PENERAPAN MODEL *YOU ONLY LOOK ONCE* (YOLO) DAN *TRANSFORMERS BASED ON OPTICAL CHARACTER RECOGNITION* (TROCR) UNTUK PENDETEKSIAN DAN PEMBACAAN PELAT NOMOR KENDARAAN

(SKRIPSI)

Oleh
ZIYAD MUHAMMAD ADZIN AZZUFARI

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024

ABSTRACT

IMPLEMENTATION OF THE YOU ONLY LOOK ONCE (YOLO) MODEL AND TRANSFORMERS-BASED OPTICAL CHARACTER RECOGNITION (TROCR) FOR VEHICLE LICENSE PLATE DETECTION AND RECOGNITION.

BY

ZIYAD MUHAMMAD ADZIN AZZUFARI

License plate recognition plays a crucial role in various applications of Intelligent Transportation Systems (ITS), such as traffic management, vehicle tracking, and parking security systems. While general approaches focus on real-time detection and recognition of license plates for traffic monitoring, this research is limited to processing images captured through cameras. The main challenges in developing such systems include variations in plate size, shape, lighting, and environmental conditions, which impact the accuracy of automatic detection and character recognition. This study aims to develop a system that integrates You Only Look Once version 8 (YOLOv8) for license plate detection and Transformer-based Optical Character Recognition (TrOCR) for character recognition. Using YOLOv8, the system achieved high evaluation results with a mean Average Precision (mAP50) of 99.5%, mAP50-95 of 83.78%, and a recall of 100% on the validation data. For character recognition, TrOCR yielded the best Character Error Rate (CER) of 0.011 on validation data and 1.12% on test data. The findings show that the combined approach of YOLOv8 and TrOCR offers an efficient and accurate system for detecting and recognizing license plates, with potential for real-time traffic monitoring applications.

Keywords: License plate recognition, YOLOv8, TrOCR, Intelligent Transportation Systems, Object detection, Character recognition, Real-time traffic monitoring, Computer vision.

ABSTRAK

PENERAPAN MODEL *YOU ONLY LOOK ONCE* (YOLO) DAN *TRANSFORMERS BASED ON OPTICAL CHARACTER RECOGNITION* (TROCR) UNTUK PENDETEKSIAN DAN PEMBACAAN PELAT NOMOR KENDARAAN

OLEH

ZIYAD MUHAMMAD ADZIN AZZUFARI

Pengenalan pelat nomor kendaraan memainkan peran penting dalam berbagai aplikasi sistem transportasi cerdas, seperti manajemen lalu lintas dan sistem keamanan parkir. Secara umum, pendekatan yang digunakan berfokus pada pendeteksian dan pengenalan pelat nomor secara real-time untuk pemantauan lalu lintas. Namun, penelitian ini hanya berfokus pada pengolahan gambar yang diambil melalui kamera. Tantangan utama dalam pengembangan sistem ini mencakup variasi ukuran, bentuk, pencahayaan, serta kondisi lingkungan yang mempengaruhi akurasi pendeteksian dan pengenalan karakter secara otomatis. Penelitian ini bertujuan untuk membangun sistem yang mengintegrasikan *You Only Look Once version 8* (*YOLOv8*) untuk pendeteksian pelat nomor dan *Transformer-based Optical Character Recognition* (*TrOCR*) untuk pengenalan karakter. Dengan menggunakan YOLOv8, sistem mampu mencapai hasil evaluasi yang sangat baik dengan nilai *mean Average Precision* (*mAP50*) sebesar 99,5%, mAP50-95 sebesar 83,78%, dan *recall* sebesar 100% pada data validasi. Penggunaan TrOCR untuk pengenalan karakter menghasilkan nilai *Character Error Rate* (CER) terbaik sebesar 0,011 pada data validasi dan 1,12% pada data pengujian. Hasil penelitian menunjukkan bahwa pendekatan berbasis YOLOv8 dan TrOCR mampu menghadirkan sistem yang efisien dan akurat dalam mendeteksi dan mengenali pelat nomor kendaraan, dengan potensi untuk diaplikasikan pada sistem pemantauan lalu lintas secara real-time.

Kata Kunci: Pengenalan pelat nomor, YOLOv8, TrOCR, Sistem Transportasi Cerdas, Deteksi objek, Pengenalan karakter, Pemantauan lalu lintas real-time, Visi komputer

PENERAPAN MODEL *YOU ONLY LOOK ONCE* (YOLO) DAN *TRANSFORMERS BASED ON OPTICAL CHARACTER RECOGNITION* (TROCR) UNTUK PENDETEKSIAN DAN PEMBACAAN PELAT NOMOR KENDARAAN

Oleh
ZIYAD MUHAMMAD ADZIN AZZUFARI
2017031052

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024

Judul Skripsi	:	**PENERAPAN MODEL *YOU ONLY LOOK ONCE* (YOLO) DAN *TRANSFORMERS BASED ON OPTICAL CHARACTER RECOGNITION* (TROCR) UNTUK PENDETEKSIAN DAN PEMBACAAN PELAT NOMOR KENDARAAN**
Nama Mahasiswa	:	**ZIYAD MUHAMMAD ADZIN AZZUFARI**
Nomor Pokok Mahasiswa	:	**2017031052**
Jurusan	:	**Matematika**
Fakultas	:	**Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Dr. Dian Kurniasari, S.Si., M.Sc.
NIP. 196903051996032001

Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc.
NIP. 197101291997021001

2. Ketua Jurusan

Dr. Aang Nuryaman, S.Si., M.Si
NIP. 197403162005011001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. Dian Kurniasari, S.Si., M.Sc.** .....

Sekertaris : **Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc.** .....

Penguji
Bukan Pembimbing : **Dr. Ahmad Faisol, S.Si., M.Sc.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 10 September 2024

PERNYATAAN SKRIPSI MAHASISWA

Saya yang bertanda tangan dibawah ini:

Nama	:	**Ziyad Muhammad Adzin Azzufari**
Nomor Pokok Mahasiswa	:	**2017031052**
Jurusan	:	**Matematika**
Judul Skripsi	:	**PENERAPAN MODEL *YOU ONLY LOOK ONCE* (YOLO) DAN *TRANSFORMERS BASED ON OPTICAL CHARACTER RECOGNITION* (TROCR) UNTUK PENDETEKSIAN DAN PEMBACAAN PELAT NOMOR KENDARAAN**

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri dan semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah karya penulisan ilmiah Universitas Lampung.

Bandar Lampung, 10 September 2024

Penulis

Ziyad Muhammad Adzin Azzufari

NPM. 2017031052

RIWAYAT HIDUP

Penulis memiliki nama lengkap Ziyad Muhammad Adzin Azzufari yang lahir di Cilegon pada tanggal 5 Februari 2001. Pernulis merupakan anak pertama dari pasangan Bapak Dadan Sudani dan Ibu Muyasaroh.

Penulis menyelesaikan pendidikan di TK Tarbiyatul Aulad pada tahun 2005 sampai dengan tahun 2007. Selanjutnya, penulis melanjutkan pendidikan Sekolah Dasar di SD Persis Serang pada tahun 2007 – 2016. Kemudian menempuh pendidikan di Madrasah Aliyah Negeri 1 Kota Serang pada tahun 2016 sampai 2019.

Pada tahun 2019, penulis melanjutkan pendidikan di perguruan tinggi Politeknik Negeri Bandung D3 Teknik Mesin. Namun penulis tidak melanjutkan studinya di tahun 2020. Selanjutnya pada tahun yang sama, penulis mendaftar SBMPTN dan diterima pada program Sarjana jurusan matematika Universitas Lampung. Selama menjadi mahasiswa, beberapa kegiatan yang diikuti adalah sebagai berikut:

1. Mengikuti lomba Satria Data bidang *Big Data Challenge* pada tahun 2022.
2. Pada tahun 2023 menjalankan kerja praktik di perusahaan BUMN yang bergerak dalam bidang produksi baja. Penulis diterima di Divisi *Production Planning and Control*. Laporan selama kerja praktik untuk perusahaan berjudul Penerapan Metode *Autoregressive Integrated Moving Average* (ARIMA) Pada Peramalan Produksi Produk *Heavy Soft, Lite Soft, Medium Soft, Heavy Hard, Lite Hard,* dan *Medium Hard* di Pabrik *Cold Roll Mill* PT Krakatau Steel (Persero) Tbk Tahun 2023.
3. Pada tahun 2023 menjadi asisten dosen praktikum mata kuliah Pengantar *Data Mining*.
4. Mengikuti kegiatan Kerja Kuliah Nyata (KKN) di Desa Teratas, Kecamatan Kota Agung Pusat, Kabupaten Tanggamus pada tahun 2023.

5. Mengikuti lomba Satria Data bidang *Big Data Challenge* pada tahun 2023.
6. Ikut serta dalam program Kampus Merdeka dengan kegiatan studi independen di Bangkit Academy led by Google, Tokopedia, Gojek, & Traveloka dengan *learning path machine learning.*
7. Menjadi asisten praktikum pada *workshop text mining* dengan *audiens* dari mahasiswa tamu Universiti Teknologi Mara Malaysia.

Selain kegiatan dalam kampus, penulis juga memiliki kegiatan diluar kampus seperti:

1. Pemilik toko online Jajanan Digital. Jajanan Digital adalah toko yang menjual kebutuhan produk digital seperti pulsa, pembayaran listrik, voucher games dan lain – lain. Jajanan Digital terdapat pada e-commerce seperti Bukalapak, Itemku, Blibli, dan VcGamers.
2. Penulis memiliki sertifikat Pelaksanaan Analisis Efek atau *Regular Securities Analyst* (RSA). Sertifikasi Kompetensi Pelaksanaan Analis Efek merupakan kompetensi yang ditujukkan kepada analis junior, yang ditugaskan untuk menganalisa satu atau beberapa instrumen keuangan yang melakukan fungsi Analisis Efek baik secara Fundamental maupun Teknikal, baik Saham, Obligasi, maupun efek lainnya.

KATA INSPIRASI

"Dan (ingatlah juga), tatkala Tuhanmu memaklumkan; "Sesungguhnya jika kamu bersyukur, pasti Kami akan menambah (nikmat) kepadamu, dan jika kamu mengingkari (nikmat-Ku), maka sesungguhnya azab-Ku sangat pedih"
(Q.s Ibrahim:7)

"Barang siapa yang tidak mensyukuri yang sedikit, maka ia tidak akan mampu mensyukuri sesuatu yang banyak."
(HR. Ahmad, 4/278. Syaikh Al Albani mengatakan bahwa hadits ini hasan sebagaimana dalam As Silsilah Ash Shohihah no. 667)

"*An investment in knowledge pays the best interest*."
(Benjamin Franklin)

"Sucikanlah nama Tuhanmu Yang Mahatinggi, yang menciptakan (semua mahluk) dan menyempurnakannya, yang memberi takdir kemudian mengarahkan(nya)"
(Q.S Al-A'la:1-3)

"Boleh jadi kamu membenci sesuatu, padahal ia amat baik bagimu, dan boleh jadi (pula) kamu menyukai sesuatu, padahal ia amat buruk bagimu; Allah mengetahui, sedang kamu tidak mengetahui."
(Q.S Al-Baqarah:216)

PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT yang telah memberikan kekuatan, kesabaran, dan kemudahan sehingga dapat menyelesaikan skripsi ini dengan baik. Saya persembahkan rasa terima kasih saya kepada:

Umi dan Abi

Terima kasih yang tak terhingga untuk Umi dan Abi atas segala doa, cinta, dan dukungan yang tak pernah putus sejak awal hingga saya bisa menyelesaikan skripsi ini. Doa-doa tulus kalian yang bisa mampu melewati segala tantangan dan ujin ini. Semoga Allah SWT membalas kebaikan dan pengorbanan kalian dengan dengan berlipat ganda dan menempatkan Umi dan Abi di surga-Nya.

Dosen Pembimbing dan Pembahas

Saya mengucapkan terima kasih yang sebesar-besarnya kepada dosen pembimbing dan dosen pembahas atas bimbingan, arahan, serta dukungan selama proses penyusunan skripsi ini. Berkat kesabaran dan masukan yang berharga, saya dapat menyelesaikan penelitian ini dengan baik. Semoga ilmu dan arahan yang diberikan menjadi bekal berharga bagi masa depan saya. Terima kasih.

Seluruh keluarga

Sahabat-sahabat

Almamater tercinta, Universitas Lampung

SANWACANA

Puji syukur kehadirat Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis dapat meyelesaikan skripsi ini yang berjudul "Penerapan Model *You Only Look Once* (Yolo) Dan *Transformers Based On Optical Character Recognition* (Trocr) Untuk Pendeteksian Dan Pembacaan Pelat Nomor Kendaraan". Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan Nabi besar Muhammad SAW.

Penulis menyadari bahwa dalam proses penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang tulus kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc., selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan, dan masukan yang berharga dalam proses penyusunan skripsi ini.
2. Bapak Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc., selaku Dosen Pembimbing II yang telah memberikan arahan, bimbingan dan dukungan sehingga dapat menyelesaikan skripsi ini.
3. Bapak Dr. Ahmad Faisol, S.Si., M.Sc., selaku Dosen Penguji yang telah memberikan kritik, saran, serta evaluasi dalam penulisan skripsi ini.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Ibu Widiarti , S.Si., M.Sc., selaku dosen pembimbing akademik.
6. Seluruh dosen, staf dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Lampung.

7. Umi, Abi, dan adik-adik yang selalu memberikan motivasi, dukungan dan do'a kepada penulis.
8. Terima kasih kepada diri saya sendri karena sudah berjuang dan bisa melalui tahapan ini.
9. Teman-teman bimbingan, teman seperjuangan selama kuliah, teman SMA dan seluruh pihak yang tidak bisa saya sebutkan satu persatu telah membantu penulis dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, oleh karena itu kritik dan saran yang membangun sangat diharapkan demi perbaikan karya ini di masa yang akan datang. Semoga skripsi ini dapat memberikan manfaat dan kontribusi positif, baik bagi penulis sendiri maupun bagi pembaca.

Bandar Lampung, 10 September 2024
Penulis

Ziyad Muhammad Adzin Azzufari
NPM. 2017031052

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR KODE PROGRAM

Kode Program Halaman

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

Optical character recognition atau yang disebut dengan OCR, adalah sistem konversi elektronik dari teks yang diketik atau ditulis tangan kemudian didokumentasikan dan diubah menjadi teks yang dikodekan oleh mesin, baik dari dokumen yang dipindai, foto dokumen, foto adegan, atau dari teks *subtitle* yang ditambahkan pada gambar (Minghao Li dkk., 2021). Beberapa contoh metode pengenalan karakter adalah Tesseract, EasyOCR, dan *Transformers based on OCR*.

Transformers based on OCR (TrOCR) pertama kali dikenalkan oleh Li dkk. (2021). TrOCR terdiri dari *encoder* Transformer gambar dan *decoder* Transformer teks autoregresif untuk melakukan OCR pada tahun 2021, TrOCR adalah model OCR berbasis transformator *end-to-end* untuk pengenalan teks dengan model *pre-trained computer vision* (CV) dan n*atural language processing* (NLP).

Penggunaan TrOCR dalam mendeteksi objek melibatkan dua tahap: pertama, pendeteksian kotak pembatas (*bounding box*) yang mengelilingi objek pada gambar dan kedua pengenalan karakter yang terdapat dalam objek tersebut. Metode yang digunakan dalam mendeteksi sebuah objek beberapa diantaranya adalah *Region-based Convolutional Neural Network* (R-CNN), *Faster R-CNN*, *Fast R-CNN*, dan YOLO. Penelitian kali ini menggunakan metode YOLOv8.

YOLOv8 dirilis pada Januari 2023 oleh Ultralytics. YOLOv8 menyediakan lima versi: YOLOv8n (*nano*), YOLOv8s (*small*), YOLOv8m (*medium*), YOLOv8l (*large*) dan YOLOv8x (*extra large*). YOLOv8 mendukung berbagai tugas visi

seperti deteksi objek, segmentasi, estimasi pose, pelacakan, dan klasifikasi (Terven dan Cordova-Esparza., 2023).

Tanda nomor kendaraan bermotor (TNKB) atau sering disebut juga pelat nomor merupakan pelat yang terbuat dari bahan alumunium sebagai identifikasi resmi suatu kendaraan yang dikeluarkan oleh Korlantas Polri. Penggunaan pelat nomor diatur dalam undang-undang nomor 22 tahun 2009 tentang Lalu Lintas dan Angkutan. TNKB membuat Nomor Registrasi Kendaraan Bermotor (NRKB) yang umumnya terdiri dari satu sampai dua huruf diawal yang mewakili kode wilayah, diikuti dengan satu hingga empat angka ditengah sebagai nomor urut registrasi, dan diakhiri satu hingga tiga huruf berarti sub-wilayah dari provinsi yang terdaftar. Urutan registrasi yang berupa angka dan terdiri dari satu sampai empat angka bukan hanya angka acak, namun juga secara umum memiliki arti. Ada empat kategori yang sesuai dengan jenis kendaraannya. Nomor 1 sampai 2999 merupakan nomor urut registrasi untuk mobil jenis penumpang, nomor 3000 sampai 6999 untuk jenis sepeda motor, nomor 7000 sampai 7999 untuk jenis bus, nomor 8000 sampai 8999 untuk jenis mobil barang, nomor 9000 sampai 9999 untuk jenis kendaraan khusus.

Pengenalan pelat nomor kendaraan memiliki peran penting dalam bidang sistem transportasi cerdas. Teknologi ini sering digunakan dalam berbagai aplikasi seperti manajemen lalu lintas, pelacakan kendaraan, sistem keamanan parkir dan sistem lainnya yang menggunakan pengenalan pelat nomor. Pendeteksian dan pembacaan karakter menjadi topik yang menarik karena meningkatnya jumlah kendaraan dan penyebaran kamera pemantau lalu lintas. Pemantauan kendaraan secara manual sangat sulit untuk dilakukan oleh manusia karena melelahkan, dan hal ini mengakibatkan terjadinya *human error*. Sistem otomatisasi dibutuhkan untuk mengidentifikasi pelat nomor kendaraan secara real-time dan efisien. Namun beberapa tantangan menghambat kinerja dari deteksi pelat nomor dan pengenalan karakter otomatis, seperti kecepatan dan pergerakan, variasi dari pelat nomor seperti ukuran, tulisan, kemiringan, berisi bingkai atau sekrup, pencahayaan dan kondisi lingkungan. Kemudian dari pendekatan umum yang menitik beratkan pada deteksi dan pengenalan pelat nomor secara *real-time* untuk monitoring lalu lintas,

terjadi pergeseran menuju pendekatan khusus yang membatasi analisis hanya pada gambar yang diambil melalui kamera.

Penelitian mengenai sistem deteksi pelat nomor kendaraan sudah pernah dilakukan dalam banyak penelitian diantaranya dilakukan oleh Lu dkk. (2019) menggunakan *deep convolutional neural network* (CNN). Model yang dibangun untuk mendeteksi pelat nomor mencapai 99.99% *mean average precision* (mAP) pada dataset OpenITS.

Berikutnya penelitian dengan judul *Automatic License Plate Recognition for Indian Roads Using Faster-RCNN* oleh Praveen dkk. (2019). Faster-RCNN sebagai basis CNN berupa Resnet-50 bekerja lebih baik daripada VGG16. Deteksi pelat nomor diuji dengan menggunakan 597 gambar dengan nilai mAP 94.98% dan untuk segmentasi karakter diuji dengan 113 pelat nilai mAP sebesar 99.55%. CNN digunakan untuk pengenalan karakter dengan akurasi sebesar 98.6%.

Kemudian penelitian mengenai pendataan pelat nomor kendaraan menggunakan metode YOLO dan Tesseract-OCR dilakukan oleh Tirtana dkk. (2021). Model YOLOv3 yang sudah dilatih memiliki mAP sebesar 100% pada nilai threshold 50%. Tingkat akurasi pembacaan karakter mencapai 71,46% dan membutuhkan waktu rata-rata selama 11,55 detik.

Selanjutnya metode pengenalan karakter bahasa Arab *end-to-end* menggunakan vision transformers sebagai encoder yaitu BEiT dan vanila transformers sebagai decoder dilakukan oleh Mostafa dkk. (2022) Dataset OCR bahasa Arab, yang terdiri dari 30,5 juta gambar berupa baris teks, 270 juta kata, dan 1,6 miliar karakter. Model tersebut mencapai *character error rate* (CER) sebesar 4.46%.

Terakhir metode menggunakan TrOCR pada gambar struk belanja yang dipindai satu halaman penuh dilakukan oleh Zhang dkk. (2023) Menggunakan dataset *Scanned Receipts OCR and Information Extraction* (SROIE) dari kompetisi *International Conference on Document Analysis and Recognition* (ICDAR) 2019. Model terbaik yang membagi penuh gambar menjadi 15 potongan berukuran sama menghasilkan F1 *score* sebesar 87.8% dan 4.98% CER.

1.2 Rumusan Masalah

Dari permasalahan latar belakang tersebut, perumusan masalah untuk penelitian yang akan dilakukan adalah:

1. Bagaimana cara membangun sistem pendeteksian pelat nomor menggunakan YOLOv8.
2. Bagaimana performa dari YOLOv8 untuk mendeteksi pelat nomor kendaraan.
3. Bagaimana cara membangun sistem pembacaan karakter dari pelat nomor menggunakan TrOCR.
4. Bagaimana performa dari TrOCR untuk pembacaan karakter dari pelat nomor.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Membuat sistem deteksi pelat nomor menggunakan YOLOv8.
2. Mengetahui pengujian pada sistem yang dibangun dengan YOLOv8 dari mAP dan Recall.
3. Membuat sistem mengenali karakter dari pelat nomor menggunakan TrOCR.
4. Menguji hasil pembacaan karakter menggunakan TrOCR dari metriks CER.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Dapat memberikan referensi baru untuk pembelajaran selanjutnya.
2. Menjadi referensi inovasi teknologi untuk sistem parkir, melacak kendaraan dalam kegiatan kriminal atau aktivitas mencurigakan.

3. TrOCR mencapai akurasi canggih dengan model encoder-decoder berbasis transformator standar, yang bebas konvolusi dan tidak bergantung pada langkah pra/pasca pemerosesan yang rumit.
4. Teknologi OCR menghilangkan kebutuhan entri data manual.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu yang telah dilakukan digunakan sebagai bahan acuan dalam penelitian ini. Berikut merupakan ringkasan dari penelitian terdahulu yang akan dijelaskan pada Tabel 1 berikut:

Tabel 1. Penelitian terdahulu

Penelitian	Data	Metode	Hasil
License plate detection and recognition using hierarchical feature layers from CNN (Lu dkk., 2019)	*Chinese license plate*. *Train*: 42.000 *Test*: 1403 (open ITS)	CNN	Deteksi :99.99% mAP Pengenalan karakter: 96.7% Akurasi
Automatic License Plate Recognition for Indian Roads Using Faster-RCNN (Ravirathinam dan Patawari., 2019)	*Indian license plate*s: *Train*: 2209 *Test*: 597 *Character Segmentation*: *Train*: 711 images *Test*: 113	Faster R-CNN base VGG16 dan Faster R-CNN base resnet-50	LPD: VGG16= 91,43% mAP Resnet-50=94.98% mAP CS: VGG16=95.63% mAP Resnet-50=99.55%

Penerapan Metode Yolo Dan Tesseract-Ocr Untuk Pendataan Plat Nomor Kendaraan Bermotor Umum Di Indonesia Menggunakan Raspberry Pi (Tirtana dkk., 2021).	Pelat nomor indonesia Train:20 Test:-	YOLOv3 dan tesseract	YOLOv3: 100% mAP Tesseract: 77.18% akurasi
An End-to-End OCR Framework for Robust Arabic-Handwriting Recognition using a Novel Transformers-based Model and an Innovative 270 Million-Words Multi-Font Corpus of Classical Arabic with Diacritics (Mostafa dkk., 2022).	30,5 juta gambar baris teks arabic, 270 juta kata dan 1,6 miliar karakter	Encoder (BEIT) - decoder (vanilla Transformers)	CER: 4.64%
Extending TrOCR for Text Localization-Free OCR of Full-Page Scanned Receipt Images (Zhang dkk., 2023)	626 gambar untur training dan 361 gambar testing (ICDAR 2019)	*Transformers Based on OCR* (TrOCR)	CER: 4.98% Presisi: 87.9% Recall: 87.7% F1: 87.8%

Berikut adalah hasil penelitian yang dijadikan bahan referensi. Penelitian berjudul "License plate detection and recognition using hierarchical feature layers from CNN" yang dilakukan oleh Lu dkk (2019) memfokuskan pada dua komponen utama: deteksi pelat nomor dan pengenalan karakter. Pendekatan ini melibatkan pelatihan *Convolutional Neural Network* (CNN) untuk mengidentifikasi pelat

nomor dalam bingkai video yang dimasukkan. Penelitian ini menggunakan dataset OpenITS yang dikembangkan oleh Universitas ZhongShan, yang terdiri dari 1,403 gambar dengan anotasi pelat nomor yang mencakup seluruh provinsi dan kota di daratan Tiongkok. Hasil pengujian pada dataset OpenITS menunjukkan nilai *mean Average Precision* (mAP) mendekati 100%, mengungguli sistem yang menggunakan metodologi *Faster* R-CNN dan HOG + SVM. Ketika mengevaluasi kinerja secara menyeluruh, terutama dalam skenario yang lebih kompleks yang melibatkan berbagai sudut kamera, resolusi gambar, dan variasi kondisi pencahayaan, dilakukan pengujian pada dataset terdiri dari 962 gambar. Performa model pada dataset pengujian yang lebih rumit juga melampaui performa metodologi Faster R-CNN dan HOG + SVM. Pengujian pengenalan karakter, hampir 42.000 pelat nomor mendapatkan nilai akurasi sebesar 96,78%.

