Vol. 11, No. 2, Oktober 2026, Page. 689-705
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Kinerja Arsitektur Ekstraksi Tanggal Kedaluwarsa pada Produk Kemasan Pangan Menggunakan YOLOv11 Nano, Filter Citra Adaptif, dan Google ML Kit

Vanessa Angelika[*1], Riki Afriansyah[2], Putri Armilia Prayesy[3]

[1,2,3]Program Studi Sarjana Terapan Teknologi Rekayasa Perangkat Lunak,
Jurusan Teknik Informatika dan Bisnis, Politeknik Manufaktur Negeri Bangka Belitung, Indonesia
Email: [1]vanessaangelika22@gmail.com, [2]riki.afriansyah@polman-babel.ac.id,
[3]putriprayesy@gmail.com

Abstrak

Tanggal kedaluwarsa pada kemasan pangan sulit dibaca secara otomatis karena ukuran teks kecil, posisi tidak seragam, format bervariasi, serta permukaan kemasan yang silau atau bercetak titik-titik (*dot-matrix)*. Penelitian ini bertujuan menganalisis pengaruh filter citra (*Grayscale*, *Solid Sharp*, dan *Dot-Matrix*) terhadap akurasi pembacaan OCR pada tanggal kedaluwarsa melalui skema filter adaptif berbasis Google ML Kit, dengan menerapkan ketiga filter OpenCV secara bertahap tanpa ground truth pada tahap pemilihan filter. Skema diterapkan pada arsitektur YOLOv11 Nano, Google ML Kit *on-device*, dan *Regular Expression* (Regex), disertai analisis galat pada tahap lokalisasi, OCR, dan normalisasi. Model dilatih pada 869 citra kemasan (631 latih, 159 validasi, 79 uji) dan mencapai mAP@50 0,917 pada data uji. Hasil pada uji makro (50 sampel produk, dengan koreksi filter berdasarkan *ground truth*), keberhasilan lokalisasi mencapai 96%, Character Accuracy 84%, dan *Date Accuracy* 70%. Pada uji mikro (20 sampel), *Date Accuracy* naik dari 50% pada citra asli menjadi 95% dengan skema adaptif, sedangkan filter tunggal mencapai 60% sampai 65%. Pada sampel yang sama, *Date Accuracy* PaddleOCR adalah 75% pada citra asli dan 70% dengan skema adaptif, sehingga manfaat filter adaptif bergantung pada mesin OCR, dan waktu baca PaddleOCR 2,24 sampai 4,30 kali lebih lama daripada ML Kit. Temuan ini bersifat awal dan belum dapat digeneralisasikan karena pengujian mikro hanya mencakup 20 sampel (14 di antaranya *dot-matrix*) yang dipilih berdasarkan ketersediaan produk.

Kata kunci: *computer vision*, filter citra, Google ML Kit, OCR, tanggal kedaluwarsa, YOLOv11

Abstract

Expiration dates on food packaging are difficult to read automatically due to small text size, inconsistent positions, varying formats, and packaging surfaces that may cause glare or use dot-matrix printing. This study aims to analyze the effect of image filters (Grayscale, Solid Sharp, and Dot-matrix) on OCR accuracy for expiration date recognition through an adaptive filtering scheme based on Google ML Kit. The three OpenCV filters are applied sequentially without ground truth during filter selection. The scheme uses YOLOv11 Nano, Google ML Kit on-device, and Regular Expression (Regex), with error analysis at the localization, OCR, and normalization stages. The model was trained on 869 packaging images (631 training, 159 validation, and 79 test images) and achieved an mAP@50 of 0.917 on the test data. In the macro test (50 product samples, with filter correction based on ground truth), localization success reached 96%, with Character Accuracy of 84% and Date Accuracy of 70%. In the micro test (20 samples), Date Accuracy increased from 50% on original images to 95% with adaptive scheme, while single filters achieved 60% to 65%. On the same samples, PaddleOCR achieved 75% on original images and 70% using the adaptive scheme, indicating that the benefit of adaptive filtering depends on the OCR engine, with reading time 2.24–4.30 times longer than ML Kit. These findings are preliminary and cannot be generalized because the micro test covered only 20 samples (14 dot-matrix) selected by product availability.

Keywords: *computer vision, expiration date, Google ML Kit, image filtering, OCR, YOLOv11*

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)

1. PENDAHULUAN

Tanggal kedaluwarsa menjadi acuan utama untuk memastikan suatu produk pangan olahan masih berada dalam batas aman konsumsi dan mutunya tetap terjaga. BPOM menegaskan bahwa produk yang telah melewati tanggal kedaluwarsa tidak lagi layak dikonsumsi maupun diedarkan [1]. Data dari WHO menyebutkan bahwa pangan yang tidak aman menyebabkan sekitar 866 juta kasus penyakit dan 1,52

juta kematian pada tahun 2021 [2]. Meski demikian, di rumah tangga maupun usaha kecil, tanggal tersebut umumnya masih dicatat secara manual dengan memeriksa label pada setiap kemasan satu per satu. Cara pencatatan manual tersebut tidak efisien, rentan terhadap kesalahan, dan berpotensi menyebabkan produk terabaikan hingga melewati batas simpannya yang dapat berkontribusi terhadap tingginya angka limbah pangan (*food waste*). Kondisi ini mempersulit penerapan prinsip *First Expired First Out* (FEFO), yaitu menggunakan terlebih dahulu produk dengan masa simpan terpendek apabila dijalankan tanpa pencatatan tanggal secara teratur [3].

Membaca tanggal kedaluwarsa secara otomatis tidaklah sederhana. Tanggal kedaluwarsa pada kemasan pangan umumnya berukuran cetak kecil, diletakkan pada posisi yang tidak teratur, dan berformat beragam, mulai dari dd/mm/yyyy, mm/yyyy, hingga kode EXP yang bercampur dengan kode produksi. Seker dan Ahn dalam penelitiannya menyampaikan bahwa keragaman format, ukuran karakter, dan kualitas cetak merupakan penghambat utama dalam pembacaan tanggal kedaluwarsa pada kemasan [4]. Tantangan visual lainnya berasal dari material kemasan berbahan plastik fleksibel atau kaleng logam yang memantulkan cahaya (*glare*) sehingga menurunkan kontras teks. Selain itu, metode pencetakan industri kerap menggunakan tinta *dot-matrix* berupa titik-titik terputus-putus sehingga mesin OCR kerap gagal membaca karakter tanggal.

Berbagai penelitian terdahulu telah mengembangkan sistem deteksi dan ekstraksi tanggal kedaluwarsa, namun masih memiliki sejumlah keterbatasan. Penelitian oleh Agatha dan Lina menggunakan segmentasi histogram dan *Convolutional Neural Network* (CNN) pada kemasan kopi instan, dengan tingkat pemotongan karakter 94,55%, tetapi akurasi pengenalan hanya 77,5% akibat karakter yang menempel, gambar miring, dan bentuk karakter yang serupa [5]. Richo menggabungkan YOLOv11m dan PaddleOCR dengan mAP50 sebesar 98,44%, tetapi menggunakan 350 citra dari satu produk dan hanya menguji 5 sampel, sehingga cakupan pengujiannya masih terbatas dan belum teruji pada kemasan produk yang berkerut, silau, atau berformat berbeda [6]. Madake dkk. menggabungkan MobileNet dan PaddleOCR dengan Gaussian blur serta *adaptive threshold*, dan mencapai akurasi 97,7% pada lebih dari 5.000 citra. Namun, posisi dan orientasi teks pada data relatif seragam [7]. Dospinescu dkk. menggunakan *Google Cloud Vision* dan berhasil membaca tanggal pada 25 dari 30 produk (83,3%), tetapi kesalahan masih muncul pada permukaan mengilap atau melengkung serta cap tanggal yang kecil dan berkontras rendah, terutama karena tidak dilengkapi dengan proses perbaikan kualitas citra. Sistem tersebut juga bergantung pada koneksi internet karena menggunakan *cloud* [8]. Penelitian oleh Emel dkk. menunjukkan bahwa kombinasi deteksi objek, OCR, dan Regex dapat mendukung ekstraksi tanggal pada faktur [9] berupa dokumen datar, berbeda dengan kemasan. Sedangkan perbaikan citra sebelum OCR juga telah diterapkan pada kartu identitas dan tulisan tangan digital [10], [11]. Dari penelitian-penelitian yang telah dikaji, masih terbatas penelitian yang membandingkan beberapa kondisi citra secara langsung untuk ekstraksi tanggal pada kemasan pangan, khususnya pada permukaan silau dan cetakan *dot-matrix*.

Bila [5], [6], [7], [8] dibandingkan secara kritis, penelitian terdahulu menunjukkan celah yang berbeda. Richo [6] dan Madake dkk. [7] sama-sama menggabungkan detektor dengan satu mesin OCR, yaitu PaddleOCR, tetapi belum menguji bagaimana kinerja OCR berubah ketika kondisi cetakan dan karakteristik kemasan berbeda. Agatha dan Lina [5] menggunakan satu metode pengenalan setelah segmentasi, sedangkan Dospinescu dkk.[8] menggunakan satu jalur pembacaan tanpa mekanisme pemilihan kondisi citra ketika hasil OCR tidak berhasil. Selain itu, [5], [6], [7], [8] belum memisahkan kegagalan sistem berdasarkan tahapan proses, sehingga belum dapat diketahui secara spesifik apakah kesalahan berasal dari lokalisasi, OCR, atau normalisasi. Celah tersebut menunjukkan perlunya pendekatan yang dapat menyesuaikan kondisi citra sebelum OCR sekaligus mengidentifikasi tahapan yang menjadi sumber kegagalan dalam proses ekstraksi tanggal.

Guna mengatasi permasalahan tersebut, penelitian ini merancang arsitektur sistem *computer vision* terpadu untuk mengekstraksi tanggal kedaluwarsa pada berbagai kondisi kemasan. Model YOLOv11 Nano digunakan untuk melokalisasi area teks tanggal secara akurat dengan beban komputasi rendah [12], [13]. Citra hasil *crop* diolah melalui tiga opsi filter pemrosesan adaptif berbasis OpenCV sesuai dengan kondisi fisik kemasan maupun format cetak tanggal. Tahap dasar memanfaatkan filter *Grayscale* yang memadukan *upscaling* untuk meningkatkan kualitas citra serta melakukan konversi ke skala keabuan. Untuk kemasan plastik atau kaleng yang silau atau berbayang, sistem menerapkan filter *Solid Sharp* yang mengombinasikan *Contrast Limited Adaptive Histogram Equalization* (CLAHE), *Gaussian Blur*, dan *Otsu Thresholding*. Sementara itu, untuk menyatukan kembali titik-titik karakter yang terpisah

pada cetakan *dot-matrix*, diterapkan filter *Dot-matrix* melalui *adaptive thresholding* dan operasi morfologi (*erode-dilate*). Teks pada citra tersebut kemudian dibaca oleh Google ML Kit yang berjalan *on-device* [14]. Teks mentah luaran OCR selanjutnya disaring dan dinormalisasi ke dalam format tanggal baku menggunakan algoritma *Regular Expression* (Regex)[15] .

