

## 2. LANDASAN TEORI

Bab ini dijelaskan beberapa teori tentang batik sebagai objek penelitian, proses *training* untuk mengenali pola batik dengan menggunakan *Convolutional Neural Network – Residual Network*, dan Augmentasi data.

### 2.1. Tinjauan Pustaka

#### 2.1.1. Batik

Motif batik yang akan digunakan pada skripsi ini terdiri dari 2 kategori. Yaitu motif geometri dan motif non geometri.

Motif batik geometri adalah motif yang memiliki ornamen dasar yang tersusun oleh bentuk geometri seperti persegi, layang-layang, atau lingkaran. Contoh 5 motif batik geometri yang juga akan digunakan pada skripsi ini yaitu Ceplok, Kawung, Lereng, Nitik, dan Parang. Gambar kelima motif ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 Contoh berbagai motif batik geometris

Motif ceplok memiliki desain yang didasarkan pada bentuk bintang, mawar, atau bentuk lainnya. Motif kawung merupakan motif yang menyerupai buah kawung (sejenis kelapa atau aren) selain itu juga kadang ditafsirkan menyerupai bunga teratai lotus. Kemudian motif lereng dan parang, kedua motif ini sangat mirip. Persamaan keduanya adalah pada umumnya memiliki motif parang (menyerupai huruf 'S') dan memanjang secara diagonal. Namun pola parang selalu mengandung ragam hias mlinjon sedangkan lereng tidak selalu. Sedangkan motif lereng terdapat garis diagonal yang tidak lebar dan dipenuhi oleh pola-pola kecil. Terakhir adalah motif nitik yang dapat dikenali melalui lingkaran maupun

ceplokannya yang tersusun atas banyak garis halus, balok kecil, atau segi empat (Paluseri, 2018).

Kemudian ada motif batik non geometri, yaitu motif batik yang memiliki susunan bentuk pola yang tidak teratur. Adapun pola hias yang biasanya ada di motif batik non geometri adalah yang tersusun dari ornamen tumbuhan, candi, binatang dan lainnya. 2 contoh batik non geometri yang juga akan digunakan sebagai objek pada skripsi ini adalah lung-lungan dan semen, dapat dilihat pada Gambar 2.2.



Batik Lung-lungan

Batik semen

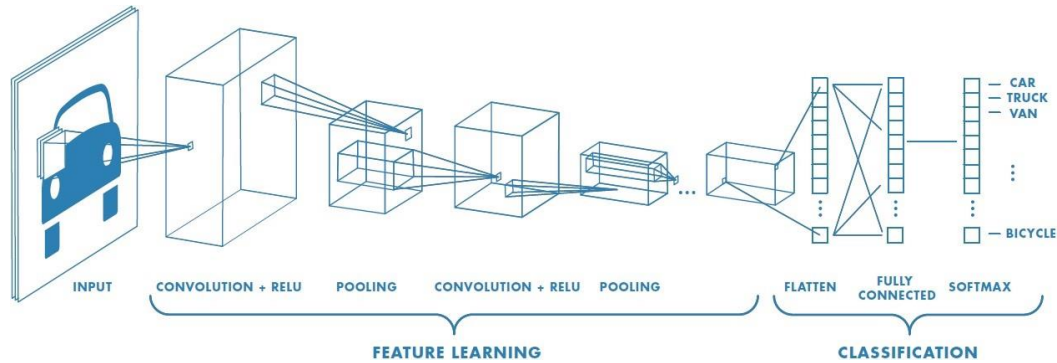
Gambar 2.2 Motif batik non-geometris Lung-Lungan dan Semen

Motif batik lung-lungan memiliki ornamen tumbuhan sebagai ornamen utamanya. Karena Lung-lungan sendiri memiliki arti tunas atau kuncup yang menjalar. Sedangkan motif semen memiliki 3 ornamen. Yaitu ornamen yang berhubungan dengan daratan seperti binatang berkaki empat atau tumbuhan, lalu ornamen kedua yaitu yang berhubungan dengan udara seperti burung garuda atau awan. Kemudian ornamen ketiga berhubungan dengan laut yaitu hewan – hewan air seperti ikan, katak, atau ular.

