



TUGAS AKHIR - EC 184801

**PENGENALAN TULISAN TANGAN NASKAH AKSARA JAWA
ATURAN PENULISAN MARDI KAWI MENJADI HURUF ALFABET
PADA MEDIA KERTAS MENGGUNAKAN METODE CONVOLUTIONAL
NEURAL NETWORK (CNN) ARSITEKTUR YOU ONLY LOOK ONCE**

Lestari Ekky Damayanti
NRP 0721154000008

Dosen Pembimbing
Prof. Dr. Ir. Yoyon K. Suprpto, M.Sc.
Dr. Eko Mulyanto Yuniarno, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



TUGAS AKHIR - EC 184801

**PENGENALAN TULISAN TANGAN NASKAH AKSARA JAWA
ATURAN PENULISAN MARDI KAWI MENJADI HURUF
ALFABET PADA MEDIA KERTAS MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) ARSITEKTUR
YOU ONLY LOOK ONCE (YOLO)**

Lestari Ekky Damayanti
NRP 07211540000008

Dosen Pembimbing
Prof. Dr. Ir. Yoyon K. Suprpto, M.Sc.
Dr. Eko Mulyanto Yuniarno, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 202



FINAL PROJECT - EC 184801

***Javanese Sript Handwritting Mardi Kawi Writing
Recognition Rules Become Alphabet Letters On
Paper Media Using The Convolutional Neural
Network (CNN) Method With You Only Look Once
(YOLO) Architecture***

Lestari Ekky Damayanti
NRP 0721154000008

Advisors

Prof. Dr. Ir. Yoyon K. Suprpto, M.Sc.
Dr. Eko Mulyanto Yuniarno, ST., MT.

DEPARTEMENT OF COMPUTER ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Pengenalan Tulisan Tangan Naskah Aksara Jawa Aturan Penulisan Mardi Kawi Menjadi Huruf Alfabet Pada Media Kertas Menggunakan Metode Convolutional Nerural Network (CNN) Arsitektur You Only Look Once (YOLO)”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutipmaupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2020



Lestari Ekky Damayanti
NRP. 0721154000008

LEMBAR PENGESAHAN

Pengenalan Tulisan Tangan Naskah Aksara Jawa Aturan Penulisan Mardi Kawi Menjadi Huruf Alfabet Pada Media Kertas Menggunakan Metode Convolutional Neural Network Arsitektur You Only Look Once

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh : Lestari Ekky Damayanti (NRP. 07211540000008)

Tanggal Ujian : 06 Januari 2020

Periode Wisuda : Maret 2020

Disetujui oleh :

Prof. Dr. Ir. Yoyon K. Suprpto, M.Sc
NIP. 195409251978031001

(Pembimbing I)

(Pembimbing II)

Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP. 196806011995121009

(Penguji I)

Dr. Diah Puspito Wulandari, ST., M.Sc.
NIP. 198012192005012001

(Penguji II)

Dr. Supeno Mardi Susiki Nugroho., ST., MT.
NIP. 197003131995121001

Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.
NIP. 197003131995121001

Halaman ini sengaja dikosongkan

ABSTRAK

Nama Mahasiswa : Lestari Ekky Damayanti
Judul Tugas Akhir : Pengenalan Tulisan Tangan Naskah Aksara Jawa Aturan Penulisan *Mardi Kawi* Menjadi Huruf Alfabet Pada Media Kertas Menggunakan Metode *Convolutional Neural Network* (CNN) Arsitektur *You Only Look Once* (YOLO)
Pembimbing : 1. Prof. Dr. Ir. Yoyon K. Suprpto, M.Sc.
2. Dr. Eko Mulyanto, ST., MT.

Pelestarian kebudayaan perlu dilakukan, salah satunya melestarikan bukti-bukti sejarah dalam wujud naskah. Pengenalan aksara Jawa diperlukan sebagai upaya pelestarian peninggalan kebudayaan bangsa. Dewasa ini pengenalan aksara Jawa masih menggunakan metode sederhana yaitu dengan membaca satu persatu lalu menuliskan kembali. Pada tugas akhir ini akan dilakukan pengenalan aksara Jawa yang setiap aksaranya diambil dari gambar hasil pemindaian sebuah naskah teks media kertas dengan aturan penulisan Aksara Jawa *Mardi Kawi* menjadi huruf alfabet sehingga dapat diketahui bagaimana cara membacanya. Menggunakan metode *Convolutional Neural Network* (CNN) sebagai metode pemrosesan data.

Kata Kunci – tulisan tangan, aksara Jawa aturan penulisan *Mardi Kawi*, *Convolutional Neural Network* (CNN).

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Lestari Ekky Damayanti
Title : *Javanese Script Handwriting Mardi Kawi Writing Recognition Rules Become Alphabet Letters On Paper Media Using The Convolutional Neural Network (CNN) Method With You Only Look Once (YOLO) Architecture*
Advisors : 1. Prof. Dr. Ir. Yoyon K. Suprpto, M.Sc.
2. Dr. Eko Mulyanto, ST., MT.

Cultural preservation needs to be done, one of which is preserving historical evidence in the form of manuscripts. The introduction of Javanese script is needed as an effort to preserve the cultural heritage of the nation. Today, the introduction of Javanese script still uses a simple method, which is to read one by one and then rewrite it. In this final project, an introduction to Javanese script will be carried out, with each script taken from a scanned image of a paper text paper with the rules of writing Javanese script Mardi Kawi into alphabet letters so that it can be known how to read it. Using the Convolutional Neural Network (CNN) method as a data processing method.

Keywords– *handwriting, Javanese script rules for writing Mardi Kawi, Convolutional Neural Network (CNN).*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas segala limpahan berkah, kebijaksanaan, serta penyertaanNya, penulis dapat menyelesaikan penelitian dengan judul **“Pengenalan Tulisan Tangan Naskah Aksara Jawa Aturan Penulisan Mardi Kawi Menjadi Huruf Alfabet Pada Media Kertas Menggunakan Metode *Convolutional Neural Network* (CNN) Arsitektur *You Only Look Once* (YOLO)”**.

Penelitian ini disusun dalam rangka pemenuhan tugas akhir Departemen Teknik Komputer ITS, Bidang Studi Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa
2. Kedua orangtua, Bapak Kristoto Ari Santoso dan Ibu Sarmini yang selalu memberikan dukungan baik material dan spiritual dalam mengerjakan penelitian ini.
3. Bapak Kepala Departemen Teknik Komputer Dr. Supeno Mardi Susiki Nugroho, ST., MT
4. Prof. Dr. Ir. Yoyon Kusnendar Suprpto M.Sc. dan Dr. Eko Mulyanto Yuniarno, ST., MT. selaku dosen pembimbing, atas dukungan dan bimbingan selama mengerjakan penelitian ini.
5. Bapak-ibu dosen pengajar serta staff Departemen Teknik Komputer atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini
6. Teman-teman S2 dan S3 Teknik Elektro yang membantu penulis mempelajari materi penunjang penelitian ini.
7. Sahabat terkasih Liestria, Sari, Josephine dan Dyah yang selalu ada dalam suka dan duka penulis selama pengerjaan penelitian ini.

8. Teman-teman laboratorium B201 dan B401 yang sudah memperbolehkan penulis untuk berkunjung ke laboratorium dalam pengerjaan penelitian ini.
9. Keluarga Chany Risqina, keluarga Dewi Ayu, dan kamar kos Oni Eka yang menjadi tempat istirahat penulis dalam proses pengerjaan penelitian ini.
10. Para sahabat Teknik Komputer 2015 Fiya, Yaya, Vipril, Selvy, Ricky, Kywa, Ishmat, Habib, Randy, dan William yang tetap memberikan semangat kepada penulis walaupun telah menjadi alumni.
11. Anggota KITT, Kak Imel, Nadya, Debora, Merisa, Enzy, dan Florence yang selalu mendoakan penulis.
12. Teman-teman HIMATEKTRO, PD FTE, dan juga PMK ITS yang memberikan support dan juga semangat serta doa kepada penulis.

Penulis memohon segenap kritik dan saran yang membangun. Harapannya penelitian ini dapat berguna sebagai acuan penelitian-penelitian selanjutnya dan bermanfaat bagi kita semua.

Surabaya, Januari 2020

Penulisi

DAFTAR ISI

Abstrak	i
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
DAFTAR NOMENKLATUR	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	2
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Sistematika Penulisan	4
BAB 2 DASAR TEORI.....	5
2.1 <i>Related Works</i>	5
2.1.1 <i>Related Works</i> Terkait dengan Naskah	5
2.1.2 <i>Related Works</i> Terkait dengan Segmentasi Tulisan Tangan.....	5
2.1.3 <i>Related Works</i> Terkait dengan CNN Aksara Jawa.....	6
2.1.4 <i>Related Works</i> Terkait dengan Pengenalan Aksara Jawa.....	7
2.2 Aturan Penulisan Aksara Jawa Mardi Kawi	8
2.2.1 Aksara Wyanjana atau aksara Konsonan	8
2.2.2 Aksara Swara	9
2.3 <i>Artificial Neural Network</i>	10
2.3.1 <i>Convolutional Neural Networks</i>	12
2.3.2 <i>Convolutional Layer</i>	13
2.3.3 <i>Pooling Layer</i>	15
2.3.4 <i>Activation Function</i>	16

2.4	Arsitektur <i>Convolutional Neural Networks</i>	16
2.4.1	<i>Feature Extraction Layer</i>	16
2.4.2	<i>Fully Connected Layer</i>	17
2.4.3	YOLO (<i>You Only Look Once</i>)	17
BAB 3 DESAIN DAN IMPLEMENTASI SISTEM		25
3.1	Desain Sistem	25
3.2	Alur Kerja	26
3.3	Pembuatan Skenario Pengujian.....	26
3.4	Pelabelan Aksara Jawa.....	28
3.5	<i>Training</i> dan <i>Testing</i>	31
3.5.1	<i>Training</i>	31
3.5.2	<i>Testing</i>	36
3.6	Analisis Hasil Pendeteksian Aksara Jawa.....	36
3.6.1	<i>Intersection over Union (IoU)</i>	37
3.6.2	<i>Confusion Matrix</i>	38
BAB 4 PENGUJIAN DATA DAN ANALISA		41
4.1	Hasil Metodologi Penelitian.....	41
4.1.1	Pengumpulan Data	41
4.1.2	Anotasi Data atau <i>labelling image</i>	54
4.1.3	Hasil Anotasi Data	56
4.2	Training Data	59
4.2.1	<i>Weight</i> (bobot)	59
4.3	Analisis Data Hasil Pengujian	59
4.3.1	Pengujian Pendeteksian Aksara Jawa Konsonan	59
4.3.2	Pengujian Aksara Jawa Menggunakan Pasangan atau <i>Sandhangan</i>	66
4.3.3	Pengujian Pembuktian Gambar Dataset Training dan Testing.....	72
Bab 5 KESIMPULAN		79

5.1 Kesimpulan	79
5.2 Saran	80
DAFTAR PUSTAKA.....	83
LAMPIRAN	85

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 Aksara Konsonan Jawa Mardi Kawi.....	9
Gambar 2.2 Aksara Swara Mardi Kawi.....	9
Gambar 2.3 Aksara Sandhangan.....	9
Gambar 2.4 Operasi Perceptron pada ANN.....	11
Gambar 2.5 <i>An simple CNN Architecture</i>	12
Gambar 2.6 <i>Convolutional Neural Networks</i>	13
Gambar 2.7 Proses <i>Stride</i>	14
Gambar 2.8 <i>Max Pooling</i>	15
Gambar 2.9 Sistem pendeteksian YOLO.....	18
Gambar 2.10 Skematik Arsitektur YOLOv3 <i>multi-scale detector</i>	19
Gambar 2.11 Arsitektur YOLOv3.....	20
Gambar 2.12 Diagram contoh YOLOv3 <i>Network Architecture</i>	21
Gambar 2.13 Model Pendeteksian pada YOLO.....	22
Gambar 3.1 Blok Diagram Alir Sistem.....	26
Gambar 3.2 Diagram Alir Secara Keseluruhan.....	27
Gambar 3.3 Diagram Alir <i>labelling</i>	29
Gambar 3.4 Program labelImg.py untuk <i>labelling</i>	30
Gambar 3.5 <i>Output</i> Anotasi Data.....	30
Gambar 3.6 Tahapan secara umum <i>pattern recognition</i>	31
Gambar 3.7 Diagram Alir Secara Umum Pendeteksian Objek oleh CNN (<i>Convolutional Neural Network</i>).....	32
Gambar 3.8 Prediksi Lokasi <i>Bounding Box</i>	32
Gambar 3.9 Diagram Alir Deteksi Objek.....	33
Gambar 3.10 Visualisasi <i>Intersection over Union</i>	37
Gambar 3.11 Gambar Contoh Variasi Hasil IoU.....	38
Gambar 4.1 Tatacara, Padmasusastra.....	42

Gambar 4.2 Babad Tanah Jawi.....	43
Gambar 4.3 Nutilen Parepatan Pangreh	44
Gambar 4.4 Serat Gancaran Warna-warni Ing Jaman Punika	45
Gambar 4.5 Aksara Jawi.....	46
Gambar 4.6 Aksara Jawi.....	47
Gambar 4.7 <i>Lelamphanipun R. Ng. Ranggawarsita, Pujangga Karaton ing Surakarta</i>	48
Gambar 4.8 Proses <i>labelling</i> pada kertas1.jpg.....	56
Gambar 4.9 Proses <i>labelling</i> pada kertas2.jpg	57
Gambar 4.10 Proses <i>labelling</i> pada kertas3.jpg	57
Gambar 4.11 Hasil Anotasi 33 <i>Class</i>	58
Gambar 4.12 Hasil Anotasi 47 <i>Class</i>	58
Gambar 4.13 Hasil Training berupa bobot dari tiap epoch	59
Gambar 4.14 Grafik nilai IoU hasil deteksi 33 class.....	60
Gambar 4.15 Gambar hasil test 33 class.....	61
Gambar 4.16 Grafik nilai IoU hasil deteksi 33 class.....	62
Gambar 4.17 Gambar hasil test 33 class.....	63
Gambar 4.18 Grafik nilai IoU hasil deteksi 33 class.....	66
Gambar 4.19 Gambar hasil test 33 class.....	67
Gambar 4.20 Grafik nilai IoU hasil deteksi 47 class.....	68
Gambar 4.21 Gambar hasil test 47 class.....	69
Gambar 4.22 Kertas1.jpg Tatacara Padmasusastra (hlm. 300)	72
Gambar 4.23 Kertas9.jpg Nutilen Parepatan.....	73
Gambar 4.24 Kertas13.jpg Tatacara Padmasusastra (hlm. prolog)	73
Gambar 4.25 Kertas25.jpg Serat Gancaran Warni-warni (hlm. 114)	74
Gambar 4.26 Kertas41.jpg Aksara Jawa.....	74
Gambar 4.27 Test4.jpg Aksara Jawa.....	75
Gambar 4.28 Test61.jpg Tatacara Padmasusastra (hlm. 301)	76
Gambar 4.29 Test65.jpg Tatacara Padmasusastra (hlm. 303)	76

Gambar 4.22 Test25.jpg Ranggawarsita (hlm. 1) 77

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Susunan <i>Layer Darknet53</i>	35
Tabel 3.2 Tabel Penjelasan <i>Confusion Matrix</i>	38
Tabel 4.1 Tabel Naskah Serat Tatacara.....	49
Tabel 4.2 Tabel Naskah Babad Tanah Jawi.....	49
Tabel 4.3 Tabel Naskah Nutilen Parepatan Pangreh.....	50
Tabel 4.4 Tabel Naskah Serat Gancaran Warni-Warni Ing Jaman Punika.....	51
Tabel 4.5 Tabel Naskah Lelampahanipun R. Ng. Ranggawarsita Pujangga Karaton Ing Surakarta.....	51
Tabel 4.6 Daftar Gambar untuk <i>Training</i> 600x800.....	52
Tabel 4.7 Daftar Gambar untuk <i>Testing</i> 600x800.....	52
Tabel 4.8 Daftar Gambar untuk <i>Training</i> 416x416.....	53
Tabel 4.9 Daftar Gambar untuk <i>Testing</i> 416x416.....	53
Tabel 4.10 Tabel Naskah Mardi Kawi 33 <i>class</i> huruf Jawa Konsonan.....	54
Tabel 4.11 Tabel Naskah Mardi Kawi 47 <i>class</i> huruf Jawa Konsonan dan Huruf “Ka” Pasangan.....	55
Tabel 4.12 Analisis <i>Confusion Matrix</i> 33 <i>class</i>	65
Tabel 4.13 Analisis <i>Confusion Matrix</i> 47 <i>class</i>	70

Halaman ini sengaja dikosongkan

NUMENKLATUR

W = panjang atau tinggi *input*

N = panjang atau tinggi *filter*

P = *zero padding*

S = *stride* (ukuran *featured map*)

B = *batch* (jumlah *bounding box*)

C = jumlah *class*

x_i = nilai x mentahan *output*

\hat{x}_i = nilai x label

y_i = nilai y mentahan *output*

\hat{y}_i = nilai y label

C_i = nilai *confidence* mentahan *output*

\hat{C}_i = nilai *confidence* label

$p_i(c)$ = nilai probabilitas *class* mentahan *output*

$\hat{p}_i(c)$ = nilai probabilitas *class* label xv

cx = jarak *box* x dari pojok kiri atas

cy = jarak *box* y dari pojok kiri atas

bx = nilai x output model setelah diproses b

by = nilai y output model setelah diproses et

pw = ukuran *anchor* w

ph = ukuran *anchor* h

bw = nilai w *output* model setelah diproses

bh = nilai h *output* model setelah diproses

tw = nilai x mentahan *output* model

th = nilai y mentahan *output* model

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar Belakang

Terdapat beberapa perkembangan penulisan aksara Jawa di Indonesia. Mardi Kawi merupakan salah satunya. Mardi Kawi adalah salah satu karya buku monumental karya W. J. S. Poerwadarminta Mardi. Buku ini ditulis pada tahun 1860, merupakan kumpulan tulisan tentang tata tulis aksara Jawa, yang lazim digunakan pada karya sastra Jawa Kuna pada masa kejayaan kerajaan Hindu di pulau Jawa[1]. Naskah sendiri merupakan cerita tentang hasil yang diungkapkan oleh teks klasik dapat dibaca dalam peninggalan-peninggalan yang berupa tulisan[2]. Dengan mengingat betapa pentingnya upaya-upaya penyelamatan, pelestarian, dan penyebarluasan naskah kuno, maka penelitian mengenai upaya digitalisasi dan pengolahan pada hasil digitalisasi naskah. Pada penelitian yang sebelumnya telah dilakukan, penyelamatan naskah dilakukan dengan pengumpulan hasil perbaikan citra dari naskah teks dengan menggunakan berbagai macam metode, salah satunya adalah *adaptive local thresholding*[3]. Tahap yang selanjutnya dapat dilakukan yaitu melakukan tahapan pengenalan aksara pada teks hasil metode penelitian sebelumnya. Kesulitan pembacaan aksara Jawa muncul dikarenakan aksara Jawa tidak lagi dipergunakan sebagai aksara dalam kehidupan sehari-hari[4]. Pengenalan aksara Jawa telah dilakukan oleh Yayasan Sastra Lestari Surakarta yang didirikan oleh John Paterson bersama dengan alumni Fakultas Sastra Universitas Sebelas Maret (UNS) Surakarta. Yayasan Sastra Lestari Surakarta mempekerjakan beberapa karyawan dengan kemampuan membaca naskah Aksara Jawa baik dan kemudian karyawan mengetikkan cara pembacaannya ke media komputer

menggunakan aksara alfabet[5]. Pada penelitian ini akan digunakan *Convolutional Neural Network* (CNN) adalah metode *neural network* yang biasa digunakan pada *data image* untuk mendeteksi dan mengenali *object* pada sebuah image hasil pemindaian teks naskah tulisan tangan bertuliskan aksara Jawa yang kemudian dapat digunakan sebagai pemrosesan pengenalan aksara Jawa. Karena pada penelitian ini akan dilakukan menggunakan konsep *handwriting recognition* atau pengenalan tulisan tangan, akan dilakukan penelitian menggunakan metode *convolutional neural network*. Beberapa penelitian terkait mengenai pengenalan huruf dengan menggunakan CNN telah dilakukan oleh Rismiyati, dkk[6] melakukan penelitian mengenai pengenalan Jawa baru yang terdiri dari 20 aksara dengan media berupa kertas. Penelitian tersebut membandingkan pengenalan aksara menggunakan 2 metode yaitu CNN dan DNN (*Deep Neural Network*) dengan masing-masing hasil akurasi adalah 70.22% dan 64.41%. Hasil penelitian ini menunjukkan CNN memiliki akurasi yang lebih baik. Tugas akhir ini fokus kepada pengenalan aksara Jawa yang kemudian dapat diketahui cara baca dari teks aksara Jawa tersebut. Hasil penelitian ini diharapkan dapat menunjang keberlangsungan dan pelestarian kebudayaan sejarah bangsa Indonesia dan selain itu juga dapat digunakan oleh masyarakat untuk dapat membaca dan mengetahui sejarah Indonesia yang dituliskan dengan menggunakan Aksara Jawa.

1.2 Perumusan Masalah

Pengenalan teks tulisan tangan aksara Jawa ke huruf alfabet dilakukan sebagai penunjang digitalisasi bukti sejarah yang ada di Indonesia. Namun saat ini pengenalan masih dilakukan secara manual oleh para ahli Sastra Jawa. Sehingga diperlukan adanya sistem yang dirancang guna efisiensi dan efektifitas proses pengenalan Aksara Jawa.

