

Identifikasi Nomor Kontainer Menggunakan Resnet-50 Pada *Gate-In* PT. Pelindo Berlian Jasa Terminal Indonesia (Bjti) Berbasis *Website*

1st Muhammad Ali Ridho
Program Studi Teknologi Informasi
Universitas Telkom
Surabaya, Indonesia
muhammadaliridho@student.telkomuniversity.ac.id

2nd Yohanes Setiawan
Program Studi Teknologi Informasi
Universitas Telkom
Surabaya, Indonesia
yohanessetiawan@telkomuniversity.ac.id

3rd Farah Zakiyah Rahmanti
Program Studi Teknologi Informasi
Universitas Telkom
Surabaya, Indonesia
farahzakiyah@telkomuniversity.ac.id

Abstrak — Proses pencatatan nomor kontainer secara manual di area *gate-in* PT Pelindo Berlian Jasa Terminal Indonesia (BJTI) masih menimbulkan berbagai kendala seperti kesalahan pencatatan, keterlambatan pemrosesan data, dan kesulitan validasi dalam kasus klaim kerusakan. Untuk mengatasi masalah tersebut, penelitian ini mengembangkan sistem identifikasi nomor kontainer otomatis berbasis *deep learning* yang terdiri dari tiga tahap utama, yaitu: deteksi area nomor kontainer menggunakan model YOLOv11, segmentasi karakter dengan model YOLOv11 lainnya, serta klasifikasi karakter menggunakan arsitektur ResNet-50. Dataset pada setiap tahap dikumpulkan dari lingkungan nyata di pelabuhan dan telah dianotasi secara manual. Model deteksi area nomor kontainer yang dilatih dengan optimizer AdamW selama 200 epoch menghasilkan mAP@0.5 sebesar 96.8%, precision 97.1%, dan recall 95.4%. Pada tahap segmentasi karakter, model mencatatkan mAP@0.5 sebesar 94.3%, precision 95.2%, dan recall 93.7%. Sedangkan pada tahap klasifikasi karakter menggunakan ResNet-50, akurasi tertinggi yang diperoleh mencapai 98.6%, dengan precision 98.9%, recall 98.3%, dan F1-score 98.6%. Seluruh sistem kemudian diintegrasikan ke dalam platform berbasis web untuk mendukung operasional, memungkinkan penyimpanan hasil deteksi dalam bentuk teks dan gambar, serta dilengkapi fitur login, pencarian data, dan tampilan statistik. Evaluasi menunjukkan bahwa sistem mampu beroperasi dengan efisien dan akurat di berbagai kondisi pencahayaan dan sudut pandang kamera. Sistem ini diharapkan dapat meningkatkan efisiensi, akurasi, dan transparansi pencatatan nomor kontainer di lingkungan operasional pelabuhan.

Kata kunci— sistem deteksi otomatis, YOLOv11, ResNet-50, nomor kontainer, pengolahan citra, pelabuhan, website.

I. PENDAHULUAN

Dalam industri logistik modern, pencatatan nomor kontainer merupakan aspek penting yang mendukung kelancaran, akurasi, dan efisiensi operasional, khususnya di area *gate-in* pelabuhan. Di PT Pelindo Berlian Jasa Terminal Indonesia (BJTI), pencatatan masih dilakukan secara manual,

yang menimbulkan potensi kesalahan, keterlambatan input, serta kesulitan pelacakan saat jumlah kontainer meningkat drastis. Permasalahan ini memperlambat alur kerja dan menyulitkan verifikasi data, terutama saat terjadi klaim kerusakan pada kontainer.

Untuk menjawab tantangan tersebut, diperlukan sistem identifikasi nomor kontainer yang otomatis dan akurat di lingkungan kerja nyata, guna menggantikan proses manual dan mempercepat proses verifikasi. Berbagai studi telah dilakukan untuk mengatasi permasalahan pengenalan karakter pada objek bergerak, seperti pelat nomor kendaraan [6], kontainer [7], dan sistem identifikasi visual *real-time* berbasis *deep learning* [8]. Namun, sebagian besar penelitian tersebut masih memiliki keterbatasan, seperti ketergantungan pada kualitas citra tinggi, ketahanan rendah terhadap gangguan visual di lapangan, serta keterbatasan dalam menghadapi kondisi pencahayaan yang buruk atau objek yang rusak secara fisik [7], [8].

Penelitian ini menawarkan pendekatan berbasis *deep learning* untuk membangun sistem identifikasi nomor kontainer yang dapat diimplementasikan secara langsung di lingkungan operasional BJTI. Sistem ini terdiri dari tiga tahap utama. Tahap pertama adalah proses deteksi area nomor kontainer menggunakan model YOLOv11, yang mampu bekerja dengan baik meskipun dalam kondisi pencahayaan yang tidak ideal [6]. Tahap kedua adalah segmentasi karakter dari area nomor menggunakan model YOLOv11 lainnya. Segmentasi dilakukan untuk memisahkan masing-masing karakter dari nomor kontainer secara akurat, menggantikan metode tradisional seperti thresholding yang tidak konsisten di lingkungan nyata. Tahap ketiga adalah klasifikasi karakter menggunakan arsitektur ResNet-50, yang bertugas mengenali karakter hasil segmentasi dan mengklasifikasikannya ke dalam 36 kelas, yaitu huruf A sampai Z dan angka 0 sampai 9. Pendekatan ini dipilih karena kemampuannya dalam menangani karakter yang mengalami distorsi, *noise*, atau variasi bentuk yang umum terjadi pada kontainer yang telah lama digunakan [7].