### 2.1.2. Convolutional Neural Network (CNN)

*Convolutional Network* atau *Convolutional Neural Network* adalah *neural network* spesialisasi untuk memroses data yang memiliki topologi *grid*. Contohnya seperti data *time-series* yang merupakan grid 1 dimensi, sedangkan data gambar merupakan grid 2 dimensi dari piksel-piksel yang terkandung di dalamnya (Goodfellow, et al, 2017).

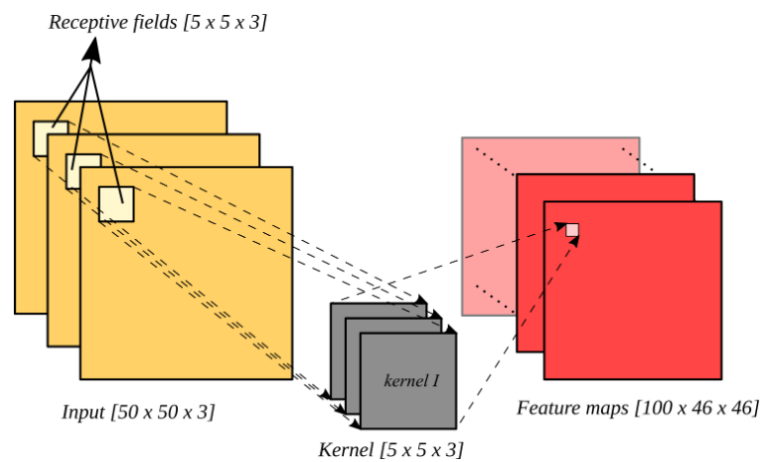
Pada umumnya, CNN memiliki 3 *layer* utama dalam arsitekturnya yaitu, *Convolutional*, *Pooling*, dan *Fully-Connected* (Li, F. F. et al. 2019) dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ilustrasi arsitektur Convolutional Neural Network

Sumber: Mathworks. url: <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html> (diakses pada 16 Desember 2019)

Pada layer *convolutional* memiliki beberapa neuron yang tersusun sehingga membentuk sebuah filter. Filter ini akan dioperasikan dengan menggunakan *dot product* terhadap sebagian area pada input gambar (Gambar 2.4) yang disebut dengan *receptive field*.



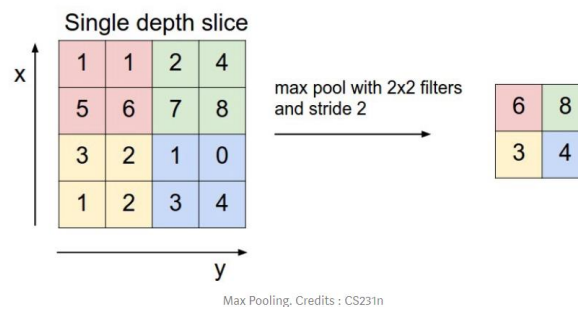
Gambar 2.4 Ilustrasi *Receptive Field*

Sumber: stackexchange, url: <https://ai.stackexchange.com/questions/8701/what-is-the-difference-between-a-receptive-field-and-a-feature-map> (diakses pada 16 Desember 2019)

sehingga akan menghasilkan output atau *activation map* atau *feature map*. Jumlah *feature map* yang dihasilkan adalah sebanyak jumlah filter yang digunakan.

Sebelumnya, filter ini akan diinisialisasi dengan nilai-nilai tertentu, dan ketika tahap *learning* filter merupakan salah satu parameter yang akan di-*update*. Dimensi dari *feature map* yang dihasilkan bergantung pada variabel  $W$  (ukuran Input),  $N$  (ukuran filter),  $P$  (*zero padding*), dan  $S$  (*Stride*)

Kemudian lanjut ke *pooling layer*. *Pooling layer* biasanya berada diantara *convolutional layer* dengan tujuan untuk mereduksi parameter dari *feature map* agar komputasi bisa berjalan semakin cepat tanpa mengorbankan informasi penting yang terkandung dalamnya. Dan tujuan mendasar lainnya adalah untuk mengurangi kemungkinan *overfitting*. Operasi *pooling* yang populer digunakan adalah *max-pooling*. Ilustrasi dari proses *pooling* ini dapat dilihat pada Gambar 2.5.



