

CONVOLUTIONAL NEURAL NETWORK KLASIFIKASI BATIK TULIS MOTIF PARANG MENGGUNAKAN MOBILENET

TUGAS AKHIR

Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika Universitas Pembangunan Nasional “Veteran” Yogyakarta



Disusun Oleh :

NAMA : MUHAMMAD FATHURRAHMAN ZUHDI

NIM : 123150027

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" YOGYAKARTA
2021**

CONVOLUTIONAL NEURAL NETWORK KLASIFIKASI BATIK TULIS MOTIF PARANG MENGGUNAKAN MOBILENET

TUGAS AKHIR



Disusun Oleh :

NAMA : MUHAMMAD FATHURRAHMAN ZUHDI

NIM : 123150027

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" YOGYAKARTA
2021**

HALAMAN PENGESAHAN PEMBIMBING

**CONVOLUTIONAL NEURAL NETWORK KLASIFIKASI BATIK TULIS MOTIF
PARANG MENGGUNAKAN MOBILENET**

Disusun oleh:

Muhammad Fathurrahman Zuhdi

123150027

Telah diperiksa dan disetujui oleh pembimbing untuk disidangkan
pada tanggal:

Menyetujui,
Pembimbing I

Menyetujui,
Pembimbing II

Hidayatulah Himawan, S.T., M.M., M.Eng.

NIP. 1976 12 24 2005 01 1001

Dr. Herlina Jayadianti, S.T., M.T.

NIK. 2 7708 02 0235 1

Mengetahui,
Ketua Program Studi Informatika

Heriyanto, A.Md., S.Kom., M.Cs

NIK. 2 7706 11 0301 1

HALAMAN PENGESAHAN PENGUJI

**CONVOLUTIONAL NEURAL NETWORK KLASIFIKASI BATIK TULIS MOTIF
PARANG MENGGUNAKAN MOBILENET**

Disusun oleh:

Muhammad Fathurrahman Zuhdi
123150027

Telah diuji dan dinyatakan lulus pada tanggal oleh :

Menyetujui,
Pembimbing I

Pembimbing II

Hidayatulah Himawan, S.T., M.M., M.Eng.
NIP. 1976 12 24 2005 01 1001

DR. Herlina Jayadianti, S.T., M.T.
NIK. 2 7708 02 0235 1

Penguji III

Dr. Ayang Hendrianto Pratomo, S.T., M.T.
NIP. 1977 07 25 2005 01 1001

Penguji IV

Bambang Yuwono, S.T., M.T.
NIK. 2 7302 00 0225 1

PERNYATAAN BEBAS PLAGIAT

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Fathurrahman Zuhdi
NPM : 123150027
Fakultas/Prodi : Teknik Industri/Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir

CONVOLUTIONAL NEURAL NETWORK KLASIFIKASI BATIK TULIS MOTIF PARANG MENGGUNAKAN MOBILENET

Adalah hasil kerja saya sendiri dan benar bebas dari plagiat kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Yogyakarta, 28 Maret 2021
Yang Membuat Pernyataan



Muhammad Fathurrahman Zuhdi
NPM. 123150027

SURAT PERNYATAAN

KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan di bawah ini, saya :

Nama : Muhammad Fathurrahman Zuhdi

No. Mhs : 123150027

Menyatakan bahwa karya ilmiah saya yang berjudul:

CONVOLUTIONAL NEURAL NETWORK KLASIFIKASI BATIK TULIS MOTIF PARANG MENGGUNAKAN MOBILENET

Merupakan karya asli saya dan belum pernah dipublikasikan di manapun. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa pun yang diberikan Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada tanggal : 28 Maret 2021



Muhammad Fathurrahman Zuhdi

HALAMAN PERSEMBAHAN

Puji Syukur kepada Allah SWT Tuhan yang Maha Esa atas segala rahmat serta hidayahNya dalam keberhasilan menyelesaikan tugas akhir dan tentunya berakhir juga masa studi di Teknik Informatika UPN ‘Veteran’ Yogyakarta. Kupersembahkan karya sederhana ini beserta ucapan terimakasih kepada orang yang sangat kusayangi dan berperan dalam proses pembuatan tugas akhir ini.

- Kedua orang tua saya Ummi Siti Rahayu dan Ayah Nurachman Effendi yang selalu mendukung dan tiada hentinya mendoakan saya selama proses pengerjaan skripsi serta adik-adik saya.
- Para Dosen Informatika yang membantu menyelesaikan masa studi saya terutama Pak Hidayatullah Himawan, Bu Herlina Jayadianti, Pak Awang, Pak Bambang Yuwono dan dosen wali saya Bu Hafsah dan Pak Mangaras.
- Teman yang menemani saya berproses selama satu tahun belakangan (Bimek, Mamad, Vigar dan Reno) yang memberikan support dan motivasi kepada saya.
- Teman, sahabat dan pacar yang memberikan support dan motivasi kepada saya, Dyah Ayu Chariesta Permata Yuana.
- Keluarga “KANTOR FAMS 2015” yang selalu menjadi motivasi saya untuk segera lulus, banyak membantu saya dalam proses pengerjaan skripsi serta selalu membagikan canda tawa ketika skripsi saya tak kunjung selesai.
- Keluarga besar IF 2015 yang menjadi teman dalam suka maupun duka selama berkuliah di Informatika UPN ‘Veteran’ Yogyakarta.

ABSTRAK

Batik merupakan kesenian tradisional Indonesia yang telah diakui UNESCO pada tanggal 2 Oktober 2009 sebagai warisan budaya Indonesia pada kancah Internasional. Motif batik di Indonesia sangat beragam, batik *parang* merupakan salah satu motif batik tertua di Indonesia. Ciri utama motif batik *parang* adalah garis diagonal yang tersusun jalin-menjalin tidak terputus menyerupai huruf S.

Metode *Convolutional Neural Network* (CNN) digunakan untuk mengekstraksi ciri fitur pada motif batik serta melakukan klasifikasi terhadap 6 kelas batik *parang* yaitu *parang rusak*, *parang barong*, *parang curigo*, *parang kusumo* dan *parang tuding*. Pengolahan citra digital dilakukan pada pembuatan *dataset* untuk menjaga esensi corak pada motif batik parang. Arsitektur jaringan CNN yang digunakan adalah *Mobilenet*. *Prototype* aplikasi klasifikasi yang dibuat pada platform *desktop* untuk data *training* dan *android* untuk klasifikasi. Data citra yang digunakan berjumlah 240 foto batik dibagi menjadi 6 kelas yang digandakan jumlahnya dengan proses *data augmentation* berupa rotasi, *shear*, *zoom* serta *horizontal* dan *vertical flip*.

Hasil penelitian *Convolutional Neural Network* dapat melakukan klasifikasi pada batik motif *parang*. Dari 240 data *testing* pada aplikasi desktop didapat hasil akurasi sebesar 99,58% *prototype* aplikasi android dengan 15 data *testing* didapat hasil akurasi sebesar 73,33%. Pengujian *prototype* dilakukan dengan mengambil foto batik secara langsung menggunakan kamera *smartphone android*. Berdasarkan pengujian tersebut, dapat disimpulkan bahwa sistem telah cukup baik dalam mengklasifikasikan batik motif parang.

Kata Kunci: batik *parang*, *Convolutional Neural Network*, *Mobilenet*, *Android*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan yang Maha Esa yang telah memberikan rahmat dan hidayahNya sehingga penulis dapat menyelesaikan penelitian yang berjudul “CONVOLUTIONAL NEURAL NETWORK KLASIFIKASI BATIK TULIS MOTIF PARANG MENGGUNAKAN MOBILENET”.

Penulisan skripsi ini diajukan untuk memenuhi salah satu syarat kelulusan dalam jenjang perkuliahan Strata 1 Universitas Pembangunan Nasional “Veteran” Yogyakarta. Dalam penulisan skripsi ini tidak lepas dari hambatan dan kesulitan, namun berkat bimbingan, bantuan, nasihat dan saran serta kerjasama dari berbagai pihak, khususnya pembimbing, segala hambatan tersebut akhirnya dapat terselesaikan dengan baik.

Dalam penulisan skripsi ini tentunya tidak lepas dari kekurangan, baik aspek kualitas maupun aspek kuantitas dari materi penelitian yang disajikan. Semua ini didasarkan dari keterbatasan yang dimiliki penulis.

Penulis menyadari bahwa skripsi ini jauh dari sempurna sehingga penulis membutuhkan kritik dan saran yang bersifat membangun untuk penelitian ini. Selanjutnya dalam penulisan skripsi ini penulis banyak diberi bantuan oleh berbagai pihak. Dalam kesempatan ini, penulis dengan tulus hati mengucapkan terimakasih kepada:

1. Hidayatulah Himawan, S.T., M.M., M.Eng. selaku pembimbing I
2. Dr. Herlina Jayadianti, S.T., M.T. selaku pembimbing II
3. Bapak dan Ibu dosen program studi Informatika Fakultas Teknik Industri yang telah memberikan ilmu, pengetahuan, dan pengalaman
4. Berbagai pihak yang telah memberikan bantuan dan dorongan serta berbagi pengalaman pada proses penyusunan skripsi ini.

Terakhir, semoga segala bantuan yang telah diberikan, sebagai amal soleh senantiasa mendapat Ridho Allah SWT. Sehingga pada akhirnya semoga skripsi ini dapat bermanfaat dalam bidang informatika dan dapat dikembangkan menjadi lebih baik lagi.

Yogyakarta, 28 Maret 2021

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
PERNYATAAN BEBAS PLAGIAT	iv
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR.....	v
HALAMAN PERSEMBAHAN	vi
ABSTRAK.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
DAFTAR MODUL PROGRAM.....	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	4
1.6.1. Studi Literatur	4
1.6.2. Pengumpulan Data Secara Langsung.....	4
1.7. Metode Pengembangan Sistem	4
1.8. Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	6
2.1. Batik.....	6
2.2. Motif Batik Parang.....	6
2.2.1. Sejarah Motif Batik <i>Parang</i>	6
2.2.2. Macam-Macam Motif Batik <i>Parang</i>	6
2.3. Citra Digital	9
2.3.1. Definisi Citra Digital.....	9
2.3.2. Representasi Citra Digital	10
2.3.3. Pengolahan Citra Digital.....	10
2.4. <i>Image Processing</i>	10
2.5. <i>Deep Learning</i>	11
2.6. <i>Neural Network</i>	11
2.6.1. <i>Activation Function</i>	12
2.6.1.1. Rectified Linear Unit (ReLU)	12
2.6.1.2. Softmax Layer.....	13
2.6.2. Augmentasi Data.....	13
2.7. <i>Convolutional Neural Network (CNN)</i>	13
2.7.1. <i>Convolution Layer</i>	14
2.7.2. <i>Pooling Layer</i>	15
2.7.3. <i>Training Data</i>	15
2.8. <i>MobileNet</i>	16
2.8.1. <i>Depthwise Separable Convolution</i>	16
2.8.2. Struktur Jaringan <i>MobileNet</i>	17

2.8.3.	<i>Width Multiplier</i>	18
2.8.4.	<i>Resolution Multiplier</i>	18
2.9.	<i>Python</i>	18
2.10.	<i>TensorFlow</i>	18
2.11.	<i>Keras</i>	19
2.12.	<i>Android</i>	19
2.13.	<i>Prototype</i>	19
2.14.	Penelitian Sebelumnya.....	20
BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM.....		23
3.1.	Tahap Pengumpulan dan Pengolahan Data.....	23
3.1.1.	Studi Pustaka.....	24
3.1.2.	Pengumpulan Data	25
3.1.3.	Pengolahan Data	25
3.1.3.1.	Inisialisasi Data Citra Batik Parang	26
3.1.3.2.	<i>Cropping</i> dan <i>Resizing</i> Data Citra Batik.....	27
3.1.3.3.	<i>Data Augmentation</i>	27
3.1.3.4.	<i>Training Data Split</i>	30
3.2.	Tahap Pengembangan Sistem	30
3.2.1.	Perancangan Sistem	30
3.2.2.	Pembuatan Model Batik Tulis Motif Parang	31
3.2.2.1.	Sub-proses Input	32
3.2.2.2.	Sub-proses <i>Training</i>	32
3.2.2.3.	Sub-proses Output.....	38
3.2.3.	Pengujian Model	39
3.2.4.	Implementasi Model	39
3.2.4.1.	Perancangan Proses.....	39
3.2.4.2.	Perancangan Antarmuka	40
BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN.....		42
4.1.	Hasil Penelitian	42
4.1.1.	Aplikasi <i>Desktop</i> untuk Pembentukan Model.....	42
4.1.2.	<i>Prototype</i> Aplikasi <i>Android</i>	47
4.2.	Pengujian Aplikasi	49
4.2.1.	Pengujian Aplikasi <i>Desktop</i> untuk Pembuatan Model.....	49
4.2.2.	Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	51
4.2.3.	Perbandingan Hasil Pengujian Model dan <i>Prototype</i> Aplikasi <i>Android</i>	56
BAB V PENUTUP		57
5.1.	Kesimpulan	57
5.2.	Saran	57
DAFTAR PUSTAKA.....		58

DAFTAR TABEL

Tabel 2.1 Tabel Penelitian Sebelumnya.	21
Tabel 2.2 Tabel Kelebihan dan Kekurangan Penelitian Sebelumnya.	22
Tabel 3.1 Pola Pengenalan Corak Batik Motif Parang.	37
Tabel 3.2 Lanjutan Pola Pengenalan Corak Batik Motif Parang.....	38
Tabel 4.1 Tabel Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	51
Tabel 4.2 Lanjutan tabel Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	52
Tabel 4.3 Lanjutan tabel Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	53
Tabel 4.4 Lanjutan tabel Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	54
Tabel 4.5 Lanjutan tabel Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	55
Tabel 4.6 Lanjutan tabel Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	56
Tabel 4.7 Lanjutan tabel Pengujian <i>Prototype</i> Aplikasi <i>Android</i>	56

DAFTAR GAMBAR

Gambar 2.1. Motif Batik Parang Rusak.	7
Gambar 2.2. Motif Batik Parang Rusak Barong.....	7
Gambar 2.3. Motif Batik <i>Parang Kusumo</i>	8
Gambar 2.4. Motif Batik Parang Klithik.	8
Gambar 2.5. Motif Batik <i>Parang Tuding</i>	9
Gambar 2.6. Motif Batik <i>Parang Curigo</i>	9
Gambar 2.7. Contoh representasi suatu citra $f(x, y)$ dalam 2 dimensi.	10
Gambar 2.8. Contoh sederhana Arsitektur <i>Neural Network</i>	12
Gambar 2.9. Grafik fungsi ReLU.	13
Gambar 2.10. Ilustrasi arsitektur CNN dalam melakukan klasifikasi.	14
Gambar 2.11. Perbedaan antara <i>average pooling</i> dan <i>max pooling</i>	15
Gambar 2.12. Lapisan <i>Depthwise Convolution</i> dan <i>Pointwise Convolution</i>	16
Gambar 2.13. Arsitektur model <i>MobileNet</i>	17
Gambar 2.14. Langkah-langkah <i>Prototype</i> Ogedebe & Jacob (2012).	19
Gambar 3.1. Tahapan Metodologi Penelitian.....	23
Gambar 3.2. Diagram Alir Pengolahan Data.....	25
Gambar 3.3. Visualisasi Foto Batik Motif <i>Parang</i> dari Setiap Kelas.	26
Gambar 3.4. <i>Cropping</i> dan <i>Resizing</i> citra batik.....	27
Gambar 3.5. Diagram alir untuk menghitung rotasi gambar.	28
Gambar 3.6. Perbandingan Citra sebelum rotasi dengan citra batik hasil rotasi.	28
Gambar 3.7. Diagram alir untuk menghitung pergeseran gambar.	29
Gambar 3.8. Perbandingan Citra sebelum <i>shear</i> dengan citra batik hasil <i>shear</i>	30
Gambar 3.9. Arsitektur jaringan <i>MobileNet</i>	31
Gambar 3.10. Diagram Alir Pembuatan Model Batik Tulis Motif Parang.....	32
Gambar 3.11. Diagram Alir Ekstraksi Fitur dengan <i>MobileNet</i>	33
Gambar 3.12. Ilustrasi proses konvolusi	34
Gambar 3.13. Perhitungan Konvolusi.	34
Gambar 3.14. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-1.....	35
Gambar 3.15. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-4.....	35
Gambar 3.16. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-7.....	36
Gambar 3.17. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-10 dan ke-13.....	36
Gambar 3.18. Diagram Alir Pengujian Model Batik Tulis Motif Parang.	39
Gambar 3.19. Output proses pengujian model.	39
Gambar 3.20. Diagram Alir Proses <i>Input</i> dan <i>Output</i>	40
Gambar 3.21. Halaman Pembuatan Model Batik Tulis Motif Parang.....	41
Gambar 3.22. Halaman Pindai Batik.	41
Gambar 4.1. Halaman Aplikasi <i>Desktop</i> untuk pembentukan Model.	42
Gambar 4.2. Tampilan <i>checkDirectory</i> setelah direktori dataset terpilih.....	43
Gambar 4.3. Tampilan dari fungsi tombol <i>Create Dataset</i>	44
Gambar 4.4. Proses Data <i>Training</i>	45
Gambar 4.5. Grafik Hasil proses <i>training</i>	46
Gambar 4.6. Tampilan <i>Confusion Matrix</i> hasil pengujian.	47
Gambar 4.7. Halaman Klasifikasi Batik pada <i>Prototype</i> aplikasi <i>android</i>	48
Gambar 4.8. Diagram Pengujian <i>White-Box</i> pada Aplikasi <i>Desktop</i> Pembuatan Model. ...	50

DAFTAR MODUL PROGRAM

Modul Program 4.1 <i>Source code Select Dataset Directory</i>	43
Modul Program 4.2 Modul Program dari fungsi tombol <i>Create Dataset</i>	44
Modul Program 4.3 Lanjutan Modul Program Proses <i>training</i>	44
Modul Program 4.4 Proses <i>Testing</i>	46
Modul Program 4.5 Proses <i>export_model</i>	47
Modul Program 4.6 Modul <i>Classifier</i>	48
Modul Program 4.7 <i>Method recognizeImage</i>	48
Modul Program 4.8 <i>Method recognizeImage</i>	49

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Batik, secara etimologi, adalah singkatan dua suku kata yang diambil dari kata Bahasa Jawa, yaitu *Amba* yang artinya menulis dan *Titik* yang artinya titik. Dalam KBBI, batik memiliki arti ”*kain bergambar yang pembuatannya secara khusus dengan menuliskan atau menerakan malam pada kain itu, kemudian pengolahannya melalui proses tertentu*”. Batik merupakan kesenian tradisional yang telah diakui UNESCO pada tanggal 2 Oktober 2009 sebagai warisan budaya Indonesia pada kancah Internasional.

Indonesia memiliki motif batik yang sangat beragam, berasal dari berbagai daerah. Setiap motif batik memiliki sejarah dan filosofi tersendiri. Salah satu contohnya adalah motif batik *parang*. Batik *parang* merupakan salah satu motif batik tertua di Indonesia yang sudah ada sejak masa Keraton Mataram Kertasura, yaitu sekitar tahun 1600-an (Koeswadji, 1981). Motif batik *parang* diciptakan oleh pendiri Keraton Mataram Kertasura, yang kemudian pada masa itu menjadi pedoman dalam menentukan derajat kebangsawanan seseorang. Motif batik *parang* diambil dari kata *pereng* yang artinya lereng. Ciri utama dari motif batik *parang* adalah garis diagonal menyerupai huruf S yang terinspirasi dari ombak samudra sebagai lambang semangat yang tidak pernah padam (Koeswadji, 1981). Garis-garis tersebut tersusun jalin-menjalin tidak terputus melambangkan kesinambungan. Motif batik *parang* memiliki banyak jenis, antara lain *parang rusak*, *parang barong*, *parang centung*, *parang kusumo*, *parang klithik*, *parang tuding*, dan *parang curigo*.

Setiap jenis motif batik memiliki ciri khas dan filosofinya masing-masing, namun memiliki banyak kemiripan pula (Samsi, 2011). Sebagai contoh antara motif *parang rusak*, *parang klithik* dan *parang barong*, yang pembedanya hanya terdapat pada ukuran corak *parang*-nya. *Parang barong* memiliki ukuran corak *parang* paling besar yaitu 15cm atau lebih. Corak *parang* pada motif *parang klithik* berukuran paling kecil diantara semua motif *parang*, sedangkan motif *parang rusak* memiliki corak berukuran sedang (Samsi, 2011). Ketiga motif tersebut memiliki makna dan filosofi yang berbeda. Corak berbentuk huruf ‘S’ yang terus menyambung pada motif *parang rusak* menggambarkan tentang hubungan yang terus tersambung. Makna kata ‘*barong*’ dalam Bahasa Jawa adalah singa. Motif *parang barong* hanya boleh digunakan oleh raja pada saat ritual keagamaan dan meditasi. Corak *parang* yang berukuran besar pada motif *parang barong* menggambarkan tentang kebijaksanaan tindakan dalam hal pengendalian diri secara terus menerus (Samsi, 2011). Sedangkan, pada motif *parang klithik* yang memiliki corak *parang* dengan ukuran kecil menggambarkan kesan feminim, kelemah-lembutan dan perilaku yang halus untuk mencerminkan sosok para putri kerajaan.

Corak dalam motif batik *parang* yang kompleks serta kemiripan corak yang terdapat antara satu motif batik *parang* dengan motif batik *parang* yang lainnya akan membingungkan bagi kebanyakan orang dalam membedakan motif batik *parang*, terutama bagi yang tidak paham tentang batik secara mendalam (Samsi, 2011).

Seiring dengan berkembangnya batik, teknologi komputasi juga berkembang dengan pesat. Saat ini teknologi komputasi sudah mulai mulai dikembangkan untuk bisa belajar

dengan sendirinya tanpa arahan dari penggunanya. Teknologi komputer seperti itu disebut *Machine Learning*. Dalam *Machine learning* terdapat implementasi berupa sebuah metode yang spesifik digunakan untuk mengenali dan mempelajari gambar atau data citra, yaitu *Convolutional Neural Network* atau CNN. Metode ini mengadopsi kemampuan otak manusia untuk mengklasifikasikan atau mendeteksi objek dalam bentuk citra digital.

Penggunaan metode CNN dalam mengklasifikasikan data gambar atau citra digital secara otomatis akan melibatkan pengolahan citra digital. Pengolahan citra digital merupakan bidang ilmu yang mempelajari tentang bagaimana suatu citra dibentuk, diolah, serta dianalisis sehingga dapat dikenali oleh manusia (Sutoyo, dkk, 2009). Pengolahan citra batik digital menjadi tantangan tersendiri untuk dilakukan. Keberagaman bentuk corak pada motif batik, warna pada motif batik yang sangat beragam, serta adanya kemiripan bentuk corak batik pada beberapa jenis motif batik (Samsi, 2011). karena kompleksitas dalam motif dan corak yang ada pada batik itulah diperlukan pengolahan citra batik digital untuk mendapatkan ciri khas dari masing-masing motif batik tanpa mengurangi esensi dari motif batik tersebut.

Teknologi komputasi saat ini juga sudah terintegrasi dengan telepon seluler sebagai alat komunikasi, yaitu *smartphone* (Gifary, 2015). *Smartphone* memiliki kemampuan komputasi setingkat dibawah komputer namun dengan fleksibilitas penggunaan yang jauh lebih baik daripada komputer personal (Ismanto, dkk, 2017). Meskipun memiliki kemampuan komputasi setingkat dibawah komputer personal, *smartphone* saat ini lebih banyak digunakan oleh masyarakat hanya sebagai alat komunikasi dan media hiburan (Gifary, 2015). Untuk memaksimalkan kemampuan komputasi *smartphone* selain sebagai alat komunikasi dan media hiburan, maka dibuat sebuah aplikasi pada *smartphone* yang dapat mengenali pola corak pada batik tulis motif parang.

Beberapa penelitian terkait dengan klasifikasi motif batik dan penggunaan metode CNN telah banyak dilakukan sebelumnya. Salah satu penelitian yang membahas tentang klasifikasi batik adalah penelitian yang berjudul Implementasi Metode Klasifikasi *K-Nearest Neighbor* (K-Nn) Untuk Pengenalan Pola Batik Motif Lampung (Naufal, 2017). Motif batik yang digunakan pada penelitian tersebut adalah motif batik khas Lampung, antara lain motif *Jung Agung*, *Siger Kembang Cengkih*, *Siger Ratu Agung* dan *Sembagi* yang berjumlah 100 data citra digital. Fokus dari penelitian ini adalah bagaimana membedakan antara batik khas Lampung atau bukan, dengan mengekstrak fitur (corak) pada motif batik khas Lampung yang kemudian membandingkannya dengan selain batik khas Lampung. Penelitian lainnya dengan judul *Batik Pattern Classification Using Gray Level Co-occurrence Matrix and Modified K-Nearest Neighbor* (Dewangga S, 2018) menggunakan 191 data citra. Motif batik yang digunakan dalam penelitian ini adalah motif *parang*, *kawung* dan *truntum*. Hasil dari penelitian tersebut menunjukkan bahwa metode GLCM dapat mengekstrak fitur corak batik melalui teksturnya dengan baik, namun memiliki kelemahan yaitu komponen warna dari citra diabaikan karena menggunakan input citra berupa citra *grayscale*. Dari hasil tersebut juga dapat diketahui bahwa metode *Modified K-Nearest Neighbor* mendapatkan hasil yang cenderung lebih baik daripada metode *K-Nearest Neighbor*.