Sebagai bagian dari rancangan sistem, hasil deteksi disimpan dalam bentuk teks dan citra yang mencakup visualisasi area deteksi, susunan karakter hasil klasifikasi, dan citra kontainer yang dikenali. Dokumentasi ini dapat digunakan untuk keperluan pelacakan atau verifikasi, terutama ketika terdapat klaim kerusakan pada badan kontainer. Dengan pengembangan sistem identifikasi nomor kontainer ini, proses pencatatan dapat dilakukan secara otomatis, akurat, dan terdokumentasi dengan baik. Hal ini diharapkan dapat meningkatkan efisiensi kerja di area *gate-in* serta memperkuat keandalan dan kecepatan operasional logistik di lingkungan PT Pelindo Berlian Jasa Terminal Indonesia (BJTI).

II. KAJIAN TEORI

A. Nomor Kontainer



GAMBAR 1
(NOMOR KONTAINER)

Nomor kontainer menjadi identitas unik yang wajib dicantumkan pada setiap peti kemas. Pada gambar 1 ditunjukkan Nomor kontainer ini berupa kode alfanumerik yang memuat informasi pemilik, nomor seri, dan digit pemeriksa sesuai standar ISO 6346 [1]. Fungsi utamanya adalah untuk mempermudah pelacakan, pencatatan, serta menjamin keamanan selama pengiriman.

B. You Only Look Once (YOLO)

You Only Look Once (YOLO) adalah metode deteksi objek yang memperlakukan proses deteksi sebagai masalah regresi, di mana bounding box dan probabilitas kelas diprediksi secara langsung dari keseluruhan citra dalam satu kali evaluasi. Konsep utamanya adalah dengan membagi citra menjadi sebuah grid, dengan setiap sel grid bertugas mendeteksi objek yang ada di dalamnya [11]. Pada Gambar 2 ditunjukkan arsitektur dari YOLO.

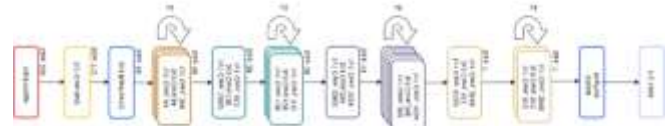
Iterasi terbarunya, YOLOv11 (rilis Oktober 2024), membawa peningkatan arsitektur signifikan seperti blok C3k2, SPPF, dan C2PSA untuk ekstraksi fitur yang lebih kuat. Model ini mengusung pendekatan anchor-free dengan decoupled head yang memisahkan tugas deteksi, klasifikasi, dan regresi untuk meningkatkan akurasi. Selain itu, arsitektur YOLOv11 juga mengimplementasikan transformer-based backbone dan dynamic head untuk fleksibilitas yang lebih baik terhadap variasi citra [12], [13].



GAMBAR 2
(ARSITEKTUR YOLO)

C. Resnet-50

ResNet50, yang pertama kali diperkenalkan oleh He dkk. pada tahun 2015, adalah jaringan saraf konvolusi dalam dengan 50 lapisan yang dirancang untuk menangani berbagai tugas visi komputer seperti klasifikasi citra, deteksi objek, dan pelokalan objek [2], [11]. Model ini adalah varian dari ResNet yang terdiri dari 48 lapisan konvolusi, satu lapisan MaxPool, dan satu lapisan Average Pool, dengan kemampuan melakukan sekitar $3,8 \times 10^9$ operasi titik mengambang [3], [11]. Dalam arsitekturnya, ResNet50 menggunakan lapisan awal untuk mengurangi ukuran spasial citra input dan meningkatkan jumlah saluran, diikuti oleh empat blok residual serta lapisan fully connected untuk proses klasifikasi [3], [11]. Blok residual ini mengadopsi desain bottleneck tiga lapisan yang menggantikan desain dua lapisan pada ResNet34, memungkinkan pelatihan jaringan yang lebih dalam dengan efisiensi lebih tinggi [11]. Seperti yang ditunjukkan pada gambar 3 merupakan arsitektur dari Resnet-50



GAMBAR 3
(ARSITEKTUR RESNET-50)

D. Confusion Matrix

Confusion matrix dimanfaatkan untuk menilai kinerja model klasifikasi, dengan menghasilkan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Masing-masing metrik dihitung berdasarkan nilai-nilai yang terdapat dalam matriks tersebut [4], [14]. Berikut merupakan rumus dari keempat metrik yang sudah disebutkan sebelumnya:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall \times Precision} \quad (4)$$

Keterangan:

- TP (*True Positive*) : Nilai ketika model dengan tepat mengklasifikasikan data yang memang seharusnya positif.
- TN (*True Negative*) : Nilai ketika model mampu mengenali dan memutuskan dengan benar bahwa data tersebut adalah negatif.
- FP (*False Positive*) : Nilai di mana model salah memprediksi hasil nilai negatif sebagai positif.
- FN (*False Negative*) : Nilai saat model salah dalam mengklasifikasikan data positif sebagai negatif.
- F1-Score : Perbandingan antara seberapa tepat model dalam membuat prediksi positif (*precision*) dengan seberapa baik model menangkap seluruh data positif yang ada (*recall*).

E. Framework Website untuk Implementasi

Flask, sebuah framework web minimalis, sangat ideal untuk menciptakan API yang menghubungkan sistem identifikasi nomor kontainer berbasis AI dengan antarmuka web. API ini memungkinkan pengguna di lapangan, seperti di pelabuhan, untuk mengakses hasil identifikasi (contoh: nomor dan kondisi kontainer) secara *online* dan fleksibel. Pendekatan ini memfasilitasi implementasi sistem verifikasi dan pencatatan kontainer yang otomatis, efisien, dan akurat dalam lingkungan operasional yang dinamis [9], [10].

