

PENENTUAN d DAN θ PADA GLCM UNTUK KLASIFIKASI TUMOR PAYUDARA PADA GAMBAR ROI MAMMOGRAM MENGGUNAKAN METODE BPNN

Untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun oleh:

Syam Julio A. Sarosa

NIM:176150100111027



MAGISTER ILMU KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

MALANG

2019

PENENTUAN d DAN θ PADA GLCM GAMBAR ROI MAMMOGRAPHY

Diajukan untuk memperoleh

Syam NIM: 1

Tesis ini telah diperbaiki

Dosen Pembimbing I

Dr. Eng. Fitri Utaminingrum, S.T.,
NIP: 19820710 200812 2 001

Ketua Jurusan

Tri Astoto Kurniadi, S.T., M.T.
NIP: 19730101 200812 2 001

PENGESAHAN

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun Oleh :

Syam Julio Akbar Sarosa
NIM:176150100111027

Tesis ini telah diuji dan dinyatakan lulus pada

26 April 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Fitri Utaminingrum, S.T., M.T. Dr. Eng. Fitra A. Bachtiar, S.T., M.Eng.

NIP: 19820710 200812 2 001

NIK: 2012018406281001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah tesis ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disisipi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah tesis ini dapat dibuktikan terdapat unsur-unsur plagiari, saya bersedia tesis ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 Mei 2019

Syam Julio Akbar Sarosa

NIM: 176150100111027

Repository Universitas Brawijaya
Repository Universitas Brawijaya

KATA PENGANTAR

Penulis panjatkan ke hadirat Allah SWT, karena hanya dengan rahmat dan bimbinganNya Penulis dapat menyelesaikan tesis dengan judul **“PENENTUAN d DAN θ PADA GLCM UNTUK KLASIFIKASI TUMOR PAYUDARA PADA GAMBAR ROI MAMMOGRAM MENGGUNAKAN METODE BPNN”**.

Penulis sangat bersyukur atas dukungan serta bantuan dari berbagai pihak selama proses penulisan tesis. Penulis juga menyampaikan penghargaan lewat rasa hormat dan terimakasih kepada semua pihak yang telah ikut andil dan berkontribusi, baik dalam bentuk dukungan tenaga, pikiran, maupun moral selama penulisan tesis sehingga dapat terselesaikannya tesis ini. Pihak-pihak tersebut antara lain:

1. Ibu Dr. Eng. Fitri Utaminingrum, S.T., M.T. dan Dr. Eng. Fitra A. Bachtiar, S.T., M.Eng. selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan ilmu, saran dan dukungan moral dalam penyusunan tesis ini.
2. Orang tua Penulis, M. Sarosa dan Yunia M. S. yang telah memberi semangat, motivasi dan dukungan moril dan material kepada Penulis.
3. Seluruh dosen Informatika/Illu Komputer Universitas Brawijaya yang telah memberikan tambahan wawasan kepada Penulis.
4. Seluruh civitas akademika Informatika/Illu Komputer Universitas Brawijaya yang telah membantu selama penulisan kripsi ini.
5. Fika Farha Dina dan Reza Rahardian, dan semua teman-teman terutama Magister Informatika 2017, terima kasih atas segala bantuan dan dukungannya.
6. Semua pihak yang tidak dapat Penulis sebutkan satu per satu baik yang terlibat secara langsung maupun tidak langsung demi terselesaikannya tesis ini.

Sebagai manusia biasa, penulis sadar bahwa begitu banyak kekurangan entah itu disengaja atau tidak dari ketidakmampuan penulis menulis tesis ini. Oleh karena itu Penulis menerima saran dan kritik yang bertujuan untuk memperbaiki tesis ini agar bisa lebih baik. Penulis berharap tesis ini dapat bermanfaat bagi penulis sendiri dan bagi semua pihak.

