**TUGAS AKHIR**



Oleh:

JEVON ANANTA GUNAWAN NRP: 160420097

**DATA SCIENCE & ARTIFICIAL INTELLIGENCE PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS**

**TEKNIK**

**UNIVERSITAS SURABAYA**

**2024**

**TUGAS AKHIR**

Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana



Oleh:

**JEVON ANANTA GUNAWAN NRP: 160420097**

**DATA SCIENCE & ARTIFICIAL INTELLIGENCE PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS**

**TEKNIK**

**UNIVERSITAS SURABAYA**

**2024**

**HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh :

Nama : Jevon Ananta Gunawan

NRP : 160420097

Program Studi : Teknik Informatika

Judul Skripsi : Pembuatan Aplikasi Sistem Pengenalan Aksara Bali Menggunakan Metode Convolutional Neural Network

**Telah diperiksa Dosen Pembimbing dan berhasil dipertahankan dihadapan Tim Penguji untuk diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Surabaya**

Dosen Pembimbing I Dosen Pembimbing II Tanda tangan Tanda tangan

(……….Nama………) (……….Nama………)

Dosen Penguji I Dosen Penguji II Tanda tangan Tanda tangan

(……….Nama………) (……….Nama………)

Ditetapkan di : …………….. Tanggal : …………….

Mengetahui

Ketua Program Studi

Tanda tangan

(……….Nama………)

iii

**PERNYATAAN KEASLIAN**

Nama : Jevon Ananta Gunawan

NRP : 160420097

Program Studi : Peminatan Data Science and Artificial Intelligence

Program Studi Teknik Informatika

Fakultas : Teknik

Judul Skripsi : Pembuatan Aplikasi Sistem Pengenalan Aksara Bali Menggunakan Metode Convolutional Neural Network

Skripsi ini adalah hasil karya saya sendiri, semua sumber kutipan dan rujukan telah saya tulis dengan benar dan disebutkan dalam sumber kutipan dan daftar pustaka. Apabila dikemudian hari penulisan Skripsi/Tesis ini merupakan hasil plagiat atau jiplakan terhadap karya orang lain, maka saya bersedia bertanggung jawab atas nama diri sendiri dan menerima sanksi berdasarkan ketentuan yang berlaku di Universitas Surabaya

Tanggal : 26 Januari 2024

Jevon Ananta Gunawan

iii

JUDUL : PEMBUATAN APLIKASI SISTEM PENGENALAN AKSARA BALI MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK

Nama : Jevon Ananta Gunawan

Jurusan/ Program Studi : Teknik Informatika

Pembimbing 1 : Mohammad Farid Naufal, S.Kom., M.Kom. Pembimbing 2 : Dr. Joko Siswantoro, S.Si., M.Si.

**ABSTRAK**

Aksara Bali adalah suatu bentuk tulisan tradisional yang berasal dari Bali dan digunakan sebagai media komunikasi tertulis. Akan tetapi, seiring perkembangan waktu, tingkat literasi masyarakat Bali dalam mengenali tulisan aksara Bali semakin rendah. Sebagian besar orang membutuhkan waktu yang sangat lama serta melalui proses yang panjang dan berulang-ulang dalam melakukan proses pengenalan dari tulisan aksara Bali ke tulisan latin. Masalah ini dapat diselesaikan menggunakan algoritma *Convolutional Neural Network* (CNN) dengan memanfaatkan berbagai macam arsitektur untuk melakukan pengenalan karakter pada tulisan aksara Bali dengan batasan uji hanya dapat mengenali aksara *wreastra*, *pengangge suara* dan *tengenan*. Sistem akan melakukan proses segmentasi untuk mendapatkan masing-masing karakter menggunakan salah satu teknik *Pre-processing* citra, yaitu *projection profile segmentation*. Tiga model klasifikasi tulisan aksara Bali yang masing-masing dibuat menggunakan arsitektur MobileNet V2, ResNet50, dan Xception memperoleh tingkat akurasi sebesar 99.8%. Sistem transliterasi aksara Bali yang dibuat kemudian diuji menggunakan 100 citra uji coba untuk menghitung *Levenshtein Distance* dari hasil transliterasi. Nilai *Levenshtein Distance* yang didapat adalah 85.9 untuk model MobileNet V2, 86.34 untuk model ResNet50, dan 83.78 untuk model Xception. Sistem divalidasi oleh 15 responden dan disimpulkan bahwa aplikasi dapat membantu pengguna dalam melakukan pengenalan dari tulisan aksara Bali ke tulisan latin. Hasil dari uji hipotesis t-*test* yang membandingkan skor hasil transliterasi yang dilakukan masyarakat Bali sebelum dan setelah menggunakan aplikasi transliterasi menunjukkan bahwa terdapat perbedaan berupa meningkatnya tingkat literasi masyarakat secara signifikan setelah menggunakan aplikasi sehingga dapat disimpulkan bahwa aplikasi yang dibuat dapat membantu meningkatkan literasi pengguna dalam mengenali tulisan aksara Bali.

**Kata Kunci:** **Aksara Bali, *Convolutional Neural Network*, Pengenalan, *Pre-processing* Citra**

v

TITLE: CREATING BALINESE SCRIPT RECOGNITION SYSTEM APPLICATION USING CONVOLUTIONAL NEURAL NETWORK METHOD

Name: Jevon Ananta Gunawan

Dicipline/Study Programme: Informatics Engineering Contributor 1: Mohammad Farid Naufal, S.Kom., M.Kom. Contributor 2: Dr. Joko Siswantoro, S.Si., M.Si.

**ABSTRACT**

Balinese script is a form of traditional script that originated from Bali and used as a medium of written communication. But, over time, the Balinese people’s literacy level in recognizing Balinese script is getting lower. Most people need a lot of time and go through a long and repetitive process in recognizing from Balinese script to latin script. This problem can be solved using Convolutional Neural Network algorithm by using various architecture to perform Balinese script character recognition with limitation that it’s only able to recognize wreastra script, pengangge suara and tengenan. The system will conduct segmentation process to get each character that is applied using an image pre-processing technique called projection profile segmentation. Three Balinese script classification models that are each built using MobileNet V2, ResNet50, and Xception architecture obtain 99.8% accuracy rate. The Balinese script transliteration system then tested using 100 testing images to calculate the Levenshtein Distance of the transliteration results. The Levenshtein Distance score obtained is 85.9 for MobileNet V2 model, 86.34 for ResNet50 model, and 83.78 for Xception model. The system was validated by 15 respondents and it was concluded that the application could help users in recognizing the Balinese script into Latin script. The result of the t-test hypothesis which compared the transliteration score of the Balinese people before and after using the transliteration application showed that there was a difference in the form of a significant increase in the community's literacy level after using the application, so it was concluded that the application can help improve user’s literacy in recognizing Balinese script.

**Keywords: Balinese script, Convolutional Neural Network, image pre-processing, recognition**

vi

**KATA PENGANTAR**

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat serta kesempatan bagi saya untuk menyelesaikan tugas akhir yang berjudul “Pembuatan Aplikasi Sistem Pengenalan Aksara Bali Menggunakan Metode *Convolutional Neural Network*”. Penulisan tugas akhir ini tidak dapat berjalan dengan baik tanpa adanya kontribusi dari berbagai pihak. Oleh karena itu, melalui kata pengantar ini, saya ingin menyampaikan rasa terima kasih yang sebesar-besarnya dan penghargaan yang mendalam kepada:

1. Bapak Dr. Joko Siswantoro, S.Si., M.Si., selaku Ketua Jurusan Teknik Informatika Universitas Surabaya, dosen wali, dan dosen pembimbing kedua tugas akhir saya yang telah memberikan bimbingan, arahan, serta masukan yang sangat berharga selama perkuliahan serta proses pembuatan dan penulisan tugas akhir ini.
2. Bapak Mohammad Farid Naufal, S.Kom., M.Kom., selaku dosen pembimbing pertama tugas akhir saya yang telah memberikan bimbingan, arahan, serta masukan yang sangat berharga dalam pembuatan serta penulisan tugas akhir ini.
3. Ibu I Gusti Ayu Made Sintia Dewi, S.Pd., selaku guru Bahasa Bali SMAK Santo Yoseph Denpasar yang telah bersedia meluangkan waktu menjadi validator dalam penelitian ini.
4. Seluruh responden yang telah berkenan meluangkan waktu serta memberikan tanggapan yang sangat berharga terhadap penelitian ini.

vii

1. Kedua orang tua saya yang telah memberikan semangat serta dukungan sehingga saya dapat menyelesaikan penelitian ini dengan baik.
2. Dan kepada semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa masih terdapat banyak kekurangan dan keterbatasan dalam penyusunan skripsi ini. Oleh karena itu, kami mengharapkan segala saran, kritik, dan masukan yang membangun untuk pengembangan di masa mendatang.

Akhir kata, penulis berharap skripsi ini dapat berkontribusi serta memberikan manfaat bagi pengembangan ilmu pengetahuan dan pembelajaran di bidang teknologi.