Gambar 2.5 Proses *Max-Pooling*

Sumber: mc.ai, url: <https://mc.ai/deep-learning-klasifikasi-image-dengan-convolutional-neural-network-cnn-menggunakan-keras-di/> (diakses pada 16 Desember 2019)

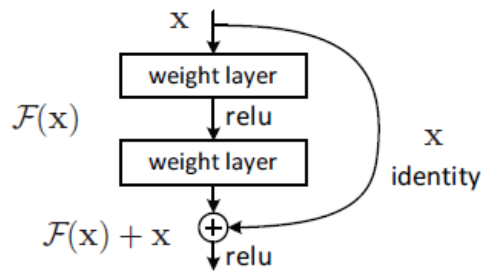
*Feature map* yang dihasilkan masih dalam bentuk array multidimensi sehingga akan dilakukan *flattening* kedalam bentuk vektor sebagai input untuk *Fully Connected Layer* yang kemudian selanjutnya akan ada beberapa *hidden layer*, *activation function*, *output layer*, dan *loss function*.

### 2.1.3. Convolutional Neural Network Residual Network (CNN ResNet)

*Deep Residual Network* atau *Residual Network* (ResNet) dikembangkan oleh Kaiming He. Melalui *paper* pada 2016 yang berjudul “Deep Residual learning for image recognition” ResNet adalah pemenang dari *ILSVRC 2015 Competition* yang mampu mengungguli arsitektur VGG, GoogleNet, dan lainnya (He, K., et al.

2016). Ide utama dibalik pengembangan model Resnet adalah untuk mengatasi fenomena *vanishing gradient descent* yang terjadi pada *deep CNN*.

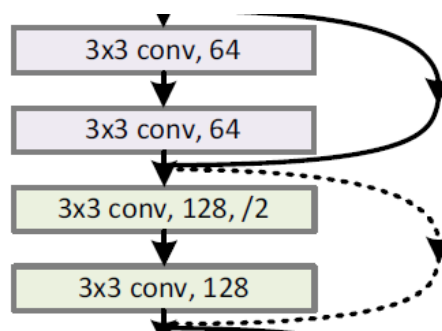
ResNet memodifikasi *network* dengan adanya *identity connection*. Identity connection yang membentangi dari input layer hingga output layer disebut sebagai *residual block* (Gambar 2.6).



Gambar 2.6 Ilustrasi *Residual Block*

Sumber: He, K., et al. 2016

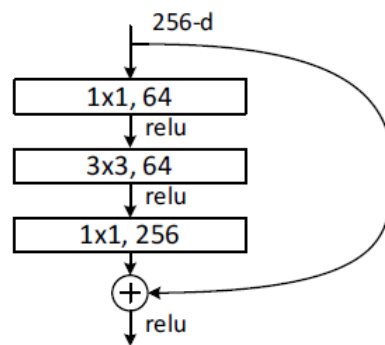
Resnet memiliki 2 jenis *connection* pada *residual block* berdasarkan ukuran dimensi *input* dan *output*. Pertama, adalah *identity shortcut connection* (Gambar 2.7, diilustrasikan dengan garis hitam), digunakan apabila *input* dan *output* memiliki dimensi yang sama. Kedua, apabila dimensi *output* lebih besar dari *input*, maka akan ada 2 pilihan *connection* yaitu menggunakan metode *zero-padding* atau *projection shortcut connection*. Adapun metode *zero-padding connection* tidak merupakan *connection* yang tidak memiliki parameter. Sedangkan *projection shortcut connection* memiliki parameter karena menerapkan *convolution layer* ukuran 1 x 1.



Gambar 2.7 Ilustrasi *shortcut connection*

Sumber: He, K., et al. 2016

Selain susunan layer pada blok seperti pada Gambar 2.8, Resnet juga mengimplementasi arsitektur *bottleneck* pada bloknya. Resnet-34 yang menggantikan bloknya menjadi *bottleneck* seperti Gambar 2.10 menghasilkan model Resnet-50. Desain *bottleneck* ini memiliki 3 *layers* pada sebuah *block* seperti yang digambarkan pada Gambar 2.10. *Kernel filter* berukuran  $1 \times 1$  berperan untuk mengurangi dan mengembalikan dimensi ke semula, diantaranya diterapkan *kernel filter* berukuran  $3 \times 3$  *convolution layer*. Desain *bottleneck* membuat *Convolution Layer*  $3 \times 3$  dapat mengerjakan bagiannya dengan tidak memakan *resource* yang besar.



