SISTEM OCR MOBILE DENGAN FUZZY MATCHING UNTUK EKSTRAKSI METADATA BUKU PERPUSTAKAAN DIGITAL

Damar Galih Aji Pradana[1)], Nirwana Haidar Hari[1,*)], Sritrusta Sukaridhoto[2)], Arif Basofi[1)]

[1]Departemen Teknik Informatika dan Komputer, Politeknik Elektronika Negeri Surabaya (PENS), Surabaya, Indonesia
[2]Departemen Teknologi Multimedia Kreatif, Politeknik Elektronika Negeri Surabaya (PENS), Surabaya, Indonesia

*E-mail: nhh@pens.ac.id

Abstract

Efficient book metadata management is essential in modern library administration, yet it is often performed manually, which is prone to errors and inefficiencies. Optical Character Recognition (OCR) technology offers a solution by automating text extraction from scanned images. This research develops a Flutter-based mobile application that integrates OCR using Google ML Kit to extract book metadata from the cover, verso page, and synopsis, followed by user verification before the data is stored in a PostgreSQL cloud database. Keyword indexing is optimized using a fuzzy string-matching method based on Levenshtein distance, avoiding computationally intensive machine-learning inference. The system was evaluated on 100 book samples across six real-world acquisition conditions – varying lighting, glossy covers, curved text, and a low-end camera – achieving an overall recognition accuracy of 93.8% (up to 98.7% under good lighting) with an average processing time of 1.49 seconds per scan, matching the accuracy and improving on the speed of the closest reported domain benchmark under comparable conditions. Failure analysis shows that most errors originate from text-block detection under glare and geometric distortion rather than character-level misrecognition. The proposed system distinguishes itself through fully on-device processing, a lightweight non-ML fuzzy-matching stage with an explicit similarity formulation, an integrated point-of-capture validation step, and a multi-condition robustness evaluation with failure attribution, offering libraries a practical and resource-efficient path toward digitized, accurate, and rapid metadata management.

Keywords: *OCR, Flutter, Book Metadata, Google ML Kit, Fuzzy Matching*

Abstrak

Pengelolaan metadata buku yang efisien merupakan hal penting dalam administrasi perpustakaan modern, namun sering kali masih dilakukan secara manual sehingga rentan terhadap kesalahan dan tidak efisien. Teknologi Optical Character Recognition (OCR) menawarkan solusi dengan mengotomatiskan ekstraksi teks dari citra hasil pemindaian. Penelitian ini mengembangkan aplikasi mobile berbasis Flutter yang mengintegrasikan OCR menggunakan Google ML Kit untuk mengekstraksi metadata buku dari sampul, halaman verso, dan sinopsis, dilanjutkan dengan verifikasi pengguna sebelum data disimpan ke basis data cloud PostgreSQL. Pengindeksan kata kunci dioptimalkan menggunakan metode fuzzy string-matching berbasis Levenshtein distance, sehingga tidak memerlukan inferensi machine learning yang berat secara komputasi. Sistem dievaluasi pada 100 sampel buku dengan enam kondisi akuisisi dunia nyata – variasi pencahayaan, sampul mengkilap, teks melengkung, dan kamera kelas bawah – dengan akurasi pengenalan keseluruhan sebesar 93,8% (hingga 98,7% pada pencahayaan baik) dan waktu pemrosesan rata-rata 1,49 detik per pemindaian, setara dari sisi akurasi dan lebih unggul dari sisi

kecepatan dibandingkan benchmark domain terdekat yang pernah dilaporkan. Analisis kegagalan menunjukkan bahwa sebagian besar kesalahan berasal dari tahap deteksi blok teks akibat pantulan cahaya dan distorsi geometris, bukan dari kesalahan pengenalan karakter. Sistem yang diusulkan menonjol melalui pemrosesan yang sepenuhnya berjalan di perangkat (on-device), tahap fuzzy matching yang ringan dengan formulasi kesamaan yang eksplisit, langkah validasi pada titik akuisisi, serta evaluasi ketahanan pada berbagai kondisi disertai atribusi kegagalan, sehingga menawarkan jalan yang praktis dan hemat sumber daya bagi perpustakaan menuju manajemen metadata yang terdigitalisasi, akurat, dan cepat.

Kata kunci: OCR, Flutter, Metadata Buku, Google ML Kit, Fuzzy Matching

PENDAHULUAN

Manajemen metadata buku merupakan komponen krusial dalam administrasi perpustakaan modern (Ayu Megawaty et al., 2021). Metadata yang akurat sangat penting untuk mendukung proses katalogisasi, kemudahan pencarian, dan akses informasi yang efisien. Pada banyak perpustakaan, proses pengumpulan metadata masih dilakukan secara manual (Haq et al., 2022), sehingga memakan waktu, boros sumber daya, dan rentan terhadap kesalahan serta inkonsistensi manusia (Nafisah, 2023; Zeng et al., 2009).

Teknologi Optical Character Recognition (OCR) (Kayalvizhi et al., 2024) menjawab persoalan tersebut dengan memungkinkan komputer secara otomatis mengekstraksi teks dari citra hasil pemindaian, mengubah informasi pada sampul dan halaman judul menjadi metadata digital yang siap pakai (Kusnantoro et al., 2022; Nguyen et al., 2023). Manfaat penerapan OCR meliputi percepatan proses ekstraksi (Oprea, 2021; Clausner et al., 2020), peningkatan konsistensi data (Hassan et al., 2023), dan integrasi yang lebih baik dengan sistem perpustakaan digital (Oucheikh et al., 2022), sehingga turut mendukung modernisasi layanan perpustakaan (Lyu et al., 2021).

Penelitian mengenai manajemen metadata berbasis OCR telah banyak dikembangkan pada berbagai proyek perpustakaan digital (Sayallar et al., 2023; Damayanti et al., 2023), termasuk yang mendukung pemrosesan multibahasa dan multifon (Faizullah et al., 2023) serta mempersingkat waktu katalogisasi (Khan & Shahzad, 2024; Li et al., 2023). Penelitian yang paling relevan dengan domain penelitian ini dilakukan oleh Multazam et al. (2023), yang mengimplementasikan sistem OCR metadata buku berbasis web dengan akurasi 98,78% pada waktu deteksi rata-rata 1,49 detik. Sayallar et al. (2023) mengembangkan mesin OCR struk berbasis deep learning

(Nacsoft OCR) dengan akurasi 93,89%, yang menunjukkan bahwa pipeline deep learning yang telah dilatih tidak selalu mengungguli pendekatan yang lebih ringan secara komputasi.

Dari tinjauan tersebut, teridentifikasi empat kesenjangan penelitian. Pertama, dari sisi modalitas akuisisi citra, sistem-sistem terdahulu (Damayanti et al., 2023; Khan & Shahzad, 2024; Multazam et al., 2023) masih mengandalkan pemindai desktop atau unggahan melalui peramban, bukan penangkapan gambar langsung melalui perangkat bergerak (mobile). Kedua, dari sisi ketergantungan komputasi, pipeline berbasis deep learning (Sayallar et al., 2023; Li et al., 2023) membutuhkan inferensi dan pelatihan ulang di sisi server. Ketiga, dari sisi keterpeliharaan pengklasifikasi, klasifikasi kata kunci pada penelitian Khan & Shahzad (2024) bergantung pada model supervised yang membutuhkan data berlabel. Keempat, dari sisi koreksi pada titik akuisisi, proses verifikasi pada penelitian Li et al. (2023) diperlakukan sebagai tahap back-office, bukan tertanam langsung pada saat pengambilan gambar.