Malang, 22 Mei 2019

Syam Julio Akbar Sarosa

NIM: 176150100111027

ABSTRAK

Kanker payudara adalah kanker paling umum dan paling mematikan pada wanita. Pada tahun 2018, diperkirakan ada 22.692 wanita Indonesia meninggal akibat kanker payudara. Bagi wanita yang sudah terserang kanker payudara, tingkat pemulihan dan tingkat kelangsungan hidup dapat ditingkatkan dengan cara deteksi dini melalui pemeriksaan rutin. Cara deteksi dini kanker payudara yang paling efektif adalah melalui mammografi. Proses mammografi menghasilkan gambar mammogram. Gambar mammogram akan dibaca oleh radiologis. Masalah yang umum ditemui radiologis ketika membaca mammogram adalah perbedaan bentuk dan struktur jaringan payudara setiap wanita. Selain itu, efisiensi radiologis dalam membaca mammogram berkaitan erat dengan pengalaman dan beban kerja. Dalam kasus terburuk, kesalahan pembacaan mammogram bisa menyebabkan kematian pasien. Untuk menyelesaikan permasalahan itu, dalam penelitian ini dikembangkan sebuah sistem *Computer-Aided Diagnosis* (CADx) untuk membantu radiologis mendeteksi dan mengklasifikasikan kelainan dalam payudara. Metode yang diusulkan dalam penelitian ini mengkombinasikan algoritme *Gray-Level Co-occurrence Matrix* (GLCM) dan algoritme *Backpropagation Neural Network* (BPNN) untuk mengklasifikasikan pasien kedalam kelas normal-abnormal berdasarkan gambar mamogram. GLCM digunakan untuk mengekstrak fitur dari mammogram. Fitur yang dihasilkan GLCM akan dijadikan masukkan pada BPNN. GLCM memiliki dua parameter, yaitu d dan θ . Dalam penelitian ini, dicari kombinasi d dan θ pada GLCM yang bisa menghasilkan performa BPNN terbaik. Agar bisa mendapat hasil optimal, *learning rate* dan jumlah *epoch* BPNN harus diuji terlebih dahulu. Hasilnya, *learning rate* dan jumlah *epoch* yang memberikan nilai MSE terkecil adalah 0,1 dan 40.000. Pengujian *learning rate* dan jumlah *epoch* dilakukan untuk semua kombinasi d dan θ GLCM. Penelitian ini menggunakan data yang disediakan oleh *Mammography Imaging Analysis Society* (MIAS). Hasilnya, penelitian ini memiliki *Accurration* sebesar 96,60%, *Sensitivity* sebesar 97,72% dan *Specificity* sebesar 93,53%. Hasil tersebut didapatkan ketika $d = [1,2]$ dan $\theta = 135^\circ$.

Kata kunci: Kanker payudara, Mammogram, MIAS dataset, GLCM, BPNN

PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR	iii
ABSTRAK	iv
DAFTAR ISIERSitas Brawijaya	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	x
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	5
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Tumor Payudara	8
2.2.1 Pengertian	8
2.2.2 Faktor yang Mempengaruhi Tumor Payudara	9
2.2.3 Deteksi Dini Tumor Payudara	9
2.3 MIAS	11
2.3.1 Kelas Mammogram	11
2.4 <i>Region of Interest</i> (ROI)	15
2.5 <i>Histogram Equalization</i>	15
2.5.1 <i>Contrast Limited Adaptive Histogram Equalization</i> (CLAHE)	16
2.6 <i>Median Filter</i>	20
2.7 <i>Gray-level Co-occurrence Matrix</i> (GLCM)	20
2.8 <i>Back Propagation Neural Network</i> (BPNN)	22
2.9 Pengujian	25
2.9.1 <i>Accururation</i>	25
2.9.2 <i>Sensitivity</i>	26
2.9.3 <i>Specificity</i>	26
BAB 3 METODOLOGI	27
3.1 Tahapan Penelitian	27