Penelitian lain yang membahas tentang klasifikasi batik dan penggunaan metode CNN adalah penelitian dengan judul *The Influence Of Different Preprocessing Techniques*

For Classifying Batik Using Convolutional Neural Networks (Achmad, 2018) menjelaskan tentang perbedaan teknik *preprocessing data* citra motif batik *ceplok*, *kawung* dan *parang*. Teknik yang digunakan antara lain *grayscale conversion*, *canny edge-detection*, dan *histogram equalization*. Hasil dari penelitian menunjukkan bahwa teknik *preprocessing data* citra batik dapat mempengaruhi tingkat akurasi menjadi lebih rendah dibandingkan menggunakan dataset yang mentah, terutama dengan menggunakan *canny edge-detection*. Teknik *grayscale conversion* dan *histogram equalization* memiliki tingkat akurasi yang cenderung sama jika dibandingkan dengan penggunaan *dataset* mentah. Berdasarkan beberapa penelitian yang pernah dilakukan, maka penelitian ini akan dilakukan dengan menggunakan metode *Convolutional Neural Network* (CNN) menggunakan dataset mentah pada *smartphone*.

Pemilihan metode CNN dengan arsitektur *MobileNet* pada penelitian ini dilandasi dengan kemampuan CNN dalam mendeteksi, mengenali serta mengklasifikasikan objek citra digital dengan tingkat akurasi yang baik. CNN merupakan salah satu bagian dari *Deep Learning* yang dapat mempelajari data citra lebih dalam dibandingkan metode *Machine Learning*, dengan tingkat representasi dan abstraksi yang lebih banyak.

Penggunaan *platform android* dapat digunakan untuk melakukan *scanning* objek secara *realtime* serta lebih fleksibel digunakan dimana saja. Selain kedua hal tersebut, penelitian ini akan berfokus pada ekstraksi fitur pada lapisan konvolusi yang terdapat dalam metode CNN sehingga nilai esensi dari masing-masing motif batik dapat dipertahankan sesuai dengan aslinya. Perpaduan antara metode CNN menggunakan arsitektur *MobileNet* dan *platform android* diharapkan dapat melakukan klasifikasi pada aplikasi *desktop* kemudian dilakukan deteksi motif batik *parang* dengan *smartphone* menggunakan hasil klasifikasi pada aplikasi *desktop*.

1.2. Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah dikemukakan, maka dapat rumusan masalah dalam penelitian ini adalah sebagai berikut.

1. Bagaimana cara mengklasifikasikan batik motif *parang* dengan metode CNN menggunakan arsitektur *MobileNet*?
2. Bagaimana cara mengimplementasikan model hasil klasifikasi kedalam aplikasi berbasis *android*?

1.3. Batasan Masalah

Setelah didapatkan rumusan masalah dibuatlah batasan permasalahan agar masalah yang dibahas menjadi lebih terfokus dan jelas. Batasan masalah penelitian ini adalah sebagai berikut :

1. Citra motif batik yang digunakan memiliki dimensi yang sama, atau memiliki ukuran panjang dan lebar yang sama.
2. Batas maksimum ukuran citra digital batik yang digunakan adalah 224 x 224 *pixel*.
3. Motif batik *parang* yang diteliti antara lain *parang rusak*, *parang rusak barong*, *parang curigo*, *parang klithik*, *parang kusumo*, dan *parang tuding*.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah mengklasifikasikan batik motif *parang* serta menganalisa dan menguji tingkat akurasi untuk mengklasifikasikan batik motif *parang* dengan metode CNN menggunakan arsitektur model *Mobilenet* berbasis aplikasi *android*. Penelitian ini akan berfokus pada mempertahankan kualitas hasil ekstraksi fitur citra foto batik tanpa adanya perubahan esensi yang berarti.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah agar dapat mempermudah masyarakat untuk membedakan kain batik bermotif *parang* yang sesuai dengan filosofi dan makna yang diinginkan secara cepat, mudah serta akurat menggunakan *smartphone android*.

Selain itu dapat memberikan informasi kepada akademisi sebagai referensi penelitian mengenai penggunaan metode CNN untuk mengklasifikasikan batik motif *parang*.

1.6. Metodologi Penelitian

1.6.1. Studi Literatur

Studi literatur dilakukan dengan mencari sumber-sumber pustaka yang mendukung penelitian dan memberikan informasi yang memadai dalam menyelesaikan penelitian ini. Studi kepustakaan yang digunakan antara lain: buku, jurnal, artikel dan *paper* yang berkaitan dengan batik motif *parang*.

1.6.2. Pengumpulan Data Secara Langsung

Data yang didapatkan berasal dari Kampung Batik Giriloyo, Kabupaten Bantul. Data dikumpulkan dengan cara mengambil gambar menggunakan kamera lalu dipisahkan dan diberi label berdasarkan jenisnya. Kemudian data yang telah terkumpul dibagi menjadi dua yaitu data latih dan data uji. Data latih sudah diberikan label lalu dimasukkan kedalam kelas berdasarkan motifnya, sedangkan data uji adalah data yang akan diuji cobakan oleh sistem.

1.7. Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah metode *prototype*. Tahapan dalam metode *prototype* adalah :

1. Pengumpulan Kebutuhan
Pengumpulan kebutuhan sistem berupa data yang diperlukan untuk perancangan sistem yang akan dibuat sesuai dengan tujuan.
2. Proses Desain
Proses Desain adalah pembuatan desain secara umum untuk selanjutnya dikembangkan kembali. Setelah desain umum telah dibuat, selanjutnya adalah mengkodekan sistem.
3. Membangun Prototipe
Membangun *prototype* sesuai dengan desain yang telah dibuat, termasuk pengujian dan penyempurnaan.
4. Evaluasi dan Perbaikan
Mengevaluasi, melakukan penyesuaian terhadap hasil pengujian dan memperbaiki

hasil *prototype*.

Pada penelitian ini, pengembangan sistem dengan model *prototype* dilakukan sampai tahap evaluasi terhadap *prototype*.

1.8. Sistematika Penulisan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang pemilihan topik mengenai implementasi metode CNN untuk mengklasifikasikan batik motif *parang*, rumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan mengenai teori-teori dan referensi penelitian yang digunakan untuk mendukung penulisan tugas akhir ini. Pada bagian ini akan dibahas mengenai motif batik *parang*, konsep dasar dari metode CNN sebagai metode dalam klasifikasi motif batik dan *android* sebagai antarmuka, serta konsep dasar pendukung pembangunan sistem lainnya.

BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

Bab ini berisi tentang tahapan metodologi penelitian dan pengembangan sistem yang digunakan dalam perancangan sistem. Bab ini juga menjelaskan tentang proses analisis data dan *design*, logika pembuatan program serta antarmuka pada sistem.

BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN

Bab ini menguraikan tentang implementasi dari analisis dan perancangan sistem menjadi sebuah perangkat lunak serta pengujian dan Analisa terhadap hasil yang diperoleh pada tampilan implementasi sistem.

BAB V PENUTUP

Bab ini memberikan kesimpulan dari analisis yang telah dilakukan dan saran untuk pengembangan yang lebih lanjut terkait dengan penelitian yang dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1. Batik

Batik, secara etimologi, adalah singkatan dua suku kata yang diambil dari kata Bahasa Jawa, yaitu *Amba* yang artinya menulis dan *Titik* yang artinya titik. Arti batik adalah melekatkan lilin malam pada kain sebelum kain diberi warna (Samsi, 2011). Batik merupakan kesenian tradisional yang telah diakui UNESCO pada tanggal 2 Oktober 2009 sebagai warisan budaya Indonesia pada kancah Internasional. Indonesia memiliki motif batik yang sangat beragam, berasal dari berbagai daerah. Setiap motif batik memiliki sejarah dan filosofi tersendiri.

2.2. Motif Batik Parang

Batik motif *parang* memiliki ciri utama yaitu susunan motif menyerupai huruf S yang saling terkait satu dengan yang lainnya, serta saling menjalin sebagai lambang kesinambungan. Susunan motif menyerupai huruf S terinspirasi dari ombak samudra sebagai lambang semangat yang tidak pernah padam. Terdapat pula garis-garis diagonal, memberi gambaran bahwa manusia harus memiliki cita-cita yang luhur, memiliki pendirian yang kokoh dan teguh, serta setia dalam nilai kebenaran. Dinamika dalam pola motif batik parang juga menggambarkan ketangkasan dan kewaspadaan.

2.2.1. Sejarah Motif Batik Parang

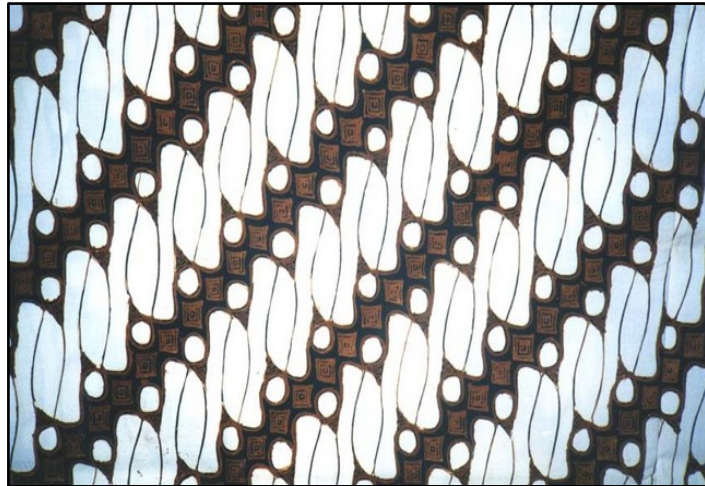
Motif batik *parang* sudah ada sejak masa Keraton Mataram Kertasura sekitar tahun 1600-an, juga menjadi salah satu batik tertua di Indonesia. Motif batik *parang* diambil dari kata *pereng* yang artinya lereng. Motif batik *parang* diciptakan oleh pendiri Keraton Mataram Kertasura, yang kala itu sedang bertapa disuatu tempat di pantai selatan. Kemudian ia memperhatikan ombak yang menerjang karang secara terus menerus tetapi karang tersebut tidak goyah sedikitpun (Koeswadji, 1981). Pada masa Keraton Mataram Kertasura, batik *parang* menjadi pedoman dalam menentukan derajat kebangsawanan seseorang (Sutiyati, 2016).

2.2.2. Macam-Macam Motif Batik Parang

Batik *parang* memiliki motif yang beraneka ragam. Ada lebih dari 50 motif batik *parang* (Samsi, 2011). Berikut ini adalah berapa motif batik *parang* antara lain *parang rusak*, *parang barong*, *parang kusumo*, *parang klithik*, *parang tuding*, dan *parang curigo*.

1. Motif Parang Rusak

Motif ini hanya digunakan oleh para bangsawan pada masa lampau untuk upacara kerajaan (Sutiyati, 2016). *Parang rusak* memiliki arti perang atau menyingkirkan segala hal yang rusak, atau melawan segala macam hal buruk. Motif ini memiliki makna bahwa sebagai manusia harus memiliki watak dan perilaku yang berbudi luhur sehingga dapat mengendalikan segala nafsu dan melawan godaan (Koeswadji, 1981). Pada Gambar 2.1 adalah motif batik *parang rusak* (Samsi, 2011).



Gambar 2.1. Motif Batik Parang Rusak.

2. Motif *Parang Barong*

Batik motif *parang barong* diduga sudah ada sejak masa pemerintahan Sultan Agung dari kerajaan Mataram Islam, tepatnya setelah perpecahan kekuasaan menjadi dua kerajaan (Hasan, 2012). Motif batik ini memiliki ciri khas yaitu ukuran corak *parang* yang paling besar diantara motif batik parang yang lainnya. Ukuran corak tersebut menggambarkan keagungan derajat seseorang. Oleh karena itu, motif batik *parang rusak barong* pada masa kerajaan Mataram Islam menjadi motif batik *larangan*, dilarang digunakan selain raja (Hasan, 2012). Pada Gambar 2.2 adalah motif batik *parang rusak barong* (Samsi, 2011).



Gambar 2.2. Motif Batik Parang Rusak Barong.

3. Motif *Parang Kusumo*

Motif parang kusumo memiliki arti kembang atau bunga. Pada jaman dahulu motif batik parang kusumo hanya boleh digunakan oleh keturunan raja apabila sedang berada di lingkungan keraton. Makna yang terkandung dalam motif batik *parang kusumo* adalah keharuman batin dan perilaku, taat pada norma-norma yang berlaku dan bisa membawa diri agar dapat terhindar dari berbagai bencana. Motif batik *parang kusumo* biasanya digunakan pada acara *lamaran* atau acara tukar cincin. Pada Gambar 2.3 adalah motif batik *parang kusumo* (Samsi, 2011).



Gambar 2.3. Motif Batik *Parang Kusumo*.

4. Motif *Parang Klithik*

Motif batik *parang klithik* merupakan motif batik yang memiliki pola dengan bentuk stilasi yang lebih halus dibanding motif batik *parang rusak*. Corak *parang* dalam motif *parang klithik* memiliki bentuk yang lebih sederhana serta ukuran yang lebih kecil dibanding motif *parang rusak*. Ukuran corak *parang* yang kecil menggambarkan citra feminisme, kelembutan, perilaku yang halus serta kebijaksanaan. Oleh karena itu, pada masa keraton Mataram Islam motif batik *parang klithik* biasanya digunakan oleh putri raja. Pada Gambar 2.4 adalah motif batik *parang klithik* (Samsi, 2011).



Gambar 2.4. Motif Batik *Parang Klithik*.

5. Motif *Parang Tuding*

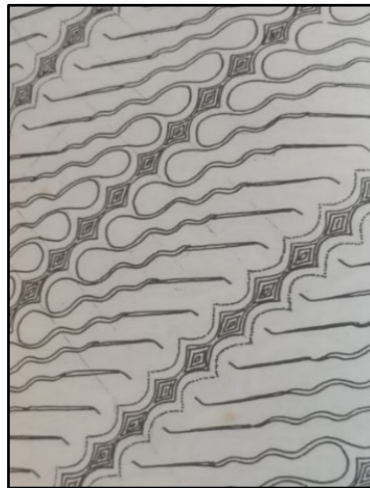
Motif batik *parang tuding* memiliki arti jari telunjuk yang tergambar pada corak *parang* yang menyerupai jari telunjuk yang disusun secara berjajar dan berkesinambungan. Makna kata “*tuding*” yaitu siapapun yang menggunakan batik motif *parang tuding* dapat memberikan petunjuk kepada hal yang menimbulkan kebaikan. Motif batik *parang tuding* biasa digunakan oleh orang tua. Pada Gambar 2.5 adalah motif batik *parang tuding* (Samsi, 2011).



Gambar 2.5. Motif Batik *Parang Tuding*.

6. Motif *Parang Curigo*

Kata “*curigo*” adalah nama lain dari bilah keris yang tergambar pada corak *parang* yang ada dalam motif batik *parang curigo*. Pola hiasan dalam motif batik *parang curigo* disusun sejajar dengan sudut 45° . Selain itu selalu ada ragam hias *mlinjon* berbentuk belah ketupat yang sejajar dengan pola corak *parang*. Makna dari motif batik *parang curigo* adalah kecerdasan, kewibawaan serta ketenangan. Motif batik ini banyak digunakan untuk menghadiri pesta pernikahan. Pada Gambar 2.6 adalah motif batik *parang curigo* (Samsi, 2011).



Gambar 2.6. Motif Batik *Parang Curigo*.

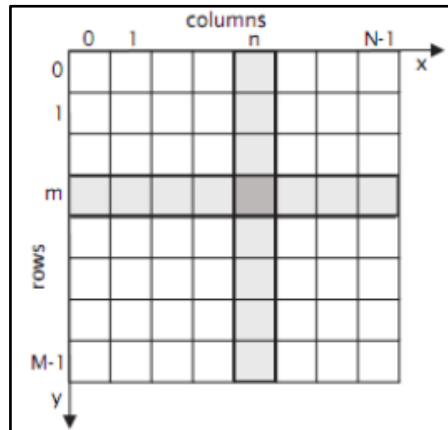
2.3. Citra Digital

2.3.1. Definisi Citra Digital

Citra dapat didefinisikan sebagai fungsi $f(x, y)$ kontinu dengan intensitas cahaya pada bidang dua dimensi dimana nilai x dan y menyatakan koordinat spasial dan elemen matriksnya menyatakan tingkat keabuan pada koordinat tersebut (Sutoyo et al., 2009). Citra digital adalah citra $f(x, y)$ yang setiap elemennya direpresentasikan secara numerik dengan nilai-nilai diskrit (Kusumanto & Tomponu, 2011). Elemen citra digital disebut *pixel* (*picture elements*).

2.3.2. Representasi Citra Digital

Sebuah citra digital dapat diwakili dengan sebuah matriks dua dimensi $f(x, y)$ yang terdiri dari N kolom dan M baris. Setiap nilai *pixel* diwakili oleh dua buah bilangan bulat yang menunjukkan lokasi *pixel* dalam bidang citra seperti digambarkan pada Gambar 2.7 (Kusumanto & Tompunu, 2011). Pada gambar tersebut, koordinat $f(0, 0)$ digunakan untuk pojok kiri atas citra dan koordinat $f(N-1, M-1)$ digunakan untuk pojok kanan bawah dalam citra yang berukuran $N \times M$ *pixel* (Ratnasari, 2018). Contoh representasi citra $f(x, y)$ dalam dua dimensi seperti yang ditunjukkan pada gambar 2.7 (Kusumanto & Tompunu, 2011).



Gambar 2.7. Contoh representasi suatu citra $f(x, y)$ dalam 2 dimensi (Kusumanto & Tompunu, 2011).

2.3.3. Pengolahan Citra Digital

Pengolahan citra digital adalah suatu bidang ilmu yang mempelajari tentang teknik-teknik mengolah citra digital menggunakan komputer (Kusumanto & Tompunu, 2011). Karena pengolahan citra digital dilakukan menggunakan komputer, maka citra terlebih dahulu ditransformasikan kedalam bentuk besaran diskrit berupa matriks dua dimensi yang mencakup tingkat keabuan serta lokasi *pixel*. Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi tidak akan sama persis seperti aslinya yang dipengaruhi oleh beberapa faktor. Faktor-faktor tersebut merupakan penurunan kualitas atau degradasi yang dapat berupa tingkat kontras yang terlalu tinggi atau terlalu rendah, efek distorsi, efek kabur (*Blur*), *noise* atau gangguan yang disebabkan oleh interfensi pengolah data citra maupun pada perangkat elektronik yang digunakan untuk mengambil data citra (Putri, 2016).

Teknik dan proses untuk mengurangi atau menghilangkan efek degradasi pada objek citra meliputi teknik perbaikan atau peningkatan citra (*image enhancement*), restorasi citra (*image restoration*) dan transformasi spesial (*special transformation*). Subjek lain dari pengolahan citra antara lain pengkodean citra, segmentasi citra, representasi dan deskripsi citra (Putri, 2016).

2.4. *Image Processing*

Image processing adalah sebuah transformasi input data yang mentah untuk membantu kemampuan komputasional dan pencari ciri serta mengurangi noise (derau) (Putra, 2008). *Image processing* adalah sebuah teknik untuk menemukan orientasi dari sebuah citra, untuk menghilangkan noise kemudian meningkatkan kualitas dari sebuah citra

sehingga memudahkan dalam proses pengklasifikasian dan penelitian lebih lanjut tentang sebuah citra.

Resizing merupakan salah satu metode yang sering dijumpai ketika citra dengan berbagai ukuran perlu diubah sesuai dengan ukuran yang diinginkan. Secara umum, diperlukan pemetaan dari sumber citra ke citra *resized* tujuan menjadi semulus mungkin (Bradski & Kaehler, 2008).

2.5. Deep Learning

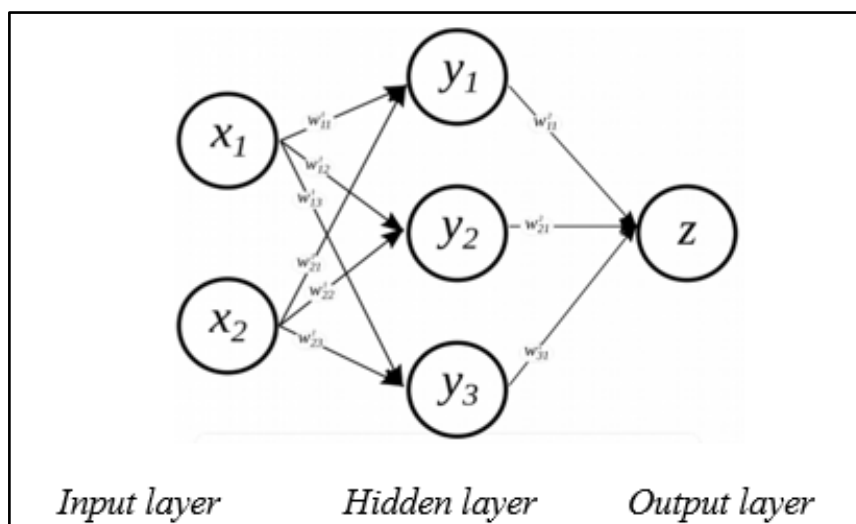
Deep Learning atau pembelajaran mendalam adalah salah satu teknik dari *machine learning* yang memanfaatkan banyak lapisan (*layer*) pengolahan informasi *non-linear* untuk melakukan pengenalan pola, ekstraksi fitur dan klasifikasi (Deng & Yu, 2013).

Deep Learning merupakan pengembangan dari *machine learning* yang telah lebih terarah pada *artificial intelligence* atau kecerdasan buatan. *Machine learning* memiliki keterbatasan kapasitas untuk pengenalan pola objek dan klasifikasi dalam memproses data dengan format aslinya, sehingga kemudian dikembangkan suatu teknik canggih untuk merepresentasikan pembelajaran lebih mendalam (Lecun et al., 2015).

Deep Learning memungkinkan model komputasi yang terdiri dari banyak lapisan untuk mempelajari representasi data dengan beberapa tingkatan abstraksi. Metode ini telah meningkatkan cakupan area penelitian secara signifikan seperti pengenalan suara, deteksi objek atau pengenalan objek visual. *Deep Learning* menemukan pola struktur yang rumit dalam suatu kumpulan data yang sangat besar menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin harus mengubah bobot pada masing-masing parameter didalamnya. Bobot tersebut digunakan untuk menghitung representasi pada setiap lapisan dari representasi pada lapisan sebelumnya (Goodfellow et al., 2016).

2.6. Neural Network

Neural Network adalah suatu sistem pengenalan informasi dengan karakteristik yang sama dengan jaringan syaraf manusia (Fausset, 1994). Seperti halnya pada jaringan syaraf manusia, informasi dalam *Neural Network* diproses pada elemen-elemen sederhana yang disebut *neuron*. Informasi yang berupa sinyal akan berpindah dari satu *neuron* ke *neuron* lainnya melalui sebuah penghubung. Setiap penghubung memiliki bobot yang akan mengubah sinyal yang ditransmisikan. Penentuan bobot tersebut dilakukan berdasarkan metode khusus yaitu pembelajaran atau pelatihan. Nilai dari bobot akan berubah seiring dengan proses pembelajaran atau pelatihan yang dilakukan sehingga sistem akan menghasilkan akurasi yang semakin baik dalam penyelesaian masalah. Setiap *neuron* memiliki fungsi aktivasi yang akan menentukan *output* dari *neuron*.



Gambar 2.8. Contoh sederhana Arsitektur *Neural Network* (Howard et al., 2017).

Pada gambar 2.8 adalah sebuah contoh sederhana arsitektur *Neural Network* yang terdiri dari 3 lapisan (*layer*) antara lain lapisan masukan (*Input Layer*), lapisan tersembunyi (*Hidden Layer*) dan lapisan keluaran (*Output Layer*) (Ratnasari, 2018). Pada gambar 2.8 terlihat bahwa *neuron-neuron* dalam masing-masing lapisan saling terhubung dengan bobot tertentu, misalkan *neuron* [x_1] pada *input layer* terhubung dengan *neuron* [y_1], [y_2] dan [y_3] pada *hidden layer* berbobot sejumlah [w_{11}], [w_{12}] dan [w_{13}]. Dalam mengimplementasikan konsep *neural network*, *hidden layer* dapat dibuat lebih dari satu lapisan yang membuat suatu *neural network* semakin dalam (Ratnasari, 2018).