Penelitian selanjutnya berjudul "Automatic License Plate Recognition for Indian Roads Using Faster-RCNN" yang dilakukan oleh Ravirathinam dkk (2019), mengungkap bahwa kebanyakan sistem *Automatic License Plate Recognition* (ALPR) terdiri dari tiga tugas kunci, yaitu *License Plate Detection* (LPD), *Character Segmentation* (CS) dan *Character Recognition* (CR). Penelitian ini menerapkan metode Faster-RCNN untuk LPD dan CS, sementara CNN digunakan untuk CR, dengan setiap bagian memiliki bobot yang terlatih secara khusus. Dataset yang digunakan untuk LPD terdiri dari 2806 gambar, dengan 2209 data digunakan untuk pelatihan dan 597 data untuk pengujian. Selanjutnya, dataset untuk CS mencakup 824 pelat, di mana 711 pelat digunakan untuk pelatihan dan 113 pelat digunakan untuk pengujian. Penggunaan jaringan ResNet-50 menghasilkan presisi sebesar 94,98% untuk LPD dan 99,55% untuk CS. Adapun untuk CR, presisi yang dicapai adalah 98,6%. Secara keseluruhan, akurasi dalam membaca nomor pelat kendaraan mencapai 91%.

Kemudian studi yang berjudul "Penerapan Metode Yolo Dan Tesseract-Ocr Untuk Pendataan Plat Nomor Kendaraan Bermotor Umum Di Indonesia Menggunakan Raspberry Pi" dilaksanakan oleh Tirtana dkk. (2021). Penelitian ini melibatkan penggunaan 20 gambar untuk melakukan deteksi pelat nomor kendaraan menggunakan model YOLO, yang diujikan pada nilai threshold IoU sebesar 50%,

60%, dan 70%. Pada nilai threshold 50%, nilai mAP mencapai 100%, sementara pada nilai threshold 60% mAP mencapai 94,5%. Nilai mAP menurun menjadi 80,31% saat menggunakan nilai threshold 70%. Sementara itu, dalam proses pengenalan karakter menggunakan Tesseract, akurasi yang diperoleh adalah sebesar 71,46%.

Berikutnya penelitian berjudul "An End-to-End OCR Framework for Robust Arabic-Handwriting Recognition using a Novel Transformers-based Model and an Innovative 270 Million-Words Multi-Font Corpus of Classical Arabic with Diacritics" oleh Mostafa dkk. (2022) mengeksplorasi fase kedua dalam serangkaian penelitian terkait pengembangan *Optical Character Recognition* (OCR) untuk dokumen sejarah berbahasa Arab. Penelitian ini mendalami interaksi antara prosedur pemodelan yang beragam dengan tantangan yang ada. Dataset yang digunakan dalam OCR untuk bahasa Arab mencakup 30,5 juta gambar berisi baris teks dan 270 juta kata yang tersusun dengan benar. Pendekatan pengenalan teks secara end-to-end memanfaatkan *Vision Transformers* sebagai encoder, yakni *BERT Pre-Training of Image Transformers* (BEIT), dan vanilla Transformer sebagai decoder. Pendekatan ini mengeliminasi penggunaan (*convolutional neural network*) CNN untuk ekstraksi fitur dan mengurangi kompleksitas model. Meskipun terdapat 30 juta gambar baris teks, hanya sekitar 100,000 gambar yang dapat digunakan dalam pelatihan model karena keterbatasan sumber daya. Selain itu, model OCR dilatih menggunakan 12 jenis font yang berbeda, termasuk diakritik, dengan variasi urutan panjang dan pendek, berhasil mencapai CER sebesar 4,46.

Terakhir penelitian yang berjudul "Extending TrOCR for Text Localization-Free OCR of Full-Page Scanned Receipt Images" dilakukan oleh Hongkuan Zhang dkk. (2022). Tujuan dari digitalisasi gambar struk yang dipindai adalah untuk mengekstrak teksnya dan menyimpan informasi tersebut dalam bentuk dokumen terstruktur. Proses ini umumnya terbagi menjadi dua tahapan: pelokalan teks dan OCR. Eksperimen yang dilakukan pada dataset OCR penerimaan SROIE dari kompetisi ICDAR 2019, terdapat 626 gambar dalam set pelatihan dan 361 gambar dalam set pengujian. Model yang telah diperbaiki dengan strategi yang diusulkan

mencapai skor F1 sebesar 64,4 dan CER sebesar 22,8%. Hasil ini mengungguli hasil dasar yang memiliki skor F1 sebesar 48,5 dan CER sebesar 50,6%. Model terbaik, yang membagi gambar penuh menjadi 15 bagian yang sama ukurannya, mencapai skor F1 sebesar 87,8 dan CER sebesar 4,98% dengan tambahan minimal sebelum atau sesudah pemrosesan keluaran.

2.2 Tanda Nomor Kendaraan Bermotor

Tanda nomor kendaraan bermotor (TNKB) atau yang biasa dikenal pelat nomor adalah identifikasi resmi berbahan aluminium yang dikeluarkan oleh kantor bersama Sistem Adminstrasi Manunggal Satu Atap (SAMSAT). Pelat nomor digunakan sejak masa pemerintahan Hindia Belanda, di mana penggunaannya melibatkan kode wilayah berdasarkan pembagian administratif keresidenan.

Undang-undang nomor 22 tahun 2009 pasal 68 menyatakan bahwa Setiap kendaraan bermotor yang dioperasikan dijalan wajib dilengkapi dengan surat tanda nomor kendaraan bermotor dan tanda nomor kendaraan bermotor. Surat tanda nomor kendaraan bermotor memuat data kendaraan bermotor, identitas pemilik,nomor registrasi kendaraan bermotor,dan masa berlaku. Tanda nomor kendaraan bermotor memuat kode wilayah, nomor registrasi,dan masa berlaku. Tanda nomor kendaraan bermotor harus memenuhi syarat bentuk, bahan, warna, dan cara pemasangan.

Satu sampai dua huruf diawal menunjukan kode wilayah pendaftaran kendaraan bermotor. Satu sampai empat sebagai nomor urut registrasi. Satu sampai tiga huruf terakhir adalah sub-wilayah dari provinsi yang terdaftar. Penentuan nomor urut ini sesuai dengan wilayah hukum Polda Metro Jaya dijelaskan pada Tabel 2.

Tabel 2. Nomor urut pelat nomor

No	Nomor urut registrasi	Jenis Kendaraan Bermotor
1	1 – 2999 , 8000 – 8999	Mobil penumpang
2	3000 – 6999	Sepeda motor
3	7000 – 7999	Mobil bus
4	9000 – 9999	Mobil barang dan kendaraan khusus

Warna pada pelat nomor ditetapkan sebagai berikut:

- Kendaraan pribadi dan rental menggunakan warna hitam dengan tulisan putih. Peraturan Kapolri Nomor 7 Tahun 2021 mewajibkan kendaraan pribadi dan rental menggunakan warna putih dengan tulisan hitam untuk memudahkan kamera tilang elektronik
- Kendaraan umum berwarna kuning dengan tulisan hitam.
- Kendaraan dinas pemerintah memiliki warna dasar merah dengan tulisan putih.
- Kendaraan korps diplomatik negara asing berwarna dasar putih dengan tulisan biru.
- Kendaraan staf operasional korps diplomatik negara asing menggunakan warna hitam dengan tulisan putih, dengan lima angka dan kode angka negara yang dicetak lebih kecil di bagian tertentu.
- Kendaraan di kawasan perdagangan bebas yang mendapatkan fasilitas pembebasan bea masuk memiliki warna hijau dengan tulisan hitam.

2.3 *Computer Vision*

Menurut Jähne dkk. (1999) *Computer vision* (visi komputer) dipahami sebagai serangkaian teknik untuk memperoleh, memproses, menganalisis dan memahami data kompleks berdimensi lebih tinggi dari lingkungan untuk eksplorasi ilmiah dan teknis. Visi komputer adalah subbidang ilmu komputer yang berfokus pada masalah membantu komputer untuk dapat mensimulasikan cara manusia melihat dan memahami lingkungan.

Terdapat empat jenis utama dari visi komputer yaitu:

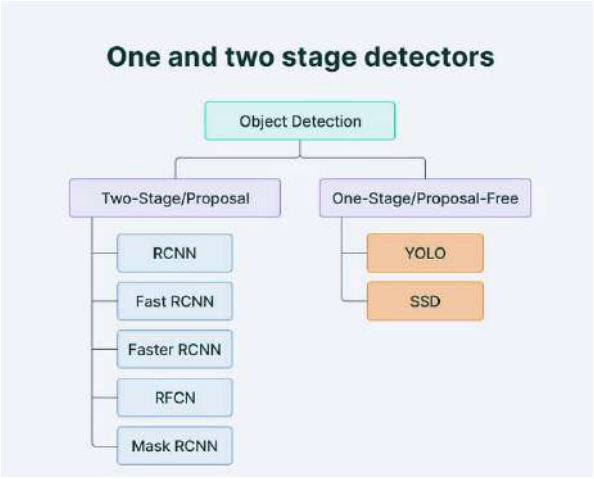
1. Klasifikasi

Klasifikasi adalah tahap di mana objek yang teridentifikasi pada bingkai gambar diidentifikasi sesuai dengan kategori objek yang diinginkan. Identifikasi objek dapat dilakukan berdasarkan parameter yang berbeda seperti bentuk, gerak, warna dan tekstur. Jadi berdasarkan parameter yang digunakan, dapat melakukan klasifikasi berdasarkan bentuk, klasifikasi berdasrkan gerak, klasifikasi berdasarkan warna, atau klasifikasi berdasarkan tekstur (Ragland dan Tharcis, 2014).

2. Deteksi

Deteksi objek adalah teknologi komputer yang berhubungan dengan pendeteksian contoh objek semantik dari kelas tertentu (seperti manusia, bangunan, atau mobil) dalam gambar dan video digital (Ragland dan Tharcis, 2014). Deteksi objek dapat dilakukan dengan menggunakan beberapa teknik dasar seperti pembedaan bingkai, aliran optik, dan pengurangan latar belakang.

Gambar 1 menjelaskan algoritme deteksi objek secara luas diklasifikan menjadi dua kategori berdasarkan beberapa kali gambar input yang dilewatkan melalui jaringan.



Gambar 1. Kategori *object detection* (Gillani dkk., 2022)

Two-stage detector membagi proses pendeteksian menjadi dua tahap: ekstrasi/pengambilan fitur diikuti dengan regresi dan klasifikasi untuk memperoleh

output (Du, 2020). Meskipun hal ini dapat memberikan akurasi yang tinggi, namun disertakan dengan kebutuhan komputasi yang tinggi sehingga tidak efisien untuk *deployment* secara *real-time*. *Single-stage detector* disisi lain menggabungkan dua tahap menjadi satu yang menggunakan satu lintasan gambar input untuk membuat prediksi tentang keberadaan gambar dan lokasi objek dalam gambar sehingga mengurangi komputasi. Namun, *single-stage detector* pada umumnya kurang akurat dan kurang efektif dalam mendeteksi objek kecil (Sultana, 2020).

3. Segmentasi

Segementasi gambar adalah teknik yang lebih lanjut dari deteksi objek karena dapat mendeteksi objek melalui segmentasi gambar karena mengambil semua piksel yang membentuk sebuah gambar dan mengelompokkannya berdasarkan apakah piksel tersebut merupakan bagian dari objek yang dicari.

4. Pelacakan objek

Pelacakan objek merupakan teknik untuk mengamati dan mencatat pergerakan suatu objek melalui serangkaian gambar berturut-turut, dengan tujuan memahami bagaimana objek tersebut bergerak relatif terhadap objek lain. Metode ini sering melibatkan pengukuran posisi pusat massa objek pada koordinat (x, y) di setiap bingkai gambar. Jenis-jenis pelacakan objek termasuk pelacakan berbasis titik, pelacakan berbasis kernel, dan pelacakan berbasis siluet (Ragland dan Tharcis, 2014).

2.4 Anotasi

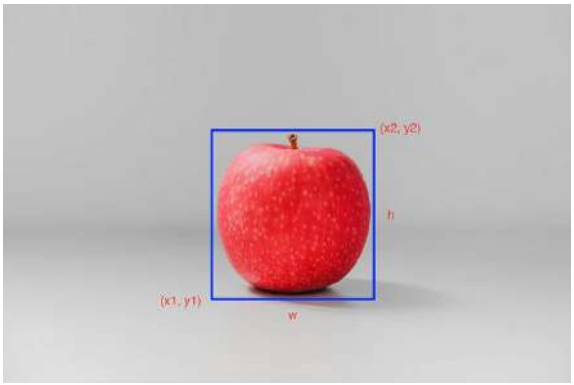
Anotasi gambar merupakan penjelasan atau deskripsi tentang data piksel dalam sebuah gambar, yang diberikan oleh manusia atau mesin (Channin dkk., 2009). Pelabelan data sangat penting dalam pengembangan model *machine learning*, terutama untuk tugas-tugas yang membutuhkan pemahaman visual. Anotasi dibuat dengan menggunakan aplikasi *software*. Berikut adalah alat yang bisa digunakan untuk membuat anotasi gambar:

1. MakeSense.AI
2. LabelImg
3. VGG image annatator
4. Roboflow
5. LabelMe
6. Scalable
7. RectLabel

Ada beberapa jenis anotasi gambar:

1. *Bounding box*

Anotasi *bounding box* merupakan teknik umum dalam komputer visi yang memerlukan penandaan objek dalam suatu citra dengan menggunakan kotak pembatas. Metode annotasi ini sering digunakan dalam tugas deteksi objek untuk menandai batas-batas dari suatu objek. Bounding box biasanya direpresentasikan oleh koordinat (x1, y1) dan (x2, y2), atau dengan koordinat (x1, y1) serta lebar (w) dan tinggi (h) dari kotak pembatas tersebut (Zheng dkk., 2022). Gambar 2 merupakan contoh *bounding box* dari jenis anotasi gambar.

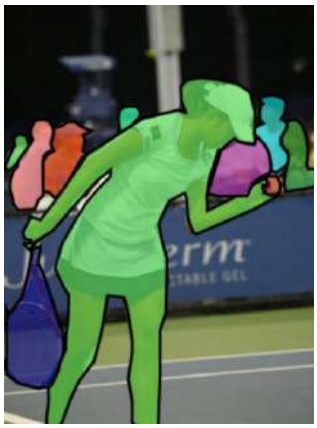


Gambar 2. Contoh *bounding box* pada buah apel (Pokhrel, 2020)

2. *Polygon segmentation*

Objek tidak selalu berbentuk persegi panjang. Mirip dengan *bounding box, polygon* mencoba menutupi objek dalam gambar dengan bantuan *polygon*. Segementasi poligon adalah jenis anotasi data lain di mana *polygon* kompleks yang digunakan sebagai pengganti persegi panjang untuk menentukan bentuk dan lokasi objek

dengan cara yang lebih tepat. Poligon dapat digunakan dalam anotasi untuk gambar data medis (Pokhrel, 2020). Gambar 3 merupakan contoh *polygon segmentation* dari jenis anotasi gambar.



Gambar 3. Contoh segementasi objek (Pokhrel, 2020)

3. 3D *cuboid*

Anotasi kuboidal merupakan evolusi dari metode topeng dalam deteksi objek dalam konteks ruang tiga dimensi. Anotasi ini memiliki signifikansi yang besar ketika melakukan tugas deteksi objek pada data tiga dimensi, terutama dalam domain medis yang sering kali melibatkan citra pemindaian. (Bandyopadhyay, 2020).

Anotasi-anotasi ini juga bisa berguna dalam melatih Algoritme untuk menggerakkan robot dan mobil, serta dalam penggunaan lengan robot di lingkungan tiga dimensi. Gambar 4 merupakan contoh anotasi 3D *cuboid*.

Gambar 4. Contoh anotasi kubus objek (Pokhrel, 2020)

4. *Semantic segmentation*

Segmentasi semantik adalah bentuk anotasi piksel demi piksel, di mana setiap piksel dalam gambar ditugaskan ke dalam kelas tertentu. Masker semantik ini memiliki berbagai aplikasi dalam berbagai bentuk segmentasi dan juga dapat diperluas untuk melatih algoritme deteksi objek. Masker semantik tersedia dalam bentuk dua dimensi dan tiga dimensi, dan dikembangkan sesuai dengan Algoritme yang diperlukan.

Segmentasi semantik digunakan dalam visi komputer untuk mobil otonom dan pencitraan medis. Pencitraan medis, segmentasi membantu mengidentifikasi dan lokalitas sel, memungkinkan pembentukan pemahaman tentang fitur-fitur bentuk seperti kebulatan, luas, dan ukuran mereka (Bandyopadhyay, 2020).

Segmentasi adalah teknologi yang penting dalam mobil otonom karena membantu mengidentifikasi dengan tepat pejalan kaki dan hambatan di jalan, yang secara signifikan mengurangi kecelakaan lalu lintas (Bandyopadhyay, 2020). Gambar 5 merupakan contoh anotasi dari *semantic segmentation*.

Gambar 5. Contoh *semantic-segmentation* objek (Pokhrel, 2020)

5. *Keypoint*/*landmark*

Anotasi titik kunci atau *landmark* hadir dalam bentuk koordinat yang menunjukkan lokasi fitur atau objek tertentu dalam gambar. Anotasi *landmark* secara khusus digunakan untuk melatih Algoritma dalam analisis data wajah guna mendeteksi fitur seperti mata, hidung, dan bibir, serta memungkinkan korelasi untuk memprediksi postur dan aktivitas manusia (Bandyopadhyay., H, 2020).

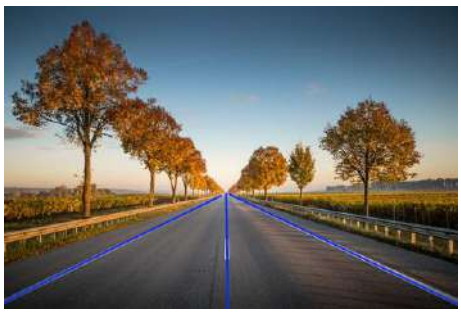
Anotasi *landmark* juga dimanfaatkan dalam pengenalan gerakan, identifikasi postur manusia, serta perhitungan objek serupa dalam citra (Bandyopadhyay, 2020). Gambar 6 adalah contoh anotasi *keypoint* objek.

Gambar 6. Contoh *keypoint* objek (Pokhrel, 2020)

6. *Polyline*

Anotasi poliline hadir sebagai serangkaian garis yang dibuat menutupi gambar input. Penggunaan poliline ini terfokus pada memberikan anotasi pada batas objek dan sering dipakai dalam tugas-tugas seperti deteksi jalur, di mana Algoritme perlu memprediksi garis daripada kelas (Bandyopadhyay, 2020).

Anotasi poliline yang memiliki tingkat presisi tinggi memiliki peran penting dalam pelatihan Algoritma mobil otonom. Anotasi ini membantu dalam pemilihan jalur yang akurat dan penentuan "wilayah yang dapat dilalui" yang vital untuk navigasi yang aman di jalan. Gambar 7 merupakan contoh dari anotasi *polyline*.



Gambar 7. Contoh polyline pada jalan (Pokhrel, 2020)

Anotasi gambar memiliki lima jenis dalam format COCO, yaitu untuk deteksi objek, *keypoint detection*, segmentasi objek, segmentasi panoptik, dan penjelasan

gambar. Anotasi-anotasi ini disimpan menggunakan format JSON (Pokhrel, 2020). Gambar 8 menjelaskan format COCO dari deteksi objek.

```
annotation{
"id" : int,
"image_id": int,
"category_id": int,
"segmentation": RLE or [polygon],
"area": float,
"bbox": [x,y,width,height],
"iscrowd": 0 or 1,
}
categories[{
"id": int,
"name": str,
"supercategory": str,
}]
```

Gambar 8. Format COCO

Pascal VOC menyimpan anotasi dalam berkas XML. Gambar 9 merupakan contoh berkas anotasi Pascal VOC untuk deteksi objek (Pokhrel, 2020).

```
<annotation>
  <folder>Train</folder>
  <filename>01.png</filename>
  <path>/path/Train/01.png</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>224</width>
    <height>224</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>36</name>
    <pose>Frontal</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <occluded>0</occluded>
    <bndbox>
      <xmin>90</xmin>
      <xmax>190</xmax>
      <ymin>54</ymin>
      <ymax>70</ymax>
    </bndbox>
  </object>
</annotation>
```

Gambar 9. Format Pascal VOC

Format label YOLO membuat berkas .txt dengan nama yang sama untuk setiap gambar dalam direktori yang sama. Setiap berkas .txt berisi anotasi untuk gambar tersebut, termasuk kelas objek, koordinat, tinggi, dan lebar objek (Pokhrel, 2020). Gambar 10 menunjukkan format anotasi YOLO.

```
<object-class> <x> <y> <width> <height>
```

Gambar 10. Format anotasi YOLO

Gambar 11 merupakan contoh anotasi dalam format YOLO.

```
0 45 55 29 67
1 99 83 28 44
```

Gambar 11. Contoh format YOLO

2.5 *You Only Look Once* (YOLO)

Model YOLO telah diterapkan pada berbagai bidang ilmu seperti pada bidang pertanian untuk medeteksi kematangan tomat (Camacho dkk., 2023), kemudian digunakan untuk mengklasifikasikan tanaman apel (Tian dkk., 2019), selanjutnya digunakan untuk deteksi hama dan penyakit (Wu dkk., 2021).

Selain itu, model YOLO juga digunakan untuk deteksi wajah secara *real time* dalam sistem biometrik keamanan dan pengenalan wajah yang pernah dilakukan dalam penelitian Yang dan Jiachun. (2020) dan Chen dkk. (2021).

Dibidang medis, YOLO digunakan untuk deteksi kanker payudara (Al-Masni dkk., 2018), deteksi melanoma (Nie dkk., 2019), dan identifikasi pil yang benar untuk memastikan pemberian obat yang aman kepada pasien (Tan dkk., 2021). Sedangkan pada aplikasi lalu lintas, model YOLO digunakan untuk tugas-tugas deteksi pelat nomor yang dilakukan oleh Aprilino dkk. (2022) dan pengenalan rambu lalu lintas dilakukan oleh Dewi dkk. (2022).

YOLO pertama kali dikembangkan oleh Joseph Redmond dkk. pada tahun 2015 dengan judul 'You Only Look Once: Unified, Real-Time Object Detection'. Model ini efisien dan cepat, populer di antara model deteksi objek lainnya seperti R-CNN, MobileNet, dan AlexNet. Joseph Redmond kemudian menghentikan penelitiannya untuk menghindari potensi penyalahgunaan teknologi. YOLOv4 dikembangkan oleh Alexei Bochkovskiy, Chien Yao Wang, dan Hong Yuan Mark Liao. Tim Ultralytics, dipimpin oleh Glenn Jocher, meluncurkan YOLOv5 yang dianggap sebagai versi terbaik. Versi lain seperti PP-YOLO (2020), YOLOv6 (2022), YOLOX (2021), YOLOR (2021), dan YOLOv7 (2022) merupakan pengembangan dari versi sebelumnya. YOLOv8 dirilis pada 2023 oleh Ultralytics, memberikan hasil yang lebih baik (Tamang dkk., 2023).

Tabel 3 merupakan ringkasan arsitektur YOLO, metrik untuk YOLO dan YOLOv2 menggunakan dataset VOC2007, sedangkan sisanya menggunakan dataset COCO2017 (Terven dan Cordova-Esparza, 2023).