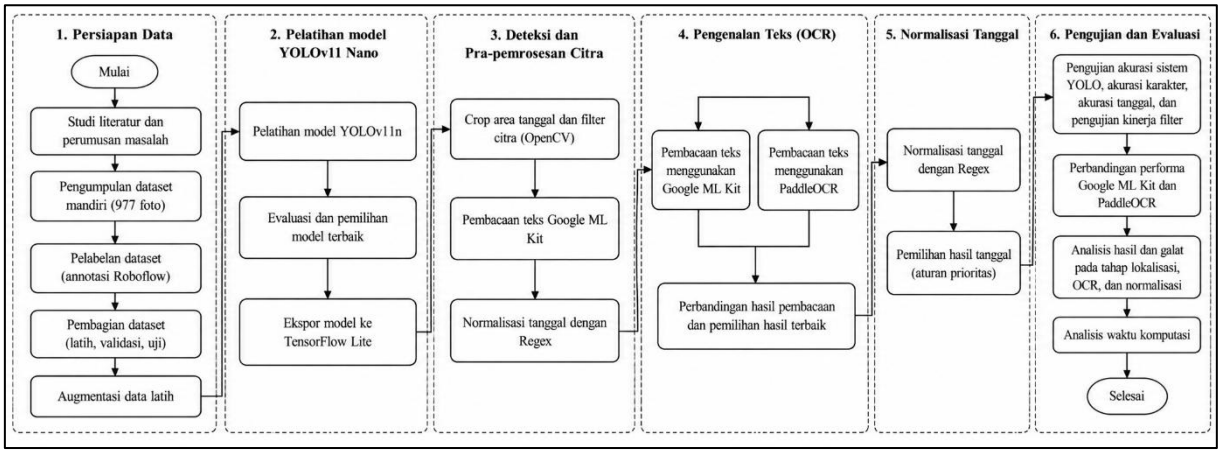
Penelitian ini bertujuan untuk mengukur kinerja arsitektur tersebut sekaligus menguji apakah pemilihan filter yang tepat dapat meningkatkan akurasi baca ML Kit. Untuk itu, evaluasi dilakukan dalam dua tahap: uji makro yang menggunakan 50 sampel produk untuk menilai lokalisasi YOLO dan kinerja sistem secara keseluruhan dengan koreksi filter berdasarkan *ground truth*, serta uji mikro yang menggunakan 20 sampel untuk membandingkan citra asli dengan ketiga filter dan skema pemilihan filter secara bertahap, dengan ML Kit dan PaddleOCR sebagai OCR pembanding pada sampel yang sama, baik dari sisi akurasi maupun waktu pembacaan. Dengan pengukuran *Character Accuracy* dan *Date Accuracy*, penelitian ini mengukur efektivitas penambahan filter citra adaptif dalam menurunkan tingkat kegagalan OCR pada kondisi visual yang sulit, seperti pantulan cahaya dan karakter yang terputus (*dot-matrix*), disertai analisis galat yang memisahkan kegagalan pada tahap lokalisasi, OCR, dan normalisasi tanggal.

Kebaruan penelitian ini tidak terletak pada penggunaan YOLOv11 Nano atau Google ML Kit, melainkan pada integrasi pengolahan citra adaptif dan analisis kinerja secara bertahap. Kontribusi penelitian mencakup tiga aspek, yaitu (1) skema filter adaptif yang memilih kondisi citra secara bertahap dengan kriteria penghentian berupa keluaran tanggal yang memenuhi format Regex, sehingga tidak memerlukan *ground truth* saat penggunaan; (2) analisis galat yang memisahkan kegagalan pada tahap lokalisasi, OCR, dan normalisasi Regex; serta (3) evaluasi menggunakan baseline tanpa filter, filter tunggal, dan OCR alternatif untuk menilai kontribusi filter adaptif secara lebih objektif.

2. METODE PENELITIAN

2.1. Rancangan Penelitian dan Data

Penelitian ini bersifat eksperimental, dengan kondisi citra (asli, *Grayscale*, *Solid Sharp*, *Dot-matrix*, dan filter adaptif) sebagai variabel yang diuji, serta *Character Accuracy* dan *Date Accuracy* sebagai ukuran hasil. Alur penelitian diawali studi literatur dan persiapan data berupa pengumpulan dataset, anotasi, pembagian data latih, validasi, dan uji, serta augmentasi data latih. Tahap kedua meliputi pelatihan YOLOv11n, pemilihan model terbaik berdasarkan hasil evaluasi, dan ekspor ke TensorFlow Lite. Tahap ketiga, mengintegrasikan model dengan filter OpenCV, Google ML Kit, dan Regex dalam aplikasi Android berbasis Flutter. Tahap terakhir menguji sistem pada 50 sampel (uji makro) dan 20 sampel (uji mikro), kemudian menganalisis akurasi dan galat.



Gambar 1. Alur Penelitian

Foto dataset diambil sendiri menggunakan Samsung Galaxy A04 dan Asus Zenfone 11 Ultra di beberapa toko ritel di Pangkalpinang, yaitu Hypermart, Alfamart, dan Indomaret. Objeknya berupa kemasan plastik fleksibel, karton, dan kaleng yang mencantumkan tanggal numerik. Setiap produk difoto tegak lurus maupun dengan kemiringan sekitar 5° sampai 15° ke kiri dan ke kanan, pada jarak 10

sampai 15 cm, dalam orientasi vertikal dan pencahayaan ruangan. Hasilnya didapatkan 977 foto, terdiri atas 414 kemasan plastik, 308 kemasan karton, dan 255 kemasan kaleng. Foto yang buram, terlalu gelap, tanggalnya tertutup pantulan cahaya, atau miring secara ekstrem kemudian disingkirkan, sehingga tersisa 869 citra.

Anotasi dilakukan secara manual di Roboflow dengan satu kelas, yaitu *expiry_area*. Bila tanggal kedaluwarsa dapat dipisahkan dari kode produksi atau nomor *batch*, hanya area tersebut yang diberi kotak anotasi. Sebaliknya, jika menyatu dengan tanggal produksi, seluruh blok dianotasi sebagai satu objek. Margin kotak dibuat sekitar 3 sampai 5 piksel dari karakter, dan aturan ini memengaruhi galat. Dataset dibagi menjadi 631 citra latih, 159 citra validasi, dan 79 citra uji (Tabel 1). Semua data diseragamkan menjadi 768 × 768 piksel, lalu data latih diaugmentasi dengan tiga keluaran per citra (rotasi ±15°, kecerahan ±15, *blur* maksimum 1,1 piksel, dan noise 1,101%, sementara data validasi dan uji dibiarkan tanpa augmentasi agar evaluasi dilakukan menggunakan citra asli.

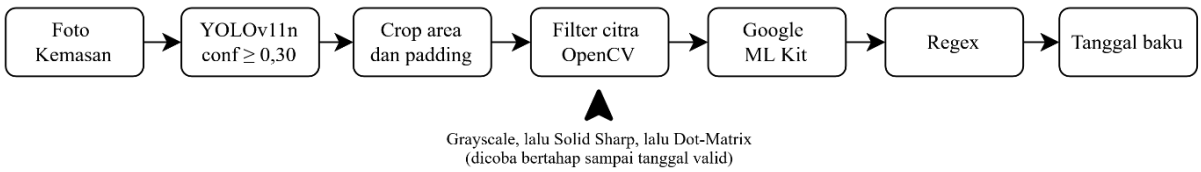
Tabel 1. Jumlah Citra pada Setiap Subset Dataset

Subset	Sebelum Augmentasi	Setelah Augmentasi
Latih (*Train*)	631	1.893
Validasi (*Valid*)	159	159
Uji (*Test*)	79	79
Total	**869**	**2.131**

2.2. Konfigurasi YOLOv11 Nano

Model yang digunakan adalah YOLOv11 Nano dengan bobot awal *yolo11n.pt*. Varian nano dipilih karena hanya memiliki sekitar 2,6 juta parameter dan 6,5 GFLOPs pada resolusi 640 [13], sehingga komputasinya cukup ringan untuk inferensi pada ponsel [12]. Arsitektur YOLOv11 terdiri atas *backbone*, *neck*, dan *head* [16]. Pelatihan berjalan 100 epoch dengan batch size 16, ukuran citra 768 × 768, dan *optimizer* otomatis dari Ultralytics, di Google Colab dengan GPU Tesla T4. Selain augmentasi dari Roboflow, Ultralytics menerapkan augmentasi bawaan selama pelatihan, yaitu mosaik, perubahan HSV, translasi dan skala, pembalikan horizontal, serta random *erasing*. Bobot terbaik (*best.pt*) didapatkan pada epoch ke-94 dan diekspor ke TensorFlow Lite (float32) tanpa kuantisasi.

Alur sistem dari foto kemasan hingga tanggal baku ditunjukkan pada Gambar 2. Karena hanya ada satu kelas, tugas model adalah murni melokalisasi area tanggal. Di aplikasi, citra diubah dengan *letterbox* menjadi 768 × 768, lalu sistem mengambil satu kotak dengan *confidence* tertinggi asalkan nilainya minimal 0,30. Ambang *confidence* ini sengaja dibuat lebih rendah daripada titik F1 terbaik karena kotak yang terlewat tidak dapat dipulihkan pada tahap berikutnya. Kotak tersebut kemudian dipetakan kembali ke citra asli dan dipotong dengan *padding* tambahan 5% pada lebar dan 20% pada tinggi kotak agar karakter di tepi tidak terpotong.