F. Metrik Performa

Beberapa parameter akurasi diterapkan untuk menghitung kemampuan metode identifikasi objek ini untuk wadah identifikasi. *Intersection Over Union* (IoU) bertanggung jawab untuk mengukur seberapa banyak kebenaran dasar dalam sebuah citra yang telah teridentifikasi. Persamaan menunjukkan cara menghitung IoU. Untuk menghitung IoU diberlakukan rumus seperti berikut.

$$IoU = \frac{A(B_p \cap B_{gt})}{A(B_p \cup B_{gt})} = \frac{\text{Overlapping Region}}{\text{Combined Region}} \quad (5)$$

Dimana A adalah fungsi untuk menghitung area. B_p menunjukkan bounding box yang diprediksi dan B_{gt} menunjukkan *ground truth* kotak pembatas [15]. Presisi adalah kemampuan model untuk mengenali hanya objek yang relevan. Ini adalah persentase prediksi positif dan diberikan sebagai persamaan berikut.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{\text{all detection}} \quad (6)$$

$$= \frac{\text{Correctly detected container number}}{\text{all detection}}$$

Recall adalah kemampuan model untuk mengenali semua kebenaran. *Recall* mengukur persentase positif yang benar dari semua kebenaran dasar.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{\text{all detection}} \quad (7)$$

$$= \frac{\text{Correctly detected container number}}{\text{all detection}}$$

Kinerja detektor dievaluasi menggunakan kurva presisi-recall, yang luas areanya menunjukkan keseimbangan antara presisi dan *recall*. Evaluasi ini didasarkan pada *True Positive* (TP) untuk deteksi yang benar, *False Positive* (FP) untuk deteksi keliru, dan *False Negative* (FN) untuk objek yang terlewat. Detektor yang baik akan mempertahankan presisi tinggi saat *recall* meningkat, menghasilkan area di bawah kurva yang luas. Sebaliknya, upaya untuk meningkatkan *recall* sering kali menaikkan jumlah *false positive*, yang menyebabkan presisi menurun. Karena itu, grafik umumnya menunjukkan tren presisi yang menurun seiring naiknya *recall* [5], [16].

III. METODE

Metodologi penelitian ini disusun secara sistematis untuk mengembangkan dan mengimplementasikan sistem identifikasi nomor kontainer otomatis di PT Pelindo Berlian Jasa Terminal Indonesia (BJTI). Penelitian ini bertujuan mengatasi inefisiensi dan kerentanan kesalahan pada proses pencatatan manual dengan menerapkan model *deep learning* YOLOv11 dan arsitektur ResNet-50. Alur penelitian secara keseluruhan digambarkan pada gambar 4 di bawah ini.



GAMBAR 4
(FLOWCHART PROSEDUR PENELITIAN)

Gambar di atas menunjukkan alur kerja penelitian yang dimulai dari identifikasi masalah, yaitu pencatatan nomor kontainer manual yang lambat dan rentan kesalahan, terutama dalam kondisi pencahayaan dan sudut pandang kamera yang bervariasi. Tahapan dilanjutkan dengan studi literatur untuk mengkaji model YOLOv11 untuk deteksi dan ResNet-50 untuk klasifikasi karakter. Setelah itu, dilakukan perancangan sistem, pengumpulan data, *preprocessing*, pengembangan dan pengujian model, hingga integrasi sistem dan evaluasi akhir.

A. Perancangan Sistem

Sistem dirancang dengan kerangka kerja berbasis modul untuk menangani setiap tahapan proses identifikasi nomor kontainer, mulai dari akuisisi data hingga integrasi dengan antarmuka pengguna. Seperti yang digambarkan pada gambar 5.

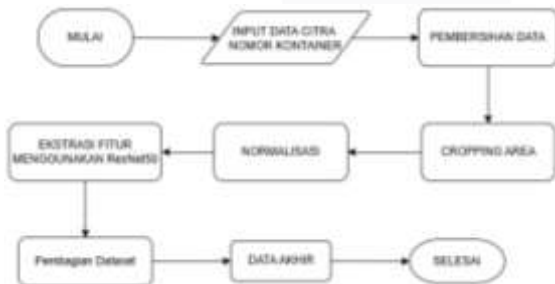


GAMBAR 5 (FLOWCHART PERANCANGAN SISTEM)

Rancangan sistem ini terdiri dari beberapa modul utama. Modul Akuisisi Data bertugas menangkap citra kontainer menggunakan kamera di area *gate-in*. Modul Preprocessing Data berfungsi untuk melakukan penyesuaian pencahayaan, pengurangan *noise*, dan pelabelan data untuk memastikan kualitas dataset. Modul Deteksi dan Klasifikasi menjadi inti sistem, menggunakan YOLOv11 untuk mendeteksi area nomor kontainer dan ResNet-50 untuk mengklasifikasikan setiap karakter ke dalam 36 kelas (A-Z, 0-9). Terakhir, Modul Integrasi Sistem menghubungkan hasil deteksi ke antarmuka berbasis web yang dapat dipantau oleh operator secara *real-time* dan menyimpan hasilnya ke *database*.

B. Pengumpulan dan Preprocessing Data

Tahap pengumpulan data bertujuan untuk memperoleh dataset yang representatif. Data primer berupa citra nomor kontainer dikumpulkan langsung dari lokasi penelitian di BJTI menggunakan kamera CCTV, dengan fokus pada variasi pencahayaan, sudut pandang, dan kondisi fisik kontainer yang rusak. Untuk memperkaya variasi, data sekunder juga diperoleh dari sumber daring seperti Roboflow. Seperti pada gambar 6



GAMBAR 6 (FLOWCHART PENGUMPULAN DATA)

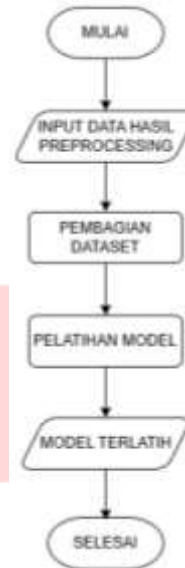
Setelah data terkumpul, dilakukan tahap *preprocessing* seperti yang digambarkan pada alur di atas. Proses ini meliputi pembersihan data untuk menghilangkan *noise*, *cropping* area nomor kontainer, dan normalisasi citra agar memiliki ukuran dan kontras yang konsisten. Langkah ini krusial untuk memastikan data dalam kondisi optimal sebelum digunakan untuk melatih model. Hal ini digambarkan pada gambar 7.