Tabel 3. Ringkasan arsitektur YOLO

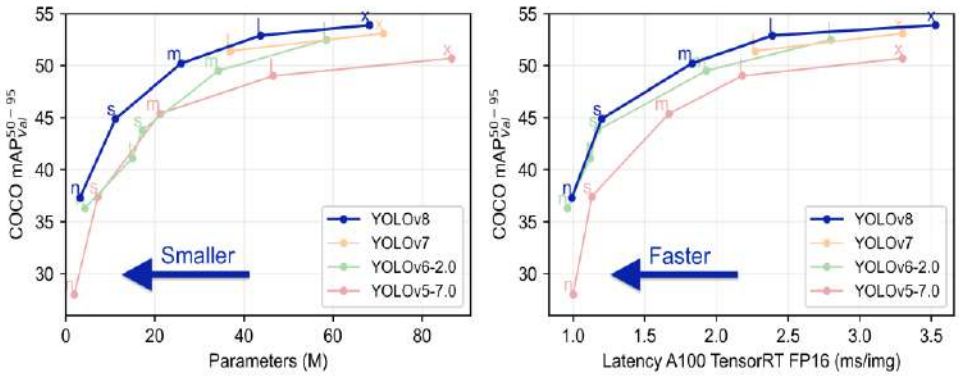
Versi	Tanggal	Anchor	Framework	Backbone	AP (%)
YOLO	2015	Tidak	Darknet	Darknet24	63.4
YOLOv2	2016	Ya	Darknet	Darknet24	63.4
YOLOv3	2018	Ya	Darknet	Darknet53	36.2
YOLOv4	2020	Ya	Darknet	CSPDarknet53	43.5
YOLOv5	2020	Ya	Pytorch	YOLOv5CSPDarknet	55.8
PP-YOLO	2020	Ya	PaddlePadd1e	ResNet50-vd	45.9
Scaled-YOLOv4	2021	Ya	Pytorch	CSPDarknet	56
PP-YOLOv2	2021	Ya	PaddlePadd1e	ResNet101-vd	50,3
YOLOR	2021	Ya	Pytorch	CSPDarknet	55.4
YOLOX	2021	No	Pytorch	YOLOXCSPDarknet	51,2
PP-YOLOE	2022	Tidak	PaddlePadd1e	CSPRepResNet	54.7
YOLOv6	2022	Tidak	Pytorch	EfficientRep	52.5
YOLOv7	2022	Tidak	Pytorch	YOLOv7Backbone	56.8
DAMO-YOLO	2022	Tidak	Pytorch	MAE-NAS	50
YOLOv8	2023	Tidak	Pytorch	YOLOv8CSPDarknet	53.9
YOLO-NAS	2023	Tidak	Pytorch	NAS	52.2

YOLOv8 adalah *state-of-the-art* model *deep learning* mendukung tugas visual seperti pengenalan, segmentasi, estimasi pose, pelacakan dan klasifikasi (Terven dan Cordova-Esparza, 2023). Tabel 4 menampilkan jenis dari YOLOv8.

Tabel 4. Jenis dari YOLOv8

Jenis Model	Bobot *pre-trained*	Tugas
YOLOv8	yolov8n.pt, yolov8s.pt, yolov8m.pt, yolov8l.pt, yolov8x.pt	Deteksi
YOLOv8-seg	yolov8n-seg.pt, yolov8s-seg.pt, yolov8m-seg.pt, yolov8l-seg.pt, yolov8x-seg.pt	*Segmentasi Instance*
YOLOv8-pose	yolov8n-pose.pt, yolov8s-pose.pt, yolov8m-pose.pt, yolov8l-pose.pt, yolov8x-pose.pt, yolov8x-pose-p6.pt	Pose/Poin Utama
YOLOv8-cls	yolov8n-cls.pt, yolov8s-cls.pt, yolov8m-cls.pt, yolov8l-cls.pt, yolov8x-cls.pt	Klasifikasi

Gambar 12 menunjukan bahwa perbandingan YOLOv8 dengan YOLOv7, YOLOv6, dan YOLOv5 yang dilatih pada ukuran piksel dengan gambar 640×640.



Gambar 12. Perbandingan antara YOLOv8 dan YOLO lainnya (Sukkar, 2024)

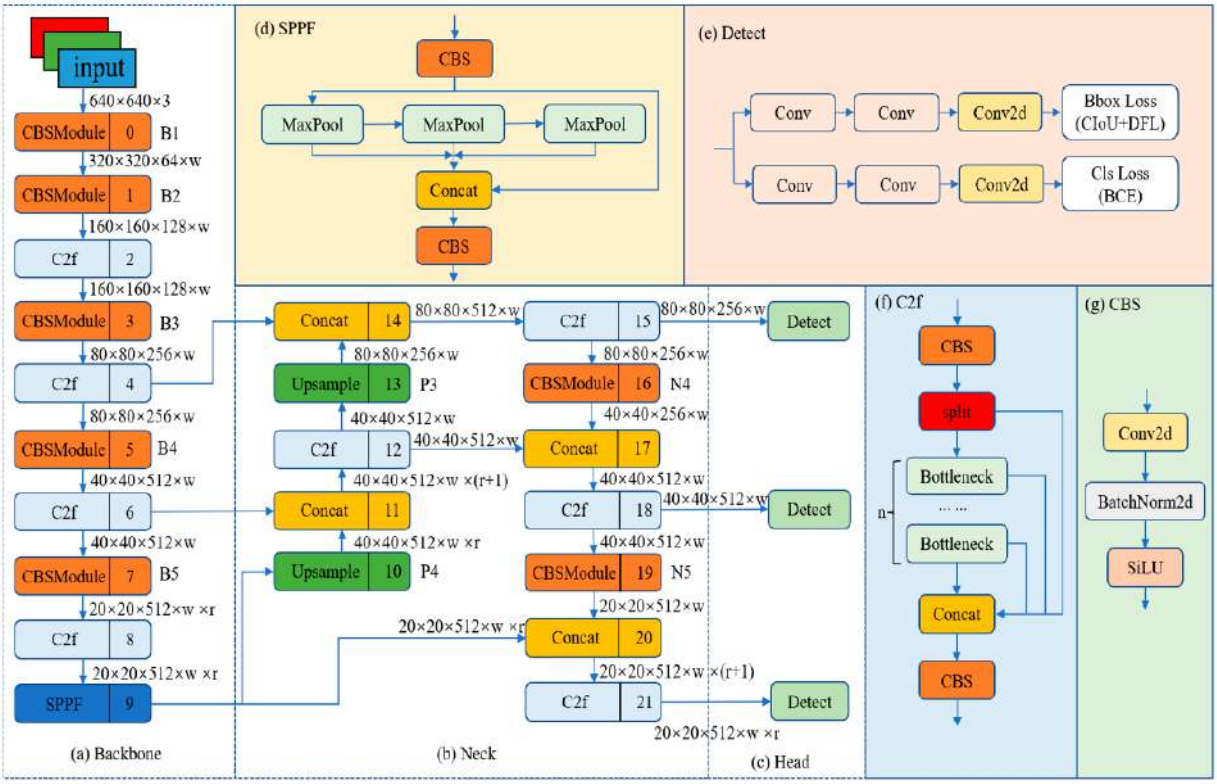
YOLOv8 dievaluasi pada dataset MS COCO test-dev 2017, Tabel 5 menjelaskan performa dari YOLOv8 untuk objek deteksi.

Tabel 5. Performa YOLOv8 untuk objek deteksi

Model	*Size* (piksels)	mAP^{val} (50 – 95)	Speed CPU ONNX (ms)	*Speed* A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80,4	0,99	3.2	8.7
YOLOv8s	640	44.9	128.4	1,20	11,2	28.6
YOLOv8m	640	50,2	234.7	1,83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

YOLOv8x mencapai *average precision* (AP) sebesar 53,9% dengan ukuran Gambar 640 piksel jika dibandingkan dengan YOLOv5 yang mencapai AP sebesar 50,7% pada ukuran input yang sama untuk kecepatan 280 FPS pada NVIDIA A100 dan TensorRT (Terven dan Cordova-Esparza, 2023).

YOLOv8 menggunakan *anchor-free* dengan *head* terpisah untuk memproses tugas objektivitas, klasifikasi, dan regresi secara independen. Desain ini memungkinkan setiap *branch* untuk fokus pada tugasnya dan meningkatkan akurasi model secara keseluruhan. Pada lapisan *output* YOLOv8, digunakan fungsi sigmoid sebagai fungsi aktivasi untuk skor objektivitas, yang mewakili probabilitas bahwa *bounding box* berisi objek. Perhitungan probabilitas dari kelas menggunakan fungsi *softmax*. YOLOv8 menggunakan CioU dan DFL untuk *bounding box loss* dan binary cross-entropy untuk *classification loss* (Terven dan Cordova-Esparza, 2023). Kerugian ini telah meningkatkan kinerja pendeteksian objek, terutama berhadapan dengan objek yang lebih kecil. Gambar 13 menunjukan arsitektur YOLOv8.



Gambar 13. Arsitektur YOLOv8 (Wang dkk., 2023)

Perbedaan dari YOLOv5 adalah sebagai berikut:

- Modul C2f digunakan sebagai pengganti modul C3.
- Ubah Conv 6 × 6 awal Backbone menjadi Conv 3 × 3.
- Hapus Conv No. 10 dan 14 dari konfigurasi YOLOv5.
- Ubah Conv 1 × 1 awal di bottleneck menjadi Conv 3 × 3.
- Menggunakan *decouple head* dan hapus *objectness branch*

Tabel 6 menyajikan parameter-parameter yang digunakan dalam objek deteksi menggunakan model YOLOv8.

Tabel 6. Parameter YOLOv8 objek deteksi.

Model	d (*depth_multiple*)	w (*width_multiple*)	r (*ratio*)
n	0,33	0,25	2.0
s	0,33	0,50	2.0
m	0,67	0,75	1,5
l	1,00	1,00	1,0
x	1,00	1,25	1,0

a) *Backbone*

YOLOv8 adalah struktur jaringan yang menggunakan CSPDarknet53 yang telah dimodifikasi sebagai jaringan *backbone* (Wang, 2023). Jaringan ini menurunkan sampel fitur input sebanyak lima kali untuk mendapatkan lima fitur skala yang berbeda, yaitu B1 hingga B5. Modul *Cross Stage Partial* (CSP) dalam jaringan *backbone* asli digantikan oleh modul C2f, dan struktur modul C2f ditunjukkan pada Gambar 13f (n menunjukkan jumlah bottleneck). Modul C2f mengadopsi koneksi *shunt gradien* untuk memperkaya aliran informasi dari jaringan ekstraksi fitur sambil menjaga bobot yang ringan. Modul CBS melakukan operasi konvolusi pada informasi input, diikuti oleh normalisasi batch, dan akhirnya mengaktifkan aliran informasi menggunakan SiLU untuk mendapatkan hasil output, seperti yang ditunjukkan pada Gambar 13g. Jaringan *backbone* akhirnya menggunakan modul *spatial pyramid pooling fast* (SPPF) untuk mengumpulkan peta fitur input menjadi peta ukuran tetap untuk output ukuran adaptif. Dibandingkan dengan struktur *spatial pyramid pooling* (SPP), SPPF mengurangi upaya komputasi dan memiliki latensi lebih rendah dengan menghubungkan tiga lapisan *pooling* maksimum secara berurutan, seperti yang ditunjukkan pada Gambar 13d.

b) *Neck*

YOLOv8 terinspirasi oleh PANet dan mengadopsi struktur *path aggregation network* (PAN) – *feature pyramid network* (FPN) pada bagian *neck*, yang ditampilkan dalam Gambar 13b. Perbandingan dengan model YOLOv5 dan YOLOv7, YOLOv8 menghilangkan operasi konvolusi setelah pengambilan sampel dalam struktur PAN. Hal ini dilakukan untuk mempertahankan kinerja model yang asli sambil tetap menghasilkan model yang lebih ringan (Wang dkk., 2023).

Pada Struktur PAN dan FPN model YOLOv8, ditunjukan pada Gambar 13 bagian P4-P5 dan N4-N5 untuk menggambarkan dua skala fitur yang berbeda. FPN tradisional biasanya menggunakan pendekatan *top-down* untuk mengirimkan informasi semantik yang lebih dalam. Meskipun FPN meningkatkan informasi semantik fitur dengan menggabungkan B4-P4 dan B3-P3, beberapa informasi lokalisasi objek dapat hilang (Wang dkk., 2023).

Mengatasi masalah tersebut, PAN-FPN menggabungkan PAN ke dalam FPN. Struktur PAN membantu meningkatkan pembelajaran informasi lokasi dengan menggabungkan P4-N4 dan P5-N5, menciptakan peningkatan jalur dalam bentuk top-down. PAN-FPN menciptakan struktur jaringan yang mengintegrasikan pendekatan *top-down* dan *bottom-up*, yang memungkinkan komplementaritas antara informasi posisi yang dangkal dan informasi semantik yang dalam melalui fusi fitur. Hal ini menghasilkan keragaman dan kelengkapan fitur yang diperlukan dalam pendeteksian objek (Wang dkk., 2023).

c) *Head*

Bagian deteksi dalam YOLOv8 mengadopsi struktur *decoupled head*, seperti yang ditunjukkan dalam Gambar 13e. Struktur *decoupled head* ini menggunkan pendekatan dengan dua cabang terpisah, satu untuk mengklasifikasikan objek dan yang lain untuk merespons regresi *bounding box* yang diprediksi. Dua jenis tugas ini dikenakan fungsi *loss* yang berbeda. Tugas klasifikasi, digunakan *binary cross-entropy loss* (BCE Loss). Tugas regresi *bounding box* yang diprediksi, digunakan *distribution focal loss* (DFL) dan CioU (Wang dkk., 2023).

Struktur deteksi ini memiliki potensi untuk meningkatkan akurasi deteksi dan mempercepat proses konvergensi model. YOLOv8 adalah contoh model deteksi tanpa penggunaan anchor yang memberi definisi singkat mengenai sampel positif dan negatif. Model YOLOv8 juga memanfaatkan *Task-Aligned Assigner* untuk menentukan sampel secara dinamis, yang pada gilirannya meningkatkan akurasi deteksi dan daya tahan model (Wang dkk., 2023).

2.6 Fungsi Aktivasi

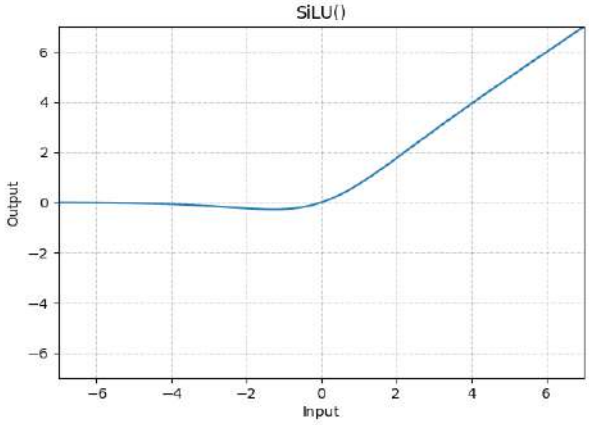
Fungsi aktivasi adalah suatu elemen yang diperkenalkan ke dalam jaringan saraf tiruan dengan tujuan membantu jaringan tersebut dalam memahami dan mempelajari pola-pola yang rumit dalam data. Ketika dibandingkan dengan model

berbasis neuron dalam otak manusia, fungsi aktivasi akhirnya menentukan sinyal mana yang akan diteruskan ke neuron berikutnya (Bag, 2021).

A. Fungsi Aktivasi *Sigmoid Linear Units*

Sigmoid Linear Units (SiLU), juga dikenal sebagai Swish, diusulkan oleh Elfwing dkk. (2018) sebagai fungsi aktivasi dalam jaringan saraf untuk reinforcement learning. Rumusnya dijelaskan pada Persamaan (2.1) dan grafiknya ditunjukkan pada Gambar 14.

$$f(x) = x \cdot sigmoid(x) = \frac{x}{(1 + e^{-x})} \tag{2.1}$$

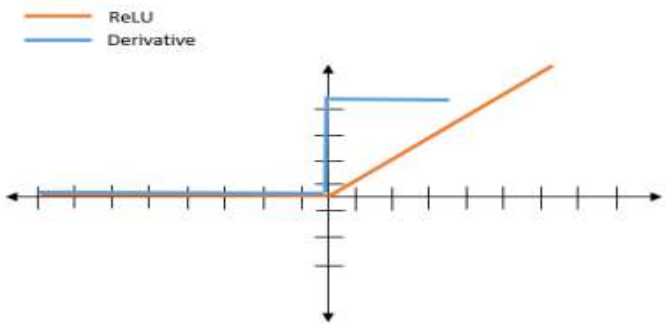


Gambar 14. Fungsi aktivasi SiLU (PyTorch Contributors, 2023)

B. Fungsi Aktivasi *Rectified Linear Units*

Rectified Linear Units (ReLU) adalah fungsi aktivasi yang diperkenalkan oleh Hahnloser dkk. (2000) yang memiliki dasar biologis dan matematis yang kuat. Pada tahun 2011, fungsi ini menunjukan kinerjanya untuk meningkatkan dalam pelatihan *deep neural network* dengan hasil tercanggih hingga saat ini (Nair dan Hinton, 2010). Fungsi aktivasi ReLU merupakan fungsi aktivasi non-linier yang dapat melakukan operasi turunan (Kiliçarslan dkk., 2021). ReLU adalah fungsi sederhana yang merupakan fungsi identitas untuk masukan positif dan nol untuk masukan negatif ,fungsi ini dijelaskan pada Persamaan (2.2) dan Gambar 15.

$$\mathrm{ReLU(x)} = \max(0, \mathrm{x}) = \begin{cases} x, & x \geq 0 \\ 0, & lainnya \end{cases} \tag{2.2}$$

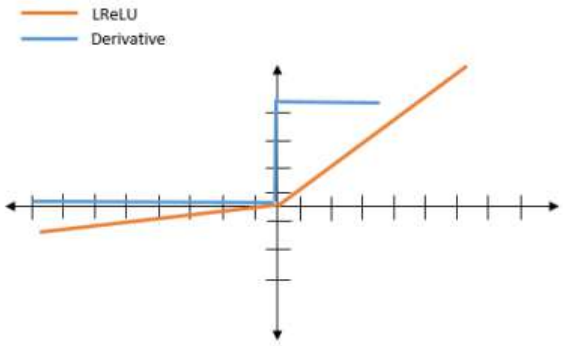


Gambar 15. Fungsi aktivasi ReLU (Kiliçarslan dkk., 2021)

C. Fungsi Aktivasi Leaky ReLU

Fungsi aktivasi ReLU memiliki masalah utama, yaitu kehilangan gradien karena mengabaikan nilai negatif dan menyebabkan gradien menjadi 0, Solusinya adalah menggunakan parameter α, yang mencegah gradien menjadi nol selama pelatihan. Leaky ReLU adalah alternatif yang mengatasi masalah ini dengan memberikan nilai gradien yang sangat kecil untuk nilai negatif, sering kali menggunakan α dalam kisaran 0,01, Leaky ReLU didefinisikan oleh Persamaan (2.3) dan grafiknya dapat dilihat pada Gambar 16.

$$LReLU(x) = \begin{cases} x, & x \geq 0 \\ 0{,}01x, & lainnya \end{cases} \quad (2.3)$$



Gambar 16. Fungsi aktivasi *Leaky ReLU (*Kiliçarslan dkk., 2021)

D. *Gaussian Error Linear Units* Activation Function

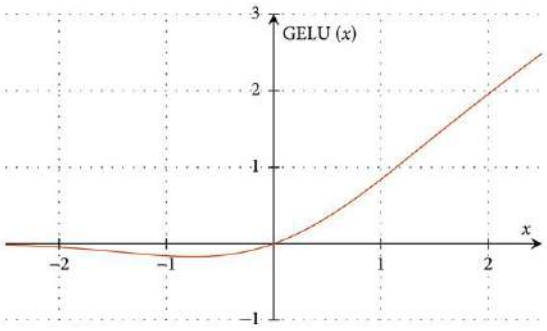
Fungsi aktivasi *Gaussian Error Linear Units* (GELU), yang diperkenalkan oleh Hendrycks dan Gimpel (2016), merupakan aproksimasi diferensial dan *smooth* dari fungsi rectifier. GELU populer dalam pembelajaran mendalam karena sifat-sifatnya yang diinginkan, seperti nonlinearitas, *differentiability*, dan *smoothness*. Oleh

karena itu, GELU digunakan dalam BERT, ViT, dan GPT. Karena fungsi distribusi kumulatif Gaussian sering dihitung dengan fungsi kesalahan, definisi GELU ditulis dengan Persamaan (2.4) sebagai berikut:

$$\begin{aligned} \mathrm{GELU}(x) &= xP(X \leq x) = x\Phi(x) \\ &= x \cdot \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \end{aligned} \tag{2.4}$$

Nilai aproksimasi dalam Persamaan (2.5) dan Gambar 17 menunjukan grafi dari fungsi GELU:

$$f(x) = 0{,}5x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}(x + 0{,}044715x^3)\right]\right) \tag{2.5}$$



Gambar 17. Fungsi aktivasi GELU (Lee, 2023)

2.7 Pengenalan Karakter

Optical character recognition (OCR) adalah konversi gambar teks yang dipindai atau dicetak (Shinde, 2012), teks tulisan tangan menjadi teks yang dapat diedit untuk proses lebih lanjut (Patel, 2012). Teknologi ini memungkinkan mesin mengenali teks secara otomatis. Seperti kombinasi antara mata dan pikiran manusia. Mata dapat melihat teks dari gambar tetapi otak memproses serta menterjemahkan teks yang diekstraksi dan dibaca oleh mata (Patel, 2012).

Menurut Islam dkk. (2016) sebuah sistem OCR umum terdiri dari beberapa fase sebagai berikut:

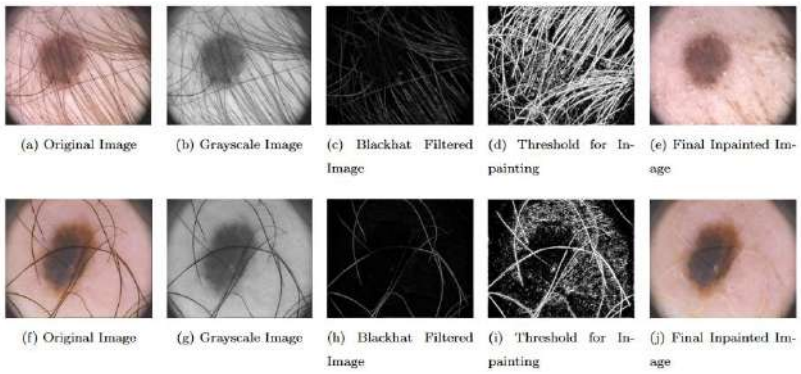
1. Akuisisi gambar: Mengambil gambar dari sumber eksternal seperti pemindai atau kamera, dll.
2. Pra-pemrosesan: Setelah gambar telah diakuisisi, berbagai langkah pra-pemrosesan dapat dilakukan untuk meningkatkan kualitas gambar. Beberapa teknik pra-pemrosesan termasuk penghilangan *noise*, pengambilan ambang batas, dan ekstraksi garis dasar gambar, dll.
3. Segmentasi karakter: Pada langkah ini, karakter-karakter dalam gambar dipisahkan sehingga mereka dapat diteruskan ke mesin pengenalan. Beberapa teknik paling sederhana termasuk analisis komponen terhubung dan profil proyeksi. Namun, dalam situasi yang kompleks di mana karakter-karakter tumpang tindih atau rusak, atau ada *noise* pada gambar, teknik segmentasi karakter lanjutan digunakan.
4. Ekstraksi fitur: Karakter-karakter yang telah di-segmentasi kemudian diproses untuk mengekstraksi berbagai fitur. Berdasarkan fitur-fitur ini, karakter-karakter dikenali. Jenis fitur yang dapat diekstraksi dari gambar termasuk momen, dll. Fitur-fitur yang diekstraksi seharusnya dapat dihitung dengan efisien, mengurangi variasi dalam kelas, dan memaksimalkan variasi antar kelas.
5. Klasifikasi karakter: Langkah ini memetakan fitur-fitur dari gambar yang telah di-segmentasi ke berbagai kategori atau kelas. Ada berbagai teknik klasifikasi karakter. Teknik klasifikasi struktural didasarkan pada fitur-fitur yang diekstraksi dari struktur gambar dan menggunakan aturan keputusan yang berbeda untuk mengklasifikasikan karakter-karakter. Metode klasifikasi pola statistik didasarkan pada model probabilistik dan metode statistik lainnya untuk mengklasifikasikan karakter-karakter.
6. Pasca-pemrosesan: Setelah klasifikasi, hasilnya tidak selalu 100% benar, terutama untuk bahasa yang kompleks. Teknik pasca-pemrosesan dapat dilakukan untuk meningkatkan akurasi sistem OCR. Teknik ini menggunakan pemrosesan bahasa alami, konteks geometri, dan bahasa untuk memperbaiki kesalahan dalam hasil OCR. Sebagai contoh, pasca-pemroses dapat menggunakan pemeriksa ejaan dan kamus, model probabilitas seperti rantai

Markov dan n-gram untuk meningkatkan akurasi. Kompleksitas waktu dan ruang dari pra-pemroses seharusnya tidak terlalu tinggi, dan penggunaan pra-pemroses seharusnya tidak menyebabkan kesalahan baru.

