bintang 2 masih dilakukan secara manual dengan mencatat setiap transaksi dalam buku tulis atau lembar kerja sederhana seperti spreadsheet. Proses tersebut memiliki beberapa kelemahan, di antaranya kesalahan input data, kehilangan dokumen, dan keterlambatan dalam penyusunan laporan keuangan. Kondisi ini menyebabkan kesulitan dalam melakukan analisis keuangan dan perencanaan anggaran secara akurat. Oleh karena itu, diperlukan solusi berbasis teknologi untuk meningkatkan efisiensi dan akurasi pencatatan. Pengembangan aplikasi mobile pencatatan keuangan dengan integrasi *Optical Character Recognition* (OCR) menjadi langkah inovatif untuk mengatasi permasalahan tersebut. Melalui teknologi ini, data dari struk pembelian dapat diekstraksi secara otomatis dan disimpan ke dalam basis data aplikasi, sehingga proses pencatatan keuangan menjadi lebih cepat, efisien, dan minim kesalahan.

Gambar 2. 4 Apotik Sinar Bintang 2

2.10 Studi Aplikasi Sejenis

Studi aplikasi sejenis dilakukan untuk mengetahui fitur-fitur yang telah tersedia pada aplikasi pencatatan keuangan yang sudah digunakan oleh masyarakat maupun pelaku usaha.

Analisis ini bertujuan untuk memperoleh referensi dalam pengembangan sistem serta mengidentifikasi kekurangan yang masih dapat diperbaiki pada aplikasi yang akan dibangun.

2.10.1 Aplikasi Buku Kas

Aplikasi Buku Kas merupakan platform pencatatan keuangan digital yang dirancang untuk mempermudah UMKM maupun individu dalam mendokumentasikan arus kas. Berdasarkan kajian penggunaannya, aplikasi ini memfasilitasi pencatatan pos pemasukan dan pengeluaran, manajemen multi-buku kas, pencarian transaksi, serta mekanisme *backup* dan *restore* data[53]. Keunggulan utama dari Buku Kas terletak pada kesederhanaan antarmuka grafisnya (*User Interface*) yang intuitif bagi pengguna awam.

Namun, dari perspektif fungsionalitas, platform ini masih memiliki keterbatasan dalam menangani manajemen keuangan berskala kompleks. Aplikasi ini belum mengadopsi sistem pembagian hak akses (*Role-Based Access Control*) dan belum mengintegrasikan teknologi otomasi seperti *Optical Character Recognition* (OCR) untuk efisiensi input data. Oleh karena itu, diperlukan pengembangan sistem yang lebih terstruktur guna mengakomodasi kebutuhan tata kelola keuangan yang spesifik.

Gambar 2. 5 Tampilan Aplikasi Buku Kas

2.10.2 Aplikasi Buku Warung

BukuWarung adalah aplikasi akuntansi digital yang berfokus pada digitalisasi pelaku usaha mikro, kecil, dan menengah (UMKM). Platform ini menyediakan fitur yang relatif komprehensif, mencakup pencatatan utang-piutang, pencatatan transaksi harian, serta pembentukan laporan keuangan berkala[54]. Selain fungsi pembukuan, BukuWarung mengintegrasikan ekosistemnya dengan fitur *fintech* seperti pembayaran digital, pengisian pulsa, dan PPOB (Payment Point Online Bank).

Kelengkapan fitur ini menjadi keunggulan utama BukuWarung dalam menyajikan data keuangan yang lebih detail. Kendati demikian, kepadatan fitur tersebut berdampak pada kompleksitas antarmuka, yang berpotensi menimbulkan *cognitive overload* (beban kognitif) bagi pengguna yang hanya membutuhkan fungsionalitas pembukuan esensial dan sederhana.

Gambar 2. 6 Tampilan Aplikasi Buku Warung

Untuk memetakan fungsionalitas dari masing-masing platform, berikut adalah tabel perbandingan fitur antara aplikasi Buku Kas dan Buku Warung:

Tabel 2. 3 Perbandingan Fitur Aplikasi Sejenis

No	Fitur	Buku Kas	Buku Warung
1	Pencatatan pemasukan	Tersedia	Tersedia
2	Pencatatan pengeluaran	Tersedia	Tersedia
3	Pencatatan Utang piutang	Tersedia	Tersedia
4	Laporan Keuangan	Sederhana	Lebih detail
5	Backup data	Tersedia	Tersedia