Gambar 2.8 Desain Bottleneck

Sumber: He, K., et al. 2016

Berikut rangkuman penjelasan setiap arsitektur Resnet berdasarkan jumlah *layer* pada Gambar 2.9. Dapat dilihat bahwa Resnet-18 dan Resnet-34 tidak mengimplementasikan *bottleneck design*.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

Gambar 2.9 Arsitektur Beberapa Resnet dengan variasi layer

Sumber: He, K., et al. 2016

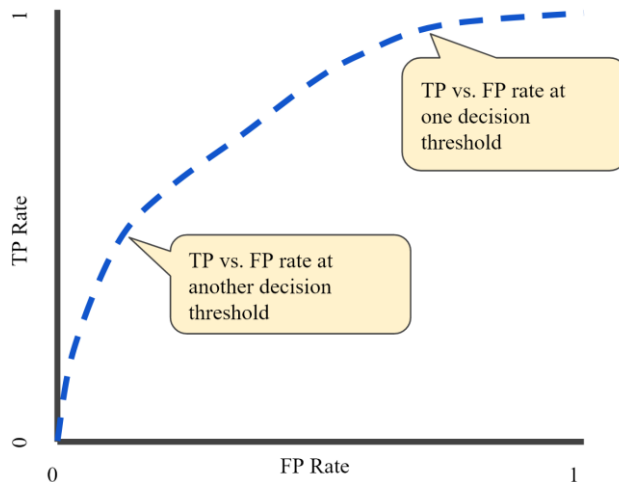
#### 2.1.4. Augmentasi Data

*Deep convolutional neural network* bergantung kepada big data untuk menghindari *overfitting*, namun big data belum tentu mudah untuk didapatkan. Sehingga *data augmentation* hadir sebagai solusi untuk meningkatkan ukuran dan kualitas dari *training dataset* yang terbatas (Shorten & Khoshgoftaar, 2019). Khususnya untuk tipe data berupa gambar ada beberapa metode augmentasi yang dapat dipertimbangkan.

Augmentasi rotasi dapat dilakukan berapapun nilai derajat dengan arah rotasi searah jarum jam atau kebalikannya. Kemudian ada metode *flipping* yang pada umumnya diterapkan pada sumbu x (*flipping horizontally*), metode ini telah terbukti berpengaruh efektif pada *dataset* seperti CIFAR-10, namun tidak memberikan hasil yang baik pada dataset MINST karena *flipping* menghasilkan informasi yang berbeda dengan data asli pada MINST (Cubuk, E. D., et al. 2019). Kemudian adalah metode Random erasing bekerja dengan cara melakukan seleksi secara acak pada bagian pada gambar lalu menghapus nilai pikselnya dengan nilai acak. *Random erasing* dapat diaplikasikan untuk semua arsitektur CNN, *parameter learning free*, dan dapat dikombinasikan dengan teknik *data augmentation* lainnya (Zhong. Z., et al. 2017). Dan yang terakhir adalah skala atau zoom yang merupakan salah satu metode untuk menyerupai data-data seperti asli, yang dimana memiliki variasi zoom yang berbeda-beda.

#### 2.1.5. ROC – AUC

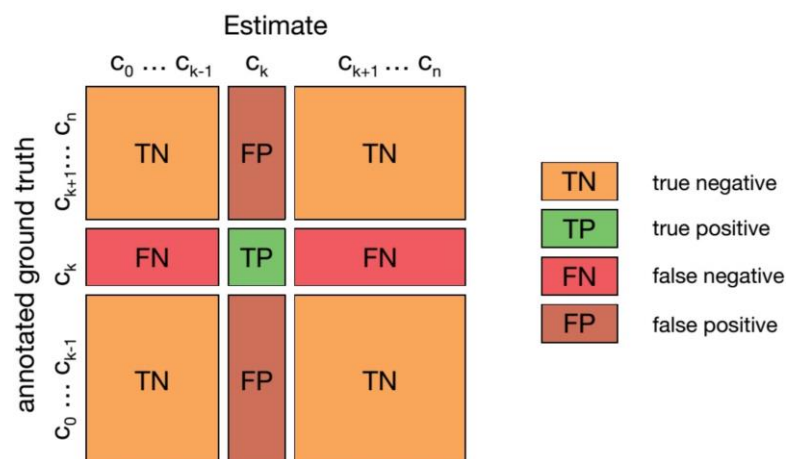
ROC merupakan singkatan dari *Receiver Operating Characteristics*. Sedangkan AUC adalah singkatan dari *Area Under Curve*. AUC merupakan nilai luas atau integral dari kurva ROC. ROC dan AUC merupakan salah satu alat ukur untuk menilai performa keyakinan dari model ketika melakukan klasifikasi.