2.8 *Image Processing*

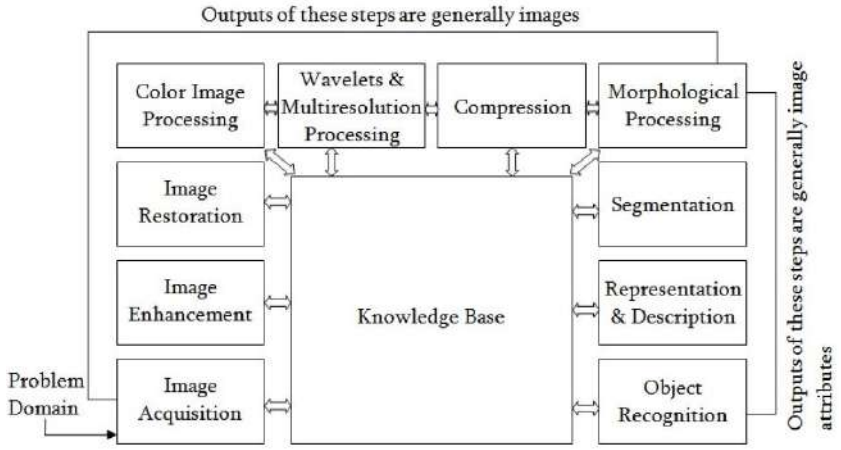
Image processing adalah kumpulan metode yang berurusan dengan manipulasi gambar digital menggunakan algoritma komputer (Sulistiyanti dkk., 2016). Langkah ini merupakan sangat penting dalam banyak aplikasi, seperti deteksi objek, pengenalan wajah, dan kompresi gambar. Tujuan utama pengolahan gambar adalah untuk mendapatkan informasi yang bermanfaat atau meningkatkan gambar asli melalui operasi yang diterapkan padanya. Pengolahan gambar adalah pemrosesan sinyal karena input yang diberikan ke program adalah gambar digital, dan keluaran yang diharapkan adalah bentuk baru dari gambar atau informasi tentang gambar tersebut (Sethy dkk., 2020).

Pada saat mempertimbangkan bidang pengolahan gambar atau visi komputer, tujuannya adalah memberikan kemampuan visual kepada mesin. Secara perspektif, pengolahan gambar menjelaskan proses mengubah sistem visual manusia ke dalam bentuk gambar digital. Beberapa tahap pengolahan pada gambar diperlukan untuk memperoleh hasil yang paling realistik dari gambar digital (Altunay, 2019). Contoh penerapan dari pengolahan gambar diilustrasikan pada Gambar 18 sebagai berikut.



Gambar 18. Contoh penerapan dari *image processing* (Basak dkk., 2022)

Gambar 19 menjelaskan langkah-langkah utama dalam pengolahan gambar sehingga dapat memahami jenis operasi yang dapat dijalankan dalam bidang ini dan hasil yang diperoleh setelah setiap operasi.



Gambar 19. Langkah fundamental dari *image processing (Gonzalez, 2009)*

Image acquisition: Pengambilan gambar digital adalah langkah pertama dalam sistem pengolahan gambar digital yang melibatkan pengambilan gambar dari berbagai perangkat keras. Gambar asli yang dihasilkan adalah gambar mentah yang belum diproses. Tahap ini melibatkan beberapa langkah pra-pemrosesan, seperti penskalaan. Hasil yang diperoleh adalah gambar yang telah diproses dan siap digunakan sebagai data masukan untuk seluruh sistem pengolahan gambar (Gonzalez, 2009).

Image enhancement: Peningkatan gambar dalam pengolahan gambar digital adalah proses penyesuaian dan manipulasi gambar untuk aplikasi tertentu. Metode yang digunakan meliputi perluasan kontras, pemrosesan histogram, penyaringan spasial, deteksi tepi, serta transformasi seperti PCA dan HSI (Zhang dan Lin, 2021). Metode ini bervariasi tergantung pada tugas dalam sistem pengolahan gambar, dan prosesnya sangat subjektif (Gonzalez, 2009).

Image restoration: Pada langkah ini, difokuskan pada meningkatkan penampilan gambar, dan melakukan operasi yang bersifat objektif karena degradasi gambar dapat dijelaskan melalui model matematis atau probabilistik (Gonzalez, 2009).

Sebagai contoh, operasi ini mencakup penghilangan *noise* atau blur dari gambar (Zhang dan Lin, 2021).

Color image processing: Pemrosesan gambar berwarna merupakan bidang yang semakin penting karena peningkatan signifikan dalam penggunaan gambar digital melalui Internet (Gonzalez, 2009).

Wavelets and multi-resolution processing: Wavelet merupakan dasar untuk merepresentasikan gambar dalam berbagai tingkat resolusi. Proses ini melibatkan subdivisi gambar secara berurutan menjadi wilayah-wilayah yang lebih kecil, yang berguna untuk kompresi data dan menciptakan representasi piramida (Gonzalez, 2009).

Image compression: Untuk mentransfer gambar ke perangkat lain atau menghemat ruang penyimpanan, gambar harus dikompresi. Ini penting untuk menampilkan gambar di internet, seperti thumbnail di Google yang merupakan versi terkompresi. Gambar asli hanya ditampilkan saat diklik, sehingga menghemat bandwidth server. (Khundu, 2022).

Morphological processing: Pemrosesan Morfologi digunakan untuk ekstraksi komponen-komponen gambar yang relevan untuk meningkatkan representasi dan deskripsi bentuk objek. Contohnya, operasi erosi dan dilasi digunakan untuk mempertajam dan menghaluskan tepi objek (Khundu, 2022).

Image segmentation: Segmentasi merupakan proses membagi gambar menjadi beberapa segmen untuk mengidentifikasi objek dalam gambar digital. Segmentasi otomatis merupakan tugas yang sulit dalam pengolahan gambar digital. Proses segmentasi yang handal memberikan solusi yang optimal terutama pada masalah pemrosesan gambar yang memerlukan identifikasi objek secara individual. Algoritma segmentasi yang lemah dapat mengakibatkan kegagalan dalam pengenalan. Sebaliknya akurasi segmentasi yang semakin tinggi, keberhasilan pengenalan semakin besar (Gonzalez, 2009).

Representation and description: representasi dan deskripsi umumnya mengikuti tahap segmentasi, yang biasanya menghasilkan data piksel mentah berupa batas wilayah atau seluruh titik dalam wilayah itu sendiri. Pada pemrosesan komputer, perubahan data ke bentuk yang sesuai merupakan kewajiban yang harus dilakukan. Menurut Gonzalez (2009) terdapat dua keputusan, keputusan pertama adalah apakah data harus direpresentasikan sebagai batas atau wilayah lengkap. Representasi batas cocok untuk karakteristik bentuk eksternal, sementara representasi wilayah cocok untuk sifat internal seperti tekstur atau bentuk kerangka. Memilih represemtasi hanyalah sebagian dari solusi untuk mengubah data mentah menjadi bentuk yang sesuai untuk pemerosesan komputer. Selanjutnya, keputusan kedua adalah menentukan metode untuk mendeskripsikan data agar fitur menarik dapat disorot. Deskripsi, atau pemilihan fitur, melibatkan ekstraksi atribut yang menghasilkan informasi kuantitatif menarik atau dasar untuk membedakan kelas objek.

Object detection and recognition: *recognition* merupakan proses pemberian label pada suatu objek berdasarkan deskripsinya (Gonzales, 2009).

Pengenalan matematika dasar yang digunakan dalam pengolahan citra memiliki dua tujuan utama menurut Gonzales (2009) :

1. Memperkenalkan berbagai alat matematika yang digunakan.
2. Membantu mulai mengembangkan tentang bagaimana atal ini digunakan dengan menerapkannya pada berbagai tugas pemrosesan gambar dasar.

Pada bagian pengenalan alat matematika dasar yang digunakan dalam pengolahan gambar digital

.

a. Operasi *elementwise* versus operasi matriks

Misalnya perhatikan gambar 2x2 pada Persamaan (2.6).

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} dan \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad (2.6)$$

Hasil kali elemen sering dilambangkan dengan simbol $\odot$ atau $\otimes$ dari kedua gambar yang ditunjukan pada Persamaan (2.7).

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix} \quad (2.7)$$

Sebaliknya, hasil kali matriks gambar dibentuk menggunakan aturan perkalian matriks yang ditunjukan pada PersamaaSn (2.8).

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{21} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix} \quad (2.8)$$

b. Operasi *linear* versus operasi *non-linear*

Salah satu yang terpenting dari metode pemrosesan gambar adalah apakah metode tersebut *linier* atau *non-linier*. Operator umum H yang menghasilkan gambar keluaran g(x,y) dari gambar masukan tertentu f(x,y) dituliskan pada Persamaan (2.9):

$$H[f(x,y)] = g(x,y) \quad (2.9)$$

H dikatakan sebagai operasi linear diberikan pada Persamaan (2.10).

$$\begin{aligned} H[af_1(x,y) + bf_2(x,y)] &= aH[f_1(x,y)] + bH[f_2(x,y)] \\ &= ag_1(x,y) + bg_2(x,y) \end{aligned} \quad (2.10)$$

Persamaan (2.10) menunjukkan bahwa keluaran suatu operasi linier yang diterapkan pada jumlah dua masukan adalah sama dengan melakukan operasi satu per satu pada masukan-masukan tersebut dan kemudian menjumlahkan hasilnya. Selain itu, keluaran operasi *linier* pada suatu konstanta dikalikan dengan suatu masukan sama dengan keluaran operasi karena masukan asli dikalikan dengan konstanta tersebut. Sifat pertama disebut sifat aditif, dan sifat kedua disebut sifat homogenitas. Menurut definisi, operator yang gagal memenuhi Persamaan (2.10) dikatakan *non-linier*.

Sebagai contoh, misalkan H adalah operator penjumlahan Σ. Fungsi yang dilakukan oleh operator ini hanya menjumlahkan input. Pengujian linearitas ditunjukan pada Persamaan (2.10) dengan melakukan pembuktian bahwa sisi kiri sama dengan sisi kanan:

$$\sum[af_1(x,y)+bf_2(x,y)] = \sum a[f_1(x,y)]+\sum b[f_2(x,y)]$$
$$= a\sum f_1(x,y)+b\sum f_2(x,y)$$
$$= a_i g_i(x,y)+a_j g_j(x,y)$$

Sehingga langkah pertama adalah mengikuti fakta bahwa penjumlahan bersifat distributif. Sehingga perluasan ruas kiri sama dengan ruas kanan. Persamaan (2.10) menyimpulkan bahwa operator penjumlahan adalah linier.

Sebaliknya, apabila bekerja dengan operasi *max*, maka fungsinya bertujuan untuk mencari nilai maksimum piksel dalam suatu gambar. Cara paling sederhana untuk membuktikan bahwa operator ini *non-linier* adalah dengan mencari contoh yang gagal dalam pengujian pada Persamaan (2.10). Perhatikan dua gambar berikut.

$$f_1 = \begin{bmatrix}0 & 2\\ 2 & 3\end{bmatrix} \; dan \; f_2 = \begin{bmatrix}6 & 5\\ 4 & 7\end{bmatrix}$$

dan misalkan diberikan a = 1 dan b = −1, Menguji linearitas, dimulai dengan sisi kiri Persamaan. (2.10):

$$\max\left\{(1)\begin{bmatrix}0 & 2\\ 2 & 3\end{bmatrix}+(-1)\begin{bmatrix}6 & 5\\ 4 & 7\end{bmatrix}\right\} = \max\left\{\begin{bmatrix}-6 & -3\\ -2 & -4\end{bmatrix}\right\}$$
$$= -2$$

Selanjutnya dengan sisi kanan, diperoleh

$$(1)\max\left\{\begin{bmatrix}0 & 2\\ 2 & 3\end{bmatrix}\right\}+(-1)\max\left\{\begin{bmatrix}6 & 5\\ 4 & 7\end{bmatrix}\right\} = 3+(-1)(7)$$
$$= -4$$

Sisi kiri dan kanan Persamaan. (2.10) tidak sama dalam kasus ini, jadi telah dibuktikan bahwa operator maks adalah *non-linier*.

c. Operasi artimatika

Operasi aritmatika antara dua gambar f(x,y) dan g(x,y) dinotasikan pada Persamaan (2.11) – (2.14).

$$s(x,y) = f(x,y)+g(x,y) \quad (2.11)$$
$$d(x,y) = f(x,y)-g(x,y) \quad (2.12)$$

$$p(x,y) = f(x,y) \times g(x,y) \quad (2.13)$$

$$v(x,y) = f(x,y) \div g(x,y) \quad (2.14)$$

Operasi elemen-per-elemen pada Persamaan (2.11) – (2.14) dilakukan antara pasangan piksel yang sesuai dalam f dan g untuk $x = 0,1,\dots,M-1$ dan $y = 0,1,\dots,N-1$, M dan N adalah ukuran baris dan kolom dari gambar. Sedangkan s,d,p, dan v merupakan bayangan $M \times N$.

Misalkan g(x, y) adalah citra rusak yang dibentuk oleh penambahan *noise* $\eta(x,y)$ pada citra tanpa *noise* f(x, y) ditunjukan pada Persamaan (2.15).

$$g(x,y) = f(x,y) + \eta(x,y) \quad (2.15)$$

Diasumsikan pada setiap pasangan koordinat (x, y) *noise* tidak berkorelasi dengan nilai citra dan nilai rata-ratanya nol.

Jika *noise* memenuhi, dapat ditunjukkan pada Persamaan (2.16) bahwa jika suatu citra g(x, y) dibentuk dengan rata-rata K citra *noise* yang berbeda.

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y) \quad (2.16)$$

Persamaan (2.17) menyatakan nilai ekspetasi sama dengan citra tanpa *noise*

$$E\{\bar{g}(x,y)\} = f(x,y) \quad (2.17)$$

Nilai varian pada rata-rata citra rusak diberikan pada Persamaan (2.18)

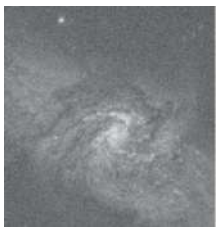
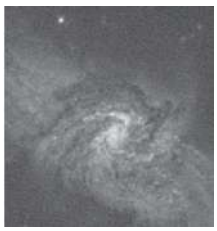
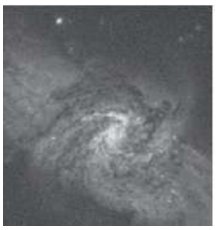
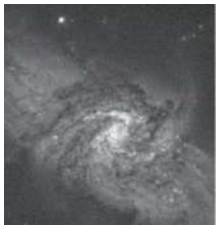
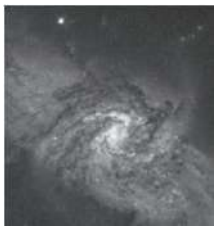
$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)} \quad (2.18)$$

$E\{\bar{g}(x,y)\}$ adalah nilai yang diharapkan dari $\bar{g}(x,y)$. $\sigma^2_{\bar{g}(x,y)}$ dan $\sigma^2_{\eta(x,y)}$ adalah varians dari $\bar{g}(x,y)$ dan $\eta(x,y)$ pada koordinat (x, y). Varians ini merupakan array dengan ukuran yang sama dengan gambar masukan, dan terdapat nilai varians skalar untuk setiap lokasi piksel.

Simpangan baku adalah akar kuadrat dari varians pada setiap titik (x, y) dalam gambar rata-rata ditunjukan melalui Persamaan (2.19).

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)} \tag{2.19}$$

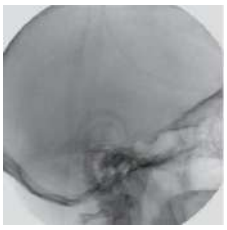
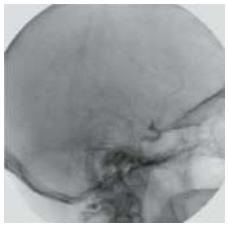
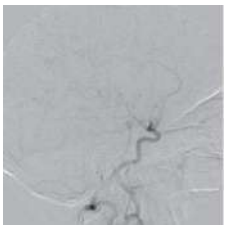
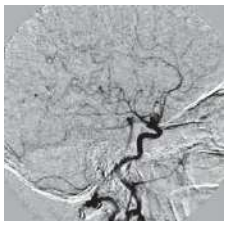
Ketika K meningkat, Persamaan. (2.18) dan (2.19) menunjukkan bahwa variabilitas nilai piksel di setiap lokasi (x, y) menurun. Hal ini disebabkan $E\{\bar{g}(x,y)\} = f(x,y)$, yang berarti bahwa $\bar{g}(x,y)$ mendekati citra tanpa *noise* f(x, y) seiring bertambahnya jumlah citra *noise* yang digunakan dalam proses rata-rata. Gambar 20 merupakan penerapan dari penambahan nilai citra *noise*.

 (a) Gambar Pasangan Galaksi NGC 3314 yang rusak akibat Gaussian aditif.	 (b) Hasil rata-rata 5 gambar *noise.*	 (c) Hasil rata-rata 10 gambar *noise.*
 (d) Hasil rata-rata 20 gambar *noise.*	 (e) Hasil rata-rata 50 gambar *noise.*	 (f) Hasil rata-rata 100 gambar *noise.*

Gambar 20. Galaxy (Gambar asli milik NASA)

Image subtraction adalah teknik yang digunakan untuk meningkatkan perbedaan antara dua gambar. Bayangkan terdapat dua gambar yang hampir sama, tetapi ada beberapa perbedaan kecil di antaranya. Menggunakan *image subtraction*, dapat mengurangi bagian-bagian yang sama dari kedua gambar sehingga hanya

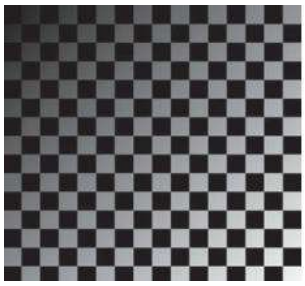
perbedaannya saja yang terlihat jelas. Penerapan *image subraction* dibidang medis seperti pada gambar 21 tentang *mask mode radiography.*

 (a) Gambar mask X-ray kepala pasien sebelum yodium diinjeksikan.	 (b) Contoh gambar yang diambil setelah diinjeksi.
 (c) Perbedaan antara (a) dan (b)	 (d) Gambar perbedaan yang ditingkatkan.

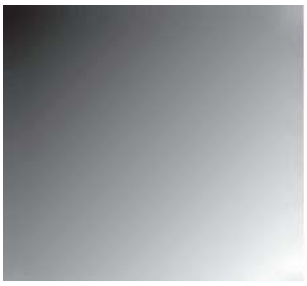
Gambar 21. Angiografi pengurangan digital. (Gambar (a) dan (b) milik Image Sciences Institute, University Medical Center, Utrecht, Belanda.)

Perkalian dan pembagian gambar adalah operasi matematis penting dalam pemrosesan gambar. Perkalian gambar melibatkan pengalian nilai piksel dari dua gambar pada posisi yang sama, yang sering digunakan untuk menggabungkan informasi dari kedua gambar tersebut. Pembagian gambar melibatkan pembagian nilai piksel dari satu gambar dengan nilai piksel dari gambar lain pada posisi yang sama, yang berguna untuk menghilangkan efek yang tidak diinginkan, seperti *shading correction*. Misalkan sensor pencitraan menghasilkan gambar yang dapat dimodelkan sebagai produk dari "gambar sempurna", dilambangkan dengan f(x, y), dikali fungsi bayangan, h(x, y), yaitu g(x, y) = f(x, y)h(x, y). apabila h(x, y) diketahui atau dapat diestimasi, maka f(x,y) diperoleh dengan cara mengalikan bayangan dengan invers dari h(x, y) atau dengan membagi g oleh h menggunakan pembagian elemen.

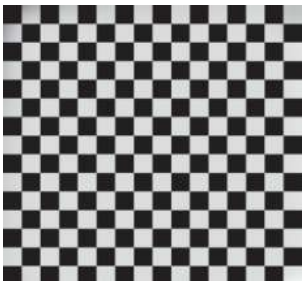
Gambar 22 menunjukkan contoh *shading correction* dengan menggunakan estimasi pola arsiran.



(a) Pola pengujian yang diarsir.



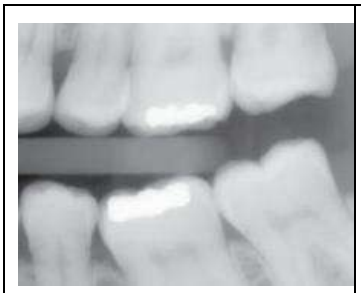
(b) Perkiraan pola shading.



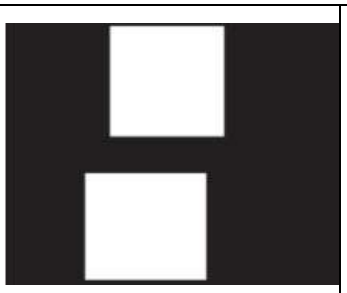
(c) perkalian dari (a) dengan invers dari (b).

Gambar 22. *Shading correction*.

Kegunaan lain dari perkalian gambar adalah dalam operasi masking, yang disebut juga *region of interest* (ROI) seperti ditunjukkan pada Gambar 23.



(a) Gambar rontgen gigi digital.



(b) Mask ROI untuk mengisolasi gigi dengan tambalan

(c) Perkalian dari (a) dan (b).

Gambar 23 aplikasi masking ROI pada rontgen gigi digital.

d. Himpunan dan operasi logika

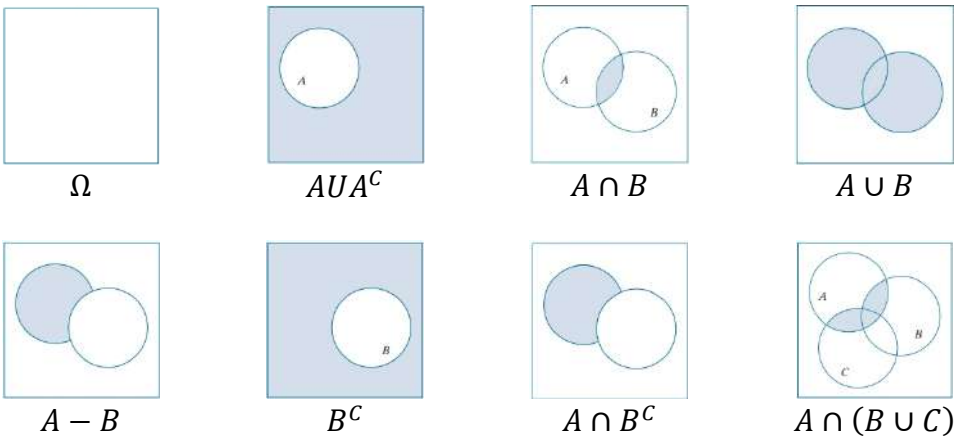
Pada bagian ini membahas tentang dasar-dasar teori himpunan, beberapa himpunan penting dan operasi logika.

Ruang sampel, Ω yang juga disebut himpunan semesta adalah himpunan semua elemen himpunan yang mungkin dalam suatu aplikasi tertentu. Pemrosesan gambar, biasanya mendefinisikan Ω sebagai persegi panjang yang berisi semua piksel dalam sebuah gambar. Tabel 7 menjelaskan tentang operasi dan hubungan himpunan penting.

Tabel 7. Operasi dan hubungan himpunan penting.