DAFTAR ISI

Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	3.1.1 Studi Literatur
Repository Universitas Brawijaya	3.1.2 Perancangan
Repository Universitas Brawijaya	3.2 Analisa Kebutuhan
Repository Universitas Brawijaya	3.2.1 Kebutuhan <i>Software</i>
Repository Universitas Brawijaya	3.2.2 Kebutuhan <i>Hardware</i>
Repository Universitas Brawijaya	3.3 Data yang digunakan
Repository Universitas Brawijaya	3.4 Skenario Pengujian
Repository Universitas Brawijaya	3.4.1 Pengujian Parameter BPNN
Repository Universitas Brawijaya	3.4.2 Pengujian Kombinasi α dan θ
Repository Universitas Brawijaya	3.4.3 Perbandingan Metode yang Diusulkan dengan Metode Pembanding
BAB 4 PERANCANGAN DAN IMPLEMENTASI	35
4.1 Perancangan	35
4.1.1 <i>Preprocessing</i>	36
4.1.2 <i>Feature Extraction</i>	44
4.1.3 BPNN	48
4.2 Implementasi	56
4.2.1 <i>Preprocessing</i>	56
4.2.2 <i>Feature Extraction</i>	58
4.2.3 BPNN	60
4.2.4 <i>User Interface</i>	63
BAB 5 HASIL DAN PEMBAHASAN	67
5.1 Pengujian Parameter BPNN	67
5.1.1 Pengujian <i>Learning Rate</i>	67
5.1.2 Pengujian Jumlah Epoch	73
5.2 Pengujian Kombinasi α dan θ	82
5.3 Perbandingan Metode yang Diusulkan dengan Algoritme Pembanding	84
BAB 6 Penutup	86
6.1 Kesimpulan	86
6.2 Saran	86
DAFTAR RUJUKAN	xii

Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	Repository Universitas Brawijaya
Repository Universitas Brawijaya	DAFTAR GAMBAR
Gambar 2.1 Kelenjar payudara (sumber: (Dang, 2018)).....	8
Gambar 2.2 Ilustrasi pembuatan gambar mammogram (Sumber: (Institute, 2019)	10
Gambar 2.3 Contoh gambar mammogram.....	10
Gambar 2.4 Gambar MIAS dataset (a) Kelas Abnormal, (b) Kelas Normal.....	11
Gambar 2.5 Mammogram <i>calcification</i>	12
Gambar 2.6 Mammogram <i>well-defined/circumscribed masses</i>	12
Gambar 2.7 Mammogram <i>spiculated masses</i>	13
Gambar 2.8 Mammogram <i>ill-defined masses</i>	13
Gambar 2.9 Mammogram <i>architectural distortion</i>	13
Gambar 2.10 Mammogram <i>asymmetry</i>	14
Gambar 2.11 Mammogram normal	14
Gambär 2.12 Contoh pengambilan ROI dari gambar mammograms.....	15
Gambar 2.13. Mengubah gambar jadi histogram (a) Gambar grayscale, (b) Histogram	15
Gambar 2.14 <i>Histogram Equalization</i> (a) Sebelum proses <i>histogram equalization</i> , (b) Setelah proses <i>histogram equalization</i> (Sumber : (doxygen.org, 2015))	16
Gambar 2.15 <i>Histogram equalization</i> (a) Sebelum proses <i>histogram equalization</i> , (b) Setelah proses <i>histogram equalization</i> (Sumber : (doxygen.org, 2015))	17
Gambar 2.16 Gambar F dengan dimensi 512 × 512.....	17
Gambär 2.17 Struktur region gambar F yang dibagi kedalam 64 region berukuran sama	18
Gambar 2.18 (a) IR dengan semua region tetangganya. (b) Pixel p dari kuadran 1 dari (i,j) region dan relasinya dengan titik tengah empat tetangga terdekatnya. (Sumber: (Reza, 2004)).....	18
Gambar 2.19. (a) BR dengan semua region tetangganya. (b) Pixel p dari kuadran 2 dari (i,j) region dan relasinya dengan titik tengah dua tetangga terdekatnya. (Sumber: (Reza, 2004)).....	19
Gambar 2.20. CR pada posisi kiri atas dan struktur ketetanggaannya (Sumber: (Reza, 2004)).	19
Gambar 2.21 Hasil CLAHE pada Gambar 2.15 (a) (Sumber : (doxygen.org, 2015))	20
Gambar 2.22 Urutan nilai <i>grayscale</i> Gambar 2.13 (a).....	20
Gambar 2.23 Kombinasi D dan Θ dalam GLCM	21

Gambar 5.3 Hasil pengujian *Learn*
Sumbu X adalah *Learning Rate* dan

Gambar 5.4 Hasil pengujian *Learn*
Sumbu X adalah *Learning Rate* dan

Gambar 5.5 Hasil pengujian jumlah
X adalah jumlah *Epoch* (dalam ribu)

Gambar 5.6 Hasil pengujian jumlah
Sumbu X adalah jumlah *Epoch* (dalam ribu)

Gambar 5.7 Hasil pengujian jumlah
Sumbu X adalah jumlah *Epoch* (dalam ribu)