Surabaya, 3 Januari 2024

Penulis

viii

**DAFTAR ISI**

HALAMAN PENGESAHAN................................................................................ iii PERNYATAAN KEASLIAN................................................................................ iv ABSTRAK .............................................................................................................. v KATA PENGANTAR .......................................................................................... vii DAFTAR ISI.......................................................................................................... ix DAFTAR GAMBAR ........................................................................................... xiii DAFTAR TABEL................................................................................................ xvi DAFTAR LISTING ............................................................................................. xix BAB I PENDAHULUAN.....................................................................................I-1

1.1 LATAR BELAKANG ..........................................................................I-1

1.2 RUMUSAN MASALAH......................................................................I-4

1.3 TUJUAN ...............................................................................................I-5

1.4 MANFAAT...........................................................................................I-5

1.5 RUANG LINGKUP..............................................................................I-5

1.6 RENCANA KEGIATAN......................................................................I-6

1.7 SISTEMATIKA PENULISAN.............................................................I-7

BAB II DASAR TEORI...................................................................................... II-1

2.1 Analisis Sentimen................................................................................ II-1

2.2 *Crawling*.............................................................................................. II-2

2.3 *Preprocessing*...................................................................................... II-3

2.3.1 Case Folding.................................................................................... II-3

ix

2.3.2 Data Cleaning.................................................................................. II-4

2.3.3 Slang Words.................................................................................... II-4

2.3.4 Stopword Removal.......................................................................... II-5

2.3.5 Negation .......................................................................................... II-5

2.3.6 Stemming ........................................................................................ II-6

2.4 Pembobotan TF-IDF ........................................................................... II-6

2.5 Support Vector Machine ..................................................................... II-9

2.6 Evaluasi............................................................................................. II-14

2.7 Twitter ............................................................................................... II-17

2.8 Youtube............................................................................................. II-18

BAB III ANALISIS SISTEM ............................................................................ III-1

3.1 Analisis Kondisi Saat Ini.................................................................... III-1

3.2 Analisis Sistem Sejenis ...................................................................... III-9

3.3 Analisis Permasalahan ..................................................................... III-17

3.4 Analisis Kebutuhan Sistem .............................................................. III-17

BAB IV DESAIN SISTEM ...............................................................................IV-1

4.1 Desain Data ........................................................................................IV-1

4.1.1 Tabel Data *Users*............................................................................IV-2

4.1.2 Tabel Data Komentar .....................................................................IV-2

4.1.3 Tabel Data Komentar Sementara ...................................................IV-3

4.2 Desain Proses .....................................................................................IV-4

4.2.1 Desain Proses Umum .....................................................................IV-4

4.2.2 Desain Proses Ambil Data (Crawling)...........................................IV-5

x

4.2.3 Desain Proses Preprocessing..........................................................IV-5

4.2.4 Desain Proses Feature Extraction...................................................IV-6

4.2.5 Desain Proses Klasifikasi...............................................................IV-6

4.3 Desain Tampilan ................................................................................IV-7

4.3.1 Desain Tampilan Halaman Login ..................................................IV-8

4.3.2 Desain Tampilan Halaman Beranda...............................................IV-9

4.3.3 Desain Tampilan Halaman Ambil data ........................................IV-11

4.3.4 Desain Tampilan Halaman Komentar..........................................IV-14

4.3.5 Desain Tampilan Halaman Cek Komentar ..................................IV-15

4.3.6 Desain Tampilan Halaman Validasi.............................................IV-16

BAB V IMPLEMENTASI SISTEM................................................................... V-1

5.1 Implementasi Data .............................................................................. V-1

5.1.1 Implementasi Data Komentar ......................................................... V-1

5.1.2 Implementasi Data User.................................................................. V-2

5.1.3 Implementasi Data Komentar Sementara........................................ V-3

5.2 Implementasi Proses............................................................................ V-4

5.2.1 Implementasi Proses Crawling........................................................ V-5

5.2.2 Implementasi Proses Preprocessing ................................................ V-8

5.2.3 Implementasi Proses Feature Extraction....................................... V-11

5.2.4 Implementasi Proses Klasifikasi ................................................... V-13

BAB VI UJI COBA DAN EVALUASI.............................................................VI-1

6.1 Verifikasi............................................................................................VI-1

6.1.1 Fitur Login .....................................................................................VI-1

xi

6.1.2 Fitur Ambil Data ............................................................................VI-3

6.1.3 Fitur Cek Komentar......................................................................VI-11

6.1.4 Fitur Tampilan Semua Komentar.................................................VI-12

6.1.5 Fitur Validasi................................................................................VI-13

6.1.6 Verifikasi Kebenaran Sistem........................................................VI-15

6.1.7 Verifikasi Perhitungan Metode Support Vector Machine............VI-23

6.2 Validasi ............................................................................................VI-36

6.2.1 Validasi dengan Pihak KEMENDIKBUDRISTEK.....................VI-36

6.2.2 Validasi dengan Pihak Guru, Orang Tua, dan Siswa ...................VI-38

BAB VII KESIMPULAN DAN SARAN........................................................ VII-1

7.1 Kesimpulan ...................................................................................... VII-1

7.2 Saran................................................................................................. VII-2

DAFTAR PUSTAKA ........................................................................................... xx

LAMPIRAN........................................................................................................ xxv

x

**DAFTAR GAMBAR**

Gambar 3.1. Tampilan *Spreadsheet* Transaksi Pembelian........................ 3-2

Gambar 3.2. Tampilan *Spreadsheet* *Detail* Transaksi Pembelian ............ 3-3

Gambar 3.3. Tampilan *Spreadsheet* Transaksi Penjualan......................... 3-3

Gambar 3.4. Tampilan *Spreadsheet Detail* Transaksi Penjualan ............. 3-4

Gambar 3.5. Tampilan *Spreadsheet* Daftar Barang................................... 3-5

Gambar 3.6. Tampilan *Spreadsheet* Daftar Mutasi Barang....................... 3-5

Gambar 4.1. Desain ERD *Database* ......................................................... 4-2

Gambar 4.2. Alur Proses *Login*.................................................................4-6

Gambar 4.3. Alur Proses Penambahan Data Transaksi Pembelian...........4-7

Gambar 4.4. Alur Proses Ubah Status Pembayaran pada Transaksi Pembelian...................................................................................................4-7

Gambar 4.5. Alur Proses Ubah Status Pengiriman Hingga Hitung Harga Jual Barang ........................................................................................................4-8

Gambar 4.6. Alur Proses Penambahan Data Transaksi Penjualan ............4-9

Gambar 4.7. Alur Proses Ubah Status Pembayaran pada Transaksi Penjualan..................................................................................................4-10

Gambar 4.8. Alur Proses Ubah Status Pengiriman pada Transaksi Penjualan ..................................................................................................................4-11

Gambar 4.9. Alur Proses Opname Stok Barang ......................................4-12

Gambar 4.10. Desain Halaman *Login*......................................................4-12

Gambar 4.11. Desain Halaman *Home* ......................................................4-13

Gambar 4.12. Desain Halaman *Master* ....................................................4-14

Gambar 4.13. Desain Halaman Transaksi Pembelian..............................4-14

Gambar 4.14. Desain Halaman *Detail* Transaksi Pembelian ...................4-15

Gambar 4.15. Desain Halaman Tambah Transaksi Pembelian ...............4-15

Gambar 4.16. Desain Halaman Transaksi Penjualan ...............................4-16

Gambar 4.17. Desain Halaman *Detail* Transaksi Penjualan....................4-16

Gambar 4.18. Desain Halaman Tambah Transaksi Penjualan .................4-17

Gambar 4.19. Desain Halaman *Customer* ...............................................4-17

Gambar 4.20. Desain Halaman *Supplier* ..................................................4-18

Gambar 6.1. Tampilan *Login* Pengguna....................................................6-2

Gambar 6.2. Tampilan Halaman Utama....................................................6-2

Gambar 6.3. Tampilan Halaman Tambah Transaksi Pembelian...............6-3

Gambar 6.4. Tampilan Halaman Transaksi Pembelian Setelah Dilakukan Penambahan ...............................................................................................6-4

Gambar 6.5. Tampilan Halaman *Detail* Transaksi Pembelian..................6-4

Gambar 6.6. Tampilan Ubah Status Pembayaran pada Transaksi Pembelian...................................................................................................6-4

Gambar 6.7. Tampilan Ubah Status Pengiriman pada Transaksi Pembelian...................................................................................................6-5

Gambar 6.8. Tampilan Transaksi Pembelian Setelah Ubah Status Pembayaran dan Pengiriman......................................................................6-5

Gambar 6.9. Tampilan Faktur Beli............................................................6-5

Gambar 6.10. Tampilan Halaman Tambah Transaksi Penjualan..............6-6

Gambar 6.11. Tampilan Halaman Transaksi Penjualan Setelah Dilakukan Penambahan...............................................................................................6-7

Gambar 6.12. Tampilan Halaman *Detail* Penjualan..................................6-7