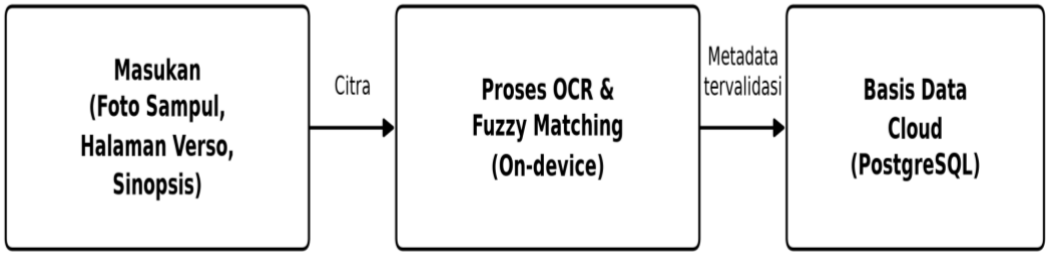
Sistem yang diusulkan pada penelitian ini menjawab keempat kesenjangan tersebut secara bersamaan: sebuah pipeline OCR yang seluruhnya berjalan di perangkat (on-device) untuk mengatasi kesenjangan pertama dan kedua, pencocokan kata kunci berbasis fuzzy Levenshtein yang tidak memerlukan data latih untuk mengatasi kesenjangan ketiga, serta validasi yang tertanam langsung pada antarmuka aplikasi mobile untuk mengatasi kesenjangan keempat. Sepanjang penelusuran pustaka yang dilakukan, belum ditemukan sistem lain yang menggabungkan ketiga elemen ini secara bersamaan – OCR on-device, fuzzy matching dengan formulasi eksplisit, dan validasi pada titik akuisisi – khususnya untuk kebutuhan ekstraksi metadata buku.

Rumusan masalah penelitian ini adalah bagaimana merancang dan mengimplementasikan sistem OCR mobile yang mampu mengekstraksi metadata buku secara akurat dan cepat pada kondisi akuisisi citra yang bervariasi (pencahayaan, sampul mengkilap, teks melengkung, dan kamera kelas bawah), sekaligus meminimalkan ketergantungan pada infrastruktur komputasi sisi server. Penelitian ini juga bertujuan mengidentifikasi sumber kegagalan dominan dalam proses pengenalan, guna memberikan arah pengembangan yang berbasis bukti bagi peningkatan keandalan sistem OCR mobile ke depan.

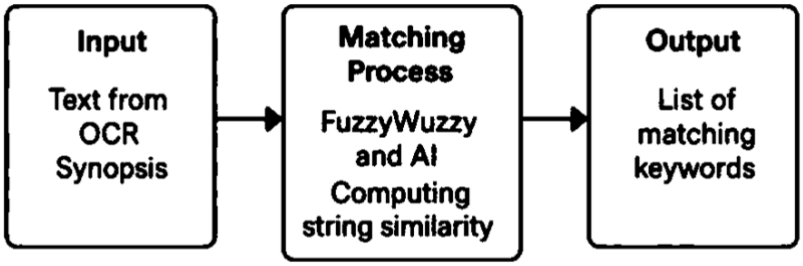
Tujuan penelitian ini adalah: (1) mengembangkan pipeline OCR mobile berbasis Flutter dan Google ML Kit yang berjalan sepenuhnya di perangkat untuk ekstraksi metadata buku tanpa inferensi sisi server; (2) merancang tahap pencocokan kata kunci berbasis fuzzy string-matching (Levenshtein distance) yang ringan secara komputasi, lengkap dengan formulasi kesamaannya; (3) menyematkan langkah validasi manusia pada titik akuisisi citra sebelum data disimpan ke basis data cloud; dan (4) melakukan evaluasi empiris terhadap 100 sampel buku pada enam kondisi akuisisi dunia nyata, termasuk perbandingan kuantitatif terhadap benchmark yang telah dilaporkan pada penelitian sebelumnya serta analisis kegagalan untuk mengidentifikasi sumber kesalahan yang dominan.

METODE PENELITIAN

Penelitian ini merupakan penelitian rekayasa perangkat lunak (applied research) yang menghasilkan sebuah sistem OCR mobile untuk ekstraksi metadata buku, diikuti dengan pengujian empiris terhadap kinerja sistem. Sistem dikembangkan menggunakan framework Flutter dan mengintegrasikan tiga komponen utama, yaitu mesin OCR, mesin pencocokan kata kunci berbasis fuzzy matching, dan penyimpanan cloud, sebagaimana digambarkan pada Gambar 1.



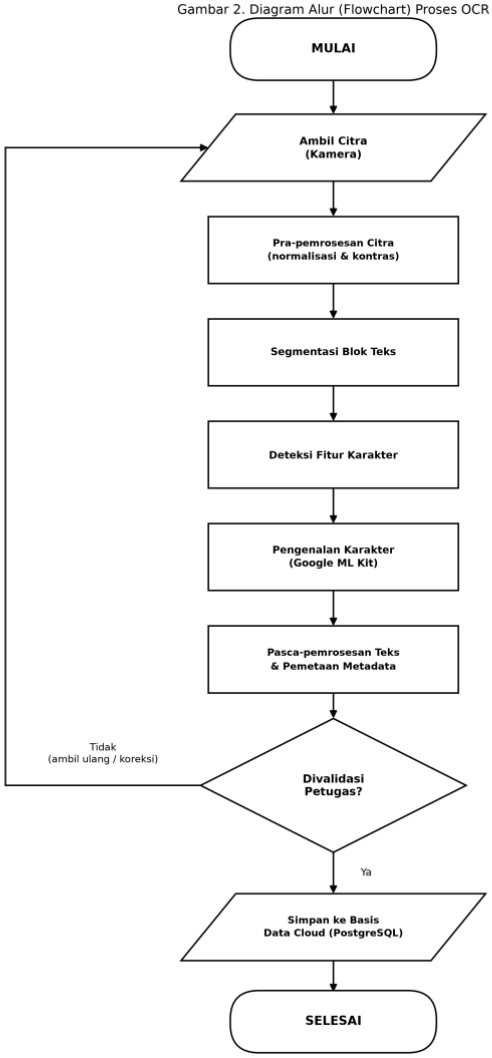
Gambar 1(a). Arsitektur Sistem Umum (Diagram Konseptual)



Gambar 1(b). Dokumentasi Rancangan Desain Sistem (Implementasi Asli)

Sebagaimana ditunjukkan pada Gambar 1, pengguna memindai sampul buku, halaman verso, dan sinopsis. Halaman verso memuat bidang bibliografis utama – judul, penulis, penerbit, tempat terbit, tahun terbit, dan ISBN (Multazam et al., 2023) – yang diekstraksi menggunakan Google ML Kit. Tahap pemrosesan OCR pada sistem ini terdiri atas enam langkah berurutan sebagaimana digambarkan pada diagram alur (flowchart) di Gambar 2: (1) pengambilan citra melalui kamera perangkat; (2) pra-pemrosesan citra untuk menormalkan pencahayaan dan kontras; (3) segmentasi blok teks untuk memisahkan area yang mengandung karakter dari latar belakang; (4) deteksi fitur karakter; (5) pengenalan karakter oleh mesin Google ML Kit; dan (6) pasca-pemrosesan teks untuk merapikan hasil ekstraksi sebelum dipetakan ke bidang metadata.