1.3 Tujuan

Tujuan dilakukannya penelitian ini adalah untuk alih aksara dari aksara dasar huruf Jawa Mardi Kawi gambar hasil pemindaian sebuah naskah yang bertuliskan tangan aksara Jawa yang kemudian dapat diketahui cara bacanya dengan menggunakan bantuan CNN (*Convolutional Neural Network*) arsitektur *You Only Look Once* dan juga dilakukan analisis performansi kegunaan YOLO pada *multi-class* objek aksara Jawa. Adapun juga manfaat penelitian adalah:

1. Sebagai penunjang penelitian mengenai sejarah di Jawa di Indonesia.
2. Sebagai upaya pelestarian Aksara Jawa.
3. Mendukung masyarakat agar dapat membaca teks dalam Aksara Jawa.

1.4 Batasan Masalah

Batasan Masalah disusun agar penelitian dapat fokus hanya kepada permasalahan yang diangkat. Batasan-batasan permasalahan tersebut diantaranya adalah:

1. Kondisi naskah berisi tulisan tangan bertuliskan huruf Jawa Mardi Kawi yang diteliti dalam keadaan bagus, yaitu kalimat dalam naskah dengan keadaan baik dan hasil tulisan relatif jelas terbaca.
2. Penelitian memperhatikan kaidah aturan penulisan Aksara Jawa Mardi Kawi dengan judul yang telah ditetapkan yaitu *Tatacara Padmasusastra, Babad Tanah Jawi Meinsma, Serat Gancaran Warni-warni Ing Jaman Punika, Nutilen Parepatan Pangreh, dan Babad Ranggawarsita*
3. Hasil penelitian hanya sampai mengenali aksara Jawa dari pengelompokkan kata yang kemudian dapat dibaca dalam huruf Alfabet.

1.5 Sistematika Penulisan

Laporan penelitian Tugas Akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

1. **BAB I Pendahuluan**

Bab I berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan, dan metodologi penelitian.

2. **BAB II Dasar Teori**

Pada bab II berisi tentang uraian secara sistematika teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian. Teori-teori yang berkaitan dengan penelitian ini adalah terkait CNN, teori tentang *Handwriting Recognition*, kaidah aturan penulisan Aksara Jawa Mardi Kawi, dan teori-teori penunjang lainnya.

3. **BAB III Perancangan Sistem dan Implementasi**

Bab III berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah pengambilan data proses pengelompokkan aksara dan proses pengenalan aksara Jawa. Guna mendukung proses ini, digunakanlah blok diagram atau *work flow*.

4. **BAB IV Pengujian dan Analisa**

Bab keempat dijelaskan mengenai pengujian eksperimen yang dilakukan terhadap proses pengambilan data dari gambar, pengumpulan katanya serta pengenalan dari setiap aksara tersebut.

5. **BAB V Penutup**

Pada bab yang terakhir ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

BAB 2

DASAR TEORI

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Related Works

Pada sub bab *related works* dijelaskan mengenai beberapa penelitian yang berelasi dengan Tugas Akhir ini.

2.1.1 Related Works Terkait dengan Naskah

Penelitian Kiki Fatimah yang berjudul “*Brittle Ancient Document Using Adaptive Local Thresholding*”. Penelitian ini bertujuan untuk memperbaiki citra gambar naskah yang sedikit sulit terbaca. Penelitian ini menggunakan lima data uji dan metode yang digunakan adalah Metode Mean-C, Metode Sauvola, dan Metode Niblack.

2.1.2 Related Works Terkait dengan Segmentasi Tulisan Tangan

1. Proyek ini dilakukan oleh Agustinus Rudatyo H. Dilakukan contoh penerapan modifikasi *profile projection segmentation* gambar dokumen aksara Jawa dari seluruh 87 halaman buku Hamong Tani.
2. Pada dokumen teks miring terdapat ketidakkonsistenan antara spasi baris dan ukuran huruf Aksara Jawa disebabkan oleh huruf-huruf pasangan, vokal terakhir dan huruf konsonan dalam satu fonem dilakukan penelitian transformasi Hough untuk memprediksi tepi area teks. Berdasarkan hasil pengujian yang berjudul “*Hanacaraka Characters Using Double Projection Profile and Hough Transform*” oleh Lilana,dkk dilakukan metode profil proyeksi.

3. Penelitian SSD (*Single-Shot Multibox Detector*) berjudul “*Chinese Character Detection Using Modified Single Shot Multibox Detector*” yang dilakukan oleh Junhwan Ryu and Sungho Kim. Penelitian dilakukan dengan memodifikasi yang kemudian digunakan untuk mendeteksi *dataset* karakter huruf Cina. Digunakan Resnet dengan jaringan VGG pada jaringan dasar. Aspek rasio dan juga *bounding box* diatur *default* oleh SSD, dan juga mengecualikan lapisan terakhir dan diprediksi.

2.1.3 Related Works Terkait dengan CNN Aksara Jawa

1. Penelitian yang berjudul “*Deep Learning for Handwriting Javanese Character Recognition*” dilakukan oleh Khadijah Rismiyati. Membandingkan pengenalan aksara Jawa yang terdiri dari 20 aksara dengan media berupa kertas menggunakan 2 metode yaitu CNN dan DNN (*Deep Neural Network*).
2. Penelitian Chandra Kusuma D. yang berjudul “*Convolutional Neural Networks for Handwritten Javanese Character Recognition*”. Digunakan modul CNN selanjutnya melakukan proses klasifikasi terhadap citra yang telah disegmentasi ke dalam 20 kelas. Untuk evaluasi, kinerja CNN dibandingkan dengan kinerja dari model Multilayer Perceptron (MLP) dari sisi akurasi klasifikasi dan waktu latih.
3. Secara khusus pada penelitian yang berjudul “*CNN Based Common Approach to Handwritten Character Recognition of Multiple Scripts*” yang dilakukan oleh Dujoy Sen M, CNN dari penelitian ini dilatih menggunakan sampel dari basis data standar karakter dasar Bangla 50-kelas dan fitur-fitur telah diekstraksi untuk 5 perbedaan masalah pengenalan angka bahasa Inggris, Devanagari, Bangla, Telugu dan Oriya yang masing-masing adalah aksara India resmi. Pendekatan yang diusulkan untuk pengenalan karakter tulisan tangan telah diuji pada 6 database karakter yang berbeda.

2.1.4 Related Works Terkait dengan Pengenalan Aksara Jawa

1. Penelitian yang berjudul “Pengenalan Aksara Jawa Menggunakan *Digital Image Processing*” yang diteliti oleh Ratna Yani, dilakukan 2 tahapan percobaan penelitian, percobaan 1 bertujuan untuk mendeteksi tulisan tangan aksara Bali dengan membuat aplikasi berbasis OCR (Optical Character Recognition). Dengan menggunakan metode jaringan saraf tiruan (*contour analysis*).
2. Penelitian berjudul “*Javanese Character Recognition Using Hidden Markov Model*” yang dilakukan oleh Anastasia Rita W, bereksperimen secara off-line untuk 20 jenis legena, dengan 50 sampel untuk setiap karakter. Hidden Markov Model (HMM) adalah metode stokastik yang telah digunakan dalam berbagai pemrosesan sinyal dan pengenalan karakter. Penelitian ini mengusulkan untuk menggunakan HMM untuk mengenali karakter Jawa dari sejumlah tulisan tangan yang berbeda, di mana HMM digunakan untuk mengoptimalkan jumlah ekstraksi keadaan dan fitur.
3. Pengenalan Pola Tulisan Tangan Aksara Jawa “ha na car a ka” Menggunakan Multilayer Perceptron yang dilakukan oleh Madha Christian, meneliti dengan menggunakan *Optical character recognition* untuk Aksara Jawa dapat dilakukan dengan metode pengenalan pola menggunakan MLP dan *back propagation*.

Keberhasilan beberapa penelitian terkait yang dijelaskan, dalam melakukan pengenalan karakter dan juga identifikasi huruf Jawa, melatarbelakangi penelitian dengan menggunakan CNN dengan arsitektur YOLO sebagai proses pengenalan aksara Jawa pada media kertas.

2.2 Aturan Penulisan Aksara Jawa Mardi Kawi

Tulisan Jawa dan Bali adalah perkembangan modern aksara Kawi salah satu turunan aksara Brahmi yang berkembang di pulau Jawa. Pada masa periode Hindu-Buddha, aksara tersebut terutama digunakan dalam literatur keagamaan dan terjemahan Sanskerta yang biasa ditulis dalam naskah daun lontar. Pada abad ke-17, tulisan tersebut telah berkembang menjadi bentuk modernnya dan dikenal sebagai *Carakan*[7] atau Hanacaraka berdasarkan lima aksara pertamanya *Carakan* terutama digunakan oleh penulis dalam lingkungan kraton kerajaan Surakarta dan Yogyakarta untuk menulis naskah berbagai subjek, diantaranya antaranya cerita-cerita (*serat*), catatan sejarah (*babad*), tembang kuno (*kakawin*), atau ramalan (*primbon*). Subjek yang populer akan berkali-kali ditulis ulang[8]. Hiasan dapat berupa tanda baca yang sedikit dilebih-lebihkan atau pigura halaman (disebut *wadana*) yang rumit dan kaya warna. Keberadaan aksara Jawa dengan tata penulisan yang beragam pernah terjadi di Indonesia. Terdapat beberapa sejarah pedoman penulisan aksara Jawa, salah satunya adalah pedoman penulisan aksara Jawa Mardi Kawi[9]. Munculnya pedoman Mardi Kawi paling sebagai pijakan untuk bisa memahami tata tulis dalam kedua karya sastra Kakawin dan Kidung. Pedoman ini ditulis dan diresmikan penggunaannya tahun 1860, oleh W.J.S Poerwadarminta.

2.2.1 Aksara Wyanjana atau aksara Konsonan

Pada dasarnya terdapat beberapa pembagian kelompok jenis aksara dalam aksara Jawa. Yang pertama adalah aksara konsonan, Pada aturan penulisan aksara Jawa Mardi Kawi, aksara konsonan disebut dengan aksara wyanjana. Terdapat sejumlah 33 jenis aksara konsonan pada aturan penulisan Mardi Kawi. Aksara nya adalah ka, kha, ga, gha, nga, ca, cha, ja, jha, ña, ya, śa, ṭa, ṭha, ḍa, dha, ṇa, ṇa, ra, ṣa, pa, pha, ba, bha, ma, wa, ha. Aksara ini sedikit berbeda dengan aksara konsonan yang biasanya dipelajari saat murid Indonesia duduk di bangku Sekolah Dasar. Karena aksara ini biasanya digunakan pada masa era jaman tahun 1800 hingga

1930. Gambar 2.1 berisikan daftar dari aksara inti atau aksara konsonan. Sedangkan aksara swara dan aksara sandhangan ditunjukkan pada Gambar 2.2 dan Gambar 2.3.

PEMBAGIAN KELOMPOK WARGA AKSARA WYANJANA								
	Pañcawalmukha					Ardha- swara	Usma	Wisarga
	Aghosa		Ghosa		Anu- swara			
	Alpa- prana	Maha- prana	Alpa- prana	Maha- prana				
Kañtya	𑀓𑀭 ka	𑀓𑀭𑀮 kha	𑀓𑀭𑀮𑀭 ga	𑀓𑀭𑀮𑀭𑀮 gha	𑀓𑀭𑀮𑀭𑀮𑀭 nga			𑀓𑀭𑀮𑀭𑀮𑀭𑀮 ha
Talawya	𑀓𑀭𑀮𑀭𑀮 ca	𑀓𑀭𑀮𑀭𑀮𑀮 cha	𑀓𑀭𑀮𑀭𑀮𑀭𑀮 ja	𑀓𑀭𑀮𑀭𑀮𑀭𑀮𑀮 jha	𑀓𑀭𑀮𑀭𑀮𑀭𑀮𑀮𑀮 ña	𑀓𑀭𑀮𑀭𑀮𑀭𑀮𑀮𑀮𑀮 ya	𑀓𑀭𑀮𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮 śa	
Murdhanya	𑀓𑀭𑀮𑀭𑀮𑀮 ṭa	𑀓𑀭𑀮𑀭𑀮𑀮𑀮 ṭha	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮 ḍa	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮 ḍha	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮 ṇa	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮 ra	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮𑀮 ṣa	
Dentya	𑀓𑀭𑀮𑀭𑀮𑀮𑀮 ta	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮 tha	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮 da	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮 dha	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮 na	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮𑀮 la	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮𑀮 sa	
Osthya	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮 pa	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮 pha	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮 ba	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮 bha	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮𑀮 ma	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮𑀮𑀮𑀮𑀮 wa		

Gambar 2.1: Aksara Konsonan Jawa Mardi Kawi

2.2.2 Aksara Swara

Aksara Swara 14 aksara							
hrswara pendek	swara						
𑀓𑀭𑀮𑀭𑀮𑀮𑀮	a	e/é	i	u	o	rê	lê
	𑀓𑀭𑀮𑀭𑀮𑀮𑀮						
dirgha panjang	swara						
𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮	ā	ai	ī	ii	au	rêu	lêu
	𑀓𑀭𑀮𑀭𑀮𑀮𑀮𑀮						

Gambar 2.2: Aksara Swara Jawa Mardi Kawi

Sandhangan Mandaswara 3 aksara		
Mandaswara (semi vokal)	Visarga	Anuswara
𑀓𑀭𑀮𑀭𑀮𑀮𑀮 am̐	wignyan [h]	cecak [ŋ]
	𑀓𑀭𑀮𑀭𑀮𑀮𑀮	.

Gambar 2.3: Aksara Sandhangan

2.3 *Artificial Neural Networks*

Artificial Intelligence (kecerdasan buatan) atau biasa disebut AI ditujukan untuk menciptakan suatu proses komputasi serba bisa diadaptasi dari kemampuan cara kerja otak manusia. Beberapa hasil riset yang merupakan cabang AI antara lain *Knowledge Based System* (sistem berbasis pengetahuan), *Artificial Neural Networks* (jaringan saraf tiruan), *Fuzzy Logic* (logika fuzzy), dan *Genetic Algorithm* (algoritma genetika).

Dalam AI terdapat 2 jenis kategori pembelajaran[10]: yaitu:

1. *Supervised Learning*

Supervised Learning merupakan metode pembelajaran yang dimana sudah ada data yang akan dilatih dan juga telah terdapat *variable* yang ditargetkan, sehingga tujuan dari pembelajaran ini adalah model prediksi pengelompokkan suatu data ke data yang telah ada. Contoh dari *Supervised Learning* antara lain adalah *Decision Tree*, *Nearest Neighbour Classifier*, *Support Vector Machine*, dan *Artificial Neural Networks*.

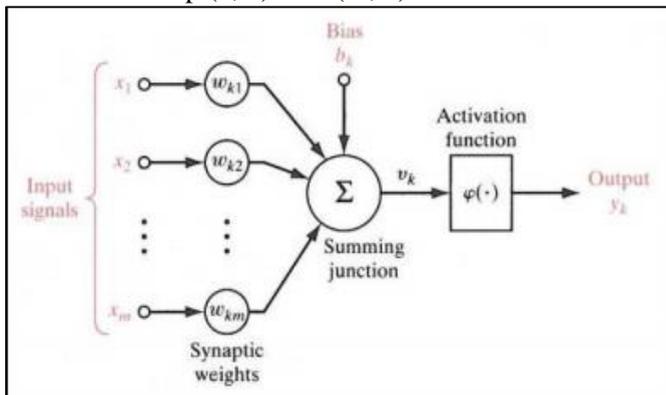
2. *Unsupervised Learning*

Unsupervised Learning merupakan metode pembelajaran yang tidak memiliki data latih sehingga perlu dilakukan analisis dan klasifikasi data terlebih dahulu. Contoh dari *Unsupervised Learning* antara lain adalah K-Means, Hierarchical Clustering, DB-SCAN, Fuzzy C-Means, dan Self Organizing Map.

Ada beberapa metoda *Supervised Learning* diantaranya adalah perseptron dan *backpropagation* (BP). Perseptron menghitung jumlah nilai perkalian penimbang dan masukan dari parameter permasalahan yang kemudian dibandingkan dengan nilai *threshold*. Bila nilai keluaran lebih besar dari *threshold* maka keluarannya adalah satu, sebaliknya adalah nol. *Artificial Neural Networks* (ANN) atau Jaringan Saraf Tiruan adalah salah satu sistem komputer cerdas yang erat keterkaitannya dengan AI. ANN menggunakan struktur pemrosesan data seperti struktur jaringan otak manusia. Secara umum jaringan saraf manusia

terbentuk dari satu trilyun (bahkan lebih) struktur dasar neuron yang terkoneksi dan terintegrasi antara satu dengan yang lainnya sehingga dapat melaksanakan aktivitas menyimpan (*memorize*) pengetahuan (*knowledge*) secara teratur dan terus menerus sesuai dengan kebutuhan. *Neuron*, adalah satuan unit pemroses terkecil pada otak. ANN terdiri atas lapisan masukan (*input layer*) dan lapisan keluaran (*output layer*). *Neuron* pada ANN adalah unit pengolahan informasi dasar bagi operasi ANN. Setiap layer terdiri atas satu atau beberapa unit neuron yang mempunyai sebuah fungsi aktivasi yang nanti akan menentukan *output* dari unit tersebut. Pada tahun 1958, Frank Rosenblatt mengenalkan ANN ini sebagai *perceptron*. Gambar 2.4 menjelaskan contoh proses singkat dari *perceptron* pada ANN. Terdapat 3 elemen dasar dari neuron atau *perceptron* yaitu:

1. Sinapsis atau penghubung yang dikategorikan berdasarkan bobot atau kekuatannya
2. *Adder* yang berfungsi untuk menjumlahkan sinyal *input* yang telah diberikan bobot oleh sinapsis.
3. *Activation function* yang berfungsi sebagai pembatas amplitudo *output* dari sebuah *perceptron*. Biasanya batas amplitudo normal dari output ditulis dalam bentuk interval satuan tertutup (0, 1) atau (-1, 1).

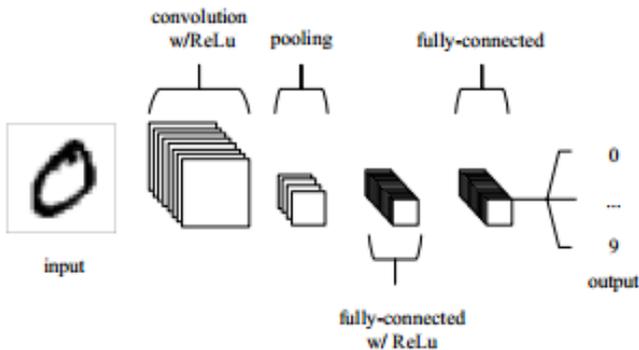


Gambar 2.4: Operasi Perceptron pada ANN

ANN bisa dilatih dengan menggunakan data *training*. Semakin banyak data *training* maka semakin baik performa dari ANN tersebut. Semakin banyak jumlah layer maka semakin tinggi kapasitas ANN tersebut. Semakin banyak layer juga mengakibatkan kelemahan yaitu semakin banyaknya jumlah iterasi atau *training* data yang dibutuhkan.

2.3.1 Convolutional Neural Networks

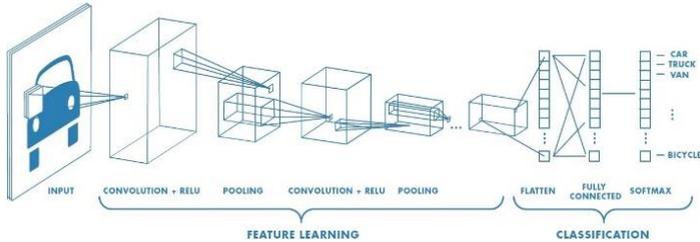
Convolutional Neural Networks atau CNN's adalah variasi dari *Multilayer Perceptron* yang terinspirasi dari jaringan syaraf manusia[11]. CNN secara khusus digunakan untuk menyelesaikan hal-hal khusus seperti *pattern recognition* yang berdasarkan arsitektur ANN. Gambar 2.5 menjelaskan contoh sederhana dari proses CNN yang mendeteksi angka dengan *input* data adalah sebuah gambar tulisan dari angka 0.



Gambar 2.5 An simple CNN architecture

Dari input gambar vector gambar asli akan diproses menggunakan CNN menghasilkan *output* dari gambar. CNN merupakan suatu *layer* yang memiliki susunan *neuron* 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran *layer* sedangkan kedalaman mengacu pada jumlah *layer*. Harus diingat bahwa perbedaan dari CNN dan ANN adalah CNN secara khusus digunakan untuk mempelajari mengenai *pattern recognition* dengan images sebagai

input data. Gambar 2.6 menunjukkan ilustrasi dari proses *learning* dan *classification* pada sistem *Convolutional Neural Networks*.