Neural Network memiliki karakteristik yang membedakan satu dengan lainnya. Yang pertama adalah pola koneksi antar *neuron* atau biasa disebut arsitektur. Pembeda selanjutnya adalah metode yang digunakan dalam menentukan nilai bobot yang biasa disebut proses pembelajaran (*Data Training*). Pembeda terakhir adalah fungsi aktivasi yang digunakan.

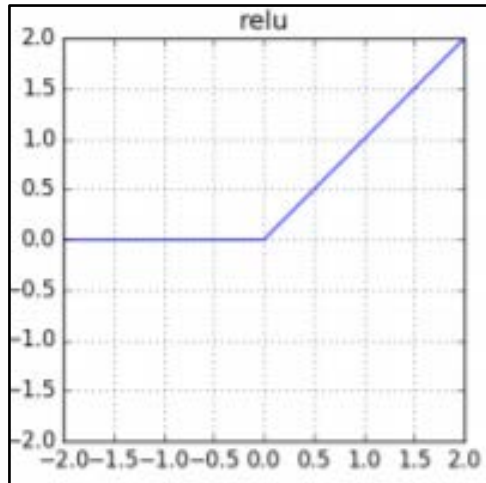
2.6.1. Activation Function

Activation Function atau fungsi aktivasi adalah suatu fungsi yang menunjukkan tingkat keaktifan suatu *neuron* dengan menerapkannya pada sinyal *input* terbobot. Nilai keaktifan hasil fungsi tersebut yang kemudian menjadi *output neuron* tersebut untuk diteruskan ke *neuron* lain pada lapisan selanjutnya atau menjadi penentu *output* sistem.

2.6.1.1. Rectified Linear Unit (ReLU)

Fungsi ReLU merupakan suatu fungsi aktivasi dengan perhitungan yang sederhana. Proses maju (*forward*) dan mundur (*backward*) dalam fungsi ReLU dapat dibuat hanya menggunakan kondisi *if*. Jika elemen x bernilai negatif, maka nilai akan diatur menjadi 0 dan tidak akan ada operasi eksponensial, perkalian atau pembagian.

Dengan karakteristik seperti yang sudah dijelaskan sebelumnya, fungsi ReLU memiliki kelebihan dalam hal waktu *training* dan *testing* data yang secara signifikan lebih cepat ketika berhadapan dengan jaringan yang memiliki *neuron* banyak.



Gambar 2.9. Grafik fungsi ReLU (Goodfellow et al., 2016).

Seperti yang dapat dilihat pada Gambar 2.9 (Goodfellow et al., 2016), ReLU setengah diperbaiki (dari bawah). Ini adalah $f(z)$ adalah nol bila z kurang dari nol dan $f(z)$ sama dengan z bila z di atas atau sama dengan nol. Rentang: (0 sampai tak terbatas) Fungsi dan turunannya keduanya bersifat monoton. Tapi masalahnya adalah semua nilai negatif menjadi nol segera yang menurunkan kemampuan model agar sesuai atau melatih dari data dengan benar. Itu berarti setiap masukan negatif yang diberikan pada fungsi aktivasi ReLU mengubah nilainya menjadi nol segera dalam grafik, yang pada gilirannya mempengaruhi grafik yang dihasilkan dengan tidak memetakan nilai negatif secara tepat (Sharma, 2017).

2.6.1.2. Softmax Layer

Softmax layer merupakan lapisan yang biasanya menjadi lapisan terakhir dari sebuah *neural network* untuk mendapatkan output. Secara umum, *softmax layer* menerima input kemudian melakukan pembobotan dan penambahan *bias* dengan menerapkan fungsi *softmax*. Sehingga output pada *softmax layer* dapat dikatakan sebagai distribusi probabilitas.

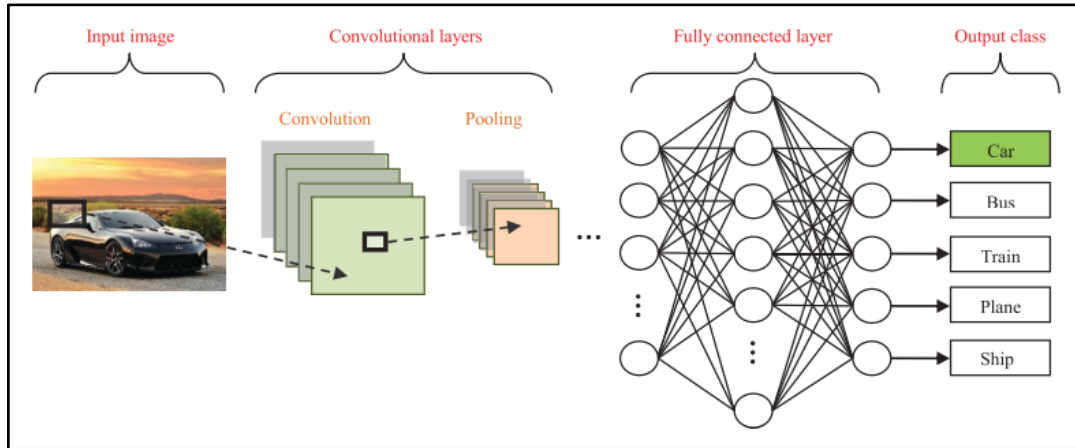
2.6.2. Augmentasi Data

Augmentasi data adalah suatu proses dalam pengolahan data gambar, augmentasi merupakan proses mengubah atau memodifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun manusia masih dapat mengetahui bahwa gambar yang diubah tersebut adalah gambar yang sama. Augmentasi dapat meningkatkan akurasi dari model CNN yang dilatih karena dengan augmentasi model mendapatkan data-data tambahan yang dapat berguna untuk membuat model yang dapat melakukan generalisasi dengan lebih baik. Augmentasi yang dilakukan pada penelitian ini adalah membalikan gambar secara horizontal, melakukan pergeseran atau *shear* secara acak, dengan maksimal zoom sebesar 0.8 dari posisi awal gambar, dan juga melakukan rotasi gambar secara acak dengan derajat maksimal 90° .

2.7. Convolutional Neural Network (CNN)

CNN adalah jaringan syaraf yang *feedforward*, artinya alur informasi yang diolah hanya terjadi dalam satu arah saja, dari *input* sampai *output* (Rawat & Wang, 2017). Arsitektur CNN sama seperti *Artificial Neural Network* (ANN) yang terinspirasi dari

jaringan syaraf manusia, terdiri dari lapisan sederhana dan kompleks yang saling bergantian seperti pada lapisan pada otak manusia. Secara umum, CNN terdiri dari lapisan konvolusi (*Convolution Layer*) dan lapisan penggabungan (*Pooling Layer*). Pada Gambar 2.10 menunjukkan ilustrasi arsitektur CNN dalam melakukan klasifikasi (Rawat & Wang, 2017).



Gambar 2.10. Ilustrasi arsitektur CNN dalam melakukan klasifikasi (Rawat & Wang, 2017).

Pada gambar 2.10 menunjukkan bahwa sebuah gambar dimasukkan kedalam jaringan CNN secara langsung, kemudian dilanjutkan dengan beberapa tahap *convolution* dan *pooling*. Kemudian hasil representasi dari operasi sebelumnya akan diproses dalam *fully connected layer*. Setelah diproses dalam *fully connected layer*, lalu sinyal hasil representasi dari proses *fully connected layer* akan mengarahkan kepada output (Rawat & Wang, 2017).

CNN hingga saat ini telah banyak digunakan sebagai metode klasifikasi gambar, terbukti dengan banyaknya penelitian yang menggunakan CNN sebagai metode klasifikasi. Dalam beberapa tahun terakhir telah banyak pengembangan arsitektur CNN dengan tujuan untuk meningkatkan akurasi dalam klasifikasi gambar atau untuk mengurangi biaya pembuatan sistem. Beberapa contoh pengembangan arsitektur CNN adalah *MobileNet* untuk platform android, *Recurrent Convolutional Neural Network (R-CNN)*, dan lain sebagainya.

2.7.1. Convolution Layer

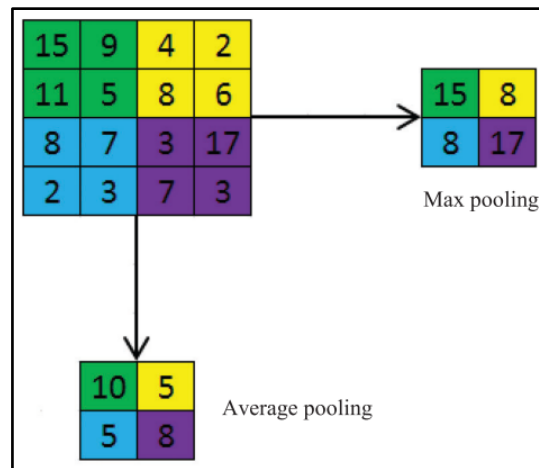
Lapisan konvolusi atau *convolution layer* dalam CNN bertindak sebagai lapisan yang mengekstrak fitur, sekaligus mempelajari representasi fitur dari gambar yang di-*input*. *Neuron* dalam lapisan konvolusi tersusun membentuk *feature map*. Setiap *neuron* dalam sebuah *feature map* memiliki sebuah penerima sinyal, dimana penerima sinyal tersebut terhubung dengan sekumpulan *neuron* dari lapisan sebelumnya yang telah memiliki bobot untuk melakukan *training* dan dalam beberapa kasus *neuron* tersebut ditunjuk sebagai patokan (Lecun et al., 2015).

Input tergabung dengan bobot yang telah di-*training* untuk melakukan *feature map* baru, dan hasil penggabungan tersebut dikirim melalui suatu fungsi aktivasi *non-linear*. Semua *neuron* dalam *feature map* akan memiliki bobot yang disamakan. Akan tetapi, *feature map* yang berbeda dalam lapisan konvolusi yang sama akan memiliki bobot yang berbeda pula, sehingga beberapa fitur dapat diekstrak dilokasi masing-masing *feature map* (Lecun et al., 2015). Fungsi aktivasi *non-linear* memungkinkan ekstraksi fitur *non-linear*. Fungsi aktivasi *non-linear* yang paling sering digunakan adalah fungsi *sigmoid*, *tanh* dan ReLU.

2.7.2. Pooling Layer

Tujuan dari *pooling layer* adalah untuk mengurangi resolusi spasial dari *feature map* sehingga terbentuk resolusi spasial yang tidak beragam pada distorsi *input* dan translasi (Lecun et al., 2015). Pada awalnya penggunaan *average pooling layer* untuk mencari rata-rata persebaran sinyal yang berisi nilai *input* dalam sekumpulan kecil gambar pada lapisan selanjutnya adalah hal yang umum (Lecun et al., 2015). Namun, dalam model yang lebih baru, penggunaan *max pooling layer* dalam menghitung persebaran sinyal berisi nilai maksimal dengan *receptive field* pada lapisan selanjutnya.

Pada *max pooling* $Y_{ki j}$ akan memilih elemen terbesar dari setiap *receptive field*, dimana *output* dari operasi *pooling* terasosiasi dengan *feature map* ke- k . Elemen pada titik (p, q) diwakili dengan $x_{k_{pq}}$, terkandung didalam daerah *pooling* \mathcal{R}_{ij} , dimana terbentuk suatu *receptive field* disekitar titik (i, j) (Deng & Yu, 2013). Pada gambar 2.11 menunjukkan perbedaan antara *max pooling* dan *average pooling* (Rawat & Wang, 2017).



Gambar 2.11. Perbedaan antara *average pooling* dan *max pooling* (Rawat & Wang, 2017).

Pada Gambar 2.11 dapat dilihat bahwa *input* yang diberikan adalah gambar dengan ukuran 4 x 4, dengan menerapkan filter dan *stride* berukuran 2 x 2. Hasil *average pooling* yang ditunjukkan pada gambar adalah nilai rata-rata yang dibulatkan dari setiap daerah sub sampel, sedangkan hasil *max pooling* adalah nilai maksimal dari setiap daerah sub sampel.

2.7.3. Training Data

CNN dan ANN secara umum menggunakan algoritma pembelajaran untuk menentukan parameter lepasnya; yaitu bias, bobot dan lain sebagainya, untuk mencapai hasil yang diinginkan. Algoritma pembelajaran yang paling umum digunakan adalah *backpropagation* (Deng & Yu, 2013; Lecun et al., 2015). *Backpropagation* memperhitungkan gradien yang terdapat dalam fungsi objektif untuk menentukan bagaimana cara mengatur parameter dari jaringan tersebut supaya meminimalisir error yang berpengaruh pada performansi sistem (Rawat & Wang, 2017). Permasalahan yang biasa muncul dalam pelatihan data CNN adalah *overfitting*, yang memiliki performa buruk ketika mencoba sistem dengan *test set* setelah melakukan pelatihan terhadap *dataset* yang kecil atau bahkan *dataset* yang besar. Permasalahan *overfitting* dapat mempengaruhi kemampuan model CNN dalam menyamaratakan data yang tidak terlihat (Rawat & Wang, 2017).

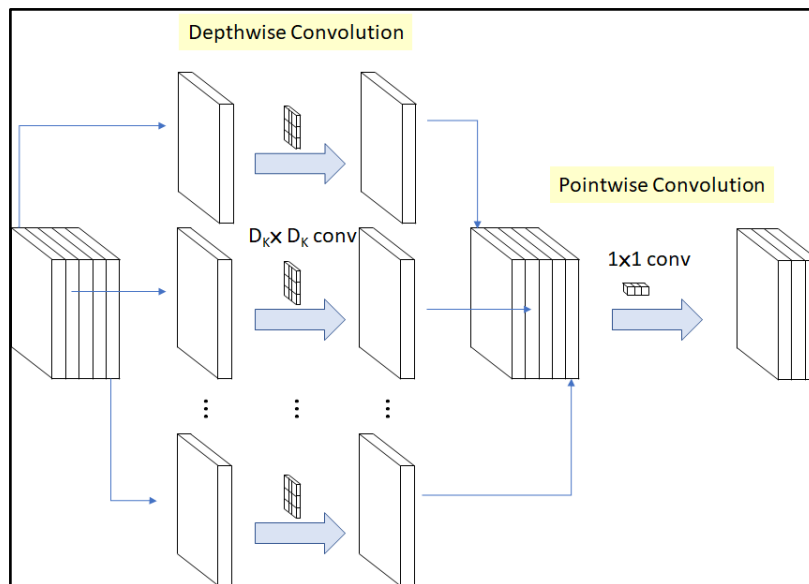
2.8. MobileNet

MobileNet adalah salah satu pengembangan metode CNN yang arsitektur jaringannya lebih efisien untuk membuat model yang dengan mudah disesuaikan dengan kebutuhan desain dari aplikasi *mobile* dan *embedded vision* (Howard et al., 2017). Jika dibandingkan model CNN standar, model *MobileNet* memiliki jumlah operasi dan ukuran memori yang secara signifikan lebih kecil namun dengan tingkat akurasi yang mendekati model CNN standar.

MobileNet berfokus kepada bagaimana cara mengoptimasi latensi dari suatu jaringan serta tetap berukuran kecil. Komponen paling utama untuk membangun *MobileNet* adalah lapisan *depthwise separable convolution*. Struktur jaringan *MobileNet* terdiri dari dua buah model penyusut yang digunakan sebagai *hyper-parameter* dalam jaringan, yaitu *width multiplier* dan *resolution multiplier*.

2.8.1. Depthwise Separable Convolution

Depthwise separable convolution merupakan kunci utama dalam membangun suatu jaringan syaraf yang efisien. Pada prinsipnya, lapisan ini dibuat untuk menggantikan lapisan konvolusi standar dengan memisahkannya menjadi 2 lapisan yang terpisah. Lapisan-lapisan tersebut adalah *depthwise convolution* yang didalamnya dilakukan *filtering* dengan mengaplikasikan filter konvolusi pada setiap kanal input, serta lapisan berupa matriks konvolusi berukuran 1×1 yang disebut *pointwise convolution* untuk menggabungkan fitur-fitur baru melalui kombinasi komputasi linear dari kanal input (Howard et al., 2017). Lapisan *depthwise convolution* dan *pointwise convolution* dapat dilihat pada gambar 2.12.



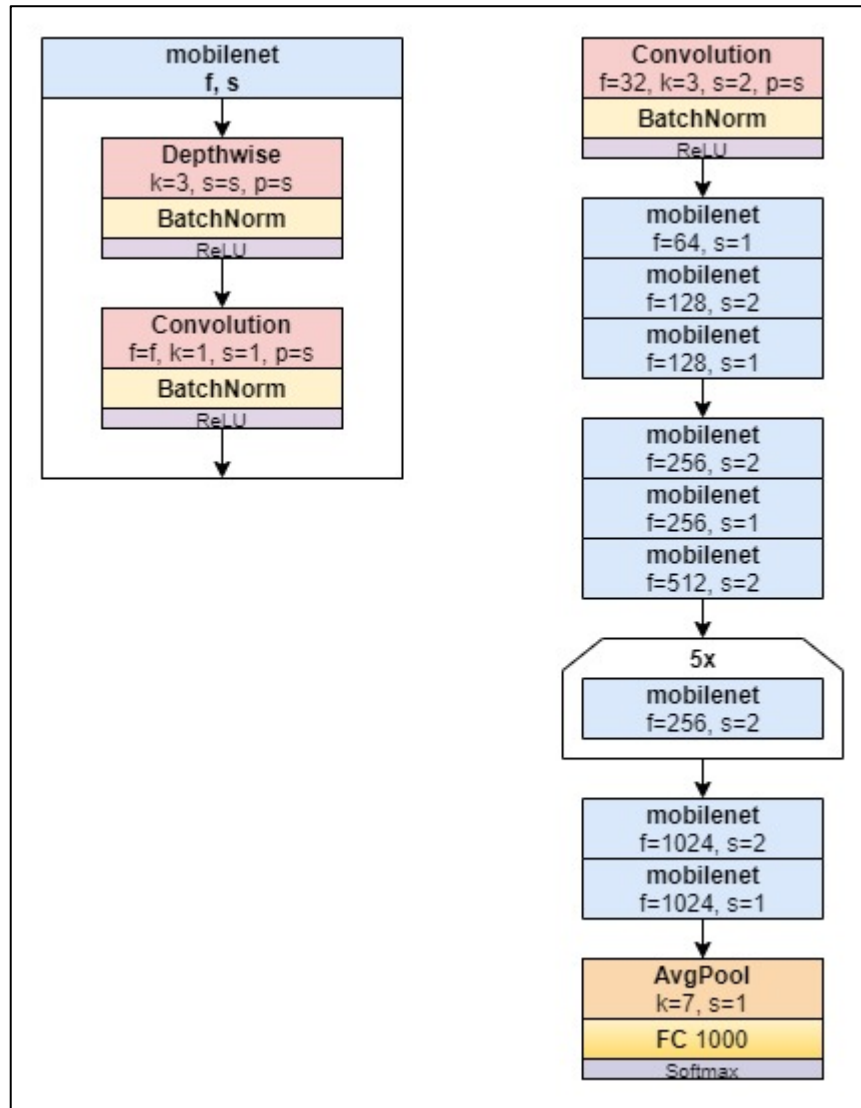
Gambar 2.12. Lapisan *Depthwise Convolution* dan *Pointwise Convolution* (Howard et al., 2017).

Suatu lapisan *depthwise separable convolution* membutuhkan biaya komputasi $D_K \times D_K \times M \times D_F \times D_F$ untuk *depthwise convolution* yang kemudian ditambahkan dengan $M \times N \times D_F \times D_F$ untuk *pointwise convolution* sehingga menghasilkan suatu output $D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$ dimana M adalah jumlah kanal input, N adalah jumlah kanal output, D_K adalah ukuran kernel dan D_F adalah ukuran *feature map*.

Jika dibandingkan dengan lapisan konvolusi standar, lapisan *depthwise separable convolution* secara efektif mengurangi biaya komputasi 8-9 kali lebih rendah dengan hanya penurunan akurasi yang tidak signifikan (Howard et al., 2017).

2.8.2. Struktur Jaringan *MobileNet*

Struktur jaringan *MobileNet* tersusun dari lapisan konvolusi *depthwise separable* terkecuali pada lapisan pertama yang merupakan lapisan konvolusi sepenuhnya. Pada Gambar 2.13 merupakan struktur jaringan *MobileNet*.



Gambar 2.13. Arsitektur model *MobileNet* (Howard et al., 2017).

Pada gambar diatas, dapat dilihat bahwa pada setiap lapisan dalam jaringan *MobileNet* diikuti dengan *batch normalization* serta fungsi aktivasi ReLU non-linier terkecuali pada lapisan *fully connected* yang terhubung dengan lapisan *softmax* untuk klasifikasi. Terdapat lapisan *average pooling* untuk mereduksi nilai resolusi spasial sebelum lapisan *fully connected*. Jika *depthwise* dan *pointwise* dihitung secara terpisah, jaringan *MobileNet* memiliki 29 lapisan konvolusi.

2.8.3. *Width Multiplier*

Width multiplier (α) merupakan bagian untuk mengontrol lebar input dari suatu lapisan dalam model *MobileNet*, yang mengubah jumlah kanal input M menjadi αM . Dengan *width multiplier*, biaya komputasi dari lapisan *depthwise separable convolution* menjadi $D_K \times D_K \times \alpha M \times D_F \times D_F + \alpha M \times \alpha N \times D_F \times D_F$ dimana nilai α yaitu antara 0 sampai 1, dengan pengaturan khas 1.0, 0.75, 0.60 dan 0.25. Pengaturan dasar dari *MobileNet* yaitu nilai $\alpha = 1.0$. Semakin kecil nilai α , akan berpengaruh pada biaya komputasi dan jumlah parameter yang dapat berkurang secara kuadrat. Nilai akurasi juga akan berkurang dari nilai $\alpha = 1.0$ sampai yang paling kecil pada nilai $\alpha = 0.25$.

2.8.4. *Resolution Multiplier*

Resolution multiplier (ρ) merupakan bagian untuk mengontrol resolusi dari gambar yang menjadi input. Dengan menambahkan *resolution multiplier* (ρ) kedalam jaringan, maka biaya komputasi berubah menjadi $D_K \times D_K \times \alpha M \times \rho D_F \times \rho D_F + \alpha M \times \alpha N \times \rho D_F \times \rho D_F$ dimana nilai ρ yaitu antara 0 sampai 1, dengan dengan resolusi input sebesar 224, 192, 160 dan 128. Pengaturan dasar dari *MobileNet* yaitu nilai $\rho = 1.0$. Semakin kecil resolusi input, maka akan berpengaruh pada nilai akurasi yang berkurang dari resolusi input 224 sampai yang paling kecil pada resolusi input 128.

2.9. *Python*

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* juga didukung oleh komunitas yang besar.

Python mendukung pemrograman multi paradigma, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada *python* adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, *python* umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai *platform* sistem operasi.

2.10. *TensorFlow*

TensorFlow adalah perpustakaan perangkat lunak, yang dikembangkan oleh Tim *Google Brain* dalam organisasi penelitian Mesin Cerdas *Google*, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam (Agarwal et al., 2015). *TensorFlow* kemudian menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan. Berdasarkan penelitian yang dilakukan oleh Agarwal, dkk pada tahun 2015, fitur utama *TensorFlow* meliputi :

1. Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan *array* multi dimensi (*tensor*).
2. Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin
3. Penggunaan GPU yang transparan, mengotomatisasi manajemen dan optimalisasi memori dan data yang digunakan. *Tensorflow* bisa menulis kode yang sama dan menjalankannya baik di CPU atau GPU. *TensorFlow* dapat mengetahui bagian perhitungan mana yang harus dipindahkan ke GPU.
4. Skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar.

2.11. *Keras*

Keras adalah perpustakaan berbasis sumber terbuka yang dirancang untuk menyederhanakan model dari kerangka Deep Learning. *Keras* ditulis dengan bahasa pemrograman *Python*, dapat dijalankan di atas *framework* (kerangka kerja) kecerdasan buatan seperti TensorFlow, Microsoft Cognitive Toolkit, dan Theano.

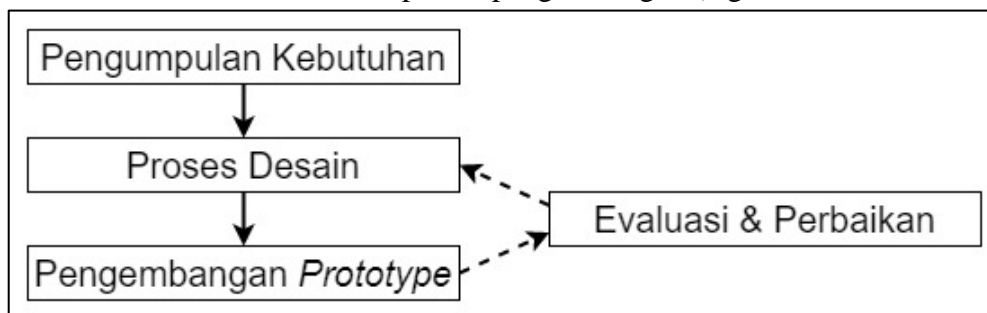
Saat ini, *Keras* dianggap sebagai salah satu perpustakaan pembelajaran mesin terbaik di Python. *Keras* juga menyediakan beberapa utilitas terbaik dalam hal menyusun model, memproses dataset, memvisualisasikan grafik, dan hal lainnya.