Gambar 6.13. Tampilan Ubah Status Pembayaran pada Transaksi...........6-7

Gambar 6.14. Tampilan Ubah Status Pengiriman pada Transaksi Penjualan ....................................................................................................................6-8

Gambar 6.15. Tampilan Transaksi Penjualan Setelah Ubah Status Pembayaran dan Pengiriman......................................................................6-8

Gambar 6.16. Tampilan *Invoice*................................................................6-8

Gambar 6.17. Tampilan Halaman Daftar Barang......................................6-9

Gambar 6.18. Tampilan Halaman Opname Stok.......................................6-9

Gambar 6.19. Tampilan Hasil Uji Coba Opname Stok............................6-10

Gambar 6.20. Tampilan Halaman Daftar *Supplier*...................................6-10

Gambar 6.21. Tampilan Halaman Tambah *Supplier*................................6-11

Gambar 6.22. Tampilan Halaman *Edit Supplier*......................................6-11

Gambar 6.23. Tampilan Halaman Daftar *Customer*.................................6-12

Gambar 6.24. Tampilan Halaman Tambah *Customer*..............................6-12

Gambar 6.25. Tampilan Halaman *Edit Customer*....................................6-12

xvii

xvi

xiv

xv

**DAFTAR TABEL**

Tabel 2. 1 Contoh Input dan Output Tahapan Case Folding.............................. II-4

Tabel 2. 2 Contoh Input dan Output Tahapan Data Cleaning............................. II-4

Tabel 2. 3 Contoh Input dan Output Tahapan Slang Words ............................... II-5

Tabel 2. 4 Contoh Input dan Output Tahapan Negation ..................................... II-5

Tabel 2. 5 Contoh Input dan Output Tahapan Stopword Removal.................... II-6

Tabel 2. 6 Contoh Input dan Output Tahapan Stemming ................................... II-6

Tabel 2. 7 Contoh Perhitungan Pembobotan TF-IDF ......................................... II-9

Tabel 2. 8 Contoh Perhitungan SVM................................................................ II-13

Tabel 2. 9 Confusion Matrix ............................................................................. II-15

Tabel 2. 10 Contoh Data Uji ............................................................................. II-15

Tabel 2. 11 Contoh Perhitungan Presisi............................................................ II-16

Tabel 2. 12 Contoh Perhitungan Recall ............................................................ II-17

Tabel 4. 1 Tabel Data Users...............................................................................IV-2

Tabel 4. 2 Tabel Data Komentar ........................................................................IV-2

Tabel 4. 3 Tabel Data Komentar Sementara ......................................................IV-3

Tabel 6. 1 Tabel Evaluasi Dataset dengan Kamus Slangword Awal...............VI-16

Tabel 6. 2 Hasil Evaluasi Dataset dengan Kamus Slangword Modifikasi.......VI-17

Tabel 6. 3 Contoh 10 Kalimat Uji....................................................................VI-18

Tabel 6. 4 Hasil Evaluasi Pengujian 10 Kalimat..............................................VI-19

Tabel 6. 5 Confussion Matrix 10 Kalimat Uji per Kernel................................VI-22

Tabel 6. 6 Perhitungan Evaluasi 10 Kalimat Uji..............................................VI-23

Tabel 6. 7 Contoh 10 Kalimat Perhitungan......................................................VI-23

xvi

Tabel 6. 8 Perhitungan TF-IDF........................................................................VI-24

Tabel 6. 9 Fungsi Kernel One vs One..............................................................VI-26

Tabel 6. 10 Perhitungan Fungsi Kernel Model 1 .............................................VI-26

Tabel 6. 11 Perhitungan Fungsi Kernel Model 2 .............................................VI-27

Tabel 6. 12 Perhitungan Fungsi Kernel Model 3 .............................................VI-27

Tabel 6. 13 Perhitungan Matrix Hessian untuk Model 1 .................................VI-28

Tabel 6. 14 Perhitungan Matrix Hessian untuk Model 2 .................................VI-29

Tabel 6. 15 Perhitungan Matrix Hessian untuk Model 3 .................................VI-29

Tabel 6. 16 Perhitungan Nilai Eror Model 1....................................................VI-30

Tabel 6. 17 Perhitungan Nilai Eror Model 2....................................................VI-30

Tabel 6. 18 Perhitungan Nilai Eror Model 3....................................................VI-30

Tabel 6. 19 Perhitungan Delta Alpha Model 1 ................................................VI-31

Tabel 6. 20 Perhitungan Delta Alpha Model 2 ................................................VI-31

Tabel 6. 21 Perhitungan Delta Alpha Model 3 ................................................VI-32

Tabel 6. 22 Perhitungan Alpha Baru Model 1 .................................................VI-32

Tabel 6. 23 Perhitungan Alpha Baru Model 2 .................................................VI-32

Tabel 6. 24 Perhitungan Alpha Baru Model 3 .................................................VI-33

Tabel 6. 25 Perhitungan W+, W-, dan Bias Model 1.......................................VI-33

Tabel 6. 26 Perhitungan W+, W-, dan Bias Model 2.......................................VI-34

Tabel 6. 27 Perhitungan W+, W-, dan Bias Model 3.......................................VI-34

Tabel 6. 28 Perhitungan Fungsi Kernel Kalimat 13 Model 1 ..........................VI-34

Tabel 6. 29 Perhitungan Fungsi Kernel Kalimat 13 Model 2 ..........................VI-34

Tabel 6. 30 Perhitungan Fungsi Kernel Kalimat 13 Model 3 ..........................VI-35

xvii

Tabel 6. 31 Hasil Perhitungan Hyperplane Tiap Model ..................................VI-35

Tabel 6. 32 Hasil Validasi Pihak KEMENDIKBUDRISTEK.........................VI-37

Tabel 6. 33 Hasil Wawancara dengan Pihak Guru ..........................................VI-38

Tabel 6. 34 Hasil Wawancara dengan Pihak Orang Tua .................................VI-39

Tabel 6. 35 Hasil Wawancara dengan Siswa ...................................................VI-39

xviii

**DAFTAR LISTING**

Listing 5. 1 Potongan Kode Program Proses Crawling Twitter.......................... V-5

Listing 5. 2 Potongan Kode Program Proses Crawling Youtube........................ V-6

Listing 5. 3 Potongan Kode Program Proses Preprocessing............................... V-8

Listing 5. 4 Potongan Kode Program Proses Feature Extraction...................... V-11

Listing 5. 5 Potongan Kode Program Proses Klasifikasi .................................. V-13

xix

**BAB 2**

**DASAR TEORI**

Pada bab ini akan dijelaskan berbagai macam teori yang diperlukan dalam pembuatan aplikasi sistem pengenalan aksara Bali. Teori yang dibahas yakni mengenai aksara Bali, metode-metode yang digunakan pada tahap *pre-processing*, serta algoritma *Convolutional Neural Network* (CNN) untuk klasifikasi citra.

**2.1 Aksara Bali**

Menurut Widiantana (2022), aksara Bali adalah sebuah tulisan aksara tradisional yang berasal dan berkembang di pulau Bali. Aksara ini adalah bentuk turunan dari aksara Brahmi yang berasal dari India melalui perantara aksara Kawi. Hingga saat ini, aksara Bali masih diajarkan dan digunakan dalam penulisan karya sastra serta aktivitas penulisan sehari-hari oleh masyarakat Bali sejak abad ke-15, walaupun telah terjadi pengurangan tingkat penerapan aksara Bali dalam aktivitas sehari-hari. Aksara Bali dikelompokkan menjadi beberapa bagian berdasarkan kegunaan atau fungsi mereka masing-masing. Berdasarkan bentuk dan fungsinya, aksara Bali dibedakan menjadi dua, yaitu aksara suci dan aksara biasa. Aksara suci hanya digunakan dalam hal yang bersifat keagamaan atau magis, sedangkan aksara biasa adalah jenis aksara Bali yang biasa digunakan dalam kehidupan sehari-hari. Salah satu aksara biasa yang paling sering digunakan oleh masyarakat Bali adalah aksara *wreastra*. Aksara *wreastra* merupakan salah satu jenis aksara Bali yang digunakan untuk menulis bahasa Bali biasa atau umum. Golongan aksara ini memiliki 18 huruf dan semuanya memiliki bentuk dan arti yang berbeda-beda. Berikut adalah tabel yang menggambarkan bentuk dan arti dari setiap aksara *wreastra*.

