TUGAS AKHIR - IF184802

DETEKSI POSISI TUMPAL PADA CITRA KAIN BATIK DENGAN MENGGUNAKAN FITUR *LOCAL BINARY PATTERN* (LBP)

RAFIDAH RAFALIA
NRP 05111540000139

Dosen Pembimbing
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.
Dini Adni Navastara S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]

TUGAS AKHIR – IF184802

DETEKSI POSISI TUMPAL PADA CITRA KAIN BATIK DENGAN MENGGUNAKAN FITUR *LOCAL BINARY PATTERN* (LBP)

RAFIDAH RAFALIA
NRP 5114100075

Dosen Pembimbing
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.
Dini Adni Navastara S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

RBTC

[Halaman ini sengaja dikosongkan]

FINAL PROJECT – IF184802

TUMPAL POSITION DETECTION IN BATIK IMAGE USING *LOCAL BINARY PATTERN* (LBP) FEATURES

RAFIDAH RAFALIA
NRP 5114100075

Advisor
Dr.Eng. Nanik Suciati, S.Kom., M.Kom
Dini Adni Navastara S.Kom., ., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

RBTC

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

DETEKSI POSISI TUMPAL PADA CITRA KAIN BATIK DENGAN MENGGUNAKAN FITUR *LOCAL BINARY PATTERN* (LBP)

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Teknik Informatika
Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:
RAFIDAH RAFALIA
NRP: 05111540000139

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr.Eng. Nanik Suciati, S.Kom., M.Kom. NIP: 19710428 199412 2 001	..... (pembimbing 1)
Dini Adni Navastara S.Kom., M.Sc. NIP: 19851017 201504 2 001	..... (pembimbing 2)

SURABAYA
JULI 2019

RBIC

[Halaman ini sengaja dikosongkan]

RBTC

DETEKSI POSISI TUMPAL PADA CITRA KAIN BATIK DENGAN MENGGUNAKAN FITUR *LOCAL BINARY PATTERN* (LBP)

Nama Mahasiswa	: Rafidah Rafalia
NRP	: 05111540000139
Departemen	: Informatika FTIK-ITS
Dosen Pembimbing 1	: Nanik Suciati, S.Kom., M.Kom., Dr.Eng
Dosen Pembimbing 2	: Dini Adni Navastara S.Kom.,

ABSTRAK

Tumpal adalah corak yang beda diantara corak utama pada sebuah batik. Dalam pencetakkan busana dengan menggunakan kain batik, pentingnya keserasian pola dalam pembuatan busana. Keserasian ini didapatkan dengan mengidentifikasi letak tumpal pada kain batik yang akan digunakan dalam pencetakkan busana. Maka perlunya dibuat sistem pencetakkan busana yang dapat mendeteksi lokasi tumpal pada citra kain batik.

Pada tugas akhir ini akan dibangun sebuah sistem yang bertujuan untuk mendeteksi lokasi tumpal pada batik. Dengan melakukan ekstraksi fitur LBP pada citra kain batik dan melakukan klasifikasi menggunakan metode KNN, sistem akan dapat mengenali perbedaan pola pada citra kain batik. Sistem akan menerima input berupa citra kain batik dan output berupa citra kain batik yang telah ditandai letak posisi tumpalnya.

Penulis melakukan pengujian akurasi dengan mengganti parameter k dan jumlah data training yang digunakan untuk mendapatkan hasil yang paling optimal, serta mencari data uji citra yang susah untuk terdeteksi tumpalnya. Berdasarkan hasil uji coba, penulis menemukan bahwa nilai k pada klasifikasi KNN bernilai 1 memberikann hasil yang paling optimal, dan jumlah

data trainng 200 memberikan hasil setelah optimalisasi terbaik. Penulis juga mendapatkan kesimpulan bahwa citra batik yang memiliki lebih dari satu pola pada area tumpal akan lebih susah untuk diidentifkasi area tumpalnya agar tepat 100%. Hal ini dikarenakan pada pengambilan data training sistem biasanya hanya akan mengambil satu pola pada tumpal citra uji coba.

***Kata kunci: Batik, Tumpal Batik, Aplikasi Pembuatan Busana Batik, Local Binary Pattern* (LBP).**

RBTC

TUMPAL POSITION DETECTION IN BATIK IMAGE USING *LOCAL BINARY PATTERN* (LBP) FEATURES

Nama Mahasiswa	: Rafidah Rafalia
NRP	: 05111540000139
Department	: Informatics FTIK-ITS
Supervisor 1	: Nanik Suciati, S.Kom., M.Kom., Dr.Eng
Supervisor 2	: Dini Adni Navastara S.Kom.,

ABSTRACT

Tumpal is a pattern that is different between the main features of batik. In printing clothes using batik, the pattern harmony in fashion making is very important. This harmony is obtained by identifying the location of tumpal in batik that will be used in printing clothes. Which is why a clothing printing system that can detect the location of tumpal in batik is much needed.

In this final project, a system that aims to detect the location of tumpal on batik will be built. By extracting LBP features in the image of batik cloth and classifying using the KNN method, the system will be able to recognize differences in patterns in the image of batik cloth. The system will receive input in the form of a batik cloth image and output in the form of an image of a batik cloth that has been marked with the location of the tumpal position.

. The author conducts accuracy testing by changing the parameters k and the amount of training data used to get the most optimal results, and looking for image test data that is difficult to detect the spill. Based on the results of the trial, the authors found that the value of k in the KNN classification of value 1 gave the most optimal results, and the number of data

trainings 200 gave the best results after optimization. The author also gets the conclusion that batik images that have more than one pattern in the area of the tumpal will be more difficult to identify to be 100% precise. This is because the retrieval of system training data will usually only take one pattern on the trial image.

***Keywords: Batik, Tumpal Batik, Batik Fashion Printing Applications, Local Binary Pattern* (LBP).**

RBTC

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wata'ala karena atas karunia, nikmat dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

DETEKSI POSISI TUMPAL PADA CITRA KAIN BATIK DENGAN MENGGUNAKAN FITUR *LOCAL BINARY PATTERN* (LBP)

Sholawat serta salam semoga tak lupa kita sampaikan kepada Rasulullah Shallallahu 'Alaihi Wassalam, yang telah membawa risalah Islam dan Qur'an kepada seluruh umat manusia.

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT serta junjungan Nabi Muhammad SAW, karena limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Informatika ITS.
2. Orang tua dan keluarga yang senantiasa mendoakan, memotivasi dan mendukung lahir maupun batin penulis dalam menyelesaikan tanggung jawab ini.
3. Ibu Nanik Suciati, S.Kom., M.Kom., Dr.Eng. dan Ibu Dini Adni Navastara S.Kom., selaku dosen pembimbing I dan II penulis, yang senantiasa membimbing, memberikan saran, arahan, serta bantuan-bantuan lainnya dalam menyelesaikan tugas akhir ini.
4. Seluruh staf dan karyawan Departemen Informatika ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Teman-teman TC 2015 yang banyak membantu penulis dalam segala hal, yang berjuang bersama-sama penulis dari awal masuk perkuliahan hingga akhir masa perkuliahan ini.

6. Teman-teman anggota kelompok Tugas Akhir, Davin, Nirmala dan Vivi yang banyak membantu dan menyusahkan penulis dalam banyak hal.
7. Serta pihak-pihak lain yang mohon maaf tidak dapat disebutkan satu per satu, yang telah turut andil dalam membantu penulis selama perkuliahan.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyelesaikan tugas akhir ini. Namun, penulis mohon maaf apabila terdapat kekurangan ataupun kesalahan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan untuk ke depannya.

Surabaya, Juli 2019

RBTC

DAFTAR ISI

LEMBAR PENGESAHAN vii
ABSTRAK x
ABSTRACT xii
KATA PENGANTAR xiv
DAFTAR ISI xvi
DAFTAR GAMBAR xix
DAFTAR KODE SUMBER.....xxvi
DAFTAR TABEL xxviiiii
DAFTAR PERSAMAAN xxviiiix
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Permasalahan 2
1.3. Batasan Permasalahan 2
1.4. Tujuan 3
1.5. Manfaat 3
1.6. Metodologi 3
1.7. Sistematika Penulisan 4
BAB II TINJAUAN PUSTAKA 7
2.1. Batik 7
2.2. Deteksi Tepi Laplacian 7
2.3. *Local Binary Pattern (LBP)* 8
2.4. *Euclidean Distance* 9
2.5. K-Nearest Neighbor (KNN) 10
2.6. Processing 10
BAB III ANALISIS DAN PERANCANGAN SISTEM 13
3.1. Perancangan Data 13
3.2. Desain Umum Sistem 13
3.2.1 Preprocessing 15
3.2.2 Deteksi Awal Tumpal dan Non Tumpal 16
3.2.3 Identifikasi Pola Subcitra 18
3.2.4 Optimalisasi dan Penandaan 19
BAB IV IMPLEMENTASI SISTEM 21
4.1. Lingkungan Implementasi 21

4.2. Implementasi Praproses Data 21
4.3. Implementasi Tahap Deteksi Awal Tumpal dan Non Tumpal 23
4.3.1. Crop Gambar Menjadi Beberapa Potongan 23
4.3.2. Ekstraksi Fitur LBP.....23
4.3.3. Tentukan Area Tumpal 25
4.4. Implementasi Tahap Identifikasi Pola Subcitra 30
4.4.1. Pengambilan Data Training.....30
4.4.2. Pemotongan dan Klasifikasi KNN 32
4.5. Implementasi Tahap Optimalisasi dan Penandaan 36
4.5.1. Identifikasi Kelas Pola Perbaris 30
4.5.2. Hitung Deretan Tumpal Terbanyak.....32
4.5.3. Penandaan 30
BAB V UJI COBA DAN EVALUASI 21
5.1. Lingkungan Uji Coba.....41
5.2. Dataset 41
5.3. Pengujian Sistem.....42
5.3.1. Uji Koefisien Matriks dan Kecepatan Data 1 43
5.3.2. Uji Koefisien Matriks dan Kecepatan Data 2 499
5.3.3. Uji Koefisien Matriks dan Kecepatan Data 3 566
5.3.4. Uji Koefisien Matriks dan Kecepatan Data 4 62
5.3.5. Uji Koefisien Matriks dan Kecepatan Data 5 699
5.3.6. Uji Koefisien Matriks dan Kecepatan Data 6 75
5.3.7. Uji Koefisien Matriks dan Kecepatan Data 7 82
5.3.8. Uji Koefisien Matriks dan Kecepatan Data 8 88
5.4. Pengujian Sistem.....95
BAB VI KESIMPULAN DAN SARAN 957
6.1. Kesimpulan 977
6.2. Saran 988
LAMPIRAN.....99
L.1. Dataset Uji Coba 100
L.2. Hasil Uji Coba Data 1 100
L.3. Hasil Uji Coba Data 2 101
L.4. Hasil Uji Coba Data 3 102
L.5. Hasil Uji Coba Data 4 102

L.6. Hasil Uji Coba Data 5 103
L.7. Hasil Uji Coba Data 6 104
L.8. Hasil Uji Coba Data 7 104
L.9. Hasil Uji Coba Data 8 105
DAFTAR PUSTAKA 107
BIODATA PENULIS 109

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR GAMBAR

Gambar 2.1 Contoh Jenis Motif Batik 7
Gambar 2.2 Ilustrasi Proses LBP 9
Gambar 2.3 Penggeseran Kombinasi Nilai Biner Untuk Mencari Kombinasi Nilai Terkecil 9
Gambar 3.1 Contoh Citra Batik 13
Gambar 3.2 Diagram Alir Sistem yang Dibangun 134
Gambar 3.3 Diagram Alir Preprocessing 135
Gambar 3.4 Hasil Tahap Preprocessing 136
Gambar 3.5 Proses Pemotongan dan Penetapan Referensi 136
Gambar 3.6 Diagram Alir Deteksi Awal Tumpal dan Non Tumpal 17
Gambar 3.7 Diagram Alir Identifikasi Pola Subcitra 18
Gambar 3.8 Hasil Tahap Identifikasi Pola Subcitra 20
Gambar 3.9 Diagram Alir Optimalisasi dan Penandaan 20
Gambar 3.10 Hasil Tahap Optimalisasi dan Penandaan 20
Gambar 5.1 Contoh Citra Dataset 42
Gambar 5.2 Citra Data 1 43
Gambar 5.3 Output Uji Coba Data 1 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 44
Gambar 5.4 Output Uji Coba Data 1 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 44
Gambar 5.5 Output Uji Coba Data 1 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 45
Gambar 5.6 Output Uji Coba Data 1 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 46
Gambar 5.7 Output Uji Coba Data 1 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 46
Gambar 5.8 Output Uji Coba Data 1 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 47
Gambar 5.9 Output Uji Coba Data 1 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 48
Gambar 5.10 Output Uji Coba Data 1 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi 48

Gambar 5.11 Output Uji Coba Data 1 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....49
Gambar 5.12 Citra Data 2.....49
Gambar 5.13 Output Uji Coba Data 2 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....51
Gambar 5.14 Output Uji Coba Data 2 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....51
Gambar 5.15 Output Uji Coba Data 2 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....51
Gambar 5.16 Output Uji Coba Data 2 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....53
Gambar 5.17 Output Uji Coba Data 2 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....53
Gambar 5.18 Output Uji Coba Data 2 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....53
Gambar 5.19 Output Uji Coba Data 2 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....55
Gambar 5.20 Output Uji Coba Data 2 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....55
Gambar 5.21 Output Uji Coba Data 2 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....55
Gambar 5.22 Citra Data 3.....56
Gambar 5.23 Output Uji Coba Data 3 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....57
Gambar 5.24 Output Uji Coba Data 3 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....57
Gambar 5.25 Output Uji Coba Data 3 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....58
Gambar 5.26 Output Uji Coba Data 3 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....59
Gambar 5.27 Output Uji Coba Data 3 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....59
Gambar 5.28 Output Uji Coba Data 3 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....60
Gambar 5.29 Output Uji Coba Data 3 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....61

Gambar 5.30 Output Uji Coba Data 3 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 61
Gambar 5.31 Output Uji Coba Data 3 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 62
Gambar 5.32 Citra Data 4..... 62
Gambar 5.33 Output Uji Coba Data 4 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 64
Gambar 5.34 Output Uji Coba Data 4 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 64
Gambar 5.35 Output Uji Coba Data 4 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 64
Gambar 5.36 Output Uji Coba Data 4 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 66
Gambar 5.37 Output Uji Coba Data 4 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 66
Gambar 5.38 Output Uji Coba Data 4 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 66
Gambar 5.39 Output Uji Coba Data 4 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 68
Gambar 5.40 Output Uji Coba Data 4 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 68
Gambar 5.41 Output Uji Coba Data 4 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 68
Gambar 5.42 Citra Data 5..... 69
Gambar 5.43 Output Uji Coba Data 5 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 70
Gambar 5.44 Output Uji Coba Data 5 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 70
Gambar 5.45 Output Uji Coba Data 5 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 71
Gambar 5.46 Output Uji Coba Data 5 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 72
Gambar 5.47 Output Uji Coba Data 5 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 72
Gambar 5.48 Output Uji Coba Data 5 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi.... 73

Gambar 5.49 Output Uji Coba Data 5 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....74
Gambar 5.50 Output Uji Coba Data 5 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....74
Gambar 5.51 Output Uji Coba Data 5 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....75
Gambar 5.52 Citra Data 6.....75
Gambar 5.53 Output Uji Coba Data 6 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....77
Gambar 5.54 Output Uji Coba Data 6 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....77
Gambar 5.55 Output Uji Coba Data 6 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....77
Gambar 5.56 Output Uji Coba Data 6 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....79
Gambar 5.57 Output Uji Coba Data 6 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....79
Gambar 5.58 Output Uji Coba Data 6 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....79
Gambar 5.59 Output Uji Coba Data 6 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....81
Gambar 5.60 Output Uji Coba Data 6 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....81
Gambar 5.61 Output Uji Coba Data 6 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....81
Gambar 5.62 Citra Data 7.....82
Gambar 5.63 Output Uji Coba Data 7 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....83
Gambar 5.64 Output Uji Coba Data 7 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....83
Gambar 5.65 Output Uji Coba Data 7 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....84
Gambar 5.66 Output Uji Coba Data 7 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....85
Gambar 5.67 Output Uji Coba Data 7 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....85

Gambar 5.68 Output Uji Coba Data 7 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....85
Gambar 5.69 Output Uji Coba Data 7 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....86
Gambar 5.70 Output Uji Coba Data 7 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....87
Gambar 5.72 Output Uji Coba Data 7 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....87
Gambar 5.73 Citra Data 8.....88
Gambar 5.74 Output Uji Coba Data 8 k=1 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....90
Gambar 5.75 Output Uji Coba Data 8 k=3 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....90
Gambar 5.76 Output Uji Coba Data 8 k=5 Training 150. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....90
Gambar 5.77 Output Uji Coba Data 8 k=1 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....92
Gambar 5.78 Output Uji Coba Data 8 k=3 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....92
Gambar 5.79 Output Uji Coba Data 8 k=5 Training 200. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....92
Gambar 5.80 Output Uji Coba Data 8 k=1 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....94
Gambar 5.81 Output Uji Coba Data 8 k=3 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....94
Gambar 5.82 Output Uji Coba Data 8 k=5 Training 250. (a) Sebelum Optimlisasi. (b) Setelah Optimalisasi....94

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR KODE SUMBER

Kode Sumber 4.1 Resize, Grayscale dan Penajaman Gambar .. 21
Kode Sumber 4.2 Crop Gambar Menjadi Potongan 23
Kode Sumber 4.3 Nilai Piksel LBP 24
Kode Sumber 4.4 Nilai Output LBP 24
Kode Sumber 4.5 Mencari Nilai Potongan 25
Kode Sumber 4.6 Mencari Area Tumpal 28
Kode Sumber 4.7 Pengambilan Data 30
Kode Sumber 4.8 Implementasi Tahap Klasifikasi 33
Kode Sumber 4.9 Implementasi Menentukan Kelas Motif 36
Kode Sumber 4.10 Implementasi Hitung Deretan Tumpal Terbanyak 38
Kode Sumber 4.11 Implementasi Penandaan 39

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR TABEL

Tabel 5.1 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 1 Training 150 43
Tabel 5.2 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 1 Training 200 45
Tabel 5.3 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 1 Training 250 47
Tabel 5.4 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 2 Training 150 50
Tabel 5.5 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 2 Training 200 52
Tabel 5.6 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 2 Training 250 54
Tabel 5.7 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 3 Training 150 56
Tabel 5.8 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 3 Training 200 58
Tabel 5.9 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 3 Training 250 60
Tabel 5.10 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 4 Training 150 63
Tabel 5.11 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 4 Training 200 65
Tabel 5.12 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 4 Training 250 67
Tabel 5.13 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 5 Training 150 69
Tabel 5.14 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 5 Training 200 71
Tabel 5.15 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 5 Training 250 73
Tabel 5.16 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 6 Training 150 76
Tabel 5.17 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 6 Training 200 78

Tabel 5.18 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 6 Training 250 80
Tabel 5.19 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 7 Training 150 82
Tabel 5.20 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 7 Training 200 84
Tabel 5.21 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 7 Training 250 86
Tabel 5.22 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 8 Training 150 89
Tabel 5.23 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 8 Training 200 91
Tabel 5.24 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data Data 8 Training 250 93
Tabel 5.25 Hasil Rekapitulasi Berbasis *k* 95
Tabel 5.25 Hasil Rekapitulasi Berbasis Jumlah Citra Training .. 95
Tabel 5.25 Hasil Rekapitulasi Gabungan 96

RBTC

DAFTAR PERSAMAAN

(2.1).....8
(2.2).....10

[Halaman ini sengaja dikosongkan]

RBTC

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam bidang pembuatan busana, pola sangat penting artinya. Keserasian pola pada letak posisi tumpal merupakan hal yang sangat penting untuk diperhatikan. Beberapa batik di Indonesia, terutama di pulau Jawa terdapat tumpal sebagai penghias batik. Motif tumpal sering digunakan sebagai hiasan tepi (pinggiran) suatu bidang. Tumpal merupakan susunan/deretan segi tiga sama kaki [1]. Posisi tumpal juga tidah hanya di satu sisi saja, bisa juga di sisi kain panjang, sisi lebar dan di tengah kain. Hingga saat ini, pembuatan busana dengan memperhatikan letak posisi tumpal umumnya masih dilakukan secara manual. Maka dari itu, pada tugas akhir ini dibuat suatu sistem yang dapat mendeteksi posisi tumpal pada buasana kain busana untuk dapat mengembangkan sistem pencetakkan busana yang sudah ada.

Dalam pengenalan pola, segmentasi citra merupakan proses yang pertama kali dilakukan sebelum proses selanjutnya yaitu analisis citra. Fungsi utama dari segmentasi citra adalah membagi citra kedalam bagian-bagian yang mempunyai kesamaan fitur tekstur, warna, bentuk, dan lain sebagainya. Proses segmentasi citra dapat dilakukan dengan banyak metode, salah satunya menggunakan *Local Binary Pattern (LBP)* yang digunakan pada Tugas Akhir ini. LBP adalah metode yang cukup sederhana, namun efisien dalam merepresentasikan ciri tekstur. LBP didasarkan pada konsep tekstur primitif. Pendekatan ini secara teoritis, metodologi komputasi sederhana dan efisien untuk analisis tekstur. LBP menangkap pola mikro pada gambar dengan menghitung perbedaan sebuah piksel dengan piksel tetangganya yang pada akhirnya menghasilkan histrogam untuk dianalisis lebh lanjut lagi. Sifat LBP inilah yang digunakan untuk menentukan posisi tumpal pada kain batik.