2.12. *Android*

Android adalah sebuah sistem operasi berbasis *linux* untuk perangkat mobile yang mencakup sistem operasi, *middleware* dan aplikasi (Murwiyati & Lauren, 2013). *Android* menyediakan *platform* yang terbuka, sehingga siapapun bebas untuk menciptakan mengembangkan aplikasi dengan basis *android*. Pada awalnya, perusahaan teknologi *Google* membeli perusahaan *Android* yang pada masa itu merupakan pendatang baru dalam dunia pengembangan perangkat lunak berbasis *mobile*. Kemudian *android* dikembangkan oleh *Google* dengan membentuk *Open Handset Alliance*, korosium dari 34 perusahaan perangkat keras, perusahaan perangkat lunak dan perusahaan telekomunikasi.

2.13. *Prototype*

Prototype adalah suatu metode pengembangan perangkat lunak yang menggunakan model berupa model fisik dari kerja sistem dan berfungsi sebagai versi awal dari sistem (Ogedebe & Jacob, 2012). *Prototype* mempermudah analisis dan desain sistem dengan menggunakan dialog yang lebih mudah dipahami. Semakin banyak interaksi antara pengembang dan pengguna, maka proses pengembangan sistem menjadi lebih cepat dan pengguna akan lebih interaktif dalam proses pengembangan (Ogedebe & Jacob, 2012).



Gambar 2.14. Langkah-langkah *Prototype* Ogedebe & Jacob (2012).

Berikut adalah langkah-langkah dalam metode *prototype* :

1. Pengumpulan Kebutuhan. Pengumpulan kebutuhan sistem mencakup pertemuan antara pengembang dan pengguna untuk menentukan perancangan sistem secara keseluruhan yang akan dibuat agar sesuai dengan tujuan serta mengidentifikasi kebutuhan secara umum yang dibutuhkan untuk pengembangan sistem.
2. Proses Desain adalah pembuatan desain secara umum untuk selanjutnya dikembangkan kembali. Proses desain berfokus pada representasi dari aspek-aspek yang berorientasi pada pengguna, yaitu input, proses dan format output.
3. Membangun *Prototype*. Pembentukan *prototype* sesuai dengan desain yang telah dibuat, termasuk pengujian dan penyempurnaan.
4. Evaluasi dan Perbaikan. Mengevaluasi, melakukan penyesuaian terhadap hasil pengujian dan memperhalus hasil *prototype*.

2.14. Penelitian Sebelumnya

Penyusunan tugas akhir ini mengambil beberapa referensi dari penelitian sebelumnya termasuk jurnal-jurnal penelitian yang terkait dengan topik penelitian ini.

Penelitian yang dilakukan oleh Naufal pada tahun 2017 berjudul Implementasi Metode Klasifikasi *K-Nearest Neighbor* (K-Nn) Untuk Pengenalan Pola Batik Motif Lampung membahas tentang cara membedakan antara batik khas Lampung atau bukan, dengan mengekstrak fitur pada motif batik khas Lampung yang kemudian membandingkannya dengan selain batik khas Lampung. Motif batik yang digunakan pada penelitian tersebut adalah motif batik khas Lampung, antara lain motif *Jung Agung*, *Siger Kembang Cengkih*, *Siger Ratu Agung* dan *Sembagi* yang berjumlah 100 data citra digital.

Penelitian lain dengan objek batik berjudul *Batik Pattern Classification Using Gray Level Co-occurrence Matrix and Modified K-Nearest Neighbor* yang dilakukan oleh Dewangga S. tahun 2018 membahas tentang metode GLCM untuk fitur corak batik melalui teksturnya menggunakan 191 data citra batik motif *parang*, *kawung* dan *truntum*. Hasil penelitian ini menunjukkan bahwa metode *Modified K-Nearest Neighbor* mendapat hasil yang cenderung lebih baik daripada metode *K-Nearest Neighbor* untuk klasifikasi motif batik.

Penelitian berjudul *The Influence Of Different Preprocessing Techniques For Classifying Batik Using Convolutional Neural Networks* yang dilakukan oleh Achmad pada tahun 2018 membahas tentang perbedaan teknik *preprocessing* atau pengolahan citra motif batik *ceplok*, *kawung* dan *parang* dengan metode *Convolutional Neural Networks*. Menggunakan teknik *grayscale conversion*, *canny edge-detection*, dan *histogram equalization*. Hasil penelitian menunjukkan bahwa teknik pengolahan citra batik menggunakan *canny edge-detection* dapat mempengaruhi tingkat akurasi menjadi lebih rendah. Sedangkan teknik *grayscale conversion* dan *histogram equalization* memiliki tingkat akurasi yang cenderung sama jika dibandingkan dengan penggunaan *dataset* mentah.

Penelitian lain dengan metode klasifikasi yang sama berjudul *Deep Learning Convolutional Neural Network Untuk Klasifikasi Pengenalan Objek Menggunakan Mxnet* oleh Ratnasari pada tahun 2018 membahas tentang performansi metode *Deep Learning* yaitu *Convolutional Neural Network* dalam melakukan klasifikasi pengenalan motif batik keraton dan motif batik pesisir. Citra batik diubah menjadi *grayscale* sebelum melakukan klasifikasi.

Hasil penelitian menunjukkan bahwa model CNN dengan dua lapisan konvolusi dan dua *fully connected layer* dengan 300 *epoch* menghasilkan nilai akurasi sebesar 52.5%.

Tabel 2.1 Tabel Penelitian Sebelumnya.

No	Penelitian	Hasil Penelitian
1	Judul : Klasifikasi Batik Menggunakan Metode <i>K-Nearest Neighbour</i> Berdasarkan <i>Gray Level Co-Occurrence Matrices</i> (GLCM) (Wijayanto, 2015)	Penelitian ini menggunakan metode GLCM untuk ekstraksi fitur dan metode <i>K-Nearest Neighbour</i> untuk mengklasifikasi batik. Menggunakan 5 kelas batik, yaitu batik Cirebon, Jakarta, Pekalongan, Solo dan Yogyakarta dengan jumlah keseluruhan 100 data. Metode evaluasi yang digunakan adalah <i>confusion matrix</i> untuk mengukur tingkat akurasi dalam proses klasifikasi. Hasil klasifikasi dengan nilai tertinggi pada sudut 0° dan nilai akurasi terendah pada sudut 90°.
2	Judul : Pengenalan Motif Batik Indonesia Menggunakan Deteksi Tepi <i>Canny</i> Dan <i>Template Matching</i> (Flaurensia, dkk, 2016)	Penelitian ini menggunakan metode deteksi tepi <i>canny</i> pengolahan citra motif batik serta metode <i>template matching</i> untuk pengenalan pola motif batik. Terdapat 9 motif batik yang digunakan sebagai pengujian. Hasil dari penelitian ini menunjukkan bahwa persentase kemiripan dalam pengujian mengenali motif batik sebesar 89,44%.
3	Judul : Implementasi Metode Klasifikasi <i>K-Nearest Neighbor</i> (K-Nn) Untuk Pengenalan Pola Batik Motif Lampung (Naufal, 2017)	Penelitian ini berfokus pada bagaimana membedakan antara batik khas Lampung atau bukan, dengan mengekstrak fitur pada motif batik khas Lampung yang kemudian membandingkannya dengan selain batik khas Lampung. Motif batik yang digunakan pada penelitian tersebut adalah motif batik khas Lampung, yaitu motif <i>Jung Agung</i> , <i>Siger Kembang Cengkih</i> , <i>Siger Ratu Agung</i> dan <i>Sembagi</i> yang berjumlah 100 data citra digital.
4	Judul : Batik <i>Pattern Classification Using Gray Level Co-occurrence Matrix and Modified K-Nearest Neighbor</i> (Dewangga S, 2018)	Penelitian ini menggunakan 191 data citra batik motif <i>parang</i> , <i>kawung</i> dan <i>truntum</i> . Hasil penelitian menunjukkan bahwa metode GLCM dapat mengekstrak fitur corak batik melalui teksturnya dengan baik. Dari hasil tersebut diketahui bahwa metode <i>Modified K-Nearest Neighbor</i> mendapatkan hasil yang cenderung lebih baik daripada metode <i>K-Nearest Neighbor</i> untuk klasifikasi motif batik.
5	Judul : <i>The Influence Of Different Preprocessing Techniques For Classifying Batik Using Convolutional Neural Networks</i> (Achmad, 2018)	Penelitian ini bertujuan untuk mengetahui perbedaan teknik <i>preprocessing</i> atau pengolahan citra motif batik <i>ceplok</i> , <i>kawung</i> dan <i>parang</i> dengan metode <i>Convolutional Neural Networks</i> . Menggunakan teknik <i>grayscale conversion</i> , <i>canny edge-detection</i> , dan <i>histogram equalization</i> . Hasil penelitian menunjukan bahwa teknik pengolahan citra batik dapat mempengaruhi tingkat akurasi menjadi lebih rendah dibandingkan menggunakan dataset yang mentah, jika menggunakan <i>canny edge-detection</i> . Teknik <i>grayscale conversion</i> dan <i>histogram equalization</i> memiliki tingkat akurasi yang cenderung sama jika dibandingkan dengan penggunaan <i>dataset</i> mentah.
6	Judul : <i>Deep Learning Convolutional Neural Network Untuk Klasifikasi Pengenalan Objek Menggunakan Mxnet</i> (Ratnasari, 2018)	Penelitian ini bertujuan untuk mengetahui performansi metode <i>Deep Learning</i> yaitu <i>Convolutional Neural Network</i> dalam melakukan klasifikasi pengenalan motif batik keraton dan motif batik pesisir. Citra batik diubah menjadi <i>grayscale</i> sebelum melakukan klasifikasi. Hasil penelitian menunjukkan bahwa model CNN dengan dua lapisan konvolusi dan dua <i>fully connected layer</i> dengan 300 <i>epoch</i> menghasilkan nilai akurasi sebesar 52.5%.

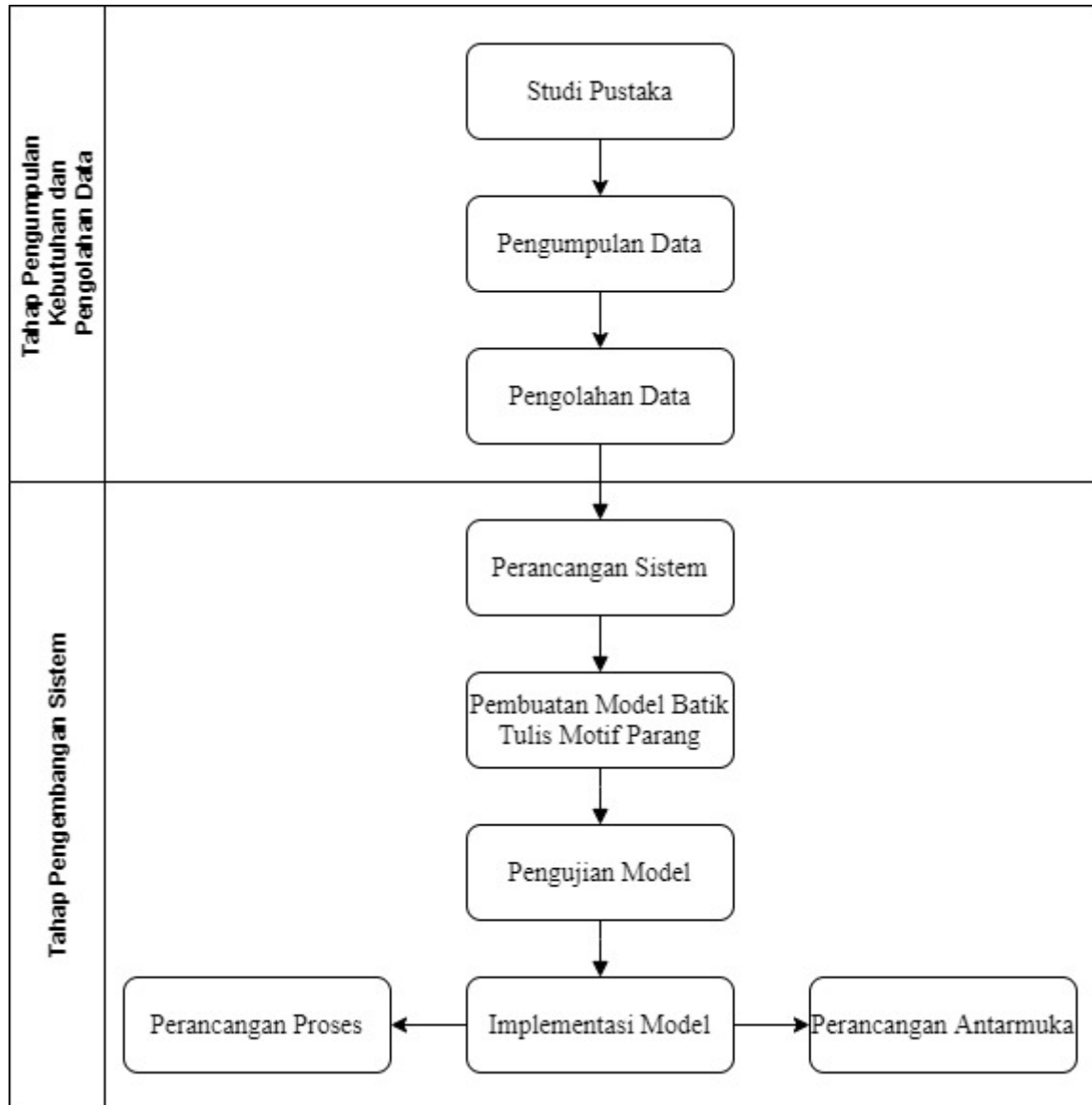
Tabel 2.2 Tabel Kelebihan dan Kekurangan Penelitian Sebelumnya.

No	Penelitian	Kelebihan dan Kekurangan
1	Judul : Klasifikasi Batik Menggunakan Metode <i>K-Nearest Neighbour</i> Berdasarkan <i>Gray Level Co-Occurrence Matrices</i> (GLCM) (Wijayanto, 2015)	(+) Cara pengujian detail. (-) Hasil klasifikasi memiliki tingkat akurasi yang cenderung rendah.
2	Judul : Pengenalan Motif Batik Indonesia Menggunakan Deteksi Tepi Canny Dan Template Matching (Flaurensia, dkk, 2016)	(+) Pengujian mencakup banyak motif batik dan mendapatkan hasil pengujian yang baik dengan rata-rata nilai akurasi sebesar 89,44%.
3	Judul : Implementasi Metode Klasifikasi <i>K-Nearest Neighbor</i> (K-Nn) Untuk Pengenalan Pola Batik Motif Lampung (Naufal, 2017)	(+) Tingkat akurasi tertinggi mencapai 98,182% pada nilai $k=17$ dan sudut 45° . (-) Metode pengujian data batik masih harus input data kedalam aplikasi dalam komputer kemudian baru dapat dibaca oleh program.
4	Judul : <i>Batik Pattern Classification Using Gray Level Co-occurrence Matrix and Modified K-Nearest Neighbor</i> (Dewangga S, 2018)	(+) Hasil klasifikasi memiliki rata-rata tingkat akurasi yang cukup baik, yaitu 75,44%.
5	Judul : <i>The Influence Of Different Preprocessing Techniques For Classifying Batik Using Convolutional Neural Networks</i> (Achmad, 2018)	(+) Hasil penelitian dapat digunakan sebagai acuan untuk penelitian dengan topik deteksi batik, dan pengolahan citra batik.
6	Judul : <i>Deep Learning Convolutional Neural Network Untuk Klasifikasi Pengenalan Objek Menggunakan Mxnet</i> (Ratnasari, 2018)	(+) Pemahaman konsep deep learning dijelaskan secara mendetail. (-) Jumlah lapisan konvolusi yang digunakan hanya 2, serta jumlah data yang digunakan hanya sedikit sehingga hasil akurasi cenderung rendah.

BAB III

METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

Pada bab ini dibahas tentang metodologi penelitian dan metodologi pengembangan sistem *prototype* yang digunakan sebagai acuan dari penelitian *Convolutional Neural Network* Klasifikasi Batik Tulis Motif *Parang* Menggunakan *MobileNet*. Berikut tahapan metodologi penelitian yang dilakukan dalam penelitian ini dalam gambar 3.1:



Gambar 3.1. Tahapan Metodologi Penelitian

3.1. Tahap Pengumpulan dan Pengolahan Data

Metodologi pengumpulan dan pengolahan data dibutuhkan untuk mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan dalam penelitian ini, gambaran umum tentang penelitian yang dilakukan, serta metodologi yang digunakan dalam pengolahan data citra batik. dengan menggunakan beberapa metode berikut ini:

3.1.1. Studi Pustaka

Studi Pustaka pada penelitian ini digunakan untuk mencari berbagai sumber referensi dari jurnal, buku, dan penelitian terdahulu yang terkait dengan penelitian yang dilakukan serta mencari *state of the art* dari penelitian yang dilakukan. Pada tahap ini juga digunakan sebagai pendalaman materi tentang apa saja yang diperlukan dan dibutuhkan untuk membuat penelitian yang dilakukan.

Pada tahap ini dilakukan pencarian informasi mengenai metode dan teknik mengklasifikasikan motif batik parang dengan pengolahan citra digital untuk mencari data perbandingan dan informasi lainnya yang dapat mendukung pengembangan sistem. Pengumpulan dan pencarian data dilakukan dengan menelaah buku, jurnal, penelitian terdahulu, serta sumber dan referensi kepustakaan lain yang terkait dengan masalah penelitian yang dilakukan.

Beberapa referensi terkait penelitian ini seperti pada buku berjudul Teknik dan Ragam Hias Batik Yogya dan Solo yang ditulis oleh Samsi pada tahun 2011 tentang batik motif parang. Pada penelitian tersebut dijelaskan tentang tata cara membuat batik tulis menggunakan lilin malam. Selain itu, dijelaskan juga tentang berbagai macam motif batik parang beserta contoh sketsa motif parang tersebut.

Referensi penelitian berikutnya adalah penelitian berjudul Konsep Pengolahan Citra Digital dan Ekstraksi Fitur yang dilakukan oleh Purnomo dan Muntasa pada tahun 2010 tentang pengolahan citra digital. Pada penelitian tersebut dijelaskan konsep pengolahan citra digital, jenis-jenis citra digital, representasi citra digital serta ekstraksi fitur pada citra digital yang dicontohkan dalam kasus pengenalan citra wajah.

Referensi penelitian berikutnya adalah penelitian berjudul *Visualizing and Understanding Convolutional Neural Networks* yang dilakukan oleh Zeiler dan Fergus pada tahun 2014 tentang *Convolutional Neural Network* (CNN). Pada penelitian tersebut dijelaskan tentang dasar-dasar yang terdapat dalam jaringan syaraf CNN, antara lain lapisan konvolusi, lapisan *fully connected*, lapisan *pooling*, fungsi aktivasi, proses *filtering* serta visualisasi alur kerja CNN.

Referensi penelitian berikutnya adalah penelitian berjudul *MobileNet: Efficient Convolutional Neural Network for Mobile Vision Applications* yang dilakukan oleh Howard, dkk pada tahun 2017 tentang implementasi jaringan syaraf konvolusi untuk perangkat *mobile* yaitu *MobileNet*. Pada penelitian tersebut dijelaskan bahwa jaringan *MobileNet* memiliki ukuran model yang lebih kecil dari model CNN standar, yang disebabkan oleh reduksi biaya komputasi yang terjadi pada lapisan *depthwise separable convolution*. Karena biaya komputasi yang lebih rendah tanpa pengurangan akurasi yang signifikan, maka model *MobileNet* cocok digunakan pada aplikasi *android*.

Referensi penelitian berikutnya adalah penelitian berjudul *Mobile Object Detection using Tensorflow Lite and Transfer Learning* yang dilakukan oleh Alsing pada tahun 2018 tentang implementasi *Tensorflow Lite* untuk deteksi objek berbasis perangkat *mobile*. Pada penelitian tersebut dijelaskan implementasi *transfer learning* pada beberapa metode turunan CNN serta *Tensorflow Lite* yang dapat digunakan sebagai penghubung antara model *MobileNet* berbasis *python* dengan aplikasi perangkat *mobile* berbasis *android*.

3.1.2. Pengumpulan Data

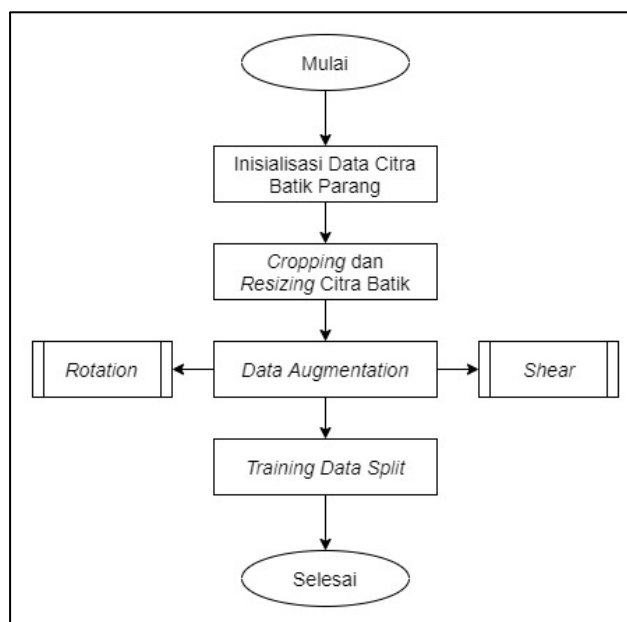
Pada penelitian ini data yang dikumpulkan adalah foto kain batik motif *parang*, yaitu *parang barong*, *parang curigo*, *parang klithik*, *parang kusumo*, *parang rusak* serta *parang tuding*. Pengumpulan data dilakukan dengan cara memfoto batik menggunakan kamera *handphone* dari beberapa sumber, yaitu Kampung Batik Giriloyo, Kabupaten Bantul, butik yang menjual batik di kabupaten Sleman, serta data dari internet. Ukuran citra yang diambil menggunakan kamera ponsel adalah 3600x4500 *pixel* dengan jarak pengambilan gambar sekitar 50-80cm. Pada foto batik yang diambil dari internet, ukurannya menyesuaikan ukuran foto yang tersedia.

Data yang telah dikumpulkan adalah 120 foto batik tulis motif *parang*, berasal dari Kampung Batik Giriloyo, Kabupaten Bantul berjumlah 77 foto batik *parang*, dari butik yang menjual batik di kabupaten Sleman berjumlah 34 foto batik *parang* serta data dari internet sejumlah 29 foto batik *parang*. Ukuran citra yang diambil menggunakan kamera ponsel adalah 3600x4500 *pixel*, sedangkan pada foto batik yang diambil dari internet menyesuaikan ukuran foto yang tersedia.

Kampung Batik Giriloyo merupakan sebuah perkampungan didaerah Imogiri, kabupaten Bantul yang menjadi salah satu pusat pengrajin batik tulis di D.I. Yogyakarta. Di daerah kampung batik Giriloyo terdapat kelompok-kelompok pengrajin batik tulis yang ilmu membatiknya didapatkan secara turun menurun. Salah satu ciri khas dari batik tulis yang dibuat dari kampung batik Giriloyo adalah kombinasi motif batik tulis yang beragam serta nuansa warna batik yang khas Yogyakarta, yaitu warna kecoklatan.