                                        Tabel 2.1. Kumpulan Aksara *Wreastra*

|  |  |  |
| --- | --- | --- |
| **No** | **Bentuk** | **Berbunyi** |
| 1 | h | A |
| 2 | n | Na |
| 3 | c | Ca |
| 4 | r | Ra |
| 5 | k | Ka |
| 6 | d | Da |
| 7 | t | Ta |
| 8 | s | Sa |
| 9 | w | Wa |
| 10 | l | La |
| 11 | m | Ma |
| 12 | g | Ga |
| 13 | b | Ba |
| 14 | \ | Nga |
| 15 | p | Pa |
| 16 | j | Ja |
| 17 | y | Ya |
| 18 | z | Nya |

Penulisan aksara *wreastra* umumnya melibatkan dua macam *pengangge*, yaitu *pengangge* *suara* dan *tengenan*. *Pengangge suara* adalah kumpulan aksara yang penulisannya dapat disatukan dengan aksara lain untuk mengubah suara atau vokal dari suatu huruf aksara Bali. *Pengangge suara* terdiri dari *ulu, suku, taleng, tedong*, dan *pepet* yang memiliki karakteristik vokal masing-masing. Untuk penulisan *taleng* dan *tedong* dapat digunakan secara berpasangan sehingga berbunyi o. Jika suatu aksara *wreastra* dibubuhi dengan *pengangge suara*, maka akan terjadi perubahan vokal pada aksara tersebut. Sebagai contoh, jika aksara Na dibubuhi dengan *pengangge ulu* yang berbunyi i, maka aksara tersebut akan dibaca sebagai Ni. Berikut adalah tabel yang berisi tulisan dan arti dari setiap *pengangge suara*.

           Tabel 2.2. Kumpulan *Pengangge Suara*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Bentuk** | **Nama** | **Berbunyi** |
| 1 | i | Ulu | i |
| 2 | u | Suku | u |
| 3 | e | Taleng | é |
| 4 | o | Tedong | o |
| 5 | ) | Pepet | ě |

Berbeda dengan *pengangge suara*, *pengangge tengenan* berfungsi untuk memberikan huruf tambahan atau menghilangkan vokal dari suatu aksara. *Pengangge tengenan* terdiri dari *bisah, cecek,* dan *surang*. Sebagai contoh, jika aksara Ma dibubuhi dengan aksara *cecek* yang berbunyi ng, maka aksara tersebut akan dibaca sebagai Mang. Tulisan dan arti dari setiap *pengangge tengenan* dapat dilihat pada tabel berikut.

  Tabel 2.3. Kumpulan *Pengangge Tengenan*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Bentuk** | **Nama** | **Berbunyi** |
| 1 | ; | Wisah/Bisah | h |
| 2 | ( | Surang | r |
| 3 | \* | Cecek | ng |

**2.2 *Data Pre-Processing***

Menurut Stefanus (2020), *data pre-processing* adalah suatu tahapan untuk mengubah data mentah menjadi data dengan format yang lebih berguna dan efisien agar data tersebut dapat digunakan untuk membentuk model sehingga hasil akurasi yang didapatkan menjadi lebih maksimal. Pada *pre-processing* data citra, tahapan *pre-processing* yang perlu diterapkan terdiri dari *grayscale, gaussian blurring, projection profile segmentation, thresholding*, dan dilasi.

**2.2.1 *Grayscale***

Menurut Stefanus (2020), *Grayscale* adalah proses untuk mengubah format suatu citra menjadi format dengan rentang warna hitam hingga putih. Proses tersebut diperlukan untuk menyederhanakan model citra. Citra berwarna memiliki 3 layer matriks yaitu R, G, dan B *layer* sehingga pada proses-proses selanjutnya juga perlu memperhatikan ketiga *layer* tersebut. Hal tersebut menyebabkan terjadinya perulangan proses perhitungan yang sama sebanyak 3 kali. Oleh karena itu, diperlukan proses pengubahan 3 *layer* matriks menjadi 1 *layer* yang menghasilkan citra *grayscale*.

Citra *grayscale* hanya memiliki derajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matriks r, g, dan b menjadi citra *grayscale* dengan nilai s, maka proses konversi dapat diterapkan dengan mengambil rata-rata nilai r, g, dan b  seperti pada persamaan (2.1) (Stefanus, 2020).

*s =*            **(2.1)**

**2.2.2 *Gaussian Blurring***

Menurut Tulasi (2023), *Gaussian blurring* merupakan sebuah fungsi yang diterapkan untuk mengurangi *noise* atau derajat pada citra dengan menambahkan efek *blur*. Dengan ditambahkannya efek tersebut, *noise* akan berkurang dengan menghapus beberapa detail pada gambar yang tidak diinginkan. Proses *Gaussian blurring* atau *gaussian filtering* dilakukan melalui proses konvolusi, yaitu perkalian antara matriks *kernel gaussian* dengan matriks citra (Yuwono, 2015). Matriks *kernel gaussian* diperoleh dari formula distribusi *gaussian* seperti pada persamaan (2.2) di bawah ini (Muhammad, 2007).

*G(x,y)*       **(2.2)**

Keterangan:

e = Bilangan konstan euler (2,71828…)

π = simbol *phi* yang bernilai (3,14…)

σ = standar deviasi dari distribusi *gaussian*

*G(x,y)* = elemen matriks *kernel gaussian* pada posisi (x,y)

*x =* jarak dari titik awal ke sumbu x

y = jarak dari titik awal ke sumbu y

Setelah itu, dilakukan proses perkalian antara bobot matriks citra dengan bobot matriks *kernel gaussian* melalui persamaan (2.3) berikut ini  (Yuwono, 2015).

*Pixel B(i,j)* =  **(2.3)**

Keterangan:

*Pixel A* = Gambar citra

*Pixel B(i,j)* = Bobot hasil perkalian pada posisi (i,j)

*N* = Jumlah kolom pada matriks *kernel*

*M* = Jumlah baris pada matriks *kernel*

*K* = Hasil penjumlahan semua bobot di G

*G(p,q)* = Elemen matriks *kernel gaussian* pada posisi (p,q)

Σ = Simbol *sigma* yang menyatakan penjumlahan

Citra akan dibagi menjadi 2 jenis *pixel*, yaitu *pixel* batas dan *pixel* dalam. Untuk *pixel* dalam, proses perkalian akan dilakukan menggunakan persamaan (3) diatas untuk menjadikan *pixel* yang dicari nilai barunya sebagai nilai tengah dan bobotnya akan dikalikan dengan bobot *pixel* tengah matriks *kernel*, hasilnya akan dijumlahkan dengan hasil perkalian bobot *pixel-pixel* tetangganya dengan bobot *pixel* matriks *kernel*. Untuk *pixel-pixel* batas, dilakukan proses pencarian bobot pada *pixel-pixel* luar terlebih dahulu. Bobot *pixel-pixel* luar ini dapat dicari dengan teknik interpolasi yang dilakukan dengan melihat dua *pixel* di dekatnya yang searah secara horizontal atau vertikal. Jika ada *pixel* yang mempunyai bobot lebih kecil dari 0, maka nilai bobotnya akan dijadikan 0. Jika ada *pixel* yang bobotnya lebih besar dari 255, maka bobotnya juga akan bernilai 255. Berikut adalah contoh proses konvolusi matriks citra dengan matriks *kernel gaussian*  (Yuwono, 2015).

Sebuah gambar berisi teks, Persegi, kotak

Deskripsi dibuat secara otomatis

Gambar 2.1 Ilustrasi Proses Konvolusi Matriks Citra dengan Matriks *Kernel*

Keterangan:

A = Matriks citra asal

G = Matriks *kernel gaussian*

B = Matriks hasil perkalian antara matriks A dan G

C = Matriks citra yang telah diberi efek *blur*

**2.2.3 *Projection Profile Segmentation***

Menurut Mamatarani (2023), proses segmentasi adalah tahapan penting dalam memperoleh setiap karakter dengan membagi citra atau memisahkan area pengamatan. Karakter-karakter yang diperoleh nantinya akan dimanfaatkan dalam proses klasifikasi. *projection profile* merupakan salah satu metode yang digunakan untuk segmentasi tiap karakter. *projection profile* adalah representasi satu dimensi dari citra dua dimensi. Jenis *projection* terdiri dari *horizontal projection* dan *vertical projection*. *Horizontal projection* digunakan pada proses *line segmentation* dengan menghitung jumlah total semua *pixel* di setiap baris. Pada proses *horizontal projection profile*, citra akan dikonversi terlebih dahulu menjadi citra biner. Dari proses tersebut, dihasilkan sebuah matriks dengan baris “m” dan kolom “n”. Total *pixel* di setiap baris dihitung menggunakan rumus pada persamaan (2.4). Besar dimensi *array* yang dihasilkan akan sama dengan jumlah baris pada citra.

*HPP*(x) = **(2.4)**

Keterangan:

*HPP*(x) = Fungsi *horizontal projection profile*

Σ = Simbol *sigma* yang menyatakan penjumlahan

*m* = Baris dari matriks hasil representasi citra tulisan

*n* = Kolom dari matriks hasil representasi citra tulisan

Dari perhitungan *pixel* tersebut, *histogram* dari suatu citra dapat dikonstruksi. Setelah itu, nilai *threshold* akan didapat dari mengukur celah antara dua *histogram* sesuai yang digunakan untuk memisahkan tiap *histogram* agar mendapatkan hasil segmentasi citra secara horizontal. Berikut adalah contoh hasil dari proses *horizontal projection* yang dilakukan terhadap salah satu citra tulisan tangan aksara Bali (Darma, 2019).