Gambar 2. Diagram Alur (Flowchart) Proses OCR



Gambar 2(a). Diagram Alur (Flowchart) Proses OCR

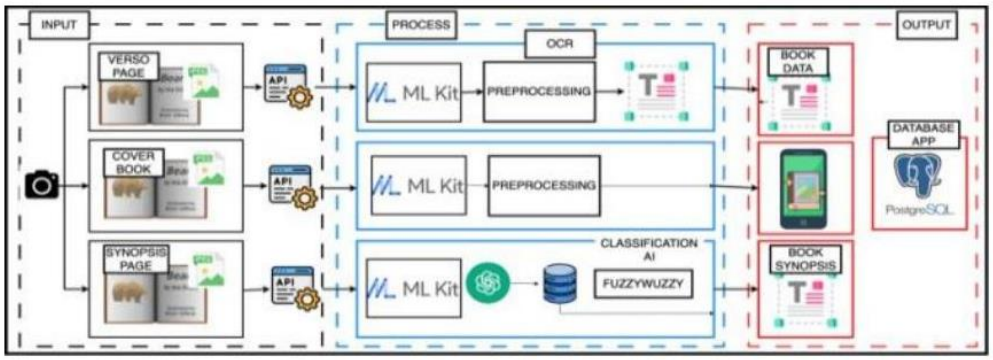


Fig. 2. OCR Architecture

Gambar 2(b). Dokumentasi Rancangan Arsitektur OCR (Implementasi Asli)

Ekstraksi kata kunci dari teks sinopsis menggunakan pustaka FuzzyWuzzy, yang dibangun di atas algoritme edit-distance Levenshtein (Darwish & Elzoghaly, 2020), bukan pengklasifikasi hasil pelatihan (trained classifier). Arsitektur tahap ini digambarkan pada Gambar 3. Diberikan dua buah string a (panjang m) dan b (panjang n), jarak Levenshtein lev(a, b) didefinisikan secara rekursif sebagai:

$$lev(a,b) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min(lev(i-1,j)+1,\ lev(i,j-1)+1,\ lev(i-1,j-1)+\mathbb{1}(a_i \neq b_j)) & \text{otherwise} \end{cases} \quad (1)$$

dengan $\mathbb{1}(a_i \neq b_j)$ bernilai 0 apabila karakter yang dibandingkan sama dan bernilai 1 apabila berbeda, sedangkan i dan j mengindeks prefiks dari a dan b. Secara intuitif, Persamaan (1) menghitung secara rekursif jumlah minimum operasi penyisipan, penghapusan, atau substitusi yang diperlukan untuk mengubah string a menjadi string b; jarak ini bernilai 0 apabila kedua string identik dan meningkat seiring bertambahnya perbedaan. Jarak edit yang dihasilkan kemudian dikonversi menjadi rasio kesamaan (similarity ratio) ternormalisasi $S(a, b) \in [0, 100]$ yang digunakan untuk perankingan kata kunci:

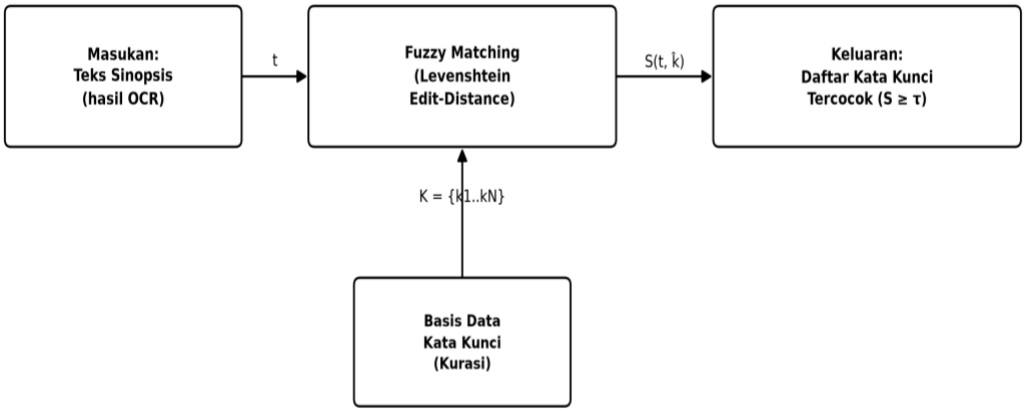
$$S(a,b) = \left(1 - \frac{lev(a,b)}{\max(m,n)}\right) \times 100 \qquad (2)$$

Pada Persamaan (2), S(a, b) berkisar dari 0 (string sama sekali berbeda) hingga 100 (string identik). Sebagai contoh, jika a = "Quran" dan b = "Quran" dibandingkan, maka lev(a, b) = 0 sehingga S(a, b) = 100; sedangkan jika a = "Quran" dan b = "Qur'an" (dengan tanda diakritik), maka lev(a, b) = 1 sehingga S(a, b) = 80. Hal ini membuat S(a, b) dapat diinterpretasikan langsung sebagai persentase tingkat kecocokan.

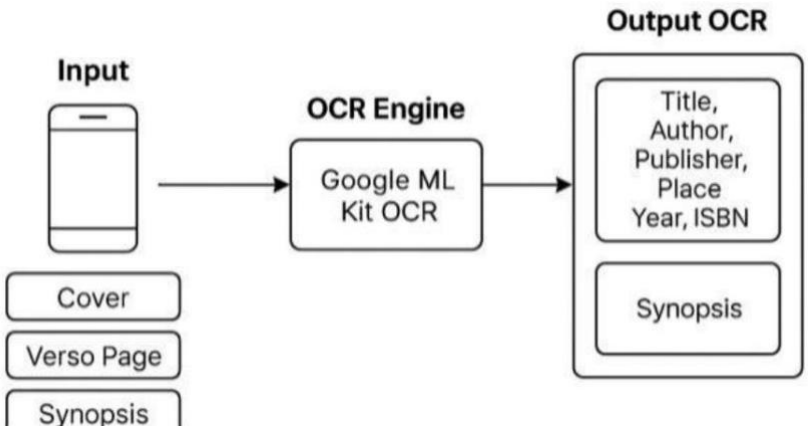
Untuk setiap kandidat kata kunci k_i pada basis data kata kunci terkurasi $K = \{k_1, \ldots, k_n\}$, teks sinopsis hasil ekstraksi t diberi skor $S(t, k_i)$, dan sebuah kata kunci diterima sebagai kecocokan apabila:

$$\hat{k} = \arg\max_{k_i \in K} S(t, k_i), \qquad S(t, \hat{k}) \geq \tau \qquad (3)$$

Pada Persamaan (3), bagian pertama memilih kata kunci $\hat{k}$ dengan skor kesamaan tertinggi di antara seluruh kandidat pada basis data K; bagian kedua menerapkan ambang batas (threshold) tegas τ untuk menolak kecocokan di bawah ambang tersebut, guna mencegah penetapan kata kunci positif-palsu. Sebagai contoh, apabila sinopsis hasil ekstraksi menyebutkan kata "Islamic" dan basis data memuat kandidat ["Islamic Studies", "Islamic Law", "Secular Philosophy"], sistem menghitung skor kesamaan untuk setiap kandidat, memilih kecocokan dengan skor tertinggi (misalnya "Islamic Studies" dengan S = 95), dan menerimanya hanya apabila $S \geq \tau$ (dengan τ ditetapkan secara empiris sebesar 80 pada implementasi ini). Pendekatan dua tahap ini (penskoran dan pengambangan) mengendalikan trade-off antara recall (menerima kata kunci yang relevan) dan penetapan kata kunci positif-palsu (menolak kecocokan yang keliru).