Pada tugas akhir ini metode klasifikasi yang digunakan adalah klasifikasi *K-Nearest Neighbor* (KNN). Prinsip kerja KNN adalah mencari jarak terdekat antara data yang dievaluasi dengan *k* tetangga terdekatnya dalam citra latihan. Dengan bantuan metode klasifikasi KNN dicari jarak *Euclidean Distance* yang nantinya jarak tersebut digunakan sebagai penentu kasifikasi pola citra. Metode klasifikasi KNN yang cukup sederhana ini dikiranya sesuai dalam pengembangan sistem yang menggunakan bahasa pemrograman Processing, dikarenakan sedikitnya *library* kecerdasan visual pada bahasa pemrograman tersebut.

1.2 Rumusan Permasalahan

Rumusan masalah dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara mendapatkan letak posisi pola tumpal yang berbeda dengan pola utamanya pada suatu citra?
2 Bagaimana penggunaan ekstraksi fitur *Local Binary Pattern* pada pengenalan citra?
3. Bagaimana mengimplementasikan rancangan sistem dengan menggunakan bahasa pemrograman *Processing*?

1.3. Batasan Permasalahan

Adapun batasan-batasan masalah dalam penulisan tugas akhir ini antara lain:

1. Citra batik yang digunakan adalah citra batik yang memiliki satu buah tumpal berbentuk *horizontal*.
2. Aplikasi dibangun dengan menggunakan *Processing* versi 3.5.
3. Aplikasi dikembangkan untuk *desktop* komputer.

1.4. Tujuan

Tujuan dari tugas akhir ini adalah untuk membuat sistem yang dapat membedakan pola tumpal pada citra kain batik dari

pola utamanya serta mengetahui letak posisi tumpal pada citra kain.

1.5. Manfaat

Dengan adanya penelitian ini diharapkan dapat membantu pengembangan sistem aplikasi pencetakkan busana batik agar dapat mencetak busana dengan memperhatikan letak posisi tumpal pada citra kain batik.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1.6.1. Studi literatur

Pada studi literatur ini dipelajari sejumlah referensi yang relevan terhadap tugas akhir yang dikerjakan, seperti mempelajari referensi *paper* yang diadaptasi oleh tugas akhir ini, mempelajari deteksi citra pola pada busana kain batik dengan bahasa pemrograman *processing* dengan bantuan ekstraksi fitur *Local Binary Pattern* dan metode klasifikasi KNN. Studi literatur ini didapatkan dari buku, internet serta materi-materi kuliah yang berhubungan dengan sistem yang dibangun.

1.6.2. Analisis dan desain sistem

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dilakukan desain sistem dan desain proses-proses yang ada.

1.6.3. Implementasi sistem

Pada tahap ini dilakukan pembangunan sistem. Sistem yang dimaksud disini, yaitu aplikasi deteksi posisi tumpal batik dengan menggunakan fitur ekstraksi *Local Binary Pattern* dan metode

klasifikasi KNN. Perangkat pendeteksi posisis tumpal pada pola busana kain batiki ini dibangun dengan menggunkan bahasa pemrograman *processing.*

1.6.4. Pengujian dan evaluasi

Tahap pengujian dan evaluasi pada perangkat lunak yang telah dibuat bertujuan untuk mengetahui kemampuan algoritma yang dipakai, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin timbul. Parameter yang diujikan adalah parameter k pada klasifikasi KNN dan jumlah data *training* yang diperlukan.

1.6.5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, tinjauan pustaka, metode, implementasi, proses yang telah dilakukan, pengujian, evaluasi dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.7. Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I **Pendahuluan**

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan dan manfaat pembuatan tugas akhir, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II **Tinjauan Pustaka**

Bab ini menjelaskan beberapa pustaka-pustaka yang dijadikan penunjang dan berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir.

Bab III **Analisis dan Perancangan Sistem**

Bab ini membahas mengenai analisis dan perancangan sistem yang dibangun.

Bab IV **Implementasi Sistem**

Bab ini membahas mengenai bagaimana implementasi sistem dari analisis dan desain yang sudah dirancang.

Bab V **Pengujian dan Evaluasi**

Bab ini membahas pengujian dari metode yang ditawarkan dalam tugas akhir untuk mengetahui kesesuaian metode dengan data yang ada.

Bab VI **Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang telah dilakukan. Bab ini juga membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi data atau daftar istilah yang penting pada tugas akhir ini.

[Halaman ini sengaja dikosongkan]

BAB II
TINJAUAN PUSTAKA

Bab ini membahas pustaka/teori-teori yang menjadi dasar dalam pembuatan tugas akhir.

2.1. Batik

Dalam Kamus Besar Bahasa Indonesia, batik dijelaskan sebagai kain bergambar yang dibuat secara khusus dengan menuliskan atau menerakan malam (lilin) pada kain, kemudian pengolahannya diproses dengan cara tertentu. Setiap daerah di Indonesia memiliki ciri khas tertentu pada motif batik dan pakem (cara motif diorganisasi) [2].

Gambar 2.1 Contoh Jenis Motif Batik

2.2. Deteksi Tepi *Laplacian*

Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra. Tujuan operasi pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Salah satu operator deteksi tepi

yang biasa dgunakan adalah opertaor *Laplacian* atau operator Laplace [3].

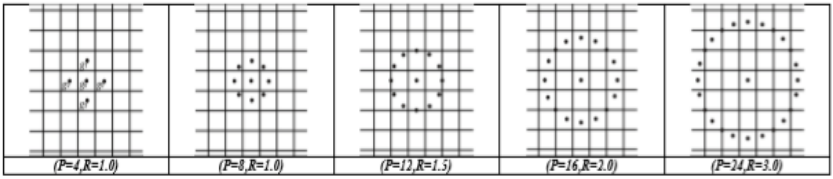
Operator Laplace mendeteksi lokasi tepi lebih akurat khususnya pada tepi yang curam. Pada tepi yang curam, turunan keduanya mempunyai persilangan nol (zero-crossing), yaitu titik di mana terdapat pergantian tanda nilai turunan kedua sedangkan pada tepi yang landai tidak terdapat persilangan nol [3]. Persilangan nol merupakan lokasi tepi yang akurat. Turunan kedua fungsi dengan dua peubah dapat dilihat pada Persamaan (2.1). Terdapat beberapa pendekatan operator laplacian daiantaranya *Four-neighbor laplacian*, *Eight-neighbor laplacian*, *Separeble eight-neighbor laplacian*, dan *Laplacian of Gaussian*.

$$v^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (2.1)$$

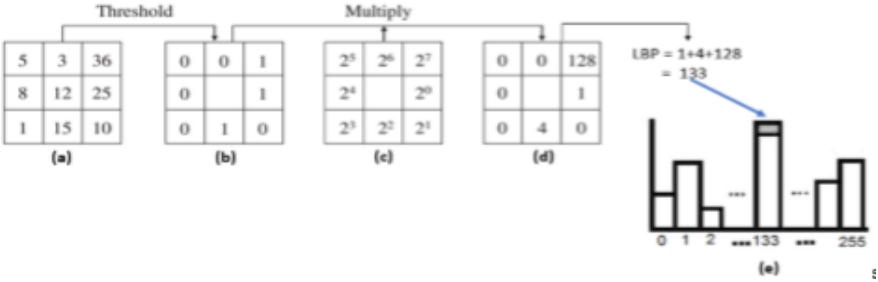
2.3. *Local Binary Pattern* (LBP)

LBP merupakan salah satu metode ekstraksi ciri yang mendeskripsikan tekstur. LBP membandingkan nilai biner piksel pada pusat citra dengan nilai piksel tetangganya [4]. Langkah-langkah LBP dalam mengekstraksi ciri diilustrasikan pada Gambar 1. Pertama, setiap piksel, dihitung nilai LBP dengan membandingkan intensitas piksel antara intensitas piksel pusat dengan intensitas piksel-piksel tetangganya pada radius tertentu, seperti yang ditunjukkan pada Gambar 2. Intensitas piksel pusat menjadi thresholding untuk menyusun nilai LBP di setiap piksel citra.

Bila nilai intensitas piksel pusat lebih besar dibandingkan nilai intensitas piksel tetangganya maka nilai transformasi biner untuk piksel pusat adalah 1. Sebaliknya, bila nilai intensitas piksel pusat lebih kecil dibandingkan nilai intensitas piksel tetangganya maka nilai transformasi biner untuk piksel pusat adalah 0. Susunan nilai biner tersebut dikonversi ke nilai desimal, dengan cara mengalikan nilai biner dengan bobotnya [5].



Gambar 2.2 Ilustrasi Proses LBP



Gambar 2.3 Penggeseran Kombinasi Nilai Biner Untuk Mencari Kombinasi nilai Terkecil LBP

2.4. *K-Nearest Neighbor* (KNN)

KNN merupakan sebuah metode untuk melakukan klasifikasi terhadap objek baruberdasarkan data latih yang jaraknya paling dekat dengan objek tersebut [6]. Tujuan algoritma KNN adalah mengklasifikasikan objek baru berdasarkan atribut dan training sample [7]. Prinsip kerja KNN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (neighbor) terdekatnya dalam pelatihan [8].

Metode KNN banyak digunakan karena mempunyai beberapa kelebihan, diantaranya yaitu dapat menghasilkan data yang lebih akurat dan efektif apabila training data cukup besar [9]. Namun, metode ini juga mempunyai beberapa kekurangan, seperti biaya komputasi yang cukup tinggi karena diperlukan perhitungan jarak query instance pada keseluruhan training sample [9]. Langkah-langkah dalam

implementasi perhitungan metode K-Nearest Neighbor adalah sebagai berikut:

1. Menentukan parameter *k* (jumlah tetangga terdekat).
2. Menghitung kuadrat jarak euclidean (*euclidean distance*) antara data yang akan dievaluasi dengan semua pelatihan.
3. Mengurutkan jarak yang terbentuk (urut naik).
4. Memilih alternative jarak terdekat sebanyak *k*.

2.5. *Euclidean Distance*

Euclidean Distance adalah matriks yang paling sering digunakan untuk menghitung kesamaan dari dua vektor dengan menghasilkan nilai yang berupa jarak dari kedua vektor tersebut. Nilai Euclidean Distance diperoleh dari akar kuadrat selisih 2 vektor yang akan dihitung jaraknya. Untuk menghitung nilai Euclidean Distance dari vektor fitur masukan dan vektor fitur pembanding digunakan Persamaan (2.2) [10].

$$d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \quad (2.2)$$

Keterangan:
d_{ij}= nilai/besaran jarak
n = panjang vektor
x_{ik} = vektor fitur masukan
x_{jk}= vektor fitur pembanding

Semakin kecil nilai dij maka semakin mirip kedua vektor yang dicocokkan. Sebaliknya, semakin besar nilai dij maka semakin berbeda pula kedua vektor yang dicocokkan.

2.6. Processing

Processing adalah sebuah perangkat lunak yang berfungsi dalam pembuatan gambar, animasi, dan interaksi. Perangkat lunak ini juga menawarkan cara untuk belajar pemrograman melalui pembuatan grafik interaktif. *Processing* menggunakan bahasa pemrograman yang mirip dengan bahasa

pemrograman *Java*. *S*intaks bahasa hampir identik, tetapi *Processing* menambahkan fitur khusus yang terkait dengan grafik dan interaksi. Elemen grafis dari *Processing* berhubungan dengan PostScript dan OpenGL.

Processing terdiri dari banyak alat yang dapat bekerjasama dalam kombinasi yang berbeda. *Processing* memiliki lebih dari 100 *library* yang memperluas kemampuan *Processing* dalam pengelolahan suara, visi komputer, dan fabrikasi digital [11].

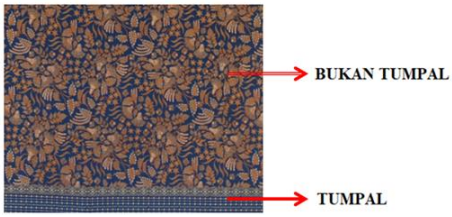
[Halaman ini sengaja dikosongkan]

BAB III
ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan mengenai perancangan sistem pendeteksi posisi tumpal pada pola busana batik menggunakan ekstraksi fitur *Local Binary Pattern* (LBP) dan metode klasifikasi KNN. Pada bab ini pula akan dijelaskan gambaran umum sistem dalam bentuk diagram alir.

3.1. Perancangan Data

Data yang digunakan dalam sistem ini diambil secara manual dengan mengambil foto busana batik yang memiliki tumpal. Dataset adalah citra kain batik yang bertumpal satu dengan bentuk tumpal melebar. Dataset berjumlah delapan dan dapat dilihat di lampiran. Contoh dataset beserta letak tumpal dan non tumpalnya dapat dilihat pada gambar 3.1.



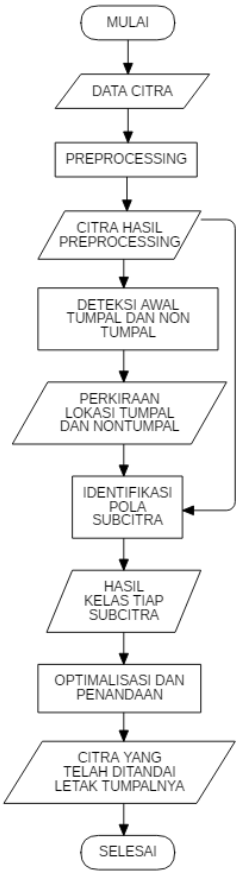
Gambar 3.1 Contoh Citra Batik

3.2. Desain Umum Sistem

Pada tugas akhir ini akan dibuat suatu aplikasi berbasis *Processing* yang bertujuan untuk mengenali posisi tumpal pada kain busana. Aplikasi ini menerima *input* berupa gambar busana batik dalam bentuk *Joint Photographic Grup* (JPG) dan *ouput* berupa lokasi posisi tumpal dengan pemberian tanda pada lokasi tumpal pada gambar pola busana batik.

Pada tahap awal citra uji coba akan melalui tahap *preprocessing*. Pada tahap ini citra akan di-*resize* serta di

konversikan ke *grayscale* dan ditajamkan dengan metode deteksi tepi. Kemudian pada tahap deteksi awal tumpal dan non tumpal citra akan dicari area perkiraan tumpal dan nontumpalnya untuk diambil data *training* pola tumpal dan non tumpal. Data *training* ini akan digunakan sebagai pembanding pada tahap identifikasi subcitra. Subcitra disini merupakan potongan-potongan kecil citra uji dengan ukuran 10x10.

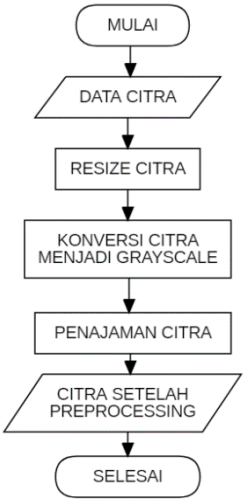


Gambar 3.2 Diagram Alir Sistem yang Dibangun

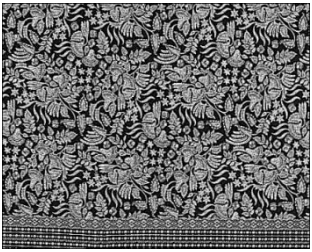
Pada tahap optimalisasi dan penandaan sistem akan mengevaluasi kemungkinan error yang terdapat pada hasil tahap sebelumnya dan membuang kemungkinan hasil error tersebut agar mendapatkan hasil yang lebih optimal. Optimalisasi dilakukan dengan mengimplementasikan batasan masalah pada sistem. Hasil keluaran akhir pada sistem ini berupa citra uji yang telah ditandai lokasi tumpalnya. Diagram alir pada sistem aplikasi yang akan dibangun ditunjukkan pada Gambar 3.2.

3.2.1 Preprocessing

Pada tugas akhir kali ini, praproses data yang dilakukan adalah melakukan perubahan ukuran data gambar menjadi 500x400, kemudian mengubah gambar menjadi *grayscale* dan melakukan penajaman gambar dengan menggunakan metode deteksi tepi *Laplacian* dengan pendekatan operator *Eight-neighbor laplacian*. Diagram alir dari proses ditunjukan pada Gambar 3.3 dan contoh hasil dari tahap *preprocessing* dapat dilihat pada Gambar 3.4.



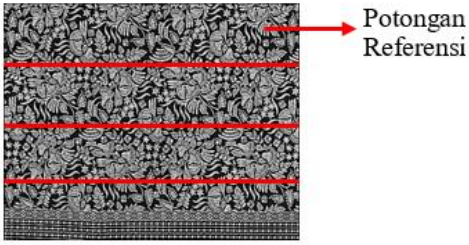
Gambar 3.3 Diagram Alir Preprocessing



Gambar 3.4 Hasil Tahap Preprocessing

3.2.2 Deteksi Awal Tumpal dan Non Tumpal

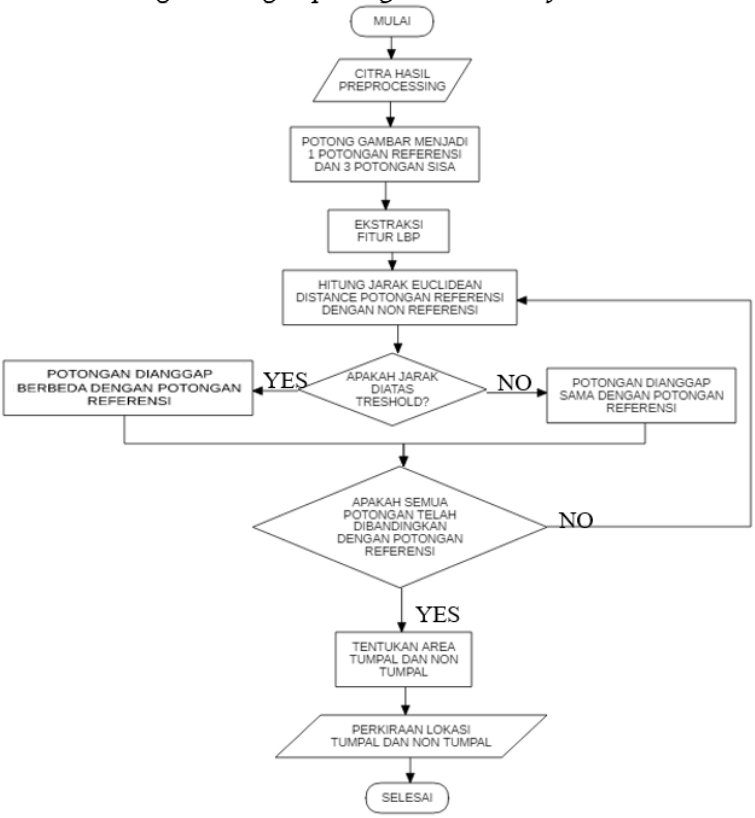
Pada tahap ini citra yang telah melalui tahap sebelumnya akan dibagi menjadi beberapa potongan. Potongan-potongan tersebut kemudian dicari nilai LBP-nya dengan ekstraksi fitur LBP. Potongan citra teratas kemudian akan dijadikan sebagai gambar referensi yang akan diberikan nilai kelas 0. Proses pemotongan dapat dilihat pada Gambar 3.5



Gambar 3.5 Proses Pemotongan dan Penetapan Referensi

Tiap potongan baik yang referensi dan yang bukan referensi diekstraksi fitur LBP-nya untuk dapat dibandingkan. Perbandingan dilakukan dengan mencari jarak *euclidean distance* citra potongan referensi dengan citra potongan lainnya. Jika jarak *euclidean distance* masih dibawah *treshold* maka nilai kelas citra potongan yang bukan referensi akan sama dengan nilai kelas citra referensi pembanding, sedangkan jika jarak lebih dari *treshold*

maka citra potongan akan diberi nilai kelas yang berbeda dari citra referensi pembanding dan citra tersebut kemudian akan di jadikan sebagai citra referensi baru. *Nilai treshold* didapatkan dari percobaan-percobaan yang telah dilakukan sebelumnya. Proses ini dilakukan berulangkali hingga semua potongan citra referensi telah dibandingkan dengan potongan citra sisanya.