Sebuah gambar berisi cuplikan layar, teks, hitam dan putih, hitam

Deskripsi dibuat secara otomatis

Gambar 2.2 Contoh Hasil Proses *Horizontal Projection*

Dari hasil horizontal projection diatas, dilakukan proses *Vertical projection* yang diterapkan secara berulang tiap barisnya dengan menghitung jumlah semua *pixel*. Berikutnya, akan dihasilkan *vertical histogram* dari citra tersebut agar mendapatkan nilai *threshold* yang dimanfaatkan dalam proses pemisahan tiap *histogram*. Kemudian akan dihasilkan citra yang berisi karakter-karakter yang telah dipisahkan dari suatu baris. Berikut adalah contoh berupa hasil dari proses *vertical projection* tulisan tangan aksara Bali (Darma, 2019).



Gambar 2.3 Contoh Hasil Proses *Vertical Projection*

**2.2.4 *Thresholding***

Menurut Salem (2007), *thresholding* adalah salah satu metode yang sering digunakan dalam proses segmentasi citra karena dapat memisahkan objek dan latar pada citra. Proses pemisahan tersebut dilakukan dengan memilih nilai ambang batas yaitu T yang dapat mengkonversi citra keabuan menjadi citra biner. Semua informasi penting mengenai posisi dan bentuk suatu objek pada citra disimpan pada citra biner. Keuntungan dari dihasilkannya citra biner adalah mengurangi kompleksitas data dan menyederhanakan proses klasifikasi. Cara untuk mengkonversi citra keabuan menjadi citra biner adalah dengan memilih satu nilai ambang batas (T). Setelah proses tersebut, maka semua nilai keabuan di bawah atau sama dengan nilai T akan diklasifikasikan sebagai hitam (0) dan yang memiliki nilai di atas T akan menjadi putih (1). Persamaan (2.5) berikut mendefinisikan metode *thresholding* (Salem, 2007).

T = T[x, y, p(x,y), f(x,y)]     **(2.5)**

Keterangan:

  T = Nilai *threshold*

x,y = Koordinat dari titik nilai *threshold*

p(x,y), f(x,y) = Titik *pixel* citra keabuan

Salah satu metode untuk menentukan nilai *threshold* yang optimal adalah dengan menggunakan metode Otsu. Menurut Yousefi (2011), metode tersebut mengasumsikan bahwa citra hanya terdiri dari objek (latar depan) dan latar belakang. Otsu mengatur nilai *threshold* dengan sedemikian rupa sehingga dapat meminimalkan terjadinya tumpang tindih pada distribusi kelas. Berdasarkan definisi diatas, metode Otsu mensegmentasi citra menjadi dua wilayah, yaitu terang dan gelap (T0 dan T1). Wilayah terang (T0) berisi nilai intensitas dari yang bernilai 0 hingga t, sedangkan wilayah gelap (T1) memiliki nilai intensitas dengan rentang nilai t hingga l, yaitu nilai keabuan maksimum pada citra. Metode Otsu mengamati semua kemungkinan *thresholding* dan memilih nilai *threshold* dengan *within class variance* yang paling minimum. *Within class variance* didefinisikan pada persamaan (2.6) berikut (Yousefi, 2011).

  wb(t) \* b(t) + wf(t) \* f(t)       **(2.6)**

Dengan:

wb(t) =             **(2.7)**

wf(t) =     **(2.8)**

μb(t) =     **(2.9)**

μf(t) = **(2.10)**

Keterangan:

wb(t) = Bobot kelas T0

μb(t) = Mean kelas T0

b(t) = Varians kelas T0

wf(t) = Bobot kelas T1

μf(t) = Mean kelas T1

f(t) = Varians kelas T1

w = Total bobot dari varians kelompok

*P(i)* = Peluang munculnya nilai intensitas i

**2.2.5 Dilasi**

Dilasi adalah sebuah metode yang menyebabkan objek pada suatu citra biner mengalami penebalan. Hal tersebut dipengaruhi oleh *kernel* dilasi yang menentukan tingkat ketebalan suatu objek. Langkah-langkah *image processing* sebelumnya seperti *filtering* dan *thresholding* dapat menyebabkan kerusakan pada objek suatu citra. Maka dari itu, proses dilasi kemudian diterapkan untuk memperbaiki kerusakan umumnya berupa adanya keretakan atau jarak pada suatu objek citra. Persamaan (2.11) menunjukkan proses dilasi dari A terhadap B (Soille, 2004).

A ⊗ B = {z | (Bs)z ∩ A ≠ Ø}   **(2.11)**

Keterangan:

⊗ = Simbol yang menyatakan operasi dilasi

A = Citra *input*

B = *Kernel* dilasi

z = Koordinat *pixel* pada citra *input*

(Bs)z = Proses pergeseran *kernel* menuju koordinat tertentu pada citra *input*

(Bs)z ∩ A ≠ Ø = Fungsi pengecekan untuk mendeteksi apakah terdapat tumpang tindih antara kedua himpunan. Jika ada, maka kondisinya terpenuhi.

**2.3 *Convolutional Neural Network* (CNN)**

Menurut Saxena (2022), algoritma *Convolutional Neural Network* (CNN) merupakan salah satu algoritma *deep learning* yang sejalan dengan algoritma *Artificial Neural Network* (ANN). *Convolutional Neural Network* (CNN) sejalan dengan ANN tradisional karena keduanya terdiri dari banyak *neuron* yang dapat mengoptimalkan dirinya sendiri melalui proses pembelajaran. Setiap *neuron* akan menerima *input* dan melakukan operasi perkalian diikuti dengan sebuah fungsi *non-linear*. Dari *input* berupa vektor citra mentah menuju *output*, seluruh jaringan masih akan memanfaatkan fungsi skor persepsi tunggal atau bobot. Pada lapisan terakhir berisi *loss function* yang berkaitan dengan kelas dan semua kaidah reguler yang dikembangkan untuk ANN tradisional masih berlaku. Perbedaan utama antara CNN dan ANN adalah CNN umumnya digunakan pada bidang klasifikasi data gambar. Perbedaan lainnya adalah setiap node pada ANN terpisah satu sama lain, sedangkan node-node pada algoritma CNN saling terhubung sehingga lebih mudah memproses data gambar. CNN memiliki 3 macam *layer* utama yang menjalankan fungsi mereka masing-masing. *Layer-layer* tersebut yakni *convolution layer, pooling layer,* dan *fully connected layer*.

**2.3.1 *Convolution Layer***

Menurut Rismiyati (2021), lapisan konvolusi umumnya memiliki ukuran kernel 3x3. Setiap lapisannya dapat memiliki jumlah *filter* yang berbeda satu sama lain. Pada lapisan tertentu, diterapkan proses pooling  menggunakan metode *max pooling* yang hasil akhirnya dihubungkan ke *fully connected layer* dan selanjutnya akan terhubung ke *classifier* menggunakan metode *softmax*. Tujuan akhirnya adalah untuk menghasilkan probabilitas kelas *output*. Kelas dengan probabilitas tertinggi akan dipilih sebagai *output*.

Di setiap lapisan konvolusi terdapat operasi *convolution* yang dilakukan dengan menggeser *filter* di atas *input*. Pada setiap lokasi, diterapkan proses perkalian nilai *pixel* citra *input* dengan nilai *pixel* citra *output* dan semua hasilnya akan dijumlahkan ke dalam peta fitur. Beberapa operasi *convolution* pada citra *input* dilakukan menggunakan filter yang berbeda-beda. Hasilnya juga akan diletakkan dalam peta fitur yang berbeda. Pada akhirnya, semua peta fitur akan digabungkan sebagai hasil akhir dari proses *convolution*. Pada operasi *convolution* di *layer* pertama, diterapkan proses perkalian atau *dot product* antara *filter* dengan sebuah blok pada citra yang masing-masing memiliki ukuran yang sama. Rumus dari proses *convolution* di *layer* pertama dapat dilihat pada persamaan (2.12) di bawah ini (Alzubaidi *et al.*, 2021).

𝐶𝑜𝑛𝑣𝑥,𝑦 = (𝐼 ⊗ 𝐶 + 𝐵𝑗), dimana ⊗ = ∑𝑢=1 ∑ 𝑣=1  𝐼𝑥+𝑢,𝑦+𝑣 ∗ 𝐶𝑢,𝑣   **(2.12)**

Keterangan:

*I* = matriks dari citra *input*

*C* = matriks dari *convolution filter*

*Bj* = nilai *bias* ke-j

*Ix,y* = nilai *pixel* dari *input* ke (x,y)

*Cu,v* = nilai *pixel* dari *filter* ke (u,v)

⊗ = simbol yang menyatakan operasi *convolution.*

∑ = simbol *sigma* yang menyatakan penjumlahan

𝐶𝑜𝑛𝑣𝑥,𝑦 = Hasil dari proses *convolution* pada *layer* pertama

Pada operasi *convolution* di *layer-layer* selanjutnya, terdapat perubahan pada persamaan yang digunakan. Setiap blok dari hasil proses *convolution* pada *layer* sebelumnya akan dikalikan dengan *convolution filter* dan semua hasilnya akan dijumlahkan*.* Ukuran dari setiap blok akan disesuaikan dengan ukuran dari *filter* yang digunakan. Proses *convolution* pada *layer-layer* berikutnya dilakukan menggunakan persamaan (2.13) berikut ini (Alzubaidi *et al.*, 2021).