Gambar 2. Arsitektur Sistem dari Foto Kemasan sampai Tanggal Baku

2.3. Skema Filter Adaptif

Hasil *crop* dibaca dalam 4 kondisi. Citra asli (ORI) adalah citra tanpa pemrosesan dan menjadi pembanding, sedangkan tiga filter lainnya berangkat dari citra dasar berupa *crop* yang diperbesar tiga kali, lalu diubah ke skala keabuan. Aplikasi menampilkan keempat pilihan tersebut sebagai tombol filter, lengkap dengan citra hasil olahan, teks mentah ML Kit, dan tanggal hasil Regex (Gambar 3). Filter *Grayscale* menghitung intensitas dari kanal warna dengan bobot luminans standar,

$$Y = 0{,}299\,R + 0{,}587\,G + 0{,}114\,B \quad (1)$$

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

dengan R,G, dan B sebagai nilai kanal merah, hijau, dan biru. Filter ini sekaligus menjadi keluaran awal aplikasi.

Filter *Solid Sharp* ditujukan untuk kemasan yang silau atau berkontras rendah. Tahap pertamanya adalah CLAHE dengan *clip limit* 2,0 dan kisi 8 × 8 petak [17], artinya citra dibagi menjadi 64 petak yang masing-masing diproses sendiri. Histogram h(k) setiap petak dipotong pada batas β = 2,0 A/256, dengan A luas petak, dan kelebihannya dibagi rata ke seluruh 256 tingkat keabuan,

$$h'(k) = \min\left(h(k), \beta + \frac{1}{256}\sum_{j=0}^{255} \max\left(h(j) - \beta, 0\right)\right) \quad (2)$$

lalu fungsi pemetaan intensitas dibentuk dari distribusi kumulatif histogram hasil pemotongan $h'(k)$,

$$s(k) = round\left(\frac{255}{A}\sum_{j=0}^{k} h'(j)\right) \quad (3)$$

sehingga kontras lokal naik tanpa memperbesar derau seperti pada penyetaraan histogram global. Citra kemudian dihaluskan dengan *Gaussian blur* berkernal 5 × 5, dengan nilai σ yang dihitung otomatis oleh OpenCV (σ = 1,1),

$$G(x, y) = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right) \quad (4)$$

dan dibinerkan dengan ambang Otsu [18], yaitu nilai t yang memaksimalkan varians antarkelas,

$$t^* = \arg\max_t \ \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \quad (5)$$

dengan ω dan μ sebagai proporsi dan rerata intensitas piksel yang berada sampai ambang t (indeks 0) dan diatasnya (indeks 1).

Filter *Dot-matrix* ditujukan untuk cetakan titik. Citra lebih dulu diperhalus dengan *median blur* 5 × 5, lalu dibinerkan dengan *adaptive threshold* Gaussian berblok 45 × 45 dan konstanta C = 25 [19]. Piksel bernilai 255 bila lebih terang daripada ambang lokal T(x,y) dan bernilai 0 bila tidak,

$$T(x, y) = \sum_{i,j} w(i, j)I(x + i, y + j) - C \quad (6)$$

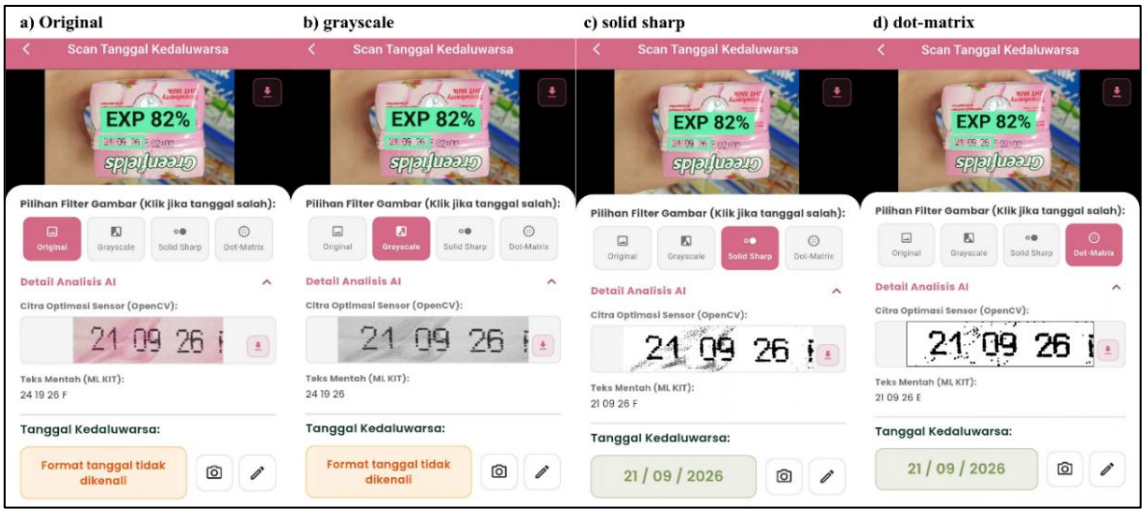
dengan w(i,j) adalah bobot Gaussian yang berjumlah 1 pada jendela 45 × 45. Titik-titik dihubungkan dengan erosi berkernel persegi 6 × 6, lalu dilasi 3 × 3 [20],

$$(A \ominus B)(x, y) = \min_{(i,j)\in B} A(x + i, y + j) \quad (7)$$

$$(A \oplus B)(x, y) = \max_{(i,j)\in B} A(x + i, y + j) \quad (8)$$

dengan B sebagai elemen struktur. Karena teks berwarna hitam di atas latar putih, operasi minimum pada erosi justru memperluas area gelap sehingga titik-titik yang berdekatan menyatu menjadi goresan, dan dilasi 3 × 3 sesudahnya sedikit mengurangi ketebalan garis tersebut.

Vol. 11, No. 2, Oktober 2026, Page. 689-705
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI



Gambar 3. Tampilan Aplikasi pada Satu Sampel untuk Empat Pilihan Filter

Aplikasi menggunakan filter *Grayscale* sebagai tahap awal dan menyediakan pilihan filter lain apabila tanggal belum berhasil dikenali dengan benar. Dalam evaluasi, mekanisme tersebut diterapkan secara bertahap dengan urutan *Grayscale*, *Solid Sharp*, kemudian *Dot-matrix*. Proses dihentikan ketika salah satu filter menghasilkan tanggal yang memenuhi pola regex. Pendekatan ini sendiri tidak memerlukan *ground truth* dalam proses pemilihan filter dan selanjutnya disebut sebagai filter adaptif.

2.4. Ekstraksi Tanggal dengan Regex

Teks mentah dari Google ML Kit (skrip Latin) [21] diproses oleh kelas *DateExtractor* melalui 4 langkah. Langkah pertama: mengoreksi karakter yang bentuknya mirip angka. O, Q, D, dan U menjadi 0; I, L, | dan] menjadi 1; Z menjadi 2; A menjadi 4; S menjadi 5; G menjadi 6; ? menjadi 7; dan B menjadi 8. Pemetaan ini menyasar pertukaran huruf dan angka yang juga dilaporkan dalam penelitian lain, misalnya 0 dengan O atau Q, 1 dengan I, dan 4 dengan A [5]. Langkah kedua memisahkan huruf dan angka yang saling menempel. Pada langkah ketiga, sistem mencari kata kunci seperti EXP, ED, BB, dan *BEST BEFORE*, lalu membuang teks sebelum kata kunci terakhir yang ditemukan, termasuk tanggal produksi yang menyertainya. Langkah keempat mencocokkan delapan pola tanggal, yakni DD MM YYYY atau YY, YYYY MM DD, MM YYYY, DDMMYYYY, DDMMYY, MMYY, DDMM YYYY, dan YYYYMMDD; kandidat hanya diterima apabila kombinasi hari dan bulan valid serta tahunnya berada pada rentang 2026 sampai 2040. Apabila terdapat beberapa hasil ekstraksi tanggal, tanggal lengkap diprioritaskan daripada format bulan-tahun; jika terdapat lebih dari satu tanggal lengkap, dipilih tanggal yang paling akhir, sedangkan format bulan-tahun dinormalisasi menjadi tanggal dengan hari ke-01. Jika pencarian pertama gagal menemukan pola tanggal yang sesuai, angka yang terpisah oleh spasi digabungkan dan proses pencarian diulangi. Langkah ini digunakan untuk membantu Regex dalam mengolah hasil OCR [15]. Hasil akhir ditampilkan dalam format DD/MM/YYYY.

2.5. Skenario Evaluasi

Kinerja model deteksi dievaluasi menggunakan *precision*, *recall*, mAP@50, dan mAP@50-95 pada data validasi dan data uji [22]. Uji makro dilakukan menggunakan 50 sampel produk yang bervariasi dari segi jenis kemasan, format tanggal, ukuran teks, posisi cetak, dan kondisi pencahayaan. Sampel uji makro berasal dari produk yang berbeda dari produk pada 869 citra pelatihan. Pada pengujian ini, hasil *crop* setiap sampel diperbesar tiga kali, lalu dibaca melalui aplikasi dengan tiga filter secara berurutan, yaitu *Grayscale*, *Solid Sharp*, dan *Dot-matrix*, mengikuti urutan skema adaptif: pembacaan dihentikan pada filter pertama yang menghasilkan tanggal berformat valid. Setelah pemindaian selesai, aplikasi langsung menampilkan hasil Grayscale sebagai filter pertama, sedangkan filter berikutnya dipilih secara manual melalui tombol filter. Jika tanggal valid tersebut tidak sesuai dengan *ground truth*, filter lain tetap diperiksa. Jika terdapat filter yang menghasilkan tanggal valid sekaligus sesuai, hasil tersebutlah yang dipakai, dan bila tidak ada, maka akan dipakai hasil valid yang pertama. Oleh karena itu, hasil uji makro menggambarkan kinerja ketika pengguna dapat mengoreksi pilihan filter dengan memeriksa