Gambar 2.6 *Convolutional Neural Networks* (Sumber: Matlab)

2.3.2 Convolutional Layer

Seperti CNN, *Convolutional Layer* juga memiliki hal-hal spesifik seperti cara kerja dari CNN dengan beberapa layer yang beroperasi yaitu[12]:

1) Zero Padding

Padding atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *convolutional layer* (*Feature Map*). Tujuan dari penggunaan *zero padding* adalah:

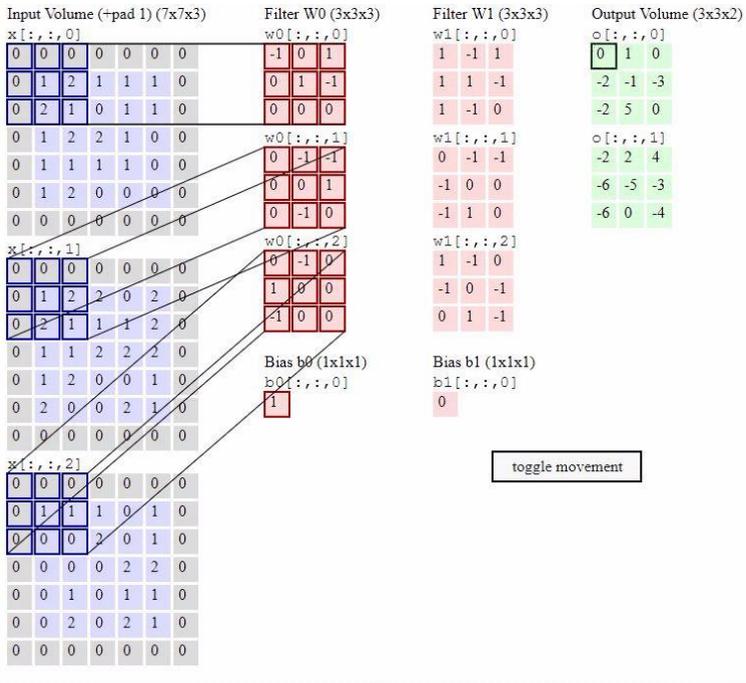
- I. Dimensi *output* dari *convolutional layer* selalu lebih kecil dari inputnya (kecuali penggunaan 1x1 filter dengan stride 1). *Output* ini akan digunakan kembali sebagai *input* dari *convolutional layer* selanjutnya, sehingga makin banyak informasi yang terbuang. Dengan menggunakan *padding*, kita dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis. Sehingga kita bisa menggunakan *convolutional layer* yang lebih dalam (*deep*) sehingga lebih banyak *features* yang berhasil di-*extract*.
- II. Meningkatkan performa dari model karena *convolutional filter* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara *zero padding* tersebut. Untuk menghitung

dimensi dari feature map menggunakan rumus seperti Persamaan 2.1:

$$output = \frac{W-N+2P}{S} + 1 \quad (2.1)$$

2) *Stride*

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah 1, maka *convolutional layer* akan bergeser sebanyak 1 *pixels* secara *horizontal* lalu *vertical*. Pada ilustrasi Gambar 2.7 *stride* yang digunakan adalah 2. Semakin kecil *stride* maka akan semakin detail informasi yang kita dapatkan dari sebuah *input*.



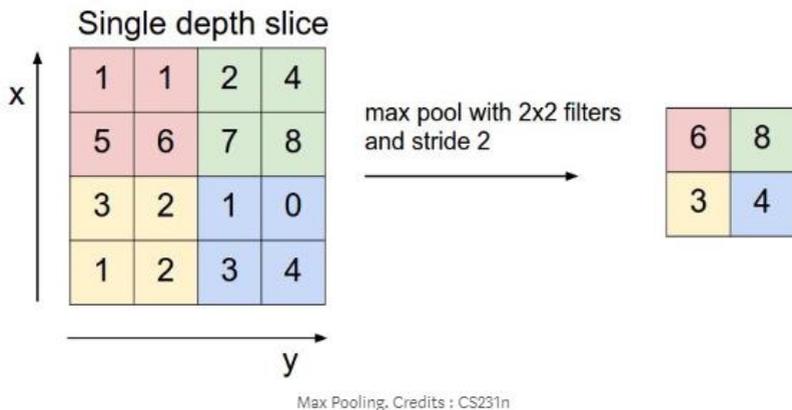
Gambar 2.7 Proses *Stride*

3) *Loss Function*

Loss function merupakan fungsi yang menggambarkan kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh sebuah model. *Loss function* ini bekerja ketika model pembelajaran memberikan kesalahan yang harus diperhatikan. *Loss function* yang baik memberikan nilai error yang terendah dari sebuah pembelajaran.

2.3.3 *Pooling Layer*

Pooling layer biasanya berada setelah *convolutional layer*. Pada prinsipnya *pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. *Pooling* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Sebagai contoh pada Gambar 2.8, jika digunakan *Max Pooling* 2x2 dengan *stride* 2, maka pada setiap pergeseran filter, nilai maximum pada area 2x2 pixel tersebut yang akan dipilih, sedangkan *Average Pooling* akan memilih nilai rata-ratanya.



Gambar 2.8 *Max Pooling*

Tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat

komputasi karena parameter yang harus di-*update* semakin sedikit dan mengatasi *overfitting*. *Overfitting* adalah keadaan dimana data yang digunakan untuk pelatihan itu adalah data yang “terbaik”. Sehingga apabila dilakukan tes dengan menggunakan data yang berbeda dapat mengurangi akurasi (hasil yang dibuat tidak sesuai yang diharapkan)

2.3.4 Activation Function

Activation function atau fungsi aktivasi berfungsi untuk menentukan apakah neuron harus “aktif” atau tidak berdasarkan penjumlahan *weight* dan *input*. Secara umum terdapat 2 jenis *activation function*, yaitu *Linear* dan *Non-Linear Activation Function*. Pada CNN, fungsi aktivasi terletak pada perhitungan akhir dari *output feature map*, sesudah proses perhitungan *pooling*, untuk menghasilkan suatu pola *feature*. Beberapa contoh fungsi aktivasi yang sering digunakan dalam riset antara lain fungsi sigmoid, fungsi tanh, dan *Rectified Linear Unit* (RELU).

2.4 Arsitektur Convolutional Neural Networks

2.4.1 Feature Extraction Layer

Gambar 2.9 adalah RGB (Red, Green, Blue) image berukuran 32x32 pixels yang sebenarnya adalah *multidimensional* array dengan ukuran 32x32x3 (3 adalah jumlah channel). Convolutional layer terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels). Sebagai contoh, layer pertama pada feature extraction layer biasanya adalah *convolutional layer* dengan ukuran 5x5x3. Panjang 5 pixels, tinggi 5 pixels dan tebal/jumlah 3 buah sesuai dengan channel dari image tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi *dot* antara input dan nilai dari filter tersebut sehingga menghasilkan sebuah output atau biasa disebut sebagai *activation map* atau *feature map*.

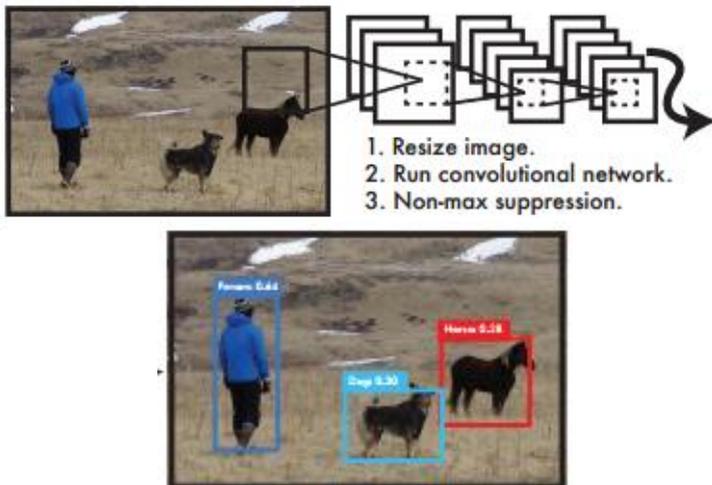
2.4.2 *Fully Connected Layer*

Fully connected layer menghubungkan seluruh neuron ke neuron lain pada layer yang berdekatan. Karena *feature map* yang dihasilkan dari proses sebelumnya masih berbentuk multidimensional array maka *feature map* harus terlebih dahulu dilakukan proses *flatten* atau *reshape feature map* menjadi sebuah *vector* sebelum dijadikan *input* dari proses *fully connected layer*. Prosesnya melakukan *training* serta *testing* dan juga memiliki *hidden layer*, *activation function*, *output layer*, dan *loss function*. Namun perbedaan antara *fully connected layer* dan *convolutional layer* adalah neuron yang berada di *convolutional layer* terhubung hanya ke daerah tertentu pada *input*, sedangkan pada *fully connected layer* memiliki *neuron* yang secara keseluruhan telah terhubung.

2.4.3 *YOLO (You Only Look Once)*

YOLO merupakan metode pendekatan untuk object detection yang dikembangkan oleh Joseph Redmon, dkk[13]. Ide dari object detection dilatarbelakangi oleh pandangan sekilas manusia yang dapat mengenali gambar atau object dengan instan, mengetahui apa gambar tersebut dan dimana lokasinya. Dibandingkan melakukan hal yang sama YOLO menganggap object detection sebagai permasalahan regresi tunggal untuk diselesaikan. Model dasar dari YOLO memproses gambar pada real-time pada 45 frame per detik, sementara versi lebih kecilnya yakni FAST YOLO memproses 155 frame per detik sambil tetap mencapai mAP dua kali lipat dari detector real time lainnya. Dibandingkan dengan sistem deteksi paling baru YOLO membuat lebih banyak kesalahan lokalisasi tetapi YOLO memiliki kemungkinan yang lebih kecil untuk salah memprediksi. Sistem pada YOLO mengubah ukuran gambar input menjadi 448 x 448. Kemudian convolutional layer tunggal secara bersamaan memprediksi bounding box dan probabilitas kelas dan masing-masing box. YOLO mendeteksi obyek dengan menggunakan *unfied model* dimana sebuah single convolutional network memprediksi beberapa bounding boxes (kotak pembatas) serta probabilitas kelas di dalam kotak-kotak tersebut secara

bersamaan. Pertama-tama, sistem YOLO membagi citra input ke dalam grid $S \times S$. Jika pusat dari sebuah objek jatuh di dalam salah satu sel grid, maka sel grid itu bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel grid memprediksi *bounding boxes* B dan *confidence score* dari tiap *bounding box* tersebut. *Confidence score* merefleksikan seberapa yakin dan akurat model bahwa terdapat sebuah objek di dalam kotak tersebut. Setiap *bounding box* terdiri 5 prediksi: x , y , w , h , dan *confidence*. Koordinat (x, y) mewakili pusat dari kotak relatif ke batas *cell grid* (w, h) atau lebar dan tinggi mewakili pusat dari kotak relatif ke gambar. Dan terakhir adalah *confidence* yang mewakili *Intersection over Union* (IoU) antara kotak prediksi dan kotak *ground-truth*. Setiap sel grid juga memprediksi probabilitas kelas C . Gambar 2.9 menjelaskan mengenai proses deteksi dari YOLO. Probabilitas dikondisikan pada sel grid yang memuat objek dan hanya satu kelas probabilitas yang dideteksi per sel grid tanpa memperhitungkan jumlah *bounding boxes* B .



Gambar 2.9: Sistem pendeteksian YOLO

Ketika melakukan pendeteksian, probabilitas kelas dikalikan dengan confidence pada kotak prediksi sesuai dengan Persamaan 2.2.

$$Pr(Class_i|Object) \times Pr(Object) \times IOU_{pred}^{truth} \quad (2.2)$$

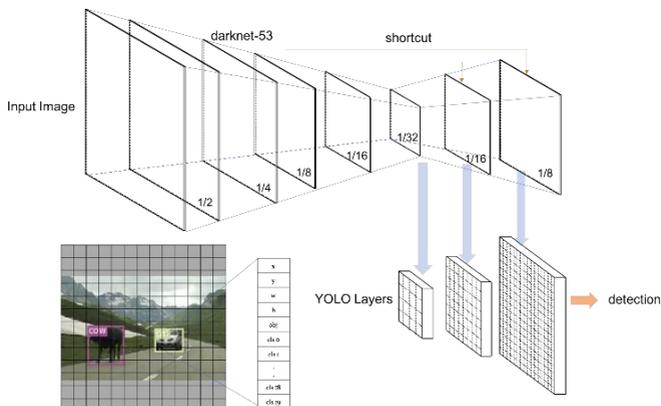
Akan yang dapat disederhanakan menjadi persamaan 2.3

$$Pr(Class_i) \times IOU_{pred}^{truth} \quad (2.3)$$

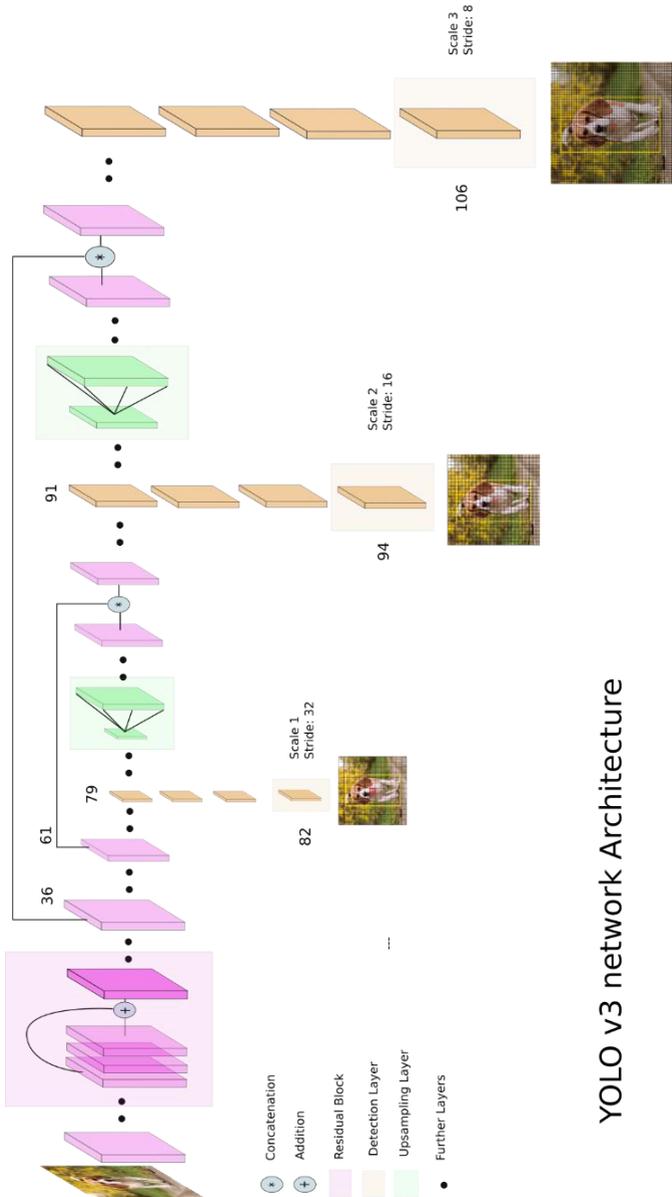
Skor dari persamaan 2.3 dapat menyimpan probabilitas dari kelas yang muncul di dalam kotak dan seberapa baik kotak prediksi tersebut menyesuaikan objek. Sistem deteksi model yang digunakan adalah penyelesaian regresi. Proses ini membagi gambar menjadi kotak-kotak $S \times S$ untuk setiap sel kisi prediksi *bounding boxes* B, konsistensi dari setiap *boxes*, dan probabilitas dari class C. Prediksi-prediksi ini dikodekan dengan rumus pencarian filters yaitu dengan Persamaan 2.4:

$$S \times S \times (B * 5 + C) \quad (2.4)$$

Arsitektur jaringan dari YOLOv3 dan juga *channel elements* dari YOLO layers (*detection layer*) untuk *multi-scale object detection* digambarkan seperti seperti Gambar 2.11.



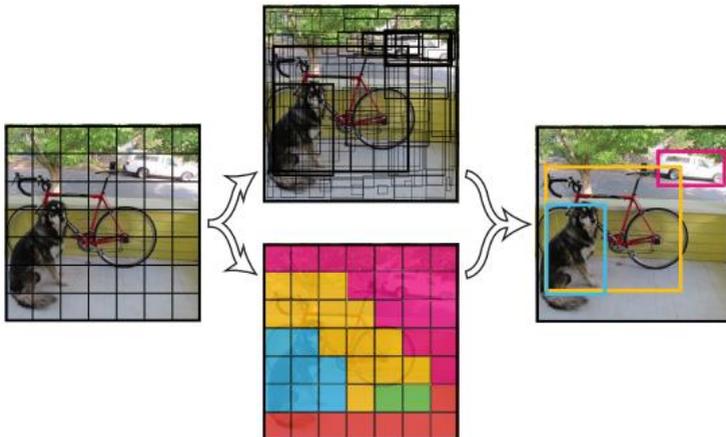
Gambar 2.10 Skematik Arsitektur YOLOv3 *multi-scale detector*



YOLO v3 network Architecture

Gambar 2.12 Diagram contoh YOLOv3 Network Architecture

Proses analisis dilakukan dari pendeteksian objek oleh sistem. Sistem pendeteksi objek akan diberikan hasil berupa *confidence*, *class*, dan jumlah *predicted bounding box*. YOLO menyatukan komponen-komponen terpisah dari *object detection* ke dalam sebuah *neural network* tunggal. YOLO menggunakan fitur dari keseluruhan gambar untuk memprediksi semua *bounding boxes* untuk gambar secara keseluruhan. Ini artinya *network* mempertimbangkan secara global tentang gambar penuh dan semua objek yang terdapat pada gambar. Model pendeteksian pada YOLO bekerja dengan membagi gambar *input* kedalam *grid* berukuran $S \times S$. Masing-masing *grid cell* memprediksi *bounding box* B dan *confidence scores* untuk *boxes* tersebut. *Confidence scores* merefleksikan seberapa tingkat keyakinan model bahwa *box* tersebut berisi objek dan seberapa akurat prediksi yang dilakukan menurut perkiraan model. Secara formal *confidence* didefinisikan sebagai $\Pr(\text{Object}) * IOU_{pred}^{truth}$. Jika tidak ada objek pada cell tersebut maka *confidence score* nya adalah nol. Model ini ditunjukkan oleh Gambar 2.13. Setiap *bounding box* terdiri dari lima prediksi : x , y , w , h , dan *confidence*.



Gambar 2.13 Model Pendeteksian pada YOLO

Koordinat x,y merepresentasikan pusat dari *box* yang relatif terhadap batas dari *grid*. w,h merupakan prediksi relatif terhadap keseluruhan gambar. *Confidence* merepresentasikan *intersection over union* (IOU) antara *box* prediksi dan *ground truth box*. Setiap *grid cell* juga memprediksi probabilitas kelas kondisional C , $\Pr(Class_i|Object)$. Sistem YOLO hanya memprediksi satu probabilitas per *grid cell*, terlepas dari banyaknya *boxes* B .

Pada pengujian, sistem mengalikan probabilitas kelas dan *confidence* dari prediksi pada *box*, yang memberikan skor *confidence* (keyakinan) kelas khusus untuk setiap *box*. Skor ini mengkodekan probabilitas kelas yang muncul di dalam *box* dan seberapa baik *box* yang diprediksi cocok dengan objek.

$$\begin{aligned} & \Pr(Class_i|Object) * \Pr(Object) * IOU_{pred}^{truth} \\ & = \Pr(Class_i) * IOU_{pred}^{truth} \end{aligned} \quad (2.5)$$

Optimasi dilakukan menggunakan *sum-squared error* karena teknik ini mudah untuk dioptimasi. Untuk mengatasi *error* yang ditimbulkan karena banyaknya *grid cell* pada gambar yang tidak berisi objek, maka YOLO meningkatkan *loss* dari prediksi koordinat *bounding box* dan mengurangi *loss* dari keyakinan prediksi untuk *boxes* yang tidak berisi objek.

Optimasi dilakukan menggunakan *sum-squared error* karena teknik ini mudah untuk dioptimasi. Untuk mengatasi *error* yang ditimbulkan karena banyaknya *grid cell* pada gambar yang tidak berisi objek, maka YOLO meningkatkan *loss* dari prediksi koordinat *bounding box* dan mengurangi *loss* dari keyakinan prediksi untuk *boxes* yang tidak berisi objek. Digunakan dua parameter untuk menyelesaikan hal tersebut, yaitu $\lambda_{coord} = 5$ dan $\lambda_{noobj} = 0.5$. YOLO memprediksi banyak *bounding box* per *grid cell*. Pada waktu *training*, diharapkan hanya satu prediktor *bounding box* untuk bertanggung jawab untuk setiap objek. Sistem menetapkan satu prediktor untuk “bertanggung jawab” memprediksi sebuah objek berdasarkan prediksi mana yang memiliki IOU tertinggi saat ini dengan *ground truth*. Hal tersebut

mengarahkan pada spesialisasi antara prediktor *bounding box*. Setiap prediktor menjadi lebih baik dalam memprediksi ukuran tertentu, aspek rasio, atau kelas objek, meningkatkan daya ingat keseluruhan. Selama *training* YOLO melakukan optimasi, seperti pada Persamaan 2.6 multi-part loss function adalah:

$$\begin{aligned}
& \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\
& + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \\
& + \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \\
& + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{s^2} 1_i^{obj} \sum_{c \in classes} ((p_i(c) + \hat{p}_i(c))^2 \tag{2.6}
\end{aligned}$$

Dimana, 1_i^{obj} menunjukkan jika objek muncul pada *cell* i , dan 1_{ij}^{obj} menunjukkan bahwa *predictor bounding box* j dalam *cell* i adalah “bertanggung jawab” untuk memprediksi *cell* tersebut.