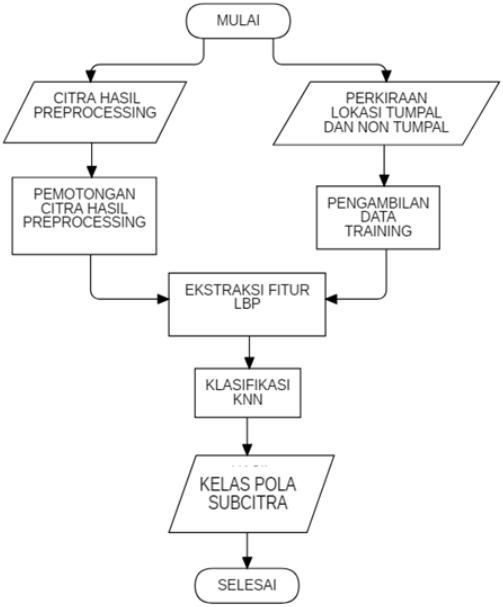
Gambar 3.6 Diagram Alir Deteksi Awal Tumpal dan Non Tumpal

Setelah tiap potongan telah diketahui nilai kelasnya, sistem akan melakukan perhitungan untuk mencari jumlah nilai kelas yang sama. Potongan yang memiliki nilai kelas dengan

jumlah paling kecil akan dianggap sebagai potongan tumpal sedangkan sisanya dianggap potongan non tumpal. Diagram alir dari deteksi awal tumpal dan non tumpal ditunjukkan oleh Gambar 3.6

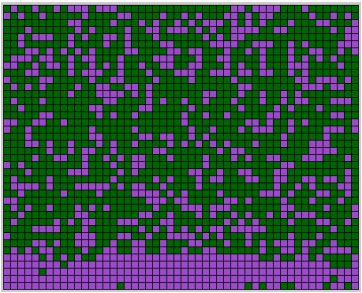
3.2.3 Identifikasi Pola Subcitra

Pada tahap ini area yang dianggap tumpal dan tidak tumpal masing-masing akan diambil sebagian datanya dalam bentuk potongan beerukuran 10x10 potongan untuk dijadikan data training. Data *training* tumpal akan disimpan pada file yang berbeda dengan data *training* non tumpal agar sistem dapat mengenali data ketika dipanggil kembali pada proses berikutnya. Jumlah data yang diambil pada masing-masing tumpal dan non tumpal bergantung pada variasi ujian percobaan. Variasi jumlah data *training* pada uji coba ini adalah 150, 200, dan 250.



Gambar 3.7 Diagram Alir Identifikasi Pola Subcitra

Setelah mendapatkan data *training* kemudian sistem akan memotong citra dengan ukuran yang sama dengan data *training* untuk dibandingkan dan diklasifikasi. Sebelum dibandingkan potongan akan diekstraksi fitur LBP-nya. Metode klasifikasi yang digunakan adalah KNN dengan *k* sesuai dengan variasi uji coba. Variasi *k* yang akan digunakan pada tugas akhir ini adalah 1, 3, dan 5. Setelah tahap klasifikasi maka ditemukanlah pola tiap subcitra untuk diproses pada tahap selanjutnya. Diagram alir dari tahap ini ditunjukkan oleh Gambar 3.7 dan contoh hasilnya dapat dilihat pada Gambar 3.8



Gambar 3.8 Hasil Tahap Identifikasi Pola Subcitra

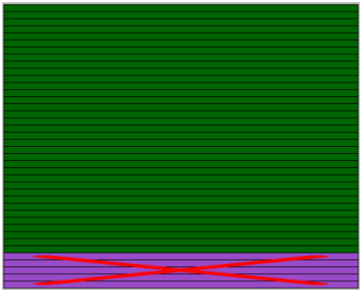
3.2.4 Optimalisasi dan Penandaan

Pada tahap ini hasil seluruh pola subcitra akan dijadikan masukkan. Dikarenakan pada batasan masalah telah diketahui bahwa tumpal citra uji coba sudah pasti berbentuk melebar maka sistem akan menentukan pola citra uji secara perbaris. Identifikasi pola citra perbaris dilakukan dengan mencari nilai kelas subcitra terbanyak dalam satu baris tersebut. Proses ini dilakukan berulangkali hingga tiap bari citra uji teridentifikasi polanya.

Kemudian pada proses selanjutnya sistem akan melakukan pembuangan prediksi *error*. Karena telah diketahui tumpal citra uji hanya berjumlah satu. Sistem akan melakukan optimalisasi dengan mencari jumlah deretan baris yang

teridentifikasi sebagai tumpal. Deretan yang memiliki jumlah terbanyak akan dianggap sebagai area tumpal yang sesungguhnya. Sistem kemudian memberikan penandaan sesuai pada letak lokasi deretan yang dianggap sebagai area tumpal sesungguhnya. Diagram alir dan hasil dari proses optimalisasi dan penandaan ditunjukkan oleh Gambar 3.9 dan Gambar 3.10

Gambar 3.9 Diagram Alir Optimalisasi dan Penandaan



Gambar 3.9 Hasil Tahap Optimalisasi dan Penandaan

BAB IV
IMPLEMENTASI SISTEM

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab 3 meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1. Lingkungan Implementasi

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Perangkat PC yang digunakan adalah desktop Lenovo dengan sistem operasi Windows 10 64-bit. Spesifikasi dari PC yang digunakan memiliki prosesor Intel Core i7-8550U dengan kecepatan 1,88 GHz, *Random Access Memory* (RAM) sebesar 8 GB. Spesifikasi PC dari sisi perangkat lunak yaitu menggunakan bahasa pemograman Processing 3.5.

4.2. Implementasi Praproses Data

Pada tahap ini implementasi membaca setiap gambar untuk dapat di-*resize* gambar dalam bentuk proses penjaman gambar dan proses konversi ke *grayscale*. Masukan dari tahap ini adalah data gambar pola batik dan keluarannya adalah gambar pola batik yang sudah di-*resize* dan dipertajam ketajamannya dalam bentuk *grayscale*. Implementasi dilakukan seperti yang ditunjukkan oleh Kode Sumber 4.1

```
1.  float[][] kernel = {{ -1, -1, -1},
2.                      { -1,  9, -1},
3.                      { -1, -1, -1}};
```

```
4.
5.  dataset.resize(500,400);
6.  dataset.loadPixels();
7.  PImage edgeImg = createImage(dataset.width, dataset.height,
8.  RGB);
9.  for (int y = 1; y < dataset.height-1; y++) {
10.   for (int x = 1; x < dataset.width-1; x++) {
11.       float sum = 0;
12. for (int ky = -1; ky <= 1; ky++) {
13.   for (int kx = -1; kx <= 1; kx++) {
14.       int pos = (y + ky)*dataset.width + (x + kx);
15.       float val = red(dataset.pixels[pos]);
16.       sum += kernel[ky+1][kx+1] * val;
17.   }
18. }
19. edgeImg.pixels[y*dataset.width + x] = color(sum, sum, sum);
20.   }
21. }
22. edgeImg.updatePixels();
23. edgeImg.save("gray.jpg");
```

Kode Sumber 4.1 Resize, Grayscale dan Penajaman Gambar

Fungsi *dataset.resize(500,400)* berfungsi untuk me-*resize* gambar menjadi ukuran 500x400. Variabel *sum* digunakan untuk menyimpan jumlah *kernel* dari dataset gambar. *(y + ky)*dataset.width + (x + kx)* digunakan untuk mendapatkan nilai ketajaman piksel. Kemudian fungsi *red(dataset.pixels[pos])* digunakan untuk mengubah gambar membuat nilai *red, green, blue* bernilai sama.

Kemudian dengan fungsi *kernel[ky+1][kx+1] * val* nilai ketajaman dikalikan dengan nilai variabel *kernel* yang sudah didefinisikan terlebih dahulu. Nilai-nilai yang didapat akhirnya diterapkan pada gambar baru *edgeImg.pixels[y*dataset.width + x] = color(sum, sum, sum).*

4.3. Implementasi Tahap Deteksi Awal Tumpal dan Non Tumpal

Pada subbab ini akan dijabarkan implementasi pada tahap *crop* gambar menjadi beberapa potongan, ekstraksi fitur LBP dan tentukan area tumpal.

4.3.1. Crop Gambar Menjadi Beberapa Potongan

Pada tahap ini citra yang telah melewati proses *preprocessing* akan dipotong menjadi beberapa bagian sama besar. Implementasi dilakukan seperti yang ditunjukkan oleh Kode Sumber 4.2

```
1.  for (int i = 0; i < Hreal; i=i+(tinggi/tot)) {
2.  for (int j = 0; j < Wreal; j=j+(lebar/tot)) {
3.  PImage myImage = imggray.get(j, i, lebar, tinggi);
4.  myImage.save("temp-"+tot+"/IMG-" + z +".png");
5.  matrix[tot-1][z-1][0]=j;
6.  matrix[tot-1][z-1][1]=i;
7.  matrix[tot-1][z-1][2]=j+(lebar/tot);
8.  matrix[tot-1][z-1][3]=i+(tinggi/tot);
9.  matrix[tot-1][z-1][5]=0;
10. z=z+1;
11. }
12. }
```

Kode Sumber 4.2 Crop Gambar Menjadi Potongan

4.3.2. Ekstraksi Fitur LBP

Pada tugas akhir digunakan metode ekstraksi fitur yaitu ekstraksi fitur tekstur menggunakan metode *Local Binary Pattern* pada tiap potongan dari tahap sebelumnya. Citra potongan tersebut kemudian diproses untuk didapatkan fitur LBP-nya. Metode LBP menghasilkan 256 fitur level keabuan untuk satu

piksel. Pada tahap ini diambil piksel tengah gambar dan 8 piksel tetangga, jika intensitas piksel tengah lebih besar dari atau sama dengan tetangganya, maka ditetapkan nilainya menjadi 1. Jika tidak, diatur ke 0. Proses ini diimplementasikan dalam Kode Sumber 4.3

```
1.
2.  int get_pixel(PImage img,int center,int x,int y){
3.     int new_value = 0;
4.     try {
5.         if (img.get(x,y) >= center){
                new_value = 1;
6.  }
7.  }
8.  catch(Exception e) {
9.  ;
10. }return new_value;
11. }
```

Kode Sumber 4.3 Nilai Piksel LBP

Dengan 8 piksel di sekitarnya, didapatkan total 2 ^ 8 = 256 kemungkinan kombinasi kode LBP. Digunakan ketenggaan 3x3 sehingga radius yang digunakan adalah 1. Dimana proses ini kemudian dilanjutkan pada tahap selanjutnya yang dapat dilihat pada Kode Sumber 4.4

```
1.
2.   int lbp_calculated_pixel(PImage img,int x,int y){
3.   int center = img.get(x,y);
4.   int[] val_ar=new int[8];
5.   int num=0;
6.   val_ar[num]=get_pixel(img, center, x-1, y+1);
7.   val_ar[num+1]=get_pixel(img, center, x, y+1);
```

```
 val_ar[num+2]=get_pixel(img, center, x+1, y+1);
 val_ar[num+3]=get_pixel(img, center, x+1, y);
 val_ar[num+4]=get_pixel(img, center, x+1, y-1);
 val_ar[num+5]=get_pixel(img, center, x, y-1);
 val_ar[num+6]=get_pixel(img, center, x-1, y-1);
 val_ar[num+7]=get_pixel(img, center, x-1, y);
 int[] power_val = {1, 2, 4, 8, 16, 32, 64, 128};
 int val=0;
for(int i=0;i<power_val.length;i++){
val += val_ar[i] * power_val[i];
}
return val;
}
```

Kode Sumber 4.4 Nilai Output LBP

Pada Kode Sumber 4.4. Setelah kedelapan tetangga sudah dibandingkan maka setiap nilai *new_value* akan dikalikan dengan 2^0 hingga 2^7 yang telah disimpan dalam variabel *val.* Selanjutnya, citra yang telah diubah menjadi citra LBP disimpan dalam variabel array.

4.3.3. Tentukan Area Tumpal

Pada tahap ini potongan yang paling awal akan dijadikan sebagai referensi dengan nilai 0, kemudian potongan-potongan lainnya akan dicari jarak *euclidean distance*-nya dengan potongan referensi. Jika jara *euclidean distance* dibawah *treshold* maka potongan tersebut akan diber nilai yang sama dengan potongan referensi pembanding, dan jika jarak di atas treshold maka potongan akan diberi nilai yang berbeda dari potongan referensi. Implementasi proses ini dituliskan dalam Kode Sumber 4.5

```
int nb[]=new int[loop];
int[][] ref=new int[loop][box[loop-1]];
int[] ArRef=new int[loop];
```

```
  for(int x=0;x<loop;x++){
    nb[x]=0;
    ref[x][0]=1;
    ArRef[x]=1;
  }
for (k=1; k<box[tot-1]+1; k++) {
    PImage image = loadImage("temp-"+tot+"/IMG-" + k +".png");
    int W = image.width;
    int H = image.height;
    int img_lbp[][][]=new int[loop][H][W];
    for (int i = 0; i < H; i++) {
      for (int j = 0; j < W; j++) {
        img_lbp[tot-1][i][j]= lbp_calculated_pixel(image, i, j);
}
}
for (int i = 0; i < nb[tot-1]+1; i++) {
      PImage image2 = loadImage("temp-"+tot+"/IMG-" + ref[tot-
1][i] +".png");
      int W2 = image2.width;
      int H2 = image2.height;
      int img_lbp2[][][]=new int[loop][H2][W2];
      for (int m = 0; m < H; m++) {
        for (int n = 0; n < W; n++) {
          img_lbp2[tot-1][m][n]= lbp_calculated_pixel(image2, m, n);
        }
      }
    double Sum = 0.0;
    for (int p=0;p<H;p++){
      for (int j=0;j<W;j++) {
        Sum = Sum+Math.pow((img_lbp[tot-1][p][j]-img_lbp2[tot-
1][p][j]),2.0);
```

```
33.         }
34.       }
35.       double dst=Math.sqrt(Sum);
36.     if(tot==1){
37.       tres=treshold1;
38.   }
39.     else if(tot==2){
40.       tres=treshold2;
41.     }
42.   println(dst);
43.       if(dst<tres){
44.           matrix[tot-1][k-1][4]=matrix[tot-1][ref[tot-1][i]-1][4];
45.       }
46.       else{
47.         matrix[tot-1][k-1][4]=matrix[tot-1][ref[tot-1][i]-1][4]+1;
48.           if(matrix[tot-1][k-1][4]>nb[tot-1]){
49.             ref[tot-1][ArRef[tot-1]]=k;
50.             ArRef[tot-1]=ArRef[tot-1]+1;
51.   nb[tot-1]+=1;
52.     }
53.     }
54.   }
55. }
```

Kode Sumber 4.5 Mencari Nilai Potongan

Setelah tiap potongan diketahui nilainya, tiap nilai yang sama akan dicari jumlahnya. Nilai yang memiliki jumlah terkecil akan dianggap sebagai nilai area tumpal dan lokasi potongan yang memiliki nilai tersebut akan dianggap sebagai area tumpal. Implementasi proses ini dituliskan dalam Kode Sumber 4.6

```
int tinggi=0;
int size=0;
int[][] a=new int[loop][box[loop-1]];
int[] temp=new int[loop];
int[] x=new int[loop];
int[] sign={0,0};
for(int i=0;i<nb[jml-1]+1;i++){
  a[jml-1][i]=0;
  }
for(int i=0;i<nb[jml-1]+1;i++){
  for(int k=1;k<box[jml-1]+1;k++){
    if(matrix[jml-1][k-1][4]==i){
      a[jml-1][i]+=1;
    }
  }
  if(a[jml-1][i]>temp[jml-1] && a[jml-1][i]>0){
    x[jml-1]=i;
    temp[jml-1]=a[jml-1][i];
    }
  }
 for(int k=1;k<box[jml-1]+1;k++){
  if(matrix[jml-1][k-1][4]!=x[jml-1]){
    sign[jml-1]=1;
    if(jml>1){
      for(int i=1;i<box[jml-2]+1;i++){
        matrix[jml-1][k-1][5]=1;
        if(matrix[jml-2][i-1][5]==1 && ((matrix[jml-1][k-1][0]>matrix[jml-2][i-1][0] && matrix[jml-1][k-1][0]<matrix[jml-2][i-1][2]) || (matrix[jml-1][k-1][2]>matrix[jml-2][i-1][0] && matrix[jml-1][k-1][2]<matrix[jml-2][i-1][2])) && ((matrix[jml-1][k-1][1]>matrix[jml-2][i-1][1] && matrix[jml-1][k-
```

```
1][1]<matrix[jml-2][i-1][3]) || (matrix[jml-1][k-1][3]>matrix[jml-2][i-1][1] && matrix[jml-1][k-1][3]<matrix[jml-2][i-1][3]))){
            tumpuk=1;
          }
         }
         }
      else{
        matrix[jml-1][k-1][5]=1;
      }
}
  }
  }
  for(int j=1;j<loop;j++){
    if(sign[j-1]==1){
      if(tumpuk==1){
      for(int i=1;i<box[j-1]+1;i++){
      for(int k=1;k<box[j]+1;k=k+2){
if(matrix[j-1][i-1][5]==1 && matrix[j][k-1][5]==1 && ((matrix[j][k-1][0]>matrix[j-1][i-1][0] && matrix[j][k-1][0]<matrix[j-1][i-1][2]) || (matrix[j][k-1][2]>matrix[j-1][i-1][0] && matrix[j][k-1][2]<matrix[j-1][i-1][2])) && ((matrix[j][k-1][1]>matrix[j-1][i-1][1] && matrix[j][k-1][1]<matrix[j-1][i-1][3]) || (matrix[j][k-1][3]>matrix[j-1][i-1][1] && matrix[j][k-1][3]<matrix[j-1][i-1][3]))){
          tinggi=matrix[j][k-1][1];
}
      }
      }
    }
else{
      for(int k=0;k<box[j-1];k=k+2){
        if(matrix[j-1][k][5]==1){
```

```
    tinggi=matrix[j-1][k][1];
  }
      }
      }
}
   for(int k=0;k<box[j];k=k+2){
        if(matrix[j][k][5]==1){
         tinggi=matrix[j][k][1];
      }
 }
  }
}
```

Kode Sumber 4.6 Mencari Area Tumpal

4.4. Implementasi Tahap Identifikasi Pola Subcitra

Pada subbab ini akan dijabarkan implementasi pada tahap *crop* gambar pengambilan data *training*, dan pemotongan dan klasifikasi KNN.

4.4.1. Pengambilan Data Training

Setelah areatumpal dan non tumpal telah ditemukan kemudian diamnil data *training* pada masing-masing area dengan jumlah sesuai dengan variasi yang diinginkan. Jumlah data *training* yang diambil bervariasi dari 150, 200, 250. Implimentasi dari prosrs ini dituliskan dalam Kode Sumber 4.7

```
if(tinggi<=200){
   nontumpal=ukuran+tinggi+ukuran;
  }
 else{
   nontumpal=tinggi-100;
  }
```

```
int z=1;
    PImage gray=loadImage("gray.jpg");
   for (int m = tinggi; m < +tinggi+(HTrain*size); m=m+size) {
     myImage.save("train1/" + z +".jpg");
      z=z+1;
      }
    }
   for (int m = ((ukuran)+tinggi-(HTrain2*size)); m < (ukuran)+tinggi-(HTrain2*size)+(HTrain2*size); m=m+size) {
    for (int n = 0+3; n <  (WTrain2*size)+3; n=n+size) {
      PImage myImage = gray.get(n, m, size, size);
      myImage.save("train1/" + z +".jpg");
  z=z+1;
      }
    }
    for (int m = ((ukuran)+tinggi-(HTrain3*size)); m < (ukuran)+tinggi-(HTrain3*size)+(HTrain3*size)-8; m=m+size) {
    for (int n = 0+28; n < (WTrain3*size)+28; n=n+size) {
      PImage myImage = gray.get(n, m, size, size);
     myImage.save("train1/" + z +".jpg");
 z=z+1;
      }
    }
  z=1;
  for (int m = nontumpal; m < nontumpal+(HTrain*size); m=m+size) {
    for (int n = 0; n < WTrain*size; n=n+size) {
      PImage myImage = gray.get(500-(n+size), 400-(m+size), size, size);
      myImage.save("train2/" + z +".jpg");
      z=z+1;
```

```
    }
  }
  for (int m = nontumpal+3; m < nontumpal+(HTrain2*size)+3; m=m+size) {
    for (int n = 0; n < WTrain2*size; n=n+size) {
      PImage myImage = gray.get(500-(n+size), 400-(m+size), size, size);
      myImage.save("train2/" + z +".jpg");
      z=z+1;
    }
}
 if(tumpuk==1 && tinggi>250){
    nontumpal=nontumpal-10;
  }
  else if(tumpuk==1 && tinggi<=250){
    nontumpal=nontumpal+10;
  }
 for (int m = nontumpal-2; m < nontumpal+(HTrain3*size)-2; m=m+size) {
    for (int n = 0; n < WTrain3*size; n=n+size) {
      PImage myImage = gray.get(500-(n+size), 400-(m+size), size, size);
      myImage.save("train2/" + z +".jpg");
      z=z+1;
```

Kode Sumber 4.7 Pengambilan Data

4.4.2. Pemotongan dan Klasifikasi KNN

Pada tahap ini citra hasil *preprocessing* dipotong dengan ukuran yang sama dengan citra data training untuk diklasifikasikan. klasifikasi dilakukan dengan metode *K-Nearest Neighbor* (KNN). Pada tugas akhir ini akan dilakukan *K-Nearest Neighbor* dengan menggunakan *euclidean distance* untuk mencari

jarak nilai LBP tiap blok citra data *test* dengan citra *training*. Implementasi dari tahap klasifikasi dituliskan dalam Kode Sumber 4.8

```
1.  for (int i = 0; i < Hreal; i=i+size) {
2.      println(i);
3.    for (int j = 0; j < Wreal; j=j+size) {
4.      for(int n=0;n<knn;n++){
5.        hslED[n]=999999999;
6.      }
7.       PImage myImage = dataGray.get(j, i, size, size);
8.      for (int r = 0; r < size; r++) {
9.        for (int s = 0; s < size; s++) {
10.         img_lbp[r][s]= lbp_calculated_pixel(myImage, r, s);
11.           hasil.println(img_lbp[r][s]);
12.          }
13. }
14. for(int m=1;m<=2;m++){
15.        for(int n=1;n<=totTrain;n++){
16.         PImage train=loadImage("train"+m+"/"+n+".jpg");
17.         for (int r = 0; r < size; r++) {
18.           for (int s = 0; s < size; s++) {
19.             img_lbp2[r][s]= lbp_calculated_pixel(train, r, s);
20. }
21.  }
22.  double Sum = 0.0;
23.        for (int r=0;r<size;r++){
24.         for (int s=0;s<size;s++) {
25.           Sum = Sum+Math.pow((img_lbp[r][s]-
    img_lbp2[r][s]),2.0);
26.         }
```

```
27.         }
28.   double dst=Math.sqrt(Sum);
29.   if(dst<hslED[knn-1]){
30.             hslpola[knn-1]=m;
31.             hslED[knn-1]=dst;
32.           }
33.           bubbleSort(hslED,hslpola);
34.    }
35.      }
36.    for(int z=0;z<knn;z++){
37.     switch(hslpola[z]){
38.      case 1:
39.       jpola[0][0]++;
40.       break;
41.      case 2:
42.       jpola[0][1]++;
43.       break;
44.       }
45.    }
46.   int tanda=0;
47.       for(int k=0;k<2;k++){
48.          if(jpola[0][k]>temp){
49.            temp=jpola[0][k];
50..   tanda=k+1;
51.            }
52.          }
53.          for(int k=0;k<2;k++){
54.          jpola[0][k]=0;
55.          }
56.   temp=0;
```

```
switch(tanda){
        case 1:
         jpola[1][0]++;
         fill(154,76,200);//ungu
         rect(535+j,i+125,size,size);
         tumpal++;
         if(i<=50){
           tumpalreal++;
         }
  break;
        case 2:
         fill(0,100,0);//hijau
         rect(535+j,i+125,size,size);
         nontumpal++;
         if(i<=50){
           nontumpalreal++;
  }
         jpola[1][1]++;
         break;
}
```

Kode Sumber 4.8 Implementasi Tahap Klasifikasi

Pada proses ini data LBP citra *testing* dan *training* yang didapatkan dari proses sebelumnya akan dipanggil dan dicari jaraknya dengan *euclidean distance.* Pada tugas akhir ini nilai *k* pada *K-Nearest Neighbor* adalah bervariasi dari 1, 3 , dan 5.