2.11 *Black Box Testing*

Black Box Testing yang juga dikenal sebagai *functional testing* merupakan pendekatan pengujian perangkat lunak yang mengevaluasi fungsionalitas sistem berdasarkan spesifikasi tanpa mempertimbangkan struktur internal kode program [55]. Dalam metode ini, tester hanya berinteraksi dengan antarmuka sistem melalui pemberian input dan memverifikasi output. Tujuan utama *black box testing* adalah memastikan bahwa perangkat lunak berjalan sesuai dengan spesifikasi kebutuhan fungsional yang telah ditetapkan. Pendekatan ini sangat cocok untuk menilai sistem dari sudut pandang pengguna akhir, karena yang diuji adalah bagaimana sistem merespon skenario penggunaan sehari-hari, bukan bagaimana kode di dalamnya bekerja [56].

Beberapa teknik umum yang digunakan dalam *black box testing* meliputi [57]:

1. *Equivalence Partitioning* adalah teknik pengujian yang membagi domain input ke dalam beberapa kelas ekivalen, baik valid maupun tidak valid. Setiap kelas diuji menggunakan satu nilai perwakilan untuk mengurangi jumlah kasus uji tanpa mengurangi efektivitas pengujian.
2. *Boundary Value Analysis* merupakan teknik pengujian yang menitikberatkan pada nilai batas suatu rentang input, seperti nilai minimum dan maksimum. Teknik ini digunakan karena kesalahan sering terjadi pada nilai ekstrem atau batas input.
3. *Decision Table Testing* digunakan untuk menguji sistem dengan kombinasi kondisi logis yang kompleks. Teknik ini menyajikan kondisi dan aksi dalam bentuk tabel keputusan untuk memastikan semua kemungkinan aturan bisnis telah diuji.
4. *State Transition Testing* merupakan teknik pengujian yang digunakan pada sistem yang perilakunya bergantung pada status sebelumnya. Pengujian dilakukan dengan memvalidasi transisi antar state berdasarkan event atau input tertentu.
5. *Cause-Effect Graphing* adalah teknik pengujian yang memetakan hubungan antara kondisi input (cause) dan output (effect). Teknik ini membantu mengidentifikasi kombinasi input yang memengaruhi perilaku sistem.
6. *Error Guessing* merupakan teknik pengujian yang bergantung pada pengalaman dan intuisi penguji untuk mengidentifikasi kemungkinan kesalahan yang tidak terdeteksi oleh teknik formal lainnya.
7. *Use Case Testing* adalah teknik pengujian yang didasarkan pada skenario penggunaan sistem oleh pengguna. Teknik ini digunakan untuk memastikan sistem berfungsi sesuai dengan kebutuhan pengguna dalam kondisi nyata.

8. *Comparison Testing* merupakan teknik pengujian yang membandingkan hasil keluaran dari dua atau lebih sistem atau versi sistem yang serupa untuk memastikan konsistensi dan kesesuaian hasil.

Secara umum, tahapan pengujian Black Box Testing meliputi beberapa langkah, yaitu [58]:

1. Menentukan fitur atau fungsi sistem yang akan diuji berdasarkan kebutuhan fungsional aplikasi.
2. Menyusun skenario pengujian dengan menentukan input yang akan diberikan kepada sistem.
3. Menentukan hasil yang diharapkan (expected output) berdasarkan spesifikasi sistem.
4. Melakukan pengujian dengan menjalankan sistem menggunakan data input yang telah ditentukan.
5. Membandingkan hasil aktual sistem dengan hasil yang diharapkan untuk mengetahui apakah fungsi berjalan dengan benar.
6. Mendokumentasikan hasil pengujian sebagai bahan evaluasi terhadap sistem yang dikembangkan.

2.12 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan tahap pengujian perangkat lunak yang dilakukan oleh pengguna akhir (end-user) untuk memverifikasi bahwa sistem yang dibangun telah berjalan sesuai dengan kebutuhan dan harapan pengguna sebelum sistem tersebut diluncurkan atau digunakan secara resmi [59]. UAT juga merupakan metode pengujian yang digunakan untuk mengukur kualitas sebuah sistem, yaitu dengan menilai sejauh mana sistem tersebut memberikan kemudahan dan kepuasan bagi penggunanya [60].

UAT biasanya dilaksanakan pada tahap akhir siklus pengembangan perangkat lunak, setelah pengujian teknis seperti *black box testing* selesai dilakukan oleh tim pengembang. Pada tahap ini, penilaian tidak lagi difokuskan pada aspek kode program, melainkan pada pengalaman nyata pengguna dalam mengoperasikan sistem, sehingga hasil pengujian mencerminkan tingkat penerimaan pengguna terhadap sistem yang telah dibangun [61].