Deskripsi	Ekspresi
Operasi antara ruang sampel dan himpunan kosong	$\Omega^c = \emptyset;\ \emptyset^c = \ \Omega;\ \Omega \cup \emptyset = \Omega; \Omega \cap \emptyset = \emptyset$
Gabungan dan perpotongan himpunan ruang nol dan ruang sampel	$A \cup \emptyset = \mathrm{A};\ A \cap \emptyset = \emptyset; \mathrm{A} \cup\ \Omega = \Omega; A \cap \Omega = A$
Gabungan dan perpotongan suatu himpunan dengan dirinya sendiri	$A \cup A = A; A \cap A = A$
Gabungan dan perpotongan suatu himpunan dengan komplemennya	$A \cup A^c = \Omega; A \cap A^c = \emptyset$
Aturan komutatif	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Aturan asosiatif	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap A)$
Aturan distributif	$(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$ $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$
Aturan DeMorgan	$(A \cup B)^c = A^c \cap B^c$ $(A \cap B)^c = A^c \cup B^c$

Ilustrasi singkat operasi himpunan yang melibatkan gambar skala abu-abu ditampilkan pada Gambar 24 sebagai berikut:



Gambar 24. Diagram venn yang sesuai dengan beberapa operasi himpunan

Misalkan elemen-elemen citra skala abu-abu diwakili oleh himpunan A yang elemen-elemennya merupakan triplet berbentuk (x,y,z), dengan x dan y adalah koordinat spasial, dan z menyatakan nilai intensitas. Komplemen dari A sebagai himpunan didefinisikan pada Persamaan (2.20).

$$A^c = \{(x, y, K - z) | (x, y, z) \in A\} \tag{2.20}$$

Komplemen A merupakan himpunan piksel A yang intensitasnya telah dikurangi dari konstanta K. Konstanta ini sama dengan nilai intensitas maksimum pada gambar $2^k - 1$, dengan k adalah jumlah bit yang digunakan untuk mewakili z yang diberikan pada Persamaan (2.21).

$$A^c = \{(x, y, 255 - z) | (x, y, z) \in A\} \tag{2.21}$$

Gambar 25(a) adalah gambar original. Gambar 25(b) adalah citra negatif yang diperoleh dengan menggunakan operasi komplemen dari Gambar 25(a). Selanjutnya gabungan antara gambar (a) dan (b) dihasilkan pada Gambar 25(c).

		
(a) Gambar original	(b) Citra negatif yang diperoleh dengan menggunakan operasi komplemen dari Gambar a	(c) Gabungan antara Gambar a dan b

Gambar 25. Penerapan dari operasi himpunan

Persamaan gabungan dua himpunan skala abu-abu A dan B yang ditunjukan dalam Persaman (2.22) dengan jumlah elemen yang sama didefinisikan sebagai himpunan yang dipahami bahwa operasi maks diterapkan pada pasangan elemen yang bersesuaian.

$$A \cup B = \{\max(a, b) \,|a \in A, b \in B\} \qquad (2.22)$$

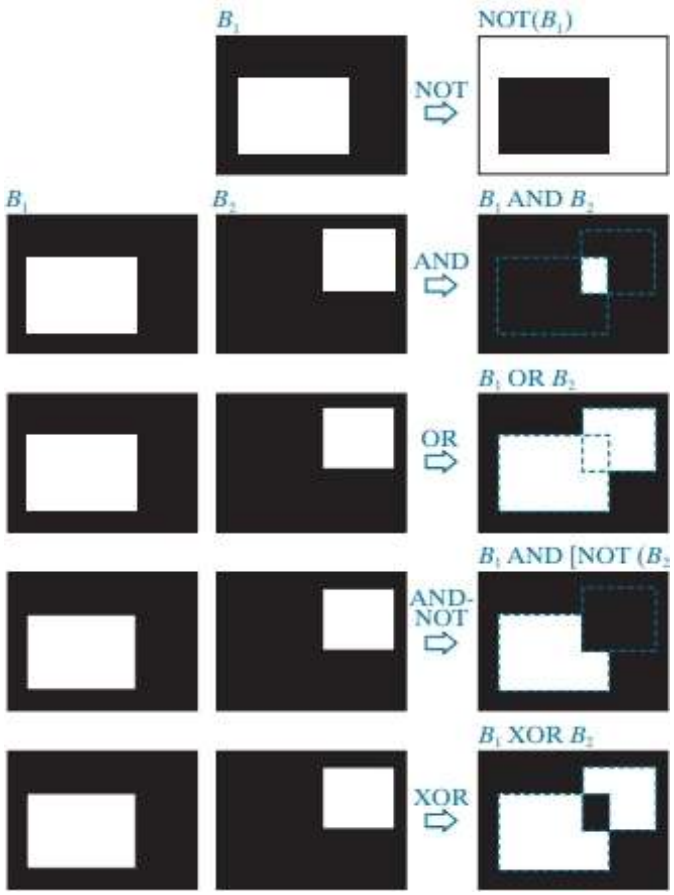
Operasi logika berhubungan dengan variabel dan ekspresi *TRUE* (biasanya dilambangkan dengan 1) dan *FALSE* (biasanya dilambangkan dengan 0). Hal ini berarti bahwa gambar biner yang terdiri dari piksel latar depan (bernilai 1), dan latar belakang terdiri dari piksel bernilai 0.

Operator logika dapat didefinisikan dalam Tabel kebenaran, seperti yang ditunjukkan dalam Tabel 8 untuk dua variabel logika a dan b.

Tabel 8. Tabel kebenaran yang mendefinisikan operator logika.

a	b	b AND b	a OR b	NOT(a)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Gambar 26 mengilustrasikan operasi logika yang melibatkan piksel latar depan (putih). Hitam mewakili biner 0 dan putih mewakili biner 1, Garis putus-putus ditampilkan hanya untuk referensi.



Gambar 26. Ilustrasi operasi logika.

e. Operasi spasial

Operasi spasial diklasifikasikan dalam 3 kategori

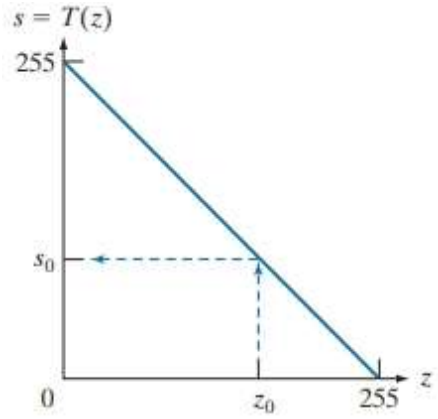
1. Operasi *single*-piksel

Operasi paling sederhana yang dilakukan pada gambar digital adalah mengubah intensitas pikselnya satu per satu menggunakan fungsi transformasi, T, diberikan dalam Persamaan (2.23):

$$s = T(z) \tag{2.23}$$

Z adalah intensitas piksel pada gambar asli dan s adalah intensitas yang dipetakan dari piksel yang bersangkutan pada gambar yang diproses. Gambar 27

menunjukkan transformasi yang digunakan untuk mendapatkan citra negatif (kadang-kadang disebut komplemen) dari gambar 8-bit. Transformasi ini dapat digunakan untuk memperoleh citra negatif pada Gambar 25, daripada menggunakan himpunan.



Gambar 27. Fungsi transformasi intensitas digunakan untuk mendapatkan padanan digital dari negatif fotografis dari Gambar 8-bit.

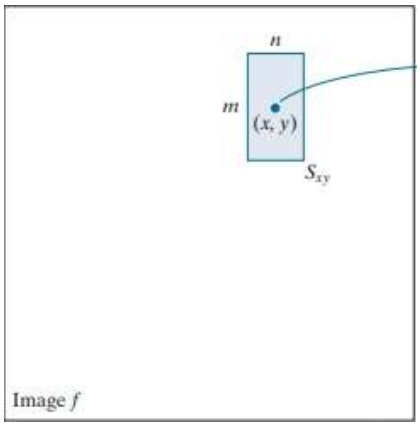
2. Operasi *neighborhood*

Misal S_{xy}, menyatakan himpunan koordinat suatu *neighborhood* yang berpusat pada titik sembarang (x, y) pada gambar f. Pemrosesan *neighborhood* menghasilkan piksel yang sesuai pada koordinat yang sama dalam gambar keluaran g. Sehingga nilai piksel tersebut ditentukan oleh operasi tertentu pada *neighborhood* piksel dalam gambar masukan dengan koordinat dalam himpunan S. Misalnya, operasi yang ditentukan adalah menghitung nilai rata-rata piksel dalam *neighborhood* persegi panjang berukuran $m \times n$ yang berpusat pada (x,y). Koordinat piksel di wilayah ini adalah elemen himpunan S. Gambar 28(a) dan (b) mengilustrasikan prosesnya. Operasi rata-rata ini dapat dinyatakan dalam Persamaan (2.24).

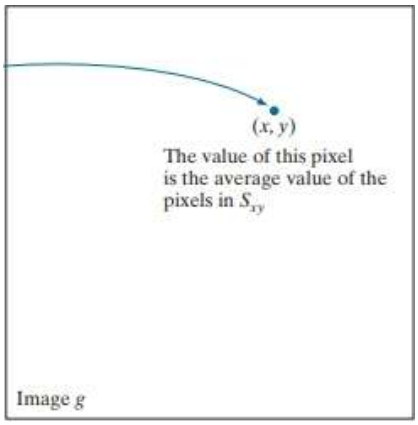
$$g(x,y) = \frac{1}{mn} \sum_{(r,c) \subseteq S_{xy}} f(r,c) \tag{2.24}$$

dengan r dan c adalah koordinat baris dan kolom dari piksel-piksel yang koordinatnya berada pada himpunan S_{xy}. Gambar g dibuat dengan memvariasikan koordinat (x, y) sehingga pusat *neighborhood* berpindah dari piksel ke piksel pada gambar f, dan kemudian mengulangi operasi *neighborhood* di setiap lokasi baru.

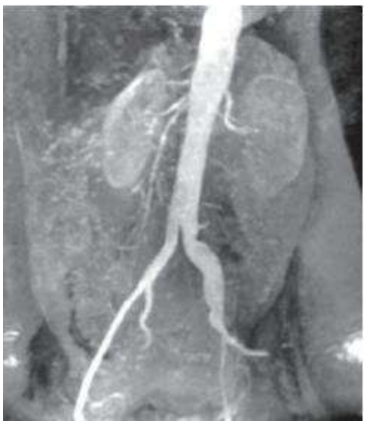
Misal Gambar 28(d) dibuat dengan cara mengacu pada proses pembentukan gambar g menggunakan *neighborhood* berukuran 41 x 41, Efek akhirnya adalah melakukan pengaburan lokal pada gambar asli. Jenis proses ini digunakan untuk menghilangkan detail kecil dan kemudian membuat "gumpalan" sesuai dengan wilayah terbesar dari suatu gambar.



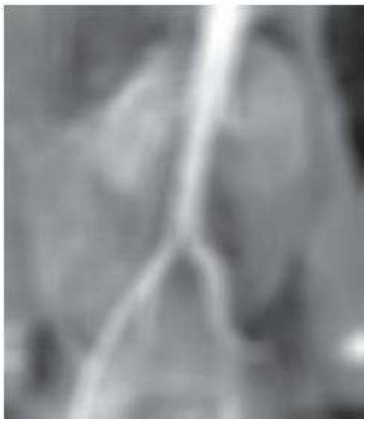
(a) Gambar asli f dengan sebuah kotak yang menandai *neighborhood* S_{xy}



(b) Nilai piksel di (x, y) dalam gambar g adalah rata-rata dari nilai piksel dalam $neighborhood\ S_{xy}$ di gambar f



(c) Gambar angiogram aorta



(d) Penerapan operasi lingkungan pada gambar f untuk menghasilkan gambar g.

Gambar 28. Rata-rata lokal menggunakan pemrosesan *neigborhood*. (Gambar asli milik Dr. Thomas R. Gest, Divisi Ilmu Anatomi, Fakultas Kedokteran Universitas Michigan.)

3. Transformasi geometrik

Transformasi geometris digunakan untuk memodifikasi susunan spasial piksel dalam suatu gambar. Transformasi geometri citra digital terdiri dari dua operasi dasar:

1. Transformasi spasial koordinat.
2. Interpolasi intensitas yang memberikan nilai intensitas pada piksel yang ditransformasikan secara spasial.

Transformasi koordinat dinyatakan pada Persamaan (2.25).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.25}$$

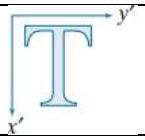
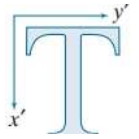
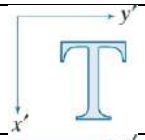
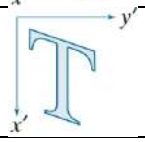
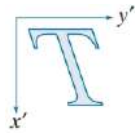
dengan (x, y) adalah koordinat piksel pada gambar asli dan (x‘,y‘) adalah koordinat piksel yang sesuai dari gambar yang diubah. Misalnya, transformasi (x‘,y‘) = (x/2,y/2) mengecilkan gambar asli menjadi setengah ukurannya di kedua arah spasial.

Transformasi affine adalah jenis transformasi geometris yang mempertahankan *collinearity* (garis lurus tetap lurus), *parallelism* (garis paralel tetap paralel), dan rasio jarak pada garis yang sama. Transformasi ini mencakup operasi seperti skala (*scaling*), rotasi (*rotation*), translasi (*translation*), dan pergeseran (*shearing*).. Persamaan (2.25) dapat digunakan untuk menyatakan transformasi affine, kecuali translasi, yang memerlukan penambahan vektor 2-D konstan pada ruas kanan persamaan. Koordinat homogen digunakan untuk mengekspresikan semua empat transformasi affine menggunakan matriks 3×3 tunggal dalam persamaan (2.26)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.26}$$

Transformasi affine dapat menskalakan, memutar, menerjemahkan, atau memperhalus suatu gambar, bergantung pada nilai yang dipilih untuk elemen matriks A. Tabel 9 menunjukkan nilai matriks yang digunakan untuk mengimplementasikan transformasi affine.

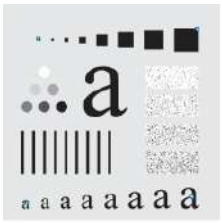
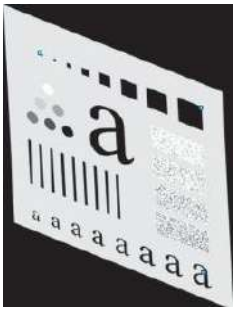
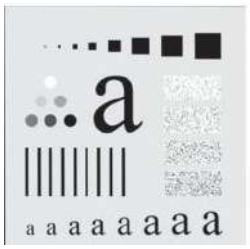
Tabel 9. Transformasi affine berdasarkan Persamaan (2.26).

Nama Transformasi	Matriks Affine, A	Persamaan kordinat	Contoh
Identitas	$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$	$x' = x$ $y' = y$	
Penskalaan/Refleksi (Untuk refleksi, atur satu faktor penskalaan ke -1 dan faktor penskalaan lainnya ke 0)	$\begin{bmatrix}c_x & 0 & 0\\0 & c_y & 0\\0 & 0 & 1\end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotasi	$\begin{bmatrix}\cos\theta & -\sin\theta & 0\\\sin\theta & \cos\theta & 0\\0 & 0 & 1\end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translasi	$\begin{bmatrix}1 & 0 & t_x\\0 & 1 & t_y\\0 & 0 & 1\end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertikal)	$\begin{bmatrix}1 & s_v & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix}1 & 0 & 0\\s_h & 1 & 0\\0 & 0 & 1\end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

4. Registrasi gambar

Registrasi gambar adalah aplikasi penting dalam pemrosesan gambar digital yang digunakan untuk menyelaraskan dua atau lebih gambar dari pemandangan yang sama. Registrasi gambar menyediakan gambar masukan dan gambar referensi. Tujuannya adalah untuk mentransformasikan citra masukan secara geometris sehingga menghasilkan citra keluaran yang sejajar atau *registerd* dengan citra acuan.

Contoh registrasi gambar diilustrasikan pada Gambar 29 mencakup penyelarasan dua gambar atau lebih yang diambil pada waktu yang hampir bersamaan, namun menggunakan sistem pencitraan yang berbeda, seperti pemindai *Magnetic Resonance Imaging* (MRI) dan pemindai *Positron Emission Tomography* (PET).

	
(a) Gambar referensi.	(b) Input (gambar yang terdistorsi secara geometris). Titik ikat yang sesuai ditampilkan sebagai kotak putih kecil di dekat sudut.
	
(c) Gambar (keluaran) terdaftar (perhatikan kesalahan pada batasnya).	(d) Perbedaan antara (a) dan (c), menunjukkan lebih banyak kesalahan registrasi.

Gambar 29. Registrasi gambar.

Estimasi transformasi antara gambar masukan dan gambar referensi dengan empat titik ikat sebagai referensi merupakan tantangan dalam pemodelan. Pendekatan umum menggunakan model interpolasi *bilinear*, seperti yang dijelaskan dalam Persamaan (2.27) dan (2.28).

$$x = c_1 v + c_2 w + c_3 vw + c_4 \quad (2.27)$$

$$y = c_5 v + c_6 w + c_7 vw + c_8 \quad (2.28)$$

Selama tahap estimasi, (v, w) dan (x, y) merupakan koordinat titik ikat pada gambar masukan dan gambar referensi. Jika kedua gambar (*input* dan referensi) mempunyai empat pasang titik ikat yang bersesuaian maka empat pasang titik ikat bersesuaian tersebut dapat ditulis sebagai delapan persamaan menggunakan Persamaan (2.27) dan (2.28) dan menggunakan keduanya untuk menyelesaikan delapan koefisien yang tidak diketahui, c_1 hingga c_8. Jika titik ikat dipilih dengan benar, gambar baru

ini harus terdaftar dalam gambar referensi, sesuai dengan akurasi model pendekatan bilinear.

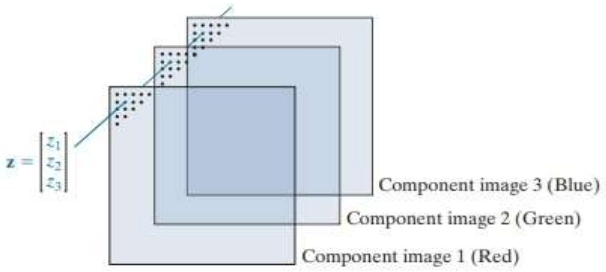
5. Operasi vektor dan matriks

Pemrosesan gambar multispektral adalah area umum di mana operasi vektor dan matriks digunakan secara rutin. Setiap piksel pada gambar RGB memiliki tiga komponen, yang dapat disusun dalam bentuk vektor kolom pada Persamaan (2.29).

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \tag{2.29}$$

z_1 adalah intensitas piksel pada gambar merah, dan z_2, dan z_3 adalah intensitas piksel yang sesuai pada gambar hijau dan biru. Gambar berwarna RGB berukuran $M \times N$ dapat diwakili oleh tiga komponen gambar seperti yang diilustrasikan pada Gambar 30. Total vektor $M \times N$ berukuran 3×1 ditunjukan pada Persamaan (2.30).

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \tag{2.30}$$



Gambar 30. Membentuk vektor dari nilai piksel yang sesuai pada tiga gambar komponen RGB.

6. Transformasi gambar

Tugas pemrosesan gambar paling baik dirumuskan dengan mentransformasikan gambar masukan, membawa tugas tertentu dalam domain transformasi, dan menerapkan transformasi invers untuk kembali ke domain spasial. Kelas transformasi linier 2-D yang sangat penting, dinotasikan T(u, v), dapat dinyatakan pada Persamaan (2.31).

$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) r(x,y,u,v) \tag{2.31}$$

dengan f(x, y) adalah gambar masukan, r(x,y,u,v) disebut *forward transformation kernel*, dan Persamaan. (2.31) dievaluasi untuk u = 0, 1, 2,..., M-1 dan v = 0, 1, 2,..., N-1. Sama seperti sebelumnya, x dan y adalah variabel spasial, sedangkan M dan N adalah dimensi baris dan kolom dari f. Variabel u dan v disebut variabel transformasi. T(u,v) disebut *forward transform* dari f(x, y). Diketahui T(u,v), dapat mengembalikan nilai f(x, y) menggunakan transformasi invers dari T(u,v) yang diberikan pada Persamaan (2.32):

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) s(x,y,u,v) \tag{2.32}$$

dengan $x = 0, 1, \ldots, M-1$ dan $y = 0, 1, \ldots, N-1, s(x,y,u,v)$ disebut *invers transformation kernel*. Secara bersama-sama, Persamaan (2.31) dan (2.32) disebut pasangan transformasi. Langkah-langkah dasar untuk melakukan pemrosesan gambar dalam domain transformasi *linier* ditunjukan pada Gambar 31. Pertama, citra masukan ditransformasikan, selanjutnya transformasi tersebut dimodifikasi dengan operasi yang telah ditentukan, dan kemudian citra keluaran diperoleh dengan menghitung *invers* dari transformasi yang dimodifikasi. Proses berpindah dari domain spasial ke domain transformasi, dan kemudian kembali ke domain spasial.

Gambar 31. Pendekatan umum untuk bekerja dalam domain transformasi linier.

Forward transformation kernel dapat dipisahkan yang ditunjukan dalam Persamaan (2.33).

$$r(x,y,u,v) = r_1(x,u) r_2(y,v) \tag{2.33}$$

Selain itu, kernel dikatakan simetris jika $r_1(x,u)$ secara fungsional sama dengan $r_1(y,v)$, diberikan pada Persamaan 2.34.

$$r(x,y,u,v) = r_1(x,u)r_1(y,v) \tag{2.34}$$

Sifat suatu transformasi ditentukan oleh kernelnya. Transformasi yang sangat penting dalam pengolahan citra digital adalah transformasi Fourier, yang mempunyai *forward* dan *invers* ditunjukan pada Persamaan (2.35) dan (2.36):

$$r(x,y,u,v) = e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2.35}$$

$$s(x,y,u,v) = \frac{1}{MN}e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2.36}$$

dengan j = √-1, sehingga kernel ini merupakan fungsi yang kompleks. Karena sebelumnya diganti menjadi formulasi transformasi umum dalam Persamaan. (2.31) dan (2.32) menghasilkan pasangan transformasi Fourier diskrit pada Persamaan (2.37):

$$T(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2.37}$$

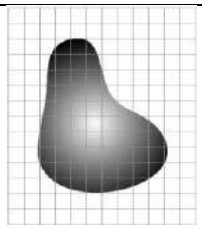
Persamaan (2.38) merupakan hasil dari subtitusi Persamaan (2.36) ke dalam Persamaan (2.32).

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} T(u,v)e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2.38}$$

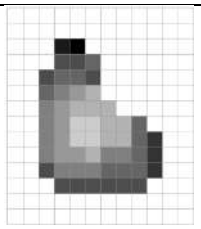
2.9 Citra Digital

Citra adalah representasi visual dalam dua dimensi. Secara perspektif matematika, citra adalah fungsi kontinu dari intensitas cahaya dalam ruang dua dimensi. Cahaya dari sumber pencahayaan mengenai objek, dan objek tersebut memantulkan

sebagian dari cahaya tersebut. Alat optik seperti mata manusia, kamera, pemindai dan sejenisnya digunakan untuk menangkap pantulan cahaya ini, sehingga menciptakan gambar objek yang dikenal sebagai citra (Munir, 2004). Langkah awal dalam menciptakan gambar digital adalah mengubah data kontinu dari penginderaan menjadi format digital. Proses ini terdiri dari dua tahap utama, yaitu pengambilan sampel dan kuantisasi. Gambar diubah menjadi bentuk digital, dilakukan dengan mengambil sampel fungsi dalam koordinat dan amplitudo. Nilai koordinat didigitalisasikan disebut sampling dan nilai amplitudo didigitalisasikan disebut kuantisasi. Konsep tersebut diilustrasikan pada Gambar 32 sebagai berikut:



. (a) Gambar kontinu.

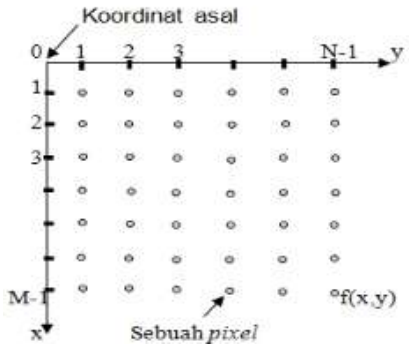


(b) Hasil dari sampling gambar dan kuantisasi

Gambar 32. Konsep citra digital

Gambar 32(a) menunjukkan gambar kontinu yang diproyeksikan ke bidang sensor array. Gambar 32(b) menunjukkan gambar setelah pengambilan sampel dan kuantisasi. Secara nyata bahwa kualitas gambar digital sebagian besar ditentukan oleh jumlah sampel dan tingkat keabuan yang digunakan dalam pengambilan sampel dan kuantisasi.

Suatu gambar dapat direpresentasikan sebagai suatu fungsi (x,y) berukuran M baris dan N kolom, x dan y adalah koordinat spasial, dan f pada koordinat (x,y) menunjukan intensitas dari gambar yang ditunjukan pada Gambar 33.