Dapat dilihat pada Kode Sumber 4.8 pencerian jarak *euclidean distance* nilai LBP dilakukan sebanyak nilai *k* pada KNN kemudian dengan menggunakan fungsi *switch(pola)* dicari nilai kelas training yang menghasilkan jarak paling kecil.

4.5. Implementasi Tahap Optimalisasi dan Penandaan

Pada subbab ini akan dijabarkan implementasi pada tahap optimalisasi dan penandaan. Tahap optimalisasi sendiri dapat dibagi menjadi dua yaitu menentukan kelas motif per-baris dan pembuangan hasil prediksi *error*. Dari dua tahap optimalisasi tersebut kemudian didapatkan prediksi lokasi tumpal yang sesungguhnya untuk ditandai.

4.5.1 Identifikasi Kelas Pola Perbaris

Dengan mengetahui bahwa tumpal berbentuk horizontal, dapat diberikan aturan untuk meningkatkan nilai kebenaran prediksi. Aturan yang diberikan ialah tiap blok pada satu baris yang sama akan dijumlahkan nilainya, kemudian nilai citra per-baris akan ditentukan dengan nilai blok yang paling banyak pada baris tersebut. Implementasi optimalisasi ini terdapat pada Kode Sumber 4.9

```
1.  if(x%div==0){
2.      int tanda2=0;
3.      for(int k=0;k<2;k++){
4.      if(jpola[1][k]>tempx){
5.        tempx=jpola[1][k];
6.         tanda2=k+1;
7.        }
8.      }
9.      if(tanda==1){
10.       ungu[i/10]=tempx;
11.     }
12.     for(int k=0;k<2;k++){
13.     jpola[1][k]=0;
14.     }
15.     tempx=0;
```

```
16.   polatumpal[i/size]=klasifikasi(j,i,tanda2,size);
17.       }
18.     }
19.      }
20.  int klasifikasi(int lebar,int tinggi,int tanda,int size){
21.    int hasil=0;
22.    switch(tanda){
23.
24.     case 1:
25.      fill(154,76,200);//ungu
26.      rect(25,tinggi+125,500,size);
27.      hasil=1;
28.      break;
29.      //nontumpal
30.     case 2:
31.      fill(0,100,0);//hijau
32.      rect(25,tinggi+125,500,size);
33.      hasil=2;
34.      break;
35.    }
36.    return hasil;
37.  }
```

Kode Sumber 4.9 Implementasi Menentukan Kelas Motif

4.5.2 Hitung Deretan Tumpal Terbanyak

Dari tahap optimasi didapatkan nilai citra per-baris, namun dikarenakan area non-tumpal yang besar mengakibatkan kemungkinan hasil prediksi *error* menjadi lebih tinggi. Maka dari itu optimalisasi pembuangan prediksi *error* dilakukan. Pada tahap ini hasil dari optimalisasi sebelumnya dipanggil, kemudian

dihitung hasil nilai kelas motif yang berurutan. Dikarenakan pada tugas akhir ini salah satu batas masalahnya adalah jumlah tumpal selalu hanya ada satu, sistem akan mengambil area tumpal yang memiliki urutan terbanyak dan membuang hasil yang dianggap tumpal dengan urutan sama yang jumlahnya lebih kecil dan didapatkan hasil prediksi area tumpal akhir. Implementasi tahap ini terdapat pada Kode Sumber 4.10

```
1.   int[] count=new int[pola.length];
2.    for(int i=0;i<pola.length;i++){
3.      count[i]=0;
4.    }
5.    for(int i=0;i<pola.length;i++){
6.      int x=pola[i];
7.      for(int j=i;j<pola.length;j++){
8.        if(pola[j]==x){
9.          count[i]++;
10.       }
11.       else{
12.         break;
13.       }
14.     }
15.   }
16.  int temp=0;

17.   int tanda=0;
18.     int[] totungu={0,0};
19.   for(int i=0;i<pola.length;i++){
20.     if(pola[i]!=2 && count[i]>temp){
21.       temp=count[i];
22.       tanda=i;
```

```
23.      for(int n=i;n<=temp+i;n++){
24.        totungu[0]=totungu[0]+ungu[i];
25.      }
26.    }
27.    else if(pola[i]!=2 && count[i]==temp){
28       for(int n=i;n<=count[i]+i;n++){
29.  totungu[1]=totungu[1]+ungu[i];
30.      }
31.      if(totungu[0]<totungu[1]){
32.      temp=count[i];
33.  tanda=i;
34.      }
35.    }
36.  }
```

Kode Sumber 4.10 Implementasi Hitung Deretan Tumpal Terbanyak

4.5.3 Penandaan

Setelah mendapatkan hasil prediksi area tumpal, sistem akan memberikan penandaan pada area tersebut yang menandakan hasil akhir dari sistem. Tahap penandaan ini dapat dilihat pada Kode Sumber 4.11

```
1.  PImage cross=loadImage("Ncross.png");
2.  cross.resize(500,size*temp);
3.  image(cross,25,125+(tanda*size));
4.  PImage cross=loadImage("Ncross.png");
```

Kode Sumber 4.11 Implementasi Penandaaa

[Halaman ini sengaja dikosongkan]

BAB V
UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

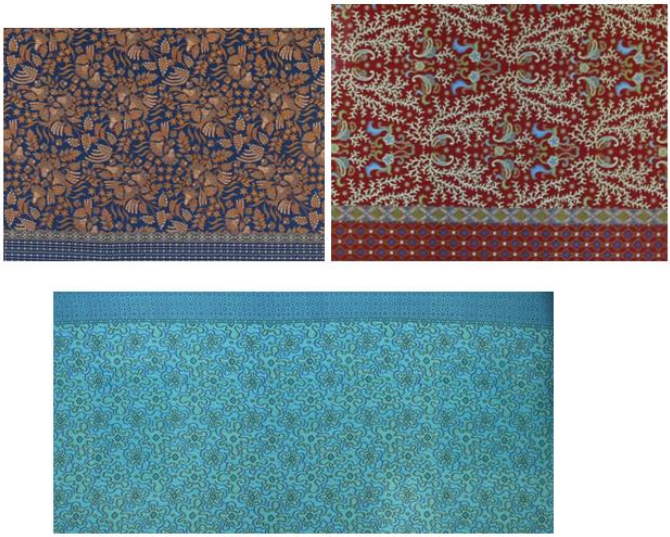
5.1. Lingkungan Uji Coba

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Perangkat PC yang digunakan adalah desktop Lenovo dengan sistem operasi Windows 10 64-bit. Spesifikasi dari PC yang digunakan memiliki prosesor Intel Core i7-8550U dengan kecepatan 1,80 GHz, *Random Access Memory* (RAM) sebesar 8 GB. Spesifikasi PC dari sisi perangkat lunak yaitu menggunakan bahasa pemograman Processing 3.5.

5.2. Dataset

Data yang digunakan untuk uji coba deteksi posisi tumpal pada citra kain batik dengan menggunakan fitur *Local Binary Pattern* (LBP) adalah citra batik yang memiliki tumpal dengan ukuran dan motif tumpal yang berbeda. Terdapat dua jenis motif batik pada satu citra batik yang dimana salah satu dari dua motif tersebut merupakan tumpal. Tumpal pada citra adalah tumpal yang berbentuk horizontal dan hanya terdapat satu tumpal pada satu citra batik.

Jumlah citra batik yang digunakan dalam tugas akhir ini ada 8 dengan 16 motif berbeda. Pada pengujian citra batik akan di-resize menjadi berukuran 500x400 dan di *crop* dengan ukuran 10x10 sehingga terdapat 2000 citra *testing* yang akan diuji coba pada satu percobaan. Untuk data *training* disediakan beberapa variasi jumlah yang akan digunakan adalah 150, 200, 250. Data *training* diambil secara otomatis dari deteksi lokasi tumpal awal yang kemudian dilanjutkan ke tahap berikutnya dan diambil potongan sebesar 10x10 pada citra uji coba. Contoh citra dapat dilihat Gambar 5.1



Gambar 5.1 Contoh Citra Dataset

5.3. Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui kesesuaian keluaran dari tiap tahap atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Pada tugas akhir ini akan diuji coba tiap citra yang tersedia dengan beberapa jumlah training dan *k* yang berbeda kemudian dicari keakurasiannya dan lama pengerjaan sistem dalam melakukan tugasnya. Variasi jumlah training yang akan digunakan pada tugas akhir ini berjumlah 150, 200, dan 250. Sedangkan variasi *k* yang akan digunakan pada tugas akhir ini yaitu *k*=1, *k*=3, dan *k*=5.

5.3.1. Uji Koefisien Matriks dan Kecepatan Data 1

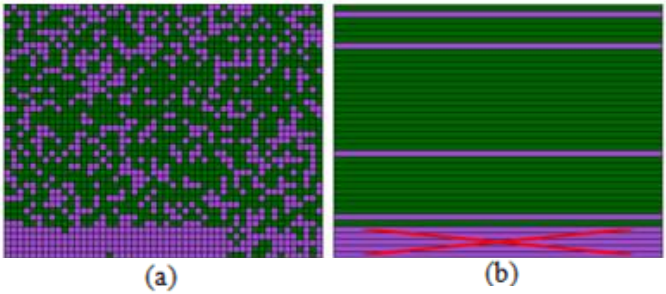
Gambar 5.2 Citra Data 1

Akurasi pada setiap uji coba data 1 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.1. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

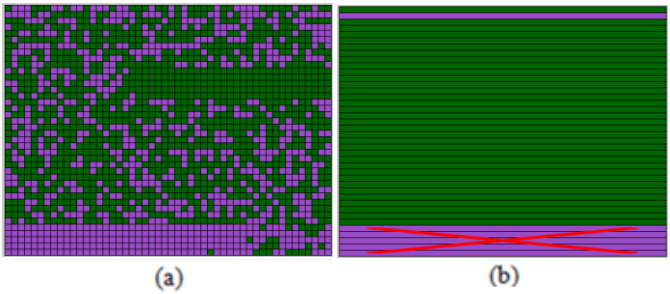
Tabel 5.1 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 1 Training 150 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	226 Potongan	622 Potongan
		Bukan Tumpal	24 Potongan	1128 Potongan
	Waktu Pengerjaan		5.24 Menit	
k=3	Kelas Prediksi	Tumpal	222 Potongan	634 Potongan
		Bukan Tumpal	28 Potongan	1116 Potongan
	Waktu Pengerjaan		4.89 Menit	
k=5	Kelas Prediksi	Tumpal	249 Potongan	51 Potongan
		Bukan Tumpal	249 Potongan	817 Potongan
	Waktu Pengerjaan		5.05 Menit	

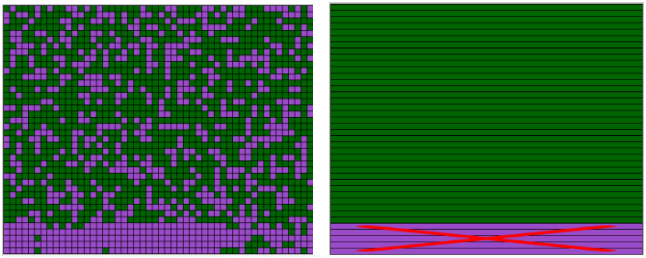
Hasil output uji coba data 1 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.3, Gambar 5.4, dan Gambar 5.5. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.3 Output Uji Coba Data 1 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.4 Output Uji Coba Data 1 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

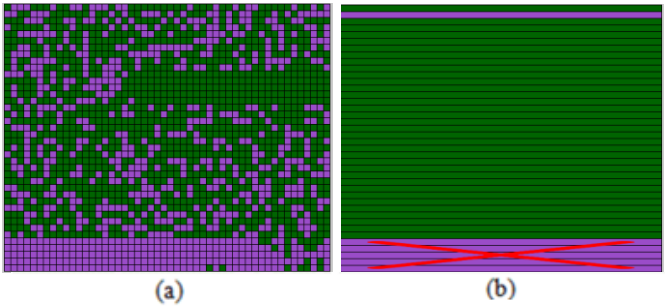
Gambar 5.5 Output Uji Coba Data 1 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 1 dengan variasi jumlah *training* 200 ditunjukkan pada Tabel 5.2. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

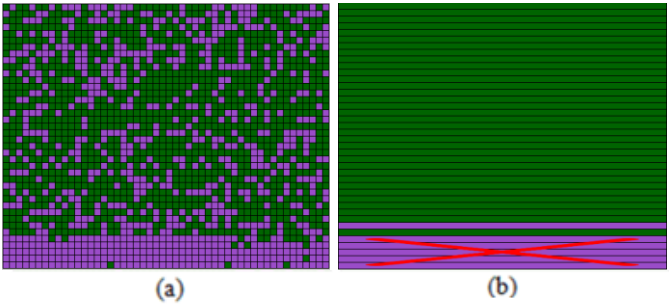
Tabel 5.2 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 1 Training 200 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	226 Potongan	622 Potongan
		Bukan Tumpal	595 Potongan	1155 Potongan
	Waktu Pengerjaan		5.24 Menit	
k=3	Kelas Prediksi	Tumpal	233 Potongan	588 Potongan
		Bukan Tumpal	17 Potongan	1162 Potongan
	Waktu Pengerjaan		4.89 Menit	
k=5	Kelas Prediksi	Tumpal	223 Potongan	570 Potongan
		Bukan Tumpal	27 Potongan	1180 Potongan
	Waktu Pengerjaan		7.62 Menit	

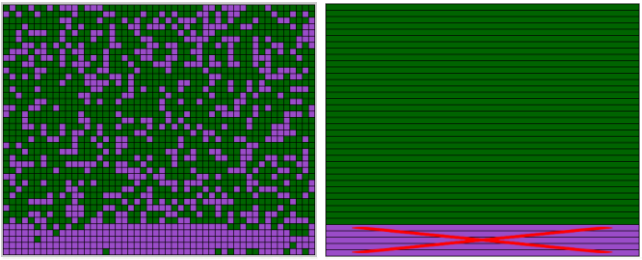
Hasil output uji coba data 1 dengan variasi jumlah training berjumlah 200 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.6, Gambar 5.7, dan Gambar 5.8. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.6 Output Uji Coba Data 1 k=1 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.7 Output Uji Coba Data 1 k=3 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

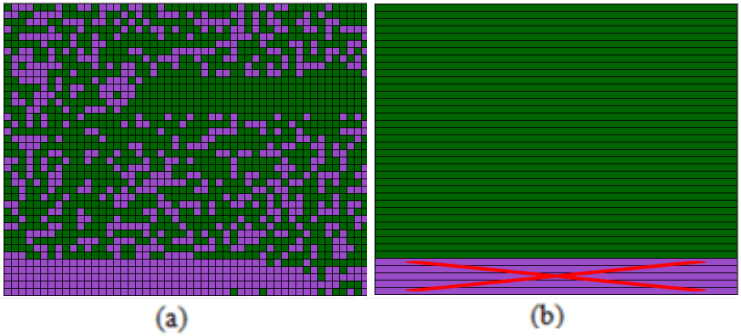
Gambar 5.8 Output Uji Coba Data 1 k=5 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 1 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.3. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

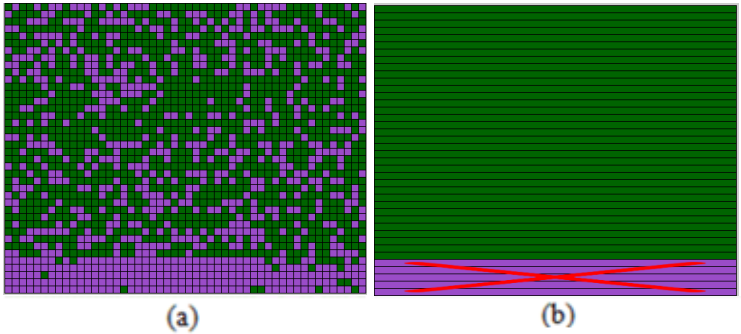
Tabel 5.3 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 1 Training 250 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	225 Potongan	594 Potongan
		Bukan Tumpal	25 Potongan	1156 Potongan
	Waktu Pengerjaan		9.6 Menit	
k=3	Kelas Prediksi	Tumpal	229 Potongan	598 Potongan
		Bukan Tumpal	21 Potongan	1152 Potongan
	Waktu Pengerjaan		9.17 Menit	
k=5	Kelas Prediksi	Tumpal	224 Potongan	585 Potongan
		Bukan Tumpal	26 Potongan	1165 Potongan
	Waktu Pengerjaan		10.8 Menit	

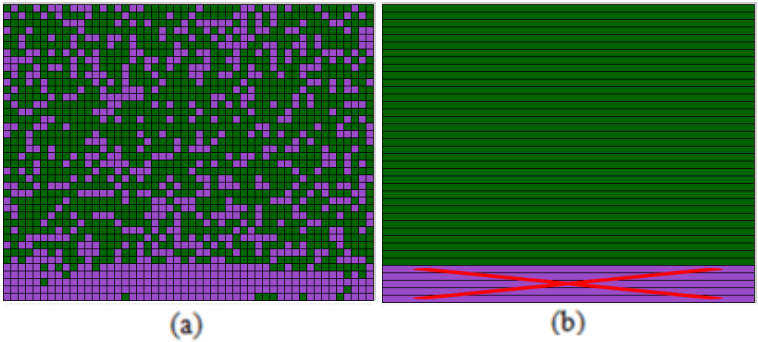
Hasil output uji coba data 1 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.9, Gambar 5.10 dan Gambar 5.11. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5. 9 Output Uji Coba Data 1 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.10 Output Uji Coba Data 1 k=3 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.11 Output Uji Coba Data 1 k=3 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

5.3.2. Uji Koefisien Matriks dan Kecepatan Data 2

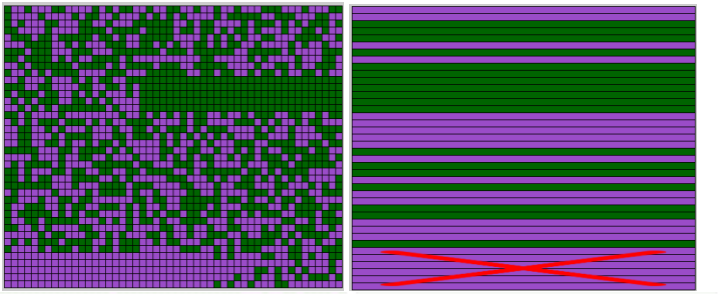
Gambar 5.12 Citra Data 2

Akurasi pada setiap uji coba data 2 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.4. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.4 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 2 Training 150 Sebelum Optimalisasi

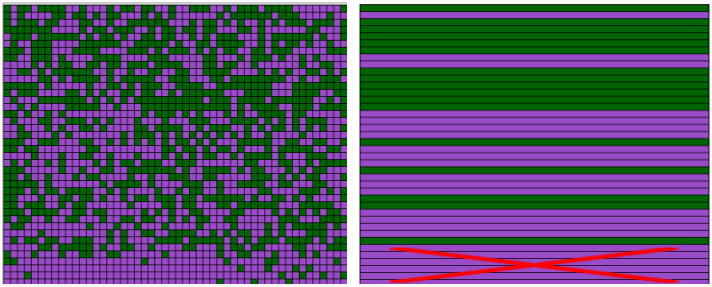
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	245 Potongan	773 Potongan
		Bukan Tumpal	55 Potongan	927 Potongan
	Waktu Pengerjaan		4.98 Menit	
k=3	Kelas Prediksi	Tumpal	244 Potongan	837 Potongan
		Bukan Tumpal	56 Potongan	863 Potongan
	Waktu Pengerjaan		5.31 Menit	
k=5	Kelas Prediksi	Tumpal	198 Potongan	934 Potongan
		Bukan Tumpal	52 Potongan	816 Potongan
	Waktu Pengerjaan		5.1 Menit	

Hasil output uji coba data 2 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.13, Gambar 5.14, dan Gambar 5.15. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) meskipun tidak semuanya bernilai benar. Pada pengujian ini dapat dilihat terdapat dua hasil setelah optimalisasi yang bernilai enar dan satu benilai salah dengan baris *error* yang cukup banyak. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap tidak cukup optimal.