3.1.3. Pengolahan Data

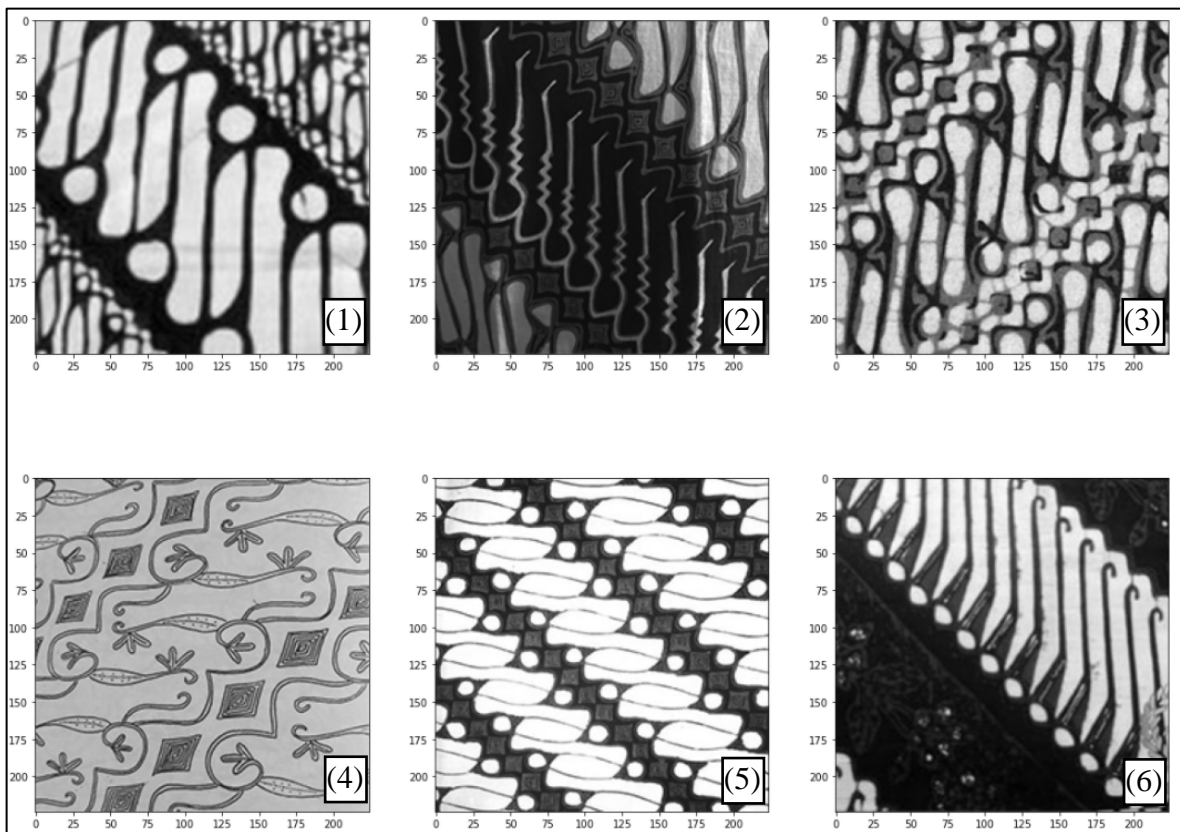
Pengolahan data pada penelitian ini dibagi menjadi beberapa tahap, yaitu inialisasi data citra batik *parang*, *Cropping* dan *Resizing* citra batik, *Data Augmentation*, serta *Training Data Split*. Pengolahan data secara keseluruhan dapat dilihat pada Gambar 3.2 yang merupakan diagram alir pengolahan data.



Gambar 3.2. Diagram Alir Pengolahan Data.

3.1.3.1. Inisialisasi Data Citra Batik Parang

Inisialisasi data citra batik *parang* dilakukan sebagai dasar pembentukan model. Data yang digunakan pada penelitian ini adalah 120 foto batik tulis motif *parang* yang dibagi menjadi 6 kelas motif batik *parang*, yaitu kelas *parang barong*, kelas *parang curigo*, kelas *parang klithik*, kelas *parang kusumo*, kelas *parang rusak* serta kelas *parang tuding*. Masing-masing data foto batik motif *parang* akan dipilah serta dipisahkan secara manual berdasarkan kelas motifnya. Kemudian batik dari masing-masing kelas diberi nama sesuai dengan nama kelasnya. Data citra berupa foto batik kemudian pisahkan dalam suatu direktori kelas motif batik *parang* dengan masing-masing kelas berisi 20 foto batik.

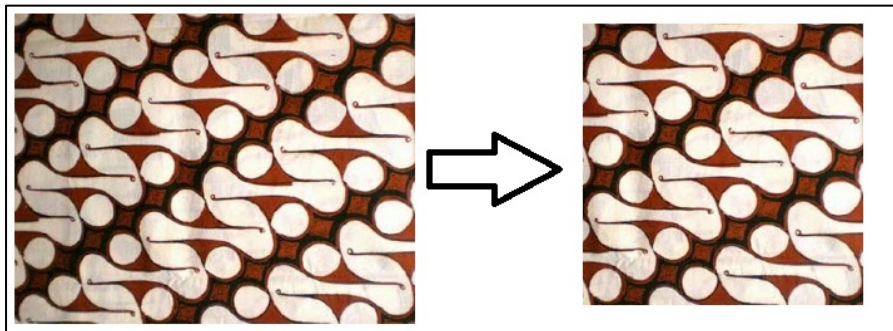


Gambar 3.3. Visualisasi Foto Batik Motif *Parang* dari Setiap Kelas.

Pada Gambar 3.3 dapat dilihat foto dari masing-masing kelas batik motif *parang*. Nomor 1 adalah dari kelas *parang barong*. Ciri khas motif *parang barong* adalah ukuran corak *parang* yang besar dan panjang dibanding motif *parang* lainnya. Nomor 2 adalah dari kelas *parang curigo*. Ciri khas motif *parang curigo* adalah corak *parang* yang menyerupai bentuk keris yang disusun sejajar dengan sudut 45° . Nomor 3 adalah dari kelas *parang klithik*. Ciri khas motif *parang klithik* adalah corak *parang* yang lebih sederhana serta ukuran yang lebih kecil dibanding motif *parang rusak*. Nomor 4 adalah dari kelas *parang kusumo*. Ciri khas motif *parang kusumo* adalah corak *parang* yang menyerupai bunga atau kembang. Nomor 5 adalah dari kelas *parang rusak*. Ciri khas dari motif *parang rusak* adalah susunan motif menyerupai huruf S yang saling terkait satu dengan yang lainnya. Nomor 6 adalah dari kelas *parang tuding*. Ciri khas motif *parang tuding* adalah corak *parang* yang menyerupai jari telunjuk yang disusun secara berjajar dan berkesinambungan.

3.1.3.2. *Cropping* dan *Resizing* Data Citra Batik

Metode *cropping* dan *resizing* pada penelitian ini dilakukan secara manual menggunakan aplikasi *adobe Photoshop CS6*. Setelah data terkumpul dalam satu direktori maka dilakukan *cropping* secara manual pada bagian foto yang terdapat motif batik parang. Metode *cropping* dilakukan supaya citra batik terfokus pada motif batik yang akan dijadikan input pada proses pelatihan model. Kemudian dilakukan *resize* pada data citra batik untuk menyamakan ukuran *pixel* data citra batik menjadi sebesar 224 x 224 px. Setelah proses *cropping* dan *resizing* selesai, jumlah keseluruhan data citra batik bertambah menjadi 300 data citra yang terdiri dari 50 data citra pada masing-masing kelas batik motif parang.



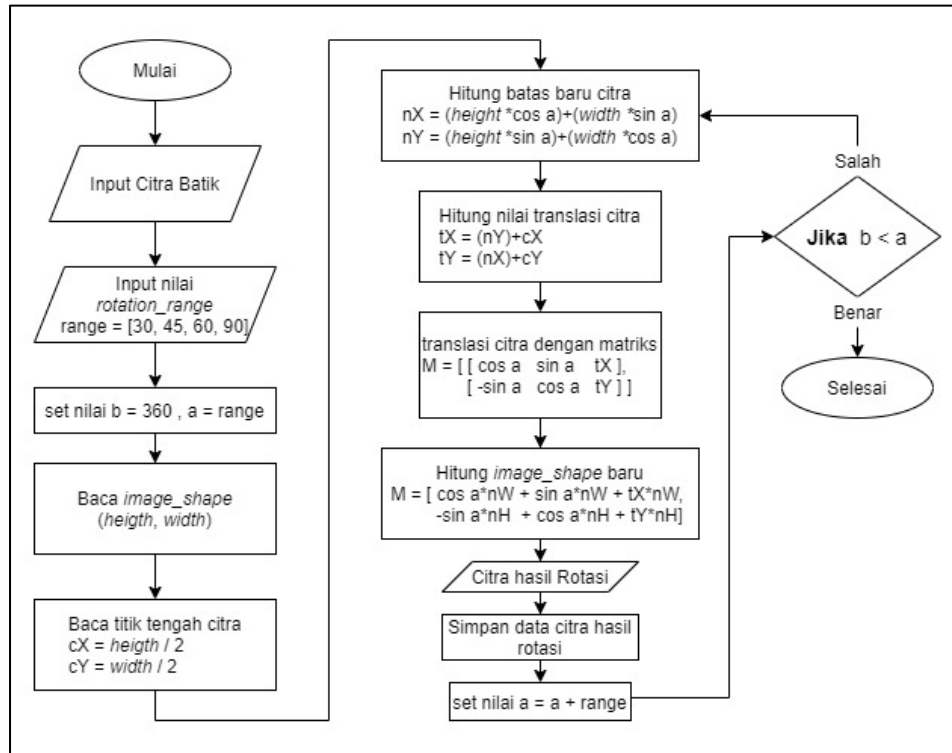
Gambar 3.4. *Cropping* dan *Resizing* citra batik.

3.1.3.3. *Data Augmentation*

Proses selanjutnya adalah *Data Augmentation* atau augmentasi data. Metode ini digunakan untuk menambahkan jumlah citra digital yang dijadikan input saat sistem melakukan pelatihan data. Penggunaan metode ini dilakukan untuk mencegah kasus *over-fitting* yang disebabkan oleh jumlah data yang sedikit. Dalam penelitian ini, metode augmentasi data yang digunakan antara lain:

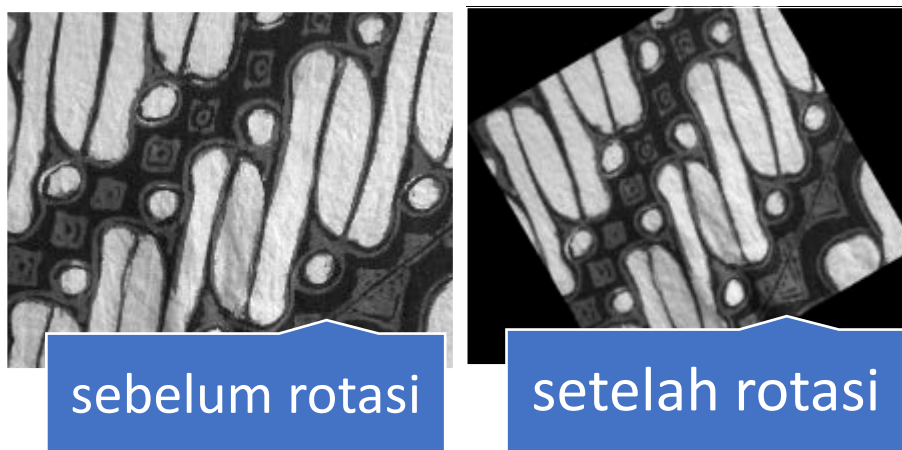
1. *Rotation*

Rotation dengan nilai dalam kisaran derajat antara $0^\circ - 360^\circ$, rentang untuk memutar gambar secara acak sebanyak n kali sesuai dengan nilai *rotation* yang ditentukan. Rentang nilai *rotation_range* yang akan digunakan adalah $5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ$ dan 30° . Berikut pada Gambar 3.5 adalah diagram alir untuk menghitung rotasi gambar.



Gambar 3.5. Diagram alir untuk menghitung rotasi gambar.

Proses melakukan rotasi citra dimulai dengan membaca ukuran citra yang diinputkan, dalam penelitian ini menggunakan citra dengan ukuran 224 x 224. Kemudian membaca input nilai rotasi dengan nama variabel range. Selanjutnya adalah mengatur nilai $b = 360$ dan $a = range$. Kemudian membuat variabel cH dan cW untuk menampung nilai titik tengah citra batik. Kemudian dilakukan perhitungan batas baru untuk citra yang dirotasi nH dan nW . Kemudian membuat matriks rotasi, dimana variabel tH dan tW merupakan nilai translasi gambar dengan rumus tH dan tW . Matriks rotasi lalu ditransformasi dengan matriks ukuran citra yang dirotasi yang menghasilkan matriks berukuran 1×2 yang berisi *image_shape* yang baru. Lalu citra disesuaikan ukurannya dengan ukuran citra input yaitu 224 x 224. Citra sudah selesai dirotasi kemudian disimpan dengan format nama *(nama_kelas)_rotasi(a).jpg*.

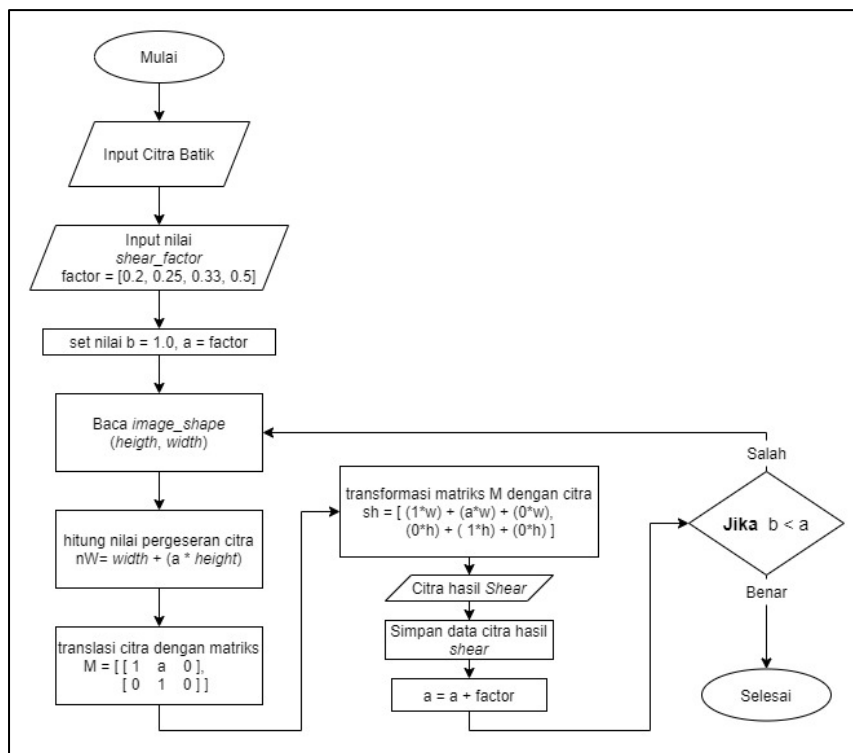


Gambar 3.6. Perbandingan Citra sebelum rotasi dengan citra batik hasil rotasi.

Langkah selanjutnya yaitu masuk kondisi *if* $b < a$ untuk perulangan proses, dimana nilai a ditambahkan terus dengan *range*. Jika kondisi ($b > a$), maka akan kembali ke proses menghitung batas baru nH dan nW . Sedangkan jika ($b < a$), maka proses rotasi citra akan selesai. Jumlah pengulangan proses (n) dapat dihitung dengan membagi nilai b dengan nilai *range*. Perbandingan citra batik sebelum rotasi dan citra batik hasil rotasi gambar dapat dilihat pada Gambar 3.6.

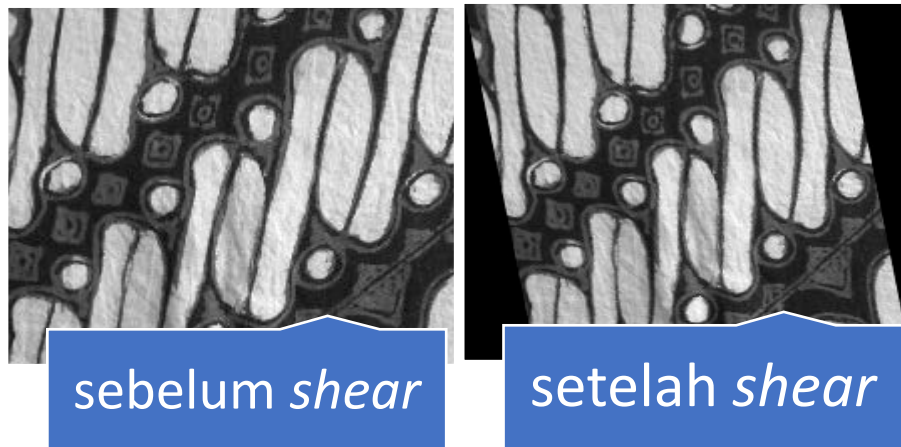
2. Horizontal Shear

Horizontal shear digunakan untuk penerapan transformasi penggeseran gambar secara acak. Nilai shear yang akan digunakan adalah 0.2, 0.25, 0.33 dan 0.5. Berikut pada Gambar 3.7 adalah diagram alir untuk menghitung pergeseran gambar.



Gambar 3.7. Diagram alir untuk menghitung pergeseran gambar.

Proses melakukan *shear* atau menggeser citra dimulai dengan membaca ukuran citra yang diinputkan, dalam penelitian ini menggunakan citra dengan ukuran 224 x 224. Kemudian membaca input nilai *shear_factor* dengan nama variabel *factor*. Selanjutnya adalah mengatur nilai $b = 1.0$ dan $a = factor$. Kemudian dilakukan perhitungan batas baru untuk citra yang dirotasi nW Kemudian membuat matriks untuk translasi, lalu matriks tersebut ditransformasikan dengan ukuran citra yang berisi *image_shape* yang baru. Lalu citra disesuaikan ukurannya dengan ukuran citra input yaitu 224 x 224. Citra sudah selesai digeser kemudian disimpan dengan format nama (*nama_kelas*)_shear(*a*).jpg.



Gambar 3.8. Perbandingan Citra sebelum *shear* dengan citra batik hasil *shear*.

Langkah selanjutnya yaitu kondisi *if* $b < a$ untuk perulangan proses, dimana nilai a akan ditambahkan terus dengan nilai *factor*. Jika kondisi ($b > a$), maka akan kembali ke proses menghitung batas baru nW . Sedangkan jika ($b < a$), maka proses menggeser citra akan selesai. Jumlah pengulangan proses (n) dapat dihitung dengan membagi nilai b dengan nilai *factor*. Perbandingan citra batik sebelum *shear* dan citra batik hasil *shear* gambar dapat dilihat pada Gambar 3.8.

3.1.3.4. *Training Data Split*

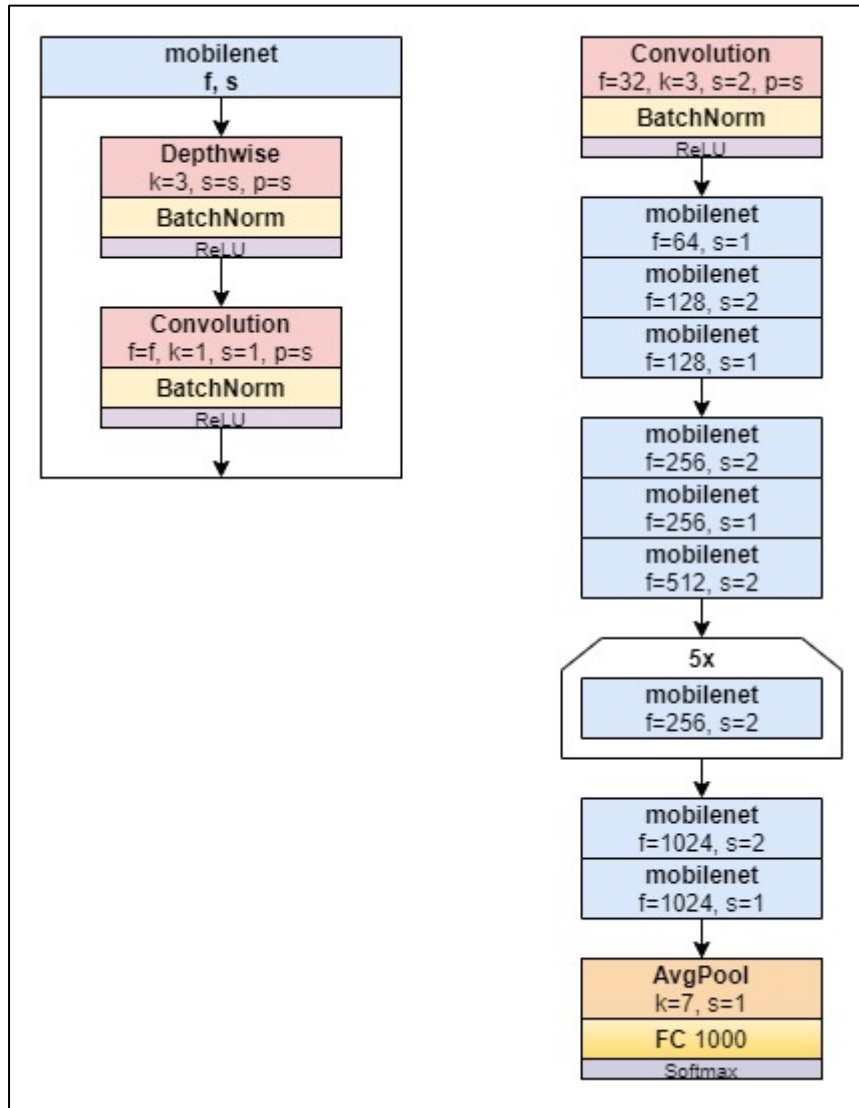
Proses pengolahan data terakhir pada penelitian ini adalah *training data split* atau membagi data menjadi data *training*, data *validation* dan data *testing*. Data *training* digunakan sebagai dasar dalam melatih sistem untuk dapat mengenali citra batik motif parang. Data *validation* digunakan untuk melakukan validasi pada setiap langkah pembelajaran dalam sistem selama proses *training*. Sedangkan data *testing* digunakan untuk menguji hasil pembelajaran setelah proses *training* selesai. Rasio perbandingan data *training* dan data *validation* pada model batik tulis motif parang yang akan dibuat adalah 80 : 20, sedangkan data *testing* menggunakan citra batik yang dibuat pada pengolahan data.

3.2. Tahap Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini menggunakan prinsip metode prototipe. Metode prototipe pada penelitian ini meliputi proses perancangan sistem, pembuatan model batik tulis motif parang, pengujian model dan implementasi model.

3.2.1. Perancangan Sistem

Pada penelitian ini digunakan arsitektur *convolutional neural network* dengan model *MobileNet* dapat berjalan dengan komputasi ringan (Howard et al., 2017). Arsitektur jaringan *MobileNet* dapat dilihat pada Gambar 3.9 berikut.

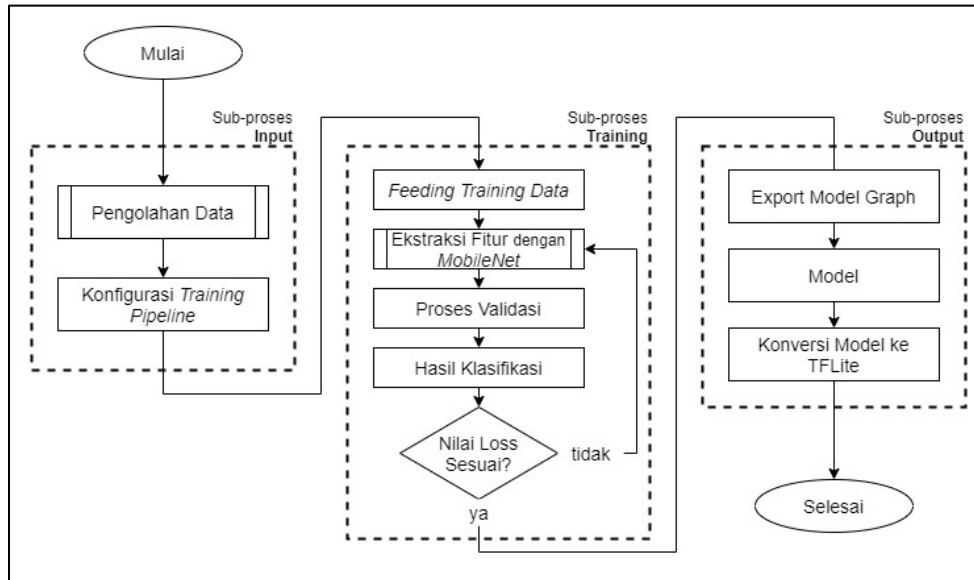


Gambar 3.9. Arsitektur jaringan *MobileNet* (Howard et al., 2017).

Arsitektur ini memiliki 13 buah lapisan konvolusi *mobilenet* yang terdiri dari lapisan *depthwise* dengan ukuran 3×3 serta lapisan *filter* konvolusi *pointwise* dengan ukuran 1×1 . Seluruh lapisan konvolusi juga memiliki fungsi aktivasi ReLU. Jumlah filter akan meningkat disetiap blok. Kemudian setelah lapisan konvolusi terakhir dengan jumlah filter sebanyak 1024 terdapat lapisan *global average pooling*. Kemudian masuk ke lapisan *fully connected* yang menghubungkan semua *hidden layer* dalam jaringan untuk masuk proses klasifikasi. Keluaran dari rangkaian proses ini akan diklasifikasikan dengan *classifier* yang menggunakan fungsi aktivasi *softmax*. *Classifier* pada rangkaian proses ini memiliki jumlah output yang disesuaikan dengan target output model batik tulis motif barang, yaitu 6 output.

3.2.2. Pembuatan Model Batik Tulis Motif Parang

Proses pembuatan model dalam penelitian ini dibagi menjadi 3 sub-proses, antara lain sub-proses input, sub-proses training dan sub-proses output. Diagram alir pembuatan model batik tulis motif parang dapat dilihat pada Gambar 3.10. berikut ini.