GAMBAR 7 (PELATIHAN MODEL)

C. Pengembangan dan Pengujian Model

Pengembangan model difokuskan pada arsitektur ResNet-50 karena kemampuannya yang unggul dalam mengenali pola visual kompleks. Seperti yang digambarkan pada gambar 6.



GAMBAR 8 (PENGEMBANGAN MODEL)

Proses pengembangan model dimulai dengan data hasil *preprocessing* yang telah dinormalisasi ke ukuran 224x224 piksel. Dataset kemudian dibagi menjadi data latih dan data uji. Proses pelatihan dilakukan menggunakan parameter yang telah ditentukan pada tabel 1.

TABEL 1 (PARAMETER PENGEMBANGAN RESNET-50)

No	Parameter	Nilai	Penjelasan
1.	Batch Size	32	Jumlah data yang diproses dalam satu kali iterasi pelatihan.
2.	Learning Rate	0.001	Kecepatan pembelajaran model, dengan pengurangan bertahap menggunakan <i>learning rate scheduler</i> .
3.	Optimizer	Adam	Algoritma optimasi yang digunakan untuk memperbarui bobot model selama pelatihan.
4.	Loss Function	Categorical Crossentropy	Fungsi kerugian yang digunakan untuk menghitung perbedaan antara prediksi dan label sebenarnya.

Tabel di atas merinci parameter kunci yang digunakan dalam pelatihan model, seperti *Batch Size* 32, *Learning Rate* 0.001, *Optimizer* Adam, dan *Loss Function* *Categorical Crossentropy*. Setelah pelatihan selesai, model yang telah terlatih disimpan dalam format file .h5 untuk digunakan pada tahap inferensi. Tahap pengujian dilakukan untuk mengevaluasi performa model ResNet-50 secara objektif menggunakan data uji yang terpisah.

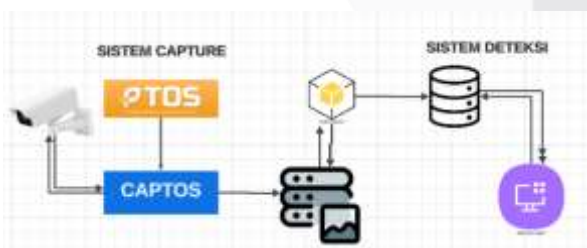
TABEL 2
(CONFUSION MATRIK UJI)

No	Pengujian	Penjelasan
1.	mAP (<i>mean Average Precision</i>)	Mengukur akurasi deteksi pada berbagai threshold IoU
2.	<i>Precision</i>	Mengukur seberapa tepat model dalam membuat prediksi, yaitu perbandingan antara prediksi benar dan semua prediksi positif.
3.	<i>Recall</i>	Mengukur kemampuan model dalam mendeteksi semua nomor kontainer yang benar, yaitu perbandingan antara prediksi benar dan semua label <i>ground truth</i> positif.
4.	<i>F1-Score</i>	Kombinasi <i>precision</i> dan <i>recall</i> yang memberikan gambaran menyeluruh tentang keseimbangan performa model.
5.	<i>Confusion Matrix</i>	Menampilkan detail jumlah prediksi benar dan salah dalam berbagai kategori, membantu mengidentifikasi pola kesalahan yang terjadi.

Kinerja model dievaluasi menggunakan beberapa metrik standar seperti yang tercantum pada tabel di atas, yaitu *mAP* (*mean Average Precision*), *Precision*, *Recall*, *F1-Score*, dan *Confusion Matrix*. Analisis hasil pengujian ini digunakan untuk mengidentifikasi kekuatan dan kelemahan model serta memastikan model telah mencapai standar yang diharapkan sebelum diimplementasikan.

D. Desain Arsitektur dan Integrasi Sistem

Arsitektur sistem secara keseluruhan dirancang untuk mengintegrasikan infrastruktur yang sudah ada di lapangan dengan sistem deteksi yang baru dikembangkan. Pada gambar 9.



GAMBAR 9
(DESAIN SISTEM)

Desain arsitektur pada gambar di atas memisahkan sistem menjadi dua bagian utama yaitu Sistem Capture dan Sistem Deteksi. Sistem Capture yang sudah ada di BJTI (terdiri dari kamera CCTV, P-TOS, dan CAP-TOS) bertugas menangkap dan menyimpan gambar kontainer ke server. Sementara itu, Sistem Deteksi yang dikembangkan dalam penelitian ini secara otomatis memantau server penyimpanan, memproses gambar baru menggunakan model YOLOv11 dan

ResNet-50, lalu menyimpan hasil identifikasi ke *database* untuk ditampilkan di *website*.

TABEL 3
(SPESIFIKASI PERANGKAT)

No.	Nama Perangkat	Spesifikasi
1.	Laptop Ideaped 3	AMD Ryzen 7 5700U With Radeon Graphics
2.	CCTV Dahua IPC-HFW5541E-ZE	5MP (2592 × 1944)@20 fps dan suport 4MP (2688 × 1520)@25/30 fps 12 VDC/PoE power supply IP67 dan IK10 protection
3.	PoE Injector PI-100H	PoE PSE Ports 1 Pass Through Data Rates 10/100/1000 Mbps Output Power Voltage 52 VDC Max User Port Power 30 Watts Max Input Power AC Input Voltage 100 ~ 240 VAC Requirements AC Input Current 0.8A @ 100-240 VAC AC Frequency 50 to 60 Hz
4.	Switch TP-Link TL-SG108	8-Port 10/100/1000Mbps

Tabel di atas merinci spesifikasi perangkat keras utama yang digunakan, termasuk laptop untuk pengembangan serta kamera CCTV Dahua, PoE Injector, dan Switch TP-Link yang digunakan dalam implementasi di lapangan. Sistem ini diintegrasikan dengan antarmuka web yang memungkinkan pengguna (operator dan staf IT) untuk memantau, mencari, dan mengelola data hasil identifikasi nomor kontainer secara efisien.