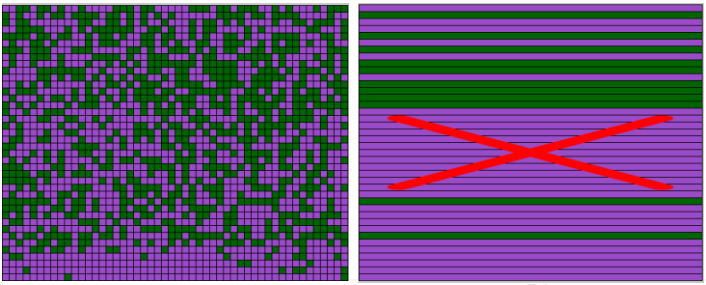
(a) (b)

Gambar 5.13 Output Uji Coba Data 2 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.14 Output Uji Coba Data 2 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

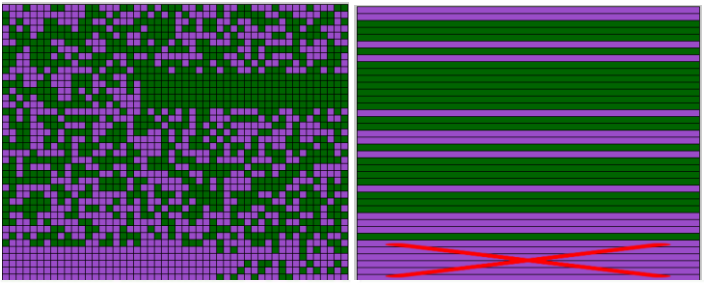
Gambar 5.15 Output Uji Coba Data 2 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 2 dengan variasi jumlah *training* 200 ditunjukkan pada Tabel 5.5. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.5 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 2 Training 200 Sebelum Optimalisasi

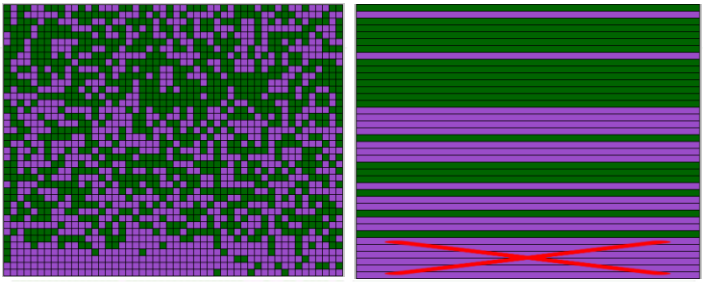
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	252 Potongan	747 Potongan
		Bukan Tumpal	48 Potongan	953 Potongan
	Waktu Pengerjaan		6.99 Menit	
k=3	Kelas Prediksi	Tumpal	245 Potongan	793 Potongan
		Bukan Tumpal	55 Potongan	907 Potongan
	Waktu Pengerjaan		6.76 Menit	
k=5	Kelas Prediksi	Tumpal	248 Potongan	840 Potongan
		Bukan Tumpal	52 Potongan	860 Potongan
	Waktu Pengerjaan		7.41 Menit	

Hasil output uji coba data 2 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.16, Gambar 5.17, dan Gambar 5.18 Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) meskipun tidak semuanya bernilai benar. Pada pengujian ini dapat dilihat terdapat dua hasil setelah optimalisasi yang bernilai enar dan satu benilai salah dengan baris *error* yang cukup banyak. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap tidak cukup optimal.



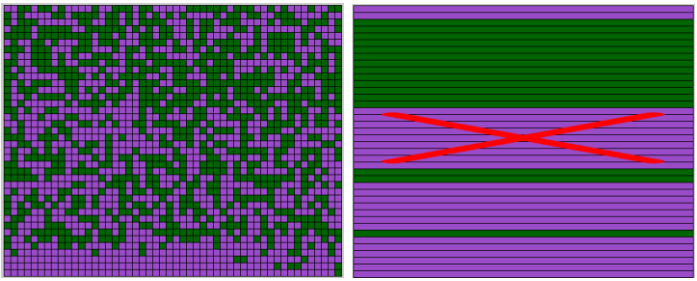
(a) (b)

Gambar 5.16 Output Uji Coba Data 2 k=1 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.17 Output Citra Data 2 k=3 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

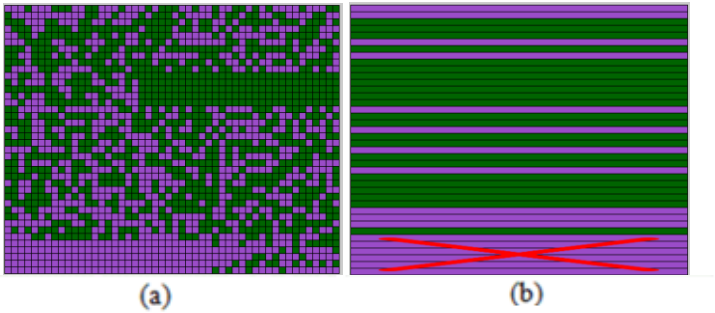
Gambar 5.18 Output Citra Data 2 k=5 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 2 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.6. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

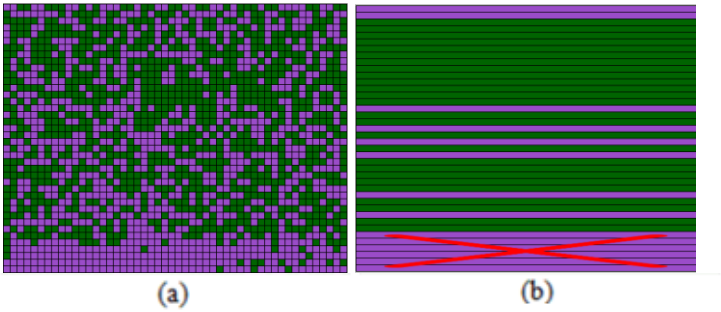
Tabel 5.6 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 2 Training 250 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	248 Potongan	722 Potongan
		Bukan Tumpal	52 Potongan	980 Potongan
	Waktu Pengerjaan		9.04 Menit	
k=3	Kelas Prediksi	Tumpal	243 Potongan	737 Potongan
		Bukan Tumpal	57 Potongan	963 Potongan
	Waktu Pengerjaan		8.88 Menit	
k=5	Kelas Prediksi	Tumpal	243 Potongan	57 Potongan
		Bukan Tumpal	773 Potongan	927 Potongan
	Waktu Pengerjaan		7.71 Menit	

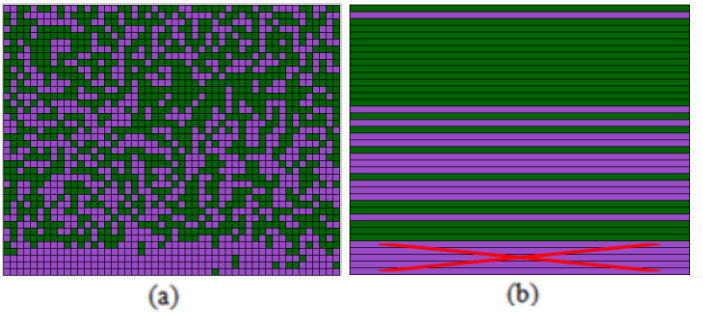
Hasil output uji coba data 2 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.19, Gambar 5.20 dan Gambar 5.21. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.19 Output Uji Coba Data 2 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

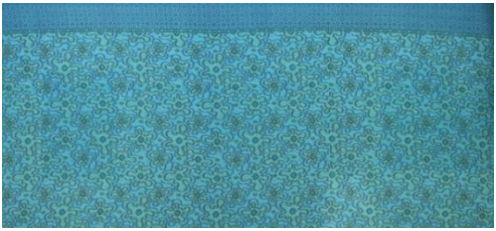


Gambar 5.20 Output Citra Data 2 k=3 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.21 Output Citra Data 2 k=5 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

5.3.3. Uji Koefisien Matriks dan Kecepatan Data 3



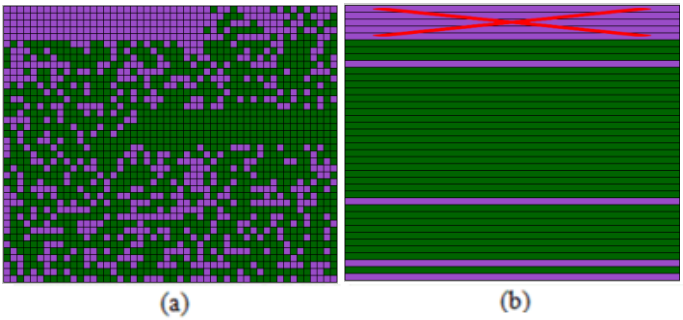
Gambar 5.22 Citra Data 3

Akurasi pada setiap uji coba data 3 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.7. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

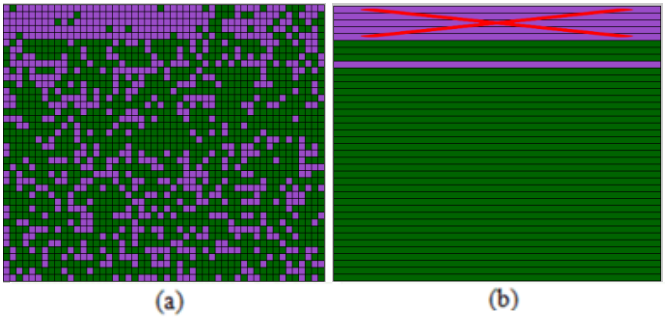
Tabel 5.7 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 3 Training 150 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	213 Potongan	578 Potongan
		Bukan Tumpal	37 Potongan	1172 Potongan
	Waktu Pengerjaan		5.31 Menit	
k=3	Kelas Prediksi	Tumpal	217 Potongan	545 Potongan
		Bukan Tumpal	83 Potongan	1115 Potongan
	Waktu Pengerjaan		5.41 Menit	
k=5	Kelas Prediksi	Tumpal	188 Potongan	523 Potongan
		Bukan Tumpal	62 Potongan	1227 Potongan
	Waktu Pengerjaan		5.06 Menit	

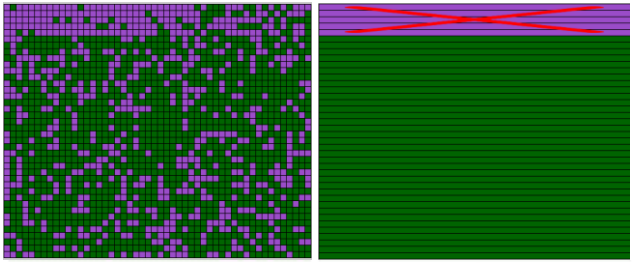
Hasil output uji coba data 3 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.23, Gambar 5.24, dan Gambar 5.25. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang jauh lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.23 Output Uji Coba Data 3 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.24 Output Uji Coba Data 3 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

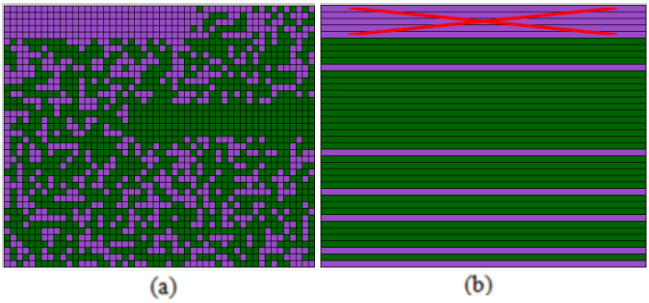
Gambar 5.25 Output Uji Coba Data 3 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 3 dengan variasi jumlah *training* 200 ditunjukkan pada Tabel 5.8. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

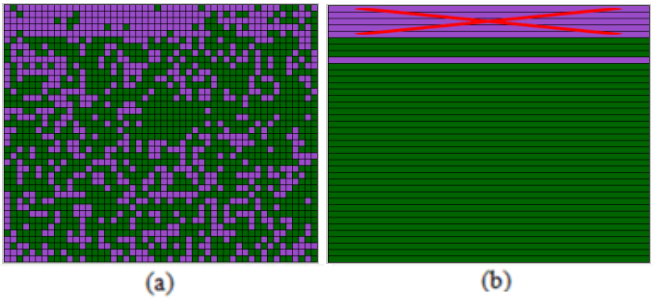
Tabel 5.8 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 3 Training 200 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	212 Potongan	640 Potongan
		Bukan Tumpal	38 Potongan	1110 Potongan
	Waktu Pengerjaan		8.25 Menit	
k=3	Kelas Prediksi	Tumpal	221 Potongan	599 Potongan
		Bukan Tumpal	79 Potongan	1101 Potongan
	Waktu Pengerjaan		8.52 Menit	
k=5	Kelas Prediksi	Tumpal	208 Potongan	580 Potongan
		Bukan Tumpal	92 Potongan	1120 Potongan
	Waktu Pengerjaan		8.45 Menit	

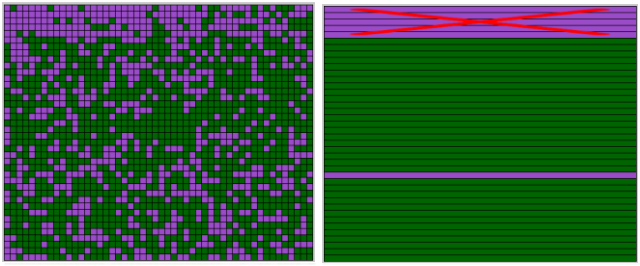
Hasil output uji coba data 3 dengan variasi jumlah training berjumlah 200 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.26, Gambar 5.27, dan Gambar 5.28. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.26 Output Uji Coba Data 3 k=1 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.27 Output Uji Coba Data 3 k=3 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

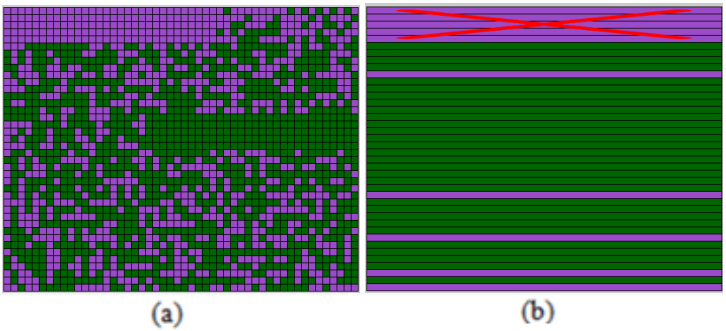
Gambar 5.28 Output Uji Coba Data 3 k=5 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 3 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.9. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

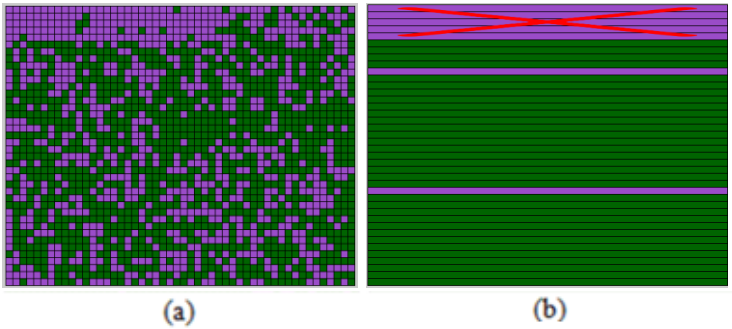
Tabel 5.9 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 3 Training 250 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	214 Potongan	668 Potongan
		Bukan Tumpal	36 Potongan	1082 Potongan
	Waktu Pengerjaan		10.32 Menit	
k=3	Kelas Prediksi	Tumpal	200 Potongan	644 Potongan
		Bukan Tumpal	50 Potongan	1106 Potongan
	Waktu Pengerjaan		9.09 Menit	
k=5	Kelas Prediksi	Tumpal	184 Potongan	640 Potongan
		Bukan Tumpal	66 Potongan	1110 Potongan
	Waktu Pengerjaan		8.88 Menit	

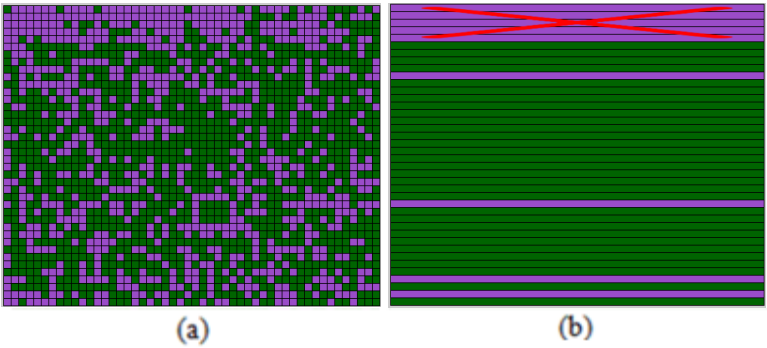
Hasil output uji coba data 3 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.29, Gambar 5.30 dan Gambar 5.31. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.29 Output Uji Coba Data 3 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.30 Output Uji Coba Data 3 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.31 Output Uji Coba Data 3 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

5.3.4. Uji Koefisien Matriks dan Kecepatan Data 4

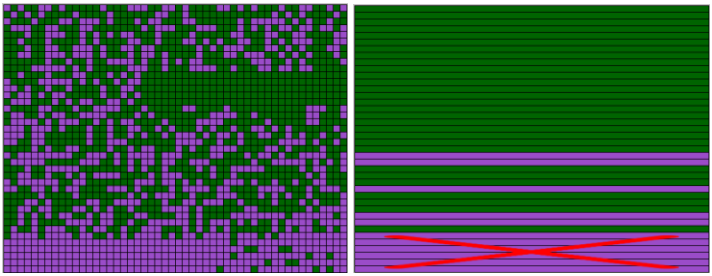
Gambar 5.32 Citra Data 4

Akurasi pada setiap uji coba data 4 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.10. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.10 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 4 Training 150 Sebelum Optimalisasi

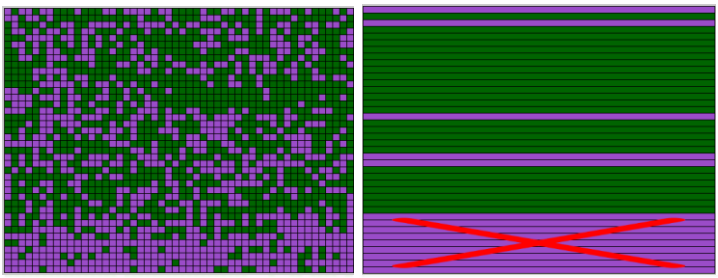
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	352 Potongan	886 Potongan
		Bukan Tumpal	98 Potongan	964 Potongan
	Waktu Pengerjaan		4.86 Menit	
k=3	Kelas Prediksi	Tumpal	329 Potongan	603 Potongan
		Bukan Tumpal	121 Potongan	947 Potongan
	Waktu Pengerjaan		4.91 Menit	
k=5	Kelas Prediksi	Tumpal	326 Potongan	617 Potongan
		Bukan Tumpal	124 Potongan	933 Potongan
	Waktu Pengerjaan		5.66 Menit	

Hasil output uji coba data 4 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.33, Gambar 5.34, dan Gambar 5.35. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan dua hasil bernilai benar dan satu hasil bernilai kurang tepat, dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