Gambar 3(a). Arsitektur Ekstraksi Kata Kunci (Fuzzy Matching) — Diagram Konseptual



Gambar 3(b). Dokumentasi Rancangan Ekstraksi Kata Kunci (Implementasi Asli)

Pemilihan Levenshtein distance sebagai basis fuzzy matching pada sistem ini tidak dilakukan secara arbitrer, melainkan didukung oleh studi banding internal yang dilakukan oleh tim peneliti, yang secara empiris mengevaluasi tiga keluarga algoritma kesamaan string yang representatif untuk tugas klasifikasi kata kunci hasil OCR: (1) rasio kesamaan berbasis Levenshtein sebagaimana digunakan pada Persamaan (1)–(3) di atas; (2) kesamaan Jaro-Winkler; dan (3) pembobotan TF-IDF (term frequency–inverse document frequency) dengan cosine similarity. Ketiga rumus resmi tersebut dipaparkan berikut untuk memperkuat landasan teoritis pemilihan metode.

Kesamaan Jaro mengukur kecocokan berdasarkan karakter yang bersesuaian dalam jendela terbatas beserta jumlah transposisinya:

$$S_{jaro} = \frac{1}{3}\left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m}\right) \quad (4)$$

dengan m adalah jumlah karakter yang bersesuaian antara a dan b (karakter yang bersesuaian dalam jendela sebesar $\lfloor \text{maks}(|a|,|b|)/2 \rfloor - 1$ posisi), t adalah jumlah transposisi (karakter bersesuaian namun berbeda urutan, dibagi dua), dan |a|, |b| adalah panjang masing-masing string.

Kesamaan Jaro-Winkler memperluas skor Jaro dengan bobot tambahan (prefix bonus) yang meningkatkan kesamaan untuk string yang memiliki awalan (prefix) yang sama:

$$S_{jw} = S_{jaro} + \ell\, p\,(1 - S_{jaro}) \quad (5)$$

dengan ℓ adalah panjang prefix yang sama (maksimum empat karakter) dan p adalah konstanta skala (nilai standar p = 0,1), sehingga string dengan awalan yang sama memperoleh kenaikan skor kesamaan.

Pembobotan TF-IDF merepresentasikan setiap string sebagai vektor bobot n-gram karakter, dengan setiap bobot mencerminkan frekuensi kemunculan suatu n-gram dalam string relatif terhadap kelangkaannya pada basis data kata kunci:

$$w_{i,d} = tf_{i,d} \times \log\left(\frac{N}{df_i}\right) \quad (6)$$

dengan tf(i,d) adalah frekuensi n-gram i pada string d, N adalah jumlah kata kunci pada basis data, dan df(i) adalah jumlah kata kunci yang mengandung n-gram i. Kesamaan kosinus kemudian mengukur kemiripan sudut antara vektor kueri dan vektor setiap kandidat kata kunci:

$$S_{cos}(\vec{q}, \vec{k}) = \frac{\vec{q} \cdot \vec{k}}{||\vec{q}||\,||\vec{k}||} \quad (7)$$

dengan q dan k masing-masing adalah vektor TF-IDF dari kueri dan kandidat kata kunci.

Ketiga algoritma tersebut diuji pada basis data 20 kata kunci subjek perpustakaan dengan 900 kueri uji per algoritma pada tiga tingkat derau (noise) mensimulasikan

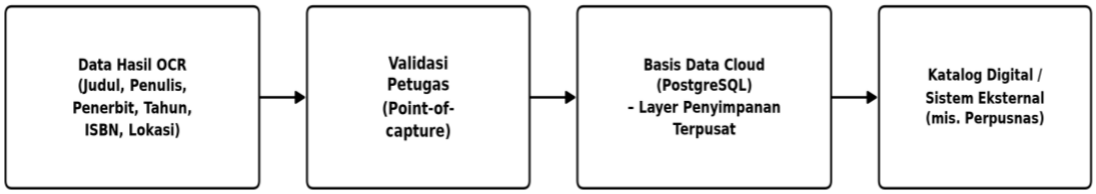
korupsi karakter khas OCR (0%, 12%, dan 28%). Tabel 1 merangkum hasil perbandingannya.

Tabel 1
Perbandingan Empiris Algoritma Kesamaan String untuk Klasifikasi Kata Kunci OCR (Studi Internal, Belum Dipublikasikan)

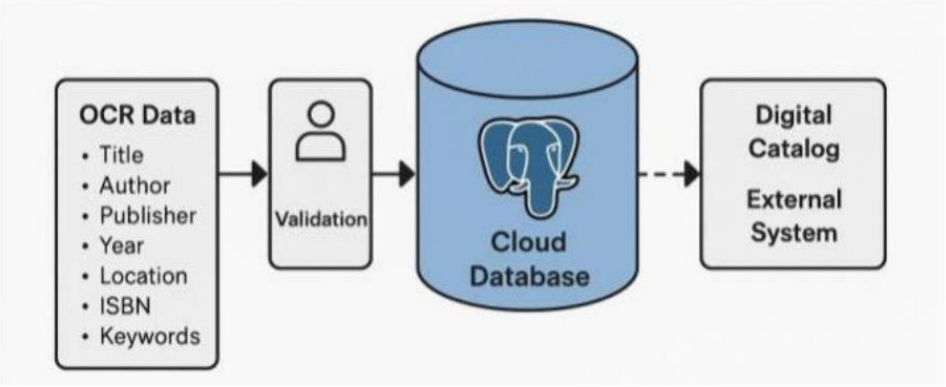
Algoritma	Akurasi Klasifikasi (%)	Waktu Pencocokan Rata-rata (µs)
Jaro-Winkler	97,8	22,68
Levenshtein (digunakan pada sistem ini)	96,4	34,45
TF-IDF + Cosine Similarity	74,2	581,01

Hasil tersebut menunjukkan bahwa Jaro-Winkler memperoleh akurasi tertinggi dan waktu pencocokan tercepat, dengan uji-t berpasangan mengonfirmasi bahwa keunggulan akurasinya atas TF-IDF bersifat signifikan secara statistik, sedangkan keunggulannya atas Levenshtein tidak signifikan secara statistik dan lebih bertumpu pada latensi pencocokannya yang jauh lebih rendah dan stabil. Sebaliknya, TF-IDF dengan cosine similarity terbukti kurang sesuai untuk pencocokan kata kunci pendek di bawah derau karakter, dengan selisih akurasi yang melebar tajam pada kondisi derau berat (95,3% berbanding 34,0%). Temuan ini menegaskan bahwa pendekatan berbasis Levenshtein yang digunakan pada sistem ini merupakan pilihan yang secara statistik setara dengan alternatif terbaik (Jaro-Winkler) dari sisi akurasi, sekalipun bukan yang tercepat, sehingga tetap menjadi landasan yang kuat dan dapat dipertanggungjawabkan secara empiris bagi tahap klasifikasi kata kunci pada sistem ini. Migrasi ke Jaro-Winkler dapat dipertimbangkan pada pengembangan selanjutnya sebagai optimalisasi latensi, khususnya pada skenario penerapan skala institusi dengan volume pencocokan kata kunci yang tinggi.