Gambar 5.8 Hasil pengujian jumlah
Sumbu X adalah jumlah *Epoch* (dalam ribu)

DAFTAR TABEL

Tabel 2.1 Kajian pustaka klasifikasi gambar menggunakan pendekatan <i>feature extraction</i>	6
Tabel 2.2 Detail data MIAS.....	11
Tabel 3.1 Kombinasi 1.....	31
Tabel 3.2 Kombinasi 2.....	31
Tabel 3.3 Kombinasi 3.....	31
Tabel 3.4 Kombinasi 4.....	31
Tabel 3.5 Kombinasi 5.....	32
Tabel 3.6 Tabel pengujian <i>Learning Rate</i>	32
Tabel 3.7 Tabel pengujian jumlah <i>Epoch</i>	33
Tabel 3.8 Tabel pengujian kombinasi d dan θ	33
Tabel 3.9 Perbandingan metode yang diusulkan dengan metode pembanding ..	34
Tabel 4.1 Hasil BPNN.....	55
Tabel 5.1 Detail hasil pengujian <i>Learning Rate</i> untuk $d = [1,2,3,4]$ dan $\theta = 00$	68
Tabel 5.2 Detail hasil pengujian <i>Learning Rate</i> untuk $d = [1,2,3,4]$ dan $\theta = 450$	69
Tabel 5.3 Detail hasil pengujian <i>Learning Rate</i> untuk $d = [1,2,3,4]$ dan $\theta = 900$	70
Tabel 5.4 Detail hasil pengujian <i>Learning Rate</i> untuk $d = [1,2,3,4]$ dan $\theta \neq 1350$	71
Tabel 5.5 Detail hasil pengujian <i>Learning Rate</i>	72
Tabel 5.6 Detail hasil pengujian jumlah <i>Epoch</i> untuk $d = [1,2,3,4]$ dan $\theta = 00$	74
Tabel 5.7 Detail hasil pengujian jumlah <i>Epoch</i> untuk $d = [1,2,3,4]$ dan $\theta = 450$	76
Tabel 5.8 Detail hasil pengujian jumlah <i>Epoch</i> untuk $d = [1,2,3,4]$ dan $\theta = 900$	77
Tabel 5.9 Detail hasil pengujian jumlah <i>Epoch</i> untuk $d = [1,2,3,4]$ dan $\theta \neq 1350$	79
Tabel 5.10 Detail hasil pengujian jumlah <i>Epoch</i>	80
Tabel 5.11 Detail waktu komputasi pengujian jumlah <i>Epoch</i>	81
Tabel 5.12 Detail hasil pengujian <i>Accururation</i>	82
Tabel 5.13 Detail hasil pengujian <i>Sensitivity</i>	83

Tabel 5.14 Detail hasil pengujian Spesifikasi
Tabel 5.15 Perbandingan metode yang
dilakukan

Tabel 5.14 Detail hasil pengujian *Specificity* 84
 Tabel 5.15 Perbandingan metode yang diusulkan dengan metode pembanding 85

Repository Universitas Brawijaya
Repository Universitas Brawijaya

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kanker payudara adalah salah satu penyakit paling mematikan bagi wanita. Menurut (Gco.iarc.fr, 2018), pada tahun 2018 diperkirakan ada 626.679 wanita meninggal akibat kanker payudara diseluruh dunia. Dari jumlah tersebut, diperkirakan sebanyak 22.692 adalah wanita Indonesia. Bagi wanita yang sudah terserang kanker payudara, tingkat pemulihan dan tingkat kelangsungan hidup dapat ditingkatkan dengan cara deteksi dini melalui pemeriksaan rutin. Cara deteksi dini kanker payudara yang paling efektif adalah mammografi (Nezhadian & Rashidi, 2017). Proses mammografi menghasilkan gambar yang disebut mammogram. Mammogram nanti akan dibaca oleh radiologis. Tantangan yang sering ditemui radiologis ketika membaca mammogram adalah perbedaan bentuk dan struktur jaringan payudara setiap wanita (Ball & Bruce, 2007). Dalam beberapa kasus, kelainan dalam payudara juga bisa tersamarkan oleh struktur jaringan payudara (Verma, McLeod, & Klevansky, 2010). Selain itu, efisiensi radiologis dalam membaca mammogram berkaitan erat dengan pengalaman dan beban kerja (Elmore, et al., 2009). Kesalahan pembacaan mammogram dapat menyebabkan biopsi yang tidak diperlukan, bahkan lebih buruk lagi, kanker yang tidak terdeteksi. Dalam kasus terburuk, keterlambatan deteksi kanker payudara dapat mengakibatkan kematian pasien. Oleh karena itu, perlu dikembangkan sebuah sistem pendukung keputusan yang bisa membantu radiologis mendeteksi dan mengklasifikasikan kelainan dalam payudara. Sistem seperti ini biasa disebut sistem *Computer Aided Diagnosis* (CADx).