GAMBAR 10
(USECASE DIAGRAM)

Pada gambar 10 merupakan *Use case diagram* yang menggambarkan interaksi antara pengguna dan sistem. Staf

Operasional dapat melakukan login, mencari data kontainer, dan mencetak laporan, sementara Staf IT memiliki hak akses tambahan untuk mengelola data dan pengguna.

IV. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan tiga jenis dataset utama yang telah melalui proses anotasi dan validasi, di mana setiap dataset memiliki peran spesifik dalam alur kerja sistem identifikasi nomor kontainer. Seperti yang digambarkan pada gambar 11 sampai 13.



GAMBAR 11
(DATASET DETEKSI AREA
NOMOR KONTAINER
YOLOv11)



GAMBAR 12
(DATASET SEGMENTASI
YOLOv11)



GAMBAR 13
(DATASET KLASIFIKASI KARAKTER RESNET-50)

Gambar di atas secara berurutan menampilkan contoh dari ketiga dataset yang digunakan. Gambar 11 menunjukkan sampel dari dataset deteksi area, yang berisi citra kontainer utuh dalam berbagai kondisi lapangan. Gambar 12 adalah contoh dari dataset segmentasi, yang berisi potongan area nomor kontainer untuk melatih model agar dapat mendeteksi setiap karakter secara individual. Sementara itu, Gambar 13 menampilkan sampel dari dataset klasifikasi, yang terdiri dari ribuan gambar karakter tunggal (A-Z, 0-9) untuk melatih arsitektur ResNet-50.

1. Dataset Deteksi Nomor Kontainer

Dataset untuk deteksi area nomor kontainer dikumpulkan dari citra nyata di Pelabuhan BJTI dan sumber daring, berjumlah lebih dari 300 gambar. Dataset ini mencakup tantangan kondisi nyata seperti variasi pencahayaan siang-malam dan kondisi fisik kontainer (kotor, berkarat). Seperti pada gambar 14.



GAMBAR 14
(BOUNDING BOX AREA NOMOR KONTAINER)

2. Dataset Segmentasi Karakter

Dataset kedua untuk segmentasi karakter diperoleh dari pemotongan manual area nomor kontainer dan dari platform Roboflow, dengan total lebih dari 900 citra karakter individu. Setiap karakter pada dataset ini telah dianotasi dengan *bounding box* untuk pelatihan model YOLOv11 tahap kedua. Seperti pada gambar 15.



GAMBAR 15
(BOUNDING BOX SEGMENTASI NOMOR KONTAINER)

3. Dataset Klasifikasi Karakter

Dataset ini diambil dari Kaggle dan dikombinasikan dengan augmentasi data internal, menghasilkan total lebih dari 36.000 citra yang terbagi dalam 36 kelas (A-Z dan 0-9). Hal ini dapat dilihat pada gambar 16.



GAMBAR 16
(DATASET KLASIFIKASI KARAKTER)

Setiap dataset melalui tahap pengolahan data yang spesifik. Untuk deteksi area, citra diubah ukurannya menjadi 640x640 piksel dan dilakukan augmentasi data berupa rotasi ($\pm 15^\circ$) serta penyesuaian kecerahan ($\pm 25\%$). Dataset kemudian dibagi menjadi 70% data latih, 20% validasi, dan 10% uji. Dataset untuk klasifikasi karakter dibagi dengan proporsi 90% untuk data latih dan 10% untuk data uji.

B. Hasil Pelatihan Model

1. Hasil Pelatihan Deteksi Area Nomor Kontainer (YOLOv11)

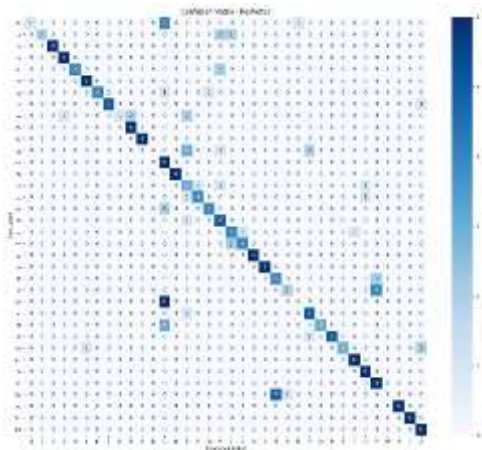
Tabel 4
Hasil metrik pelatihan model deteksi YOLOv11

Percobaan ke	Precision(B)	Recall(B)	mAP50(B)
1	86%	85%	88%
2	88%	84%	89%
3	92%	80%	88%
4	82%	54%	68%

Tabel di atas menampilkan hasil evaluasi dari empat percobaan pelatihan model deteksi. Berdasarkan metrik yang ada, percobaan kedua dipilih sebagai model terbaik karena menunjukkan kinerja yang paling seimbang. Model ini mencapai *precision* sebesar 88%, *recall* 84%, dan mAP50 sebesar 89%. Meskipun percobaan ketiga memiliki *precision* lebih tinggi (92%), nilai *recall*-nya lebih rendah (80%), sehingga secara keseluruhan tidak lebih seimbang dibandingkan percobaan kedua.