Gambar 2.10 Kurva ROC

Sumber: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc> (diakses pada 4 juli 2020)

Kurva ROC adalah kumpulan titik dari nilai TP dan FP yang diplotkan pada berbagai nilai *threshold* (Bradley, 1997). Diilustrasikan melalui Gambar 2.10. TP adalah singkatan dari *True Positive* yang merupakan hasil dari sebuah keadaan dimana motif ceplok pada data asli juga diklasifikasikan sebagai motif ceplok oleh model. sedangkan FP adalah singkatan dari *False Positive Rate* yang adalah hasil dari sebuah keadaan dimana model mengklasifikasikan motif lain (selain ceplok) sebagai motif ceplok. Ilustrasi tentang FP dan TP pada *multiclass classification* dapat dilihat melalui Gambar 2.11.



Gambar 2.11 *Confusion matrix* multiclass classification

Sumber: <https://devopedia.org/confusion-matrix> (diakses pada 4 juli 2020)

Adapun *threshold* adalah sebuah batas dari nilai yang ditentukan secara manual untuk dijadikan sebagai batas ketika dilakukan konversi ke hasil kategorikal. Contohnya adalah, apabila *threshold* yang ditetapkan adalah 0,6. Kemudian apabila motif Ceplok memiliki nilai probabilitas 0,4 sedangkan motif parang memiliki nilai 0,6, maka model akan memberikan *output* prediksi adalah motif parang.

Salah satu yang menjadi keunggulan AUC untuk digunakan sebagai alat ukur evaluasi adalah sifat *scale invariant* yakni, AUC mengukur seberapa baik persebaran ranking dari hasil prediksi dibandingkan hanya melihat nilai absolut dari suatu hasil prediksi. Ini akan relevan dengan data yang akan digunakan pada skripsi ini dimana, pada beberapa gambar batik memiliki lebih dari 1 motif atau kelas atau label.

## 2.2. Tinjauan Studi

Pada tahun 2015, Nurhaida, I., et al. melakukan pengenalan motif batik dengan menggunakan metode dari bidang *image processing* yaitu SIFT. Permasalahan yang dihadapi oleh peneliti adalah karakteristik motif batik yang berulang. Hal ini menyebabkan SIFT kurang mampu untuk mengenali motif batik dengan baik. Sehingga, peneliti menambahkan algoritma *Hough Transform* dan hasil yang didapatkan jauh lebih baik yakni dari nilai *Equal Error Rate* sebesar 26,60% menjadi 8,47%.

Karimah, F. U., & Harjoko, A. (2017) mengerjakan pengklasifikasian untuk motif batik kain besurek menggunakan metode dari bidang *image processing* yaitu *Speed Up Robust Features* (SURF) dan *Gray Level Co-occurrence Matrix* (GLCM) untuk digunakan sebagai *feature extractor* kemudian dilanjutkan dengan menggunakan metode k-Nearest Neighbour (kNN) untuk melakukan klasifikasi motif batik. Model yang dibangun menerapkan kombinasi weight 0,1 pada GLCM dan 0,9 pada SURF. Kemudian untuk kNN ditetapkan nilai k=3. Hasil akurasi yang didapatkan adalah sebesar 95,47%. Model yang dikembangkan memanfaatkan 5 kelas yang terdiri dari 5 motif batik Besurek yaitu, Kaligrafi, Raflesia, Relung Paku, Burung Kuau, dan Rembulan. Dan 1 kelas yang digolongkan sebagai batik non Besurek.

Penggabungan metode *image processing* dengan dari *machine learning* dilakukan juga oleh Minarno, A. E., Maulani, A. S., Kurniawardhani, A.,

Bimantoro, F., & Suciati, N. (2018). Penelitian ini dilakukan untuk membandingkan performa kombinasi MTH-SVM dan MTH-kNN pada 300 gambar batik dengan pembagiannya adalah 50 kelas batik (tanpa label nama motif). Hasil yang mereka dapatkan adalah MTH-SVM memiliki tingkat akurasi sebesar 76%. Sedangkan MTH-kNN memiliki akurasi optimum pada nilai 82%.