𝐶𝑜𝑛𝑣𝑗 = ∑i=1 𝑝𝑜𝑜𝑙𝑖 ⊗ 𝐶𝑖,𝑗 + 𝐵𝑗    **(2.13)**

Keterangan:

*Convj* = *Convolution layer* ke-j

∑ = Simbol *sigma* yang menyatakan penjumlahan

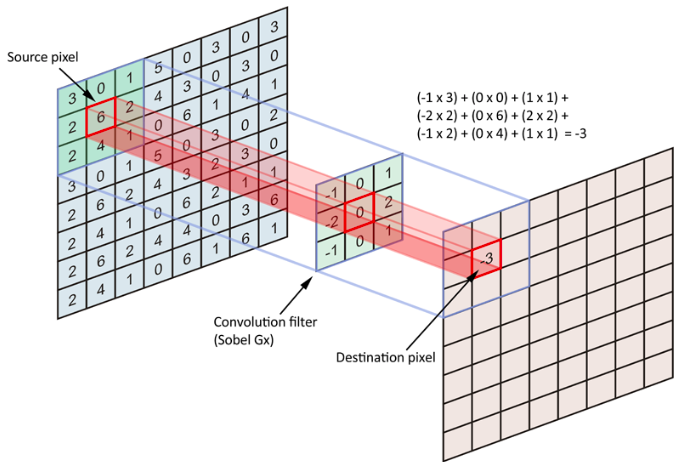
*pooli* = *Pooling layer* ke-i

*Ci,j* = *Kernel* indeks ke-j dan *filter* ke-i

Bj = *Bias* ke-j

⊗ = Simbol yang menyatakan operasi *convolution*

Pada awal operasi *convolution*, *filter* akan ditempatkan pada *pixel* pertama citra *input*, kemudian dilakukan proses perkalian setiap angka pada *filter* dengan beberapa angka pada citra *input*. Jumlah angka yang terlibat pada proses perkalian dipengaruhi oleh ukuran dari *filter* yang digunakan. Selanjutnya, semua hasil perkalian akan dijumlahkan untuk mendapatkan angka tunggal. Angka tersebut akan diinputkan pada citra *output* dengan posisi yang sesuai dengan *pixel* citra *input* yang dipilih pada proses penempatan *filter* sebelumnya. Setelah mendapatkan nilai intensitas pada *pixel* pertama citra *output*, *filter* akan bergeser sejauh satu *pixel* dan proses perkalian akan kembali dijalankan. Proses tersebut diterapkan pada setiap *pixel* di citra *input*, kecuali untuk *pixel* yang berada di tepi. *Pixel-pixel* tersebut akan digunakan untuk menghasilkan citra *output* dengan ukuran yang sama seperti citra *input*. Ilustrasi dari operasi *convolution* dapat dilihat pada gambar di bawah ini.



Gambar 2.4 Ilustrasi Proses *Convolution*

**2.3.2 *Pooling Layer***

Setelah proses *convolution*, dilakukan tahap penyatuan (*pooling*) terhadap peta fitur yang dihasilkan dari proses *convolution.* Setiap *pooling layer* dapat menggunakan metode yang berbeda-beda, yakni diantara *average pooling* atau *max pooling* yang keduanya berfungsi untuk mengurangi dimensi serta jumlah parameter dan komputasi dalam jaringan agar dapat mempersingkat waktu *training* dan mencegah terjadinya *overfitting*. Pada metode *max pooling*, dilakukan proses pengambilan nilai maksimum di setiap blok untuk mengurangi ukuran peta fitur dan menyimpan informasi yang penting dan signifikan untuk proses klasifikasi. Ukuran dari setiap blok perlu ditentukan sebelumnya. Berikut merupakan rumus dari metode *max pooling* yang dituliskan pada persamaan (2.14) (Alzubaidi *et al.*, 2021).

𝑃𝑜𝑜𝑙𝑥,𝑦 = 𝑀𝑎𝑥(𝐶𝑜𝑛𝑣𝑥,𝑦, 𝐶𝑜𝑛𝑣𝑥+1,𝑦, 𝐶𝑜𝑛𝑣𝑥,𝑦+1, 𝐶𝑜𝑛𝑣𝑥+1,𝑦+1)   **(2.14)**

Keterangan:

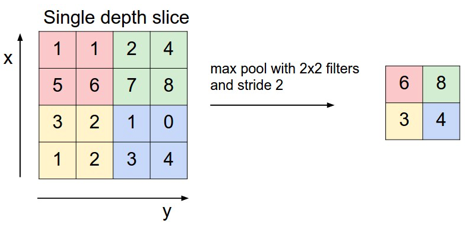
𝐶𝑜𝑛𝑣𝑥,𝑦 = nilai pada peta fitur dengan posisi *pixel* (x,y)

𝐶𝑜𝑛𝑣𝑥+1,𝑦 = nilai pada peta fitur di sebelah kanan *pixel* (x,y)

𝐶𝑜𝑛𝑣𝑥,𝑦+1 = nilai pada peta fitur di sebelah bawah *pixel* (x,y)

𝐶𝑜𝑛𝑣𝑥+1,𝑦+1 = nilai pada peta fitur di sebelah kanan bawah *pixel* (x,y)

𝑃𝑜𝑜𝑙𝑥,𝑦 = Hasil proses *pooling*



Gambar 2.5 Ilustrasi Proses *Pooling* Menggunakan *Max Pooling*

**2.3.3 *Fully Connected Layer***

Setelah melewati *pooling layer*, tahap berikutnya adalah klasifikasi yang terdiri dari beberapa *fully connected layer* yang hanya dapat menerima data berdimensi 1. Maka dari itu, diperlukan fungsi *flatten* yang dapat mengkonversi data 3 dimensi menjadi 1 dimensi. Berikut adalah ilustrasi proses konversi yang Sebuah gambar berisi teks, cuplikan layar, diagram

Deskripsi dibuat secara otomatisdilakukan pada fungsi flatten.

Gambar 2.6 Ilustrasi Proses Flatten

Pada *fully connected layer*, setiap node terhubung dengan semua node pada layer sebelumnya. Operasi perhitungan yang terjadi pada *layer* ini adalah perkalian antara transpose dari matriks *w* yang berisi *weight* antar node pada *layer* dengan vektor input *x* yang hasilnya kemudian ditambahkan dengan bias *b* dan akhirnya diterapkan faktor penskalaan atau bobot *f*. Rumus dari operasi pada *fully connected layer* dapat dilihat pada persamaan (2.15) berikut ini (Alzubaidi *et al.*, 2021).

**(2.15)**

Keterangan:

*y* = Nilai *output* atau prediksi

*f* = faktor penskalaan atau bobot

= Transpose dari vektor *weight* atau bobot

*x* = vektor *input*

*b* = bias

**2.3.4 *Activation Function***

*Activation function* adalah sebuah fungsi yang pada umumnya terletak pada lapisan konvolusi dan lapisan *fully connected*. Fungsi tersebut memperkenalkan non-linearitas ke dalam jaringan sehingga dapat membentuk model yang mampu menangani relasi yang kompleks dan membuat prediksi yang lebih akurat. *Activation function* yang digunakan pada arsitektur MobileNet V2, ResNet50, dan Xception adalah *Rectified Linear Unit* (ReLU). ReLU menerapkan sebuah fungsi yang melakukan *thresholding* terhadap nilai *pixel* pada citra input dengan nilai 0. Fungsi ini diterapkan untuk mengubah seluruh nilai *pixel* yang kurang dari nol menjadi bernilai 0. Persamaan (2.16) menunjukkan persamaan dasar dari fungsi ReLU (Ilahiyah, 2018).

frelu(z) = max(0,z) **(2.16)**

**2.3.5 *Loss Function***

*Loss function* merupakan sebuah fungsi matematika yang menghitung perbedaan antara nilai prediksi (*output*) pada suatu model dan nilai target sebenarnya pada *dataset*. Tujuan utama dari diterapkannya *loss function* adalah untuk mengukur performa model dan memandu proses optimisasi selama proses *training*. Dalam hal klasifikasi data *multiclass*, *loss function* yang dapat digunakan yakni *cross-entropy loss* dan *softmax loss*. Pada fungsi *cross-entropy loss*, dilakukan proses perhitungan *loss* atau kerugian untuk setiap titik data dan setiap kelas serta kemudian menjumlahkan semua titik data dan kelas tersebut. Proses diatas menyebabkan tingkat probabilitas prediksi untuk kelas sebenarnya menjadi tinggi dan rendah untuk kelas lainnya. Persamaan (2.17) menjabarkan rumus perhitungan *cross-entropy loss* untuk data *multiclass* (Zhang, 2018).