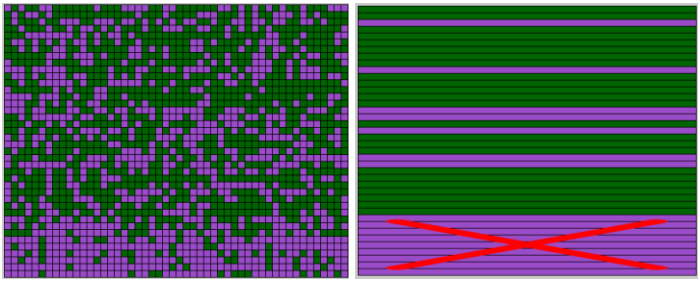
(a) (b)

Gambar 5.33 Output Uji Coba Data 4 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.34 Output Uji Coba Data 4 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

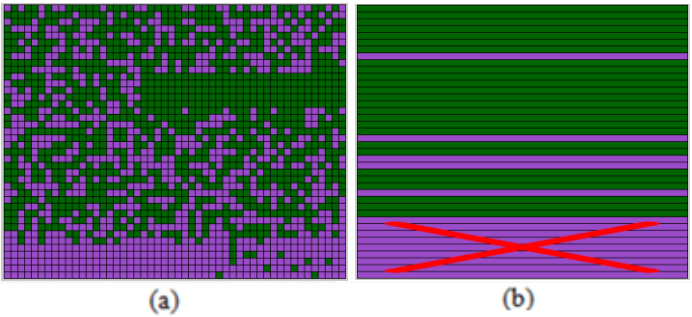
Gambar 5.35Output Uji Coba Data 4 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 4 dengan variasi jumlah *training* 200 ditunjukkan Tabel 5.11. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

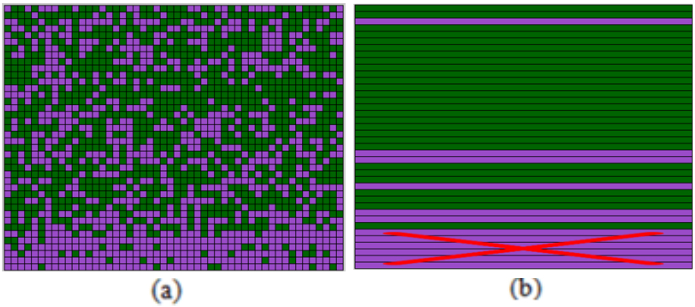
Tabel 5.11 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 4 Training 200 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	358 Potongan	577 Potongan
		Bukan Tumpal	92 Potongan	973 Potongan
	Waktu Pengerjaan		8.17 Menit	
k=3	Kelas Prediksi	Tumpal	334 Potongan	582 Potongan
		Bukan Tumpal	116 Potongan	968 Potongan
	Waktu Pengerjaan		6.58 Menit	
k=5	Kelas Prediksi	Tumpal	355 Potongan	563 Potongan
		Bukan Tumpal	95 Potongan	987 Potongan
	Waktu Pengerjaan		7.65 Menit	

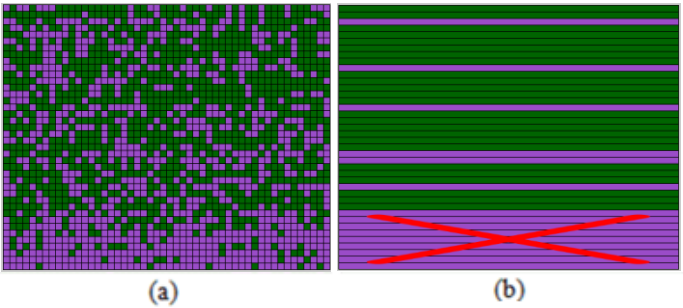
Hasil output uji coba data 4 dengan variasi jumlah training berjumlah 200 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.36, Gambar 5.37, dan Gambar 5.38. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan dua hasil bernilai benar dan satu hasil bernilai kurang tepat, dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.36 Output Uji Coba Data 4 k=1 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.37 Output Citra Data 4 k=3 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



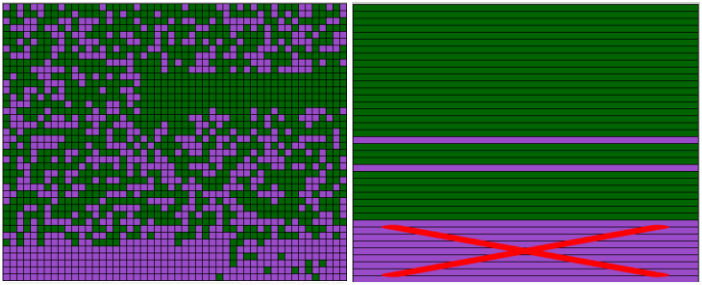
Gambar 5.38 Output Citra Data 4 k=5 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 4 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.12. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.12 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 4 Training 250 Sebelum Optimalisasi

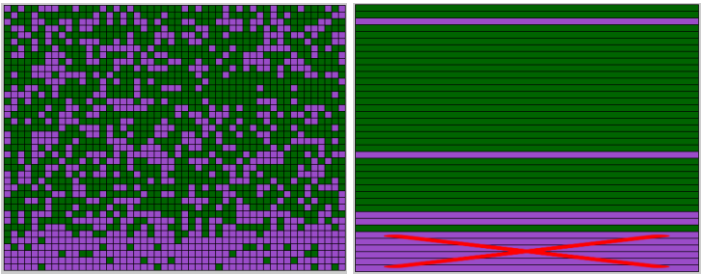
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	371 Potongan	542 Potongan
		Bukan Tumpal	129 Potongan	958 Potongan
	Waktu Pengerjaan		8.28 Menit	
k=3	Kelas Prediksi	Tumpal	349 Potongan	548 Potongan
		Bukan Tumpal	151 Potongan	952 Potongan
	Waktu Pengerjaan		8.67 Menit	
k=5	Kelas Prediksi	Tumpal	349 Potongan	553 Potongan
		Bukan Tumpal	151 Potongan	947 Potongan
	Waktu Pengerjaan		8.34 Menit	

Hasil output uji coba data 4 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.39, Gambar 5.40 dan Gambar 5.41. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan dua hasil bernilai benar dan satu hasil bernilai kurang tepat, dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



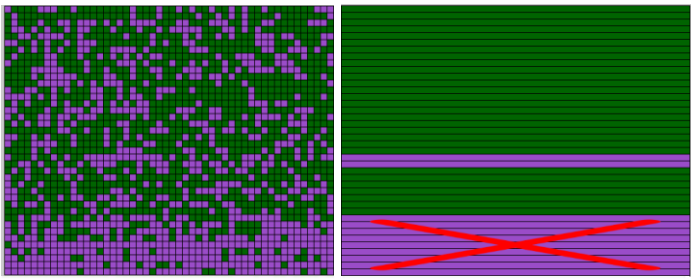
(a) (b)

Gambar 5.39 Output Uji Coba Data 4 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

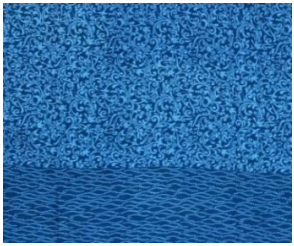
Gambar 5.40 Output Citra Data 4 k=3 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.41Output Citra Data 4 k=5 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

5.3.5. Uji Koefisien Matriks dan Kecepatan Data 5



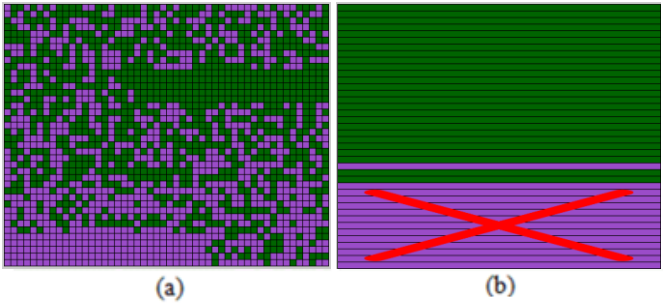
Gambar 5.42 Citra Data 5

Akurasi pada setiap uji coba data 5 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.13. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

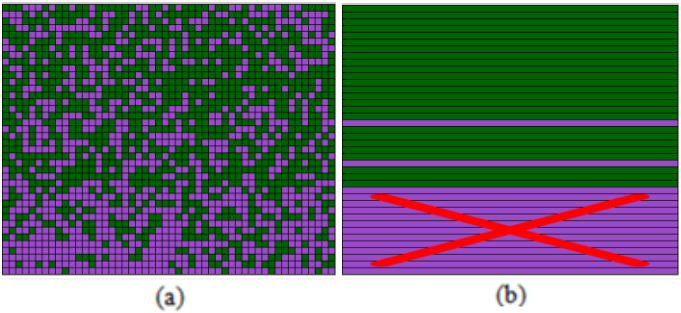
Tabel 5.13 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 5 Training 150 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	460 Potongan	506 Potongan
		Bukan Tumpal	190 Potongan	844 Potongan
	Waktu Pengerjaan		4.97 Menit	
k=3	Kelas Prediksi	Tumpal	433 Potongan	515 Potongan
		Bukan Tumpal	217 Potongan	835 Potongan
	Waktu Pengerjaan		4.72 Menit	
k=5	Kelas Prediksi	Tumpal	442 Potongan	208 Potongan
		Bukan Tumpal	533 Potongan	817 Potongan
	Waktu Pengerjaan		4.74 Menit	

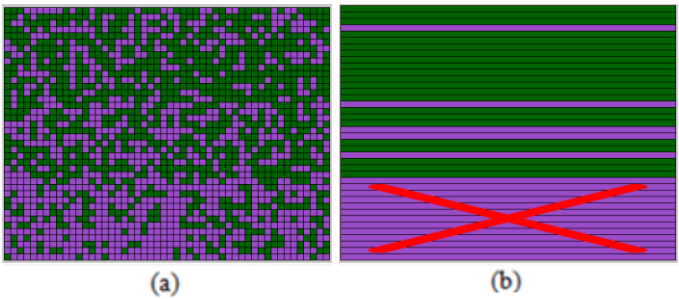
Hasil output uji coba data 5 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.43, Gambar 5.44, dan Gambar 5.45. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai benar dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.43 Output Uji Coba Data 5 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.44 Output Uji Coba Data 5 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



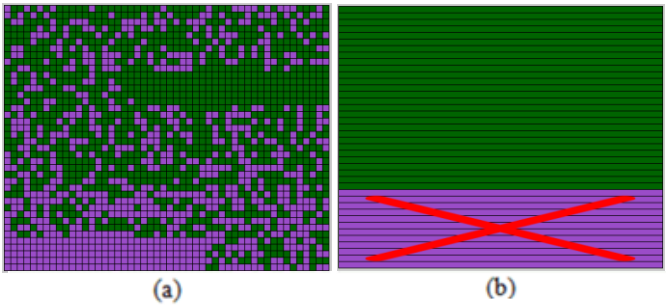
Gambar 5.45 Output Uji Coba Data 5 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 5 dengan variasi jumlah *training* 200 ditunjukkan pada Tabel 5.14. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

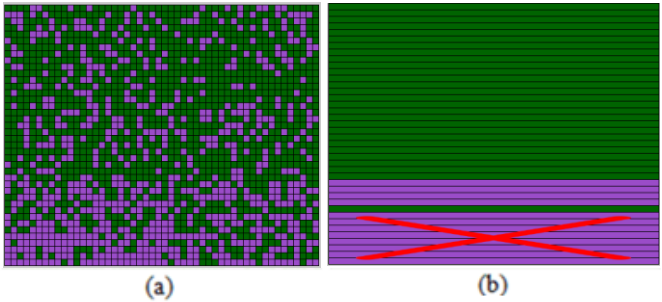
Tabel 5.14 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 5 Training 200 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	430 Potongan	428 Potongan
		Bukan Tumpal	220 Potongan	922 Potongan
	Waktu Pengerjaan		6.71 Menit	
k=3	Kelas Prediksi	Tumpal	393 Potongan	383 Potongan
		Bukan Tumpal	257 Potongan	967 Potongan
	Waktu Pengerjaan		7.04 Menit	
k=5	Kelas Prediksi	Tumpal	383 Potongan	384 Potongan
		Bukan Tumpal	267 Potongan	966 Potongan
	Waktu Pengerjaan		6.52 Menit	

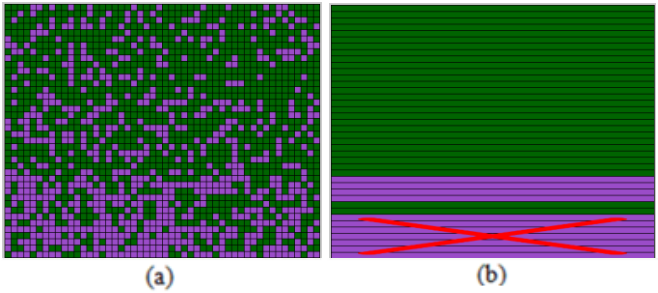
Hasil output uji coba data 5 dengan variasi jumlah training berjumlah 200 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.46, Gambar 5.47, dan Gambar 5.48. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan satu hasil bernilai benar dan dua hasil bernilai kurang tepat, dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.46 Output Uji Coba Data 5 k=1 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.47 Output Citra Data 5 k=3 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



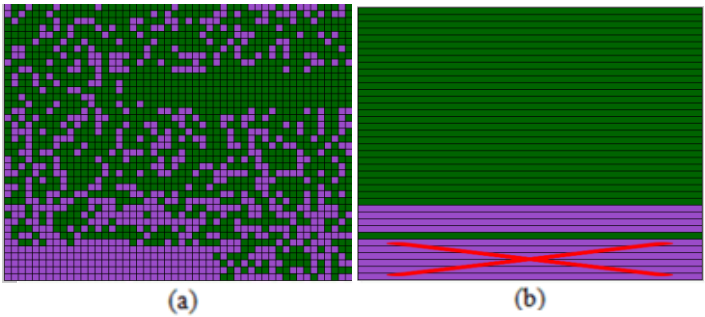
Gambar 5.48 Output Citra Data 5 k=5 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 5 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.15. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

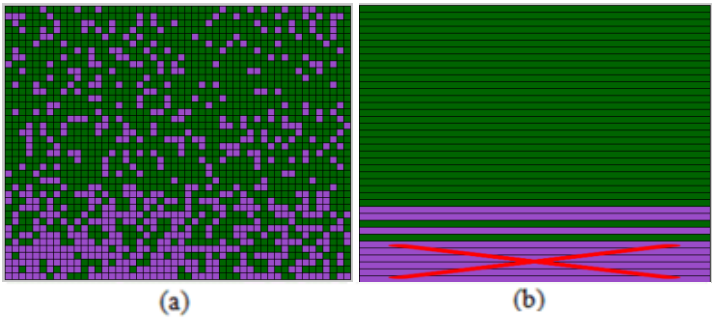
Tabel 5.15 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 5 Training 250 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	411 Potongan	367 Potongan
		Bukan Tumpal	239 Potongan	983 Potongan
	Waktu Pengerjaan		8.51 Menit	
k=3	Kelas Prediksi	Tumpal	357 Potongan	293 Potongan
		Bukan Tumpal	297 Potongan	1053 Potongan
	Waktu Pengerjaan		8.58 Menit	
k=5	Kelas Prediksi	Tumpal	337 Potongan	313 Potongan
		Bukan Tumpal	277 Potongan	1073 Potongan
	Waktu Pengerjaan		9.04Menit	

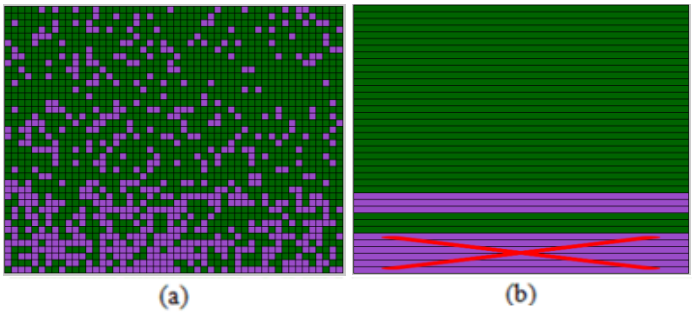
Hasil output uji coba data 5 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.49, Gambar 5.50 dan Gambar 5.51. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan tiga hasil bernilai bernilai kurang tepat, dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.49 Output Uji Coba Data 5 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.50 Output Citra Data 5 k=3 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.51 Output Citra Data 5 k=5 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

5.3.6. Uji Koefisien Matriks dan Kecepatan Data 6

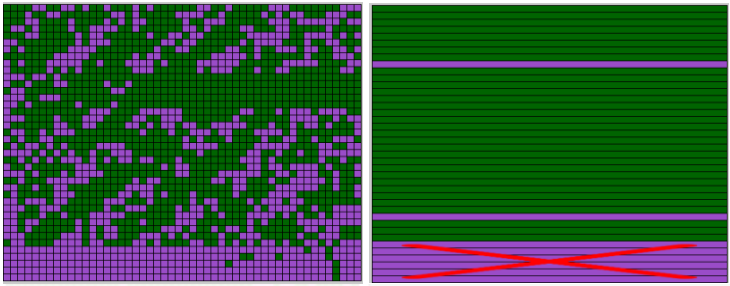
Gambar 5.52 Citra Data 6

Akurasi pada setiap uji coba data 6 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.16. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.16 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 6 Training 150 Sebelum Optimalisasi

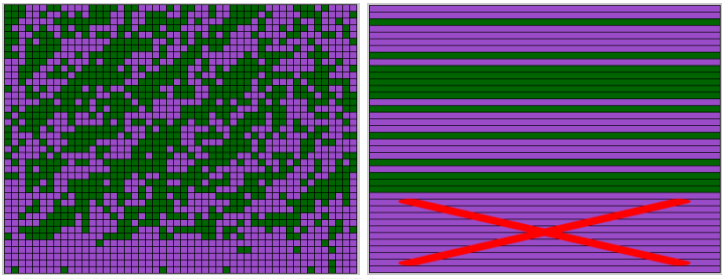
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	265 Potongan	877 Potongan
		Bukan Tumpal	35 Potongan	1088 Potongan
	Waktu Pengerjaan		5.23 Menit	
k=3	Kelas Prediksi	Tumpal	269 Potongan	824 Potongan
		Bukan Tumpal	31 Potongan	876 Potongan
	Waktu Pengerjaan		5.36 Menit	
k=5	Kelas Prediksi	Tumpal	271 Potongan	889 Potongan
		Bukan Tumpal	29 Potongan	811 Potongan
	Waktu Pengerjaan		5.73 Menit	

Hasil output uji coba data 6 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.53, Gambar 5.54, dan Gambar 5.55. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan satu hasil bernilai benar dan dua hasil bernilai bernilai kurang tepat, dan jumlah hasil predikisi baris *error* cukup banyak. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap kurang optimal.



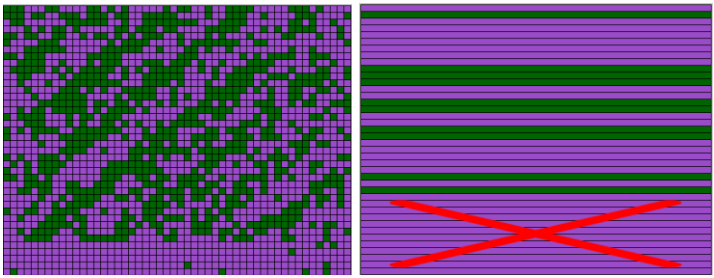
(a) (b)

Gambar 5.53 Output Uji Coba Data 6 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.54 Output Uji Coba Data 6 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

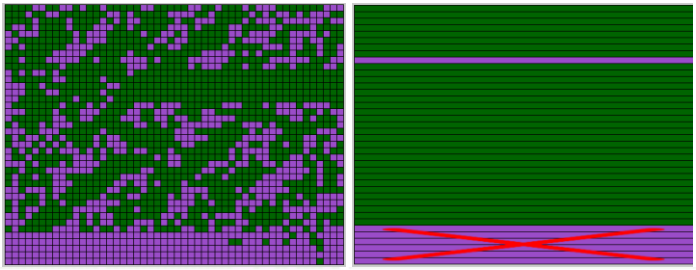
Gambar 5.55 Output Uji Coba Data 6 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 6 dengan variasi jumlah *training* 200 ditunjukkan pada Tabel 5.17. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.17 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 6 Training 200 Sebelum Optimalisasi

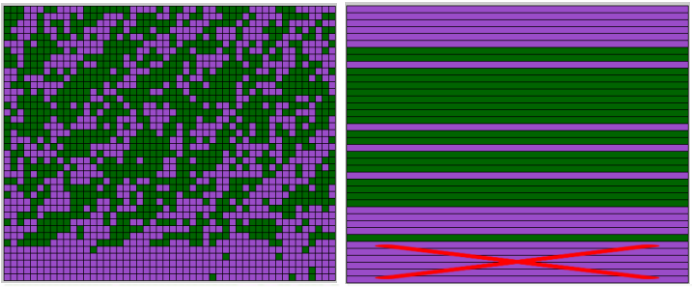
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	265 Potongan	599 Potongan
		Bukan Tumpal	35 Potongan	1101 Potongan
	Waktu Pengerjaan		6.38 Menit	
k=3	Kelas Prediksi	Tumpal	273 Potongan	771 Potongan
		Bukan Tumpal	27 Potongan	929 Potongan
	Waktu Pengerjaan		7.1 Menit	
k=5	Kelas Prediksi	Tumpal	276 Potongan	880 Potongan
		Bukan Tumpal	24 Potongan	820 Potongan
	Waktu Pengerjaan		7.48 Menit	

Hasil output uji coba data 6 dengan variasi jumlah training berjumlah 200 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.56, Gambar 5.57, dan Gambar 5.58. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan tiga hasil bernilai bernilai benar, dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



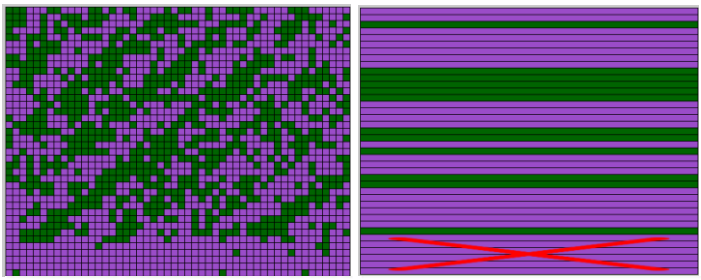
(a) (b)

Gambar 5.56 Output Uji Coba Data 6 k=1 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.