Gultom, Y., Arymurthy, A. M., & Masikome, R. J. (2018) menggunakan CNN model VGG16 untuk menyelesaikan masalah dari penelitian ini yaitu motif batik yang sangat beragam. Dan CNN VGG16 ini mampu mengatasi dilema keberagaman motif batik yang ada dengan mendapat akurasi sebesar 89%. Tetapi akurasi yang dihasilkan masih tidak lebih baik dibanding metode SIFT apabila data gambar yang digunakan memiliki rotasi dan skala(perbesaran) yang berbeda-beda.

Permasalahan yang muncul pada penelitian oleh Gultom, Y. (2018) menjadi fokus permasalahan utama untuk penelitian yang dilakukan oleh Agastya, I. M. A., & Setyanto, A. (2018). Masalahnya yaitu CNN VGG16 tidak efektif ketika menghadapi data gambar yang memiliki rotasi dan skala yang bervariasi. Sehingga peneliti melakukan *data augmentation* pada proses *training*. Hasil akurasi akhir yang didapatkan adalah sebesar 89,3% dengan standar deviasi 2%.

Rangkuman dari keempat penelitian terkait dan penjelasan terkait perbedaan dengan skripsi yang akan dilakukan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Rangkuman penelitian terkait beserta perbedaan dengan skripsi yang akan dilakukan

Nama Peneliti	Tahun	Masalah	Metode	Hasil	Perbedaan dengan skripsi yang akan dilakukan
Nurhaida, I., Noviyanto, A., Manurung, R., & Arymurthy, A. M.	2015	Metode SIFT kurang mampu mengenali motif batik yang berulang	SIFT dengan Menambahkan algoritma <i>Hough Transform</i>	Nilai <i>Equal Error Rate</i> lebih baik yakni sebesar 8,47%	Skripsi ini akan menerapkan metode CNN Resnet

Nama Peneliti	Tahun	Masalah	Metode	Hasil	Perbedaan dengan skripsi yang akan dilakukan
Karimah, F. U., & Harjoko, A.	2017	Mengklasifikasi motif batik kain besurek	Menggunakan SURF dan GLCM sebagai <i>feature extractor</i> . Dan k-Nearest Neighbour (kNN) untuk melakukan pengklasifikasian	Mendapatkan akurasi sebesar 95,47%	Skripsi ini akan menggunakan metode CNN ResNet tanpa melibatkan metode <i>image processing</i> (seperti SURF atau GLCM). Dan batik yang akan diteliti di skripsi ini motif kawung, parang, ceplik, nitik, lereng, lung-lungan dan semen
Minarno, A. E., Maulani, A. S., Kurniawardhani, A., Bimantoro, F., & Suciati, N.	2018	Peneliti ingin membandingkan performa metode <i>image processing</i> (MTH) yang digabung dengan metode dari <i>machine learning</i> (SVM dan kNN)	<i>Multi Texton Histogram</i> (MTH) yang dipadukan dengan <i>Support Vector Machine</i> (SVM) dan <i>k-Nearest Neighbour</i> (kNN)	MTH-SVM memiliki tingkat akurasi sebesar 76%. Sedangkan MTH-kNN memiliki akurasi optimum pada nilai 82%.	Dataset yang digunakan memiliki 50 kelas batik tanpa label nama motifnya. Sedangkan skripsi ini memiliki 5 kelas beserta label untuk motif batik geometri dan non-geometri
Gultom, Y., Arymurthy, A. M., & Masikome, R. J.	2018	CNN dilema ketika menghadapi tingkat keberagaman motif batik yang tinggi	Menggunakan CNN model VGG16	Akurasi sebesar 89%	Selain motif geometri, skripsi ini akan menambah motif non-geometri sebagai objek penelitian juga

Nama Peneliti	Tahun	Masalah	Metode	Hasil	Perbedaan dengan skripsi yang akan dilakukan
Agastya, I. M. A., & Setyanto, A.	2018	CNN VGG16 tidak dapat mengenali pola batik secara efektif ketika dihadapkan dengan gambar batik yang bervariasi dari segi skala dan rotasi	melakukan training CNN dengan memasukkan gambar hasil <i>data augmentation</i> juga	Akurasi sebesar 89,3% dengan standar deviasi sebesar 2,8%	Selain skala dan rotas, skripsi ini akan menerapkan <i>flipping</i> , <i>random erasing</i> , dan implementasi <i>patchshuffle regularization</i> pada CNN