L(x,y) =               **(2.17)**

Keterangan:

L(x,y) = Nilai *multiclass cross-entropy loss*

N = Jumlah titik data

K = Jumlah kelas

xi,y = Fungsi indikator yang bernilai 1 jika label kelas sebenarnya  dari titik data ke-i adalah kelas j, selain itu maka bernilai 0.

yi,j = Probabilitas prediksi bahwa titik data ke-i dimiliki oleh kelas j

**2.3.6 *Transfer Learning***

*Transfer learning* adalah suatu metode yang memanfaatkan *pre-trained model* untuk digunakan pada model baru. *pre-trained model* sendiri adalah suatu model terlatih yang telah mempelajari fitur-fitur dari kumpulan data atau *dataset* yang besar seperti ImageNet. Metode ini sangat efektif digunakan, terutama jika data yang dimiliki tergolong sedikit untuk menyelesaikan suatu permasalahan. Untuk algoritma CNN, telah tersedia beragam *pre-trained model* dengan berbagai macam arsitektur, seperti ResNet, VGGnet, MobileNet, dan AlexNet. Penggunaan *transfer learning* memberikan beberapa manfaat yaitu meningkatkan performa model, proses *training* yang lebih cepat, dan mampu mengurangi risiko *overfitting* walaupun bekerja dengan *dataset* yang lebih kecil.

**2.3.7 Arsitektur CNN**

Pengguna dapat mendesain arsitektur CNN mereka sendiri dari awal. Akan tetapi, tingkat performa yang didapat seringkali tidak maksimal. Berdasarkan penelitian-penelitian yang dilakukan sebelumnya, telah tersedia berbagai macam arsitektur CNN yang sudah dilatih menggunakan *dataset* yang besar dan kompleks dan terbukti mampu melakukan proses klasifikasi citra dengan tingkat performa yang lebih tinggi. Beberapa contoh arsitektur CNN yang sudah ada terdiri dari Xception, MobileNet, dan ResNet.

1. Xception

Menurut Chollet (2017), Arsitektur Xception merupakan hasil pengembangan dari arsitektur Inception. Perbedaan antara keduanya terletak pada penggunaan *depthwise separable convolution* pada arsitektur Xception yang menggantikan modul yang dipakai oleh arsitektur Inception. *depthwise separable convolution* adalah konvolusi spasial yang diterapkan secara independen pada setiap saluran suatu *input*. Xception kemudian melakukan proses konvolusi 1x1 untuk memproyeksikan saluran *output* dengan *depthwise convolution* menuju ruang saluran yang baru.

Sebuah gambar berisi teks, struk, Font, cuplikan layar

Deskripsi dibuat secara otomatis

Gambar 2.7 Arsitektur Xception

1. ResNet

Arsitektur ResNet dirancang menggunakan jaringan *ultra-deep* untuk mengatasi masalah hilangnya gradien. ResNet dikembangkan berdasarkan jumlah lapisan yang dimulai dari 34 lapisan hingga 1,202 lapisan. Jenis yang paling umum adalah ResNet50 yang terdiri dari 49 lapisan konvolusi ditambah dengan satu lapisan FC. Arsitektur ResNet adalah pemenang kejuaraan ILSVRC pada tahun 2015 dengan 152 lapisan. Kedalaman lapisan tersebut setara dengan 8 kali kedalaman VGG dan 20 kali kedalaman arsitektur AlexNet (Alzubaidi *et al.*, 2021).

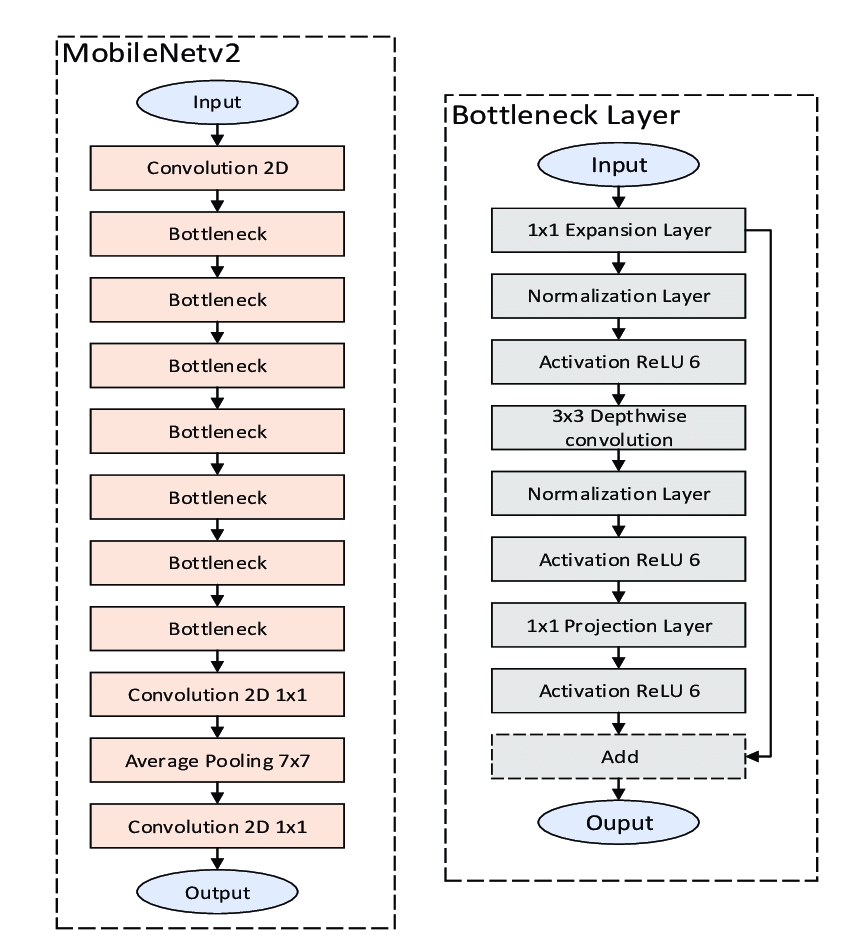
Sebuah gambar berisi teks

Deskripsi dibuat secara otomatis

Gambar 2.8 Arsitektur ResNet50

1. MobileNet

Arsitektur MobileNet menggunakan sebuah proses konvolusi yaitu *depthwise separable convolution* yang membagi proses konvolusi menjadi dua tahap berbeda yaitu *depthwise convolution* dan *pointwise convolution*. *depthwise convolution* memiliki filter-filter yang hanya menggunakan satu saluran *input* untuk melakukan proses konvolusi. *Pointwise convolution* mengintegrasikan *output* dari *depthwise convolution* dan melakukan proses konvolusi 1x1. Arsitektur ini memiliki beberapa blok residual yang disebut dengan bottleneck, dan setiap blok tersebut terdapat 3 *layer* yang terdiri dari *layer* konvolusi berdimensi 1x1 ditambah dengan fungsi aktivasi ReLU, *layer* *depthwise convolution* yang menggunakan fungsi aktivasi ReLU, dan *layer* terakhir adalah *layer* konvolusi (Sheng, 2020).



Gambar 2.9 Arsitektur MobileNet V2

**2.4 *Model Evaluation***

Model klasifikasi yang telah dibuat perlu dilakukan proses evaluasi untuk mengetahui seberapa tingkat performa dan efektivitas dari model tersebut. Beberapa metode evaluasi yang umum digunakan yakni (Zheng, 2015):

* *Confusion Matrix*: Metode evaluasi yang dapat menampilkan hasil prediksi berupa sebuah matriks untuk setiap kelas yang terlibat pada proses klasifikasi. Metode ini dapat memperlihatkan prediksi yang benar dan salah untuk setiap kelasnya. Bagian diagonal dari suatu *confusion matrix* mempresentasikan prediksi yang benar, kolom dari matriks menandakan hasil prediksi, dan baris dari matriks mewakili kelas sebenarnya. Berikut Sebuah gambar berisi teks, cuplikan layar, nomor, Font

  Deskripsi dibuat secara otomatisadalah contoh dari suatu *confusion matrix*.

Gambar 2.10 Contoh *Confusion Matrix* dengan 3 kelas

* *Accuracy*: Metode yang mengukur seberapa sering pengklasifikasi membuat prediksi yang benar. Hal tersebut diketahui dari proses perhitungan rasio antara jumlah prediksi yang benar dan total jumlah prediksi. Total prediksi didapatkan dari jumlah titik data yang terlibat dalam *test set*. Persamaan (2.18) menunjukkan rumus dasar dari metode *accuracy*.

*Accuracy* =   **(2.18)**

Keterangan:

*True Positive* (TP) = Jumlah data yang benar diprediksi sebagai bagian dari suatu kelas.

*True Negative* (TN) = Jumlah data yang benar diprediksi sebagai bagian dari kelas lainnya.

*Positive* (P) = Jumlah data yang merupakan bagian dari suatu kelas.

*Negative* (N) = Jumlah data yang merupakan bagian dari kelas lainnya.