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

tanggal sebenarnya pada kemasan, dan cenderung lebih tinggi daripada skema adaptif murni. Setiap sampel dievaluasi berdasarkan tiga parameter, yaitu keberhasilan lokalisasi, *Character Accuracy*, dan *Date Accuracy*. Keberhasilan lokalisasi bernilai 1 apabila kotak deteksi mencakup area tanggal dan hasil *crop* dapat diteruskan ke OCR. Sebaliknya, nilai 0 diberikan apabila kotak tidak terdeteksi, berada di tempat yang salah, atau menghasilkan potongan yang tidak terbaca. *Character Accuracy* sampel ke-i dihitung dari string tanggal hasil normalisasi Regex dalam delapan posisi (DDMMYYYY) terhadap *ground truth* yang dicatat secara manual,

$$CA_i = \frac{C_i}{N_i} \times 100\% \tag{9}$$

dengan C_i sebagai jumlah karakter yang benar dan N_i sebagai jumlah karakter *ground truth* [23]. Nilai akhirnya adalah rerata CA_i pada semua sampel sehingga setiap produk berbobot sama. *Date Accuracy* menilai tanggal satu kesatuan, dengan K sebagai jumlah sampel yang hari, bulan, dan tahunnya sama persis dengan *ground truth* dan n sebagai jumlah seluruh sampel,

$$DA = \frac{K}{n} \times 100\% \tag{10}$$

Uji mikro menggunakan 20 sampel yang mencakup kemasan plastik, karton, dan kaleng dengan bentuk cetakan yang beragam, yaitu *dot-matrix* tipis maupun tebal, serta tinta solid yang tebal. Setiap *crop* dibaca dalam 4 kondisi (ORI, *Grayscale*, *Solid Sharp*, dan *Dot-matrix*) serta dalam skema filter adaptif, dengan semua hasil dicatat dari tangkapan layar aplikasi. *Character Accuracy* pada uji ini dihitung dari teks mentah OCR sebelum Regex agar pengaruh filter terhadap ML Kit terlihat langsung. Karakter pengganggu di luar pola tanggal (misalnya huruf acak atau simbol dari kode *batch*/lot) diabaikan terlebih dahulu sebelum digit yang tersisa dicocokkan secara berurutan terhadap *ground truth*. *Date Accuracy* dihitung dari tanggal yang ditampilkan oleh aplikasi setelah Regex, sehingga nilai *Character Accuracy* pada kedua uji ini tidak dapat dibandingkan secara langsung. Untuk sampel yang *ground truth*-nya hanya berupa bulan dan tahun, tanggal dianggap benar bila bulan dan tahunnya sudah sesuai. Setiap proporsi dilengkapi dengan interval kepercayaan Wilson 95%, yang dipilih karena cakupannya lebih andal daripada interval Wald pada jumlah sampel kecil [24], sedangkan perbedaan antara kondisi ORI dan filter adaptif dianalisis menggunakan uji McNemar eksak untuk data berpasangan [25]. Selisih *Character Accuracy* per sampel antara kedua kondisi tersebut diuji dengan uji peringkat bertanda Wilcoxon [26]. Prosedur pemilihan filter pada uji mikro sama dengan pengujian makro, yaitu menggunakan urutan *Grayscale*, *Solid Sharp*, dan *Dot-Matrix*. Pembacaan dihentikan ketika diperoleh tanggal valid pertama. Apabila tanggal valid tersebut tidak sesuai dengan *ground truth*, pemeriksaan dilanjutkan dengan filter lainnya. Pada sampel 20 pengujian, baik pada ML Kit maupun PaddleOCR, tidak ada filter yang menghasilkan tanggal valid yang salah sebelum tanggal yang benar sehingga pemeriksaan ulang tidak mengubah hasil, dan hasil skema adaptif identik dengan aturan henti pada tanggal valid pertama. Pada ML Kit, sebanyak 12 sampel berhenti di *Grayscale*, 4 di *Solid Sharp*, 3 di *Dot-Matrix*, sedangkan 16 sampel gagal pada memperoleh tanggal valid pada seluruh filter.

2.6. Baseline Pembanding

Hasil skema adaptif dibandingkan dengan tiga baseline pada 20 sampel uji mikro yang sama: *pipeline* tanpa filter (ORI), tiga filter tunggal, dan OCR alternatif PaddleOCR [27] yang menggantikan ML Kit. PaddleOCR dipilih karena dipakai pada [6], [7] dan pernah dibandingkan dengan EasyOCR pada penelitian penulis [28]. *Crop*, urutan filter, kriteria berhenti, Regex, *ground truth*, dan rumus akurasi dipertahankan sama sehingga satu-satunya perbedaan adalah mesin OCR. PaddleOCR dijalankan pada aplikasi yang sama dengan hanya mengganti mesin OCR pada layar hasil deteksi, sehingga crop, filter OpenCV, dan *DateExtractor* identik dengan versi ML Kit. Paket yang dipakai adalah *flutter_paddle_ocr* versi 0.0.2 dengan model PP-OCRv2 Lite (PaddleLite), yaitu model deteksi *det_db.nb*, model pengenalan *rec_crnn.nb*, model klasifikasi *cls.nb*, dan kamus *ppocr_keys_v1.txt*, dengan klasifikasi arah teks dinonaktifkan. Aplikasi pada konfigurasi ini sesekali berhenti sendiri, sehingga pembacaan diulang sampai berhasil. Waktu komputasi dicatat dengan *stopwatch* yang dipasang pada kedua aplikasi hanya di sekitar pemanggilan OCR, yaitu setelah filter OpenCV dan sebelum Regex, untuk 20 sampel dalam keempat kondisi citra. Dengan demikian, waktu inferensi

YOLO, pemrosesan filter OpenCV, dan pemrosesan Regex tidak termasuk dalam pengukuran. Selisih waktu berpasangan antar OCR diuji dengan uji peringkat bertanda Wilcoxon [26]. Waktu skema adaptif tidak diukur karena pada aplikasi pergantian filter dilakukan manual oleh pengguna.

3. HASIL DAN PEMBAHASAN

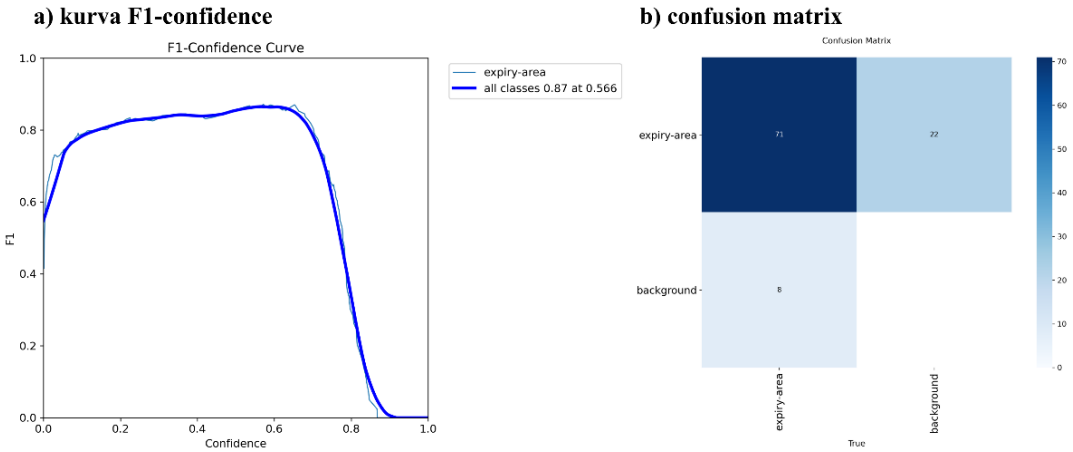
3.1. Kinerja Lokalisasi YOLOv11 Nano

Tabel 2 memuat hasil model terbaik pada data validasi dan data uji. *Checkpoint* terbaik (*best*.pt, *epoch* ke-94) memiliki mAP@50 validasi sebesar 0,958, dan nilai inilah yang dipakai pada Tabel 2 serta pembahasan berikutnya. Nilai pada data validasi lebih tinggi daripada data uji karena data validasi digunakan untuk memilih *checkpoint* terbaik. Oleh karena itu, data uji digunakan sebagai acuan untuk melihat performa model pada data yang tidak digunakan dalam proses pemilihan model. Dibandingkan dengan data validasi, pada data uji *precision* menurun sebesar 0,056 (dari 0,928 menjadi 0,872), mAP@50 sebesar 0,041 (dari 0,958 menjadi 0,917), dan mAP@50-95 sebesar 0,025 (dari 0,600 jadi 0,575), sedangkan *recall* menurun sebesar 0,026 (dari 0,887 menjadi 0,861). Nilai mAP@50-95 sebesar 0,575 yang lebih rendah dibandingkan dengan mAP@50 sebesar 0,917 menunjukkan bahwa model mampu mendeteksi area tanggal, tetapi tingkat ketepatan posisi kotaknya masih lebih rendah pada standar IoU yang lebih ketat. Pada sistem ini, perbedaan tersebut tidak terlalu memengaruhi proses OCR karena hasil deteksi masih diberi *padding* sebelum dilakukan *crop*, sehingga karakter tanggal tetap dapat masuk ke area pemotongan citra.

Tabel 2. Kinerja YOLOv11 Nano pada Data Validasi dan Data Uji

Metrik	Validasi (159 citra)	Uji (79 citra)
Precision	0,928	0,872
Recall	0,887	0,861
mAP@50	0,958	0,917
mAP@50-95	0,600	0,575

Pada *confusion matrix* data uji, 71 dari 79 objek terdeteksi dengan benar (TP), 8 objek tidak terdeteksi (FN), dan 22 objek *false positive* (FP), sehingga *recall* berdasarkan *confusion matrix* sebesar 89,87% seperti ditunjukkan pada Gambar 4b. Nilai tersebut berbeda dari *recall* 0,861 pada Tabel 2 karena keduanya diperoleh dari konfigurasi evaluasi yang berbeda. Kurva F1 pada Gambar 4a mencapai nilai tertinggi sebesar 0,87 pada *confidence* 0,566 dan berada pada kisaran 0,82 – 0,87 untuk *confidence* 0,2 – 0,65. Kondisi ini menunjukkan bahwa perubahan *confidence* dalam rentang tersebut hanya menghasilkan perubahan F1 yang relatif kecil. Aplikasi menggunakan ambang *confidence* 0,30 yang masih berada dalam rentang tersebut. Penggunaan ambang yang lebih rendah dapat menghasilkan lebih banyak prediksi keliru, tetapi pada implementasi aplikasi hanya satu kotak dengan *confidence* tertinggi yang diteruskan ke tahap berikutnya.



Gambar 4. Kurva F1-Confidence dan Confusion Matrix pada Data Uji

Pada uji 50 sampel, kotak deteksi yang mencakup area tanggal diperoleh pada 48 sampel atau sebesar 96% (CI 95%: 86,5–98,9%). Nilai tersebut lebih tinggi daripada *recall* sebesar 0,861 pada data uji. Perbedaan ini berkaitan dengan perbedaan kriteria dan konfigurasi pengujian. Pada uji 50 sampel, setiap gambar hanya memiliki satu target dan hanya kotak dengan *confidence* tertinggi yang dinilai. Selain itu, ambang *confidence* yang digunakan adalah 0,30, sedangkan nilai tersebut lebih rendah daripada titik dengan F1 tertinggi. Keberhasilan lokalisasi juga tidak mensyaratkan IoU minimal 0,5, tetapi cukup apabila kotak deteksi mencakup area tanggal sehingga hasil *crop* dapat diteruskan ke tahap OCR. Dua kegagalan yang tersisa, yaitu pada sampel 15 dan 25, dibahas pada Subbab 3.4.