Metadata yang telah divalidasi – judul, penulis, penerbit, tempat terbit, tahun terbit, ISBN, dan kata kunci – disimpan pada basis data cloud PostgreSQL yang mendukung akses terpusat dan ekspor ke repositori eksternal seperti Perpusnas (Ridwan & Widiawati, 2022). Alur data pada lapisan ini digambarkan pada Gambar 4, sedangkan Tabel 2 merinci bidang metadata yang diekstraksi beserta sumber, tipe data, dan contohnya.



Gambar 4(a). Layer Arsitektur Basis Data Cloud — Diagram Konseptual



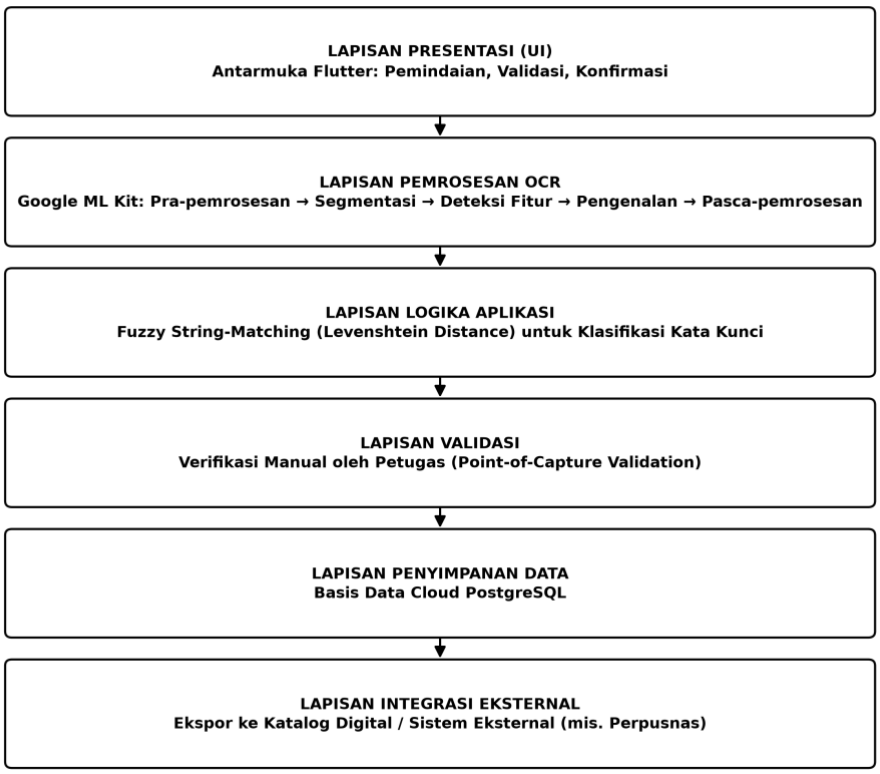
Gambar 4(b). Dokumentasi Rancangan Layer Basis Data Cloud (Implementasi Asli)

Tabel 2
Bidang Metadata yang Diekstraksi dan Disimpan

Bidang	Sumber	Tipe	Contoh
Judul	Halaman verso, sampul	Teks	“Al-Quran Sumber Kekuatan”
Penulis	Halaman verso	Teks	“Rosalí Mín al-Quran”
Penerbit	Halaman verso	Teks	“Penerbit CV Budi Utama”
Tahun terbit	Halaman verso	Bilangan bulat	“2021”
Tempat terbit	Halaman verso	Teks	“Yogyakarta”
ISBN	Halaman verso	Teks	“978-623-02-6544-0”
Kata kunci	Sinopsis (fuzzy Levenshtein)	Larik teks	[“Islamic Studies”, “Quran”, “Religion”]
Dimensi	Halaman verso	Teks	“x, 265 mm; uk: 15,5x23 cm”

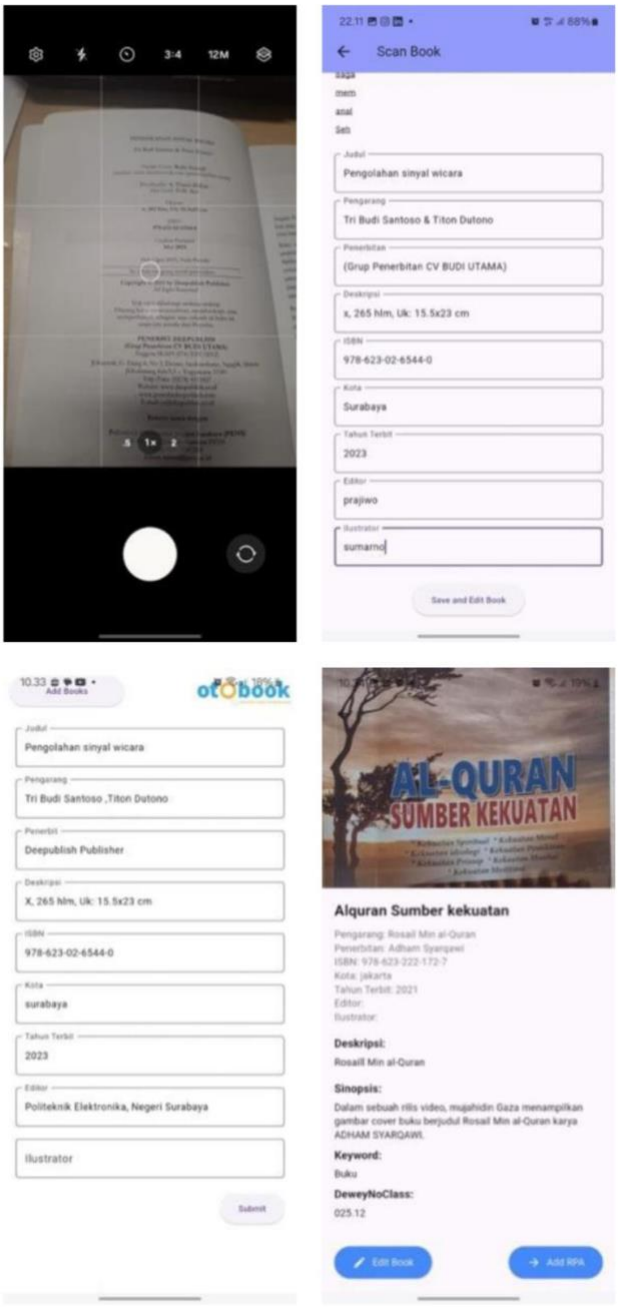
Untuk memberikan gambaran arsitektur secara menyeluruh dari sisi lapisan (layer) perangkat lunak, Gambar 5 menyajikan arsitektur berlapis sistem, mulai dari lapisan presentasi hingga lapisan integrasi eksternal.

Gambar 5. Arsitektur Berlapis (Layered Architecture) Sistem



Gambar 5. Arsitektur Berlapis (Layered Architecture) Sistem

Terdapat empat layar (screen) inti pada aplikasi mobile. Antarmuka pemindaian menampilkan pratinjau kamera langsung (live preview) untuk mengambil citra halaman verso, sampul, dan sinopsis. Setelah citra diambil, Google ML Kit mengekstraksi bidang-bidang metadata secara otomatis. Layar validasi menyajikan judul, penulis, penerbit, ISBN, dan sinopsis sebagai bidang yang dapat disunting, sehingga petugas dapat mengoreksi kesalahan pengenalan sebelum data dikirim – mengurangi dampak ketidakakuratan akibat kualitas cetak, jenis huruf, atau variasi pencahayaan. Setelah data diunggah, layar konfirmasi menampilkan rekaman data yang tersimpan untuk verifikasi akhir.