BAB 3

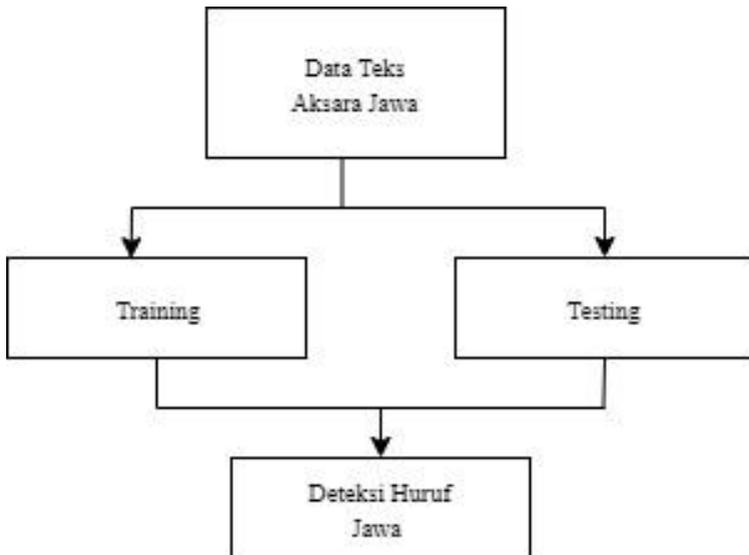
DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem juga dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

3.1 Desain Sistem

Tugas akhir ini merupakan penelitian dalam bidang visi komputer yang bertujuan untuk mendeteksi tulisan tangan aksara Jawa dengan menggunakan sistem pendeteksi berbasis *Convolutional Neural Networks (CNN)*. Berdasarkan diagram alir pada Gambar 3.1, proses pertama yang dilakukan adalah penentuan data *input* naskah teks bertuliskan Aksara Jawa. Tujuan dari tahap ini adalah untuk memastikan data file gambar yang akan diproses dalam keadaan baik dan relative dapat dibaca. Tahap yang selanjutnya yaitu melakukan persiapan penyusunan data yang akan dilakukan pada tahap proses *training*. Data gambar yang telah dipilih pada tahap pertama kemudian diberikan label atau anotasi berupa *ground-truth bounding box* yang menunjukkan dimana posisi objek tiap aksara Jawa serta kelas (*class*) aksara yang berada file gambar. Setelah memperoleh *ground-truth bounding box*, proses selanjutnya adalah ekstraksi *feature* dan klasifikasi objek yang telah terlabeli dengan menggunakan Darknet53 sebagai *feature extractor*. Pada tahap ini disebut sebagai tahap *training*, *output* proses *training* adalah mengetahui bahwa hasil *training* yang dilakukan mendapatkan *output weight* yang dibutuhkan oleh YOLOv3 untuk mendeteksi tiap *bounding box* pada objek yakni aksara Jawa. Setelah mendapatkan *final-weight* maka dilakukan tahapan *testing*. Tahap *testing* dilakukan dengan cara memproses hasil *weight* yaitu hasil *training dataset* yang kemudian hasil prediksi *bounding box* dapat terlihat di gambar *test*. Tahap terakhir

adalah dapat diketahui hasil deteksi gambar yang kemudian dapat dianalisis keakuratan sistem yang digunakan. Gambar 3.1 adalah blok diagram alir dari sistem yang telah dirancang.



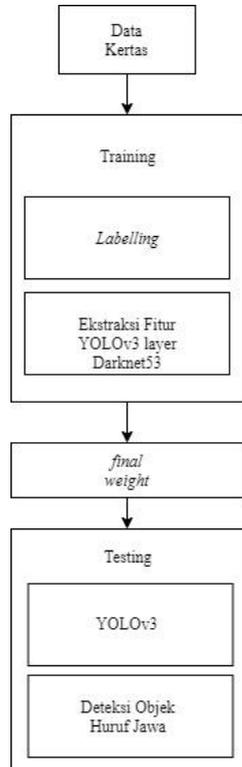
Gambar 3.1 Blok Diagram Alir Sistem

3.2 Alur Kerja

Alur implementasi dalam pengerjaan penelitian tugas akhir ini dapat dibagi menjadi beberapa tahapan berdasarkan metodologi penelitian yang telah dilakukan oleh penulis sebagai penunjang terlaksananya penelitian. Alur kerja yang telah dilakukan dapat diuraikan secara sederhana yaitu:

1. Pembuatan Skenario Pengujian
2. Pelabelan Aksara Jawa
3. *Training* dan *Testing*
4. Pendeteksian Aksara Jawa dan Analisis Hasil

Gambar 3.2 adalah blok diagram alir secara keseluruhan. Pada penelitian ini dilakukan 2 proses utama yaitu proses *training* dan *testing*.



Gambar 3.2 Diagram Alir Secara Keseluruhan

3.3 Pembuatan Skenario Pengujian

Penelitian tugas akhir ini dilakukan dua jenis percobaan pengujian dengan masing-masing pengujian terdapat dua perbedaan *dataset*. *Dataset* yang diteliti merupakan gambar yang berisikan tulisan

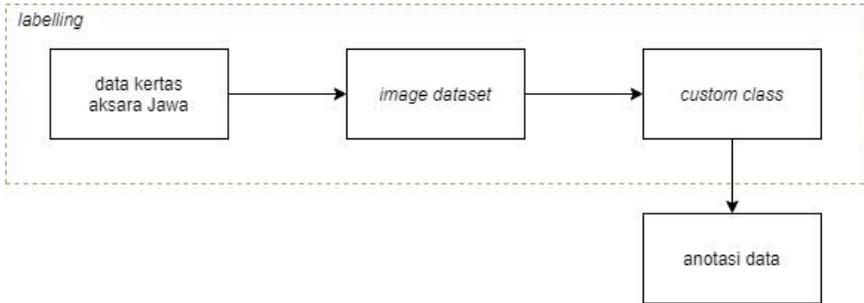
tangan aksara Jawa yang terdiri dari 5 judul dan penulis yang berbeda. Adapun rindian dari proses pengujian adalah:

1. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 33 jenis aksara Jawa Mardi Kawi menggunakan *dataset* dengan ukuran asli dari kertas.
2. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 47 jenis aksara Jawa Mardi Kawi menggunakan *dataset* dengan ukuran asli dari kertas.
3. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 33 jenis aksara Jawa Mardi Kawi menggunakan *dataset* kertas dengan membagi ukuran kertas menjadi 416x416.
4. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 47 jenis aksara Jawa Mardi Kawi menggunakan *dataset* kertas dengan membagi ukuran kertas menjadi 416x416.
5. Pengujian pembuktian jika gambar yang digunakan pada *dataset training* dapat digunakan dalam proses *testing* namun dengan gambar yang berbeda penulis dan judul tulisan.

3.4 Pelabelan Aksara Jawa

Tahap pelabelan atau anotasi objek merupakan proses pemberian informasi berupa kelas dan posisi dari objek yang akan dideteksi. Dari data hasil scan yang didapatkan, dilakukan pelabelan satu per satu terhadap objek, yaitu dalam penelitian ini adalah tiap aksara Jawa, sehingga didapatkan koordinat *ground-truth bounding box*.

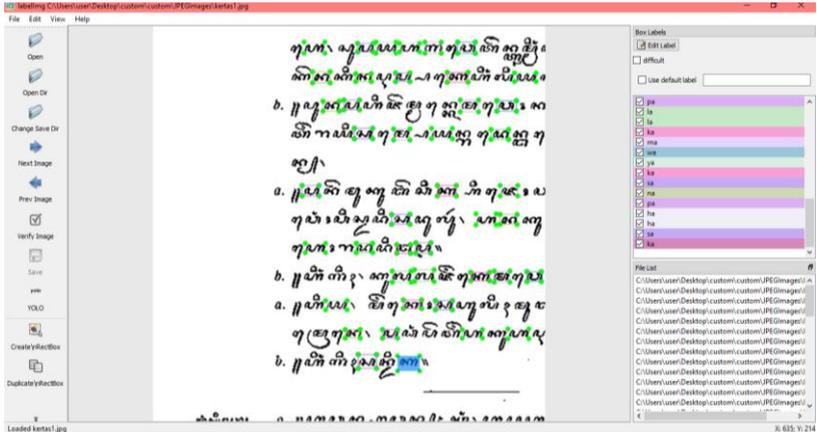
Pada Gambar 3.3 adalah salah satu contoh proses *labelling* yang dilakukan satu per-satu kepada setiap kertas yang terdapat objek aksara Jawa.



Gambar 3.3 Diagram Alir *labelling*

Proses pelabelan *bounding box* dilakukan satu per satu secara manual pada tiap gambar yang terdapat objek aksara Jawa di dalamnya. Informasi mengenai label yang telah dibuat kemudian disimpan dalam bentuk file TXT atau XML. Perlu diperhatikan ketelitian *labelling* yang harus sesuai dengan *class* yang diharapkan. Setelah dilakukan *labelling* didapatkan *output* yang dihasilkan yaitu berupa file .txt. YOLO menghasilkan prediksi berupa koordinat *bounding box* ($tx;ty;tw;th$), dan *class probability*. Penjelasan tx adalah nilai koordinat x , ty adalah koordinat y , tw adalah koordinat *width*, th adalah koordinat *height*, sedangkan *class probability* merupakan jenis kelas dari objek yang terdeteksi. File ini biasanya disebut dengan file anotasi.

Pada Gambar 3.4 ditunjukkan contoh hasil anotasi data untuk tiap kertas yang telah melalui proses *labelling*.



Gambar 3.4 Program labelImg.py untuk *Labelling*

Persamaan 3.1 adalah hasil dari proses anotasi data (*labelling*).

$$\langle \text{object-class} \rangle \langle x \rangle \langle y \rangle \langle \text{width} \rangle \langle \text{height} \rangle \quad (3.1)$$

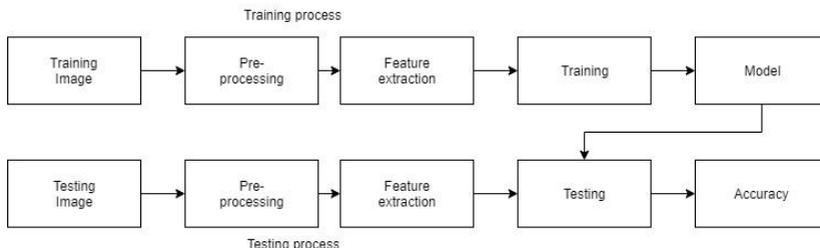
Sedangkan pada Gambar 3.4 ditunjukkan contoh hasil *file output* anotasi data untuk tiap kertas yang telah melalui proses *labelling*.



Gambar 3.5 *Output* Anotasi Data

3.5 Training dan Testing

Tahapan utama dari sistem pendeteksian yang menggunakan *Convolutional Neural Network* sebagai sistem pendeteksian objeknya. Gambar 3.6 menjelaskan tahapan yang diimplementasikan pada tugas akhir ini.



Gambar 3.6 Tahapan secara umum *pattern recognition*

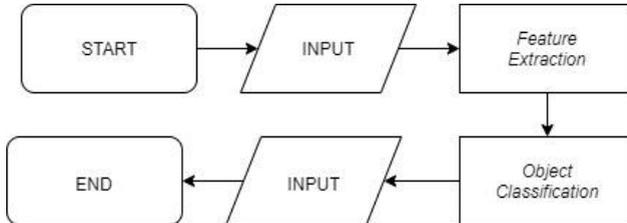
3.5.1 Training

Proses pendeteksian aksara Jawa dilakukan dengan menggunakan program yang telah dibuat sebelumnya dan kemudian dilakukan pengubahan parameter serta penambahan kode agar sesuai dengan kebutuhan analisis. Pada tahap pendeteksian aksara Jawa, data gambar dari hasil digunakan sebagai *input* yang kemudian dilewatkan pada sistem pendeteksi aksara Jawa berbasis CNN, yaitu YOLO.

a. YOLO (*You Only Look Once*)

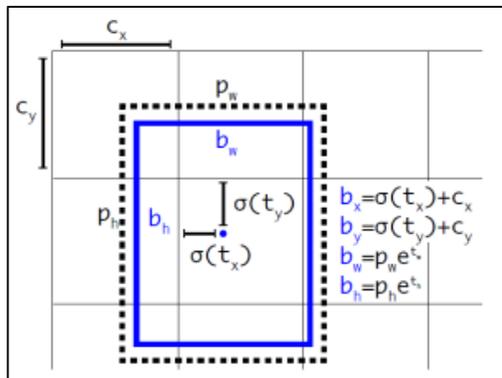
Output dari proses pendeteksian aksara Jawa berupa nama kelas serta posisi *bounding box* aksara Jawa yang terdeteksi oleh sistem. Pengolahan *input* menggunakan YOLOv3 sebagai acuan metodologi penelitian. Secara umum pada metode YOLO membutuhkan *input* gambar dengan ukuran yaitu 416×416 piksel, yang masih dapat diubah selama masih ada di dalam kelipatan 32.

Terdapat 3 aspek penting dalam pendeteksian objek yang dilakukan oleh CNN, yaitu *input*, *feature extraction*, dan juga *object classification*. Gambar 3.7 menjelaskan diagram alir secara umum proses pendeteksian objek oleh CNN.



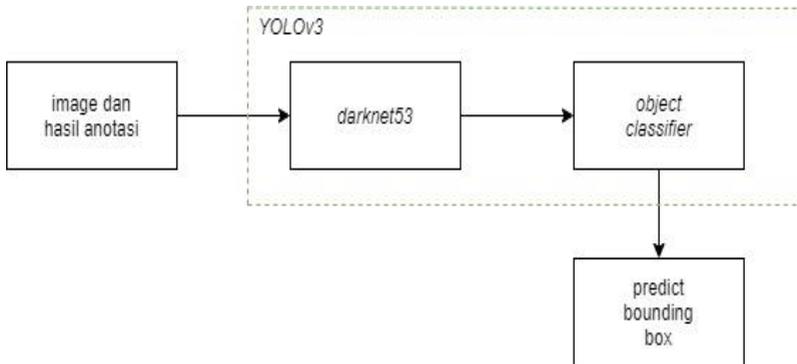
Gambar 3.7 Diagram Alir Secara Umum Pendeteksian Objek oleh CNN

YOLO membuat *bounding box* pada lokasi *feature map* dengan menggunakan *logistic activation sigmoid*. Pengolahan *bounding box* dari hasil deteksi diilustrasikan pada Gambar 3.8 dimana $(c_x; c_y)$ merupakan lokasi dari *grid*. $(b_x; b_y)$ merupakan lokasi *bounding box* yang didapatkan dari lokasi $(c_x; c_y)$ dengan hasil fungsi sigmoid dari t_x dan t_y , $(p_w; p_h)$ merupakan *anchor box* yang didapatkan dari proses *clustering*, sedangkan $(b_w; b_h)$ merupakan dimensi akhir dari *bounding box* yang didapatkan dari perkalian *anchor box* $(p_w; p_h)$ dengan $(t_w; t_h)$.



Gambar 3.8 Prediksi Lokasi *Bounding Box*

Hanya terdapat satu *predicted bounding box* yang digunakan untuk setiap *ground-truth object* yang ada. Pada pengembangan YOLO terbaru, yaitu YOLOv3, jika *input* gambar yang diberikan sebesar 416×416 , maka dihasilkan *output* dalam tiga skala dalam ukuran dimensi 13×13 , 26×26 , dan 52×52 piksel. Dari perbedaan skala tersebut, YOLOv3 dapat menghasilkan *predicted box* sebanyak 10,647 kotak. Hal ini dapat dilakukan karena YOLOv3 melakukan deteksi dengan cara mengaplikasikan *kernel* 1×1 pada *feature map* pada tiga ukuran yang berbeda di tiga tempat yang berbeda pada jaringan. Bentuk dari *kernel* deteksi tersebut adalah $1 \times 1 \times (B \times (5 + C))$, dimana B adalah jumlah bounding box yang dapat diprediksi oleh *feature map*, angka 5 mewakili 4 *bounding box* atau biasa disebut *batch*, dan 1 *object confidence*, dan C adalah jumlah *class*. Gambar 3.9 menjelaskan diagram alir metodologi *object training* yang dilakukan dengan ekstraksi fitur dan klasifikasi objek dari penggunaan Darknet53 pada YOLOv3.



Gambar 3.9 Diagram Alir Deteksi Objek

YOLO (*You Only Look Once*) adalah salah satu detector canggih yang mampu melokalisasi dan mengklasifikasikan beberapa objek dalam gambar. Tahapan *training* lebih kompleks daripada tahap *testing*. Karena harus dilakukan “pelatihan” untuk banyak tugas secara bersamaan, yaitu seperti *object localization*, *classification*, dan juga *objectness (confidence)*. Penambahan layer ini untuk

meningkatkan performa. YOLO juga menggunakan resolusi input 448x448, karena deteksi kadang membutuhkan informasi visual yang halus. Layer terakhir pada *network* memprediksi probabilitas kelas dan *koordinat bounding box*. Optimasi yang dilakukan menggunakan *sum-squared error* karena teknik ini mudah untuk dioptimasi. Untuk mengatasi *error* yang ditimbulkan karena banyaknya *grid cell* pada gambar yang tidak berisi objek, maka YOLO meningkatkan *loss* dari prediksi koordinat *bounding box* dan mengurangi *loss* dari keyakinan prediksi untuk *boxes* yang tidak berisi objek. Pada waktu *training*, diharapkan hanya satu prediktor *bounding box* untuk bertanggung jawab untuk setiap objek. Sistem menetapkan satu prediktor untuk “bertanggung jawab” memprediksi sebuah objek berdasarkan prediksi mana yang memiliki IOU tertinggi saat ini dengan *ground truth*. Setiap prediktor menjadi lebih baik dalam memprediksi ukuran tertentu, aspek rasio, atau kelas objek, meningkatkan daya ingat keseluruhan.

b. Feature Extraction dan Object Classification

Pada penelitian ini menggunakan YOLOv3 tiny model sebagai model *weight* sebelum dilakukan proses *training*. Perlu memodifikasi YOLOv3 tiny model (*yolov3-tiny.cfg*) sebelum dilakukan proses *training*. Hal ini dilakukan karena pada penelitian ini digunakan *custom class* untuk menunjang *custom detector*. Tahapan modifikasi yang dilakukan adalah:

1. Menghapus # (*uncomment*) pada baris ke 5, 6, dan 7. Kemudian ubah *training batch* menjadi 64 dan *subdivisions* menjadi 2.
2. Merubah angka *filters* untuk *convolutional layer* “[convolution]” untuk setiap *yolo output* “[yolo]” dengan rumus:

$$\text{number of filters} = \# \text{anchor} \times (5 + \# \text{numberofclasses})$$

Diubah pada baris ke 127 dan 171 “filters=...”

- Untuk setiap yolo layer [yolo] diubah jumlah *class* yang sesuai dengan *custom class* pada penelitian ini. Diubah pada baris ke 135 dan 177.

Pada *feature extractor* dari YOLOv3 penelitian ini menggunakan Darknet53, yang memiliki 53 *layer* yang telah dilatih di ImageNet. Pada Tabel 3.1 adalah susunan layer dari Darknet53.

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3/2	128 x 128
1x	Convolutional	32	1 x 1	128 x 128
	Convolutional	64	3 x 3	
	Residual			
	Convolutional	128	3 x 3/2	64 x 64
2x	Convolutional	64	1 x 1	64 x 64
	Convolutional	128	3 x 3	
	Residual			
	Convolutional	256	3 x 3/2	32 x 32
8x	Convolutional	128	1 x 1	32 x 32
	Convolutional	256	3 x 3	
	Residual			
	Convolutional	512	3 x 3/2	16 x 16
8x	Convolutional	256	1 x 1	16 x 16
	Convolutional	512	3 x 3	
	Residual			
	Convolutional	1024	3 x 3/2	8 x 8
4x	Convolutional	512	1 x 1	8 x 8
	Convolutional	1024	3 x 3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Tabel 3.1 Susunan *Layer* Darknet53

Kemudian dilakukan *training* dengan bantuan *Google Colaboratory* sebagai media pelatihan. Setiap neuron pada *Multi Layer Perceptron* saling berhubungan yang ditandai dengan tanda panah pada gambar diatas. Tiap koneksi memiliki *weight* yang

nantinya nilai dari tiap weight akan berbeda-beda. Hidden layer dan output layer memiliki tambahan “*input*” yang biasa disebut dengan bias. Pada tahap *training* setiap weight dan bias pada tiap neuron akan diupdate terus menerus hingga output yang dihasilkan sesuai dengan harapan. Pada tiap iterasi akan dilakukan proses evaluation yang biasanya digunakan untuk menentukan kapan harus menghentikan proses *training* (stopping point). Setelah dilakukan *training* sebanyak 3000 *epoch* maka didapatkan *final weight*. *Final weight* nanti akan digunakan sebagai data *testing* agar prediksi *bounding box* dapat secara akurat didapatkan pada tiap objek.

3.5.2 Testing

Testing dilakukan dengan mengaplikasikan hasil dari proses *training* yang telah dilakukan. Nilai dari bobot atau *weight* dan juga YOLOv3 dijalankan agar setiap *bounding box* yang telah di *training* dapat terdeteksi pada *test* gambar yang juga bertuliskan aksara Jawa. Dikarenakan jumlah class yang cukup banyak, maka data *test* yang diteliti hanya sebanyak 33 dan 47 *class*, 33 *class* ini adalah 33 jenis huruf konsonan aksara Jawa sedangkan 47 adalah jenis huruf konsonan berserta pasangannya, masing-masing sama-sama menganut sistem penulisan Aksara Jawa aturan penulisan Mardi Kawi.