3.2. Kinerja Ekstraksi Tanggal pada 50 Sampel

Tabel 3 merangkum hasil pengujian makro pada 50 sampel. Rata-rata *Character Accuracy* mencapai 84%, sedangkan *Date Accuracy* mencapai 70%. Perbedaan ini terjadi karena *Date Accuracy* hanya memberikan nilai benar apabila seluruh digit tanggal sesuai dengan *ground truth*. Dari 15 sampel dengan tanggal yang tidak sesuai, tujuh sampel hanya memiliki satu kesalahan pada delapan digit, yaitu sampel 3, 5, 7, 8, 9, 29, dan 45. Ketujuh sampel tersebut memiliki *Character Accuracy* sebesar 87,5%, sehingga kesalahan yang terjadi sebenarnya relatif kecil meskipun tetap menyebabkan *Date Accuracy* bernilai 0. Jika ketujuh sampel tersebut berhasil dibaca dengan tepat, *Date Accuracy* akan meningkat menjadi 84%. Delapan sampel lainnya memiliki tingkat kesalahan yang lebih beragam, terdiri atas satu sampel dengan tiga digit salah (*Character Accuracy* 62,5%), satu sampel dengan enam digit salah (25%), serta enam sampel yang tidak menghasilkan pembacaan atau mengalami kegagalan lokalisasi sehingga memperoleh nilai 0.

Tabel 3. Hasil Uji Makro pada 50 Sampel dengan Koreksi Filter Berdasarkan *Ground Truth*

Ukuran	Hasil	Keterangan
Keberhasilan lokalisasi	96%	48 dari 50; CI 95%: 86,5-98,9
Character Accuracy rata-rata	84%	4.200 / 50
Date Accuracy	70%	35 dari 50; CI 95%: 56,2-80,9
Date Accuracy pada sampel terlokalisasi	72,92%	35 dari 48
Ketepatan tanggal yang tampil	77,78%	35 dari 45 keluaran valid

Selain kesesuaian karakter dan tanggal, jumlah tanggal berformat valid yang dihasilkan oleh sistem juga perlu diperhatikan. Dari 45 sampel yang menghasilkan tanggal dengan format yang valid, 10 di antaranya masih menunjukkan tanggal yang salah. Artinya, sekitar 22,22% dari tanggal yang berhasil ditampilkan dalam format yang valid masih tidak sesuai dengan *ground truth*. Oleh karena itu, verifikasi pengguna tetap diperlukan. Untuk mendukung proses tersebut, aplikasi menyediakan teks mentah hasil OCR, pratinjau hasil *crop*, serta fitur penyuntingan tanggal secara manual.

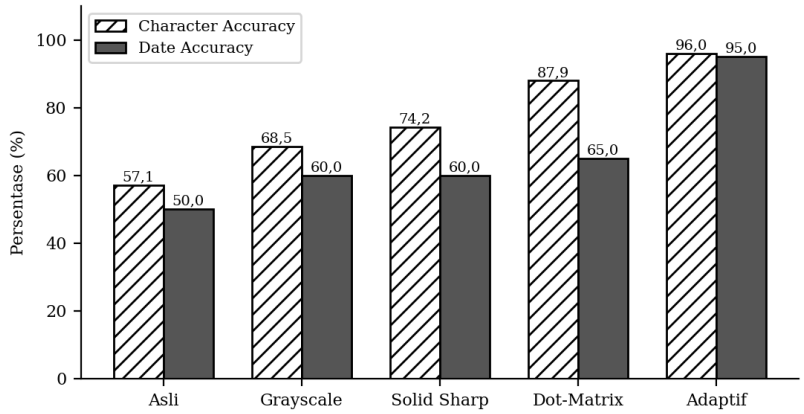
3.3. Analisis Komparatif Filter Adaptif pada 20 Sampel

Pada bagian ini kondisi ORI berperan sebagai baseline tanpa filter, dan *Grayscale*, *Solid Sharp*, serta *Dot-matrix* sebagai baseline filter tunggal terhadap skema adaptif. Tabel 4 membandingkan lima kondisi pada 20 sampel yang sama, dan Gambar 5 menyajikannya secara visual. Kedua ukuran ini naik seiring dengan citra asli hingga filter adaptif. Pada citra asli, ML Kit hanya menghasilkan tanggal yang benar pada 10 sampel (50%), sedangkan skema adaptif berhasil mencapai 19 sampel (95%) dengan *Character Accuracy* meningkat dari 57,08% menjadi 96,04%.

Tabel 4. Perbandingan Kondisi Citra pada 20 Sampel

Kondisi	Tanggal benar	Character Accuracy (%)	Date Accuracy (%)	CI 95%
Asli (ORI)	10/20	57,08	50	29,9-70,1
Grayscale	12/20	68,54	60	38,7-78,1
Solid Sharp	12/20	74,17	60	38,7-78,1
Dot-matrix	13/20	87,92	65	43,3-81,9
Adaptif	19/20	96,04	95	76,4-99,1

Vol. 11, No. 2, Oktober 2026, Page. 689-705
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI



Gambar 5. *Character Accuracy* dan *Date Accuracy* pada Lima Kondisi Citra

Tabel 5 memecah hasil menurut jenis cetakan, dan polanya sekarang terbagi menjadi dua kelompok. Pada enam sampel bercetak solid (sampel 9, 12, 13, 18, 19, dan 20), pada seluruh kondisi, termasuk citra asli, selalu membaca tanggal dengan benar, sehingga filter sama sekali tidak diperlukan pada cetakan jenis ini. Perbedaan besar justru terlihat pada 14 sampel *dot-matrix*: citra asli hanya benar pada 4 sampel (28,6%), *Grayscale* pada 6 sampel (42,9%), *Solid Sharp* pada 6 sampel (42,9%), *Dot-matrix* pada 7 sampel (50,0%), dan skema adaptif pada 13 sampel (92,9%). Skema adaptif hanya gagal pada satu sampel dari 20 sampel uji mikro, yaitu sampel 16. Angka tersebut perlu dibaca bersama rancangan skemanya: skema adaptif mendapat hingga tiga percobaan pada setiap sampel, sedangkan setiap filter tunggal hanya satu kali, sehingga sebagian keunggulannya memang berasal dari rancangan skema itu sendiri. Pada uji mikro, skema adaptif juga mencapai batas atas (oracle) dari ketiga filter, karena semua sampel kecuali sampel 16 terbaca benar pada minimal satu filter (Tabel 6).

Tabel 5. Jumlah Tanggal Benar menurut Jenis Cetakan

Jenis cetak	**Sampel**	**ORI**	***Grayscale***	***Solid Sharp***	***Dot-matrix***	**Adaptif**
Dot-matrix	14	4	6	6	7	13
Solid	6	6	6	6	6	6
Total	**20**	**10**	**12**	**12**	**13**	**19**

Perbaikan awal terlihat setelah penerapan pembesaran 3 kali dan konversi ke skala keabuan. Kedua tahapan tersebut membuat sampel 1, 2, dan 6 yang sebelumnya tidak terbaca pada citra asli dapat terbaca. Namun, pengaruh masing-masing operasi belum dapat ditentukan secara terpisah karena keduanya selalu diterapkan bersamaan. Dibandingkan dengan *Grayscale*, *Solid Sharp* menghasilkan *Character Accuracy* yang lebih tinggi, yaitu 74,17% berbanding 68,54%, meskipun jumlah tanggal yang terbaca dengan benar tetap sama, yaitu 12 sampel. Di antara ketiga kondisi filter tunggal, *Dot-matrix* menghasilkan *Date Accuracy* tertinggi sebesar 65% dan *Character Accuracy* sebesar 87,92%. Filter ini berhasil membaca sampel 3, 7, dan 8 yang tidak terbaca menggunakan *Grayscale* maupun *Solid Sharp*. Hasil tersebut sejalan dengan temuan Madake dkk. yang menunjukkan penurunan akurasi sebesar 5,6% ketika prapemrosesan tidak diterapkan [7], meskipun penelitian tersebut menggunakan data dan mesin OCR yang berbeda.

Perbedaan hasil antarfilter menunjukkan bahwa tidak ada satu filter pun yang mampu memberikan pembacaan yang benar pada seluruh sampel. Seperti ditunjukkan pada Tabel 6, *Solid Sharp* berhasil membaca sampel 10, 11, dan 14 yang gagal pada *Dot-matrix*. Sebaliknya, *Dot-matrix* berhasil pada sampel 3, 6, 7, dan 8 yang tidak berhasil dibaca menggunakan *Solid Sharp*. Penerapan filter juga dapat mengubah hasil yang sebelumnya sudah benar. Pada sampel 11, citra asli dan *Grayscale* menghasilkan pembacaan “11 03 07”, sedangkan *Dot-matrix* menghasilkan “13 03 27”. Akibatnya, aplikasi menampilkan 13/03/2027 sebagai tanggal yang valid, meskipun tanggal tersebut tidak sesuai dengan hasil yang benar. Pada sampel 14, citra asli dapat dibaca dengan benar, sementara *Grayscale* dan *Dot-matrix* justru tidak dapat dibaca. Hal serupa terlihat pada sampel 1 dan 2, ketika penerapan filter biner mengubah atau menghilangkan digit. Sebagai contoh, “011 026” terbaca menjadi “911 026”, sedangkan

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Grayscale masih dapat dibaca dengan utuh. Hasil ini menunjukkan bahwa binerisasi tidak selalu membantu, terutama pada karakter tipis yang sebenarnya sudah cukup jelas pada citra awal. Berdasarkan temuan tersebut, *Grayscale* digunakan sebagai filter awal, sedangkan *Solid Sharp* dan *Dot-matrix* digunakan sebagai alternatif jika pembacaan awal belum berhasil.