Pengembangan sistem CADx pada studi kasus klasifikasi tumor payudara sudah banyak dilakukan. (Kral & Lenc, 2016) melakukan penelitian yang menggunakan *Local Binary Pattern* (LBP) untuk mengekstraksi fitur pada gambar mammogram dan SVM pada tahap klasifikasi normal-abnormal pada dataset *Mammographic Image Analysis Society* (MIAS). Kombinasi metode *feature extraction Gray-level Co-occurrence Matrix* (GLCM) dan metode klasifikasi *Radial Basis Function Neural Network* (RBFNN) dilakukan oleh (Pratiwi, Alexander, Harefa, & Nanda, 2015) untuk mengklasifikasi dataset MIAS kedalam kelas normal-abnormal. *Laws' texture energy measures* (LAWS) dan SVM digunakan oleh (Setiawan, Elysia, Wesley, & Purnama, 2015) untuk mengklasifikasi kasus normal-abnormal dari dataset MIAS.

GLCM, LAWS dan LBP termasuk kedalam metode *feature extraction* berdasarkan tekstur gambar. LAWS dan LBP rentan terhadap rotasi (Humeau-Heurtier, 2019). Hal ini berlawanan dengan sifat struktur jaringan payudara yang berbeda-beda (Ball & Bruce, 2007). Artinya, struktur jaringan payudara tidak memiliki bentuk serta “arah” yang pasti. Selain itu, LBP memiliki kekurangan lain, yaitu sangat sensitif terhadap *noise* dan *blurring*. Dua kelemahan ini tidak dimiliki oleh GLCM. Perbandingan performa antara GLCM dan beberapa metode analisa tekstur lain pernah dilakukan, antara lain oleh (Materka & Strzelecky, 1998) dan

Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository
Repository
Repository
Repository
Repository
Repository

Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository
Repository

Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository

Repository Universitas Brawijaya
(GRIBKOV, et al.). Hasilnya, GLCM memiliki performa yang lebih baik dibanding metode lain seperti LAWS, Gaussian Markov Random Fields, Gabor Filtering, dan lain-lain. Oleh karena itu, dalam penelitian ini GLCM dipilih sebagai metode analisa tekstur untuk mengekstrak fitur dari gambar mammogram.

GLCM adalah metode statistik orde kedua untuk analisa tekstur. GLCM termasuk metode statistik orde kedua karena GLCM mempertimbangkan ketetanggaan antar pixel dalam gambar. Luaran yang dihasilkan GLCM adalah matriks *Co-occurrence* atau matriks GLCM. Matriks GLCM memiliki dua parameter, yaitu jarak (d) dan sudut ketetanggaan (θ) antara pixel utama dan pixel tetangga. Nilai d mulai dari 1 sampai setengah gambar. Semakin jauh jarak d , semakin kecil juga hubungan yang dimiliki antara piksel utama dan piksel tetangga. Hal ini menyebabkan hasil analisa tekstur semakin kasar (Humeau-Heurtier, 2019). Nilai θ adalah $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$. Hal ini dipengaruhi oleh bentuk matriks, dimana setiap elemen didalam matriks memiliki tetangga disudut $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$. Dengan menerapkan aturan simetris, setiap elemen matriks GLCM untuk sudut $0^\circ = 180^\circ, 45^\circ = 225^\circ, 90^\circ = 270^\circ, 135^\circ = 315^\circ$. Oleh karena itu, nilai θ yang sering digunakan dalam GLCM adalah $0^\circ, 45^\circ, 90^\circ, 135^\circ$. Perbedaan kombinasi d dan θ akan menghasilkan matriks GLCM yang berbeda juga. Sebagai contoh, asumsikan pixel utama berada di posisi (2,2), kombinasi $d = 1$ dan $\theta = 0^\circ$ akan melihat pixel tetangga di posisi (1,2) dan (3,2). Jika menggunakan kombinasi $d = 1$ dan $\theta = 45^\circ$, pixel utama berada di posisi (2,2), pixel tetangga yang dilihat ada di posisi (1,3) dan (3,1). Setelah matriks GLCM dibuat, fitur gambar akan diekstrak dari matriks tersebut.