2. Hasil Pelatihan Klasifikasi Karakter (ResNet-50)

Tiga percobaan pelatihan dilakukan untuk model klasifikasi ResNet-50 dengan hasil yang bervariasi. Seperti yang terlihat pada gambar 17 dan 18 merupakan percobaan pertama.



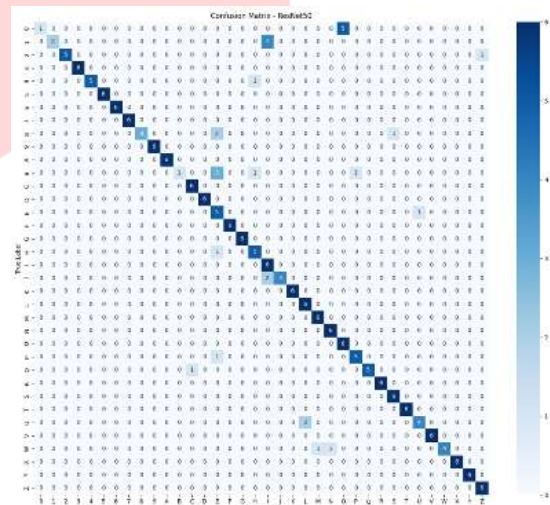
GAMBAR 17
(CONFUSION MATRIX PERCOBAAN 1)

	precision	recall	f1-score	support
0	1.00	0.17	0.33	6
1	1.00	0.23	0.58	6
2	1.00	1.00	1.00	6
3	0.90	1.00	0.97	6
4	1.00	0.67	0.80	6
5	0.90	1.00	0.97	6
6	1.00	0.67	0.80	6
7	1.00	0.21	0.42	6
8	1.00	0.33	0.50	6
9	0.75	1.00	0.86	6
A	1.00	1.00	1.00	6
B	0.90	0.90	0.90	6
C	0.57	1.00	0.69	6
D	1.00	1.00	1.00	6
E	0.33	0.58	0.45	6
F	0.90	0.67	0.78	6
G	0.90	0.67	0.78	6
H	0.40	0.83	0.58	6
I	0.70	0.67	0.68	6
J	0.90	0.67	0.78	6
K	1.00	1.00	1.00	6
L	1.00	1.00	1.00	6
M	0.44	0.67	0.55	6
N	0.67	0.33	0.48	6
O	0.90	0.90	0.90	6
P	0.63	0.67	0.65	6
Q	1.00	0.58	0.67	6
R	1.00	0.67	0.83	6
S	1.00	0.33	0.67	6
T	0.90	1.00	0.97	6
U	0.75	1.00	0.86	6
V	0.90	1.00	0.97	6
W	0.90	0.90	0.90	6
X	1.00	1.00	1.00	6
Y	1.00	1.00	1.00	6
Z	0.67	1.00	0.80	6
accuracy			0.71	216
macro avg	0.76	0.71	0.68	216
weighted avg	0.75	0.71	0.68	216

GAMBAR 18
(CLASSIFICATION REPORT PERCOBAAN 1)

Pada percobaan pertama, model mencapai akurasi keseluruhan sebesar 71%. Namun, analisis lebih dalam menunjukkan kelemahan signifikan, di mana beberapa karakter seperti 'B', 'O', dan 'W' sama sekali tidak berhasil dikenali (*recall* = 0). Hal ini menandakan model gagal melakukan generalisasi pada karakter dengan bentuk visual yang mirip.

Kemudian pada Percobaan kedua menunjukkan peningkatan performa yang sangat signifikan, dengan akurasi mencapai 87%. Model ini menunjukkan stabilitas yang jauh lebih baik dan berhasil mengklasifikasikan banyak karakter dengan sempurna, seperti '3', '5', 'D', 'F', dan lainnya dengan F1-score 1.00. Meskipun masih terdapat sedikit kebingungan pada karakter yang bentuknya mirip (misalnya 'O' dengan 'Q'), performa secara keseluruhan sangat solid. Hal ini dapat terlihat pada gambar 19 dan 20.



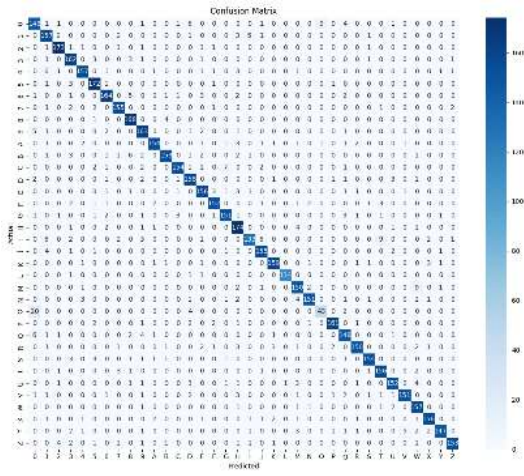
GAMBAR 19
(CONFUSION MATRIX PERCOBAAN 2)

	precision	recall	f1-score	support
0	1.00	0.17	0.33	6
1	1.00	0.23	0.51	6
2	1.00	1.00	1.00	6
3	1.00	0.83	0.91	6
4	1.00	0.83	0.91	6
5	1.00	1.00	1.00	6
6	1.00	1.00	1.00	6
7	1.00	1.00	1.00	6
8	1.00	0.58	0.87	6
9	1.00	1.00	1.00	6
A	1.00	1.00	1.00	6
B	1.00	1.00	1.00	6
C	1.00	1.00	1.00	6
D	1.00	1.00	1.00	6
E	0.43	0.63	0.54	6
F	1.00	1.00	1.00	6
G	1.00	1.00	1.00	6
H	0.71	0.67	0.69	6
I	0.70	1.00	0.87	6
J	1.00	1.00	1.00	6
K	1.00	1.00	1.00	6
L	0.73	1.00	0.86	6
M	0.44	1.00	0.62	6
N	0.60	1.00	0.72	6
O	0.53	1.00	0.71	6
P	0.63	0.67	0.65	6
Q	1.00	0.83	0.91	6
R	1.00	1.00	1.00	6
S	1.00	1.00	1.00	6
T	0.90	1.00	0.97	6
U	0.75	1.00	0.86	6
V	0.90	1.00	0.97	6
W	1.00	1.00	1.00	6
X	1.00	1.00	1.00	6
Y	1.00	1.00	1.00	6
Z	0.67	1.00	0.87	6
accuracy			0.87	216
macro avg	0.91	0.83	0.86	216
weighted avg	0.93	0.87	0.86	216