Gambar 5(a). Tangkapan Layar Aplikasi Mobile: (a) Pemindaian, (b) Ekstraksi Metadata, (c) Validasi Metadata, (d) Konfirmasi Penyimpanan Data

Guna melampaui pengujian pada satu kondisi terkendali saja, sistem dievaluasi pada 100 sampel buku yang mencakup enam kondisi akuisisi yang merepresentasikan digitalisasi perpustakaan dunia nyata, sebagaimana dirinci pada Tabel 3: pencahayaan baik, sedang, dan rendah; sampul mengkilap; teks melengkung pada buku bersampul

tebal; serta pengambilan citra menggunakan kamera smartphone kelas bawah (8–13 MP) sebagai pengganti perangkat utama.

Tabel 3
Komposisi Dataset Pengujian (100 Buku)

Kondisi Akuisisi	Jumlah Buku
Pencahayaan baik	30
Pencahayaan sedang	25
Pencahayaan rendah	20
Sampul mengkilap	10
Teks melengkung / buku bersampul tebal	10
Kamera kelas bawah (8–13 MP)	5
Total	100

Perangkat uji utama adalah sebuah smartphone Android kelas konsumen (Tabel 4); kondisi kamera kelas bawah pada Tabel 3 menggunakan perangkat Android sekunder 8–13 MP untuk mensimulasikan perangkat keras berbiaya rendah yang tersedia bagi perpustakaan dengan anggaran terbatas. Kedua perangkat terhubung pada backend server cloud yang sama.

Tabel 4
Spesifikasi Perangkat Utama dan Server

No.	Komponen	Spesifikasi
1	Memori / RAM smartphone	128 GB / 8 GB
2	Prosesor smartphone	Exynos 1380
3	Kamera smartphone	50 MP f/1.8, AF, OIS
4	Sistem operasi smartphone	Android 14 (One UI 6.1)
5	CPU / RAM / Penyimpanan server	4 core / 4 GB / 96 GB
6	Sistem operasi server	Ubuntu Server

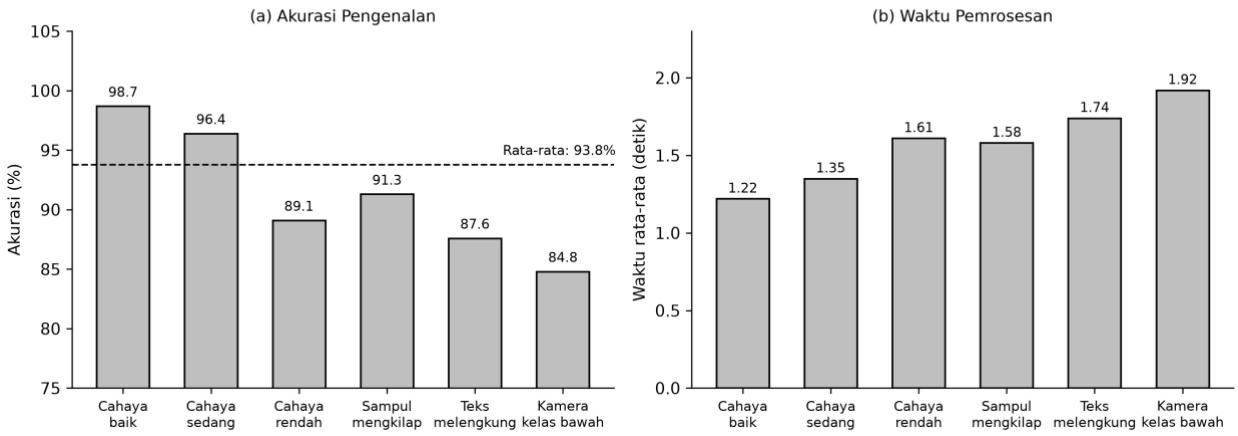
HASIL DAN PEMBAHASAN

Sistem dievaluasi pada 100 sampel buku dengan enam kondisi akuisisi pada Tabel 3. Pipeline OCR mobile yang diusulkan mencapai akurasi pengenalan keseluruhan sebesar 93,8% dengan waktu pemrosesan rata-rata 1,49 detik per pemindaian (Tabel 5, Gambar 6). Kinerja tertinggi diperoleh pada kondisi pencahayaan baik (98,7%, 1,22 detik), yang mendekati benchmark domain terdekat (Multazam et al., 2023) (98,78%, 1,49 detik), sementara kinerja terendah terjadi pada kondisi perangkat kelas bawah dan teks melengkung (84,8–87,6%). Waktu pemrosesan meningkat seiring menurunnya akurasi pada kondisi yang lebih sulit, konsisten dengan mesin OCR yang mengeluarkan lebih banyak upaya evaluasi ulang tingkat blok (block-level re-evaluation) pada masukan yang terdegradasi.

Tabel 5
Akurasi OCR dan Waktu Pemrosesan Berdasarkan Kondisi

Kondisi	Akurasi (%)	Waktu rata-rata (detik)
Pencahayaan baik	98,7	1,22
Pencahayaan sedang	96,4	1,35
Pencahayaan rendah	89,1	1,61
Sampul mengkilap	91,3	1,58
Teks melengkung	87,6	1,74
Kamera kelas bawah	84,8	1,92
Rata-rata keseluruhan	93,8	1,49

Gambar 6. Akurasi OCR dan Waktu Pemrosesan pada Enam Kondisi Akuisisi (garis putus-putus: rata-rata keseluruhan)

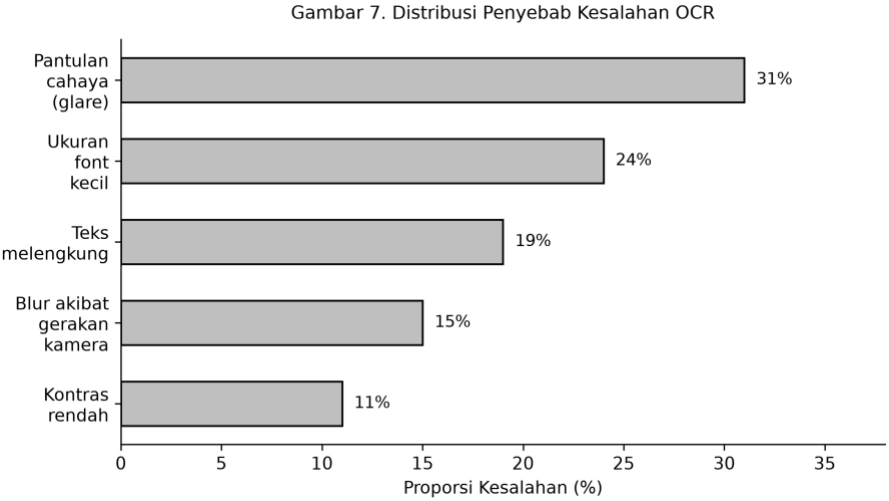


Gambar 6. (a) Akurasi OCR dan (b) Waktu Pemrosesan pada Enam Kondisi Akuisisi (garis putus-putus: rata-rata keseluruhan)

Untuk mengidentifikasi sumber kesalahan dominan dan memahami pola kegagalan pada dataset 100 buku, seluruh sampel yang salah dikenali diperiksa secara manual oleh tim peneliti dan dikategorikan berdasarkan penyebab utama kegagalan (kegagalan deteksi vs. kegagalan pengenalan, beserta penyebab spesifiknya). Hasilnya dirangkum pada Tabel 6 dan divisualisasikan pada Gambar 7. Pantulan cahaya pada sampul (glare) dan ukuran font kecil bersama-sama menyumbang 55% dari kesalahan yang teramati, diikuti oleh teks melengkung (19%), blur akibat gerakan kamera (15%), dan kontras rendah (11%). Distribusi ini bersifat krusial: hal ini mengungkapkan bahwa hambatan utama bukan pada akurasi OCR tingkat karakter (yang tetap >98% bahkan pada masukan terdegradasi), melainkan pada tahap deteksi blok teks, yang mengalami kesulitan terhadap pantulan spekular, distorsi geometris, dan blur akibat gerakan.