GLCM sudah banyak diterapkan di bidang kesehatan. (Zulpe & Pawar, 2012) mengkombinasikan GLCM dan Two-Layer Feed Forward Network untuk mengklasifikasikan kanker otak berdasarkan gambar *Magnetic Resonance Imaging* (MRI). Metode yang diusulkan oleh (Zulpe & Pawar, 2012) memiliki akurasi 97,5%. GLCM digunakan oleh (Shirvaikar, Huang, & Dong, 2016) untuk mengukur kualitas tulang untuk memprediksi kemungkinan patah tulang berdasarkan gambar *Dual-energy X-ray Absorptiometry* (DXA). Hasil pengujian yang dilakukan oleh (Shirvaikar, Huang, & Dong, 2016) menunjukkan jika matriks GLCM berhubungan erat dengan properti *microstructure* tulang. GLCM juga pernah diterapkan di bidang mammografi oleh (Pratiwi, Alexander, Harefa, & Nanda, 2015). Metode yang diusulkan oleh (Pratiwi, Alexander, Harefa, & Nanda, 2015) memiliki *Sensitivity* 97,22% dan *Specificity* 91,49%. Berdasarkan penelitian sebelumnya, kehandalan GLCM sudah teruji dalam beberapa kasus uji. Oleh karena itu, dalam penelitian ini GLCM dipilih sebagai metode *texture analysis*.

Penelitian yang dilakukan (Zulpe & Pawar, 2012), (Shirvaikar, Huang, & Dong, 2016), (Pratiwi, Alexander, Harefa, & Nanda, 2015) tidak menjelaskan secara spesifik kombinasi d dan θ yang memberikan hasil terbaik. Kesalahan pemilihan kombinasi d dan θ pada GLCM akan menyebabkan kesalahan klasifikasi. Pada bidang kesehatan, khususnya kanker payudara, kesalahan klasifikasi mammogram bisa mengakibatkan kesalahan diagnosis. Dalam kasus terburuk, kesalahan diagnosis bisa mengakibatkan kematian pasien. Sehingga pada penelitian ini, kami

Repository Universitas Brawijaya
mencari nilai d dan θ pada GLCM yang memberikan hasil terbaik pada studi kasus klasifikasi tumor payudara pada gambar ROI mammogram.

Metode yang diusulkan dalam penelitian ini terbagi kedalam tiga tahap utama, yaitu *preprocessing*, *feature extraction* dan *classification*. *Preprocessing* dilakukan untuk meningkatkan kualitas gambar mammogram. *Preprocessing* terbagi kedalam tiga tahap, yaitu *ROI Extraction*, *Contrast Limited Adaptive Histogram Equalization* (CLAHE) dan *Median Filter*. Setelah kualitas mammogram ditingkatkan, tahap selanjutnya adalah *feature extraction* menggunakan metode GLCM. *Feature extraction* menghasilkan kumpulan fitur, yaitu *mean*, *standard deviation*, *variance*, *contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy*, *maximum probability* dan *sum average*. Kumpulan fitur ini disebut *feature descriptor*. Setelah *feature descriptor* didapatkan, *feature descriptor* akan dijadikan *input neuron* pada algoritme *Backpropagation Neural Network* (BPNN). Luaran dari algoritme BPNN adalah prediksi kelas dari ROI yang dimasukkan. Luaran dari algoritme BPNN dalam penelitian ini adalah kelas normal atau abnormal, dimana normal adalah mammogram wanita sehat dan abnormal adalah mammogram wanita yang menderita kanker payudara.

Selain dipengaruhi kualitas data yang dipelajari, performa BPNN juga berbanding lurus dengan kemampuan belajar yang dimiliki BPNN. Kemampuan belajar BPNN ditentukan oleh *Learning Rate* dan jumlah *Epoch* yang ditentukan. Kesalahan penentuan *Learning Rate* atau jumlah *Epoch* akan menghambat kemampuan belajar yang dimiliki BPNN. Oleh karena itu, perlu ditentukan nilai *Learning Rate* dan jumlah *Epoch* yang menghasilkan performa BPNN terbaik. Performa BPNN akan diukur berdasarkan nilai *Mean Square Error* (MSE). Secara umum, semakin kecil nilai MSE, semakin bagus performa BPNN.