* *K-Fold Cross Validation*: Metode validasi yang membagi *training set* menjadi k bagian. Setelah itu, dilakukan proses iterasi berdasarkan nilai k. Setiap iterasinya melayani kedua *test set* dan *validation set*, sedangkan k-1 bagian sisanya digunakan untuk proses *training*. tahap *training* akan dilakukan secara berulang hingga semua bagian pernah menjadi *test* data. Performa keseluruhan didapatkan dari rata-rata performa di semua *k-fold*.
* *Precision* dan *Recall*: *Precision* adalah metode yang mengukur keakuratan prediksi positif yang dibuat oleh suatu model. Metode tersebut menjawab pertanyaan yakni dari semua data yang diprediksi masuk ke suatu kelas, berapa banyak data yang benar-benar tergolong ke dalam kelas tersebut. Persamaan (2.19) menjabarkan rumus dari *precision*.

*Precision =*    **(2.19)**

Keterangan:

*True Positives* (TP) = Jumlah data yang benar diprediksi sebagai bagian dari suatu kelas.

*False Positives* (FP) =  Jumlah data yang sebenarnya bukan bagian dari suatu kelas tetapi salah diprediksi sebagai bagian dari kelas tersebut.

*Recall* adalah metode yang mengukur kemampuan suatu model untuk mengidentifikasi semua data yang relevan dari suatu kelas. Metode ini menjawab pertanyaan yakni dari semua data yang sebenarnya masuk ke suatu kelas, berapa banyak data yang diprediksi benar masuk kelas tersebut. Persamaan (2.20) memaparkan rumus dari metode *recall*.

*Recall =*  **(2.20)**

Keterangan:

*True Positive* (TP) = Jumlah data yang sebenarnya masuk ke suatu kelas dan diprediksi benar masuk kelas tersebut.

*False Negative* (FN) = Jumlah data yang sebenarnya masuk ke suatu kelas tetapi diprediksi bukan masuk kelas tersebut.

* F1 *Score*: Merupakan rata-rata dari *precision* dan *recall*. F1 *score* dihitung menggunakan rumus pada persamaan (2.21) berikut ini.

F1 *=*              **(2.21)**

**2.5 *Similarity Score***

*Similarity score* adalah suatu nilai yang menunjukkan tingkat kesamaan antara dua objek. Pada tahap uji coba pengenalan atau transliterasi secara keseluruhan sistem tugas akhir ini, diperlukan perhitungan *similarity score* untuk mengetahui apakah hasil transliterasi yang didapat sesuai dengan hasil yang sebenarnya. Metode *similarity score* yang digunakan pada tugas akhir ini adalah *Levenshtein Distance* yang dihitung dengan cara mengurangi nilai panjang maksimum kedua string dan jarak *Levenshtein* antara kedua *string*, kemudian dibagi dengan nilai panjang maksimum untuk mendapatkan nilai rasio. *Levenshtein Distance* digunakan karena memiliki kemampuan dalam menilai kemiripan atau perbedaan antara dua *string* sehingga cocok digunakan untuk menghitung *strings similarity*. Metode ini juga dapat membandingkan dua buah *string* dengan panjang atau jumlah karakter yang berbeda. Jika s dan t menandakan masing-masing *string*, dan m adalah panjang *string* pertama dan n adalah panjang *string* kedua, maka rumus dari rasio *Levenshtein Distance* dapat dijabarkan seperti persamaan (2.22) berikut (Khin, 2020).

*Similarity Ratio* =   **(2.22)**

Keterangan:

max(m,n) = Nilai maksimum dari panjang *string* pertama dan kedua

D = *Levenshtein Distance* antara *string* s dan t

**2.6 *Library***

*Library* adalah kumpulan metode-metode berupa kelas-kelas atau fungsi yang dapat dimanfaatkan pengguna dalam pembuatan program tertentu. *Library* biasanya terkumpul dalam sebuah *package* atau modul yang dapat diimpor ke program agar mendapatkan akses ke fungsi-fungsi yang ditawarkan. Setiap *library* memiliki penjelasan berupa dokumentasi contoh penggunaan dari kelas atau fungsi yang disediakan sehingga sangat membantu pengguna dalam menggunakan kelas-kelas tersebut. Pada pembuatan metode *pre-processing* data citra serta model klasifikasi CNN menggunakan bahasa pemrograman Python, *library-library* yang bisa dimanfaatkan pengguna terdiri dari *library* OpenCV, Scikit-learn, dan PyTorch.

**2.5.1 OpenCV**

OpenCV (*Open Source Computer Vision*) pertama kali diluncurkan pada tahun 1999 oleh tim Intel. Tujuan dari dibuatnya *library* ini adalah untuk mendukung pengembangan teknologi di bidang *computer vision*, yaitu salah satu ilmu pemrograman komputer yang berfokus pada pemrosesan dan pemahaman gambar dan video, dengan menyediakan berbagai macam *function-function* pemrograman dengan *code* yang teroptimalisasi, portabel, dan tersedia secara gratis. Hingga sekarang, OpenCV memiliki lebih dari 2,500 algoritma yang khusus digunakan dalam penyelesaian masalah di bidang *computer vision* dan *machine learning* seperti identifikasi wajah, pengenalan objek, melacak pergerakan kamera, menghasilkan gambar dengan resolusi tinggi, serta pengolahan citra digital (Ivan, 2012). *Library* OpenCV menyediakan banyak fungsi yang dapat digunakan dalam pengolahan citra digital seperti pengubahan *color space, blurring, filtering*, dan sebagainya.

**2.5.2 Scikit-learn**

Scikit-learn adalah sebuah modul yang terintegrasi dengan bahasa pemrograman Python yang menyediakan berbagai macam algoritma untuk menyelesaikan masalah-masalah yang berkaitan dengan *machine learning*. Algoritma-algoritma yang disediakan pada *library* ini dapat menyelesaikan berbagai masalah yang melibatkan analisis data dan statistik (Pedregosa, 2019). *Library* ini bersifat *open source* dan sering digunakan dalam melakukan *data pre-processing*, pemilihan model, model *fitting*, serta model *evaluation*. Pada model klasifikasi citra menggunakan algoritma CNN, *library* ini digunakan pada tahapan model *evaluation* untuk mengetahui tingkat akurasi, *precision, recall*, dan *f1 score* dari model yang telah dibuat.

**2.5.3 PyTorch**

PyTorch adalah *library open source* untuk bahasa pemrograman Python yang umum digunakan pada proses *deep learning*. *Library* PyTorch memiliki komputasi yang bersifat dinamis, mencakup ruang lingkup yang luas, dan mudah untuk digunakan sehingga dapat dimanfaatkan dalam berbagai pengaplikasian *deep learning* yang kompleks. Dikarenakan mendukung pemrograman *deep learning*, *library* PyTorch menyediakan berbagai macam fungsi dengan operasi matematika yang rumit serta dapat mengakomodasi arsitektur *neural network*. *Library* ini dapat berkontribusi dalam menyusun suatu algoritma *neural network* hingga membentuk sebuah model bahkan dapat menjalankan proses *training* model *deep learning* (Paszke, 2019).

**2.5.4 Flask**

Flask adalah sebuah *library* atau *framework* yang dirancang untuk pembuatan aplikasi *web* menggunakan bahasa pemrograman Python (Ghimire, 2020). Flask menerapkan spesifikasi WSGI (*Web Service Gateway Interface*) yang mendukung komunikasi antara *web server* dan aplikasi *web* Python melalui *library* Werkzeug. *Library* tersebut menyediakan alat untuk membangun aplikasi dan *middleware* yang berbasis WSGI.

parameter normalisasi L1 dan random state 114. Hasil uji nilai kebenaran menggunakan dataset oversampling dan parameter normalisasi L1

mendapatkan hasil yaitu 29 data dari 40 data bernilai benar. Berdasarkan hasil diatas maka sistem akan menggunakan teknik oversampling dan dengan parameter normalisasi L1.

• Hasil validasi yang telah dilakukan mendapatkan hasil bahwa sistem dapat membantu pemerintah dan masyarakat dalam mengetahui polaritas sentimen masyarakat terhadap PTM pada masa pandemi dan hasil laporan klasifikasi dapat dijadikan bahan dasar evaluasi untuk menentukan kebijakan selanjutnya.

• Fitur-fitur yang terdapat dalam sistem dalam membantu dalam hal mengumpulkan data komentar dan juga mengklasifikasikan komentar tersebut kedalam sentimen negatif, netral, dan positif.

**7.2 Saran**

Pada sub bab ini berisikan saran dan masukan dari hasil verifikasi dan validasi yang telah dilakukan untuk perkembangan sistem dikemudian hari. Beberapa saran yang didapatkan diantaranya:

• Akurasi yang didapatkan terbilang masih kurang dari 80% dikarenakan dataset yang terlalu banyak *noise* karena berasal dari sosial media*.* Maka dari itu dapat dilakukan penambahan kamus *slangword* dan juga *convert* emoji.

• Penambahan kategori komentar seperti komentar tersebut termasuk kedalam kategori fasilitas, pengajar, atau lain sebagainya supaya memudahkan untuk pengkategorian masalah.

• Penambahan fitur melihat komentar berdasarkan sekolah.