GAMBAR 20
(CLASSIFICATION REPORT PERCOBAAN 2)

Lalu pada percobaan ketiga meskipun akurasi tercatat lebih tinggi (91%), hasil ini tidak mencerminkan

kualitas klasifikasi yang lebih baik. *Confusion matrix* menunjukkan bahwa model mengalami tingkat kesalahan prediksi yang tinggi pada karakter-karakter sulit seperti 'O' dan 'M', yang sering salah diklasifikasikan sebagai karakter lain dengan bentuk serupa. Seperti pada gambar 21 dan 22.







GAMBAR 21 (CONFUSION MATRIX PERCOBAAN 3)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	blank		
0	177.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
1	0.0	180.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	179.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	180.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	178.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	178.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
D	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
E	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
G	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
H	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	178.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
J	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
K	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
L	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
M	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
N	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
O	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
Q	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0		
R	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0	0.0		
S	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0	0.0		
T	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0	0.0		
U	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0		
V	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	174.0	0.0	0.0	0.0		
W	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	176.0	0.0	0.0	0.0		
X	0.0																																						

2. Performa Segmentasi Karakter (YOLOv11)

Setelah area nomor kontainer terdeteksi, tahap selanjutnya adalah melakukan segmentasi untuk mengisolasi setiap karakter. Model YOLOv11 yang berbeda digunakan untuk tugas ini. Seperti pada tabel 6.

TABEL 6
(HASIL PENGUJIAN SEGMENTASI KARAKTER PLAT NOMOR)

Citra	Aktual	Prediksi	Keterangan
	DPEU8025417	DPEU8025417	Hasil deteksi karakter dari sampel 1
	SPNU292506	SPNL292506	Terdapat 1 karakter yang kurang tepat Ketika melalui deteksi model dari sample 2
	SPNU3028172	SPNU3028172	Hasil deteksi sempurna meskipun dalam keadaan sedikit gelap dari sampel 3
	RPTU2203311	RPTU2203311	Hasil deteksi sempurna dari sampel 4
	SSYU2210448	SSYU2210448	Hasil deteksi sempurna dari sampel 5

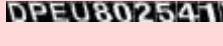
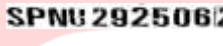
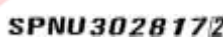
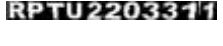
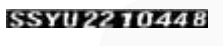
Hasil pengujian pada tabel di atas menunjukkan bahwa model segmentasi karakter memiliki performa yang sangat baik dan stabil. Pada empat dari lima sampel yang diuji,

termasuk pada kondisi pencahayaan yang sedikit gelap, model berhasil mengenali dan memisahkan seluruh karakter dengan sempurna. Hanya pada satu sampel terdapat satu karakter yang terdeteksi kurang tepat, kemungkinan disebabkan oleh kemiringan posisi tulisan pada kontainer. Secara keseluruhan, tahap segmentasi ini berjalan dengan akurasi tinggi.

3. Performa Klasifikasi Karakter (ResNet-50) dan Hasil Akhir

Tahap terakhir adalah mengklasifikasikan setiap karakter yang telah disegmentasi menggunakan model ResNet-50. Seperti yang ditunjukkan pada tabel 7.

TABEL 7
(HASIL PENGUJIAN PENGENALAN KARAKTER NOMOR KONTAINER)

Citra Karakter	Aktual	Prediksi	Keterangan
	DPEU8025417	DPEU8025417	Sampel 1
	SPNU2925062	SPNL2925062	Sampel 2
	SPNU3028172	SPNU3028172	Sampel 3
	RPTU2203311	RPTU2203311	Sampel 4
	SSYU2210448	SSYU2210448	Sampel 5

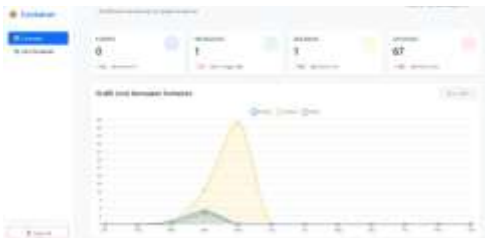
Tabel di atas menampilkan hasil akhir dari keseluruhan proses identifikasi. Model klasifikasi ResNet-50 menunjukkan tingkat akurasi yang sangat tinggi. Dari lima sampel yang diuji, empat di antaranya berhasil dikenali secara sempurna tanpa kesalahan, di mana hasil prediksi sepenuhnya sesuai dengan nomor kontainer aktual. Kesalahan hanya terjadi pada satu sampel, di mana terdapat satu kesalahan minor yaitu huruf 'U' yang terklasifikasi sebagai 'L'. Kesalahan ini kemungkinan besar disebabkan oleh kemiripan bentuk visual antara kedua karakter tersebut. Meskipun demikian, performa sistem secara keseluruhan dalam mengidentifikasi nomor kontainer menunjukkan akurasi dan konsistensi yang sangat baik untuk implementasi di lingkungan operasional.

D. Performa Website Monitoring

Pengujian *black box* pada website monitoring menunjukkan bahwa semua fungsionalitas, mulai dari login, tampilan *dashboard* rekap data kerusakan, hingga fitur pencarian kontainer, berjalan dengan stabil dan telah memenuhi kebutuhan operasional dasar. Untuk tampilan dari websitenya dapat dilihat pada gambar 23 sampai 25.