Gambar 33. Koordinat citra digital (Kumaseh dkk., 2013)

Secara matematis, citra digital dapat diungkapkan sebagai matriks seperti yang Persamaan (2.39) (Gozalez, 2009):

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.39)$$

Di mana:

M = jumlah baris $0 \leq y \leq N-1$

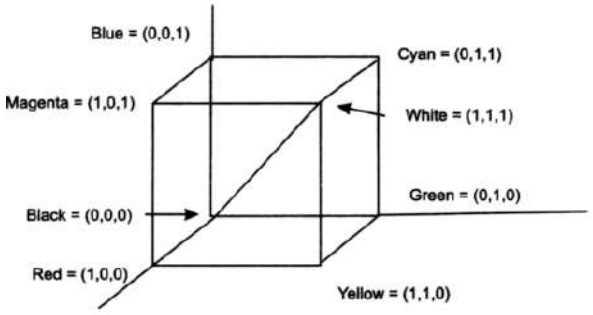
N = jumlah kolom $0 \leq x \leq M-1$

L = maksimal warna intensitas $0 \leq f(x,y) \leq L-1$

f(x,y) = derajat keabuan (*grey level*).

Mengekspresikan pengambilan sampel dan kuantisasi dalam istilah matematika yang lebih formal terkadang berguna. Misal Z dan R masing-masing menyatakan himpunan bilangan bulat real dan himpunan bilangan real. Proses pengambilan sampel direpresentasikan dengan mempartisi bidang xy menjadi sebuah grid. Koordinat pusat setiap grid adalah sepasang elemen dari hasil kali Kkrtesius Z^2(juga dilambangkan $Z \times Z$), yang merupakan himpunan semua pasangan elemen terurut (z_i, z_j), dengan z_i dan z_j merupakan bilangan bulat dari Z. Oleh karena itu, f(x, y) adalah suatu citra digital jika (x, y) adalah bilangan bulat dari Z^2 dan f adalah fungsi yang memberikan nilai tingkat keabuan (yaitu bilangan real dari himpunan bilangan real, R) pada setiap pasangan koordinat yang berbeda (x, y) (Gozalez, 2009).

Citra berwarna, setiap titik memiliki warna khusus yang terbentuk melalui kombinasi tiga warna dasar, yaitu merah, hijau, dan biru. Format citra ini sering disebut sebagai citra RGB (merah-hijau-biru). Masing-masing warna dasar memiliki intensitas dengan kisaran nilai maksimum 255 (8 bit), sehingga jumlah kemungkinan warna yang dapat direpresentasikan adalah 255x255x255 atau setara dengan 16.581.375 warna (Septiaji dan Firdausy, 2018). Skala abu-abu mengikuti garis dari hitam ke putih seperti yang ditampilkan pada Gambar 34.



Gambar 34. Diagram model warna RGB

Mengubah gambar dari warna RGB ke skala abu-abu, gunakan Persamaan (2.40):

$$gray\ scale\ intensity = 0{,}299R + 0{,}587G + 0{,}114B \qquad (2.40)$$

atau persamaan umum untuk mengkonversi dari warna RGB dengan menggunakan rata-rata seperti Persamaan (2.41).

$$gray\ scale\ intensity = 0{,}333R + 0{,}333G + 0{,}333B \qquad (2.41)$$

Sebuah piksel p pada koordinat (x, y) mempunyai dua *neighbors* horizontal dan dua vertikal yang koordinatnya diberikan pada Persamaan (2.42).

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \qquad (2.42)$$

Kumpulan piksel ini, disebut 4 *neighbors* p, dilambangkan dengan $N_4(p)$.

Keempat *neighbors* diagonal p mempunyai koordinat yang ditunjukkan pada Persamaan (2.43) dilambangkan dengan $N_D(p)$.

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1) \qquad (2.43)$$

Neighbors diagonal bersama dengan 4 tetangga disebut 8 tetangga dari p, dilambangkan dengan $N_8(p)$ ditunjukkan pada Persamaan (2.44),

$$\mathrm{N_8(p)} = \mathrm{N_4(p)} + \mathrm{N_D(p)} \tag{2.44}$$

Neighbors dikatakan tertutup apabila di dalamnya terdapat p. Jika tidak, *neighbors* tersebut dikatakan terbuka.

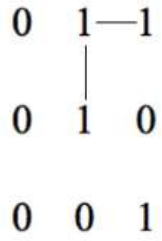
$(x-1, y+1)$	$(x, y+1)$	$(x+1, y+1)$
$(x-1, y)$	(x, y)	$(x+1, y)$
$(x-1, y-1)$	$(x, y-1)$	$(x+1, y-1)$

Misalkan V adalah himpunan nilai intensitas yang digunakan untuk mendefinisikan ketetanggaan. Citra biner, V ={1} jika mengacu pada kedekatan piksel dengan nilai 1, Citra skala abu-abu, idenya sama tetapi himpunan V biasanya berisi lebih banyak elemen.

Misalnya, dalam kedekatan piksel dengan nilai intensitas berkisar antara 0 hingga 255, himpunan V dapat berupa subset mana pun dari 256 nilai tersebut.

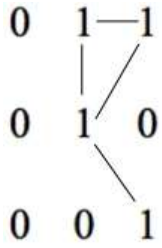
Ada tiga jenis *adjacent*:

1. 4-*adjacency*: Dua piksel p dan q dengan nilai dari V berdekatan 4 jika q ada di himpunan $\mathrm{N_4(p)}$. Gambar 35 merupakan contoh dari 4-*adjacency*.



Gambar 35. Contoh dari 4-*adjacency*.

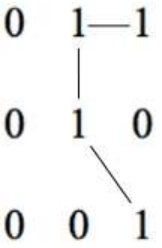
2. 8-*adjacency*: Dua piksel p dan q dengan nilai dari V bertetangga 8 jika q ada di himpunan N8(p). Gambar 36 merupakan contoh dari 8-*adjacency*.



Gambar 36. Contoh dari 8-*adjacency*.

3. m-*adjacency* (Kedekatan Campuran): Dua piksel p dan q dengan nilai dari V berdekatan dengan m jika,
   a. q ada di $N_4(p)$, atau
   b. q ada di $N_D(p)$ dan himpunan $N_4(p) \cap N_4(q)$ tidak memiliki piksel yang berasal dari V.

Contoh dari m-*adjacency* ditampilkan pada Gambar 37 sebagai berikut:



Gambar 37. Contoh dari m-adjacency.

konektivitas adalah konsep penting dalam pemrosesan citra digital untuk menghubungkan objek dan komponen wilayah. Misalkan S mewakili subset piksel dalam sebuah gambar. Dua piksel p dan q dikatakan terhubung di S jika terdapat jalur di antara keduanya yang seluruhnya terdiri dari piksel-piksel di S. Setiap piksel p adalah S, himpunan piksel yang terhubung dengannya di S disebut komponen terhubung dari S. Jika hanya mempunyai satu komponen, dan komponen tersebut terhubung, maka S disebut himpunan terhubung. Berdasarkan adjacency, ada tiga bentuk konektivitas sebagai berikut:

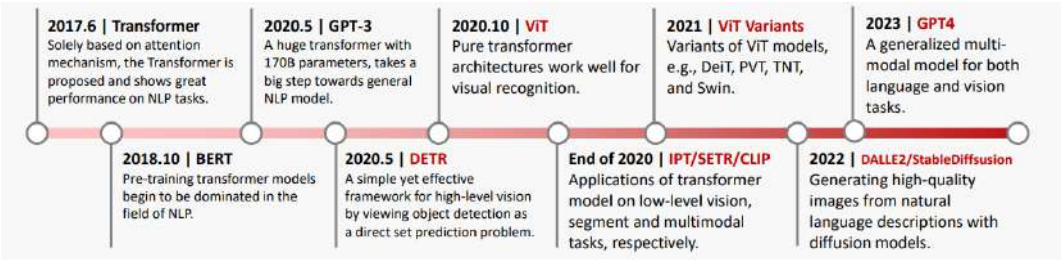
1. 4-*connectivity*: Jika dua piksel atau lebih bersebelahan 4 satu sama lain, keduanya dikatakan terhubung 4.
2. 8-*connectivity*: Jika dua piksel atau lebih bersebelahan 8 satu sama lain, keduanya dikatakan terhubung 8.

3. m-*connectivity*: Jika dua piksel atau lebih berdekatan satu sama lain, maka piksel-piksel tersebut dikatakan m-terkoneksi.

Subset R piksel dalam suatu gambar disebut Wilayah gambar jika R adalah himpunan terhubung, sedangkan batas wilayah R adalah himpunan piksel-piksel di wilayah yang mempunyai satu atau lebih tetangga yang tidak berada di R.

2.10 *Vision Transformers* (ViT)

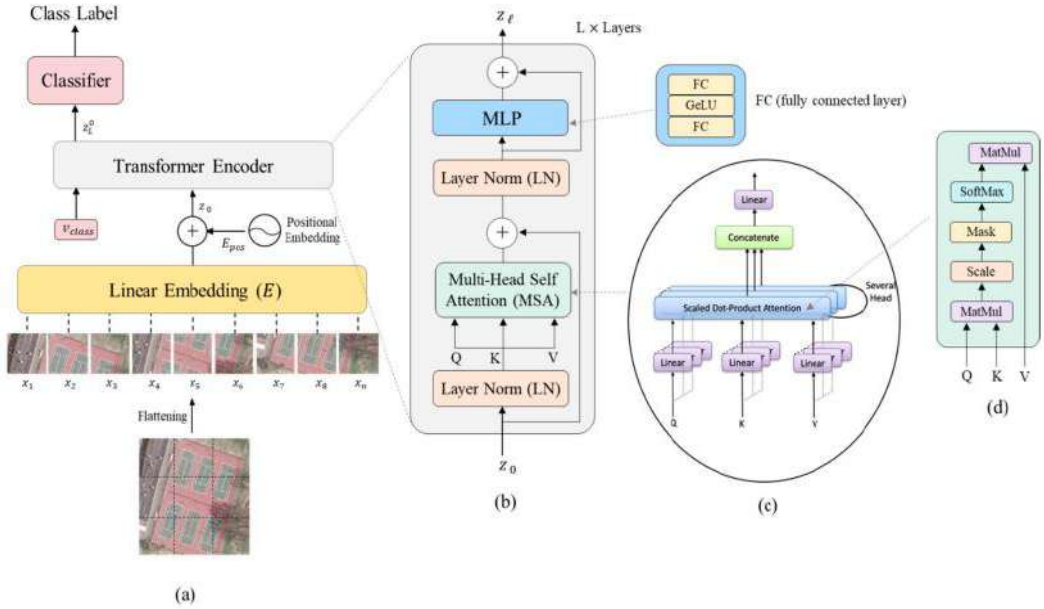
Beberapa tahun terakhir, CNN telah mencapai hasil terbaik dalam kinerja dalam tugas-tugas visi komputer. *Vision transformers* merupakan arsitektur pemrosesan gambar yang merujuk pada dasar arsitektur *transformers*. Awalnya *transformers* digunakan dalam NLP (Vaswani dkk., 2017). Terinspirasi oleh keberhasilan NLP, beberapa penelitian mencoba menggabungkan arsitektur mirip CNN dengan *self-attention* (Wang dkk., 2018; Carion dkk., 2020). Baru-baru ini, model ViT yang diusulkan oleh Dosovitskiy dan rekan-rekannya memperkenalkan sebuah makalah penelitian yang diterbitkan sebagai makalah konferensi pada ICLR 2021 berjudul "An Image is Worth 16*16 Words: Transformers for Image Recognition at Scale" telah mencapai kinerja *state-of-the-art* pada benchmark pengenalan gambar. Perkembangan transformers diilustrasikan pada Gambar 38 sebagai berikut:



Gambar 38. Perkembangan dari transformers (Han dkk., 2022)

Menurut Dosovitsky. (2021) transformers menunjukkan kemampuan untuk berkembang lebih baik dibandingkan dengan CNN. Pada saat model yang lebih besar dilatih dengan dataset yang lebih besar, maka *vision transformers* berhasil

melampaui ResNets dengan perbedaan yang besar. Arsitektur dari *vision transformers* ditunjukan pada Gambar 39.



Gambar 39. Arsitektur dari transformers (Bazi dkk., 2021)

Model transformer menggunakan mekanisme *self-attention*, tidak praktis untuk melakukan proses input piksel per piksel karena jumlahnya sangat besar. Setiap piksel harus berinteraksi dengan setiap piksel lainnya, menghasilkan jumlah yang sangat besar (Dosovitskiy, 2021).

Transformer standar menerima urutan token *embeddings* 1D sebagai masukan. Gambar diubah kembali dari x ∈ $\mathbb{R}^{WxCxH}$ menjadi urutan potongan gambar 2D yang diratakan $x_p \in \mathbb{R}^{Nx(P^2.C)}$, dengan (H, W) adalah resolusi gambar asli, C adalah jumlah *channel*, dan (P, P) adalah resolusi setiap potongan gambar. Penentuan banyaknya *patch* dijelaskan pada Persamaan (2.45)

$$N = \frac{HW}{P^2} \tag{2.45}$$

N adalah jumlah potongan yang dihasilkan, yang berfungsi sebagai panjang urutan masukan efektif untuk transformer. Transformer menggunakan ukuran vektor laten konstan D melalui semua lapisan. Potongan-potongan ini diratakan dan dipetakan ke dimensi D dengan proyeksi linear yang dapat dilatih (Dosovitskiy, 2021).

Seperti halnya dengan token [class] pada BERT, sebuah *embedding* yang dapat dipelajari yang diterapkan pada urutan potongan gambar. Sebelum memberikan urutan potongan ke *encoder*, potongan-potongan tersebut diproyeksikan secara linear ke dalam vektor dengan dimensi model d menggunakan matriks E (*embedding*) yang dapat dipelajari (Bazi, dkk., 2021). Representasi yang telah di *embedding* kemudian digabungkan bersama dengan token klasifikasi yang dapat dipelajari x_{class} yang diperlukan untuk menjalankan tugas klasifikasi. Potongan gambar yang sudah diembedding direpresentasikan oleh *transformers* sebagai kumpulan potongan tanpa pemahaman tentang urutan mereka. Mempertahankan susunan spasial potongan seperti dalam gambar asli, informasi posisi E_{pos} di-encode dan ditambahkan ke representasi potongan (Bazi, dkk., 2021). Urutan potongan yang di *embedding* hasilnya dengan token z_0 diberikan pada Persamaan (2.46) dan Persamaan (2.47).

$$z_0 = \left[x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^N E\right] + E_{pos}, E \in \mathbb{R}^{(P^2.C)xd}, E_{pos} \in \mathbb{R}^{(N+1)xd} \tag{2.46}$$

$$E_{(pos,i)} = \begin{cases} \sin\left(\dfrac{pos}{10000^{\left(\frac{i}{d}\right)}}\right) jika\ i\ genap \\ \cos\left(\dfrac{pos}{10000^{\left(\frac{i-1}{d}\right)}}\right) lainnya \end{cases} \tag{2.47}$$

Selanjutnya urutan *patches embedding* yang dihasilkan z_0 diberikan kepada transformers *encoder*. Seperti yang ditunjukkan dalam Gambar 39b, *encoder* terdiri dari L lapisan yang identik. Setiap lapisan memiliki dua subkomponen utama: (1) blok *multihead self-attention* (MSA), dan (2) *blok dense feed-forward fully connected* (MLP). Blok terakhir terdiri dari dua lapisan dense dengan aktivasi GELU di antaranya. Setiap dari kedua subkomponen encoder menggunakan koneksi lompatan residual dan didahului oleh lapisan normalisasi (LN) (Bazi dkk., 2021).

$$z_l' = MSA(LN\left(z_{(l-1)}\right) + z_{(l-1)}, l = 1, \ldots, L \tag{2.48}$$

$$z_l = MLP\big(LN(z'_l)\big) + z'_l, l = 1, \dots, L \quad (2.49)$$

$$y = LN(z_L^0) \quad (2.50)$$

Pada gambar 39b menjelaskan Blok MSA dalam *encoder* transformer. Komponen utama yang bertanggung jawab untuk menentukan sejauh mana pentingnya sebuah *patches embedding* dalam urutan dibandingkan dengan *embedding* lainnya. Blok ini terdiri dari empat lapisan, yaitu: lapisan linear, lapisan *self-attention*, *concatenation layer* yang menggabungkan keluaran dari beberapa *head attention*, dan lapisan linear terakhir (Bazi dkk., 2021).

Pada dasarnya, *attention* adalah cara untuk menentukan seberapa penting setiap bagian dari data dalam sebuah urutan. *Attention* dapat direpresentasikan dengan bobot *attention* yang dihitung melalui jumlah bobot dari nilai-nilai dalam urutan z. *Head self-attention* (SA) mempelajari bobot attention dengan menghitung perkalian *dot-product* antara *query-key-value* yang sudah dilakukakan *scaling*. Setiap elemen dalam urutan input menghasilkan tiga nilai: *query* (Q), *key* (K), dan *value* (V) dengan menggunakan tiga matriks yang telah dipelajari, yaitu U_{QKV}. Untuk mengetahui seberapa penting satu elemen dibandingkan elemen lain, dot product dihitung antara vektor Q dari satu elemen dengan vektor K dari elemen lainnya. Hasil dari dot produt vektor Q dan vektor K menentukan pentingnya relatif potongan dalam urutan. Selanjutnya dilakukan *scaling* dan dimasukkan ke dalam fungsi softmax. Proses scaling pada operasi perkalian *dot-product* yang dilakukan oleh blok SA mirip dengan perkalian *dot-product* standar, tetapi menggabungkan dimensi *key* (D_h) (Bazi, dkk., 2021). Akhirnya, nilai vektor masing-masing *patches embedding* dikalikan dengan keluaran dari fungsi *softmax* untuk menentukan potongan dengan skor *attention* tinggi. Seluruh operasi diberikan oleh Persamaan-Persamaan (2.57), (2.58), dan (2.59):

$$[Q, K, V] = zU_{QKV}, U_{QKV} \in \mathbb{R}^{dxD_h} \quad (2.51)$$

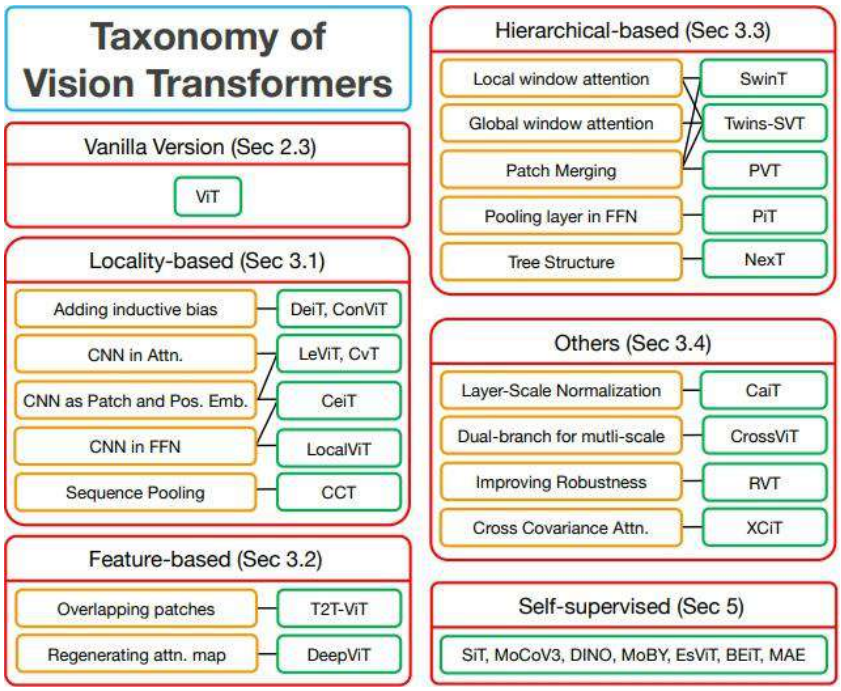
$$A = softmax\left(\frac{QK^T}{\sqrt{(D_h)}}\right), A \in \mathbb{R}^{NxN} \quad (2.52)$$

$$SA(z) = A.V \quad (2.53)$$

Hasil dari semua *head attention* digabungkan bersama dan kemudian diproyeksikan melalui lapisan *feed-forward* dengan bobot yang dapat dipelajari U_{msa} ke dimensi yang diinginkan (Bazi dkk., 2021). Operasi ini diungkapkan melalui Persamaan (2.54):

$$MSA(z) = [SA_1(z); SA_2(z); \dots; SA_k(z)]U_{msa}, U_{msa} \in \mathbb{R}^{k.D_h xD} \qquad (2.54)$$

Gambar 40 menggambarkan klasifikasi model-model *vision transformer* ke dalam tiga arah utama. Pertama, Bagian 3.1 memperkenalkan model berbasis lokalitas yang memasukkan lokalitas ke dalam arsitektur. Kemudian, Bagian 3.2 memperkenalkan model berbasis fitur yang bertujuan untuk variasi representasi fitur. Terakhir, Bagian 3.3 menyajikan model berbasis hierarki yang mengurangi ukuran fitur lapisan demi lapisan untuk meningkatkan kecepatan inferensi. Beberapa arsitektur yang tidak masuk ke dalam kategori-kategori di atas dibahas dalam Bagian 3.4. Perlu diingat bahwa model-model ini ditempatkan dalam kategori tertentu, tetapi kategori-kategori tersebut tidak eksklusif satu sama lain (Ruan dkk., 2022).



Gambar 40. Taksonomi dari *vision transformers* (Ruan dkk., 2022)

2.11 *A Robustly Optimized BERT Pretraining Approach* (RoBERTa)

A Robustly Optimized BERT Pretraining Approach (RoBERTa) merupakan varian BERT dirilis oleh Liu dkk. (2019) dengan arsitektur yang sama dengan BERT, tetapi dengan kinerja yang lebih baik. RoBERTa dilatih dengan menggunakan dynamic masking, kalimat lengkap tanpa *next sentences prediction* (NSP) *loss*, *mini-batch* besar, dan penggunaan *Byte-Pair Encoding* (BPE) berdasarkan byte yang lebih besar (Liu dkk., 2019).

A. *Dynamic vs static masking*

Model BERT menggunakan metode *static masking*, di mana pola masking token tetap tidak berubah. Sebaliknya, RoBERTa menerapkan *dynamic masking*, yang mengakibatkan perubahan posisi token pada *mask* selama pelatihan. Penggunaan *masking* dinamis ini menghasilkan peningkatan dalam tingkat keacakkan data dan, akibatnya, meningkatkan kemampuan model untuk belajar (Gao dkk., 2022). Gambar 41 merupakan perbandingan antara *static masking* dan *dynamic masking.*

Masking	SQuAD 2.0	MNLI-m	SST-2
reference	76.3	84.3	92.8
Our reimplementation:			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

Gambar 41. Perbandingan antara *static masking* dengan *dynamic masking* (Liu dkk., 2019)

B. *Full santences*

Setiap masukan diisi dengan urutan kalimat penuh yang diambil dari satu atau beberapa dokumen secara berurutan, dengan total panjang maksimal 512 token. Masukan dapat melintasi batas dokumen. Ketika mencapai akhir satu dokumen, kalimat mulai diambil dari dokumen berikutnya, dan menghapus NSP *loss* (Liu dkk., 2019). Gambar 42 menjelaskan perbandingan dengan menggunakan NSP dan tidak menggunakan NSP.

Model	SQuAD 1.1/2.0	MNLI-m	SST-2	RACE
Our reimplementation (with NSP loss):				
SEGMENT-PAIR	90.4/78.7	84.0	92.9	64.2
SENTENCE-PAIR	88.7/76.2	82.9	92.1	63.0
Our reimplementation (without NSP loss):				
FULL-SENTENCES	90.4/79.1	84.7	92.5	64.8
DOC-SENTENCES	90.6/79.7	84.7	92.7	65.6
$\text{BERT}_{\text{BASE}}$	88.5/76.3	84.3	92.8	64.3
$\text{XLNet}_{\text{BASE}}$ (K = 7)	–/81.3	85.8	92.7	66.1
$\text{XLNet}_{\text{BASE}}$ (K = 6)	–/81.0	85.6	93.4	66.7

Gambar 42. Perbandingan dengan menggunakan NSP (Liu dkk., 2019)

C. *Training* dengan *batch* yang besar

Penelitian yang dilakukan oleh Ott. dkk (2018) pada *neural machine translation* telah menunjukkan bahwa menggunakan *mini-batch* yang sangat besar dapat meningkatkan kecepatan optimisasi serta kinerja dalam tugas akhir, asalkan tingkat pembelajaran disesuaikan dengan baik.