(a) (b)

Gambar 5.57 Output Citra Data 6 k=3 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.



(a) (b)

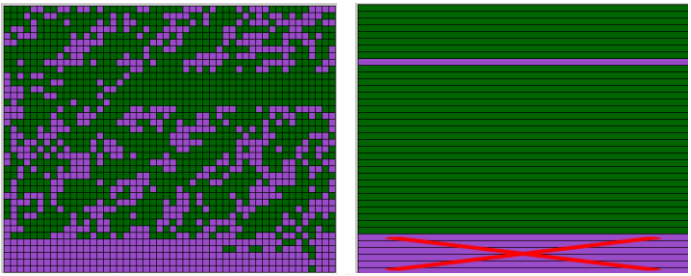
Gambar 5.58 Output Citra Data 6 k=5 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.

Akurasi pada setiap uji coba data 6 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.18. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.18 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 6 Training 250 Sebelum Optimalisasi

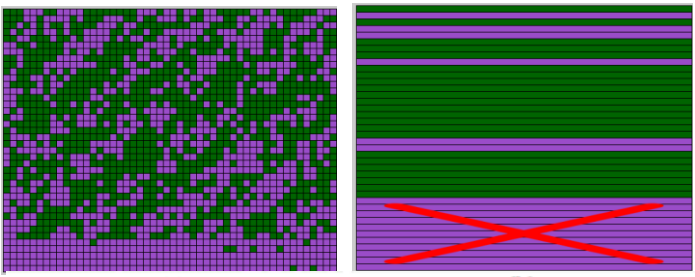
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	265 Potongan	580 Potongan
		Bukan Tumpal	35 Potongan	1120 Potongan
	Waktu Pengerjaan		9.22 Menit	
k=3	Kelas Prediksi	Tumpal	269 Potongan	739 Potongan
		Bukan Tumpal	31 Potongan	961 Potongan
	Waktu Pengerjaan		8.82 Menit	
k=5	Kelas Prediksi	Tumpal	268 Potongan	832 Potongan
		Bukan Tumpal	32 Potongan	868 Potongan
	Waktu Pengerjaan		8.88 Menit	

Hasil output uji coba data 6 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.59, Gambar 5.60 dan Gambar 5.61. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan satu hasil bernilai bernilai benar dan dua hasil bernilai kurang tepat, dan jumlah hasil predikisi baris *error* cukup banyak. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap kurang optimal.



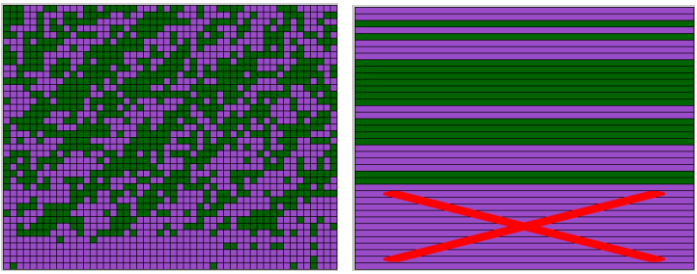
(a) (b)

Gambar 5.59 Output Uji Coba Data 6 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.60 Output Citra Data 6 k=3 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.61 Output Citra Data 6 k=5 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

5.3.7. Uji Koefisien Matriks dan Kecepatan Data 7

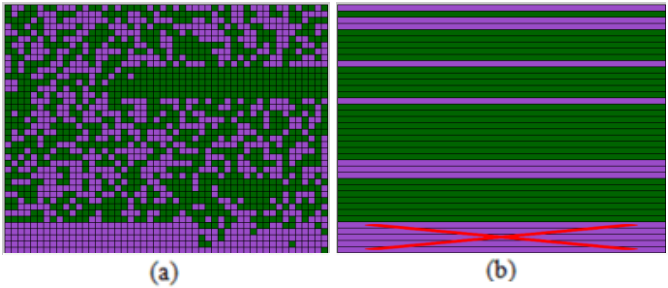
Gambar 5.62 Citra Data

Akurasi pada setiap uji coba data 7 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.19. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

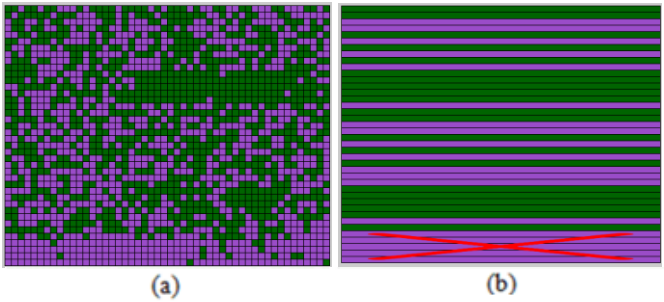
Tabel 5.19 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 7 Training 150 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	232 Potongan	734 Potongan
		Bukan Tumpal	18 Potongan	1016 Potongan
	Waktu Pengerjaan		5.16 Menit	
k=3	Kelas Prediksi	Tumpal	218 Potongan	766 Potongan
		Bukan Tumpal	32 Potongan	984 Potongan
	Waktu Pengerjaan		5.4 Menit	
k=5	Kelas Prediksi	Tumpal	215 Potongan	761 Potongan
		Bukan Tumpal	35 Potongan	989 Potongan
	Waktu Pengerjaan		5.03 Menit	

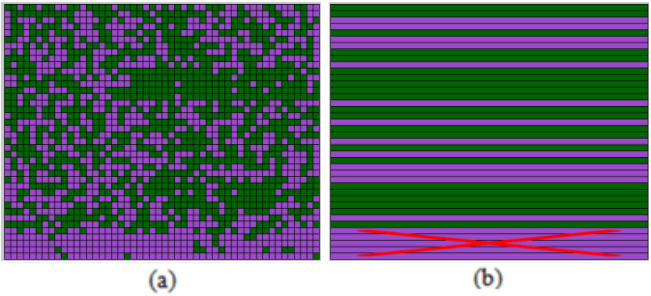
Hasil output uji coba data 7 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.63, Gambar 5.64, dan Gambar 5.65. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai bernilai benar dan jumlah hasil predikisi baris *error* cukup banyak. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.63 Output Uji Coba Data 7 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.64 Output Uji Coba Data 7 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



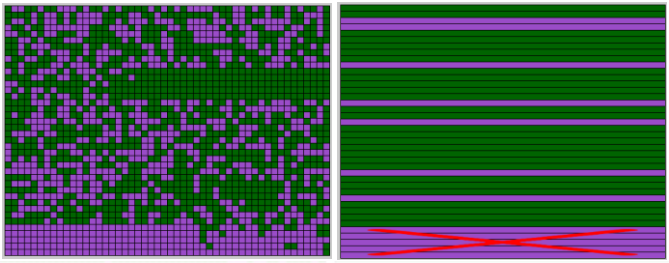
Gambar 5.65 Output Uji Coba Data 7 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 7 dengan variasi jumlah *training* 200 ditunjukkan pada Tabel 5.20. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.20 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 7 Training 200 Sebelum Optimalisasi

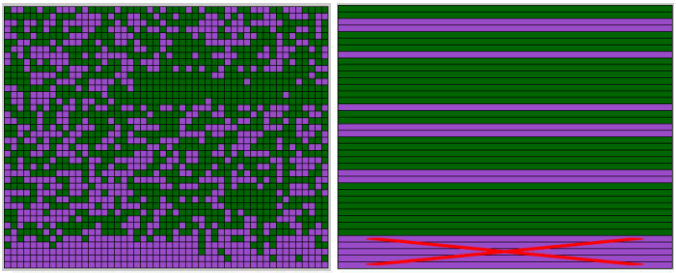
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	1053 Potongan	697 Potongan
		Bukan Tumpal	21 Potongan	229 Potongan
	Waktu Pengerjaan		6.94 Menit	
k=3	Kelas Prediksi	Tumpal	219 Potongan	714 Potongan
		Bukan Tumpal	31 Potongan	1036 Potongan
	Waktu Pengerjaan		7.02 Menit	
k=5	Kelas Prediksi	Tumpal	216 Potongan	738 Potongan
		Bukan Tumpal	34 Potongan	1012 Potongan
	Waktu Pengerjaan		6.16 Menit	

Hasil output uji coba data 7 dengan variasi jumlah training berjumlah 200 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.66, Gambar 5.67, dan Gambar 5.68. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai bernilai benar dan jumlah hasil predikisi baris *error* cukup banyak. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



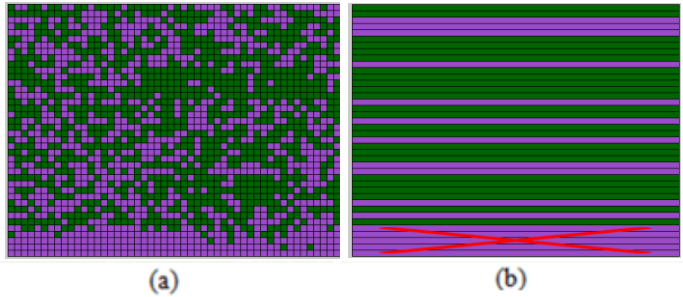
(a) (b)

Gambar 5.66 Output Uji Coba Data 7 k=1 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.



(a) (b)

Gambar 5.67 Output Citra Data 7 k=3 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.



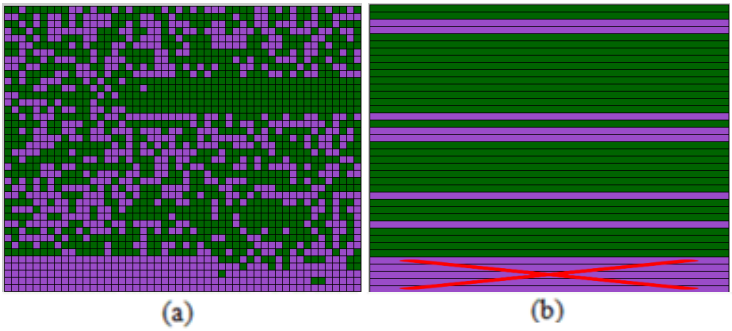
Gambar 5.68 Output Citra Data 7 k=5 Training 200. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 7 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.21. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

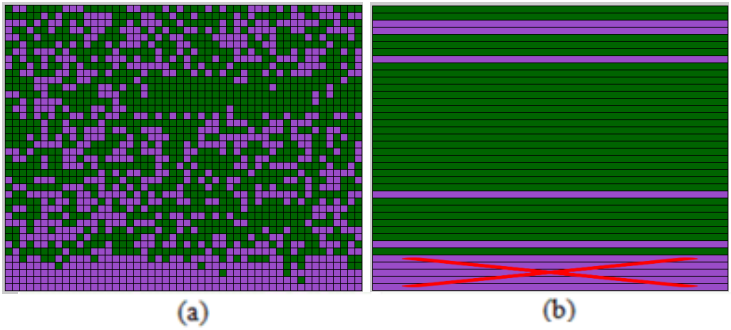
Tabel 5.21 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 7 Training 250 Sebelum Optimalisasi

			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	230 Potongan	693 Potongan
		Bukan Tumpal	20 Potongan	1057 Potongan
	Waktu Pengerjaan		9.49 Menit	
k=3	Kelas Prediksi	Tumpal	223 Potongan	678 Potongan
		Bukan Tumpal	27 Potongan	1072 Potongan
	Waktu Pengerjaan		8.07 Menit	
k=5	Kelas Prediksi	Tumpal	219 Potongan	681 Potongan
		Bukan Tumpal	31 Potongan	1069 Potongan
	Waktu Pengerjaan		7.5 Menit	

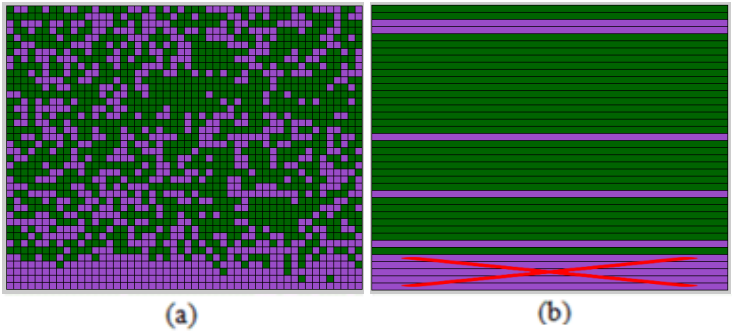
Hasil output uji coba data 7 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.69, Gambar 5.70 dan Gambar 5.71. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai bernilai benar dan jumlah hasil predikisi baris *error* cukup banyak. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



Gambar 5.69 Output Uji Coba Data 7 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.70 Output Citra Data 7 k=3 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



Gambar 5.71 Output Citra Data 7 k=5 Training 250. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.

5.3.8. Uji Koefisien Matriks dan Kecepatan Data 8

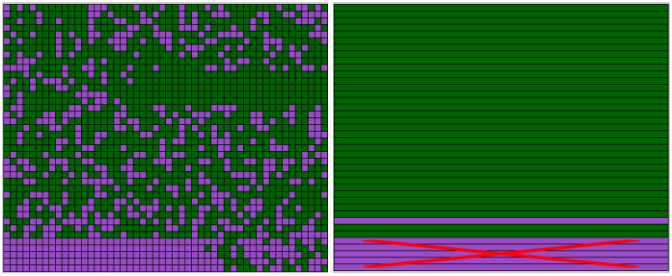
Gambar 5.72 Citra Data 8

Akurasi pada setiap uji coba data 8 dengan variasi jumlah *training* 150 ditunjukkan pada Tabel 5.22. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.22 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 8 Training 150 Sebelum Optimalisasi

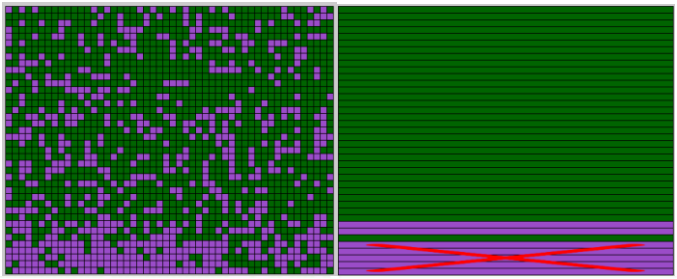
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	316 Potongan	430 Potongan
		Bukan Tumpal	184 Potongan	1070 Potongan
	Waktu Pengerjaan		4.41 Menit	
k=3	Kelas Prediksi	Tumpal	290 Potongan	419 Potongan
		Bukan Tumpal	210 Potongan	1081 Potongan
	Waktu Pengerjaan		6.25 Menit	
k=5	Kelas Prediksi	Tumpal	282 Potongan	401 Potongan
		Bukan Tumpal	218 Potongan	1099 Potongan
	Waktu Pengerjaan		5.33 Menit	

Hasil output uji coba data 8 dengan variasi jumlah training berjumlah 150 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.73, Gambar 5.74, dan Gambar 5.75. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai bernilai kurang tepat dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal



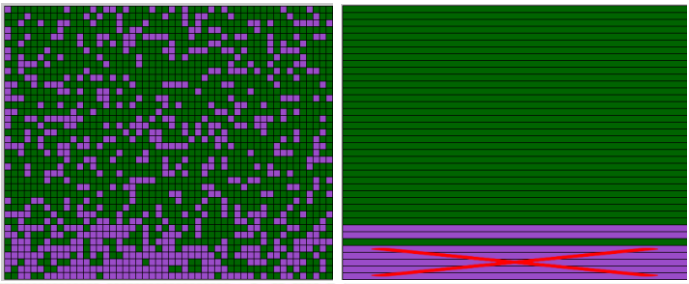
(a) (b)

Gambar 5.73 Output Uji Coba Data 8 k=1 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.74 Output Uji Coba Data 8 k=3 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

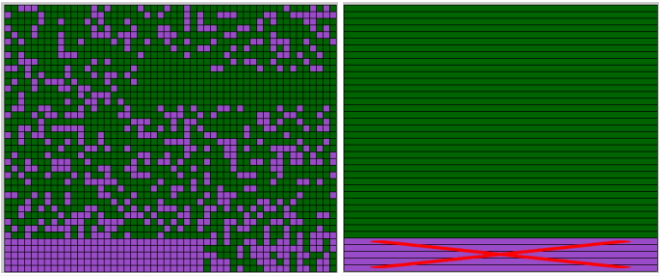
Gambar 5.75 Output Uji Coba Data 8 k=5 Training 150. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

Akurasi pada setiap uji coba data 8 dengan variasi jumlah *training* 200 ditunjukkan pada Tabel 5.23. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.23 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 8 Training 200 Sebelum Optimalisasi

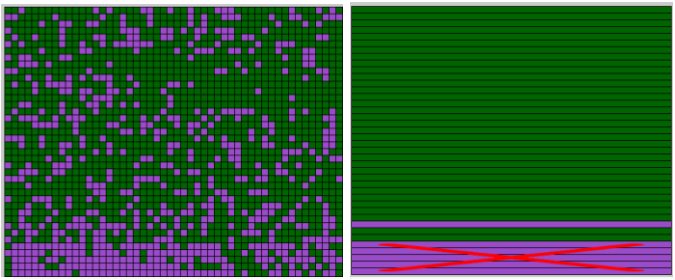
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	309 Potongan	401 Potongan
		Bukan Tumpal	191 Potongan	1099 Potongan
	Waktu Pengerjaan		7.23 Menit	
k=3	Kelas Prediksi	Tumpal	285 Potongan	389 Potongan
		Bukan Tumpal	215 Potongan	1111 Potongan
	Waktu Pengerjaan		7.8 Menit	
k=5	Kelas Prediksi	Tumpal	290 Potongan	353 Potongan
		Bukan Tumpal	210 Potongan	1147 Potongan
	Waktu Pengerjaan		7.68 Menit	

Hasil output uji coba data 8 dengan variasi jumlah training berjumlah 200 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.76, Gambar 5.77, dan Gambar 5.8. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai bernilai kurang tepat dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal



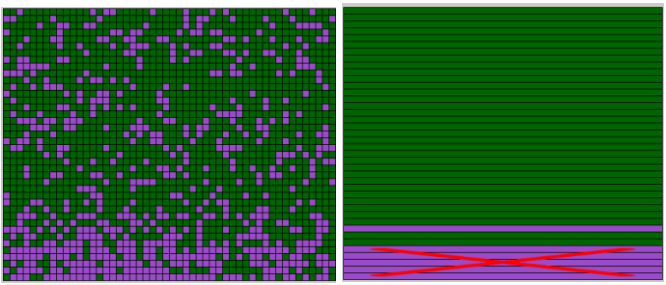
(a) (b)

Gambar 5.76 Output Uji Coba Data 8 k=1 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.



(a) (b)

Gambar 5.77 Output Citra Data 8 k=3 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.



(a) (b)

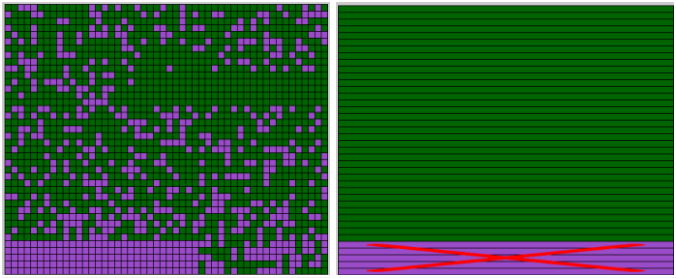
Gambar 5.78 Output Citra Data 8 k=5 Training 200. (a) *Sebelum Optimlisasi*. (b) *Setelah Optimalisasi*.

Akurasi pada setiap uji coba data 8 dengan variasi jumlah *training* 250 ditunjukkan pada Tabel 5.24. Tabel berisi kolom kelas sebenarnya yang menunjukkan kelas pola sebenarnya yang diprediksi oleh sistem, dan waktu pengerjaan yang menunjukkan lama sistem dalam menyelesaikan satu uji coba.