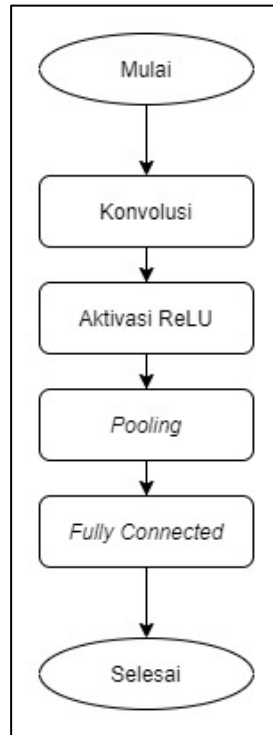
Gambar 3.10. Diagram Alir Pembuatan Model Batik Tulis Motif Parang.

3.2.2.1. Sub-proses Input

Data yang sudah dikumpulkan menjadi dataset diolah pada tahapan Pengolahan Data yang telah dibahas sebelumnya, lalu digunakan sebagai input dalam pembuatan model ini. Selanjutnya adalah konfigurasi *training pipeline* untuk mengatur parameter yang mempengaruhi hasil *training*, antara lain *epochs*, *batch size*, serta opsi untuk melakukan *early stopping*. Pada bagian *early stopping* digunakan dapat menghentikan *training* dengan memonitor nilai *loss* atau *accuracy*.

3.2.2.2. Sub-proses Training

Pada sub-proses *training*, data yang telah diproses pada sub-proses input di-*feed* kedalam arsitektur jaringan *MobileNet* untuk dilakukan *training*. Seperti yang sudah dibahas pada bagian perancangan sistem, pada arsitektur jaringan *MobileNet* memiliki 27 lapisan konvolusi. Pada lapisan konvolusi tersebut dilakukan *batch normalization* dan fungsi aktivasi ReLU setelah proses konvolusi. Diagram alir ekstraksi fitur dengan *MobileNet* dapat dilihat pada gambar 3.11.

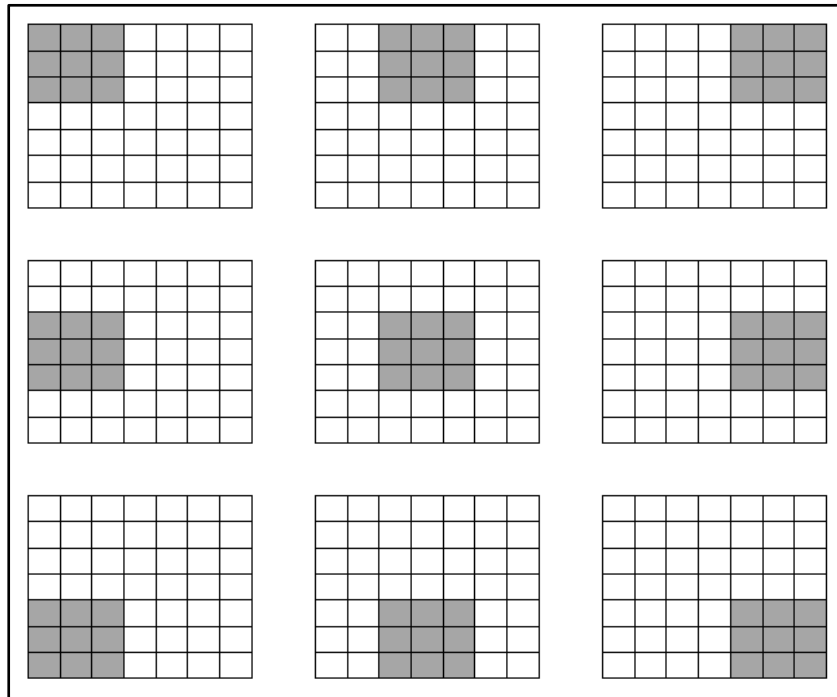


Gambar 3.11. Diagram Alir Ekstraksi Fitur dengan MobileNet.

Bagian ekstraksi fitur merupakan bagian dalam jaringan *MobileNet* yang akan melakukan identifikasi terhadap citra batik yang kemudian akan mengekstrak fitur dari masing-masing motif batik agar dapat diklasifikasikan pada bagian selanjutnya. Ekstraksi fitur terdiri dari lapisan konvolusi, lapisan *pooling*, fungsi aktivasi ReLU, dan lapisan *fully connected*.

1. Konvolusi

Proses konvolusi adalah proses untuk mengekstrak fitur dari suatu citra input, sekaligus mempelajari representasi fitur yang ada pada citra tersebut. Didalam proses konvolusi terdapat *filter* yang memiliki ukuran tinggi, lebar dan tebal tertentu. Filter dapat diinisialisasikan dengan suatu nilai atau dapat menjadi *random*. Nilai dari filter ini kemudian akan di-*update* secara terus menerus sebagai parameter proses *training*. Pada penelitian ini digunakan filter berukuran 3 x 3 pada lapisan *depthwise* dan ukuran 1 x 1 pada lapisan *pointwise*. Proses konvolusi yang diilustrasikan dengan input gambar berukuran 7 x 7, *filter* dengan ukuran 3 x 3 serta *stride*=2 dapat dilihat pada Gambar 3.12.



Gambar 3.12. Ilustrasi proses konvolusi

Proses konvolusi dilakukan dengan dimulai dari sudut kiri atas citra asli yaitu pada posisi (0,0) kemudian akan melakukan pergeseran hingga pojok kiri bawah. Pergeseran tersebut dilakukan oleh *filter* berukuran 3 x 3 yang terjadi saat proses konvolusi. *Filter* bergeser sebanyak 2 *pixel* karena jumlah *stride*=2. Dari pergeseran sesuai nilai *filter* ini akan dihasilkan nilai *pixel* baru untuk *pixel* yang terkena konvolusi. Nilai pada *pixel* yang tidak terkena konvolusi akan tetap sama atau nilainya sesuai citra input.

2	8	3	1	6	3	4
5	4	9	4	7	5	2
8	0	5	7	8	3	0
1	4	2	0	9	1	2
4	8	7	3	0	9	4
7	2	3	6	5	7	5
2	6	9	1	2	5	8

 \times

0	-1	0
1	2	1
0	-1	0

 $=$

2	8	3	1	6	13	4
5	4	9	4	7	5	2
8	0	14	16	13	3	0
11	4	3	1	1	11	2
4	8	0	16	10	9	4
7	2	3	6	5	7	5
2	6	9	1	2	5	8

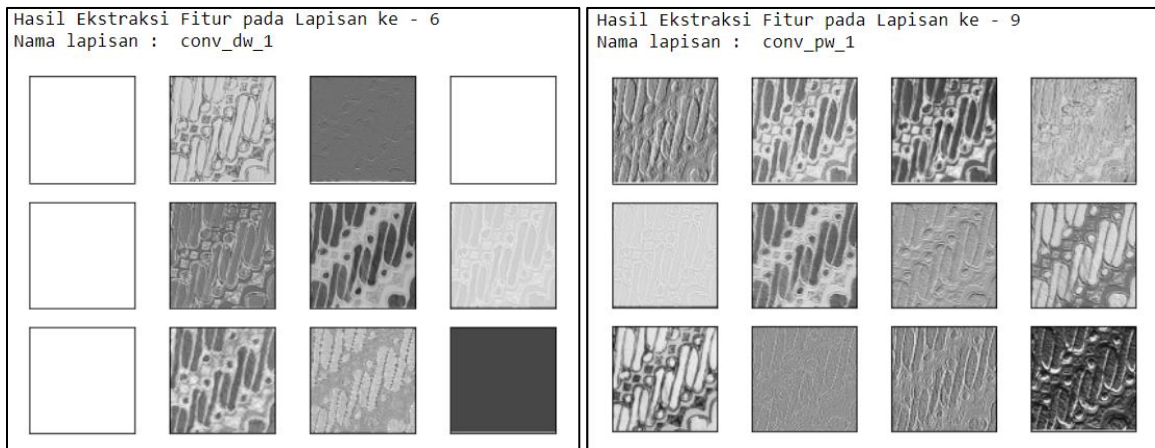
Gambar 3.13. Perhitungan Konvolusi.

Perhitungan nilai tengah pada proses konvolusi dengan *filter* berukuran 3x3 menghasilkan nilai baru pada matriks tengah, Berikut adalah perhitungan konvolusi dengan ukuran *filter* 3x3 pada Gambar 3.13:

$$\begin{array}{l}
 f(1,1) | (2 \times 0) + (8 \times (-1)) + (3 \times 0) + (5 \times 1) + (4 \times 2) + (9 \times 1) + (8 \times 0) + (0 \times (-1)) + (5 \times 0) = 14 \\
 f(1,3) | (3 \times 0) + (1 \times (-1)) + (6 \times 0) + (9 \times 1) + (4 \times 2) + (7 \times 1) + (5 \times 0) + (7 \times (-1)) + (8 \times 0) = 16 \\
 f(1,5) | (6 \times 0) + (3 \times (-1)) + (4 \times 0) + (7 \times 1) + (5 \times 2) + (2 \times 1) + (8 \times 0) + (3 \times (-1)) + (0 \times 0) = 13 \\
 f(3,1) | (8 \times 0) + (0 \times (-1)) + (5 \times 0) + (1 \times 1) + (4 \times 2) + (2 \times 1) + (4 \times 0) + (8 \times (-1)) + (7 \times 0) = 3 \\
 f(3,3) | (5 \times 0) + (7 \times (-1)) + (8 \times 0) + (2 \times 1) + (0 \times 2) + (9 \times 1) + (7 \times 0) + (3 \times (-1)) + (0 \times 0) = 1 \\
 f(3,5) | (8 \times 0) + (3 \times (-1)) + (0 \times 0) + (9 \times 1) + (1 \times 2) + (2 \times 1) + (0 \times 0) + (9 \times (-1)) + (4 \times 0) = 1 \\
 f(5,1) | (4 \times 0) + (7 \times (-1)) + (8 \times 0) + (7 \times 1) + (2 \times 2) + (3 \times 1) + (2 \times 0) + (6 \times (-1)) + (9 \times 0) = 0 \\
 f(5,3) | (7 \times 0) + (3 \times (-1)) + (0 \times 0) + (3 \times 1) + (6 \times 2) + (5 \times 1) + (9 \times 0) + (1 \times (-1)) + (2 \times 0) = 16 \\
 f(5,5) | (0 \times 0) + (9 \times (-1)) + (4 \times 0) + (5 \times 1) + (7 \times 2) + (5 \times 1) + (2 \times 0) + (5 \times (-1)) + (8 \times 0) = 10
 \end{array}$$

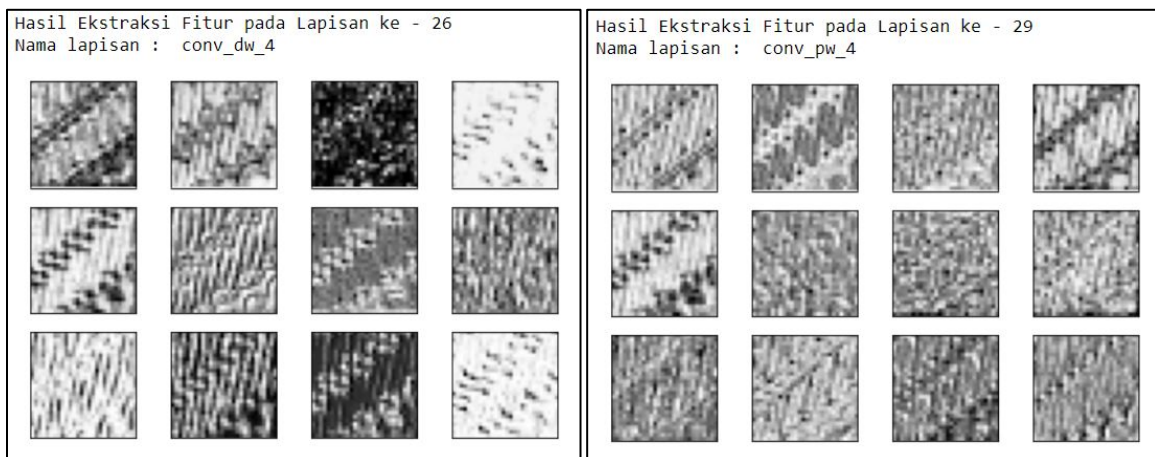
Representasi citra pada proses konvolusi berbentuk matriks dikalikan dengan matriks filter F , yang akan menghasilkan matriks *feature map* berisi fitur yang dominan berdasarkan filter yang digunakan. Pada setiap *feature map* atau *activation map* akan digunakan fungsi aktivasi ReLU untuk menentukan keluaran dari lapisan tersebut.

Pada lapisan input Mobilenet, citra diubah dalam bentuk matriks dengan 3 kanal yaitu RGB. Pada lapisan konvolusi pertama setelah lapisan input dilakukan proses konvolusi pertama dengan *stride*=2 serta *padding*=2, yang kemudian akan mengubah ukuran citra menjadi 112x112. Kemudian dilakukan normalisasi *batch* serta fungsi aktivasi ReLU. Setelah itu citra akan masukkan pada lapisan *depthwise* untuk dipisahkan menjadi 3 kanal dari masing-masing warna yaitu merah (*Red*), hijau (*Green*) dan biru (*Blue*).



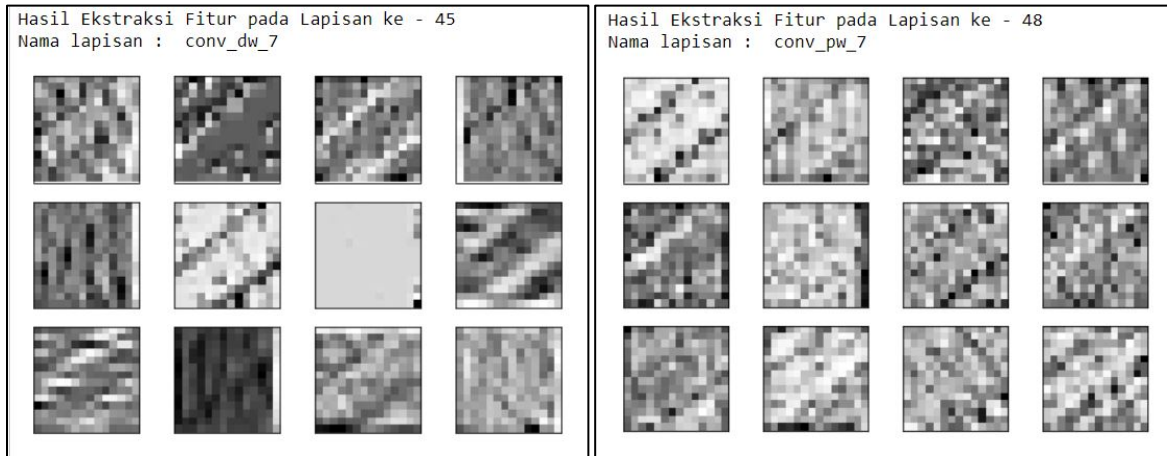
Gambar 3.14. Lapisan Konvolusi *depthwise* dan *pointwise* ke-1.

Pada lapisan konvolusi pertama akan menghasilkan output citra yang masih terlihat jelas bentuk corak batik parang rusaknya. Pada lapisan ini, garis pola akan terlihat lebih jelas pada beberapa hasilnya. Pada Gambar 3.14 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-1.



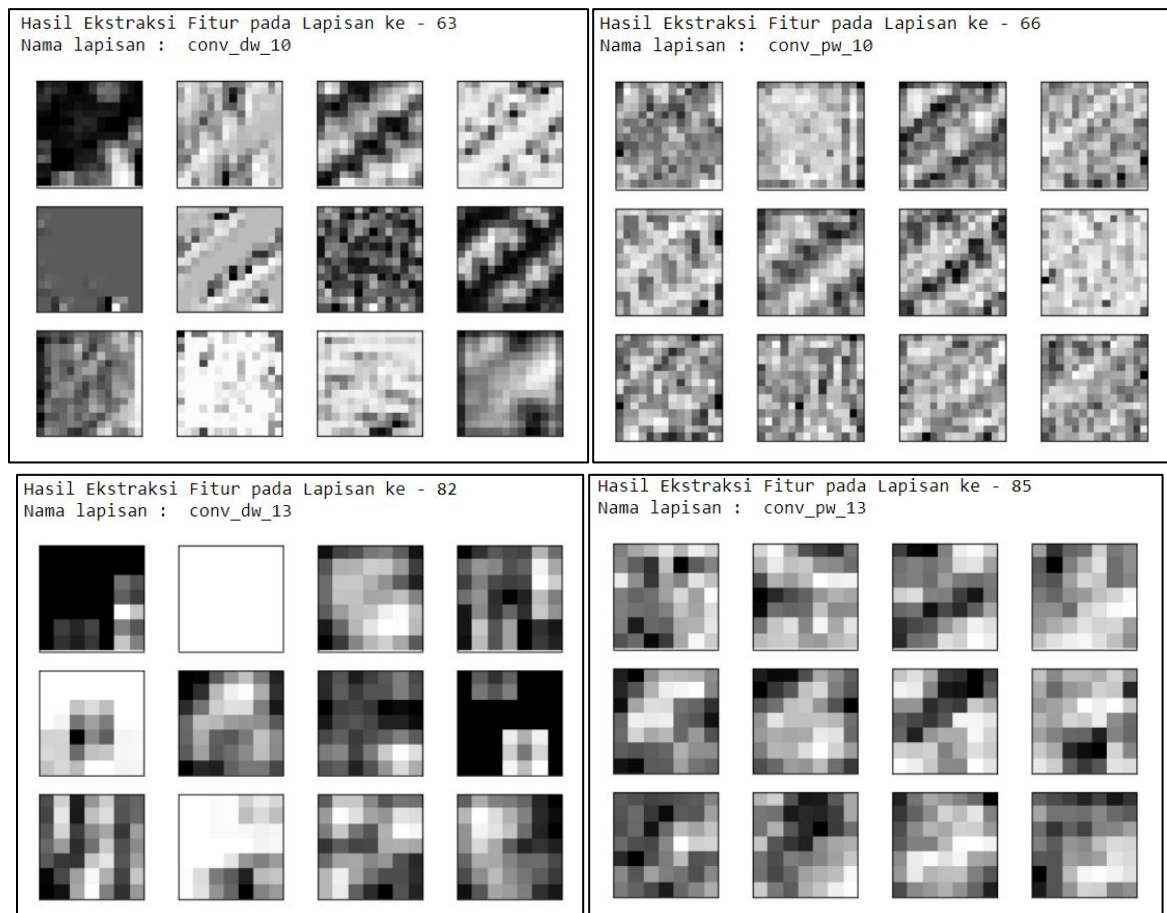
Gambar 3.15. Lapisan Konvolusi *depthwise* dan *pointwise* ke-4.

Pada lapisan konvolusi ke-4 akan menghasilkan output citra yang masih terlihat cukup jelas bentuk corak batik parang rusaknya. Pada lapisan ini, fitur dari citra batik telah sebagian diekstrak, sehingga gambar yang dihasilkan akan sedikit *blur*. Pada Gambar 3.15 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-4.



Gambar 3.16. Lapisan Konvolusi *depthwise* dan *pointwise* ke-7.

Pada lapisan konvolusi ketujuh, fitur dari citra batik telah sebagian diekstrak, sehingga gambar yang dihasilkan akan *blur*. Pada Gambar 3.16 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-7.



Gambar 3.17. Lapisan Konvolusi *depthwise* dan *pointwise* ke-10 dan ke-13.

Pada lapisan konvolusi ke-10 serta ke-13 akan menghasilkan output citra yang bentuk corak batik parang rusak *pixelated* atau sangat terlihat bagian pikselnya. Namun jika diperhatikan lebih dekat, pola corak parang masih dapat dilihat dengan jelas. Pada lapisan ini, fitur dari citra batik sebagian besar telah diekstrak, sehingga gambar yang dihasilkan

akan *blur*. Pada Gambar 3.17 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-10 dan ke-13.

2. Pooling

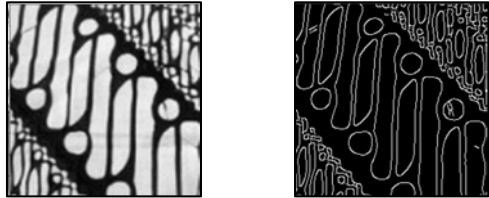
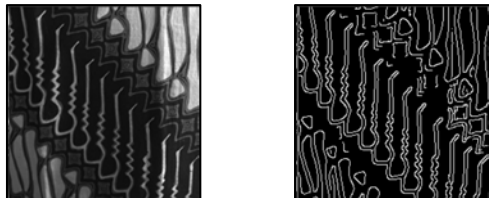
Bagian berikutnya setelah fungsi aktivasi adalah *pooling* yang berfungsi untuk mereduksi input secara spasial serta mengurangi jumlah parameter dengan operasi *down-sampling*. Pada penelitian ini digunakan *average pooling* yang mengambil nilai rata-rata dari matriks filter berukuran $F \times F$. Proses pada lapisan *pooling* mirip seperti pada lapisan konvolusi, filter akan menghitung nilai rata-rata kemudian bergerak berdasarkan *stride*.

3. Fully Connected

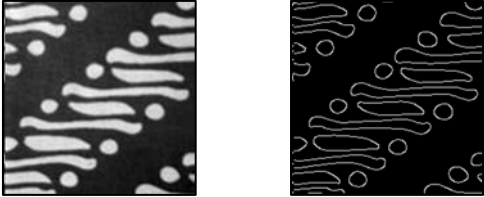
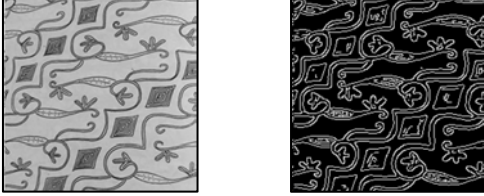
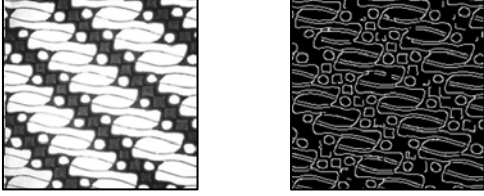
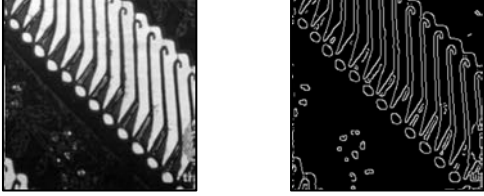
Setelah melakukan ekstraksi fitur serta klasifikasi pada lapisan *fully connected*, dilakukan validasi terhadap data yang sudah dilatih dalam jaringan. Data yang sudah dilatih pada setiap *epochs* kemudian dicocokkan dengan data validasi untuk mengurangi kesalahan selama proses pelatihan data. Hasil validasi kemudian akan meng-*update* bobot dari setiap lapisan yang terlibat selama pelatihan. Untuk hasil pelatihan yang lebih baik, serta mencegah kasus *overfitting* dilakukan *dropout* atau menon-aktifkan lapisan yang memiliki nilai bobot yang lebih rendah daripada nilai *dropout* yang telah ditentukan. Pada pelatihan model batik motif parang ini digunakan nilai *dropout* = 0.05. Data yang telah divalidasi kemudian diklasifikasikan berdasarkan kelasnya.

Ekstraksi fitur yang telah dilakukan menghasilkan pola pengenalan corak pada batik motif parang. Pola pengenalan motif batik parang dapat dilihat pada tabel 3.1 berikut ini.

Tabel 3.1 Pola Pengenalan Corak Batik Motif Parang.

Motif Batik Parang	Pola Pengenalan Corak Batik
<p><u>Parang Barong</u></p> 	<ul style="list-style-type: none"> • Ukuran Corak Parang yang lebih besar dibanding Motif Parang lainnya.
<p><u>Parang Curigo</u></p> 	<ul style="list-style-type: none"> • Bentuk meruncing pada ujung corak parang. • Bentuk motif parang yang pipih. • Terdapat pola <i>zig-zag</i> pada corak parang.

Tabel 3.2 Lanjutan Pola Pengenalan Corak Batik Motif Parang.

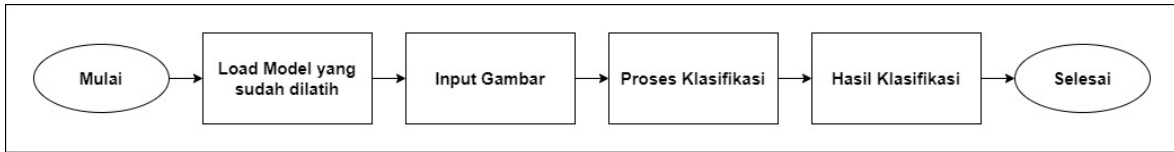
<p><u>Parang Klithik</u></p> 	<ul style="list-style-type: none"> • Corak parang yang landai, datar dan cenderung meruncing.
<p><u>Parang Kusumo</u></p> 	<ul style="list-style-type: none"> • Corak parang yang lentik. • Terdapat corak bunga atau kembang pada motif parang kusumo. • Corak mlinjon disekitar motif parang dapat dilihat dengan jelas.
<p><u>Parang Rusak</u></p> 	<ul style="list-style-type: none"> • Susunan motif parang rapat-rapat. • Corak parang cenderung membulat. • Terdapat mlinjon disekitar corak parang.
<p><u>Parang Tuding</u></p> 	<ul style="list-style-type: none"> • Corak parang yang lurus, mendatar dan rapat. • Tidak ada corak mlinjon. • Corak bulat pada ujung corak parang. • Corak parang yang saling menyambung.