Secara umum, UAT bertujuan untuk memastikan bahwa perangkat lunak yang dikembangkan benar-benar memenuhi kebutuhan fungsional serta ekspektasi pengguna sebelum sistem diimplementasikan secara luas [59]. Beberapa tujuan pelaksanaan UAT antara lain [62]:

1. Memverifikasi kesesuaian antara fitur-fitur sistem dengan kebutuhan pengguna yang telah dirumuskan sejak tahap analisis kebutuhan
2. Mengidentifikasi kekurangan, kesalahan, atau ketidaksesuaian sistem yang mungkin tidak terdeteksi pada tahap pengujian teknis sebelumnya
3. Mengukur tingkat kepuasan dan penerimaan pengguna terhadap sistem, baik dari sisi fungsionalitas, tampilan antarmuka, maupun kemudahan penggunaan.
4. Mengurangi risiko kegagalan sistem setelah diluncurkan ke lingkungan produksi
5. Menjadi dasar bagi pengguna untuk memberikan persetujuan resmi (*sign-off*) yang menandakan bahwa sistem telah siap digunakan secara operasional.

Pelaksanaan UAT umumnya dilakukan melalui beberapa tahapan sistematis. Tahapan tersebut dapat diuraikan sebagai berikut [59], [60], [62].

1. Perencanaan UAT (*UAT Planning*) yaitu identifikasi kebutuhan pengujian, penentuan ruang lingkup, serta penetapan skenario dan kriteria keberhasilan sistem.
2. Studi Literatur yaitu pengumpulan referensi dari buku, jurnal, dan publikasi ilmiah terkait untuk memperkuat landasan teori pengujian.
3. Penyusunan Instrumen dan Penentuan Responden yaitu penyusunan kuesioner yang berisi pernyataan terkait fungsionalitas, keandalan, kemudahan penggunaan, dan efisiensi sistem, serta penentuan responden dari kalangan pengguna akhir yang relevan.
4. Pelaksanaan Pengujian dimana responden menjalankan sistem secara langsung dan mengisi kuesioner berdasarkan pengalaman mereka.
5. Pengolahan dan Analisis Data yaitu data hasil kuesioner diolah menggunakan perhitungan skala Likert untuk memperoleh nilai interpretasi tingkat penerimaan pengguna.
6. Penarikan Kesimpulan yaitu hasil analisis digunakan untuk menentukan apakah sistem telah memenuhi kriteria penerimaan pengguna; apabila hasilnya baik, pengguna dapat memberikan *sign-off* sebagai tanda sistem layak dan siap digunakan.

Hasil pengujian UAT yang telah dikumpulkan melalui kuesioner selanjutnya dihitung menggunakan pendekatan skala Likert untuk memperoleh nilai interpretasi berupa persentase. langkah-langkah perhitungannya adalah sebagai berikut [63] :

Pertama, menghitung skor tiap jawaban dengan rumus:

$$Skor = T \times Pn$$

dengan T adalah jumlah responden yang memilih suatu kategori jawaban, dan Pn adalah bobot/nilai pada skala Likert untuk kategori jawaban tersebut. Kedua, menentukan skor tertinggi (Y) dan skor terendah (X):

$$Y = Skor\ tertinggi\ likert \times Jumlah\ responden$$

$$X = Skor\ terendah\ likert \times Jumlah\ responden$$

Ketiga, menghitung nilai interpretasi (Index %):

$$Rumus\ Index\% = \frac{Total\ Skor}{Y} \times 100$$

Keempat, menghitung interval interpretasi:

$$I = \frac{100}{Jumlah\ skor(likert)}$$

Nilai interval (I) ini digunakan untuk menyusun kategori interpretasi dari hasil persentase yang diperoleh. Apabila menggunakan skala Likert 1–5, interval yang dihasilkan adalah 20, sehingga kategori interpretasi terbagi menjadi lima rentang, misalnya "Sangat Tidak Setuju", "Tidak Setuju", "Netral/Cukup", "Setuju", hingga "Sangat Setuju". Nilai akhir persentase inilah yang kemudian digunakan sebagai dasar untuk menyimpulkan tingkat penerimaan pengguna terhadap sistem yang diuji.