3.6 Analisis Hasil Pendeteksian Aksara Jawa

Pada tahap ini dilakukan analisis dari hasil pendeteksian aksara Jawa oleh sistem. Sistem akan memberikan hasil pendeteksian berupa *confidence* dan *predicted bounding box*. Hasil dari pendeteksian tersebut kemudian dibandingkan dengan *ground-truth bounding box* yang telah diberikan sebelumnya pada proses *labelling*. Perbandingan ini dilakukan dengan cara menghitung nilai *Intersection over Union* (IoU) atau nilai *overlap* antara *ground-truth bounding box* dengan *predicted bounding box*. Dari hasil perhitungan IoU didapatkan nilai *True Positive*, *False Positive*, *True Negative*, dan *False Negative* yang digunakan untuk

mencari nilai *precision* dan *recall* dari hasil pendeteksian aksara Jawa.

3.6.1 Intersection over Union (IoU)

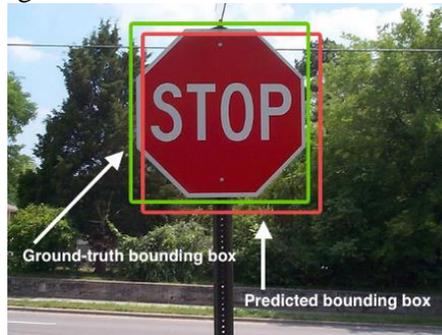
Intersection over Union (IoU) adalah matrik evaluasi yang digunakan untuk mengukur keakuratan pendeteksian objek pada *dataset*. IoU dapat digunakan untuk menilai sistem pendeteksi objek dengan ketentuan dasar yaitu:

1. Model *dataset* menghasilkan prediksi koordinat (x, y). Contohnya *bounding box* pada objek suatu gambar.
2. Memiliki *ground truth* dari *bounding box* dari *dataset* objek yang akan diteliti.
3. Memiliki *predicted* atau hasil yang diprediksi dari suatu proses.

Pada pengujian, sistem mengalikan probabilitas kelas dan *confidence* dari prediksi pada box, yang memberikan skor *confidence* (keyakinan) kelas khusus untuk setiap *box*. Skor ini mengkodekan probabilitas kelas yang muncul di dalam *box* dan seberapa baik *box* yang diprediksi cocok dengan objek. IoU merupakan perbandingan antara area potongan dengan area gabungan dari *predicted bounding box* dan *ground-truth bounding box* yang dapat dituliskan dengan Persamaan 3.2.

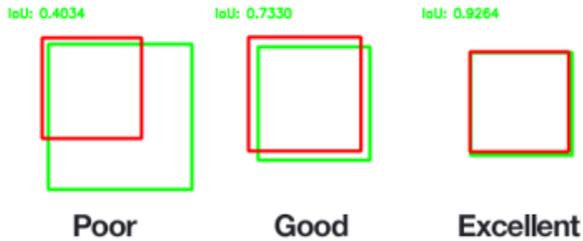
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (3.2)$$

Gambar 3.10 menjelaskan visualisasi dari proses yang terjadi pada sistem IoU yang dimiliki oleh CNN arsitektur YOLO.



Gambar 3.10 Visualisasi *Intersection over Union*

Sedangkan Gambar 3.11 adalah contoh gambar variasi hasil IoU pada *ground truth* yang ditunjukkan oleh gambar kotak berwarna hijau dan kotak berwarna merah adalah *predicted bounding box*.



Gambar 3.11 Gambar Contoh Variasi Hasil IoU

3.6.2 Confusion Matrix

Confusion Matrix adalah salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. *Confusion matrix* berisi informasi detail mengenai hasil klasifikasi oleh sistem terhadap data *testing* yang telah diketahui kelasnya. Berdasarkan jumlah keluaran kelasnya, sistem klasifikasi dapat dibagi menjadi 4 jenis yaitu *binary*, *multi-class*, *multi-label*, dan *hierarchical*[14]. Pada jenis klasifikasi *confusion matrix binary* terdapat 2 keluaran kelas yaitu yang disajikan seperti pada Tabel 3.2.

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
Negatif	FP (<i>False Positive</i>)	TN (<i>True Negative</i>)

Tabel 3.2 Tabel Penjelasan *Confusion Matrix*

Pada pengukuran kinerja menggunakan *confusion matrix*[15] terdapat 4 istilah sebagai representasi hasil proses klasifikasi. Keempat istilah menurut Tabel 3.2 adalah:

1. TP (*True Positive*) merupakan data positif yang terklasifikasi benar oleh sistem.

2. TN (*True Negatif*) merupakan jumlah data negatif yang terklasifikasi benar oleh sistem.
3. FP (*False Positive*) merupakan jumlah data positif namun terklasifikasi salah oleh sistem.
4. FN (*False Negative*) merupakan data negatif namun terklasifikasi salah oleh sistem.

Beberapa pengukuran untuk mengevaluasi kinerja sistem yang terdapat pada jenis pengukuran *confusion matrix* adalah:

1. *Precision*: Persentase prediksi positif yang benar dari seluruh hasil prediksi.
2. *Accuracy*: Persentase jumlah total prediksi benar.
3. *Recall*: Persentase data *testing* (actual) positif yang diprediksi benar (sebagai data positif).
4. *Specificity*: Persentase data *testing* (actual) negatif yang diprediksi benar (sebagai negatif).
5. *F-score* (*F-measure*): total nilai kesetaraan antara *precision* dan *accuracy*.

Untuk menghitung nilai presisi, akurasi, dan *recall* pada sistem dapat dilakukan dengan memperhatikan Persamaan 3.3, Persamaan 3.4, dan Persamaan 3.5.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (3.3)$$

$$Presisi = \frac{TP}{FP + TP} * 100\% \quad (3.4)$$

$$Recall = \frac{TP}{FN+TP} * 100\% \quad (3.5)$$

Sementara itu pada klasifikasi dengan jumlah keluaran kelas yang lebih dari dua (*multi-class*) cara untuk menghitung nilai akurasi, presisi, dan *recall* dapat dilakukan dengan menghitung rata-rata dari nilai akurasi, presisi, dan *recall* pada tiap kelas.

Perhitungan dapat didapatkan dengan menghitung menggunakan Persamaan 3.6, Persamaan 3.7, dan Persamaan 3.8.

$$Akurasi = \sum_{i=1}^1 \frac{(TP_i + TN_i)}{TP_i + TN_i + FP_i + FN_i} * 100\% \quad (3.6)$$

$$Presisi = \frac{\sum_{i=1}^1 TP_i}{\sum_{i=1}^1 (FP_i + TP_i)} * 100\% \quad (3.7)$$

$$Recall = \frac{\sum_{i=1}^1 TP_i}{\sum_{i=1}^1 (TP_i + FN_i)} * 100\% \quad (3.8)$$

$$f1 \text{ score} = \frac{Presisi * Recall}{Presisi + Recall} \quad (3.9)$$

BAB 4

PENGUJIAN DATA DAN ANALISA

Pada bab ini akan dijelaskan mengenai hasil pengujian serta analisa dari desain sistem dan implementasi. Yaitu hasil pengujian terhadap sistem pendeteksi aksara Jawa berbasis CNN dengan menggunakan arsitektur YOLO (*You Only Look Once*).

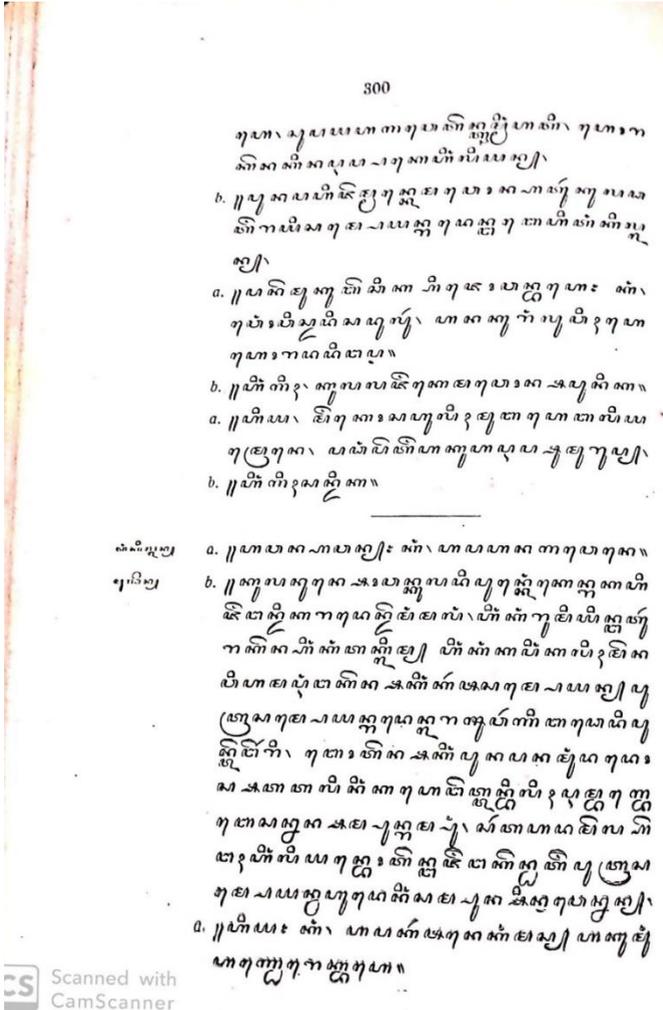
4.1. Hasil Metodologi Penelitian

Tahapan yang dilakukan pada penelitian ini dijelaskan sesuai dengan metodologi penelitian yang dijelaskan pada Gambar 3.2 sebelumnya,

4.1.1 Pengumpulan Data

Data berupa teks naskah bertuliskan Aksara Jawa yang didapatkan dengan mengunjungi Yayasan Sastra Lestari untuk mengambil data penunjang. Data uji dibedakan menjadi 2, yaitu data *training* dan data *testing*. Data *training* berupa 12 citra yang terdiri dari 5 penulis yang berbeda. naskah teks nya dalam keadaan baik dan citra sudah dilakukan perbaikan melalui proses *thresholding* dengan menggunakan bantuan *photo editor*. Sedangkan data *testing* disiapkan sejumlah 6 citra dari 2 penulis yang berbeda. Keberagaman penulis bertujuan agar *output* yang didapatkan adalah betul merupakan alih aksara dari aksara Jawa yang telah terdeteksi walaupun data yang digunakan sebagai input terdiri dari beberapa penulis yang berbeda. Keberagaman penulis juga mengakibatkan keberagaman besar kecil dan juga miring tegaknya penulisan aksara Jawa yang telah diteliti agar dapat dilihat tingkat akurasi dari penggunaan sistem penelitian. Dikarenakan data yang digunakan adalah naskah hasil *scan* huruf Jawa yang relative kecil jenis hurufnya, maka dilakukanantisipasi berupa pemotongan gambar menjadi 4 bagian. Gambar yang awalnya berukuran 600x800 kemudian dibagi menjadi 4 bagian dengan masing-masing gambar berukuran 416x416. Maka, jumlah total gambar data yang digunakan dalam penelitian ini

adalah: sejumlah 48 gambar *dataset* dan 24 gambar *datatest*. Pada Gambar 4.1 hingga Gambar 4.5 adalah merupakan citra *input dataset*.



Gambar 4.1 Tatacara, Padmasusastra, 1911, (Hlm. 300) sastra.org

Handwritten text in Jawi script, likely a land deed (Babad Tanah Jawi). The text is arranged in approximately 15 horizontal lines. It begins with a decorative wavy line at the top. The script is dense and characteristic of traditional Malay manuscripts. The text appears to be a formal document, possibly detailing land ownership or a transaction. The characters are well-defined and consistent throughout the page.

Gambar 4.2. Babad Tanah Jawi, Meinsma, 1874, (Hlm. 001)

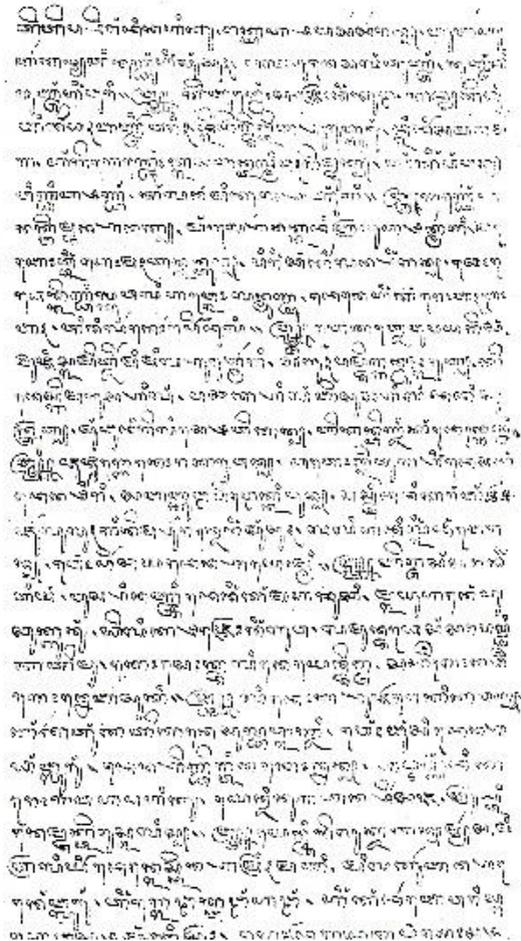
Handwritten text in a script, likely Indonesian or Malay, on aged paper. The text is arranged in several lines, with some lines starting with numbers 1 through 7. The handwriting is cursive and somewhat difficult to decipher due to the script and fading.

Handwritten text in a script, likely Indonesian or Malay, on aged paper. The text is arranged in several lines, with some lines starting with numbers 1 through 7. The handwriting is cursive and somewhat difficult to decipher due to the script and fading.

1.
2.
3.
4.
5.
6.
7.

Gambar 4.3 Nutilen Parepatan Pangreh, Radyapustaka, 1923

Sedangkan data Gambar 4.6 dan Gambar 4.7 adalah data gambar citra yang digunakan untuk *testing* hasil *training data*.



Gambar 4.6

Tabel 4.1 hingga Tabel 4.5 menjelaskan mengenai informasi dari masing-masing naskah yang digunakan sebagai *dataset* maupun *datatest*.

Tabel 4.1 Tabel Naskah *Serat Tatacara*

Judul	Serat Tatacara
Tahun Terbitan	1911
Penerbit	H. A Benyamin
Tempat	Semarang
Sub-judul	Ngadat sarta kalakuanipun tetiyang Jawi, ingkang taksih lumengket dhateng gugon-tuhon
Series dari	Serie uitgaven door bemiddeling der Commissie voor Inlandsche Volkslectuur
Bentuk	Prosa
Bahasa	Jawa
Aksara	Jawa
Penulis	Ki Padmasusastra
Kedudukan Penulis	Tiyang mardika ingkang amarsudi kasusastran Jawi ing Surakarta
Lokasi	Batawi
Tahun Ditulis	1883
Digitalisasi	Yayasan Sastra Lestari
Tanggal	5 Januari 1999
Ukuran Kertas	27.5 x 18.1 cm
Ukuran Isi Teks	20 x 10.7 cm

Tabel 4.2 Tabel Naskah *Babad Tanah Jawi*

Judul	Babad Tanah Jawi
Tahun Terbitan	1874
Penerbit	Martinus Nijhoff
Tempat	's Gravenhage
Sub-judul	Javaansche Geschiedenis Loopende tothet jaar 1647 der Javaansche Jaartelling
Series dari	Koninklijk Instituut voor de Taal, Landen Volkenkunde van Ned. Indie
Bentuk	Prosa
Bahasa	Jawa
Aksara	Jawa
Penulis	J. J. Meinsma
Kedudukan Penulis	-
Lokasi	-
Tanggal Ditulis	-
Digitalisasi	Yayasan Sastra Lestari
Tanggal	12 April 1999
Ukuran Kertas	A4
Ukuran Isi Teks	17.3 x 9.1 cm

Tabel 4.3 Tabel Naskah *Nutilen Parepatan Pangreh*

Judul	Nutilen Parepatan Pangreh
Tahun Terbitan	1923
Penerbit	Radya Pustaka
Tempat	-
Sub-judul	-
Series dari	-
Bentuk	Prosa
Bahasa	Jawa
Aksara	Jawa
Penulis	-
Kedudukan Penulis	-
Lokasi	-
Tanggal Ditulis	1883
Digitalisasi	Yayasan Sastra Lestari
Tanggal	7 Maret 2003
Ukuran Kertas	F4
Ukuran Isi Teks	28.5 x 14.7 cm

Tabel 4.4 Tabel Naskah Serat *Gancaran Warni-Warni Ing Jaman Punika*

Judul	Serat Gancaran Warni-warni Ing Jaman Punika
Tahun Terbitan	1933
Penerbit	J.B Wolters U. M.
Tempat	Betawi
Sub-judul	Kangge ing wulangan
Series dari	-
Bentuk	Prosa
Bahasa	Jawa
Aksara	Jawa
Penulis	R. L. Mallema
Kedudukan Penulis	Guru pamulangan tengahan bageyan basa wetanan
Lokasi	Ngayogyakarta
Tanggal Ditulis	1933
Digitalisasi	Yayasan Sastra Lestari
Tanggal	11 Desember 1997
Ukuran Kertas	A4
Ukuran Isi Teks	16.8 x 11 cm

Tabel 4.5 Tabel Naskah *Lelampahanipun R. Ng. Ranggawarsita*

Judul	Lelampahanipun R. Ng. Ranggawarsita Pujangga Karaton ing Surakarta
Tahun Terbitan	1931
Penerbit	Drikerei Mares
Tempat	Surakarta
Sub-judul	Serat Babad Cariyos Lelampahanipun Suwargi Raden Ngabei Ranggawarsita
Series dari	-
Bentuk	Prosa
Bahasa	Jawa
Aksara	Jawa
Penulis	Kumite Ranggawarsita
Kedudukan Penulis/Lokasi	-
Tanggal Ditulis	1931
Digitalisasi	Yayasan Sastra Lestari
Tanggal	21 Maret 1998
Ukuran Kertas	21.7 x 15.4 cm
Ukuran Isi Teks	17 x 10.9 cm

Terdapat 2 macam data gambar, yaitu data gambar untuk proses *training* dan data gambar untuk proses *testing*. Pada penelitian dilakukan dengan 2 jenis ukuran gambar yang berbeda, 600x800 dan juga 416x416. Untuk mempermudah proses sehingga distribusi data dapat dipantau, dilakukan perubahan nama file pada masing-masing jenis gambar. Tabel 4.6 dan Tabel 4.7 adalah daftar gambar dengan jenis ukuran 600x800, sedangkan Tabel 4.8 dan Tabel 4.9 adalah gambar dengan jenis ukuran 416x416.

Tabel 4.6 Daftar Gambar untuk *Training* 600x800

Judul Naskah	No. Kertas
Tatacara Padmasusastra(hlm.300)	Kertas 1
Babad Tanah Jawi	Kertas 2
Nutilen Parepatan	Kertas 3
Tatacara Padmasusastra(halaman prolog)	Kertas 4
Tatacara Padmasusastra(hlm.199)	Kertas 5
Tatacara Padmasusastra(hlm.102)	Kertas 6
Serat Gancaran Warni-warni(hlm.114)	Kertas 7
Serat Gancaran Warni-warni(hlm.116)	Kertas 8
Serat Gancaran Warni-warni(hlm.115)	Kertas 9
Serat Gancaran Warni-warni(hlm.117)	Kertas 10
Aksara Jawa	Kertas 11
Aksara Jawa	Kertas 12

Tabel 4.7 Daftar Gambar untuk *Testing* 600x800

Judul Naskah	No. Kertas
Aksara Jawa	<i>Test 1</i>
Aksara Jawa	<i>Test 2</i>
Aksara Jawa	<i>Test 3</i>
Aksara Jawa	<i>Test 4</i>
Aksara Jawa	<i>Test 5</i>
Aksara Jawa	<i>Test 6</i>

Tabel 4.8 Daftar Gambar untuk *Training* 416x416

Judul Naskah	No. Kertas
Tatacara Padmasusastra(hlm.300)	Kertas 1,2,3,4
Babad Tanah Jawi	Kertas 5,6,7,8
Nutilen Parepatan	Kertas 9,10,11,12
Tatacara Padmasusastra(halaman prolog)	Kertas 13,14,15,16
Tatacara Padmasusastra(hlm.199)	Kertas 17,18,19,20
Tatacara Padmasusastra(hlm.102)	Kertas 21,22,23,24
Serat Gancaran Warni-warni(hlm.114)	Kertas 25,26,27,28
Serat Gancaran Warni-warni(hlm.116)	Kertas 29,30,31,32
Serat Gancaran Warni-warni(hlm.115)	Kertas 33,34,35,36
Serat Gancaran Warni-warni(hlm.117)	Kertas 37,38,39,40
Aksara Jawa	Kertas 41,42,43,44
Aksara Jawa	Kertas 45,46,47,48

Tabel 4.9 Daftar Gambar untuk *Testing* 416x416

Judul Naskah	No. Kertas
Aksara Jawa	<i>Test</i> 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24
Ranggawarsita(hlm.1)	<i>Test</i> 25,26,27,28
Ranggawarsita(hlm.2)	<i>Test</i> 29,30,31,32
Ranggawarsita(hlm.4)	<i>Test</i> 33,34,35,36
Ranggawarsita(hlm.5)	<i>Test</i> 37,39,39,40
Ranggawarsita(hlm.prolog)	<i>Test</i> 41,42,43,44
Ranggawarsita(hlm.prolog)	<i>Test</i> 45,46,47,48
Ranggawarsita(hlm.prolog)	<i>Test</i> 49,50,51,52
Serat Gancaran Warni-warni(hlm.122)	<i>Test</i> 53,54,55,56
Nutilen Parepatan	<i>Test</i> 57,58,59,60
Tatacara Padmasusastra(hlm.301)	<i>Test</i> 61,62,63,64
Tatacara Padmasusastra(hlm.303)	<i>Test</i> 65,66

4.1.2 Anotasi Data atau *labelling image*

Anotasi merupakan proses membuat label dengan cara memberikan kotak batas (*bounding box*) beserta nama kelas pada objek disetiap citra. Pada penelitian ini digunakan data *custom class* yaitu sebanyak 33 dan 47 *class*. Sebelum melakukan anotasi data, dilakukan penyusunan *list class* terlebih dahulu. Tabel 4.10 dan Tabel 4.11 berisikan daftar *class* yang akan digunakan sebagai bahan penelitian.