Tabel 6. Ketepatan Tanggal per Sampel pada Uji Mikro

No	Ground Truth	Bahan	Cetak	ORI	*Grayscale*	*Solid Sharp*	*Dot-matrix*	Adaptif
1	011026	Plastik	DM	X	B	X	X	B (Gray)
2	07012027	Kaleng	DM	X	B	X	X	B (Gray)
3	16102026	Kaleng	DM	X	X	X	B	B (Dot)
4	210926	Karton	DM	X	X	B	B	B (Solid)
5	072026	Karton	DM	X	X	B	B	B (Solid)
6	270528	Kaleng	DM	X	B	X	B	B (Gray)
7	151126	Karton	DM	X	X	X	B	B (Dot)
8	220926	Kaleng	DM	X	X	X	B	B (Dot)
9	102027	Karton	Solid	B	B	B	B	B (Gray)
10	190227	Karton	DM	X	X	B	X	B (Solid)
11	110327	Kaleng	DM	B	B	B	X	B (Gray)
12	150927	Plastik	Solid	B	B	B	B	B (Gray)
13	130427	Plastik	Solid	B	B	B	B	B (Gray)
14	271126	Plastik	DM	B	X	B	X	B (Solid)
15	201026	Plastik	DM	B	B	B	B	B (Gray)
16	19012027	Kaleng	DM	X	X	X	X	X
17	31052026	Kaleng	DM	B	B	X	X	B (Gray)
18	250127	Karton	Solid	B	B	B	B	B (Gray)
19	21012027	Plastik	Solid	B	B	B	B	B (Gray)
20	200127	Plastik	Solid	B	B	B	B	B (Gray)

B = tanggal benar, x = tanggal salah atau tidak dikenali. DM = *dot-matrix*. Kolom Adaptif memuat filter yang menghentikan pencarian.

Hasil pengujian menunjukkan bahwa ketiga filter memberikan hasil yang berbeda pada setiap sampel. Dalam skema adaptif, sebanyak 12 sampel dapat diselesaikan pada percobaan pertama menggunakan *Grayscale*, 4 sampel pada percobaan kedua menggunakan *Solid Sharp*, dan 3 sampel pada percobaan ketiga menggunakan *Dot-matrix*. Dengan demikian, rata-rata diperlukan 1,6 kali pembacaan ML Kit untuk setiap sampel. Dibandingkan dengan citra asli, skema adaptif menghasilkan perbaikan pada sembilan sampel dan tidak menunjukkan penurunan pada sampel lainnya. Perbedaan tersebut signifikan berdasarkan uji McNemar eksak ($p = 0{,}004$), sedangkan perbandingan *Character Accuracy* per sampel menggunakan uji Wilcoxon menghasilkan $p = 0{,}003$.

Keberhasilan skema adaptif ditentukan oleh kriteria penghentian berupa hasil tanggal yang memenuhi pola valid. Pada 20 sampel uji, skema ini tidak menghasilkan tanggal valid yang salah. Namun, hasil pada sampel 11 menunjukkan bahwa suatu filter tetap dapat menghasilkan tanggal yang valid secara format, tetapi tidak sesuai dengan *ground truth*. Pada pengujian makro, kondisi serupa ditemukan pada sekitar 22% tanggal yang ditampilkan dalam format yang valid, dan pada sampel tersebut tidak ada filter lain yang menghasilkan tanggal valid sekaligus benar. Dengan demikian, kriteria penghentian berdasarkan validitas format dapat membantu menemukan hasil pada cetakan yang sebelumnya tidak terbaca, tetapi belum dapat memastikan bahwa setiap digit yang dihasilkan sudah benar.

Empat sampel tambahan, yaitu sampel 17-20, menunjukkan bahwa hasil pembacaan tidak hanya dipengaruhi oleh jenis cetakan, tetapi juga oleh kondisi permukaan kemasan. Sampel 18 – 20 dengan cetakan *solid* dapat dibaca dengan benar pada seluruh kondisi, sedangkan sampel 17 yang menggunakan *dot-matrix* tebal pada kaleng justru mengalami kesalahan setelah penerapan *Solid Sharp* dan *Dot-matrix*. Namun, skema adaptif tetap menghasilkan pembacaan yang benar karena prosesnya berhenti pada tahap *Grayscale*. Hal ini menunjukkan bahwa karakteristik permukaan kemasan turut memengaruhi hasil filter.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Hasil tersebut perlu dilihat dengan mempertimbangkan keterbatasan sampel uji. Interval kepercayaan beberapa kondisi masih saling tumpang tindih sehingga belum menunjukkan keunggulan statistik antarfilter, sementara 14 dari 20 sampel menggunakan cetakan *dot-matrix*. Oleh karena itu, Date Accuracy 95% pada skema adaptif belum dapat digeneralisasikan sebagai akurasi sistem untuk seluruh jenis kemasan.

3.4. Analisis Galat

Lima belas sampel uji makro yang tanggalnya salah dikelompokkan berdasarkan tahap kegagalannya (Tabel 7). Kegagalan pada OCR mendominasi dengan 10 dari 15 sampel, disusul oleh Regex dengan 3 sampel dan lokalisasi YOLO dengan 2 sampel. Pada sampel yang menghasilkan *crop*, ketiga filter sudah dicoba dan tidak ada yang menghasilkan tanggal benar, sehingga kegagalan tersebut tidak dapat diatasi hanya dengan mengganti filter.

Tabel 7. Pengelompokan 15 Galat pada Uji Makro

Tahap	Sampel	Jumlah	Gejala
Lokalisasi YOLO	15, 25	2	Kotak tidak tepat (15); tidak ada kotak di atas ambang *threshold 0*,30 (25)
OCR tanpa teks	26, 33	2	ML Kit tidak mengembalikan teks
OCR kacau	19, 32	2	Teks tercampur atau tidak bermakna
Salah digit	3, 5, 8, 9, 20, 29	6	Digit tertukar dengan digit atau huruf mirip
Regex	7, 10, 45	3	Hari hilang, pemisah hilang, kata kunci rusak

Salah membaca digit menjadi kesalahan yang paling banyak ditemukan, seperti 1 menjadi 2 pada sampel 3, 1 menjadi 3 pada sampel 5, 8 menjadi 3 pada sampel 8, dan 2 menjadi 7 pada sampel 29. Pada sampel 9 dan 20, digit 6 dan 3 terbaca sebagai huruf S yang kemudian dipetakan menjadi 5 oleh tabel koreksi. Kondisi ini menunjukkan bahwa karakter yang memiliki beberapa kemungkinan interpretasi tidak selalu dapat diperbaiki hanya dengan satu aturan pemetaan.

Tiga sampel lainnya gagal pada tahap Regex. Pada sampel 45, koreksi karakter yang dilakukan sebelum pencarian kata kunci mengubah huruf D pada “ED” menjadi 0 sehingga kata kunci tidak dikenali dan menghasilkan 01/07/2027, bukan 07/07/2027. Pada sampel 7, teks “ED 21 0428” dikenali sebagai pola MMYY sehingga menghasilkan 01/04/2028 dan menghilangkan informasi hari 21. Sementara itu, teks “D4/032028” pada sampel 10 tidak cocok dengan pola regex karena pemisah antara bulan dan tahun tidak terbaca. Ketiga kasus tersebut menunjukkan bahwa perbaikan dapat dilakukan pada urutan proses koreksi dan cakupan Regex tanpa mengubah model deteksi.



Gambar 6. Contoh Kegagalan pada Uji Makro

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Gambar 6 menunjukkan beberapa kasus galat lainnya. Pada sampel 19, kotak deteksi mencakup baris MFG dan EXP secara bersamaan sehingga ML Kit menggabungkannya menjadi "EXP 1e 2026/2029". Kondisi ini berkaitan dengan aturan anotasi blok pada Subbab 2.1 dan pemilihan satu kotak dengan *confidence* tertinggi. Ketika dua baris tanggal berada berdekatan, sistem belum dapat memisahkannya. Pada sampel 25, meskipun cetakan hitam pada label putih terlihat jelas, tidak terdapat kotak deteksi dengan *confidence* ≥ 0,30. Penulis menduga kondisi ini berkaitan dengan posisi tanggal kedaluwarsa yang berada tepat di bawah tanggal produksi serta format YYYYMMDD tanpa pemisah, tetapi dugaan tersebut belum diuji lebih lanjut. Sementara itu, pada sampel 33, kotak deteksi sudah berada pada lokasi yang tepat, tetapi ML Kit tidak menghasilkan teks pada konfigurasi *Grayscale*. *Solid Sharp* dan *Dot-matrix* juga tidak menghasilkan tanggal yang terbaca pada sampel ini, sehingga kegagalannya tidak dapat diatasi oleh filter yang tersedia.

Pada uji mikro, sampel 16 yang hanya berupa kaleng bercetak *dot-matrix* dan tanggal 19012027 tetap menghasilkan pembacaan yang salah pada semua kondisi. Pada filter *Dot-matrix*, hasil OCR menjadi "190102" karena dua digit tahun tidak terbaca. Kondisi ini menunjukkan bahwa pantulan pada permukaan logam atau plastik yang mengilap dapat menghilangkan informasi karakter sehingga sulit dipulihkan melalui pemrosesan citra. Temuan serupa terlihat pada sampel 26 dan 33 dalam uji makro. Namun, intensitas pantulan tidak diukur secara langsung dalam penelitian ini, sehingga hubungan tersebut masih bersifat dugaan. Dospinescu dkk. juga melaporkan galat OCR pada permukaan yang mengilap atau melengkung, terutama ketika cap tanggal berukuran kecil dan memiliki kontras rendah [8].

3.5. Perbandingan dengan Baseline OCR Alternatif

Tabel 8 membandingkan kedua OCR pada 20 crop yang sama. Pada citra asli, PaddleOCR membaca 15 sampel dengan benar (75%), lebih banyak daripada ML Kit (10 sampel; 50%), terutama pada cetakan *dot-matrix* (9 dibanding 4 dari 14 sampel). Setelah pemrosesan, hasilnya berbalik: *Date Accuracy* PaddleOCR turun menjadi 60% (*Grayscale*) dan 45% (*Solid Sharp* serta *Dot-matrix*), sehingga skema adaptif hanya mencapai 70%, sedangkan ML Kit mencapai 95%. Selisih pada skema adaptif ini belum signifikan pada 20 sampel (McNemar eksak, p = 0,0625).