Skala Likert sendiri adalah skala psikometrik yang umum digunakan dalam penelitian untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok responden terhadap suatu fenomena atau objek tertentu, termasuk dalam konteks pengujian penerimaan pengguna terhadap sebuah sistem [59]. Responden diminta memberikan penilaian terhadap suatu pernyataan dengan memilih salah satu tingkatan jawaban, mulai dari yang bersifat sangat negatif hingga sangat positif, misalnya "Sangat Tidak Setuju" hingga "Sangat Setuju". Setiap tingkatan jawaban tersebut diberi bobot nilai numerik sebagaimana ditunjukkan pada tabel berikut [63].

Tabel 2. 4 Kategori Skala Likert

Kategori Jawaban	Bobot/Skor
Sangat Setuju (SS)	5
Setuju (S)	4
Netral/Cukup (N)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

Penggunaan skala Likert pada UAT memungkinkan hasil penilaian pengguna yang bersifat kualitatif (persepsi dan opini) diubah menjadi data kuantitatif berupa skor, sehingga tingkat penerimaan pengguna terhadap sistem dapat diukur dan dianalisis secara lebih objektif dan terukur [60].

2.13 Pengujian Akurasi Ekstraksi Informasi OCR

Pengujian akurasi ekstraksi informasi OCR dilakukan untuk mengukur kemampuan sistem dalam mengekstraksi data transaksi dari nota pengeluaran ke dalam format data terstruktur. Evaluasi dilakukan dengan membandingkan hasil ekstraksi sistem terhadap data ground truth yang telah dibuat dan diverifikasi secara manual.

Salah satu metode evaluasi yang umum digunakan pada tugas *Information Extraction* dan *Document Understanding* adalah *Exact Match Accuracy* (EMA). *Exact Match Accuracy* merupakan metrik yang mengukur tingkat kesesuaian hasil ekstraksi terhadap data referensi (*ground truth*), dimana suatu hasil dianggap benar hanya apabila identik dengan nilai referensi tanpa memberikan nilai parsial terhadap hasil yang hampir benar. Pendekatan ini banyak digunakan pada penelitian ekstraksi informasi dokumen karena setiap field harus memiliki nilai yang tepat agar dapat digunakan pada proses bisnis selanjutnya [64], [65].

Pada penelitian ini, konsep *Exact Match Accuracy* diterapkan pada level field sehingga disebut sebagai *Field Extraction Accuracy*. Setiap field dinyatakan benar apabila nilai hasil ekstraksi identik dengan nilai pada ground truth setelah proses OCR, parsing, dan post-processing selesai dilakukan. Field yang dievaluasi meliputi vendor, tanggal, grand total, nama barang, kuantitas (qty), satuan, harga satuan, dan total harga barang.

Field Extraction Accuracy dihitung menggunakan Persamaan (2.x).

$$Field\ Extraction\ Accuracy = \frac{Jumlah\ Field\ Benar}{Jumlah\ Field\ Ground\ Truth} \times 100\%$$

Dengan Jumlah Field Benar adalah jumlah field yang nilai hasil ekstraksinya identik dengan data ground truth. Dan Jumlah Seluruh Field adalah total field yang menjadi objek evaluasi. Semakin tinggi nilai *Field Extraction Accuracy* yang diperoleh, maka semakin baik kemampuan sistem dalam mengekstraksi informasi transaksi dari nota pengeluaran. Pada penelitian ini, evaluasi dilakukan menggunakan pendekatan *Exact Match Accuracy* sehingga setiap field yang berbeda, meskipun hanya satu karakter, akan dianggap sebagai kesalahan ekstraksi.

Selain menghitung akurasi pada setiap field, penelitian ini juga menggunakan *Overall Item Accuracy* untuk menggambarkan kemampuan sistem secara keseluruhan dalam mengekstraksi informasi pada item transaksi. Nilai ini diperoleh dengan menghitung rata-rata akurasi dari lima field yang terdapat pada setiap item transaksi, yaitu nama barang, kuantitas (qty), satuan, harga satuan, dan total harga. Field pada bagian header, yaitu vendor, tanggal, dan grand total, tidak termasuk dalam perhitungan *Overall Item Accuracy* karena dievaluasi secara terpisah sebagai akurasi informasi header.

$$Overall\ Item\ Accuracy = \frac{A_{nama\ barang} + A_{qty} + A_{satuan} + A_{harga} + A_{total}}{5} \times 100\%$$

Semakin tinggi nilai *Overall Item Accuracy*, semakin baik kemampuan sistem dalam mengekstraksi informasi item transaksi dari nota pengeluaran. Metrik ini digunakan untuk memberikan gambaran performa sistem secara keseluruhan pada proses ekstraksi item transaksi tanpa dipengaruhi oleh akurasi informasi pada bagian header.