Tabel 4.10 Tabel Naskah Mardi Kawi 33 *class* huruf Jawa konsonan

No.	Nama Aksara	Nama <i>class</i>
1.	ha	ha
2.	sa	sa
3.	ya	ya
4.	ga	ga
5.	na	na
6.	ma	ma
7.	pa	pa
8.	ta	ta
9.	da	da
10.	ba	ba
11.	ca	ca
12.	ka	ja
13.	wa	wa
14.	la	la
15.	ra	ra
16.	ṭha	ttha
17.	ṇa	nna
18.	ka	ka
19.	ṭa	tta
20.	tha	tha
21.	pha	pha
22.	cha	cha
23.	ḍa	dda
24.	gha	gha
25.	jha	jha
26.	kha	kha
27.	ḍha	ddha
28.	dha	dha
29.	bha	bha
30.	nga	nga
31.	nya	nya
32.	ṣa	sha
33.	ṣa	ssa

Tabel 4.11 47 class huruf Jawa konsonan dan huruf “ka” pasangan

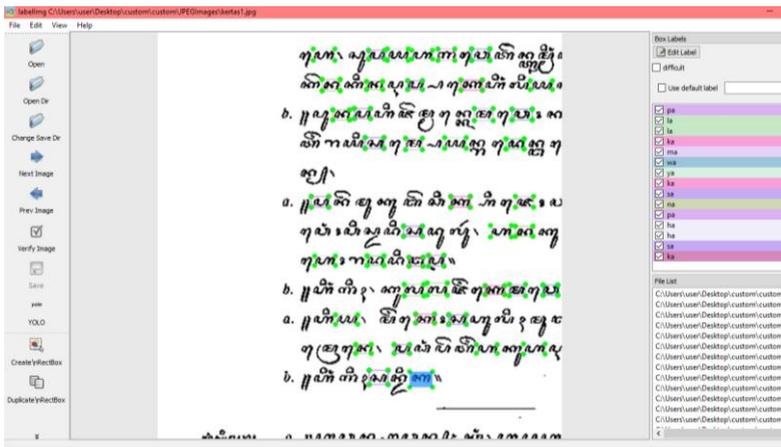
No.	Nama Aksara	Nama class
1.	ha	ha
2.	sa	sa
3.	ya	ya
4.	ga	ga
5.	na	na
6.	ma	ma
7.	pa	pa
8.	ta	ta
9.	da	da
10.	ba	ba
11.	ca	ca
12.	ka	ja
13.	wa	wa
14.	la	la
15.	ra	ra
16.	ṭha	ttha
17.	ṇa	nna
18.	ka	ka
19.	ṭa	tta
20.	tha	tha
21.	pha	pha
22.	cha	cha
23.	ḍa	dda
24.	gha	gha
25.	jha	jha
26.	kha	kha
27.	ḍha	ddha
28.	dha	dha
29.	bha	bha
30.	nga	nga
31.	nya	nya
32.	śa	sha
33.	ṣa	ssa
34.	ki	ki
35.	ku	ku
36.	kū	kuu
37.	ke	ke
38.	kō	koo
39.	kā	kaa
40.	ko	ko

41.	ké	kek
42.	kai	kai
43.	kau	kau
44.	kar	kar
45.	kaṅ	kaṅ
46.	kah	kah
47.	kya	kya

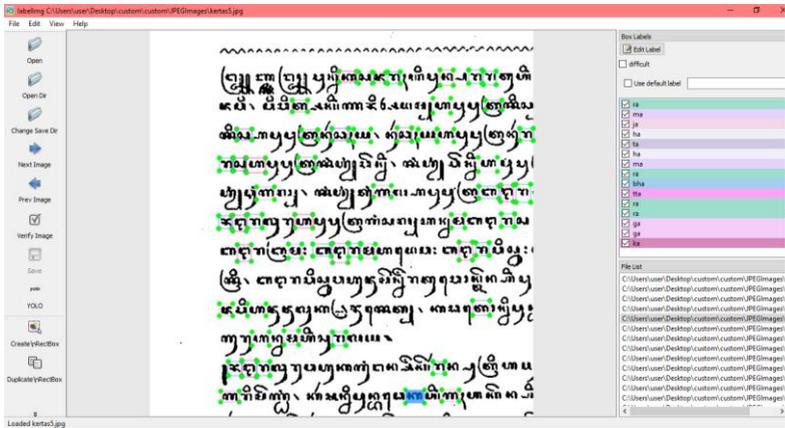
4.1.3 Hasil Anotasi Data

Dilakukan *labelling* pada setiap *dataset* yang akan di *training*. Pada penelitian ini dilakukan *labelling* kepada 12 kertas dengan isi dan penulis yang berbeda-beda. Hal ini dilakukan agar mendapatkan variasi *dataset* yang beragam. Setelah mempersiapkan data yang akan digunakan, dengan bantuan program *python* dilakukan anotasi atau *labelling* objek huruf jawa kepada setiap kertas.

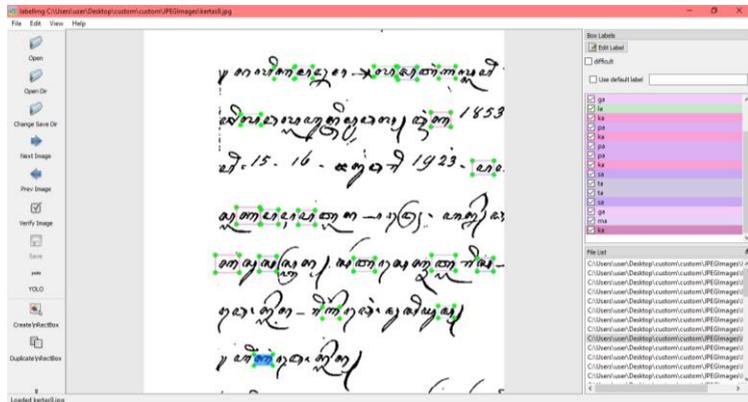
Pada Gambar 4.8 hingga Gambar 4.10 adalah proses *labelling* yang dilakukan pada beberapa kertas yang ditentukan sebagai *dataset*.



Gambar 4.8 Proses *labelling* pada kertas1.jpg



Gambar 4.9 Proses labelling pada kertas5.jpg



Gambar 4.10 Proses labelling pada kertas3.jpg

Setelah dilakukan anotasi data, *output* data hasil *labelling* adalah file .txt yang terdiri dari `<object-class> <x> <y> <width> <height>` hasil *labelling* ditunjukkan pada Gambar 4.11 dan Gambar 4.12.



```
Notepad
File Edit Format View Help
0 0.411667 0.054167 0.053333 0.028333
6 0.541667 0.054167 0.040000 0.021667
2 0.594167 0.054167 0.051667 0.028333
0 0.654167 0.054167 0.051667 0.021667
3 0.714167 0.054167 0.041667 0.028333
12 0.799167 0.054167 0.045000 0.028333
4 0.429167 0.129167 0.045000 0.028333
4 0.536667 0.130000 0.040000 0.020000
6 0.636667 0.129167 0.036667 0.021667
17 0.776667 0.129167 0.033333 0.029000
2 0.949167 0.129167 0.059000 0.021667
4 0.454167 0.202500 0.041667 0.025000
6 0.506667 0.199167 0.040000 0.025000
5 0.806667 0.201667 0.046667 0.030000
12 0.901667 0.201667 0.050000 0.026667
1 0.536667 0.271667 0.046667 0.026667
5 0.634167 0.274167 0.041667 0.028333
2 0.736667 0.271667 0.053333 0.026667
8 0.884167 0.274167 0.045000 0.025000
6 0.401667 0.416667 0.043333 0.026667
17 0.749167 0.416667 0.048333 0.026667
11 0.907500 0.416667 0.045000 0.026667
Ln 1, Col 1 100% Unix (LF) UTF-8
```

Gambar 4.11 Hasil anotasi data 33 *class*



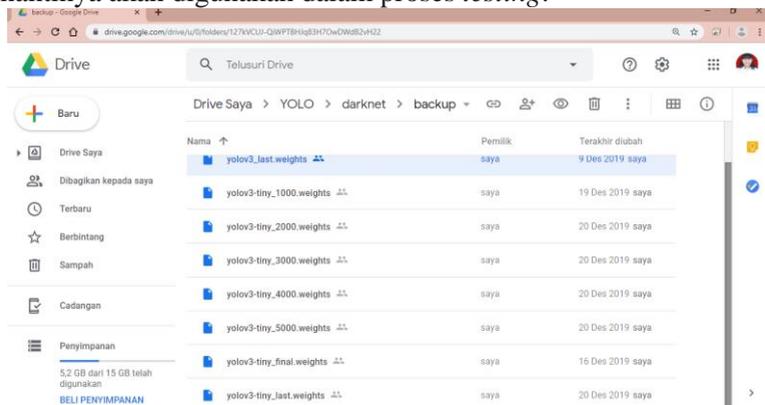
```
Notepad
File Edit Format View Help
0 0.414167 0.561667 0.051667 0.023333
8 0.539167 0.561667 0.041667 0.023333
29 0.639167 0.564167 0.039000 0.021667
6 0.689167 0.561667 0.038333 0.026667
13 0.634167 0.634167 0.048333 0.028333
13 0.691667 0.631667 0.050000 0.030000
17 0.834167 0.634167 0.048333 0.021667
5 0.890000 0.634167 0.040000 0.028333
12 0.971667 0.634167 0.036667 0.025000
2 0.467500 0.706667 0.055000 0.023333
17 0.661667 0.704167 0.046667 0.021667
1 0.741667 0.706667 0.043333 0.023333
4 0.509167 0.779167 0.038333 0.021667
6 0.604167 0.779167 0.038333 0.021667
0 0.814167 0.779167 0.051667 0.021667
0 0.931667 0.779167 0.053333 0.025000
17 0.651667 0.854167 0.056667 0.038333
1 0.546667 0.850833 0.050000 0.038333
33 0.485000 0.121667 0.053333 0.046667
34 0.568333 0.430000 0.056667 0.060000
34 0.966667 0.505833 0.056667 0.068333
34 0.574167 0.647500 0.055000 0.058333
Ln 1, Col 1 100% Unix (LF) UTF-8
```

Gambar 4.12 Hasil anotasi data 47 *class*

4.2 Training Data

4.2.1 Weight (bobot)

Convolution bekerja dengan prinsip *sliding window* dan *weight sharing* (mengurangi kompleksitas perhitungan). Pada Gambar 4.13 adalah output hasil *training* adalah bobot (*weight*) yang nantinya akan digunakan dalam proses *testing*.



Gambar 4.13 Hasil *Training* berupa bobot dari tiap *batch*.

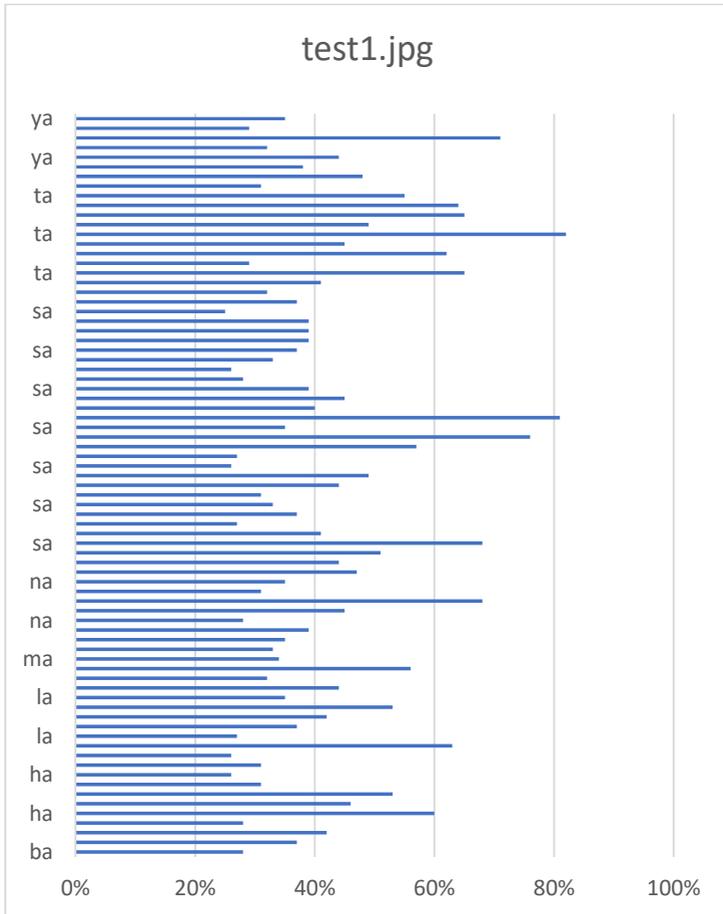
4.3 Analisis Data Hasil Pengujian

Dari hasil pengumpulan dan persiapan data akan dilakukan analisis seberapa baik performa sistem dalam mendeteksi aksara Jawa dengan menjalankan skenario-skenario yang telah ditentukan. Pengujian yang dilakukan dengan membagi tahapan pengujian penelitian menjadi beberapa bagian yaitu:

4.3.1 Pengujian Pendeteksian Aksara Jawa Konsonan

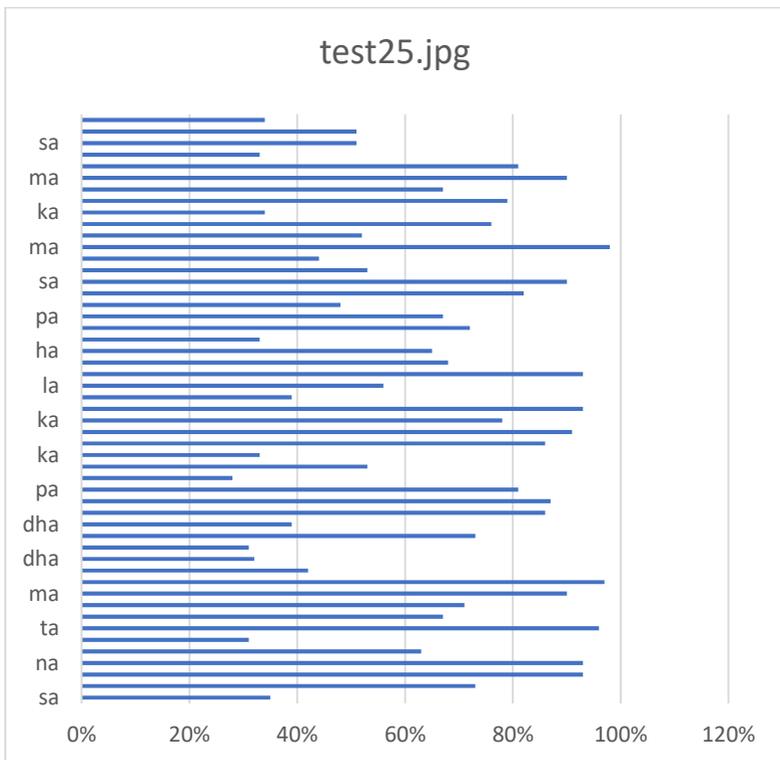
Pada pengujian pendeteksian aksara Jawa konsonan, dilakukan proses deteksi kepada 33 huruf Jawa Mardikawi. Dalam penelitian ini, jumlah *class* sama dengan jumlah huruf aksara Jawa. Pada pengujian pertama dilakukan deteksi huruf Jawa kepada 33 jenis *class* yang berbeda.

Gambar 4.14 adalah grafik dari percobaan hasil deteksi tiap huruf pada file gambar *test1.jpg*. Grafik pada Gambar 4.14 berisikan nilai dari *confidence* deteksi objek huruf, nilai yang ditampilkan berupa persentase (%).



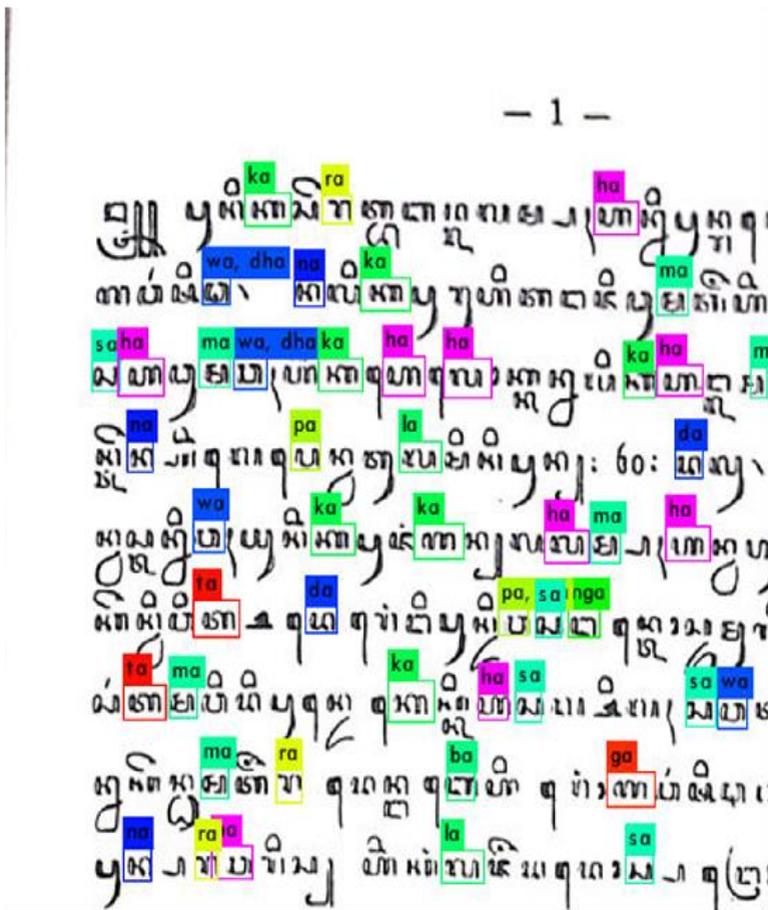
Gambar 4.14 Grafik nilai IoU hasil deteksi 33 *class*

Pada percobaan kedua digunakan file gambar yang telah diubah ukurannya menjadi 416x416. Percobaan kedua dilakukan kepada 24 jenis kertas yang berbeda isi tulisan aksara Jawa nya, pada Gambar 4.16 adalah grafik hasil deteksi dari gambar *test25.jpg*. Pada percobaan gambar *test25.jpg*. Menurut Gambar 4.16 nilai *confidence* terendah dimiliki oleh salah satu huruf “sa” sedangkan yang tertinggi dimiliki oleh huruf “ma”. Hal ini disebabkan oleh jumlah *gorund truth* yang terkumpul pada jenis *class* “sa” lebih sedikit dibandingkan dengan jenis “ma”, Total *ground truth class* “sa” adalah 197 sedangkan *class* “ma” berjumlah 223. Hal ini disebabkan oleh letak objek dan bounding box yang terdeteksi,



Gambar 4.16 Grafik nilai IoU hasil deteksi 33 *class*

Hasil pada Gambar 4.17 menunjukkan ketika dilakukan pembagian gambar dari yang awalnya berukuran 600x800 menjadi 416x416 gambar hasil deteksi terlihat lebih jelas dan letak objek lebih tepat.



Gambar 4.17 Gambar hasil test 33 class

Peramalan hasil deteksi dengan hasil alih aksara ahli sastra Jawa:

Punika sêrat babad lèlampahanipun Radèn Ngabèi Rānggawarsita, nalika puruhita ngaji dhumatêng ing kitha Panaraga, saha dhumawah ing kaelokan nalika anglampahi kungkum wontên ing lèpèn Watu laminipun 40 dalu, limrahipun dipun wastani wahyuning kapujanggan, lèlampahan wau kacariyosakên wiwit sadèrèngipun diwasa ngantos dumugi sedanipun, sarta mawi dipun dèkèki asalsilah sawatawis ingkang nurunakên dhumatêng Radèn Ngabèi Rānggawarsita wau, karanganipun para waris, ingkang lajêng dados pangrèh adêgan Rānggawarsitan, pangarangipun mênthêt pèpiritan saking pèngêtanipun para waris, ingkang anyumêrèpi piyambak dhatêng lèlampahan wau, kawontênanipun sami kapratelakakên kados ing ngandhap punika.

Dikarenakan pada Tugas Akhir ini terdapat percobaan lebih dari 2 class, maka dilakukan proses analisis *Confusion Matrix multi-class* yaitu klasifikasinya dilakukan dengan cara menghitung rata-rata dari nilai akurasi, presisi, dan *recall* pada tiap class.