Tabel 8. Date Accuracy (DA) Google ML Kit dan PaddleOCR pada 20 sampel

Kondisi	ML Kit: benar	ML Kit: DA (%)	PaddleOCR: benar	PaddleOCR: DA (%)
Asli (ORI)	10/20	50	15/20	75
Grayscale	12/20	60	12/20	60
Solid Sharp	12/20	60	9/20	45
Dot-matrix	13/20	65	9/20	45
Adaptif	19/20	95	14/20	70
Adaptif (ORI pertama)	19/20	95	16/20	80

Hasil ini menunjukkan bahwa filter adaptif berfungsi sebagai tahap bantu bagi ML Kit, dan manfaatnya tidak otomatis berlaku untuk OCR lain. ML Kit tidak mengembalikan teks pada 7 dari 20 citra asli (sampel 1, 2, 6, 7, 8, 10, dan 16), sedangkan PaddleOCR menghasilkan teks pada seluruh citra asli. Filter *Solid Sharp* dan *Dot-matrix* dirancang untuk menutup kelemahan ML Kit pada cetakan titik, dan pada ML Kit jumlah tanggal benar naik dari 10 menjadi 12 hingga 13 sampel. Pada PaddleOCR, kedua filter lebih sering mengubah atau menghilangkan digit, bahkan pada enam sampel bercetak solid, hanya 3 sampel yang tetap terbaca benar. Karena itu, penggunaan filter pada PaddleOCR tidak dapat direkomendasikan sebagai pengolahan bawaan. Meski demikian, filter masih membantu pada sebagian kecil sampel: sampel 3 dan 17 tidak terbaca pada citra asli, tetapi terbaca dengan benar setelah *Dot-matrix* dan *Solid Sharp*.

Pada PaddleOCR, kriteria penghentian berdasarkan format tanggal belum menghasilkan pemilihan hasil pembacaan yang optimal. Skema berhenti pada tanggal valid yang salah pada sampel 1 (*Grayscale*, 11/10/2026), sampel 10 (*Solid Sharp*, 09/02/2027), dan sampel 19 (*Dot-matrix*, 01/01/2027), serta tidak ada filter yang menghasilkan tanggal valid pada tiga sampel (2, 7, dan 16). Pada sampel 1, 10, dan 19,

citra asli terbaca benar tetapi skema adaptif gagal. Bila citra asli dijadikan langkah pertama (urutan ORI, *Grayscale*, *Solid Sharp*, kemudian *Dot-matrix*) dan dihitung dari pembacaan yang sama, *Date Accuracy* PaddleOCR naik dari 70% menjadi 80% (16 dari 20 sampel) dengan rata-rata 1,6 pembacaan per sampel. Pada ML Kit, urutan tersebut tetap menghasilkan 95% dengan rata-rata 2,05 pembacaan per sampel. Selisih ML Kit dan PaddleOCR pada urutan ini tetap lebih kecil daripada pada urutan yang dimulai dari *Grayscale*, tetapi ML Kit masih lebih tinggi.

3.6. Waktu Komputasi dan Trade-off Akurasi

Tabel 9 membandingkan waktu pembacaan OCR pada 20 crop yang sama. PaddleOCR lebih lambat pada keempat kondisi citra. Rata-rata waktunya 2,24 kali waktu ML Kit pada citra asli (203,25 dibanding 90,60 ms) dan 3,71 sampai 4,30 kali pada citra berfilter (520,05 sampai 697,95 dibanding 139,30 sampai 162,30 ms). Selisih berpasangan ini signifikan pada keempat kondisi (uji Wilcoxon, $p < 0,001$).

Tabel 9. Waktu Pembacaan OCR per Citra (rata-rata ± simpangan baku, ms)

Kondisi	ML Kit (ms)	PaddleOCR (ms)	Rasio Paddle/ML Kit
Asli (ORI)	90,60 ± 13,09	203,25 ± 155,88	2,24
Grayscale	140,30 ± 41,81	520,05 ± 356,75	3,71
Solid Sharp	139,30 ± 38,07	588,30 ± 560,53	4,22
Dot-matrix	162,30 ± 46,51	697,95 ± 599,30	4,30

Dari sisi kestabilan waktu pembacaan, PaddleOCR menunjukkan variasi yang lebih besar dibandingkan dengan ML Kit. Simpangan baku waktunya 155,88 sampai 599,30 ms, sedangkan ML Kit 13,09 sampai 46,51 ms. Waktu pembacaan terlama PaddleOCR mencapai 2.295 ms (sampel 8, *Dot-matrix*), dibandingkan dengan 270 ms pada ML Kit. PaddleOCR melewati 1.000 ms pada 10 dari 80 pembacaan, sedangkan ML Kit tidak pernah melewati 270 ms. Penerapan filter menaikkan waktu ML Kit 1,5 sampai 1,8 kali dibandingkan citra asli, dan waktu PaddleOCR 2,6 sampai 3,4 kali. Citra berfilter berukuran tiga kali citra asli, tetapi pengaruh ukuran masukan terhadap waktu tidak diperiksa lebih lanjut.

Hasil tersebut menunjukkan bahwa akurasi dan waktu pembacaan perlu dipertimbangkan secara bersamaan. Pada citra asli, PaddleOCR lebih akurat daripada ML Kit (75% dibanding 50%; McNemar eksak, $p = 0,125$, belum signifikan pada 20 sampel) dengan waktu baca 2,24 kali lebih lama. Pada filter tunggal, ML Kit memberikan akurasi yang sama atau lebih tinggi (60%, 60%, dan 65% dibandingkan 60%, 45%, dan 45%), dengan waktu pembacaan yang lebih singkat. Waktu PaddleOCR mencapai 3,71–4,30 kali waktu ML Kit pada ketiga kondisi tersebut. Pada skema adaptif, ML Kit mencapai 95% dibandingkan dengan 70% pada PaddleOCR, tetapi waktu skema ini tidak diukur. Dalam nilai mutlak, selisih rata-rata waktu per pembacaan berkisar antara 112,65 sampai 535,65 ms, sehingga dampaknya bagi pengguna belum diuji. Selain lebih lambat, aplikasi dengan PaddleOCR pada konfigurasi ini sesekali berhenti sendiri saat dijalankan di ponsel, sedangkan versi ML Kit tidak mengalaminya pada pengujian yang sama. Dengan mempertimbangkan akurasi, efisiensi waktu pembacaan, dan kestabilan eksekusi pada perangkat yang diuji, ML Kit dipertahankan sebagai OCR utama. Kombinasi filter adaptif dan ML Kit menghasilkan Date Accuracy tertinggi dari seluruh kombinasi yang diuji, yaitu 95%, dibandingkan dengan 70% pada PaddleOCR dengan skema adaptif. Klaim penelitian ini, karena itu dibatasi pada filter adaptif sebagai pendukung pembacaan ML Kit, bukan sebagai perbaikan untuk sembarang OCR.

3.7. Ancaman Validitas dan batas generalisasi

Hasil penelitian ini berlaku pada cakupan berikut. *Date Accuracy* 95% berasal dari 20 sampel uji mikro (14 di antaranya *dot-matrix*) yang dipilih berdasarkan ketersediaan produk, dengan CI 95% 76,4–99,1%, sehingga tidak otomatis berlaku untuk semua kemasan pangan. Seluruh citra diambil dengan dua ponsel di tiga toko ritel di Pangkalpinang pada cahaya ruangan, sehingga kinerja di luar domain tersebut dapat berbeda. PaddleOCR yang diuji adalah model ringan PP-OCRv2 Lite untuk perangkat bergerak, sehingga hasilnya tidak mewakili versi PaddleOCR yang lebih besar atau lebih baru. Ada pula risiko overfitting: data dibagi acak per citra oleh Roboflow, padahal setiap produk difoto dari tiga sudut (tegak lurus serta miring ke kiri dan ke kanan sekitar 5° sampai 15°), sehingga foto produk yang sama dapat

berada di lebih dari satu subset. Jumlah produk yang beririsan antarsubset tidak dapat dihitung karena identitas produk tidak dicatat saat pembagian data, sehingga mAP@50 pada data uji dapat lebih tinggi daripada pada produk yang sama sekali baru. Sampel uji makro dan mikro berasal dari produk yang terpisah dari 869 citra pelatihan, sehingga *Date Accuracy* dan *Character Accuracy* tidak terkena risiko ini, tetapi sebagian produk muncul pada kedua uji tersebut, sehingga hasilnya tidak sepenuhnya independen. Parameter filter disetel secara empiris dengan ML Kit sebagai acuan, tanpa pemisahan formal antara sampel penyetelan dan uji, sehingga perbandingan dengan PaddleOCR cenderung menguntungkan ML Kit. *Ground truth* dicatat dengan bantuan beberapa orang tanpa ukuran kesepakatan antarpenilai. Pada uji makro dan mikro, tanggal valid yang tidak sesuai dengan *ground truth* diperiksa ulang dengan filter lain. Pada uji mikro, hal ini tidak mengubah hasil karena tidak ada tanggal valid yang salah muncul sebelum tanggal yang benar, sedangkan pada uji makro hasilnya dapat lebih tinggi daripada skema adaptif murni. Waktu komputasi hanya mencakup tahap OCR, tanpa waktu YOLO, filter OpenCV, skema adaptif, maupun penggunaan memori, sehingga belum menggambarkan total waktu sistem. Evaluasi statistik akurasi hanya membandingkan kondisi ORI dengan skema adaptif, sedangkan uji Wilcoxon digunakan untuk membandingkan waktu pembacaan OCR. Selain itu, kriteria penghentian berbasis Regex hanya memastikan keluaran memenuhi format tanggal dan tidak dapat mendeteksi kesalahan digit yang tetap menghasilkan tanggal valid; pada uji makro, 22,22% keluaran dengan format valid masih merupakan tanggal yang salah.

4. KESIMPULAN

Penelitian ini menguji arsitektur ekstraksi tanggal kedaluwarsa yang terdiri atas YOLOv11 Nano, filter citra, Google ML Kit, dan Regex, serta menghasilkan empat temuan pokok. Pertama, lokalisasi area tanggal oleh YOLOv11 Nano sudah memadai: model mencapai mAP@50 sebesar 0,917 pada data uji dan berhasil menemukan area yang benar pada 48 dari 50 produk (96%), sehingga kegagalan pada tahap lokalisasi hanya menyumbang sebagian kecil dari kegagalan sistem pada pengujian ini. Kedua, pada uji 50 produk dengan koreksi filter berdasarkan *ground truth*, sistem menghasilkan *Character Accuracy* rata-rata 84% dan Date Accuracy 70%. Dari 15 kegagalan yang tercatat, 10 terjadi pada pembacaan OCR, 3 pada Regex, dan 2 pada lokalisasi. Sementara itu, tujuh di antaranya hanya berselisih satu digit dari tanggal yang benar, sehingga akurasi tanggal sangat peka terhadap satu karakter yang keliru.