Tujuan utama dari penelitian ini adalah mencari kombinasi d dan θ pada GLCM yang bisa menghasilkan performa BPNN terbaik. Performa BPNN diukur menggunakan *Accurrature*, *Sensitivity* dan *Specificity*. Terakhir, penelitian ini akan membandingkan kemampuan metode yang diusulkan dengan metode pembanding.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Berapa *Learning Rate* dan jumlah *Epoch* di BPNN yang memberikan nilai MSE terkecil?
2. Bagaimana kombinasi nilai d dan θ GLCM pada data latih dan data uji yang memberikan *Accurrature*, *Sensitivity* dan *Specificity* terbaik?
3. Seberapa besar *Accurrature*, *Sensitivity* dan *Specificity* metode yang diusulkan jika dibandingkan algoritme pembanding pada kasus klasifikasi tumor payudara menggunakan gambar mammogram?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Mencari *Learning Rate* dan jumlah *Epoch* di BPNN yang menghasilkan MSE terkecil.
2. Mencari kombinasi d dan θ pada data latih dan data uji yang menghasilkan *Accuratuon*, *Sensitivity* dan *Specificity* terbaik.
3. Mencari *Accuratuon*, *Sensitivity* dan *Specificity* metode yang diusulkan dibandingkan algoritme pembedahan.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat:

1. Dibidang pengolahan gambar, dapat menemukan algoritme klasifikasi gambar baru. Terutama untuk klasifikasi penyakit kanker payudara menggunakan gambar mammogram.
2. Dibidang kesehatan, dapat menjadi bahan pertimbangan atau solusi alternatif bagi ahli radiologi untuk menentukan apakah ROI yang dicurigai tergolong normal atau abnormal.

1.5 Batasan Masalah

Agar penelitian ini tidak menyimpang dari tujuan yang ingin dicapai, dibuat batasan masalah sebagai berikut:

1. Gambar mammogram yang digunakan pada tahap klasifikasi normal-abnormal didapatkan dari *Mammography Imaging Analysis Society* (MIAS) dataset. Dataset bisa diunduh di <http://peipa.essex.ac.uk/info/mias.html>.
2. Tipe gambar yang digunakan adalah *grayscale*.
3. Luaran yang didapat berupa prediksi kelas normal-abnormal dari ROI yang diolah.
4. Program dibuat dengan menggunakan Python versi 3.7.
5. Variasi d yang digunakan dalam GLCM adalah $d = [1,2,3,4]$.
6. Variasi θ yang digunakan dalam GLCM adalah $\theta = [0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}]$.
7. *Learning rate* yang digunakan pada saat pengujian metode yang diusulkan didapatkan dari pengujian *learning rate* BPNN. *Learning rate* yang diuji mulai dari 0,1 sampai 1, dengan pertambahan jumlah *learning rate* sebanyak 0,1 pada setiap pengujian.
8. Jumlah *epoch* yang digunakan pada saat pengujian metode yang diusulkan didapatkan dari pengujian jumlah *epoch* BPNN. Jumlah *epoch* yang diuji mulai dari 5.000 sampai 80.000, dengan pertambahan jumlah *epoch* sebanyak 5.000 pada setiap pengujian.

1.6 Sistematika Pembahasan

Susunan laporan tesis dari penelitian ini adalah sebagai berikut:

1. Bab I Pendahuluan

Bab I membahas latar belakang permasalahan yang diangkat, pertanyaan yang akan dijawab, tujuan, manfaat dan batasan masalah dalam penelitian ini.

2. Bab II Dasar Teori

Pada Bab II dibahas penelitian sebelumnya dalam studi kasus tumor payudara dan dasar teori tentang tumor payudara, dataset yang digunakan, *ROI Extraction, Content Linear Adaptive Histogram Equalization, Median Filter, Gray-level Co-occurrence Matrix, Backpropagation Neural Network*. Selain itu, pada Bab II juga dibahas dasar teori pengujian yang dilakukan dalam penelitian ini.