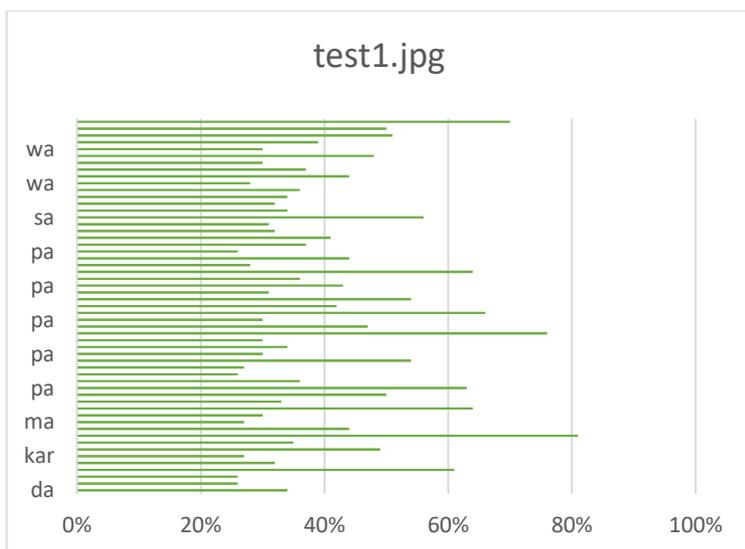
Setelah dilakukan perhitungan akurasi, didapatkan nilai akurasi sebesar 0.337 nilai presisi sebesar 0.327, nilai *recall* sebesar 0.411, dan *f1 score* sebesar 0.357. Terdapat kekosongan nilai pada aksara. Hal ini dikarenakan jumlah dataset yang berjenis aksara yang nilai akurasi, presisi, *recall*, dan *f1 score* adalah nol relatif kurang.

Tabel 4.12 Analisis *Confusion Matrix* 33 class

Class	Akurasi	Presisi	Recall	f1 score
ha	0.857143	0.85714	1	0.923077
na	0.75	0.75	1	0.857143
ca	0	0	0	0
ra	1	1	1	1
ka	1	1	1	1
da	1	1	1	1
ta	0.666667	0.66667	1	0.8
sa	1	1	1	1
wa	1	1	1	1
La	0.666667	0.5	1	0.666667
pa	1	1	1	1
dha	0	0	0	0
Ja	0	0	0	0
ya	0	0	0	0
nya	0	0	0	0
ma	0.857143	0.85714	1	0.923077
ga	0.666667	0.5	1	0.666667
pa	0.5	0.5	1	0.666667
tha	0	0	0	0
nga	0.5	0.5	1	0.666667
ttha	0	0	0	0
nna	0	0	0	0
tta	0	0	0	0
pha	0	0	0	0
cha	0	0	0	0
dda	0	0	0	0
gha	0	0	0	0
jha	0	0	0	0
kha	0	0	0	0
ddha	0	0	0	0
bha	0	0	0	0
sha	0	0	0	0
ssa	0	0	0	0
Rata-rata	0.337185	0.32738	0.411765	0.35794

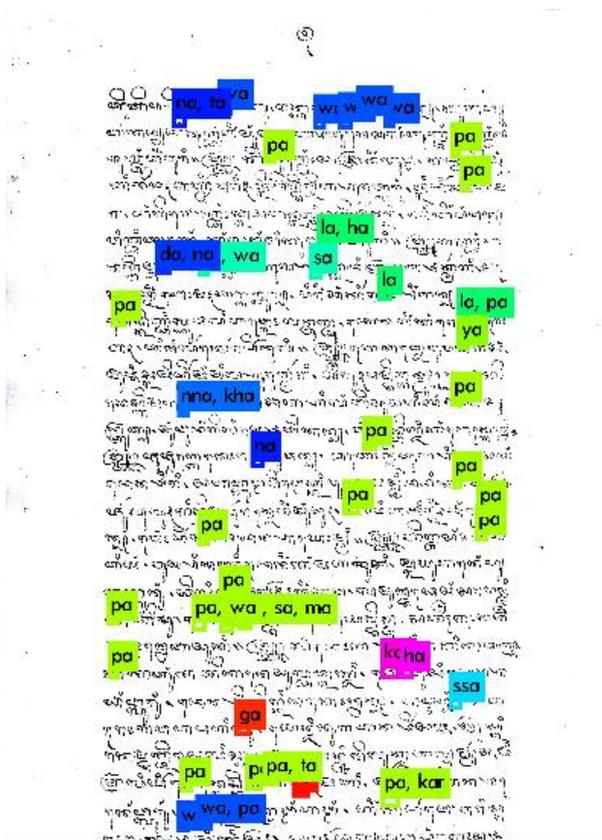
4.3.2 Pengujian Akasara Jawa Menggunakan Pasangan atau Sandhangan

Pada pengujian kedua yaitu pendeteksian aksara Jawa konsonan dengan menambahkan jenis huruf “ka” dengan pasangannya. Terdapat 14 jenis pasangan pada aturan aksara Jawa Mardikawi. Maka pada pengujian ini dilakukan proses deteksi kepada 33 huruf ditambahkan 14 jenis class huruf “ka+pasangan+ menjadi 47 jenis huruf. Dalam pengujian ini, jumlah *class* menjadi 47 jenis *class* yang berbeda. Seperti Gambar 4.14, Gambar 4.18 adalah grafik dari percobaan hasil deteksi tiap huruf pada file gambar *test1.jpg*. Namun bedanya adalah terdapat 47 *class* yang digunakan pada proses *labelling*. Grafik pada Gambar 4.18 berisikan nilai dari *confidence* deteksi objek huruf, nilai yang ditampilkan berupa persentase (%).



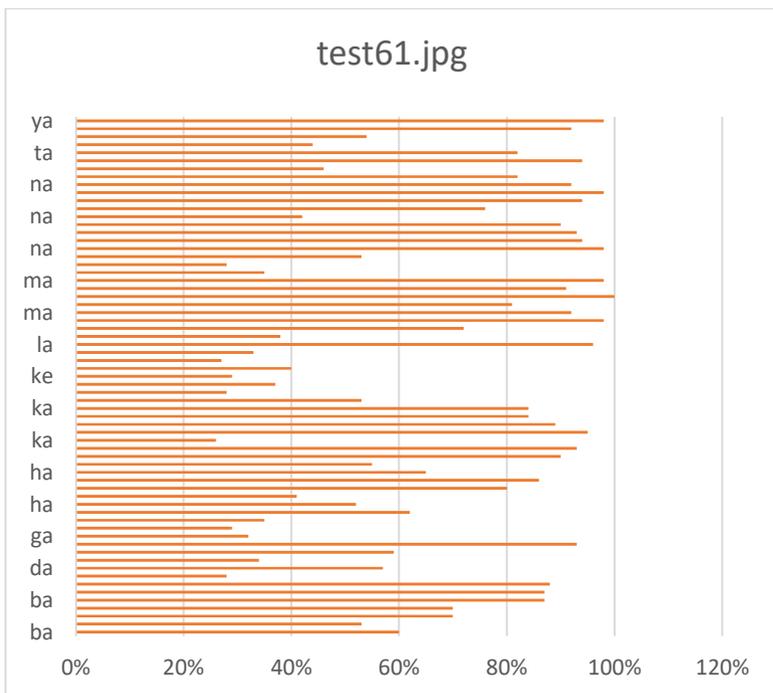
Gambar 4.18 Grafik nilai IoU hasil deteksi 47 *class*

Hasil *output* gambar dengan 47 *class* yang telah melalui proses *training* ditunjukkan pada Gambar 4.19. Pada percobaan ketiga ini digunakan file gambar asli yaitu gambar berukuran 600x800. Seperti pada hasil *output* gambar 33 *class*, hasil pada 47 *class* ini juga menunjukkan gambar hasil deteksi terlihat kurang jelas dikarenakan ukuran objek yang terlalu kecil.



Gambar 4.19 Gambar hasil *test* 47 *class*

Pada percobaan ketiga digunakan file gambar dengan yang telah diubah ukurannya menjadi 416x416. Gambar 4.20 adalah grafik dari percobaan hasil deteksi tiap huruf pada file gambar *test25.jpg*, terdapat kenaikan hasil *confidence* deteksi objek pada objek huruf “ma” sebesar 100%. Menurut Gambar 4.20 nilai *confidence* terendah dimiliki oleh salah satu huruf “ka” sedangkan yang tertinggi dimiliki oleh huruf “ma”. Hal ini disebabkan oleh jumlah *ground truth* yang terkumpul pada jenis *class* “ka” lebih sedikit dibandingkan dengan jenis “ma”, Total *ground truth class* “ka” adalah 220 sedangkan *class* “ma” berjumlah 223. Dan juga dapat terdeteksi jenis objek aksara konsonan dan aksara swara “ng” yaitu pada huruf “kang” dan juga huruf dengan aksara swara “u” yaitu huruf “ku”.



Gambar 4.20 Grafik nilai IoU hasil deteksi 47 class

Peramalan hasil deteksi dengan hasil alih aksara ahli sastra Jawa:

b: Mênggah sagêd panjênêngan nampèni têtêngêr wau benjing punapa.

a: O, la kuwi kudu dak pikir dhisik, wong bakal anampani dhayoh sarta ngakake sêsuguh ora gampang.

b: Kulu nun mênawi nampèni têngêran punika sumêrêp kula botên kenging mawi sêsêgah, sampun kalêbêt ing ila-ila botên sae, kêjawi namung sêgah wedang lajêng bibaran, makatên ugi têngêran ingkang warni panganan, punika botên kenging yèn botên dipun dumakên ing rencang tuwin tangga têpalih sarta kêdah dipun têlasakên, botên kenging nenggehakên sakêdhik-kêdhika.

Seperti pada Tabel 4.12, Tabel 4.13 juga berisi hasil proses analisis *Confusion Matrix multi-class* Jumlah *class* pada percobaan ini adalah sebanyak 47 *class*. Untuk percobaan ini didapatkan nilai akurasi, didapatkan nilai akurasi sebesar 13.2738 nilai presisi sebesar 10.29, nilai *recall* sebesar 17, dan *f1 score* sebesar 11.248.

Tabel 4.13 Analisis *Confusion Matrix* 47 *class*

Class	Akurasi	Presisi	Recall	f1 score
ha	0.5	0.5	1	0.666667
na	1	1	1	1
ca	0	0	0	0
ra	0	0	0	0
ka	0.666667	0.6	1	0.75
da	0.5	0.5	1	0.666667
ta	0.666667	0	1	0
sa	1	1	1	1
wa	1	1	1	1
la	1	1	1	1
pa	0.5	0.5	1	0.666667
dha	0	0	1	0
ja	0	0	1	0
ya	0.5	0	1	0

nya	0	0	0	0
ma	0.857143	0.857143	1	0.923077
ga	1	1	1	1
ba	0.833333	0.833333	1	0.909091
tha	0	0	0	0
nga	0.75	0	0	0
ttha	0	0	0	0
nna	0	0	0	0
tta	0	0	0	0
pha	0	0	0	0
cha	0	0	0	0
dda	0	0	0	0
gha	0	0	0	0
jha	0	0	0	0
kha	0	0	0	0
ddha	0	0	0	0
bha	0	0	0	0
sha	0	0	0	0
ssa	0	0	0	0
ki	0	0	0	0
ku	0.5	0.5	1	0.666667
kuu	0	0	0	0
ke	1	0	0	0
koo	0	0	0	0
kaa	0	0	0	0
ko	0	0	0	0
kek	0	0	0	0
kai	0	0	0	0
kau	0	0	0	0
kar	0	0	0	0
kang	1	1	1	1
kah	0	0	0	0
kya	0	0	0	0
Rata-rata	13.27381	10.29048	17	11.24883

4.3.3 Pengujian Pembuktian Gambar *Dataset Training* dan *Testing*

Pengujian pembuktian jika gambar yang digunakan pada *dataset training* dapat digunakan dalam proses *testing* namun dengan gambar yang berbeda penulis dan judul tulisan. Dilakukan pembuktian gambar *dataset* dapat digunakan sebagai data deteksi gambar *datatest* yang berbeda. Gambar 4.22, Gambar 4.23 Gambar 4.24, Gambar 4.25, dan Gambar 4.26 adalah gambar-gambar yang digunakan pada *dataset*.

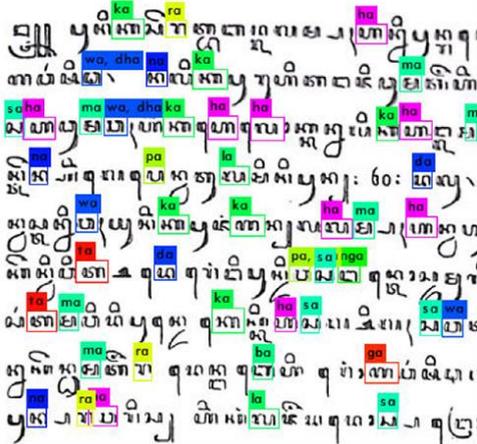
Handwritten text in Indonesian script, consisting of several lines and sub-points (a, b) that appear to be a list or set of instructions related to the dataset or testing process.

Gambar 4.22 Kertas1.jpg Tatacara Padmasusastra (halaman 300)

Pada tahap *testing* dilakukan kepada 5 jenis kertas yang berbeda. Jika ada jenis kertas yang sama dengan gambar pada tahap *training*, sebelumnya dipastikan terlebih dahulu halaman yang digunakan adalah halaman yang berbeda. Gambar 4.27, Gambar 4.28, Gambar 4.29, Gambar 4.30, adalah contoh dari hasil *test* untuk gambar yang memiliki penulis yang sama dengan gambar *dataset* namun halaman dari kertas yang berbeda. Sedangkan untuk gambar *test25*, gambar naskah penulis tidak dimiliki oleh gambar pada *dataset*.



Gambar 4.27 Test4.jpg Aksara Jawa



Gambar 4.30 Test25.jpg Ranggawarsita (halaman 1)

Terbukti jika ada perbedaan antara penulis pada gambar *dataset training* dengan gambar *dataset testing* dapat digunakan sebagai bagian dari proses pengumpulan data .

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

5.1 Kesimpulan

Pada penelitian ini, hasil pengenalan huruf Jawa dengan aturan penulisan Mardi Kawi dilakukan dengan menggunakan bantuan metode *Convolutional Neural Networks* (CNN). CNN menggunakan *image* sebagai data *input* untuk pemrosesan data. Ada berbagai macam jenis arsitektur CNN yang digunakan sebagai penunjang penelitian, salah satunya adalah YOLO (*You Only Look Once*). Seperti namanya, sistem YOLO diharapkan dapat membantu penelitian, dalam hal ini pemrosesan data citra hanya dengan sekali pengelihatan. Dengan sekali melihat, output proses yang dilakukan YOLO adalah prediksi jenis objek yang terdapat pada citra tersebut. Setelah dilakukan beberapa pengujian data, didapatkan kesimpulan:

1. Ukuran dan kualitas gambar yang digunakan pada data *training* dapat mempengaruhi keberhasilan dari hasil deteksi objek dengan menggunakan CNN arsitektur YOLO
2. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 33 jenis aksara Jawa Mardi Kawi menggunakan *dataset* dengan ukuran asli dari kertas mendapatkan hasil total rata-rata nilai IoU sebesar 46%.
3. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 47 jenis aksara Jawa Mardi Kawi menggunakan *dataset* dengan ukuran asli dari kertas mendapatkan hasil total rata-rata nilai IoU sebesar 40%.
4. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 33 jenis aksara Jawa Mardi Kawi menggunakan *dataset* kertas

- dengan membagi ukuran kertas menjadi 416x416 mendapatkan hasil total rata-rata nilai IoU sebesar 41%.
5. Pengujian pengenalan aksara Jawa Mardikawi dengan memperhatikan aksara Jawa konsonan, yaitu sebanyak 47 jenis aksara Jawa Mardi Kawi menggunakan *dataset* kertas dengan membagi ukuran kertas menjadi 416x416 mendapatkan hasil total rata-rata nilai IoU sebesar 72%.
 6. Telah terbukti jika semakin banyak variasi data ground truth aksara Jawa, semakin baik pula nilai dari *confidence* yang dihasilkan pada tahap *testing*.
 7. Menurut teori Intersection over Union pada CNN arsitektur YOLO, semakin banyak variasi data yang digunakan sebagai ground truth *dataset* yang kemudian akan di lakukan tahap *training*, maka semakin baik pula nilai *confidence* yang dihasilkan.
 8. Perbedaan gambar antara data *training* dan data *testing* dapat digunakan sebagai penunjang keanekaragaman data yang digunakan pada proses CNN deteksi tulisan tangan.

5.2 Saran

Demi pengembangan penelitian lebih lanjut mengenai tugas akhir ini, disarankan beberapa lanjutan yaitu:

1. Penambahan *dataset* aksara Mardi Kawi konsonan yang masih belum bisa terdeteksi seperti ca, ra, dha, ja, tha, nga, t̄ha, nna, ̄ta, pha, cha, ̄da, gha, jha, kha, ̄dha, bha, ̄sa, dan ̄sa.
2. Melakukan metode yang sama untuk pengenalan aksara Jawa aturan penulisan Mardi Kawi aksara konsonan beserta pasangan dan sandhangan.
3. Melakukan metode yang sama untuk pengenalan aksara Jawa aturan penulisan Mardi Kawi aksara konsonan beserta angka serta tanda baca.
4. Melakukan proses *augmentasi* data pada *dataset ground truth* yang akan digunakan pada tahap *training* guna menunjang performansi keberhasilan deteksi dan

meningkatkan variasi data yang akan digunakan dalam proses *training*.

5. Melakukan peramalan kata yang terbentuk pada hasil deteksi dan pengenalan aksara Jawa.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1]. Prasaja, Setya Amrih. Nityaksara Jawa Pengetahuan Aksara Jawa. 2014.
- [2]. Baried, dkk. Pengantar Teori Filologi. 1985
- [3]. Fatimah, Kiki. Tugas Akhir 2018. Britle Ancient Document Using Adaptive Local Thresholding. Departemen Teknik Komputer Institut Teknologi Sepuluh Nopember.
- [4]. Majelis Luhur Tamansiswa, 2011. Karya Ki Hadjar Dewantara, Bagian Kedua Kebudayaan. Yogyakarta: Yayasan Persatuan Tamansiswa.
- [5]. Heru Sri Kumoro. Kompas. 22 Juni 2019. John Paterson, Menyelamatkan Naskah Kuno
- [6]. Rismiyati, dkk., Deep Learning for Handwritten Javanese Character Recognition, 2017.
- [7]. Campbell, George L. Compendium of the World's Languages. Vol. 1. New York: Routledge, 2000.
- [8]. Gallop, Annabel T. Golden Letters: Writing Traditions of Indonesia. Jakarta: Lontar Foundation, 201
- [9]. Prasaja, Setya Amrih. Lingkar Jawa Paguyuban Pecinta Aksara Jawa. Bantul
- [10]. P. Mauridhi Hery, K. Agus. *Supervised Neural Networks* dan aplikasinya. (2006)

- [11]. Rismiyati, dkk., Deep Learning for Handwritten Javanese Character Recognition 2017.
- [12]. Samuel Sena, " Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN) ". Available: <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>
- [13]. Joseph Redmon, dkk., You Only Look Once: Unified, Real-Time Object Detection, 2015.
- [14]. M. Sokolova dan G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45.
- [15]. A. Solichin, Mengukur kinerja algoritma klasifikasi dengan *confusion matrix*.
<https://achmatim.net/2017/03/19/mengukur-kinerja-algoritma/-klasifikasi-dengan-confusion-matrix/,2017>
Diakses: 2020

LAMPIRAN

Daftar tabel pada bagian Lampiran adalah daftar tabel yang berisi hasil *labelling* 47 huruf konsonan pada 48 gambar.