Tabel 6
Analisis Kegagalan: Penyebab Kesalahan

Penyebab Kesalahan	Proporsi Kesalahan (%)
Pantulan cahaya pada sampul (glare)	31
Ukuran font kecil	24
Teks melengkung	19
Blur akibat gerakan kamera	15
Kontras rendah	11



Gambar 7. Distribusi Penyebab Kesalahan OCR

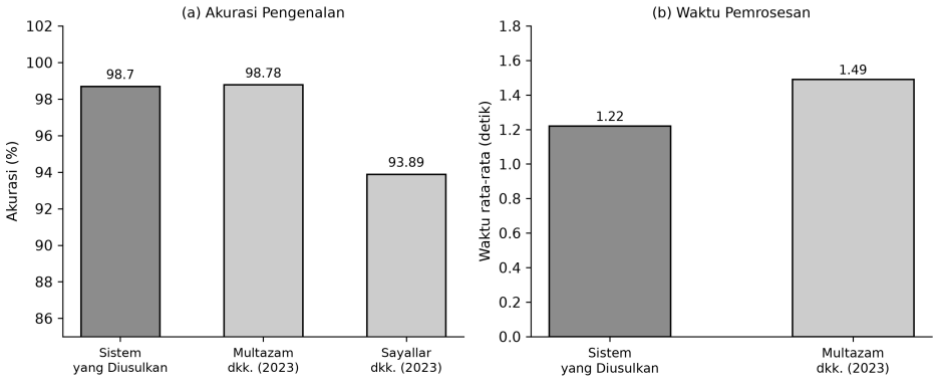
Dua temuan penting dapat digarisbawahi. Pertama, pengenalan karakter tingkat OCR tetap tinggi bahkan pada kondisi yang terdegradasi; sebagian besar kegagalan berasal dari tahap deteksi blok teks, bukan dari kesalahan pembacaan karakter, konsisten dengan pemisahan tahap deteksi/pengenalan pada Google ML Kit. Kedua, pantulan cahaya pada sampul dan distorsi geometris akibat teks melengkung merupakan dua moda kegagalan yang dominan dan sebagian besar tidak bergantung pada jenis perangkat – yang mengindikasikan bahwa fitur panduan akuisisi citra (misalnya peringatan silau dan perintah perataan halaman) merupakan investasi peningkatan ketahanan sistem yang lebih menjanjikan dibandingkan penyetelan lanjutan pada model OCR itu sendiri.

Pencahayaan memengaruhi kontras tepi karakter (character-edge contrast): akurasi menurun dari 98,7% (pencahayaan baik) menjadi 89,1% (pencahayaan rendah), merupakan perubahan terbesar yang teramati akibat kondisi tunggal. Kualitas kamera (autofocus, stabilisasi optis/OIS) menekan efek blur; ketidakhadiran fitur ini pada kondisi kamera kelas bawah berkorelasi dengan akurasi terendah (84,8%) dan waktu pemrosesan terlama (1,92 detik) pada dataset. Resolusi citra mempertukarkan detail piksel per karakter dengan jumlah blok teks yang diproses, turut berkontribusi pada peningkatan waktu pemrosesan pada kondisi yang lebih sulit. Google ML Kit melakukan deteksi (lokalisasi blok) dan pengenalan (dekode karakter) secara berurutan; analisis kegagalan menunjukkan deteksi, bukan pengenalan, sebagai hambatan utama pada kondisi pantulan cahaya dan kelengkungan.

Terkait klaim efisiensi sumber daya, penelitian ini tidak melakukan pengukuran instrumentasi langsung terhadap konsumsi baterai (mAh) maupun puncak penggunaan memori (MB) selama pemrosesan OCR berjalan di perangkat, sehingga angka kuantitatif untuk kedua besaran tersebut belum tersedia dan tidak dilaporkan di sini. Meskipun demikian, efisiensi sumber daya pada sistem ini didukung secara arsitektural oleh tiga faktor. Pertama, seluruh tahap OCR berjalan sepenuhnya on-device melalui Google ML Kit, yang dirancang untuk inferensi ringan pada perangkat kelas konsumen (Tabel 4: RAM 8 GB) tanpa memerlukan model deep learning bertumpukan besar maupun komunikasi jaringan berulang ke server, berbeda dengan pendekatan berbasis inferensi sisi server seperti pada Sayallar et al. (2023). Kedua, tahap klasifikasi kata kunci menggunakan fuzzy matching berbasis Levenshtein yang ringan secara komputasi dan tidak memerlukan proses inferensi model machine learning, sehingga beban CPU tambahan pada tahap ini minimal dibandingkan pendekatan berbasis pembelajaran terawasi. Ketiga, pemrosesan OCR hanya dipicu pada saat pengguna melakukan pengambilan citra (bukan melalui pemindaian video atau kamera yang menyala terus-menerus di latar belakang), sehingga durasi aktif pemrosesan per buku terbatas pada rata-rata 1,49 detik (Tabel 5) dan tidak menimbulkan beban komputasi maupun konsumsi daya yang berkelanjutan. Kuantifikasi langsung konsumsi baterai dan memori melalui instrumen profiling perangkat (misalnya Android Studio Profiler atau Battery Historian) pada berbagai kelas perangkat tetap diperlukan untuk memvalidasi klaim efisiensi sumber daya secara empiris, dan hal ini ditetapkan sebagai salah satu arah penelitian selanjutnya.

Gambar 8 membandingkan subset kondisi pencahayaan baik pada sistem yang diusulkan – kondisi yang paling sebanding dengan evaluasi terkendali yang dilaporkan pada literatur – terhadap dua benchmark terdekat yang dilaporkan sebelumnya. Pada kondisi yang disandingkan, sistem yang diusulkan mencapai akurasi yang sebanding dengan Multazam et al. (2023) (98,7% berbanding 98,78%) dengan waktu pemrosesan rata-rata yang lebih cepat (1,22 detik berbanding 1,49 detik), sekalipun berjalan sepenuhnya di perangkat (on-device). Berbeda dari Multazam et al. (2023) dan Sayallar et al. (2023), yang melaporkan kinerja pada satu kondisi terkendali saja, evaluasi pada penelitian ini turut mengukur penurunan kinerja pada lima kondisi yang lebih sulit dan realistis (Tabel 5), sehingga memberikan gambaran ketahanan sistem yang lebih lengkap dibandingkan benchmark satu-kondisi sebelumnya.