Setelah menentukan hasil klasifikasi, nilai akurasi dan loss akan diperbarui secara terus menerus selama epochs masih tersedia. Proses klasifikasi akan berhenti ketika pelatihan data sudah tidak dapat mengurangi nilai loss dari hasil pelatihan. Sub-proses *training* akan terus diulang sesuai dengan *training pipeline* dikonfigurasi sebelumnya. Setelah sub-proses *training* selesai dilakukan, akan dilanjutkan dengan sub-proses output.

3.2.2.3. Sub-proses Output

Sub-proses output mengubah hasil sub-proses *training* menjadi grafik dan sebuah model yaitu model batik motif parang. Grafik yang ditampilkan adalah grafik akurasi dari proses validasi. Model yang telah berhasil dibuat kemudian dikonversi menjadi bentuk file *model.tflite* yang akan digunakan untuk klasifikasi batik menggunakan aplikasi berbasis *android*.

3.2.3. Pengujian Model



Gambar 3.18. Diagram Alir Pengujian Model Batik Tulis Motif Parang.

Akurasi merupakan salah satu parameter dalam penelitian ini. Akurasi didapat setelah proses pengujian model dalam bentuk presentase. Perhitungan akurasi dilakukan pada citra batik dengan motif parang yang berbeda. Dari perhitungan tersebut dapat diketahui tingkat akurasi model batik tulis motif parang dengan arsitektur jaringan *MobileNet* dari metode *Convolutional Neural Network* dalam klasifikasi batik tulis motif parang. Berikut adalah rumus perhitungan presentasi akurasi.

$$\text{Presentase akurasi} = \frac{\text{Jumlah citra data uji benar}}{\text{Jumlah keseluruhan data citra uji}}$$

Pengujian model pada penelitian ini dilakukan secara terpisah dari proses training. Data *testing* yang diinputkan kedalam model yang sudah dilatih kemudian model akan melakukan klasifikasi. Output dari proses pengujian yang dilakukan berupa matriks *confussion* berukuran 6 x 6, sesuai dengan jumlah kelas yang terdapat pada model yang sudah dilatih, yang dapat dilihat pada Gambar 3.19.

```
240/240 [=====] - 5s 17ms/step
Confusion matrix, without normalization
[[40  0  0  0  0  0]
 [ 0 40  0  0  0  0]
 [ 0  0 40  0  0  0]
 [ 0  0  0 40  0  0]
 [ 0  0  0  0 40  0]
 [ 0  0  0  1  0 39]]
```

Gambar 3.19. Output proses pengujian model.

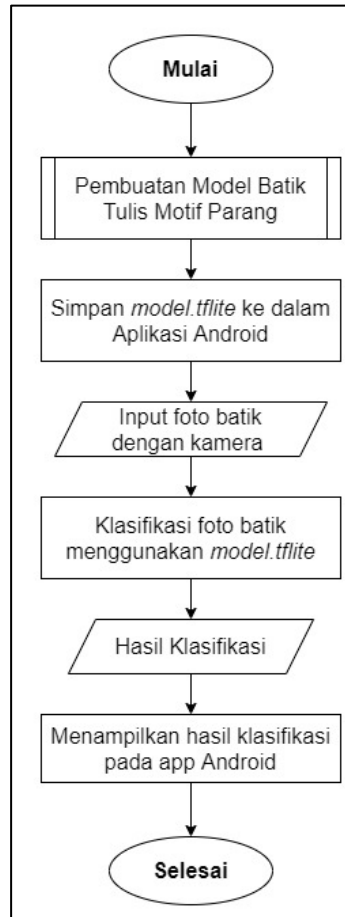
Berdasarkan data hasil pengujian sesuai dengan output pada gambar 3.19, didapatkan hasil pengujian model sebagai berikut.

$$\frac{239}{240} \times 100\% = 99.58\%$$

3.2.4. Implementasi Model

3.2.4.1. Perancangan Proses

Diagram alir yang ada pada Gambar 3.21 merupakan alur input dan output keseluruhan dalam mengklasifikasikan batik tulis motif parang. Perancangan proses dibagi menjadi 2 yaitu sub proses pembuatan model pada aplikasi berbasis *python* kemudian sub proses klasifikasi menggunakan model pada aplikasi berbasis *android*.



Gambar 3.20. Diagram Alir Proses *Input* dan *Output*.

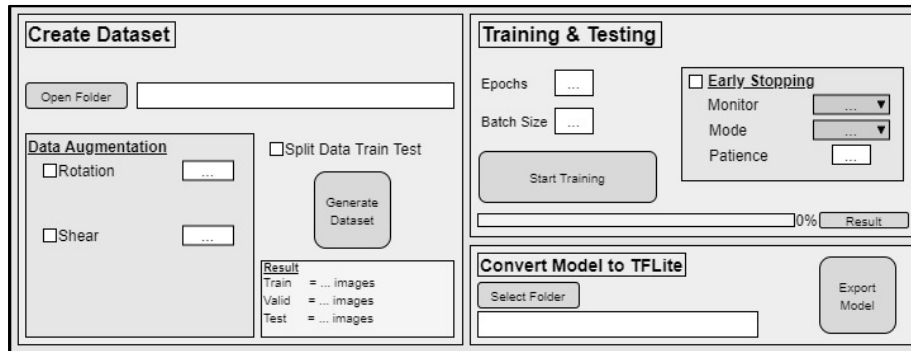
Proses dimulai pada sub proses pembuatan model dengan input data citra batik yang kemudian akan diolah pada proses pengolahan data. Proses selanjutnya adalah proses *training* data yang sudah diolah menggunakan arsitektur jaringan *MobileNet* pada aplikasi berbasis *python*. Output dari proses *training* adalah model batik tulis. Model hasil output tersebut dikonversi menjadi format *.tflite* agar dapat dikenali oleh aplikasi berbasis *android*. Proses dilanjutkan pada sub proses klasifikasi berbasis *android*.

Pada sub proses klasifikasi berbasis *android*, input yang digunakan berasal dari kamera *smartphone*. Input tersebut kemudian dilakukan klasifikasi menggunakan model batik tulis yang telah dilatih sebelumnya. Selanjutnya adalah *output*, yang akan menampilkan hasil klasifikasi berupa 3 prediksi hasil dengan persentase benar tertinggi.

3.2.4.2. Perancangan Antarmuka

Perancangan antarmuka pada penelitian ini dibagi menjadi 2 bagian, yaitu perancangan antarmuka berbasis *python* pada *platform desktop* untuk pembuatan model dan antarmuka berbasis *android* untuk klasifikasi batik tulis motif parang. Berikut rancangan *user interface* sesuai dengan *flowchart* proses pada sistem ini:

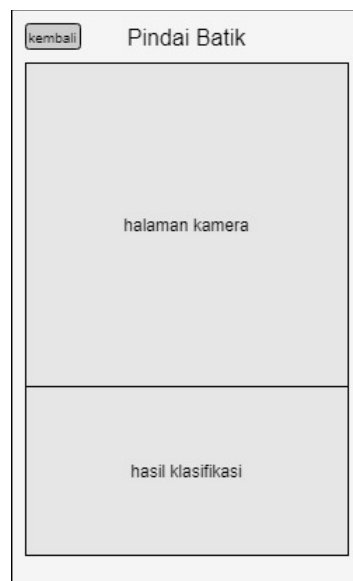
1. Perancangan Antarmuka Aplikasi *Desktop* untuk Pembuatan Model



Gambar 3.21. Halaman Pembuatan Model Batik Tulis Motif Parang.

Pada halaman ini terdapat 3 proses yaitu membuat dataset, *training & testing* dan mengubah model menjadi *TFLite*. Bagian membuat dataset digunakan untuk mempersiapkan dataset, pilihan untuk melakukan augmentasi data, serta membagi data pelatihan dan data pengujian seperti pada gambar 3.21 yaitu pra-proses data pelatihan. Pada bagian *training & testing* digunakan untuk melakukan pelatihan dengan konfigurasi jumlah *epochs*, ukuran *batch* serta *early stopping*. Pada bagian konversi model menjadi *TFLite* digunakan untuk menyimpan model yang sudah dilatih dalam bentuk *TFLite*.

2. Perancangan Antarmuka *Prototype* Aplikasi *Android* untuk Klasifikasi Batik Tulis Motif Parang



Gambar 3.22. Halaman Pindai Batik.

Halaman ini merupakan halaman dilakukannya pemindaian batik melalui kamera serta menampilkan hasil klasifikasi batik. Terdapat 2 bagian serta sebuah tombol pada halaman ini. Bagian atas adalah halaman kamera, dimana akan menampilkan pemindaian secara *real-time* sesuai yang ditangkap oleh kamera. Bagian bawah halaman kamera adalah hasil klasifikasi yang menampilkan hasil dari klasifikasi dari gambar yang diambil dari halaman kamera. Terakhir adalah tombol kembali, yaitu untuk keluar dari aplikasi klasifikasi batik tulis.

BAB IV

HASIL, PENGUJIAN DAN PEMBAHASAN

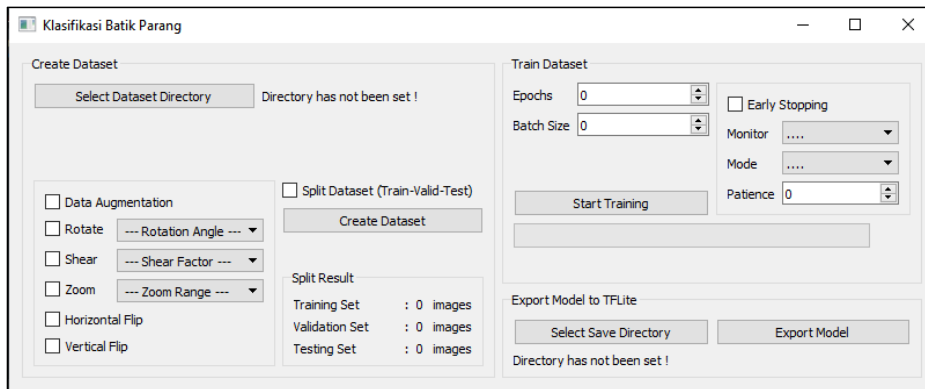
Bab hasil, pengujian dan pembahasan merupakan penjelasan dari penelitian yang telah dilakukan sesuai dengan keadaan yang sebenarnya. Pada bab ini terdiri dari hasil, pengujian dan pembahasan. Dari pembahasan ini akan dapat diketahui apakah hasil dari sistem yang dibangun mampu menghasilkan tujuan yang diinginkan berdasarkan analisis dan perancangan yang telah dilakukan.

4.1. Hasil Penelitian

Hasil penelitian mencakup pembuatan model dengan metode *Convolutional Neural Network* menggunakan arsitektur model *MobileNet* untuk mengenali pola pada batik motif parang. Pembuatan model tersebut melalui beberapa tahapan yang telah digambarkan pada bab perancangan sistem. Implementasi model dibuat dalam bentuk aplikasi *android* untuk mengklasifikasikan motif batik parang dengan model yang telah dibuat

4.1.1. Aplikasi *Desktop* untuk Pembentukan Model

Halaman aplikasi *desktop* untuk pembentukan model dibagi menjadi 3 bagian, yaitu bagian *Create Dataset* untuk pembentukan *dataset*, bagian *Train Dataset* untuk memulai proses pelatihan data, dan bagian *export model to TFLite* untuk mengekspor data dalam model yang sudah dilatih agar dibaca oleh aplikasi *android*. Tampilan dari halaman aplikasi *desktop* untuk pembentukan model dapat dilihat pada gambar 4.1.



Gambar 4.1. Halaman Aplikasi *Desktop* untuk pembentukan Model.

Pada bagian *Create Dataset* terdapat sub bagian berisi pilihan untuk melakukan *Data Augmentation*. didalam sub bagian ini terdapat beberapa opsi *data augmentation* yang dapat dilakukan, antara lain *rotate* (rotasi), *shear* (menggeser), *zoom* (memperbesar), serta *horizontal* dan *vertical flip* (memutar gambar secara horisontal/vertikal). Pada bagian *Create dataset* terdapat dua buah tombol, yaitu tombol *Select Dataset Directory* serta tombol *Create Dataset*.

1. Tombol *Select Dataset Directory*

Fungsi dari tombol tersebut untuk memilih direktori *folder* tempat dataset yang telah diolah sebelumnya tersimpan dalam komputer pengguna. Direktori yang telah dipilih dan telah dipastikan berisi gambar kemudian dijadikan sebagai acuan dalam menentukan

direktori pada langkah selanjutnya. Tampilan dari fungsi tombol *Select Dataset Directory* dapat dilihat pada gambar 4.2 dan *source code Select Dataset Directory* ditunjukkan pada modul program 4.1.

```

FUNCTION select_dir :
    GET selected_dir
    RETURN selected_dir
ENDFUNCTION

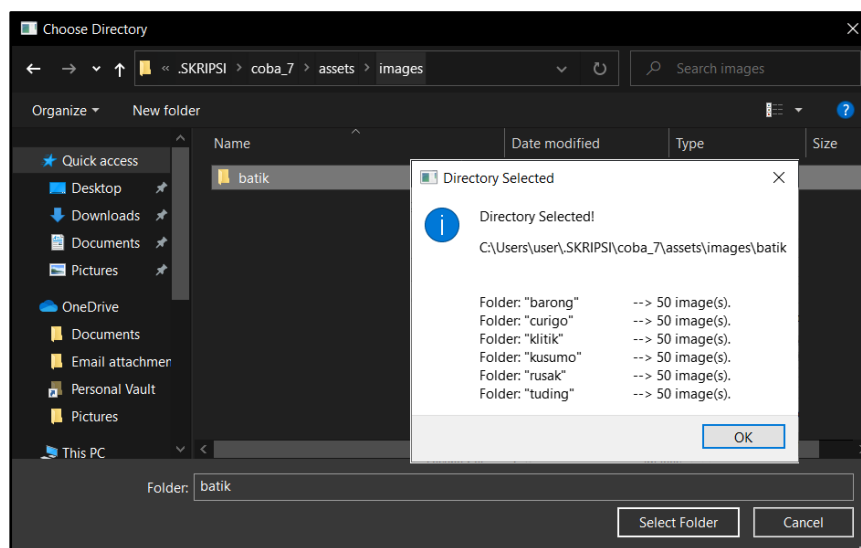
FUNCTION check_directory :
    INIT :
        directory <- STRING

    FOR item in data_dir.glob('*') THEN
        IF item.name != "LICENSE.txt" THEN
            SET folders TO item.name]
        ENDIF
    ENDFOR

    FOR folder in folders :
        SET dir_ TO directory, folder
        INCREMENT img_count
    ENDFOR
ENDFUNCTION

```

Modul Program 4.1 *Source code Select Dataset Directory.*



Gambar 4.2. Tampilan *checkDirectory* setelah direktori dataset terpilih.

2. Tombol *Create Dataset*

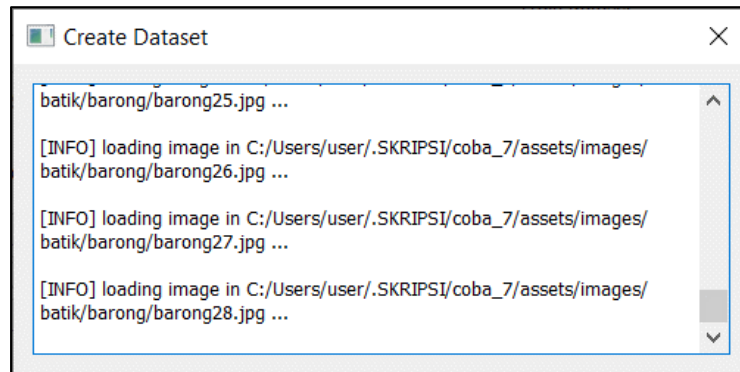
Setelah direktori dataset berhasil dipilih, akan muncul opsi untuk melakukan augmentasi data dan *split dataset* serta mengaktifkan tombol untuk membuat dataset. Tombol *create dataset* berfungsi untuk memproses parameter augmentasi data dan *split dataset*. Ketika tombol create dataset diklik, akan memunculkan *window* berisi informasi proses dalam pembuatan dataset. Tampilan dari fungsi tombol *Create Dataset* dapat dilihat pada gambar 4.3 dan *source code Create Dataset* ditunjukkan pada modul program 4.2

```

FUNCTION create_dataset :
  aug_param <- {GET shear_factor,
               GET angle,
               GET zoom_range,
               GET Hflip,
               GET Vflip}
  CALL dataset_preparations
ENDFUNCTION

```

Modul Program 4.2 Modul Program dari fungsi tombol *Create Dataset*.



Gambar 4.3. Tampilan dari fungsi tombol *Create Dataset*.

3. Tombol *Start Training*

Fungsi dari tombol ini adalah untuk memulai proses *training* dengan mengirimkan *training pipeline* sebagai parameter proses training. Ketika *training* dimulai, akan muncul *terminal window* yang menampilkan proses *training* seperti pada gambar 4.4. Proses *training* terdapat pada modul 4.3.

```

FUNCTION model_config :
  mobilenet <- MobileNet(include_top <- False,
                        input_shape <- (img_height, img_width, 3),
                        weights <- None,
                        pooling <- 'max',
                        classes <- num_classes,
                        dropout <- 0.05)

  model.compile(
    optimizer = 'adam',
    loss = SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'])
  RETURN model
ENDFUNCTION
FUNCTION train(train_pipeline):
  img_height <- 224
  img_width <- 224
  batch_size <- train_pipeline['batch_size']
  SET model TO model_config(img_height, img_width, num_classes)
  epochs=train_pipeline['epochs']
  history <- model.fit(train_ds,
                      validation_data=valid_ds,
                      epochs=epochs,
                      callbacks= my_callbacks())

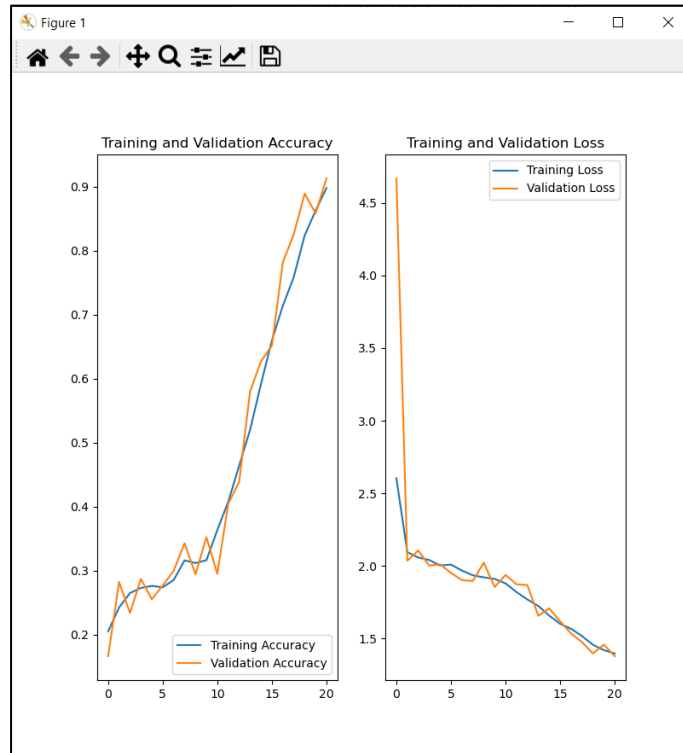
```

Modul Program 4.3 Lanjutan Modul Program Proses *training*.

```
Console 11/A
-----
630/630 [=====] - 221s 351ms/step - loss: 2.0469 - accuracy: 0.2988 - val_loss: 2.0017 -
val_accuracy: 0.2873
Epoch 5/21
630/630 [=====] - 221s 351ms/step - loss: 2.0105 - accuracy: 0.2905 - val_loss: 2.0106 -
val_accuracy: 0.2556
Epoch 6/21
630/630 [=====] - 222s 353ms/step - loss: 2.0192 - accuracy: 0.2887 - val_loss: 1.9513 -
val_accuracy: 0.2770
Epoch 7/21
630/630 [=====] - 221s 351ms/step - loss: 1.9733 - accuracy: 0.3023 - val_loss: 1.9021 -
val_accuracy: 0.3000
Epoch 8/21
630/630 [=====] - 221s 351ms/step - loss: 1.9540 - accuracy: 0.3140 - val_loss: 1.8959 -
val_accuracy: 0.3429
Epoch 9/21
630/630 [=====] - 222s 353ms/step - loss: 1.9257 - accuracy: 0.3353 - val_loss: 2.0225 -
val_accuracy: 0.2944
Epoch 10/21
630/630 [=====] - 222s 353ms/step - loss: 1.9184 - accuracy: 0.3314 - val_loss: 1.8544 -
val_accuracy: 0.3524
Epoch 11/21
630/630 [=====] - 221s 351ms/step - loss: 1.8760 - accuracy: 0.3816 - val_loss: 1.9376 -
val_accuracy: 0.2952
Epoch 12/21
630/630 [=====] - 222s 352ms/step - loss: 1.8380 - accuracy: 0.3931 - val_loss: 1.8732 -
val_accuracy: 0.4048
Epoch 13/21
630/630 [=====] - 222s 353ms/step - loss: 1.7881 - accuracy: 0.4620 - val_loss: 1.8670 -
val_accuracy: 0.4389
Epoch 14/21
630/630 [=====] - 221s 351ms/step - loss: 1.7438 - accuracy: 0.5097 - val_loss: 1.6563 -
val_accuracy: 0.5802
Epoch 15/21
630/630 [=====] - 221s 351ms/step - loss: 1.6746 - accuracy: 0.5706 - val_loss: 1.7074 -
val_accuracy: 0.6270
Epoch 16/21
630/630 [=====] - 222s 352ms/step - loss: 1.6106 - accuracy: 0.6463 - val_loss: 1.6176 -
val_accuracy: 0.6516
Epoch 17/21
630/630 [=====] - 221s 351ms/step - loss: 1.5834 - accuracy: 0.6943 - val_loss: 1.5345 -
val_accuracy: 0.7817
Epoch 18/21
630/630 [=====] - 223s 353ms/step - loss: 1.5323 - accuracy: 0.7494 - val_loss: 1.4748 -
val_accuracy: 0.8254
Epoch 19/21
630/630 [=====] - 221s 351ms/step - loss: 1.4628 - accuracy: 0.8024 - val_loss: 1.3964 -
val_accuracy: 0.8889
Epoch 20/21
630/630 [=====] - 222s 352ms/step - loss: 1.4252 - accuracy: 0.8542 - val_loss: 1.4579 -
val_accuracy: 0.8579
Epoch 21/21
630/630 [=====] - 222s 352ms/step - loss: 1.4092 - accuracy: 0.8876 - val_loss: 1.3780 -
val_accuracy: 0.9127
```

Gambar 4.4. Proses Data *Training*.

Setelah proses data *training* selesai dilakukan, akan muncul sebuah *window* yang berisikan 2 buah grafik pada gambar 4.4., yaitu grafik akurasi data *training* dan data validasi serta grafik *loss* pada *training* dan validasi. Grafik tersebut berguna sebagai evaluasi hasil *training*, apakah sudah sesuai dengan tujuan atau belum.



Gambar 4.5. Grafik Hasil proses *training*.

4. Tombol *Testing*

Tombol ini berfungsi untuk memulai proses *testing* model yang sudah di-*training* yang dapat dilihat pada gambar 4.6. Proses *testing* dapat dilihat pada modul program 4.4.

```

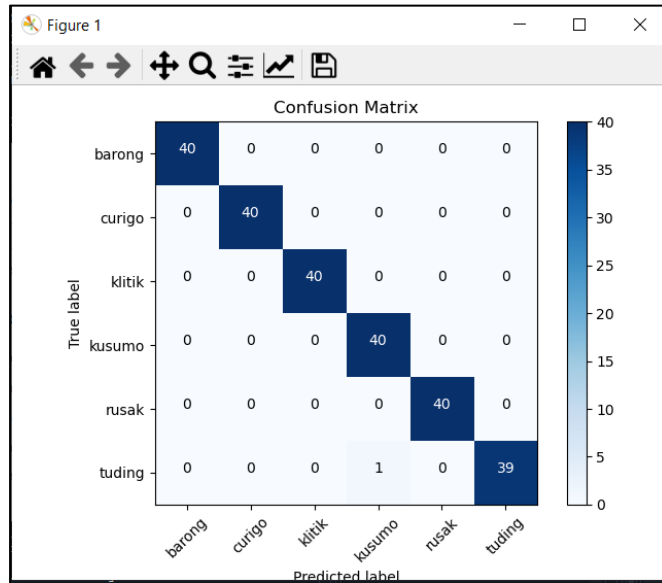
FUNCTION test_model :
  GET directory
  SET classes TO directory
  SET img_height TO 224
  SET img_width TO 224
  SET flow_from_directory TO directory,
                                target_size=(img_height, img_width),
                                classes=classes,
                                batch_size=1,
                                shuffle=TRUE

  SET test_batches TO flow_from_directory
  SET model_predict TO x=test_batches, verbose=1
  SET predictions TO model_predict
  CALCULATE predictions
  SET cm TO confusion_matrix(y_true=classes, y_pred=predictions)
  SET plot TO cm, interpolation='nearest', cmap
  IF normalize THEN
    SET m TO cm / cm[:,newaxis]
  ENDIF
  OUTPUT cm
  SET thresh TO cm.max / 2.
  FOR each row on thresh THEN
    SET plot TO (j, i, cm[i, j],
  SHOW plot

```

Modul Program 4.4 Proses *Testing*.