	kertas1	kertas2	kertas3	kertas4	kertas5	kertas6	kertas7	kertas8	kertas9	kertas10	kertas11	kertas12	SUM
ha	4	2	1	6	6	4	2	2	1	1	0	0	29
na	7	8	4	12	1	2	1	0	0	0	0	0	35
nna	1	3	1	0	0	0	0	0	0	0	0	0	5
ca	0	2	0	0	1	0	0	0	0	0	0	0	3
cha	0	0	0	0	0	0	0	0	0	0	0	0	0
ra	6	8	8	0	12	13	9	3	0	0	0	0	59
ka	4	8	2	6	4	5	7	1	1	1	2	2	43
kha	0	0	0	0	0	0	0	0	0	0	0	0	0
da	4	3	4	4	0	0	0	0	0	0	0	0	15
dda	0	0	0	0	0	0	0	0	0	0	0	0	0
ta	2	6	2	4	5	4	4	3	4	4	0	2	40
ttha	0	0	0	0	0	0	0	0	0	0	0	0	0
sa	2	3	0	3	3	1	2	0	4	4	4	3	29
sha	0	0	0	0	0	0	0	0	0	0	0	0	0
ssa	0	0	0	1	0	0	0	0	0	0	4	2	7
wa	0	5	5	4	0	0	0	0	1	1	0	0	16
la	5	0	0	3	0	0	0	0	1	1	0	0	10
pa	2	3	2	2	0	0	0	0	1	1	2	0	13
pha	0	0	0	0	0	0	0	0	0	0	0	0	0
dha	1	2	0	1	0	0	0	0	1	1	0	0	6
ddha	0	0	0	0	0	0	0	0	0	0	0	0	0
ja	1	0	1	0	2	0	0	0	0	0	0	0	4
jha	0	0	0	0	0	0	0	0	0	0	0	0	0
ya	0	2	1	3	2	2	0	1	0	0	2	2	15
nya	0	0	0	0	0	0	0	0	0	0	0	0	0
na	6	6	6	6	3	3	1	1	1	1	1	1	36
ga	7	2	5	1	4	3	4	5	1	1	0	0	33
gha	0	0	0	0	0	0	0	0	0	0	0	0	0
ba	1	1	1	2	2	1	0	0	0	0	1	1	10
bha	0	0	0	0	2	0	0	0	0	0	0	0	2
ttha	0	0	0	0	5	5	1	2	0	0	0	0	13
ttha	0	0	0	0	0	0	0	0	0	0	0	0	0
nga	1	2	0	2	0	0	0	0	0	0	0	0	5
SUM	54	66	43	40	52	43	31	18	16	16	16	13	408

	kertas13	kertas14	kertas15	kertas16	kertas17	kertas18	kertas19	kertas20	kertas21	kertas22	kertas23	kertas24	SUM
ha	5	7	8	4	4	2	1	2	4	7	2	1	47
na	7	5	6	2	7	3	8	8	3	5	2	1	57
na	0	0	0	0	0	0	0	0	0	0	0	0	0
ca	0	1	1	1	0	0	0	1	0	0	0	0	4
cha	0	0	0	0	0	0	0	0	0	0	0	0	0
ra	0	0	0	0	0	0	0	0	0	0	0	0	0
ka	4	5	5	3	3	4	6	5	2	5	4	8	54
kha	0	0	0	0	0	0	0	0	0	0	0	0	0
da	0	0	0	0	0	0	2	3	1	1	0	0	7
dda	0	0	0	0	0	0	0	0	0	0	0	0	0
ta	2	3	2	2	4	4	5	2	1	2	3	2	32
ttha	0	0	0	0	0	0	0	0	0	0	0	0	0
sa	4	4	11	5	2	0	4	6	1	2	3	3	45
sha	0	0	0	0	0	0	0	0	0	0	0	0	0
ssa	0	0	0	0	0	0	0	0	0	0	0	0	0
wa	2	1	4	5	5	2	3	1	3	3	3	2	34
la	1	1	4	2	4	3	3	5	1	1	2	2	29
pa	1	2	1	2	4	2	4	4	2	3	1	2	28
pha	0	0	0	0	0	0	0	0	0	0	0	0	0
dha	1	4	0	1	1	0	1	1	0	0	0	0	9
ddha	0	0	0	0	0	0	0	0	0	0	0	0	0
ja	2	1	3	0	0	0	1	1	0	0	1	2	11
jha	0	0	0	0	0	0	0	0	0	0	0	0	0
ya	3	2	1	1	3	2	1	2	1	2	0	0	18
nya	0	0	0	0	0	0	0	0	0	0	0	0	0
na	6	3	2	3	1	1	8	5	5	3	1	1	39
ga	1	1	3	2	3	0	3	0	3	1	1	1	19
gha	0	0	0	0	0	0	0	0	0	0	0	0	0
ba	2	3	3	3	6	3	6	4	1	1	1	1	34
bha	0	0	0	0	2	0	0	0	0	0	0	0	2
tta	0	0	1	1	0	0	0	0	0	0	0	0	2
tttha	1	0	0	0	0	0	0	0	0	0	0	0	1
nga	3	4	1	0	2	2	6	8	0	2	1	0	29
SUM	45	47	56	37	51	28	62	58	28	38	25	26	501

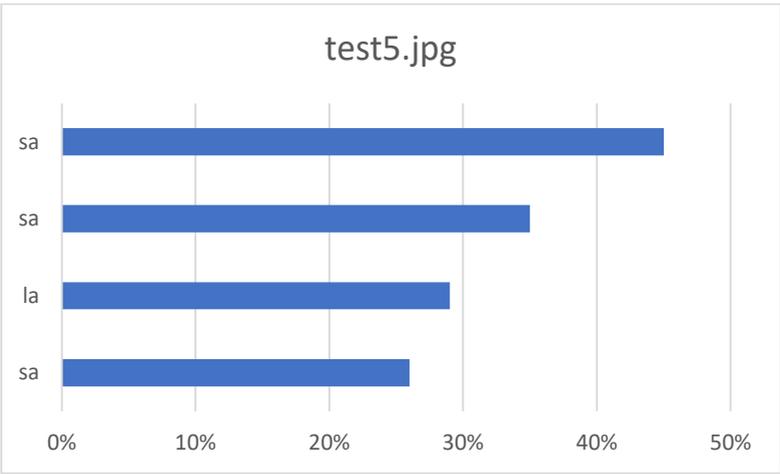
	kertas25	kertas26	kertas27	kertas28	kertas29	kertas30	kertas31	kertas32	kertas33	kertas34	kertas35	kertas36	SUM
ha	2	3	3	3	2	5	2	4	2	1	6	4	37
na	12	3	6	7	16	9	10	6	9	14	5	7	104
nna	0	0	0	0	1	2	0	0	1	0	0	1	5
ca	0	0	0	0	0	0	1	0	0	0	0	0	1
cha	0	0	0	0	0	0	0	0	0	0	0	0	0
ra	1	2	6	5	7	6	4	6	6	7	5	7	62
ka	3	7	6	9	7	6	7	6	5	3	2	4	65
kha	0	0	0	0	0	0	0	0	0	0	0	0	0
da	4	1	2	2	7	8	3	8	5	5	4	4	53
dda	0	0	0	0	0	0	0	0	0	0	0	0	0
ta	4	1	3	5	2	4	5	2	4	3	6	2	41
tha	0	0	0	0	0	0	0	1	0	0	0	0	1
sa	8	8	12	9	7	4	4	7	9	4	8	2	82
sha	0	0	2	0	0	0	0	0	1	4	0	0	7
ssa	0	0	0	0	0	0	0	0	0	0	0	0	0
wa	3	5	3	3	2	1	0	2	2	3	1	0	25
la	5	5	7	7	2	3	8	3	3	2	5	5	55
pa	2	3	1	1	1	4	3	2	0	1	2	2	22
pha	0	0	0	1	0	0	1	0	0	0	0	0	2
dha	2	1	1	1	2	1	2	1	0	0	3	1	15
ddha	0	0	0	0	0	0	0	0	0	0	0	0	0
ja	1	0	3	1	0	0	0	0	0	0	1	2	8
jha	0	0	0	0	0	0	0	0	0	0	0	0	0
ya	1	3	3	3	1	1	1	0	1	1	0	0	15
nya	0	0	0	0	0	0	0	0	0	0	0	0	0
ma	6	5	9	9	8	6	8	8	7	6	7	7	86
ga	1	0	2	3	5	6	6	2	1	2	5	7	40
gha	0	0	0	0	0	0	0	0	0	0	0	0	0
ba	5	3	3	4	1	0	3	0	5	4	1	1	30
bha	0	0	0	0	0	0	0	0	0	0	0	0	0
tta	0	0	0	0	0	0	0	0	0	0	0	0	0
tttha	0	0	0	0	0	0	0	0	0	0	0	0	0
nga	0	0	0	2	2	3	0	0	3	1	1	1	13
SUM	60	50	72	75	73	69	68	58	64	61	62	57	769

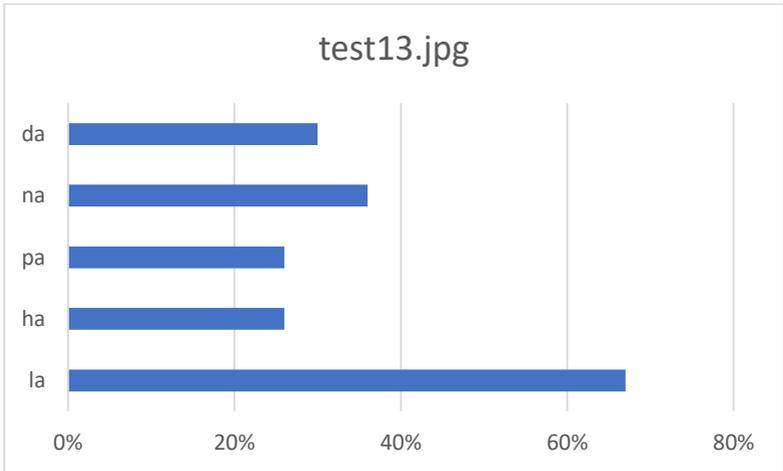
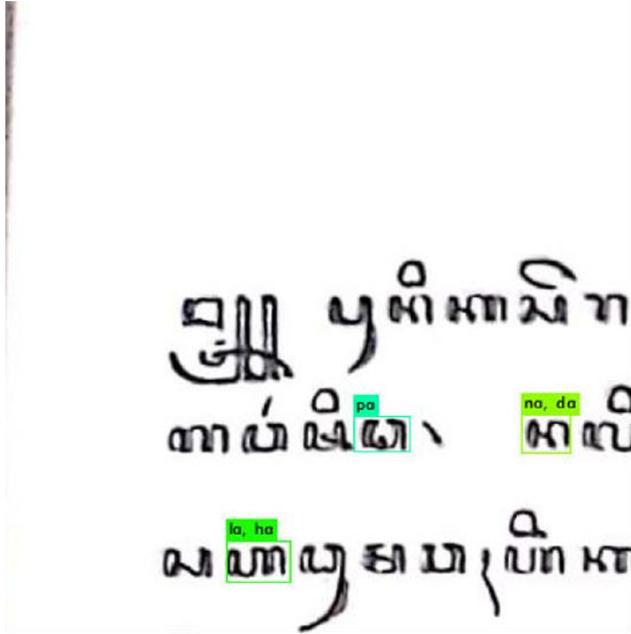
	kertas37	kertas38	kertas39	kertas40	kertas41	kertas42	kertas43	kertas44	kertas45	kertas46	kertas47	kertas48	SUM
ha	2	1	5	4	6	4	4	5	2	3	6	6	48
na	7	4	5	4	1	1	2	3	1	0	0	2	30
na	3	2	0	0	0	1	0	0	0	0	0	0	6
ca	2	0	0	0	0	0	1	1	0	1	1	1	7
cha	0	0	0	0	0	0	0	0	0	0	0	0	0
ra	8	8	1	2	0	0	0	0	0	0	0	0	19
ka	8	3	5	7	4	5	5	5	3	2	6	5	58
kha	0	0	0	0	0	0	0	0	0	0	0	0	0
da	2	4	4	6	1	0	1	1	1	1	0	0	21
dda	0	0	0	0	0	0	0	0	0	0	0	0	0
ta	5	2	0	1	4	4	6	6	4	2	2	4	40
ttha	0	0	0	0	0	0	0	0	0	0	0	0	0
sa	3	2	2	2	3	2	4	4	7	6	8	7	50
sha	0	0	0	1	0	0	0	0	0	0	0	0	1
ssa	0	0	0	0	0	0	0	0	2	3	0	0	5
wra	5	5	1	1	1	0	0	1	1	0	5	2	22
la	0	0	1	1	3	6	7	8	5	3	6	4	44
pa	3	4	0	4	0	2	5	5	1	4	2	3	33
pha	0	0	0	0	0	1	0	0	0	0	2	2	5
dha	2	0	1	1	1	1	0	0	0	0	0	0	6
ddha	0	0	0	0	0	0	0	0	0	0	0	0	0
ja	0	1	1	2	0	0	1	1	0	0	1	1	8
jha	0	0	0	0	0	0	0	0	0	0	0	0	0
ya	2	1	2	2	3	7	1	1	3	4	2	2	30
nya	0	0	0	0	0	0	0	0	0	0	0	0	0
ma	7	6	5	4	3	1	5	6	3	6	3	4	53
ga	2	5	5	1	0	0	2	1	2	1	0	0	19
gha	0	0	0	0	0	0	0	0	0	0	0	0	0
ba	1	0	2	1	1	1	1	1	1	1	3	5	18
bha	0	0	0	0	0	0	0	0	0	0	0	0	0
tta	0	0	0	0	0	0	0	0	0	0	1	0	1
ttha	0	0	0	0	0	0	0	0	0	0	0	0	0
nga	2	0	0	0	0	0	0	0	0	0	1	5	8
SUM	64	48	40	44	31	36	45	49	36	37	49	53	532

Gambar Test	Nilai Confidence
1	40%
2	37%
3	38%
4	34%
5	34%
6	34%
7	39%
8	42%
9	32%
10	35%
11	35%
12	37%
13	37%
14	47%
15	32%
16	37%
17	37%
18	40%
19	31%
20	40%
21	45%
22	35%
23	40%
24	41%
25	47%
26	43%
27	45%
28	43%
29	46%
30	42%
31	47%
32	50%
33	37%
34	47%
35	45%
36	45%
37	41%
38	37%
39	46%
40	51%
41	45%
42	42%
43	44%
44	44%
45	48%
46	49%
47	47%
48	45%
Rata-rata Total Confidence	41%

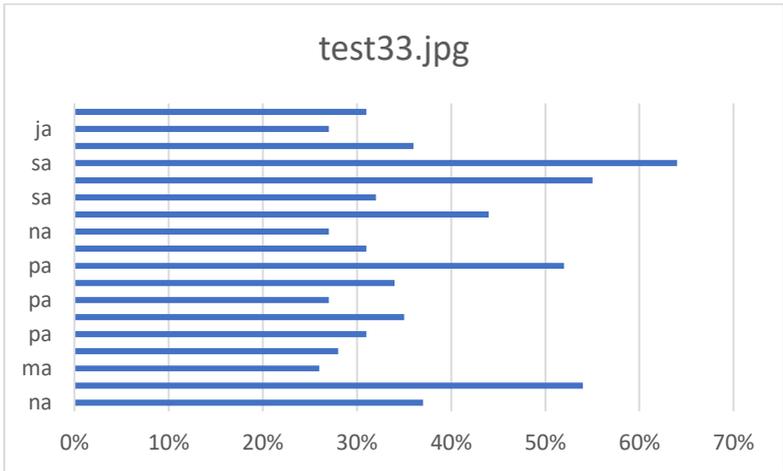
Tabel Hasil testing dengan jenis 33 class

မြေကြီးဝေး... ကြေးတူ... မြေကြီး...
 မြေကြီး... ကြေးတူ... မြေကြီး...

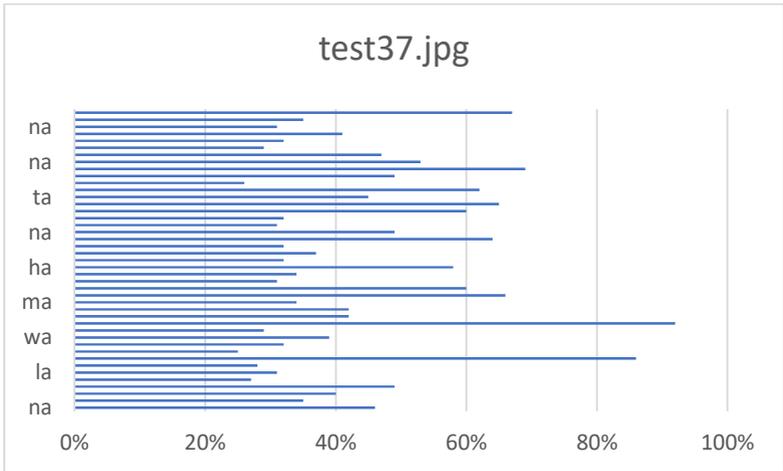




Handwritten text with highlighted syllables in various colors (green, red, pink, yellow). The text is written in a cursive style and appears to be a mix of characters and symbols. The highlighted syllables include: pa, na, sa, ba, ma, ra, ma, la, and ja.



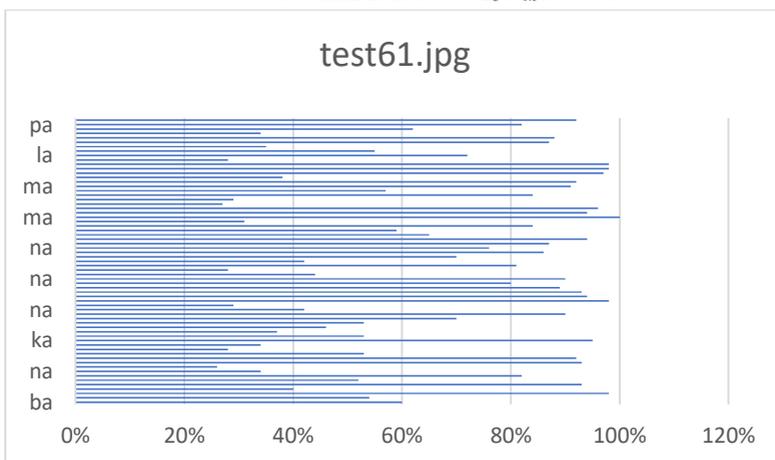
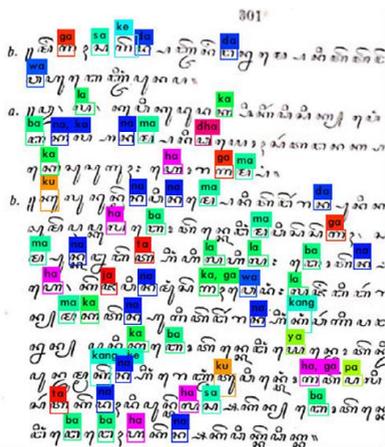
Handwritten Burmese text with colored labels (ra, pa, na, ha, sa, la, wa, na, sa, na, wa, ha, sa, na, ga, na, sa, ka, wa, ha, ma, ga) overlaid on the characters.



Tabel Hasil testing dengan jenis 47 class

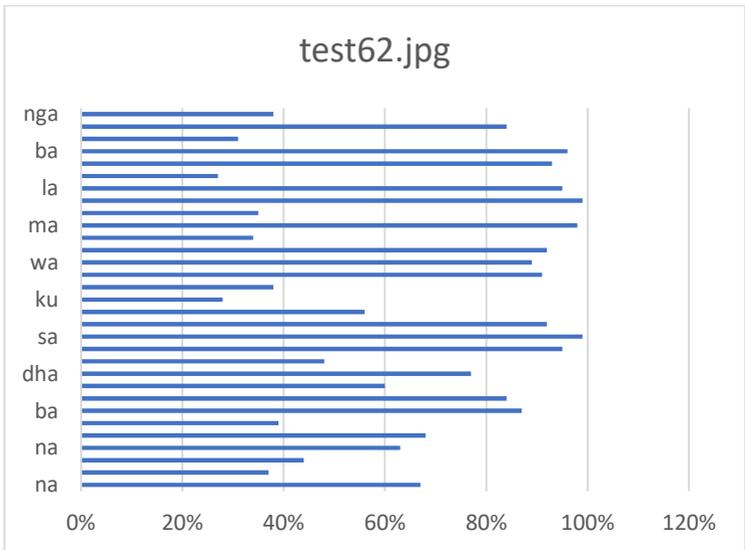
test61.jpg	test62.jpg	test63.jpg	test64.jpg	test65.jpg	test66.jpg	RATA-RATA TOTAL confidence
67%	67%	74%	74%	75%	74%	72%

Hasil testing dengan jenis 33 class

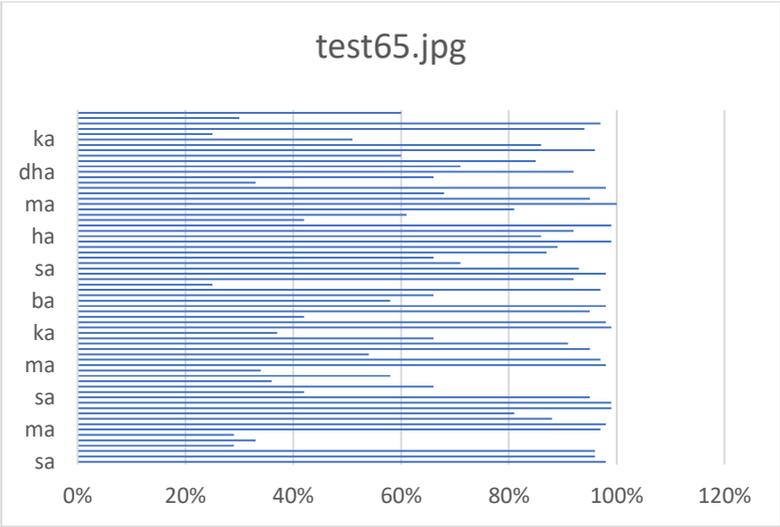


သံသယရှိသော အချက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၁။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၂။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၃။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၄။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၅။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၆။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၇။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၈။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၉။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။
 ၁၀။ အချက်အလက်များကို အောက်ဖော်ပြပါအတိုင်း ဖော်ပြပါ။

test62.jpg



ma ya no ka kanc sa ka pa
 la ma ma dha
 na la ba na ke ma
 sa ma ba ba
 a. || ma ya no ka kanc sa ka pa
 sa ma no sa ba
 la sa sa la
 ka na ma ba ma ha
 sa ma ga ya ha da
 sa ma ya ha da
 ka ka ha
 b. || sa ma ma sa ma sa
 na sa ma sa ma sa
 na sa ma sa ma sa



Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Lestari Ekky Damayanti, lahir di Sidoarjo pada tanggal 29 Mei 1997. Merupakan anak pertama dari dua bersaudara. Telah menyelesaikan pendidikan sekolah dasar di SD Kristen Petra 12 Sidoarjo (2003-2009), SMP Hang Tuah 5 Candi Sidoarjo (2009-2012), SMA Cendekia Sidoarjo (2012-2015). Penulis memiliki bidang minat dalam bidang teknologi dan seni. Selama menjadi mahasiswa Institut Teknologi Sepuluh Nopember, penulis juga aktif dalam berpartisipasi dalam beberapa kegiatan *event* seperti *Electra Event*, *Multimedia & Game Event*, dan Natal Paskah Persekutuan Mahasiswa Kristen ITS. Serta beberapa organisasi kampus seperti Persekutuan Mahasiswa Kristen (PMK) ITS dan HIMATEKTRO ITS. Penulis dapat dihubungi melalui email : lestariiekky@gmail.com.