Ketiga, penerapan filter citra memiliki pengaruh terhadap akurasi pembacaan tanggal kedaluwarsa menggunakan Google ML Kit, khususnya pada citra dengan cetakan *dot-matrix*. Pada uji mikro, enam sampel bercetak solid selalu terbaca dengan benar pada semua kondisi, sedangkan pada 14 sampel *dot-matrix*, citra asli hanya terbaca dengan benar pada 4 sampel (28,6%). Tidak terdapat satu filter tunggal yang unggul pada seluruh sampel dikarenakan ketiga filter: *Grayscale*, *Solid Sharp*, dan *Dot-matrix* memberikan hasil pembacaan yang berbeda sesuai dengan kondisi citra, sedangkan *Dot-matrix* dapat menyebabkan perubahan pada digit yang sebelumnya telah terbaca dengan benar. Dengan mencoba ketiganya secara bertahap tanpa memerlukan *ground truth* (pada uji ini hasilnya sama dengan pemeriksaan ulang berdasarkan *ground truth*), *Date Accuracy* naik dari 50% menjadi 95% dengan rata-rata 1,6 kali pembacaan per sampel, sementara *Character Accuracy* naik dari 57,08% menjadi 96,04%.

Keempat, pada 20 sampel yang sama, PaddleOCR mencapai *Date Accuracy* 75% pada citra asli, tetapi hanya 70% dengan skema adaptif (80% bila citra asli dijadikan langkah pertama), sedangkan ML Kit mencapai 95% dengan skema yang sama. Manfaat filter adaptif bergantung pada mesin OCR dan kondisi citra yang dihadapi. PaddleOCR juga membutuhkan waktu pembacaan 2,24 sampai 4,30 kali lebih lama daripada ML Kit pada keempat kondisi citra, dan aplikasi pada konfigurasi tersebut sesekali berhenti sendiri pada ponsel. Pada perbandingan skema adaptif, perbedaan *Date Accuracy* antara ML Kit dan PaddleOCR belum signifikan secara statistik (McNemar eksak, $p = 0{,}0625$).

Dalam penerapannya, hasil tersebut masih memiliki beberapa batasan. Kriteria penghentian berdasarkan tanggal berformat valid belum dapat mendeteksi kesalahan digit yang tetap menghasilkan tanggal valid. Pada uji 50 produk, sekitar 22,22% tanggal yang berhasil ditampilkan dalam format valid masih tidak sesuai dengan *ground truth*, sehingga hasil tetap perlu diverifikasi oleh pengguna sebelum disimpan. Selain itu, hasil 95% berasal dari 20 sampel uji mikro (14 di antaranya *dot-matrix*) yang dipilih menurut ketersediaan produk, sehingga belum dapat digeneralisasikan ke seluruh kemasan pangan, dan jumlah sampel yang terbatas membuat perbedaan antarfilter tunggal belum dapat dipastikan.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Pengembangan selanjutnya dapat mencakup pengujian pada lebih banyak sampel dan kondisi pencahayaan, evaluasi model OCR alternatif atau versi yang lebih baru, penyempurnaan koreksi karakter dan pencarian kata kunci pada Regex, serta pemisahan baris tanggal yang berdekatan.

5. DAFTAR PUSTAKA

[1] Badan Pengawas Obat dan Makanan, "Pentingnya Cek Tanggal Kedaluwarsa Pangan Olahan." [Online]. Available: https://serang.pom.go.id/berita/bbpom-serang-ingatkan-pentingnya-cek-tanggal-kedaluwarsa-pangan-olahan

[2] World Health Organization, "Estimating the burden of foodborne diseases." [Online]. Available: https://www.who.int/news/item/04-06-2026-unsafe-food-causes-866-million-illnesses-and-1.5-million-deaths-annually--young-children-at-highest-risk

[3] A. Najlae, A. Sedqui, and A. Lyhyaoui, "A Product Driven System to Facilitate FEFO Application in Warehouses," *Procedia Comput. Sci.*, vol. 191, pp. 451–456, 2021, doi: 10.1016/j.procs.2021.07.056.

[4] A. C. Seker and S. C. Ahn, "A generalized framework for recognition of expiration dates on product packages using fully convolutional networks," *Expert Syst. Appl.*, vol. 203, Oct. 2022, doi: 10.1016/j.eswa.2022.117310.

[5] C. Agatha and Lina, "Pengenalan Tanggal Kedaluwarsa Pada Kemasan Produk Dengan Metode Convolutional Neural Network," *Jurnal Sistem Informasi Dan Teknik Komputer*, vol. 8, no. 2, pp. 290–294, 2023, doi: 10.51876/simtek.v8i2.210.

[6] Richo, "Jurnal Rekayasa Sistem Informasi dan Teknologi Volume 2, No 3 Februari 2025 e-ISSN : 3025-888X Sistem Identifikasi Informasi Expired Date Produk Kemasan Menggunakan Kolaborasi Metode Yolo-V11m Dan Paddleocr," Feb. 2025. doi: 10.70248/jrsit.v2i3.1719.

[7] J. Madake, K. Kalamkar, and P. Katepallewar, "ExpirySense: Expiry Date Detection and Proactive Alerts," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. 12, pp. 658–671, Dec. 2025, doi: 10.22214/ijraset.2025.76176.

[8] O. Dospinescu, G. L. Grigorcea, and B. I. Lefter, "A Mobile Approach to Food Expiration Date Determination Using OCR and On-Cloud Image Classification," *Applied System Innovation*, vol. 8, no. 6, Dec. 2025, doi: 10.3390/asi8060176.

[9] M. H. Emel, M. Terzioğlu, and R. Özkan, "Efficient and Accurate Date Extraction from Invoices: A Comprehensive Three-Step Methodology Integrating Custom Object Detection, OCR, and Refined Regular Expressions," *Advances in Artificial Intelligence Research*, vol. 4, no. 1, pp. 10–17, 2024, doi: 10.54569/aair.1401234.

[10] G. Ramadhan, D. I. Mulyana, and S. Adrianto, "Optimization of Tesseract OCR for Automatic Text Extraction on Indonesian ID Cards (KTP) Through Image Quality Enhancement Using Preprocessing Techniques," *International Journal Software Engineering and Computer Science (IJSECS)*, vol. 5, no. 3, pp. 938–948, 2025, doi: 10.35870/ijsecs.v5i3.5183.

[11] L. R. Mursari and A. Wibowo, "The Effectiveness of Image Preprocessing on Digital Handwritten Scripts Recognition with The Implementation of OCR Tesseract," *Computer Engineering and Applications Journal*, vol. 10, no. 3, pp. 177–186, 2021, doi: 10.18495/comengapp.v10i3.386.

[12] Eva Urankar, "Waste Detection on Mobile Devices: Model Performance and Efficiency Comparison," *International Journal of Science and Research Archive*, vol. 15, no. 1, pp. 722–731, 2025, doi: 10.30574/ijsra.2025.15.1.1052.

[13] Ultralytics, "Ultralytics YOLO11," Ultralytics Documentations. [Online]. Available: https://docs.ultralytics.com/models/yolo11

[14] M. Viddya Zufar and N. Pratiwi, "Implementasi YOLOv11 dan Google ML Kit untuk Pembacaan Struk pada Aplikasi Keuangan Mobile," *Metik Jurnal*, vol. 9, p. 2025, 2025, doi: 10.47002/metik.v9i2.1087.

[15] B. Shi, W. Cheng, Y. Lu, C. Zhang, and D. Florencio, "Improving Structured Text Recognition with Regular Expression Biasing," pp. 1–8, 2021, [Online]. Available: http://arxiv.org/abs/2111.06738

[16] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," UK, 2024. [Online]. Available: http://arxiv.org/abs/2410.17725

[17] OpenCV, "OpenCV: Histograms - 2: Histogram Equalization," OpenCV 4.13.0 Documentation. Accessed: May 31, 2026. [Online]. Available: https://docs.opencv.org/4.13.0/d5/daf/tutorial_py_histogram_equalization.html
[18] OpenCV, "Image Thresholding." [Online]. Available: https://docs.opencv.org/4.13.0/d7/d4d/tutorial_py_thresholding.html
[19] S. Sivankalai and S. Balachandran, "Transforming Physical Archives into Searchable Digital Libraries with Optical Character Recognition," *Preservation, Digital Technology and Culture*, vol. 54, no. 4, pp. 263–284, 2025, doi: 10.1515/pdtc-2025-0025.
[20] OpenCV, "Morphological Transformations." [Online]. Available: https://docs.opencv.org/4.13.0/d9/d61/tutorial_py_morphological_ops.html
[21] Google Developers, "Mengenali teks di gambar dengan ML Kit di Android," Google for Developers. [Online]. Available: https://developers.google.com/ml-kit/vision/text-recognition/v2/android?hl=id
[22] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, no. 1, pp. 1–14, 2024, doi: 10.1038/s41598-024-56706-x.
[23] J. D. Gilbey and C.-B. Schönlieb, "An end-to-end Optical Character Recognition approach for ultra-low-resolution printed text images," 2021, [Online]. Available: http://arxiv.org/abs/2105.04515
[24] G. Shan, X. Lou, and S. S. Wu, "Continuity corrected Wilson interval for the difference of two independent proportions," vol. 22, pp. 38–53, 2023, doi: 10.1007/s44199-023-00054-8.Continuity.
[25] M. P. Fay and K. Lumbard, "Confidence intervals for difference in proportions for matched pairs compatible with exact McNemar's or sign tests," *Stat. Med.*, vol. 40, no. 5, pp. 1147–1159, 2021, doi: 10.1002/sim.8829.
[26] G. A. Ghadhban and H. A. Rasheed, "Robust tests for the mean difference in paired data using Jackknife resampling technique," *Iraqi Journal of Science*, vol. 62, no. 9, pp. 3081–3090, 2021, doi: 10.24996/ijs.2021.62.9.23.
[27] Y. Du *et al.*, "PP-OCRv2: Bag of Tricks for Ultra Lightweight OCR System," 2021, [Online]. Available: http://arxiv.org/abs/2109.03144
[28] R. Afriansyah *et al.*, "Performance Comparison Of Easyocr And Paddleocr In Yolo- Based Indonesian License Plate Recognition," vol. 12, no. 1, pp. 429–440, 2026, doi: 10.33480/jitk.v12i1.8217.