Awalnya, Devlin dkk. (2019) melatih BERTBASE selama 1 juta langkah dengan 256 urutan dalam setiap batch. Biaya komputasi ini setara dengan melatih selama 125 ribu langkah dengan 2 ribu urutan dalam setiap batch, atau 31 ribu langkah dengan 8 ribu urutan dalam setiap batch melalui akumulasi gradien. Gambar 43 merupakan perbandingan BERT dan RoBERTa menggunakan *batch size*, *steps* dan *learning rate*.

bsz	steps	lr	ppl	MNLI-m	SST-2
256	1M	1e-4	3.99	84.7	92.7
2K	125K	7e-4	**3.68**	**85.2**	**92.9**
8K	31K	1e-3	3.77	84.6	92.8

Gambar 43. Perbandingan BERT dan RoBERTa menggunakan *batch size*, *steps*, dan *learning rate* (Liu dkk., 2019).

D. *Byte-Pair Encoding* (BPE)

Byte-Pair Encoding adalah suatu metode yang menggabungkan representasi karakter dan kata untuk mengatasi kosakata besar yang sering ditemui dalam teks bahasa alami (Sennrich dkk., 2016). *Byte-Pair Encoding* mengandalkan subkata sebagai unit dasar yang diperoleh melalui analisis statistik pada korpus latihan.

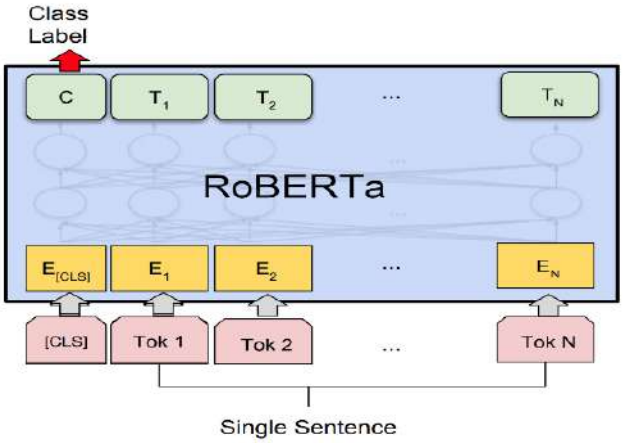
Ukuran kosakata BPE berkisar antara 10K-100K unit subkata. Namun, karakter Unicode dapat membentuk sebagian besar kosakata ini saat memodelkan korpus yang besar dan beragam, seperti yang dipertimbangkan dalam penelitian ini. Radford dkk. (2019) memperkenalkan implementasi BPE yang pintar dengan menggunakan byte sebagai unit subkata dasar daripada karakter *unicode*. Menggunakan *byte* memungkinkan untuk mempelajari kosakata subkata yang memiliki ukuran yang cukup besar (50K unit) yang masih dapat mewakili teks masukan tanpa memperkenalkan token "tidak dikenal."

Implementasi BERT menggunakan kosakata BPE tingkat karakter dengan ukuran 30K, yang dipelajari setelah melakukan pra-pemrosesan masukan dengan aturan tokenisasi heuristik. Namun, mengikuti pendekatan Radford dkk. (2019) melatih BERT dengan kosakata BPE tingkat byte yang lebih besar berisi 50K unit subkata, tanpa perlu melakukan pra-pemrosesan atau tokenisasi tambahan pada masukan. Ini mengakibatkan penambahan sekitar 15 juta dan 20 juta parameter tambahan untuk BERTBASE dan BERTLARGE. Gambar 44 merupakan perbandingan model dari *natural language processing* pada dataset.

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Gambar 44. Perbandingan dataset pada berbagai model (Liu dkk., 2019).

RoBERTa mendapatkan hasil yang lebih baik daripada BERT pada dataset *The General Language Understanding Evaluation* (GLUE), *The ReAding Comprehension from Examinations* (RACE), dan *The Standford Question Answering Dataset* (SquAD) tanpa *finetuning* multi-tugas untuk GLUE atau data tambahan untuk SquAD (Liu dkk., 2019). Gambar 45 menjelaskan arsitektur dari ROBERTa.



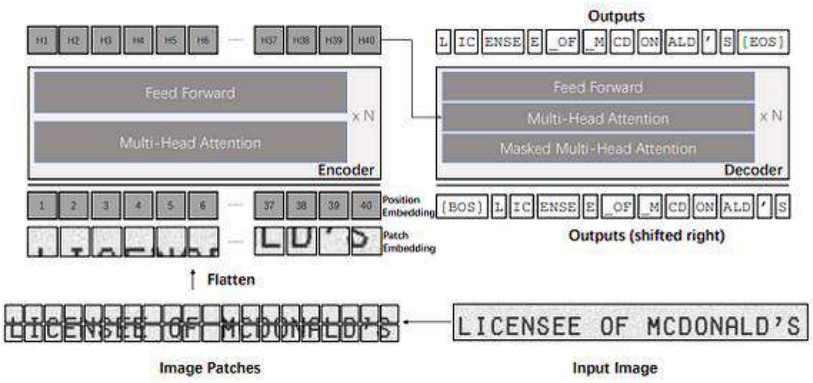
Gambar 45. Arsitektur dari RoBERTa (Khusuma dkk., 2023)

RoBERTa menerima kalimat sebagai masukan, dan setelah itu mengubahnya menjadi token untuk membuat masukan yang dapat digunakan oleh model. Tiga jenis masukan yang digunakan oleh RoBERTa termasuk *input_ids, attention_mask,* dan *token_type_ids*. Input_ids merupakan representasi numerik dari setiap token. Token [CLS] dimasukkan pada awal setiap rangkaian token untuk mengidentifikasi klasifikasi, sedangkan token [SEP] ditempatkan pada bagian akhir setiap rangkaian token. *Attention_mask* adalah representasi biner yang menunjukkan apakah suatu token adalah padding. Jika panjang urutan token kurang dari urutan terpanjang, maka padding akan ditambahkan ke urutan token tersebut. Penambahan *padding* ini disesuaikan dengan batasan jumlah maksimum token yang dapat digunakan oleh model RoBERTa, yaitu 512 token. Selanjutnya, terdapat *token_type_ids*, yang juga merupakan representasi biner untuk menunjukkan apakah dua kalimat merupakan pasangan kalimat. Biasanya, *token_type_ids* diperlukan dalam tugas tanya-jawab yang menggunakan input berupa pasangan kalimat. Input kemudian diproses oleh model RoBERTa yang terdiri dari 12 lapisan *encoder* RoBERTa, mengubah input menjadi bentuk vektor *embedding* yang melibatkan *embedding* token, *embedding segment*, dan *embedding* posisi. Hasil akhir dari lapisan tersebut disebut *last_hidden_state*, tempat semua vektor kata *embedding* disimpan. Vektor kata ini akan dilatih untuk memahami bahasa, dan kemudian model akan disesuaikan sehingga dapat digunakan untuk tugas-tugas NLP (Khusuma dkk., 2023).

2.12 *Transformers based on Optical Character Recognition* (TrOCR)

Beberapa tahun belakangan, penggunaan teknologi pembelajaran mesin dan komputer visi telah mendorong perkembangan model OCR sehingga mencapai tingkat akurasi dan efisiensi yang luar biasa (Chen dan Zhou., 2023). Model TrOCR adalah model *transformer* yang diterbitkan oleh Microsoft yang mengimplementasikan OCR (Li dkk., 2021). Model TrOCR mencapai kinerja mutakhir dalam pengenalan teks dan memanfaatkan arsitektur *transformer* modern untuk mengimplementasikan pemahaman gambar dan generasi teks dalam satu model.

Berbeda dengan model pengenalan teks yang sudah ada, TrOCR adalah model yang sederhana namun efektif yang tidak menggunakan CNN sebagai *backbone* (Li dkk., 2021). Untuk melakukan pengenalan karakter pada gambar-gambar teks yang di-*cropped*, langkah pertama adalah mengubah ukuran gambar tersebut menjadi kotak persegi dengan ukuran 384 × 384 piksel kemudian, gambar-gambar tersebut dipecah menjadi urutan 576 *patch* (DosoViTsky dkk., 2021), dan setelah itu dijalankan melalui proses encoding oleh BEiT untuk menghasilkan representasi tingkat tinggi, yang kemudian di-decode oleh RoBERTa menjadi karakter yang sesuai langkah demi langkah (Li dkk., 2021). Gambar 46 merupakan arsitektur dari TrOCR.



Gambar 46. Arsitektur TrOCR (Li dkk., 2021)

Encoder menerima gambar input dengan ukuran awal $x_{img} \in \mathbb{R}^{3xH_0xW_0}$ dan mengubah menjadi ukuran tetap (H,W). Tetapi *encoder* transformers tidak bisa

langsung memproses gambar, kecuali jika gambar tersebut dipecah menjadi sejumlah $N = \frac{HW}{P^2}$ potongan kotak dengan ukuran yang sama (P,P). Kemudian, potongan-potongan ini diubah menjadi vektor dan diteruskan melalui proyeksi linear menjadi vektor-vektor dengan dimensi D, yang sering disebut sebagai *patches embedding*. Nilai D ini merupakan ukuran tersembunyi yang digunakan oleh Transformer dalam seluruh lapisan mereka (Li dkk., 2021).

Sama seperti dalam pendekatan ViT (Dosovitskiy dkk., 2021) dan *Data-efficient Image Transformers* (DeiT) (Touvron dkk., 2021), dengan memasukkan token khusus "[CLS]" yang biasanya digunakan dalam tugas klasifikasi gambar. Token "[CLS]" ini berfungsi untuk menggabungkan informasi dari seluruh *patches embedding* dan mewakili keseluruhan gambar. Selain itu, dalam kasus penggunaan model *pre-trained* DeiT, disertakan juga token distilasi dalam urutan input. Ini memungkinkan model untuk belajar dari model guru. Memberikan posisi absolut, baik *patches embedding* maupun kedua token khusus ini diberikan dengan *embedding* posisi 1D yang dapat dipelajari. Struktur BEiT sama dengan transformer gambar dan tidak memiliki token distilasi saat dibandingkan dengan DeiT.

Tidak seperti fitur yang diperoleh dari jaringan seperti CNN yang memiliki bias induktif yang berkaitan dengan gambar, model transformer tidak memiliki bias inisial yang bersifat gambar dan mengolah gambar dengan cara memperlakukannya sebagai urutan potongan. Hal ini memungkinkan model transformer untuk lebih fleksibel dalam memberikan perhatian baik kepada seluruh gambar maupun potongan-potongan gambar yang independen (Li dkk., 2021).

Inisialisasi *decoder* menggunakan model RoBERTa (Liu dkk., 2019) dan model MiniLM untuk menginisialisasi *decoder*. RoBERTa adalah studi replikasi dari (Devlin dkk., 2019) yang secara cermat mengukur dampak banyak *hyperparameter* kunci dan ukuran data pelatihan. Berdasarkan BERT, mereka menghilangkan tujuan prediksi kalimat berikutnya dan secara dinamis mengubah pola *masking* dari *Masked Language Model*. Modul *attention encoder-decoder*, *key* dan *value* diambil dari hasil keluaran *encoder*, sementara *query* diambil dari *input decoder*. Selain itu, *decoder* memanfaatkan *masking attention* pada *self-attention* untuk mencegah

dirinya sendiri mendapatkan lebih banyak informasi selama pelatihan daripada prediksi.

Model MiniLM adalah pendekatan sederhana dan efektif untuk mengompresi model terlatih berbasis *transformer* besar (Vaswani dkk., 2017) , yang disebut sebagai distilasi *self attention* yang dalam. Model ini mampu mempertahankan akurasi sebesar 99% dari model-model *transformers*. Model ini fokus pada distilasi modul *self-attention* dalam lapisan terakhir model guru. Penggunaan metode ini memungkinkan fleksibilitas dalam jumlah lapisan model siswa dan meningkatkan kinerja dengan memasukkan asisten guru.

Model TrOCR yang telah dilatih sebelumnya disesuaikan (*fine-tuning*) untuk tugas pengenalan teks berikutnya. Konteks ini, model TrOCR yang sudah ada telah melalui pelatihan awal pada data umum atau tugas yang berbeda. Kemudian, model tersebut disesuaikan ulang untuk tugas pengenalan teks yang lebih spesifik. *Output* dari model TrOCR didasarkan pada teknik-teknik seperti BPE dan *SentencePiece*, yang digunakan untuk menghasilkan representasi teks yang lebih umum dan tidak bergantung pada kosakata terkait tugas tertentu (Li dkk., 2021). Ini memungkinkan model TrOCR untuk lebih fleksibel dalam mengenali berbagai jenis teks tanpa ketergantungan pada kosakata yang sangat spesifik. Dengan kata lain, model ini dapat digunakan untuk berbagai tugas pengenalan teks dengan berbagai kosakata tanpa perlu pelatihan ulang yang intensif.

Dataset pra-pelatihan tahap pertama, dengan mengambil dua juta halaman dokumen dari berkas PDF yang tersedia secara publik di Internet. Secara total, dataset pra-pelatihan tahap pertama mengandung 684 juta garis teks (Li dkk., 2021).

Berbagai jenis huruf tulisan tangan (5.427 jenis) digunakan untuk mensintesis gambar garis teks tulisan tangan dengan bantuan *Text recognition data generator* (TRDG), yang mengambil teks dari halaman-halaman acak di Wikipedia. Dataset tulisan tangan untuk tahap pra-pelatihan kedua berisi 17,9 juta garis teks, termasuk dataset IIIT-HWS. Selain itu, sekitar 53 ribu gambar struk belanja dunia nyata dikumpulkan dan dienali teks di dalamnya dengan mesin OCR komersial. Dataset ini kemudian digunakan untuk pelatihan. TRDG digunakan untuk mensintesis 1 juta

gambar garis teks tercetak dengan berbagai jenis huruf, termasuk huruf struk belanja dan huruf cetak bawaan. Total dataset teks tercetak berjumlah 3,3 juta garis teks. Data untuk pengenalan teks di lingkungan luar ruangan berasal dari MJSynth (MJ) dan SynthText (ST), dengan total sekitar 16 juta gambar teks (Li dkk., 2021).

Benchmarks dataset *Scanned Receipts OCR and Information Extraction* (SROIE) (Task 2) berfokus pada pengenalan teks dalam gambar-gambar struk belanja. Tabel 10 menjelaskan hasil evaluasi pada berbagai model menggunakan dataset SROIE, terdapat 626 gambar struk belanja dalam set data pelatihan dan 361 gambar struk belanja dalam set data pengujian SROIE. Karena tugas deteksi teks tidak disertakan dalam penelitian ini, digunakan gambar-gambar garis teks yang di-*crop* untuk evaluasi. Gambar garis teks tersebut diperoleh dengan meng-*crop* gambar-gambar struk belanja secara keseluruhan berdasarkan kotak pembatas *ground truth* (Li dkk., 2021). Tabel 10 menjelaskan model encoder dari tiap model transformers gambar dan model decoder dari tiap model transformers pada teks dengan menggunakan dataset SROIE. Pada Tabel 11 menjelaskan inisialisasi model pra-pelatihan, augmentasi data, dan dua tahap pra-pelatihan pada dataset SROIE. Hasil evaluasi perbandingan model pada dataset SROIE dijelaskan pada Tabel 12.

Tabel 10. Evaluasi performa model pada dataset SROIE.

Encoder	Decoder	Precision	Recall	Fl score
DeiTBASE	RoBERTaBASE	69.28	69.06	69.17
BEiTBAsE	RoBERTaBASE	76.45	76.18	76.31
ResNet50	RoBERTaBASE	66.74	67.29	67.02
DeiTBAsE	RoBERTaLARGE	77.03	76.53	76.78
BEiT13AsE	RoBERTaLARGE	79.67	79.06	79.36
ResNet50	RoBERTaLARGE	72.54	71,13	71,83

Tabel 11. Augmentasi data dan dua tahap pra-pelatihan pada dataset SROIE.

Model	Precision	Recall	Fl score
From Scratch	38.06	38.43	38.24
+ Pretrained Model	72.95	72.56	72.75
+ Data Augmentation	82.58	82.03	82.3
+ First-Stage Pretrain	95.31	95.65	95.48
+ Second-Stage Pretrain	95.76	95.91	95.84

Tabel 12. Hasil evaluasi dari SROIE tugas 2.

Model	Recali	Precision	F1 Score
CRNN	28.71	48.58	36.09
Tesseract OCR	57.5	51,93	54.57
H&H Lab	96.35	96.52	96.43
MSOLab	94.77	94.88	94.82
CLOVA OCR	94.3	94.88	94.59
TrOCRsMALL	95.89	95.74	95.82
TrOCRBASE	96.37	96.31	96.34
TrOCRLARGE	96.59	96.57	96.58

Basis data IAM Handwriting terdiri dari teks berbahasa Inggris yang ditulis tangan, dan ini adalah dataset paling populer untuk pengenalan teks tulisan tangan. Dataset Aachen terbagi menjadi: 6.161 baris dari 747 formulir dalam dataset pelatihan, 966 baris dari 115 formulir dalam dataset validasi, dan 2.915 baris dari 336 formulir dalam dataset pengujian (Li dkk., 2021). Tabel 13 menjelaskan hasil evaluasi TrOCR pada dataset IAM Handwriting.

Tabel 13. Hasil evaluasi (CER) pada dataset IAM Handwriting.

Model	***Architecture***	***Training Data***	***External* LM**	**CER**
(Bluche and Messina 2017)	GCRNN / CTC	Synthetic + IAM	Yes	3.2
(Michael et al. 2019)	LSTM/LSTM w/Attn	IAM	No	4.87
(Wang et al. 2020a)	FCN / GRU	IAM	No	6.4
(Kang et al. 2020)	Transformer w/ CNN	Synthetic + IAM	No	4.67
(Diaz et al. 2021)	S-Attn / CTC	Internal + IAM	No	3.53
(Diaz et al. 2021)	S-Attn / CTC	Internal + IAM	Yes	2.75
(Diaz et al. 2021)	Transformer w/ CNN	Internal + IAM	No	2.96
$\mathrm{TrOCR}_{\mathrm{SMALL}}$	Transformer	Synthetic + IAM	No	4.22
$\mathrm{TrOCR}_{\mathrm{BASE}}$	Transformer	Synthetic + IAM	No	3.42
$\mathrm{TrOCR}_{\mathrm{LARGE}}$	Transformer	Synthetic + IAM	No	2.89

2.13 Metriks Evaluasi

Metriks evaluasi untuk YOLO mencakup *mean average precision* (mAP), *intersection over union* (IoU), *precision*, *recall*, dan F1 *score* (Jiang dkk., 2023).

Mean average precision (mAP) merupakan metrik yang sering digunakan untuk mengevaluasi kinerja model deteksi objek. Metrik ini mengukur presisi rata-rata pada semua kategori, kemudian memberikan satu nilai tunggal untuk membandingkan model-model yang berbeda. *Mean average precision* diperoleh dari bobot rata-rata nilai *average precision* (AP) berdasarkan semua kategori sampel, yang digunakan untuk mengukur kinerja deteksi model pada semua kategori (Ragland dan Tharcis, 2014).

Perhitungan mAP50 mengevaluasi AP dengan menggunakan *threshold* IoU 0,5. Sementara mAP50-95, AP dihitung pada setiap ambang batas IoU, dimulai dari 0,5 hingga 0,95 dengan interval 0,05 kemudian diambil rata-rata. Rumus perhitungan mAP disajikan pada Persamaan (2.55) (Li dkk., 2023)

$$mAP = \frac{1}{N}\sum_{k=1}^{k=N} AP_k \tag{2.55}$$

dengan:
N= jumlah kategori sampel

Menurut Wen dkk. (2022) AP sama dengan luas pada kurva presisi-recall dan dihitung menggunakan Persamaan (2.56) berikut:

$$AP = \int_0^1 precision(recall)\, d(recall) \tag{2.56}$$

Precision adalah kemampuan model untuk mengidentifikasi objek yang relevan. Rasio jumlah sampel positif yang diprediksi oleh model terhadap jumlah sampel yang terdeteksi dijelaskan pada Persamaan (2.57) (Wen, 2022).

$$precision = \frac{TP}{TP + FP} \tag{2.57}$$

dengan:

TP = *True positive*

FP = *False negative*

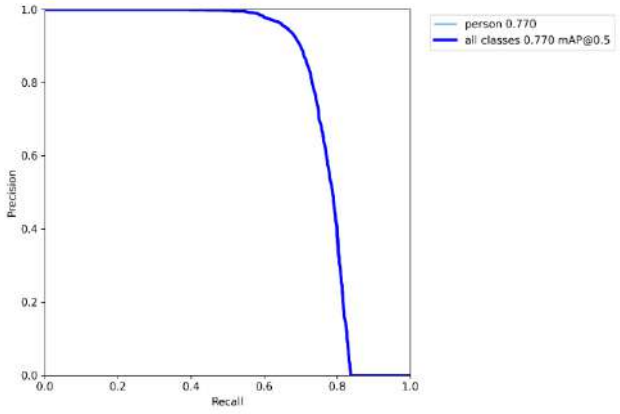
Recall adalah rasio jumlah sampel positif yang diprediksi dengan benar oleh model terhadap jumlah sampel positif yang benar-benar muncul. *Recall* mencerminkan sejauh mana model mampu untuk menemukan semua kotak pembatas *ground truth* yang dijelaskan pada Persamaan (2.58) (Padilla dkk., 2021).

$$recall = \frac{TP}{TP + FN} \tag{2.58}$$

dengan:

FN = *False Negative*

Kurva *precision-recall* adalah plot presisi pada sumbu X dan recall pada sumbu Y (Cochard, 2021). Terdapat *threshold* (ambang batas) dalam proses pendeteksian objek. Mengatur *threshold* secara lebih tinggi akan mengurangi kemungkinan pendeteksian objek yang berlebihan, namun meningkatkan peluang terlewatnya deteksi. Sebagai contoh, pada saat *threshold* diatur ke 1,0, tidak ada objek yang terdeteksi, mengakibatkan presisi mencapai 1,0, sementara *recall* mencapai 0,0, Sebaliknya, apabila ambang batasnya diturunkan ke 0,0, sejumlah objek yang tak terbatas akan terdeteksi, menghasilkan presisi sebesar 0,0 dan *recall* sebesar 1,0 diilustrasikan pada Gambar 47.



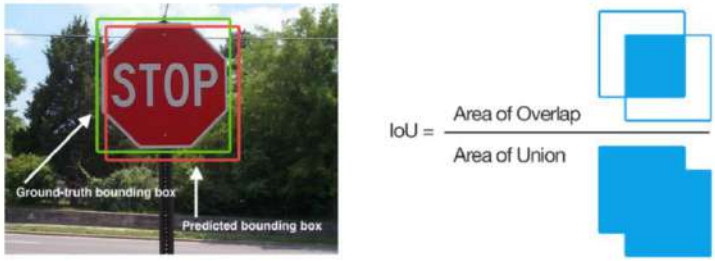
Gambar 47. Contoh kurva *precision-recall*

F1 *score* mengukur keseimbangan antara presisi dan *recall* yang diuraikan pada Persamaan (2.59) (Wen, 2022). Pada saat *F1-score* tinggi, hal ini menunjukkan bahwa presisi dan *recall* memiliki kinerja yang baik. Sedangkan apabila *F1-score* yang lebih rendah menunjukkan ketidakseimbangan antara presisi dan recall yang lebih besar. Nilai F1-score dituliskan dalam Persamaan (2.60) sebagai berikut:

$$f1 = 2\,x\,\frac{(precision\;x\;recall)}{(precision + recall)} \tag{2.60}$$

Intersection over union (IoU) atau Jaccard, mengukur kesamaan antara dua himpunan dan didefinisikan sebagai rasio perpotongan terhadap gabungan (Rezatofoghi, 2019). *Intersection Over Union* digunakan untuk menentukan apakah kotak pembatas telah diprediksi dengan benar.

Intersection Over Union menunjukkan tumpang tindih koordinat kotak pembatas yang diprediksi dengan kotak *ground truth* (Maleh dkk., 2023). Nilai IoU yang lebih tinggi menunjukkan koordinat kotak pembatas yang diprediksi sangat mirip dengan koordinat kotak *ground truth*. Rasio tumpang tindih antara *ground truth* dan prediksi kotak pembatas menjadi 1,0 ketika kedua kotak pembatas tepat dan 0,0 jika tidak ada tumpang tindih. Gambar 48 dan 49 menunjukan contoh dari *intersection over union*.