Output dari proses ini adalah matriks *confusion* berukuran sesuai dengan kelas yang terdapat pada dataset, yaitu 6 kelas. Matriks tersebut berisi persebaran hasil pengujian model dalam mengklasifikasikan citra batik motif parang.



Gambar 4.6. Tampilan *Confusion Matrix* hasil pengujian.

5. Tombol *Export Model*

Tombol ini bertugas untuk memanggil *method export_model*. Pada *method* ini model yang telah dilatih akan disimpan dalam direktori yang telah dipilih sebelumnya, sekaligus akan meng-*export* model kedalam model yang berekstensi file *.tflite* supaya dapat dibaca oleh aplikasi *android*. Proses *export_model* terdapat pada modul 4.5.

```

FUNCTION export_model :
  GET saved_model_dir
  GET classes
  GET trained_model

  GET tflite_converter

  SAVE trained_model TO saved_model_dir

  COMPUTE trained_model WITH tflite_converter AS converted_model

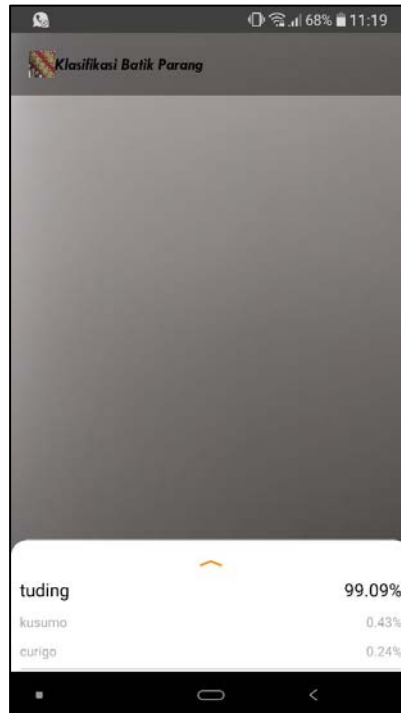
  SAVE converted_model TO saved_model_dir

```

Modul Program 4.5 Proses *export_model*.

4.1.2. *Prototype Aplikasi Android*

Pada *prototype* aplikasi *android* untuk klasifikasi batik motif parang terdapat halaman kamera untuk melakukan klasifikasi terhadap batik secara *realtime*. Pada halaman ini terdapat 3 prediksi berdasarkan nilai akurasi klasifikasi paling tinggi terhadap model yang ada dalam *prototype*. Halaman Klasifikasi dapat dilihat pada gambar 4.7.



Gambar 4.7. Halaman Klasifikasi Batik pada *Prototype* aplikasi android.

Halaman klasifikasi terhubung dengan modul *classifier* yang berfungsi untuk memproses model .tflite sehingga dapat melakukan klasifikasi berdasarkan hasil pelatihan yang ada dalam model tersebut. Modul *classifier* dapat dilihat pada modul program 4.10.

```

FUNCTION Classifier :
  GET tfliteModel, numThreads, labels
  GET imgSizeY, imgSizeX, dimPixelSize, numBytesPerChannel

  SET tfliteOptions.setNumThreads TO numThreads

  CALCULATE (imgSizeY*imgSizeX*dimPixelSize*numBytesPerChannel) AS data

  DEFINE byteBuffer <- data
  DEFINE Interpreter <- tfliteModel, tfliteOptions
  DEFINE imgData <- ByteBuffer

  COMPUTE imgData BY ByteOrder
  PRINT "Created a Tensorflow Lite Image Classifier."
ENDFUNCTION

```

Modul Program 4.6 Modul *Classifier*.

Setelah model diproses oleh modul classifier, kemudian dapat dilakukan pemindaian gambar melalui kamera. Gambar yang diambil melalui kamera kemudian diproses dalam *method recognizeImage* pada modul program 4.6 berikut.

```

FUNCTION recognizeImage :
  START "preprocessBitmap" :
    GET bitmap
    DEFINE convertBitmapToByteBuffer <- bitmap
  END

```

Modul Program 4.7 Method *recognizeImage*.

```

START "runInference" :
    GET systemClock
    COMPUTE "runInference"
    SET endTime, startTime
END
PRINT "Timecost to run model inference:" endTime - startTime
START "compareImage" :
    GET recognitionLHS, recognitionRHS
    DEFINE confidence <- CALCULATE recognitionLHS, recognitionRHS
END
DEFINE result
GET labels_size, labels, normalizedProbability
SET i <- 0
FOR i < labels_size :
    IF labels_size > i :
        BUMP result <- labels(i)
        COMPUTE normalizedProbability(i)
    ENDIF
ENDFOR
DEFINE recognitionArrayList <- List, recognitionSize <- int
SET recognitionsSize TO (CALCULATE max(result)), i <- 0
FOR i < recognitionsSize :
    BUMP recognitions <- result
ENDFOR
RETURN recognitions
ENDFUNCTION

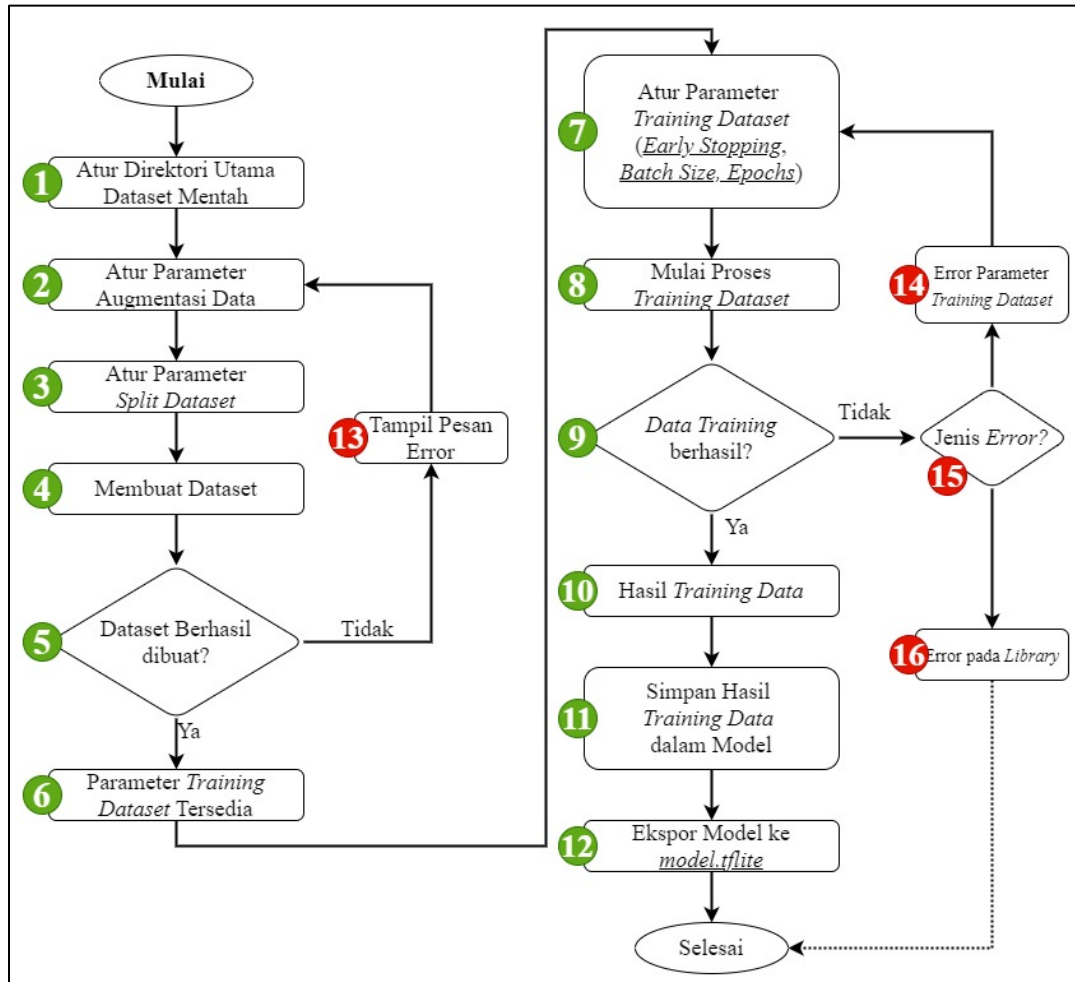
```

Modul Program 4.8 *Method recognizeImage.*

4.2. Pengujian Aplikasi

4.2.1. Pengujian Aplikasi *Desktop* untuk Pembuatan Model

Pada pengujian ini digunakan metode pengujian *White-Box* dimana pengujian dilakukan pada bagian *input-output* program apakah sudah sesuai dengan perancangan atau belum. Gambar 4.8 berikut adalah diagram alir pengujian *white-box* yang sudah dilakukan.



Gambar 4.8. Diagram Pengujian *White-Box* pada Aplikasi *Desktop* Pembuatan Model.

Proses pengujian dimulai dengan mengatur bagian input pertama yaitu variabel berjenis *string* dengan nama *dataset_dir*. Variabel tersebut digunakan untuk proses pembuatan dataset yang disertai augmentasi data dan *split dataset* sebagai data lokasi tempat dataset terletak. Proses kedua adalah input parameter augmentasi data yang ada pada kotak augmentasi data. Proses ketiga adalah input parameter *split dataset*. Proses keempat adalah membuat dataset. Selanjutnya pada proses keenam adalah pengecekan apakah proses pembuatan dataset berhasil atau gagal. Jika gagal, akan menampilkan pesan error. Error yang biasa terjadi adalah nama data citra yang sama dengan data yang telah ada, tidak dibuat opsi *overwrite* pada pembuatan data untuk menghindari data yang ada dihapus. Jika berhasil, akan berlanjut pada proses ketujuh yaitu membuka opsi untuk memilih parameter *training data*.

Setelah mengatur parameter pada proses kedelapan, akan dilakukan proses *training data*. Selanjutnya dilakukan pengecekan apakah proses *training data* berhasil atau tidak. Terdapat 2 jenis error pada proses ini. Jenis error pertama adalah kesalahan dalam mengatur *batch size* yang terlalu besar dibandingkan kemampuan *hardware* yang digunakan. Jenis error yang kedua adalah error pada library, jika terjadi akan langsung menghentikan proses program. Proses kesepuluh adalah menampilkan hasil *training data* dalam bentuk grafik. Proses kesebelas adalah menyimpan model yang sudah dilatih. Proses keduabelas adalah

mengekspor model yang sudah dilatih menjadi *model.tflite*. Setelah selesai mengekspor, seluruh proses yang ada dalam aplikasi sudah dilakukan.

4.2.2. Pengujian *Prototype Aplikasi Android*

Pengujian dilakukan menggunakan 15 data citra batik motif parang yang diambil di beberapa butik yang menjual batik di daerah Malioboro, Yogyakarta. Terdapat skenario pengujian dengan beberapa pembatasan, antara lain :

1. Gambar yang diklasifikasikan hanya gambar batik motif parang (*parang barong, parang curigo, parang klithik, parang kusumo, parang rusak, dan parang tuding*)
2. Jarak pengambilan gambar antara 20-40cm
3. Pengambilan gambar dilakukan disiang hari untuk mendapatkan cahaya yang optimal.

Adapun parameter yang digunakan dalam pengujian *prototype aplikasi android* antara lain :

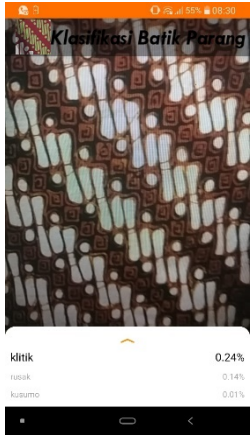
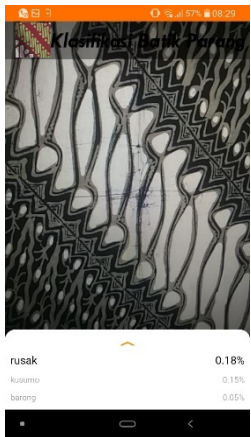
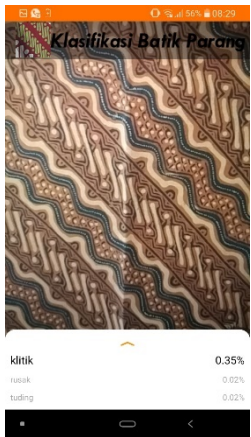
1. Tingkat kesesuaian hasil motif batik yang diambil gambarnya dengan prediksi klasifikasi motif batik dari model yang telah dibuat.
2. Kualitas pengambilan gambar yang dilakukan ketika pengujian (intensitas cahaya, jenis kamera pada *smartphone*)

Berdasarkan skenario serta parameter pengujian yang telah ditentukan, didapat hasil pengujian seperti pada tabel 4.1 berikut ini.




Tabel 4.1 Tabel Pengujian *Prototype Aplikasi Android*.

No	Citra Batik motif Parang	Motif Batik	Hasil Klasifikasi
1		Parang klithik	Parang klithik

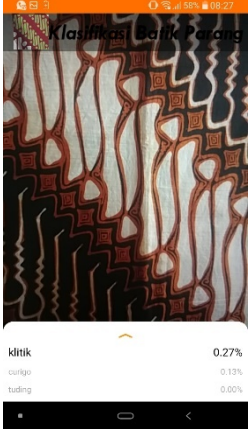

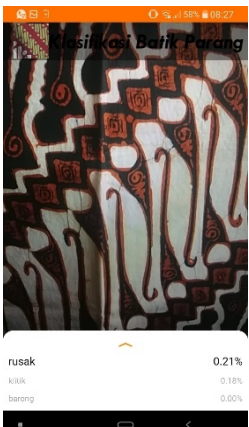
Tabel 4.2 Lanjutan tabel Pengujian *Prototype* Aplikasi *Android*.

2		Parang klithik	Parang klithik
3		Parang barong	Parang rusak
4		Parang klithik	Parang klithik

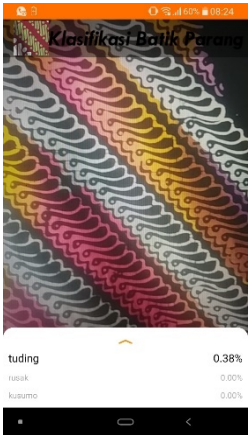

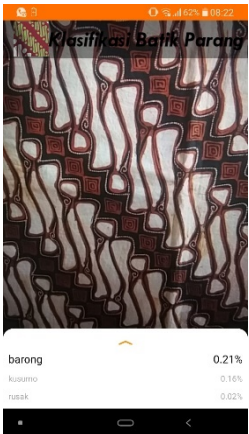
Tabel 4.3 Lanjutan tabel Pengujian *Prototype* Aplikasi *Android*.

No	Citra Batik motif Parang	Motif Batik	Hasil Klasifikasi
5		Parang klithik	Parang klithik
6		Parang curigo	Parang curigo
7		Parang klithik	Parang klithik

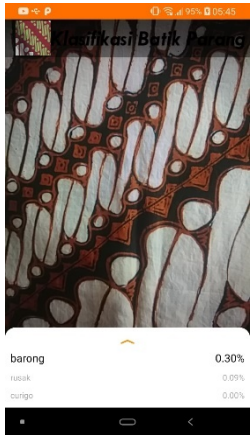

Tabel 4.4 Lanjutan tabel Pengujian *Prototype* Aplikasi *Android*.

No	Citra Batik motif Parang	Motif Batik	Hasil Klasifikasi
8		Parang rusak	Parang klitik
9		Parang rusak	Parang barang
10		Parang rusak	Parang rusak

Tabel 4.5 Lanjutan tabel Pengujian *Prototype* Aplikasi *Android*.

No	Citra Batik motif Parang	Motif Batik	Hasil Klasifikasi
11		Parang tuding	Parang tuding
12		Parang barong, Parang rusak	Parang barong
13		Parang rusak	Parang barong

Tabel 4.6 Lanjutan tabel Pengujian *Prototype* Aplikasi *Android*.

No	Citra Batik motif Parang	Motif Batik	Hasil Klasifikasi
14		Parang rusak, Parang barong	Parang barong
15		Parang kusumo	Parang kusumo

Tabel 4.7 Lanjutan tabel Pengujian *Prototype* Aplikasi *Android*.

Berdasarkan data hasil pengujian , perhitungan akurasi :

$$\frac{11}{15} \times 100\% = 73,33\%$$

Akurasi yang didapatkan sebesar dengan rincian 11 data klasifikasi benar dan data yang salah sebanyak 4 data. Citra batik tersebut tidak dapat diklasifikasikan oleh prototype aplikasi android karena noise dan pencahayaan yang kurang ketika pengambilan gambar sehingga beberapa bagian corak pada batik motif parang tidak dapat terlihat jelas.

4.2.3. Perbandingan Hasil Pengujian Model dan *Prototype* Aplikasi *Android*

Jika dibandingkan antara hasil pengujian model dengan hasil pengujian *prototype* aplikasi *android*, terdapat perbedaan yang cukup signifikan pada akurasi dalam mengklasifikasikan batik motif parang, yaitu 99,58% akurasi pada hasil pengujian model dan 73,33% akurasi pada hasil pengujian *prototype* aplikasi *android*. Hal tersebut disebabkan oleh perbedaan sumber input citra, dimana pada pengujian model data citra yang digunakan adalah data gambar yang sudah difoto, sedangkan pada aplikasi android menggunakan data realtime dilapangan. Selain itu terdapat celah antara model yang dihasilkan oleh aplikasi desktop dengan model yang sudah diekspor menjadi .tflite.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil analisis, perancangan serta pengujian yang telah dilakukan dalam *convolutional neural network* klasifikasi batik tulis motif parang menggunakan *mobilenet* berbasis android, dapat diambil kesimpulan sebagai berikut.

1. *Convolutional Neural Network* dapat mengenali pola pada batik motif parang dengan baik, dengan tingkat akurasi sebesar 99,58% dari 240 data uji pada aplikasi *desktop* dan akurasi sebesar 73,33% dari 15 data uji pada *prototype* aplikasi *android*.
2. Penggunaan metode *data augmentation* dapat mencegah terjadinya *over-fitting* pada model, serta dapat membantu meningkatkan tingkat akurasi pada model karena memperbanyak variasi data untuk proses *training*.
3. Kekurangan pada penelitian ini terdapat pada proses ekspor ke model *.tflite* untuk aplikasi android, dimana banyak terjadi *loss* sehingga tingkat akurasi berkurang secara signifikan.

5.2. Saran

Adapun saran-saran yang dapat digunakan untuk mengembangkan penelitian ini.

1. Pengembangan dapat dilakukan dengan membuat *method* khusus untuk ekspor model ke *.tflite* sehingga dapat mengurangi perbedaan akurasi yang signifikan dibanding model hasil *training*.
2. Penelitian selanjutnya dapat dilakukan dengan batik selain motif parang.
3. Dalam pembuatan *dataset*, harus memperhatikan klasifikasi batik agar tidak terjadi kesalahan pada hasil akhir klasifikasi.

DAFTAR PUSTAKA

- , *Kamus Besar Bahasa Indonesia*. [Online]. Tersedia di kbbi.kemdikbud.go.id/entri/batik. Diakses 11 Juli 2019
- Achmad, F. D. (2018). *The Influence Of Different Preprocessing Techniques For Classifying Batik Using Convolutional Neural Networks*.
- Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., ... Zheng, X. (2015). *TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems*. <http://download.tensorflow.org/paper/whitepaper2015.pdf>
- Alsing, O. (2018). *Mobile Object Detection using TensorFlow Lite and Transfer Learning*.
- Deng, L., & Yu, D. (2013). *Deep Learning : Methods and Applications*.
- Dewangga S, H. (2018). *Batik Pattern Classification Using Gray Level Co-occurrence Matrix and Modified K-Nearest Neighbor*.
- Fausset, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall, Inc.
- Gifary, S. (2015). INTENSITAS PENGGUNAAN SMARTPHONE TERHADAP PERILAKU KOMUNIKASI. *Jurnal Sosioteknologi*, 14(2).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>
- Hasan, R. V. (2012). Studi Komparasi Motif Batik Parang Rusak Barong. *Fakultas Sastra Universitas Jember*, 2(1), 71–79.
- Howard, A. G., Wang, W., Zhu, M., Weyand, T., Chen, B., Andreetto, M., Kalenichenko, D., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.
- Ismanto, E., Novalia, M., & Herlandy, P. B. (2017). Pemanfaatan Smartphone Android Sebagai Media Pembelajaran Bagi Guru Sma Negeri 2 Kota Pekanbaru. *Jurnal Pengabdian UntukMu NegeRI*, 1(1), 42–47. <https://doi.org/10.37859/jpumri.v1i1.33>
- Kasim, A. A., & Harjoko, A. (2014). Klasifikasi Citra Batik Menggunakan Jaringan Syaraf Tiruan Berdasarkan Gray Level Co- Occurrence Matrices (GLCM). *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Yogyakarta, 21 Juni 2014*, 7–13.
- Koeswadji, K. (1981). *Mengenal Seni Batik di Yogyakarta*. Proyek Pengembangan Permuseuman.
- Kusumanto, R. D., & Tomponu, A. N. (2011). *Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB*. 2011(Semantik).
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). *Deep Learning*. <https://doi.org/10.1038/nature14539>
- Murwiyati, & Lauren, G. (2013). *Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia Untuk Anak Sekolah Dasar Berbasis Android*. 12.
- Naufal, M. A. (2017). *Implementasi Metode Klasifikasi K-Nearest Neighbor (K-Nn) Untuk Pengenalan Pola Batik Motif Lampung*. 1–46.
- Ogedebe, P. M., & Jacob, B. P. (2012). *Software Prototyping : A Strategy to Use When User Lacks Data Processing Experience*. 2(6), 219–224.
- Purnomo, M. H., & Muntasa, A. (2010). Konsep pengolahan citra digital dan ekstraksi fitur. *Yogyakarta: Graha Ilmu*, 277.
- Putri, A. R. (2016). *Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya*. 1, 1–6.
- Ratnasari, M. C. D. (2018). *DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENGENALAN OBJEK MENGGUNAKAN MXNET (Studi*

Kasus : Data Citra Motif Batik Keraton dan Pesisir).

- Rawat, W., & Wang, Z. (2017). *Deep Convolutional Neural Networks for Image Classification : A Comprehensive Review*. October. <https://doi.org/10.1162/NECO>
- Samsi, S. S. (2011). *Teknik dan Ragam hias Batik Yogya & Solo* (A. Diaz (ed.)). Yayasan Titian Masa Depan (Titian Foundation).
- Sutiyati, E. (2016). *Nilai Filosofi Motif Parang Rusak Gurdo Dalam Tari Bedhaya Harjuna Wiwaha*. 1–15.
- Sutoyo, T. d, Mulyanto, E., & Suhartono, V. (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: Andi.
- Wardani, M. F. K. (2018). *Pengenalan Motif Batik Lampung Menggunakan Deteksi Tepi Canny dan Cross Power Spectrum*. 2, 58. [https://www.uam.es/gruposinv/meva/publicaciones_jesus/capitulos_espanyol_jesus/2005_motivacion para el aprendizaje Perspectiva alumnos.pdf%0Ahttps://www.researchgate.net/profile/Juan_Aparicio7/publication/253571379_Los_estudios_sobre_el_cambio_conceptual_](https://www.uam.es/gruposinv/meva/publicaciones_jesus/capitulos_espanyol_jesus/2005_motivacion_para_el_aprendizaje_Perspectiva_alumnos.pdf%0Ahttps://www.researchgate.net/profile/Juan_Aparicio7/publication/253571379_Los_estudios_sobre_el_cambio_conceptual_)
- Wijayanto, H. (2015). Klasifikasi Batik Menggunakan Metode K-Nearest Neighbour Berdasarkan Gray Level Co-Occurrence Matrices (GLCM). *Jurusan Teknik Informatika FIK UDINUS*, 5, 1–7. <https://doi.org/10.1109/ISIT.2018.8437785>
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1), 818–833. https://doi.org/10.1007/978-3-319-10590-1_53
- Riyanto, Didik. (1995). *Proses Batik, Batik Tulis-Batik Cap-Batik Printing (dari awal persiapan bahan dan alat mendesain corak sampai finishing)*. Solo: CV. Aneka.

LAMPIRAN