GAMBAR 23
(HALAMAN LOGIN)



GAMBAR 24
(TAMPILAN WEBSITE DASHBOARD)



GAMBAR 25
(TAMPILAN WEBSITE HALAMAN PENCARIAN KONTAINER)

V. KESIMPULAN

Penelitian ini telah berhasil mengembangkan sebuah sistem identifikasi nomor kontainer otomatis yang terintegrasi dengan sistem pendeteksi kerusakan bodi kontainer. Inovasi utama dari sistem ini adalah kemampuannya untuk bekerja secara selektif, di mana proses pencatatan nomor hanya akan terpicu jika sistem deteksi kerusakan mengidentifikasi adanya kerusakan pada bodi kontainer. Alur kerja yang efisien ini secara efektif menggantikan metode pencatatan manual yang rentan terhadap kesalahan, sehingga mampu mempercepat proses pelaporan dan meminimalkan intervensi manusia di lapangan.

Untuk mencapai hal tersebut, sistem ini mengimplementasikan dua teknologi *deep learning* utama: YOLOv11 yang bertugas untuk mendeteksi area karakter pada citra kontainer dan arsitektur ResNet-50 untuk mengklasifikasikan karakter dengan tingkat akurasi yang tinggi. Meskipun masih terdapat tantangan dalam membedakan karakter yang memiliki bentuk visual serupa, sistem secara keseluruhan menunjukkan performa yang stabil dalam berbagai kondisi pencahayaan dan posisi. Seluruh sistem terintegrasi dalam sebuah platform berbasis web yang intuitif, memungkinkan staf operasional untuk melakukan monitoring, validasi, dan mengakses data secara *real-time*. Dengan demikian, implementasi sistem ini memberikan manfaat operasional yang signifikan bagi pelabuhan, termasuk mengurangi risiko kesalahan pencatatan,

mempercepat alur klaim kerusakan, dan pada akhirnya meningkatkan kepercayaan pelanggan.

REFERENSI

- [1] D.-V. Licu, "Automatic container code identification using Machine Learning," 2020.
- [2] J. Liang and W. Jiang, "A ResNet50-DPA model for tomato leaf disease identification," *Front Plant Sci*, vol. 14, 2023, doi: 10.3389/fpls.2023.1258658.
- [3] G. S. Surjo, B. K. Ghosh, M. J. Alam, Mahamudullah, M. Razib, and S. Bilgaiyan, "A comparative analysis between single & dual-handed Bangladeshi Sign Language detection using CNN based approach," in *2023 International Conference on Computer Communication and Informatics, ICCCI 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICCCI56745.2023.10128583.
- [4] N. Basuni and Amril Mutoi Siregar, "Comparison of the Accuracy of Drug User Classification Models Using Machine Learning Methods," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1348–1353, Dec. 2023, doi: 10.29207/resti.v7i6.5401.
- [5] S. Bandong, Y. Y. Nazaruddin, and E. Joelianto, "Container Detection System Using CNN Based Object Detectors," in *Proceedings of the 2021 International Conference on Instrumentation, Control, and Automation, ICA 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 106–111. doi: 10.1109/ICA52848.2021.9625663
- [6] R. Shinde, A. Patil, and N. Jadhav, "Automatic number plate recognition using YOLOv11 and PaddleOCR," *International Journal of Computer Trends (IJCT)*, vol. 12, no. 3, pp. 5–10, 2025. [Online]. Available: <https://ijctjournal.org/wp-content/uploads/2025/06/IJCT-V12I3P2.pdf>
- [7] Z. Lin, C. Dong, and X. Wan, "Research on intelligent recognition algorithm of container numbers in ports based on deep learning," in *Proc. ICIC 2024, Lecture Notes in Computer Science*, vol. 14868, pp. 179–190, 2024. [Online]. Available: https://doi.org/10.1007/978-981-97-5600-1_16
- [8] Y. Zhao, J. Wu, X. Wang, and M. Li, "An improved character recognition framework for containers based on DETR-ResNeSt50-MSLC," *Sensors*, vol. 21, no. 14, p. 4678, 2021. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8272209/pdf/sensors-21-04678.pdf>
- [9] Y. Akkem, "Multi Disease Prediction Model by using Machine Learning and Flask API," *2020 International Conference on Communication and Electronics Systems (ICCES)*, pp. 962–965, 2020. DOI: 10.1109/ICCES48766.2020.9137896
- [10] U. M. Kotha, D. Bharathi, and P. K. D. Prakash, "Execution of Flask deployment on essay grading," *International Journal of Electrical and Computer*

- Engineering (IJECE)*, vol. 13, no. 3, pp. 2990–2998, Jun. 2023. DOI: 10.11591/ijece.v13i3.pp2990-2998
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90
- [12] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *arXiv preprint*, arXiv:2410.17725, Oct. 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2410.17725>
- [13] S. Balakrishna, S. Yadao, and V. K. Solanki, "Enhancing YOLOv11 for Real-Time Object Detection: Advanced Architectures and Edge-Optimized Training Pipeline," in *Proc. 9th Int. Conf. Res. Intelligent Computing in Engineering (ACSIS)*, vol. 42, pp. 89–96, May 2025, doi: 10.15439/2024R115.
- [14] A. A. Salihu, M. H. Ahmad, N. Z. Jhanjhi, M. S. Khan, and A. Abdullah, "A Comprehensive Review on Evaluation Metrics for Deep Learning Classification Models Based on Confusion Matrix," *IEEE Access*, vol. 9, pp. 145362–145377, 2021, doi: 10.1109/ACCESS.2021.3122606.
- [15] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018. [Online]. Available: <https://arxiv.org/abs/1804.02767>.
- [16] T. S. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.