Gambar 48. Contoh *Intersection over Union* (Agrawal, 2022)

Gambar 49. Ilustrasi dari pengukuran IoU (Rosebrock, 2016)

Character error rate (CER) adalah ukuran umum untuk mengevaluasi seberapa baik sistem pengenalan ucapan otomatis berperforma. Sama seperti *Word Error Rate* (WER), CER fokus pada tingkat karakter alih-alih kata. Perhitungan CER melibatkan evaluasi kesalahan dalam pengenalan karakter, termasuk penggantian, penghapusan, dan penyisipan karakter yang diuraikan pada Persamaan (2.61),

$$CER = \frac{S + D + I}{S + D + C} \tag{2.61}$$

Di mana:

S = Jumlah subtitusi

D = Jumlah penghapusan

I = Jumlah penyisipan

C = Jumlah karakter yang benar

III. METODELOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan Tempat Penelitian ini yaitu sebagai berikut:

a. Tempat Penelitian

Penelitian ini dilakukan selama Semester Ganjil tahun akademik 2023/2024 dengan melakukan studi pustaka di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Lokasi bertempat di Jalan Prof. Dr. Soemantri Brojonegoro No.1, Gedong Meneng, Bandar Lampung.

b. Waktu Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2023/2024 pada bulan Oktober 2023. Penelitian ini dibagi menjadi tiga tahap, pertama dilakukan studi literatur dengan topik penelitian yang akan digunakan sebagai referensi dalam penyusunan proposal penelitian. Selanjutnya dilakukan pengumpulan data yang digunakan dalam penelitian. Kedua adalah pengerjaan program untuk pendeteksian pelat nomor menggunakan YOLOv8 dan pengenalan karakter pelat nomor menggunakan TrOCR. Ketiga adalah penyusunan hasil penelitian dan kesimpulan penelitian dalam bentuk draf laporan yang disampaikan pada seminar hasil. Setelah itu, akan dilaksanakan sidang komprehensif.

3.2 Data dan Alat Penelitian

Data dan alat penelitian ini sebagai berikut:

1. Data

Data yang digunakan dalam penelitian ini diambil dari beberapa sumber seperti https://universe.roboflow.com, dan pengambilan gambar secara langsung menggunakan kamera *handphone*. Data yang digunakan berjumlah 666 data gambar kendaraan terdiri dari 426 data pelatihan, 106 data validasi, dan 134 data pengujian. Setelah data terkumpul dilakukan anotasi dengan menggunakan Roboflow. Gambar 50, Gambar 51 dan Gambar 52 merupakan pembagian data *training*, data validasi dan data *testing*.

Gambar 50. Contoh data training yang sudah dianotasi

Gambar 51. Contoh data evaluasi yang sudah dianotasi

Gambar 52. Contoh data testing

2. Alat

Alat yang digunakan dalam penelitian ini sebagai berikut:

a. Perangkat keras (*Hardware*)

Perangkat keras yang digunakan pada penelitian ini adalah sebuah komputer dengan spesifikasi sebagai berikut.

- Processor :Intel® Core™ i3-8145U CPU @2.10GHz
- Memory :SSD 512 GB
- RAM :8GB DDR 4 3200 MHz
- GPU :NVIDIA GeForce MX350
- GPU Memory :3.9 GB + 2 GB

b. Perangkat lunak (Software)

Perangkat lunak yang digunakan pada penelitian ini antara lain:

- Sistem operasi Windows 11
- Google colab PRO dengan menggunakan GPU A100 dengan package yang digunakan sebagai berikut:

1. NumPy : Operasi numerik dan manipulasi array.
2. Pandas : Manipulasi dan analisis data menggunakan DataFrame.
3. OpenCV (cv2) : Pemrosesan gambar dan kompter visi.
4. PIL (Pillow) : Manipulasi gambar.
5. Transformers : Menggunakan TrOCRProcessor, Vision Encoder Decoder Model, Seq2SeqTrainer, dan lainnya.
6. PyTorch : Operasi deep learning.
7. Datasets : Memuat dataset.
8. Matplotlib : Membuat visualisasi seperti plot dan grafik.
9. Difflib : Menghitung kesamaan urutan karakter (digunakan untuk SequenceMatcher).
10. Google Colab : Mengakses Google Drive dan mungkin untuk menjalankan kode di Colab environment.
11. Roboflow : Mengelola data dan annotasi menggunakan Roboflow.
12. Shutil : Operasi manipulasi file dan direktori.
13. Os : Operasi sistem seperti manipulasi file dan direktori.
14. Ultralytics : Operasi terkait YOLO dan plotting.

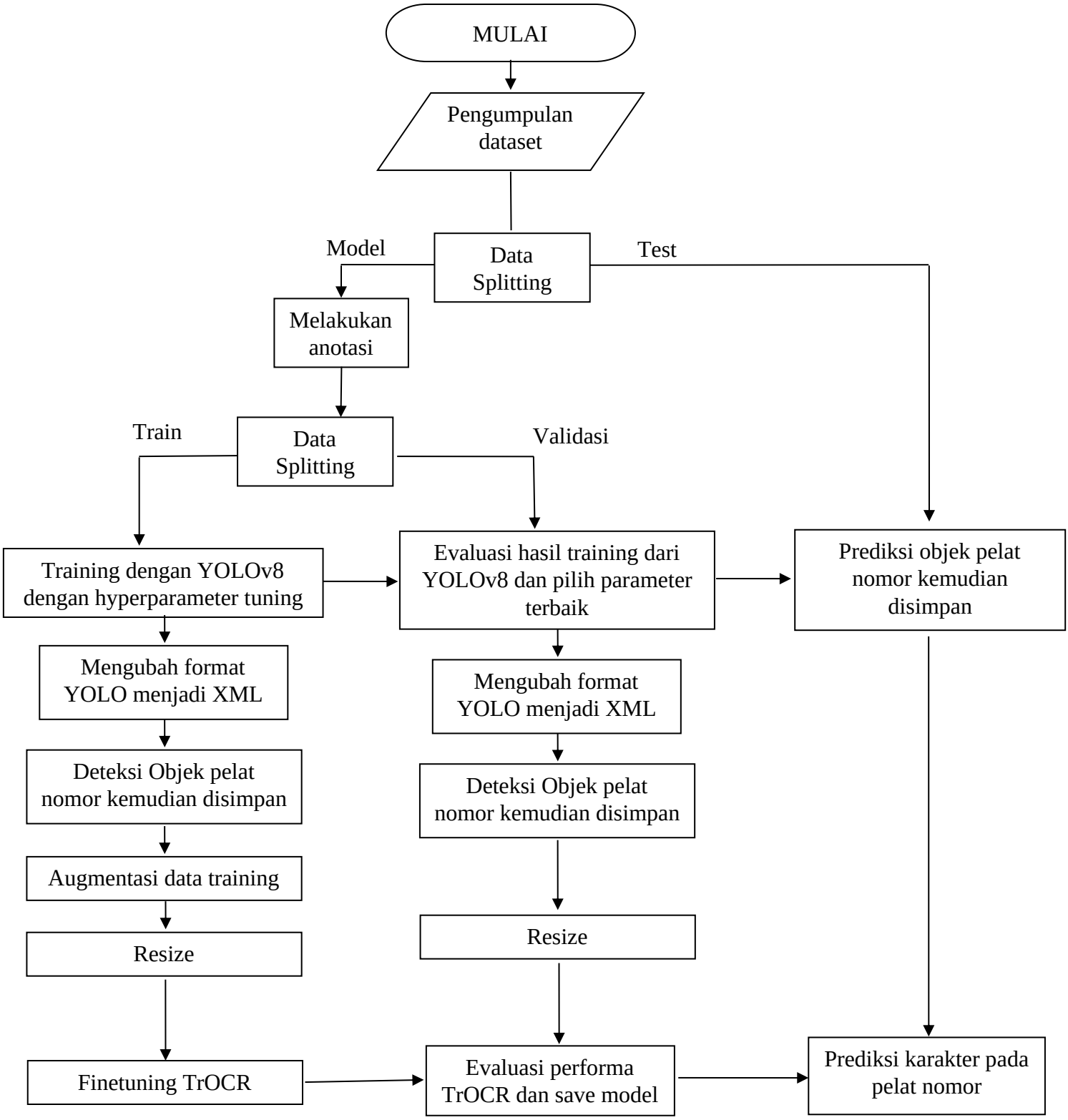
15. Plotly : Membuat visualisasi interaktif.
16. scikit-learn : Pemisahan dataset menggunakan train_test_split.

3.3 Metode Penelitian

Penelitian ini menggunakan 2 metode, pertama dengan mencari bounding box objek pelat nomor menggunakan YOLOv8 dan pembacaan pelat nomor kendaraan menggunakan TrOCR. Adapun langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Melakukan pengumpulan data gambar kendaraan.
2. Setelah semua data gambar terkumpul, setiap file gambar diberi nama sesuai dengan karakter yang terdapat pada pelat nomor kendaraan. Penamaan ini penting untuk memastikan bahwa setiap gambar memiliki label yang sesuai dengan isi dari pelat nomor kendaraan.
3. Selanjutnya Dilakukan proses pembersihan data gambar untuk memastikan konsistensi dan akurasi data. Jika terdapat gambar yang memiliki pelat nomor yang sama, maka perlu dilakukan penghapusan.
4. Membagi dataset menjadi 80% data model (532 gambar), dan data uji sebesar 20% (134 gambar).
5. Melakukan anotasi gambar kendaraan untuk mendeteksi pelat nomor kendaraan pada data model menggunakan Roboflow.
6. Pada data model dilakukan splitting data model sebesar 80% untuk data pelatihan (426 gambar) dan 20% untuk data validasi (106 gambar).
7. Resize gambar menjadi 640×640 piksel.
8. Training data dengan model YOLOv8n dengan 100 epochs, dan batch sebesar 64.
9. Menampilkan grafik training and validation loss pada localization, classification dan distributional focal.
10. Menampilkan metrics evaluasi seperti precision, recall, mAP50 dan mAP50-95.

11. Mengukur kinerja dari model yang telah dibangun dengan menggunakan data validasi.
12. Menampilkan *confusion matrix* dari data *validation*, kurva *precision-confidence*, kurva *recall-confidence*, kurva *F1-confidence*, dan kurva *precision-recall*.
13. Mengubah format label YOLO menjadi format XML.
14. Membuat prediksi pada *bounding box* setelah itu ekstraksi objek gambar pelat nomor kemudian menyimpannya.
15. Selanjutnya menerapkan augmentasi data pada data training menggunakan *adaptive thresholding* dengan mempertahankan gambar original.
16. Melakukan finetuning menggunakan seq2seqtrainer kemudian dilakukan training dan validasi dataset.
17. Menampilkan metric evaluasi seperti CER, eval_loss, eval_runtime, eval_samples_per_second, dan eval_steps_per_second.
18. Mengukur kinerja dari model TrOCR yang telah dilatih dengan menggunakan data validasi.
19. Melakukan pengujian pada data test hasil dari training YOLOv8 dan TrOCR.



Gambar 53. Flowchart Metode Penelitian.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan dari bab sebelumnya untuk deteksi pelat nomor menggunakan YOLOv8 dan pengenalan karakter pada pelat nomor menggunakan TrOCR diambil kesimpulan sebagai berikut:

1. Membangun sistem pendeteksian pelat nomor menggunakan YOLOv8 perlu dilakukan persiapan dataset, anotasi data, pelatihan model dan yang terakhir adalah evaluasi model.
2. Pada data validasi, hasil evaluasi pada metriks mAP 50 mencapai 99,5%, mAP50-95 sebesar 83,78%, dan recall sebesar 100%.
3. Membangun sistem pengenalan karakter menggunakan TrOCR melibatkan beberapa langkah, yaitu melakukan deteksi objek untuk mendapatkan koordinat bounding box, kemudian melakukan cropping gambar untuk mendapatkan region of interest pada pengenalan karakter dari pelat nomor, mengubah ukuran gambar pelat nomor menjadi 384×384, memberikan label pada gambar pelat nomor, augmentasi gambar, melatih model dan yang terakhir evaluasi model.
4. Hasil evaluasi yang didapat pada data validation, diperoleh nilai CER terbaik pada model TrOCR Large Printed 1 sebesar 0,011 dan pada data testing nilai CER sebesar 1,12%.

5.2 Saran

Adapun beberapa saran yang perlu diperhatikan untuk penelitian berikutnya adalah sebagai berikut:

1. Memperbanyak data training jika melakukan penelitian serupa, karena pada penelitian ini jumlah datanya sebanyak 666 gambar yang dibagi menjadi 3 yaitu 426 data pelatihan, 106 data validasi, dan 134 data pengujian.
2. Pada saat pengumpulan data, disarankan untuk memiliki keragaman gambar yang diambil dari berbagai kondisi. Misalkan pencahayaan (gelap dan terang), waktu (pagi, siang, dan malam), sudut pandang (tegak lurus dengan pelat nomor dan miring), jarak (jauh, dekat dan menengah) , jenis kendaraan (mobil, motor, truk, sepeda motor, dan bus) dan lain-lain. Tujuannya agar model bisa mendeteksi dan mengenali suatu karakter dari berbagai kondisi.
3. Diharapkan untuk mengecek kembali apakah ada gambar yang terlewatkan pada saat proses anotasi gambar dan juga pada saat pelabelan pelat nomor harus dipastikan sesuai dengan gambar agar pada saat dilakukan training tidak menimbulkan kesalahan.
4. Menggunakan model lain pada object detection dan pengenalan karakter.
5. Melakukan *post-processing* untuk meningkatkan akurasi pada pengenalan karakter.

DAFTAR PUSTAKA

Agrawal, J. 2022. Mean Average Precision (mAP) Explained in Object Detection. https://medium.com/@jalajagr/mean-average-precision-map-explained-in-object-detection-fb61adf67ef4 diakses pada tanggal 25 oktober 2023 pukul 13.20

Al-Masni, M. A., Al-Antari, M. A., Park, J. M., Gi, G., Kim, T. Y., Rivera, P., Valarezo, E., Choi, M. T., Han. S. M., & Kim, T. S. (2018). Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Computer methods and programs in biomedicine*, *157*, 85-94.

Altunay, D. 2019. The Basics of Image Processing and OpenCV. https://developer.ibm.com/articles/learn-the-basics-of-computer-vision-and-object-detection/ diakses pada tanggal 25 oktober 2023 pukul 13.20

Aprilino, A. (2022). Implementasi Algoritma Yolo dan Tesseract OCR Pada Sistem Deteksi Plat Nomor Otomatis. *Jurnal Teknoinfo*, ***16***(1), 54-59.

Bag, S. 2021, Activation Functions — All You Need To Know!. https://medium.com/analytics-vidhya/activation-functions-all-you-need-to-know-355a850d025e diakses pada tanggal 16 Oktober 2023 pukul 21,00

Bandyopadhyay, H. 2023. Image Annotation: Definition, Use Cases & Types [2023]. https://www.v7labs.com/blog/image-annotation-guide diakses pada tanggal 16 November 2023 pukul 16.50

Basak, H., Kundu, R., & Sarkar, R. (2022). MFSNet: A Multi Focus Segmentation Network for Skin Lesion Segmentation. *Pattern Recognition*, *128*, 108673.

Bazi, Y., Bashmal, L., Rahhal, M. M. A., Dayil, R. A., & Ajlan, N. A. (2021). Vision transformers for remote sensing image classification. *Remote Sensing*, ***13***(3), 516.

Camacho, J. C., & Morocho-Cayamcela, M. E. (2023). Mask R-CNN and YOLOv8 Comparison to Perform Tomato Maturity Recognition Task. In *Conference on Information and Communication Technologies of Ecuador* (pp. 382-396). Cham: Springer Nature Switzerland.

Chen, W., Huang, H., Peng, S., Zhou, C., & Zhang, C. (2021). YOLO-face: A Real-Time Face Detector. *The Visual Computer*, *37*, 805-813.

Chen, Y. H., & Zhou, Y. (2023). Enhancing OCR Performance through Post-OCR Models: Adopting Glyph Embedding for Improved Correction. *arXiv preprint arXiv:2308.15262*.

Cochard, D. 2021. mAP: Evaluation metric for object detection models. https://medium.com/axinc-ai/map-evaluation-metric-of-object-detection-model-dd20e2dc2472

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, 4171-4186.

Dewi, C., Chen, R. C., Jiang, X., & Yu, H. (2022). Deep Convolutional Neural Network for Enhancing Traffic Sign Recognition Developed on YOLO V4. *Multimedia Tools and Applications*, ***81***(26), 37821-37845.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *In International Conference on Learning Representations (ICLR) 2021*.

Du, L., Zhang, R., dan Wang, X. (2020), Overview of Two-Stage Object Detection Algorithms. In *Journal of Physics: Conference Series* (Vol. 1544, No. 1, p. 012033). IOP Publishing.

Eikvil, L. (1993). Optical Character Recognition. *citeseer. ist. psu. edu/142042. html*, *26*.

Elfwing, S., Uchibe, E., & Doya, K. (2018). Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural networks*, *107*, 3-11.

Gonzalez, R. C., & Woods, R. (2009). Digital Image Processing: Pearson education india. *Digital image processing: Pearson education india*.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., & Tao, D. (2022). A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, ***45***(1), 87-110.

Hendrycks, D., & Gimpel, K. (2016). Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*.

Jähne, B., Haussecker, H., & Geissler, P. (Eds.). (1999). *Handbook of computer vision and applications* (Vol. 2, pp. 423-450). San Diego: Academic press.

Jiang, H., Hu, F., Fu, X., Chen, C., Wang, C., Tian, L., & Shi, Y. (2023). YOLOv8-Peas: a lightweight drought tolerance method for peas based on seed germination vigor. *Frontiers in Plant Science*, *14*, 1257947.

Khusuma, R., Maharani, W., & Gani, P. H. (2023). Personality Detection On Twitter User With RoBERTa. *Jurnal Media Informatika Budidarma*, ***7***(1), 542-553.

Kumaseh, M. R., Latumakulita, L., & Nainggolan, N. (2013). Segmentasi Citra Digital Ikan Menggunakan Metode Thresholding. *Jurnal Ilmiah Sains*, ***13***(1), 74-79.

Kundu, R. 2023. Image Processing: Techniques, Types, & Applications. https://www.v7labs.com/blog/image-processing-guide diakses pada tanggal 25 oktober 2023 pukul 13.20

Lee, M. (2023). Mathematical analysis and performance evaluation of the gelu activation function in deep learning. *Journal of Mathematics*, ***2023***(1), 4229924.

Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). Trocr: Transformer-Based Optical Character Recognition with Pre-Trained Models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, No. 11, pp. 13094-13102).

Li, Z., Pang, C., Dong, C., & Zeng, X. (2023). R-YOLOv5: A lightweight rotational object detection algorithm for real-time detection of vehicles in dense scenes. *IEEE Access*, *11*, 61546-61559.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv, abs/1907.11692*.

Lu, Q., Liu, Y., Huang, J., Yuan, X., & Hu, Q. (2019). License Plate Detection and Recognition Using Hierarchical Feature Layers from CNN. *Multimedia Tools and Applications*, *78*, 15665-15680.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proc. icml* (Vol. 30, No. 1, p. 3).

Maleh, I. M. D., Teguh, R., Sahay, A. S., Okta, S., & Pratama, M. P. (2023). Implementasi Algoritma You Only Look Once (YOLO) untuk Object Detection Sarang Orang Utan. *Jurnal Informatika*, ***10***(1).

Mostafa, A., Mohamed, O., Ashraf, A., Elbehery, A., Jamal, S., Salah, A., & Ghoneim, A. S. (2022). An End-to-End OCR Framework For Robust Arabic-Handwriting Recognition using a Novel Transformers-Based Model and an Innovative 270 Million-Words Multi-Font Corpus of Classical Arabic with Diacritics. *arXiv preprint arXiv:2208.11484*.

Munir, R. (2004). Pengolahan Citra Digital dengan Pendekatan Algoritmik. *Informatika, Bandung, 260*.

Nie, Y., Sommella, P., O'Nils, M., Liguori, C., & Lundgren, J. (2019). Automatic Detection of Melanoma with YOLO Deep Convolutional Neural Networks. In *2019 E-Health and Bioengineering Conference (EHB)* (pp. 1-4). IEEE.

Ott, M., Edunov, S., Grangier, D., & Auli, M. (2018). Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers* (pp. 1-9). Brussels, Belgium: Association for Computational Linguistics.

Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & Da Silva, E. A. (2021). A Comparative Analysis Of Object Detection Metrics With A Companion Open-Source Toolkit. *Electronics*, ***10***(3), 279.

Patel, C., Patel, A., & Patel, D. (2012). Optical character recognition by open source OCR tool tesseract: A case study. *International journal of computer applications*, ***55***(10), 50-56.

Pokhrel, S. (2020), Image Data Labelling and Annotation — Everything you need to know. https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1 diakses pada tanggal 16 November 2023 pukul 16.54

Ravirathinam, P., & Patawari, A. (2019). Automatic License Plate Recognition for Indian Roads Using Faster-RCNN. In *2019 11th international conference on advanced computing (ICoAC)* (pp. 275-281). IEEE.

Pytorch. 2023. SILU. https://pytorch.org/docs/stable/generated/torch.nn.SiLU.html diakses pada tanggal 16 Oktober 2023 pukul 16.00

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI blog*, ***1***(8), 9.

Ragland, K., & Tharcis, P. (2014). A Survey on Object Detection, Classification and Tracking Methods. *Int. J. Eng. Res. Technol*, ***3***(11), 622-628.

Rosebrock, A. 2016. Intersection over Union (IoU) for Object Detection. https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ diakses pada tanggal 25 oktober 2023 pukul 13.20

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 658-666).

Ruan, B. K., Shuai, H. H., & Cheng, W. H. (2022). Vision Transformers: State of The Art and Research Challenges. *arXiv preprint arXiv:2207.03041*.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1715-1725). Berlin, Germany: Association for Computational Linguistics.

Septiaji, K. D., & Firdausy, K. (2018). Deteksi Kematangan Daun Selada (Lactuca Sativa L) Berbasis Android Menggunakan Nilai RGB Citra. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, ***4***(1), 20-27.

Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. (2020). Image Processing Techniques for Diagnosing Rice Plant Disease: A Survey. *Procedia Computer Science, 167*, 516-530.

Shinde, A. A., & Chougule, D. G. (2012). Text Pre-Processing and Text Segmentation for OCR. *International Journal of Computer Science Engineering and Technology*, ***2***(1), 810-812.

Simay. 2020, Object Detection vs. Image Segmentation. https://medium.com/inovako/object-detection-vs-image-segmentation-e5290e4690d diakses 10 November 2023 pukul 22.35

Soviany, P., & Ionescu, R. T. (2018). Optimizing The Trade-Off Between Single-Stage and Two-Stage Deep Object Detectors Using Image Difficulty Prediction. In *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)* (pp. 209-214). IEEE.

Sulistiyanti, S. R., Setyawan, F. X. A., & Komarudin, M. (2016). Pengolahan citra: Dasar dan Contoh Penerapannya. Yogyakarta: Teknosain.

Sultana, F., Sufian, A., & Dutta, P. (2020). A Review of Object Detection Models Based on Convolutional Neural Network. *Intelligent computing: image processing based applications*, 1-16.

Tamang, S., Sen, B., Pradhan, A., Sharma, K., & Singh, V. K. (2023). Enhancing Covid-19 Safety: Exploring YOLOV8 Object Detection for Accurate Face Mask Classification. *International Journal of Intelligent Systems and Applications in Engineering*, ***11***(2), 892-897.

Tan, L., Huangfu, T., Wu, L., & Chen, W. (2021). Comparison of Retinanet, SSD, and YOLO V3 for Real-Time Pill Identification. *BMC medical informatics and decision making*, *21*, 1-11.

Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of Yolo Architectures in Computer Vision: from YOLOV1 to YOLOV8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, ***5***(4), 1680-1716.

Tirtana, E., Gunadi, K., & Sugiarto, I. (2021). Penerapan Metode YOLO dan Tesseract-OCR untuk Pendataan Plat Nomor Kendaraan Bermotor Umum di Indonesia Menggunakan Raspberry Pi. *Jurnal Infra*, ***9***(2), 241-247.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998-6008). Curran Associates, Inc.

Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOV8 for UAV Aerial Photography Scenarios. *Sensors*, ***23***(16), 7190.

Wen, C., Chen, H., Ma, Z., Zhang, T., Yang, C., Su, H., & Chen, H. (2022). Pest-YOLO: A Model for Large-Scale Multi-Class Dense and Tiny Pest Detection and Counting. *Frontiers in Plant Science*, *13*, 973985.

Wihartasih, D., & Wibawanto, H. (2015). Pembuatan Prototipe Sistem Deteksi Plat Kendaraan Bermotor di Indonesia. *Edu Komputika Journal*, *2*(2).

Zhang, C., & Lin, L. (2021). Image Processing Methods in Agricultural Observation Systems. *Agro-geoinformatics: Theory and Practice*, 81-102.

Zhang, H., Whittaker, E., & Kitagishi, I. (2023). Extending TrOCR for Text Localization-Free OCR of Full-Page Scanned Receipt Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1479-1485).

Zheng, T., Zhao, S., Liu, Y., Liu, Z., & Cai, D. (2022). Scaloss: Side and Corner Aligned Loss for Bounding Box Regression. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, No. 3, pp. 3535-3543).