Tabel 5.24 Hasil Uji Coba Koefisien Matriks dan Kecepatan Data 8 Training 250 Sebelum Optimalisasi

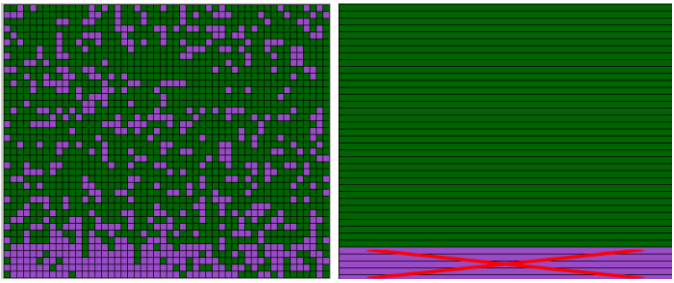
			Kelas Sebenarnya	
			Tumpal	Bukan Tumpal
k=1	Kelas Prediksi	Tumpal	307 Potongan	381 Potongan
		Bukan Tumpal	193 Potongan	1119 Potongan
	Waktu Pengerjaan		9.9 Menit	
k=3	Kelas Prediksi	Tumpal	275 Potongan	355 Potongan
		Bukan Tumpal	225 Potongan	1145 Potongan
	Waktu Pengerjaan		8.2 Menit	
k=5	Kelas Prediksi	Tumpal	275 Potongan	347 Potongan
		Bukan Tumpal	225 Potongan	1153 Potongan
	Waktu Pengerjaan		8.22 Menit	

Hasil output uji coba data 8 dengan variasi jumlah training berjumlah 250 dengan *k*=1, *k*=3 dan *k*=5 ditunjukkan pada Gambar 5.79, Gambar 5.80 dan Gambar 5.81. Terlihat hasil setelah optimalisasi (b) memberikan hasil yang lebih optimal dibandingkan sebelum optimalisasi (a) dengan ketiga hasil bernilai bernilai kurang tepat dan jumlah hasil predikisi baris *error* cukup sedikit. Sehingga dapat disimpulkan parameter pada percobaan ini dianggap cukup optimal.



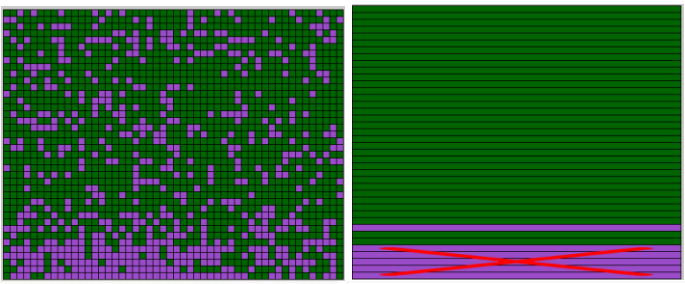
(a) (b)

Gambar 5.79 Output Uji Coba Data 8 k=1 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.80 Output Citra Data 8 k=3 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*



(a) (b)

Gambar 5.81 Output Citra Data 7 k=5 Training 250. (a) *Sebelum Optimlisasi.* (b) *Setelah Optimalisasi.*

5.4. Evaluasi Pengujian

Sub bab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Evaluasi pengujian ini menunjukkan data rekapitulasi berdasarkan hasil pengujian. Hasil pengujian menilai keefektifan tiap parameter uji coba pada aplikasi deteksi posisi tumpal pada citra kain batik dengan menggunakan fitur *Local Binary Pattern* (LBP) dan klasifikasi KNN. Data rekapitulasi pengujian berbasis *k* dapat dilihat pada Tabel 5.25 dan data rekapitulasi berbasis jumlah data *training* dapat dilihat pada Tabel 5.2

Tabel 5.25 Hasil Rekapitulasi Berbasis k

k	Akurasi Sebelum Optimalisasi (%)	Hasil Setelah Optimalisasi				Waktu Pengerjaan
		Benar	Kurang Tepat	Salah	Akurasi	
1	66.45%	19	5	0	100%	7.21 Menit
3	64.24%	15	8	1	95.83%	7.08 Menit
5	64.0%	13	9	2	91.67%	7.08 Menit

Tabel 5.26 Hasil Rekapitulasi Berbasis Jumlah Citra Training

Jumlah Citra *Training Tiap Pola*	Akurasi Sebelum Optimalisasi (%)	Hasil Setelah Optimalisasi				Waktu Pengerjaan
		Benar	Kurang Tepat	Salah	Akurasi	
150	63.4%	17	6	1	95.83%	5.17 Menit
200	65.36%	17	6	1	95.83%	7.32 Menit
250	66.01%	13	10	1	95.83%	8.88 Menit

Berdasarkan data pada Tabel 5.25, semakin tinggi nilai *k* semakin kecil akurasi sebelum optimalisasinya. Pada tabel ini juga terlihat bahwa hasil setelah optimalisasi berbanding lurus dengan akurasi sebelum optimalisasi. Nilai *k* juga tidak terlalu berpengaruh pada kecepatan kerja sistem. Sedangkan pada Tabel 5.26, semakin besar jumlah data *training* maka semakin banyak waktu yang diperlukan sistem untuk menjalankan tugasnya. Jumlah citra *training* 150 dan *training* 200 memiliki hasil setelah

optimalisasi terbaik yang sama persis hasilnya. Sedangkan citra data *training* 250 memiliki akurasi tertinggi meskipun jumlah hasil setelah optimalisasinya yang bernilai benar paling sedikit.

Tabel 5.27 Hasil Rekapitulasi Gabungan

k	Jumlah Citra *Training Tiap Pola*	Akurasi Sebelum Optimalisasi (%)	Hasil Setelah Optimalisasi				Waktu Pengerjaan
			Benar	Kurang Tepat	Salah	Akurasi	
1	150	65.74%	6	2	0	100%	5.02 Menit
3	150	61.96%	6	2	0	100%	5.28 Menit
5	150	62.49%	5	2	1	87.5%	5.21 Menit
1	200	66.58%	7	1	0	100%	7.31 Menit
3	200	64.84%	5	3	0	100%	7.29 Menit
5	200	64.67%	5	2	1	87.5%	7.37 Menit
1	250	67.04%	6	2	0	100%	9.29 Menit
3	250	65.93%	4	3	1	87.5%	8.68 Menit
5	250	65.07%	3	5	0	100%	8.67 Menit

Setelah melakukan rekapitulasi dengan gabungan dari kedua parameter uji coba yang dapat dilihat pada Tabel 5.27, gabungan parameter *k* dengan nilai 1 dan jumlah citra *training* 200 terbukti menghasilkan hasil setelah optimalisasi bernilai benar paling banyak melebihi parameter *k* bernilai 1 dengan jumlah citra *training* 250 dengan nilai akurasi sebelum optimalisasi tertinggi. Ketidak serasian akurasi sebelum optimalisasi dengan hasil setelah optimalisasi pada data *training* 250 dikarenakan jumlah data *training* yang terlalu banyak mengakibatkan subcitra yang dianggap benar tidak berfokus pada satu garis tapi lebih menyebar dalam quantity yang lebih sedikit.

Hasil uji coba pada citra data uji 8 yang dapat dilihat pada lampiran memberikan nilai hasil setelah optimalisasi terbanyak kurang tepat. Hal ini dikarenakan pada citra data 8 terdapat dua pola pada berbeda pada area tumpal. Sehingga ketika sistem mengambil data *training* yang berisi satu dari dua pola tersebut, pola kedua yang tidak terdapat pada data *training* tidak akan dapat dianggap sebagai tumpal. Sehingga dapat disimpulkan tumpal yang memiliki lebih dari satu pola akan lebih sulit untuk didteksi dibandingkan dengan tumpal yang hanya berpola satu.

BAB VI
KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

6.1. Kesimpulan

Dalam pengerjaan Tugas Akhir ini melalui tahap perancangan aplikasi, implementasi metode, serta uji coba. Sehingga mendapatkan kesimpulan sebagai berikut:

1. Penajaman citra dengan menggunakan deteksi tepi *Laplacian* sebelum ekstraksi fitur LBP dan pencarian jarak *euclidean distance,* memberikan hasil jarak yang lebih besar sehingga lebih mudah untuk menentukan *treshold*.
2. Optimalisasi dilakukan dengan menjadikan batasan masalah citra uji coba memiliki sebuah tumpal berbentuk melebar sebagai acuan dalam mengambil mayoritas kelas subcitra dalam satu baris.
3. Hasil klasifikasi dengan menggunakan KNN tanpa optimalisasi dianggap tidak cukup baik dalam mendeteksi letak posisi tumpal.
4. Semakin tinggi jumlah data citra *training* semakin tinggi hasil akurasi sebelum optimalisasi. Namun, hasi akurasi sebelum optimalisasi tidak berbanding lurus dengan hasil setelah optimalisasi.
5. Semakin kecil nilai *k* pada KNN semakin tinggi pula akurasi sebelum optimalisasinya, dan semakin banyak hasil setelah optimalisasi yang bernilai benar.

6. Nilai *k* tidak berpengaruh pada waktu pengerjaan sistem. Sedangkan jumlah data citra *training* berbanding lurus dengan waktu pengerjaan sistem
7. Gabungan parameter *k*=1 dan jumlah citra data *training* 200 memberikan hasil yang terbaik.

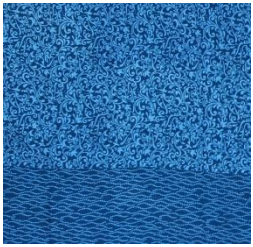
6.2. Saran

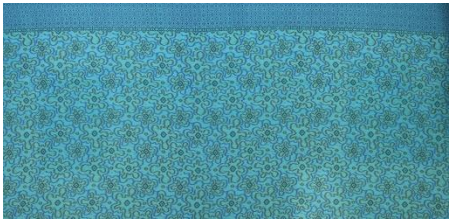
Saran yang diberikan untuk pengembangan sistem deteksi posisi tumpal pada citra kain batik dengan menggunakan fitur *Local Binary Pattern* (LBP), yaitu:

1. Perlu dilakukan uji coba lebih lanjut pada deteksi posisi tumpal pada citra kain batik dengan menggunakan metode klasifikasi lain untuk dapat ditemukan hasil yang lebih optimal.
2. Ditambahkan aturan optimalisasi yang lain untuk dapat meningkatkan hail yang lebih optimal.
3. Mencari cara agar tidak terlalu banyak menggunakan data *training* dikarenakan pembacaan data training yang banyak memakan waktu yang cukup lama.

LAMPIRAN

L.1. Dataset Uji Coba





L.2. Hasil Uji Koefisien Matriks dan Kecepatan Data 1

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 1	1	150	1354	646	67.7%	Benar	5.24 Menit
Citra 1	3	150	1338	662	66.9%	Benar	4.89 Menit
Citra 1	5	150	1353	647	67.65 %	Benar	5.05 Menit
Citra 1	1	200	1385	615	69.25 %	Benar	7.78 Menit
Citra 1	3	200	1395	605	69.75 %	Benar	7.48 Menit
Citra 1	5	200	1403	597	70.15 %	Benar	7.62 Menit
Citra 1	1	250	1381	619	69.05 %	Benar	9.6 Menit
Citra 1	3	250	1381	619	69.05 %	Benar	9.17 Menit
Citra 1	5	250	1389	611	69.45 %	Benar	10.8 Menit

L.3. Hasil Uji Koefisien Matriks dan Kecepatan Data 2

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 2	1	150	1172	828	58.6%	Benar	4.98 Menit
Citra 2	3	150	1107	913	55.35 %	Benar	5.31 Menit
Citra 2	5	150	1066	986	53.3%	Salah	5.1 Menit
Citra 2	1	200	1205	795	60.25 %	Benar	6.99 Menit

Citra 2	3	200	1152	848	57.6%	Benar	6.76 Menit
Citra 2	5	200	1108	892	55.4%	Salah	7.41 Menit
Citra 2	1	250	1228	772	61.4%	Benar	9.04 Menit
Citra 2	3	250	1206	794	60.3%	Benar	8.88 Menit
Citra 2	5	250	1170	830	58.5%	Kurang Tepat	7.71 Menit

L.4. Hasil Uji Koefisien Matriks dan Kecepatan Data 3

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 3	1	150	1385	615	69.25 %	Benar	5.31 Menit
Citra 3	3	150	1206	794	60.3%	Benar	5.41 Menit
Citra 3	5	150	1395	605	69.75 %	Benar	5.06 Menit
Citra 3	1	200	1322	678	66.1%	Benar	8.25 Menit
Citra 3	3	200	1322	678	66.1%	Benar	8.52 Menit
Citra 3	5	200	1328	672	66.4%	Benar	8.45 Menit
Citra 3	1	250	1296	704	64.8%	Benar	10.32 Menit
Citra 3	3	250	1306	694	65.3%	Benar	9.09 Menit
Citra 3	5	250	1294	706	64.7%	Benar	8.88 Menit

L.5. Hasil Uji Koefisien Matriks dan Kecepatan Data 4

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 4	1	150	1316	684	65.8%	Kurang Tepat	4.86 Menit
Citra 4	3	150	1276	724	63.8%	Benar	4.91 Menit
Citra 4	5	150	1259	741	62.95 %	Benar	5.66 Menit
Citra 4	1	200	1331	669	66.65 %	Benar	8.17 Menit
Citra 4	3	200	1302	698	65.1%	Kurang Tepat	6.58 Menit
Citra 4	5	200	1342	658	67.1%	Benar	7.65 Menit
Citra 4	1	250	1329	671	66.45 %	Benar	8.28 Menit
Citra 4	3	250	1301	699	65.05 %	Salah	8.67 Menit
Citra 4	5	250	1296	704	64.8%	Kurang Tepat	8.34 Menit

L.6. Hasil Uji Koefisien Matriks dan Kecepatan Data 5

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 5	1	150	1304	696	65.2%	Benar	4.97 Menit
Citra 5	3	150	1268	732	63.4%	Benar	4.72 Menit
Citra 5	5	150	1259	741	62.95 %	Benar	4.74 Menit
Citra 5	1	200	1352	648	67.6%	Benar	6.71 Menit

Citra 5	3	200	1360	640	68%	Kurang Tepat	7.04 Menit
Citra 5	5	200	1349	651	67.45 %	Kurang Tepat	6.52 Menit
Citra 5	1	250	1394	606	69.7%	Kurang Tepat	8.51 Menit
Citra 5	3	250	1410	590	70.5%	Kurang Tepat	8.58 Menit
Citra 5	5	250	1410	590	70.5%	Kurang Tepat	9.04 Menit

L.7. Hasil Uji Koefisien Matriks dan Kecepatan Data 6

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 6	1	150	1353	647	67.65 %	Benar	5.23 Menit
Citra 6	3	150	1145	855	57.25 %	Kurang Tepat	5.36 Menit
Citra 6	5	150	1082	918	54.1%	Kurang Tepat	5.73 Menit
Citra 6	1	200	1366	634	68.3%	Benar	6.38 Menit
Citra 6	3	200	1201	798	60.1%	Benar	7.1 Menit
Citra 6	5	200	1096	904	54.8%	Benar	7.48 Menit
Citra 6	1	250	1385	615	69.25 %	Benar	9.22 Menit
Citra 6	3	250	1230	770	61.5%	Kurang Tepat	8.82 Menit
Citra 6	5	250	1136	864	56.8%	Kurang Tepat	8.88 Menit

L.8. Hasil Uji Koefisien Matriks dan Kecepatan Data 8

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 7	1	150	1248	752	62.4%	Benar	5.16 Menit
Citra 7	3	150	1202	798	60.1%	Benar	5.4 Menit
Citra 7	5	150	1204	796	60.2%	Benar	5.03 Menit
Citra 7	1	200	1282	718	64.1%	Benar	6.94 Menit
Citra 7	3	200	1255	745	62.75 %	Benar	7.02 Menit
Citra 7	5	200	1228	772	61.4%	Benar	6.16 Menit
Citra 7	1	250	1287	713	64.35 %	Benar	9.49 Menit
Citra 7	3	250	1295	705	64.75 %	Benar	8.07 Menit
Citra 7	5	250	1288	712	64.4%	Benar	7.5 Menit

L.9. Hasil Uji Koefisien Matriks dan Kecepatan Data 7

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 7	1	150	1248	752	62.4%	Benar	5.16 Menit
Citra 7	3	150	1202	798	60.1%	Benar	5.4 Menit
Citra 7	5	150	1204	796	60.2%	Benar	5.03 Menit

Citra 7	1	200	1282	718	64.1%	Benar	6.94 Menit
Citra 7	3	200	1255	745	62.75 %	Benar	7.02 Menit
Citra 7	5	200	1228	772	61.4%	Benar	6.16 Menit
Citra 7	1	250	1287	713	64.35 %	Benar	9.49 Menit
Citra 7	3	250	1295	705	64.75 %	Benar	8.07 Menit
Citra 7	5	250	1288	712	64.4%	Benar	7.5 Menit

L.10. Hasil Uji Koefisien Matriks dan Kecepatan Data 8

Citra Uji	k	Jumlah Citra *Training Tiap Pola*	Jumlah Benar	Jumlah Salah	Nilai (%)	Hasil Setelah Optimalisasi (Benar/Salah /Kurang Tepat)	Waktu
Citra 8	1	150	1386	614	69.3%	Kurang Tepat	4.41 Menit
Citra 8	3	150	1371	629	68.55 %	Kurang Tepat	6.25 Menit
Citra 8	5	150	1381	619	69.05 %	Kurang Tepat	5.33 Menit
Citra 8	1	200	1408	592	70.4%	Kurang Tepat	7.23 Menit
Citra 8	3	200	1396	604	69.8%	Kurang Tepat	7.8 Menit
Citra 8	5	200	1437	563	71.85 %	Kurang Tepat	7.68 Menit
Citra 8	1	250	1426	574	71.3%	Kurang Tepat	9. 9 Menit
Citra 8	3	250	1420	580	71%	Kurang Tepat	8.2 Menit
Citra 8	5	250	1428	572	71.4%	Kurang Tepat	8.22 Menit

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

[1] Edin Suhaedin Purnama Giri, “Ragam Hias Kreasi” Fakultas Bahasa dan Seni, Universitas Negeri Yogyakarta. 2000.

[2] Adam Fauzi, " Pendeteksian Tepi Menggunakan Teknik Operator Robert , Laplacian dan Compass" Program Studi Teknik Geomatika, Jurusan Teknologi Infrastruktur dan Kewilayahan, Institut Teknologi Sumatera, Bandarlampung, 2015.

[3] Wulandari, A., “Batik Nusantara (Makna Filosofi, Cara Pembuatan & Industri Batik)” Yogyakarta, 2011.

[4] Amin Padmo A.M, Murinto, "Segmentasi Citra Batik Berdasarkan Fitur Tekstur Menggunakan Metode Filter Gabor dan K-means Clustering" Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Yogyakarta, 2016.

[5] Arrie Kurniawardhani, Isye Arieshanti, Nanik Suciati, “Klasifikasi Citra Batik Menggunakan Metode Ekstraksi Ciri yang Invariant Terhadap Rotasi”, JUTI vol. 12. No. 2, pp. 48 – 60 Juli 2014.

[6] Mitchell, T. "Machine Learning" New York, 1997.

[7] Larose, D.T."Discovering Knowledge in Data: An Introduction to Data Mining", John Wiley & Sons, Inc, 2005.

[8] Dr.Suvarna Nandyal, Poorvika M. Harsoor, “Perceptual Image Segmentation Using Local Binary Pattern Algorithm for Analysis of Psoriasis Skin Image” International Journal of Computer Science and Information Technologies, Vol. 7 (5), 2016.

[9] Rismawan, T., Irawan, A.W., Prabowo, W. & Usuma Dewi, S. "Sistem Pendukung Keputusan berbasis Pocket PC Sebagai Penentu Status Gizi Menggunakan Metode KNN (K-Nearest Neighbor)" Jurnal Teknoin, Vol. 12, No. 2, Juni 2008.

[10] Elia Kurniawati1, Jangkung Raharjo, Ir., M.T., Inung

Dr.Suvarna Nandyal, Poorvika M. Harsoor, “Perceptual Image Segmentation Using Local Binary Pattern Algorithm for Analysis of Psoriasis Skin Image” International Journal of Computer Science and Information Technologies, Vol. 7 (5), 2016.

[11] Casey Reas, Ben Fry, "Getting Started with Processing" O’Reilly Media, Inc, 2010.

BIODATA PENULIS

Rafidah Rafalia, lahir di Surabaya pada tanggal 24 Maret 1997. Penulis menempuh pendidikan mulai dari TK Cendekia Sidoarjo (2001-2003), SD Muhammadiyah 1 Sidoarjo (2003-2009), SMP Negeri 2 SIdoarjo (2009-2012), SMA Negeri 2 Sidoarjo (2012-2015) dan saat ini melanjutkan studinya di Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Penulis aktif dalam organisasi dan kepanitiaan Himpunan Mahasiswa Teknik Computer (HMTC) dan Schematics. Dalam menyelesaikan pendidikan sarjananya, penulis mengambil bidang minat Komputasi Cerdas dan Visi (KCV). Komunikasi kepada penulis dengan senang hati dapat dilayani melalui alamat ***e-mail*: rafidahlia6@gmail.com**.