Gambar 8. Perbandingan pada Kondisi Tersanding (Good-Lighting) dengan Benchmark Terdekat



Gambar 8. Perbandingan pada Kondisi Tersanding (Pencahayaan Baik) dengan Benchmark Terdekat: (a) Akurasi Pengenalan dan (b) Waktu Pemrosesan Rata-rata

SIMPULAN

Penelitian ini menghasilkan sebuah sistem OCR mobile untuk ekstraksi metadata buku secara otomatis, yang menggabungkan pipeline Google ML Kit yang berjalan on-device, pencocok kata kunci fuzzy berbasis Levenshtein dengan formulasi yang dinyatakan secara eksplisit, serta penyimpanan cloud berbasis PostgreSQL. Dievaluasi pada 100 sampel buku dengan enam kondisi akuisisi, sistem mencapai akurasi keseluruhan sebesar 93,8% (hingga 98,7% pada pencahayaan baik) dengan waktu pemrosesan rata-rata 1,49 detik. Pada kondisi terkendali yang disandingkan, akurasi dan kecepatannya setara dengan atau lebih baik dibandingkan benchmark terdekat yang dilaporkan (Multazam et al., 2023) (98,7% berbanding 98,78% akurasi; 1,22 detik berbanding 1,49 detik).

Selain aplikasi mobile itu sendiri, kontribusi utama penelitian ini bersifat diagnostik: analisis kegagalan menunjukkan bahwa pengenalan OCR tingkat karakter tetap konsisten tinggi bahkan pada kondisi terdegradasi, sementara sumber kesalahan yang dominan dan berulang adalah kegagalan deteksi blok teks akibat pantulan cahaya dan distorsi geometris – secara spesifik, pantulan cahaya pada sampul (31% dari kesalahan) dan distorsi geometris akibat teks melengkung (19% dari kesalahan), bersama dengan isu terkait berupa ukuran font kecil (24%) dan blur akibat gerakan (15%). Implikasi praktisnya adalah bahwa peningkatan lebih lanjut pada OCR metadata buku mobile kemungkinan besar tidak akan datang dari perbaikan model pengenalan karakter, yang telah mendekati kinerja puncak di seluruh state of the art yang ditinjau (93–99%), melainkan dari penanganan hambatan pada tahap akuisisi dan deteksi yang teridentifikasi

di sini: panduan akuisisi yang peka-silau (glare-aware capture guidance) dan koreksi kelengkungan/perataan halaman.

Dataset 100 buku dengan enam kondisi memperluas cakupan pengujian dibandingkan evaluasi pada satu kondisi tunggal, namun naskah non-Latin dan buku yang rusak parah (misalnya akibat noda air atau sobek) belum diuji. Skalabilitas basis data multi-pengguna, konflik penulisan bersamaan (concurrent write contention), dan penanganan data duplikat pada skala institusi juga belum dievaluasi. Penelitian selanjutnya disarankan untuk memperluas evaluasi pada naskah non-Latin dan buku yang terdegradasi secara fisik, membandingkan dengan OCR deep learning berbasis server pada data yang sama, mengevaluasi beban basis data multi-pengguna, serta mengeksplorasi fitur panduan akuisisi (deteksi silau dan kelengkungan) yang termotivasi langsung dari analisis kegagalan pada penelitian ini, serta melakukan profiling kuantitatif konsumsi baterai dan penggunaan memori perangkat selama pemrosesan OCR on-device untuk memvalidasi klaim efisiensi sumber daya secara empiris.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Politeknik Elektronika Negeri Surabaya (PENS) atas dukungan dan fasilitas yang diberikan selama penelitian ini berlangsung.

DAFTAR PUSTAKA

Ayu Megawaty, D., Alita, D., & Sukma Dewi, P. (2021). Penerapan digital library untuk otomatisasi administrasi perpustakaan. Jurnal Teknologi dan Sistem Informasi (JTSCS), 2(2).

Clausner, C., Antonacopoulos, A., & Pletschacher, S. (2020). Efficient and effective OCR engine training. International Journal on Document Analysis and Recognition, 23(1). https://doi.org/10.1007/s10032-019-00347-8

Damayanti, D. L., Hidayati, D., & Mandasari, O. (2023). Digital library: Upaya mewujudkan perpustakaan sekolah berbasis teknologi. Jurnal Pendidikan dan Konseling (JPDK), 5(1).

Darwish, S. M., & Elzoghaly, K. O. (2020). An enhanced offline printed Arabic OCR model based on bio-inspired fuzzy classifier. IEEE Access, 8. https://doi.org/10.1109/ACCESS.2020.3004286

Faizullah, S., Ayub, M. S., Hussain, S., & Khan, M. A. (2023). A survey of OCR in Arabic language. Applied Sciences, 13(7). https://doi.org/10.3390/app13074584

Haq, M. S., Samani, M., Karwanto, & Hariyati, N. (2022). Android-based digital library application development. International Journal of Interactive Mobile Technologies, 16(11). https://doi.org/10.3991/ijim.v16i11.32055

Hassan, O., et al. (2023). An optimized hardware inference of SABiNN. IEEE Transactions on Instrumentation and Measurement, 72. https://doi.org/10.1109/TIM.2023.3279880

Kayalvizhi, S., Akash Silas, N., Tarunaa, R. K., & Pothirajan, S. (2024). OCR-based ingredient recognition for consumer well-being. Lecture Notes in Networks and Systems. https://doi.org/10.1007/978-981-99-6906-7_41

Khan, S. A., & Shahzad, K. (2024). Key features of digital library management system (DLMS) for developing digital libraries. Journal of Librarianship and Information Science, 56(1). https://doi.org/10.1177/09610006221127033

Kusnantoro, Rohana, T., & Sulistya Kusumaningrum, D. (2022). Implementasi metode Tesseract OCR untuk deteksi plat nomor kendaraan pada sistem parkir. SSJITS, 3(1).

Li, H., et al. (2023). Deep skin diseases diagnostic system with dual-channel image and extracted text. Frontiers in Artificial Intelligence, 6. https://doi.org/10.3389/frai.2023.1213620

Lyu, L., Koutraki, M., Krickl, M., & Fetahu, B. (2021). Neural OCR post-hoc correction of historical corpora. Transactions of the Association for Computational Linguistics, 9. https://doi.org/10.1162/tacl_a_00379

Multazam, A. E., Qashlim, A., & Sarjan, M. (2023). Image processing technology in book metadata extraction system using OCR. Jurnal Sisfotek Global, 13(1). https://doi.org/10.38101/sisfotek.v13i1.865

Nafisah, S. (2023). Building a social space through a digital library. Librarianship in Muslim Societies, 2(1). https://doi.org/10.15408/lims.v2i1.31755

Nguyen, Q. D., Phan, N. M., Kromer, P., & Le, D. A. (2023). An efficient unsupervised approach for OCR error correction of Vietnamese OCR text. IEEE Access, 11. https://doi.org/10.1109/ACCESS.2023.3283340

Oprea, M. (2021). Integration of artificial intelligence in STEM education through IoT projects based on machine learning. Proceedings of the eLearning and Software for Education Conference. https://doi.org/10.12753/2066-026X-21-096

Oucheikh, R., Pettersson, T., & Löfström, T. (2022). Product verification using OCR classification and Mondrian conformal prediction. Expert Systems with Applications, 188. https://doi.org/10.1016/j.eswa.2021.115942

Ridwan, M. M., & Widiawati, W. (2022). Bimbingan teknis otomasi perpustakaan berbasis Inlislite. Maspul Journal of Community Empowerment, 4.

Sayallar, C., Sayar, A., & Babalık, N. (2023). An OCR engine for printed receipt images using deep learning techniques. International Journal of Advanced Computer Science and Applications, 14(2). https://doi.org/10.14569/IJACSA.2023.0140295

Zeng, M. L., Lee, J., & Hayes, A. F. (2009). Metadata decisions for digital libraries: A survey report. Journal of Library Metadata, 9(3–4). https://doi.org/10.1080/19386380903405074