3. Bab III Metodologi

Bab III memuat langkah-langkah penelitian yang dirancang untuk menjawab pertanyaan yang dikemukakan di subbab Rumusan Masalah. Rencana penelitian tersebut meliputi studi literatur, perancangan alur kerja program, perancangan *interface*, pembagian dataset kedalam kumpulan data *training* dan data *testing*, skenario pengujian dan spesifikasi *software* serta *hardware* yang digunakan dalam penelitian ini.

4. Bab IV Perancangan dan Implementasi

Bab IV membahas langkah-langkah perancangan metode yang diusulkan. Perancangan program yang dimaksud meliputi *flowchart* dan manualisasi untuk setiap bagian metode yang diusulkan. Hal ini dilakukan untuk mempermudah pemahaman dan implementasi algoritme yang digunakan. Terakhir, pada Bab IV juga disertakan penggalan *source code* implementasi metode yang diusulkan.

5. Bab V Pengujian dan Analisis

Memuat hasil pengujian dari sistem yang dibuat. Pengujian yang dilakukan sesuai dengan yang sudah dirancang di Bab III. Setelah hasil pengujian didapat, hasil pengujian akan dianalisis untuk menjawab pertanyaan dalam penelitian ini.

6. Bab VI Penutup

Memuat kesimpulan dari penelitian yang dilakukan dan saran untuk penelitian selanjutnya.

9. Metode pembanding yang digunakan ada 3, yaitu LBP-SVM (Kral & Lenc, 2016), GLCM-RBFNN (Pratiwi, Alexander, Harefa, & Nanda, 2015) dan LAWS-ANN (Setiawan, Elysia, Wesley, & Purnama, 2015).

Repository Universitas Brawijaya

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Topik dalam penelitian ini adalah klasifikasi gambar mammogram kedalam kelas normal-abnormal. Pada penelitian ini, gambar mammogram akan dirubah menjadi fitur. Proses ini disebut *feature extraction*. Setelah itu, fitur yang didapat akan dijadikan masukan untuk metode klasifikasi yang sudah ditentukan. Pada umumnya, sebelum bisa mengklasifikasikan data kedalam kelas yang diinginkan, metode klasifikasi harus dilatih terlebih dahulu. Proses ini disebut *training*. Pada tahap *training*, metode klasifikasi akan mempelajari pola data dan berusaha mengarahkan data tersebut kedalam kelas yang seharusnya. Setelah *training* selesai, kemampuan metode klasifikasi akan diuji pada tahap *testing*. Penelitian klasifikasi gambar mammogram menggunakan pendekatan *feature extraction* sudah banyak dilakukan antara lain dilakukan oleh (Kral & Lenc, 2016), (Pratiwi, Alexander, Harefa, & Nanda, 2015), (Setiawan, Elysia, Wesley, & Purnama, 2015). Tabel 2.1 menampilkan detail penelitian klasifikasi gambar mammogram menggunakan pendekatan *feature extraction*.

Tabel 2.1 Kajian pustaka klasifikasi gambar menggunakan pendekatan *feature extraction*

Referensi	(Kral & Lenc, 2016)
Judul	<i>LBP features for breast cancer detection</i>
Studi Kasus	Klasifikasi kanker payudara (normal-abnormal) pada MIAS dan DDSM dataset
Deskripsi	Kombinasi <i>Local Binary Pattern</i> (LBP) dan <i>Support Vector Machine</i> (SVM) digunakan oleh (Kral & Lenc, 2016). Metode Otsu digunakan dalam langkah <i>pre-processing</i> untuk menghapus latar belakang. Setelah itu, LBP digunakan untuk menampilkan vektor. Kernel berukuran 3x3 yang berpusat pada piksel yang diberikan diterapkan di LBP. Terakhir, SVM digunakan untuk mengklasifikasikan mammogram normal-abnormal pada dataset MIAS dan DDSM.
Hasil	Accurature (normal-abnormal) 84%
Referensi	(Pratiwi, Alexander, Harefa, & Nanda, 2015)
Judul	<i>Mammograms Classification Using Gray-level Co-occurrence Matrix and Radial Basis Function Neural Network</i>
Studi Kasus	Klasifikasi kanker payudara (normal-abnormal) pada MIAS dataset

Tabel 2.1 Kajian pustaka klasifikasi gambar menggunakan pendekatan feature extraction

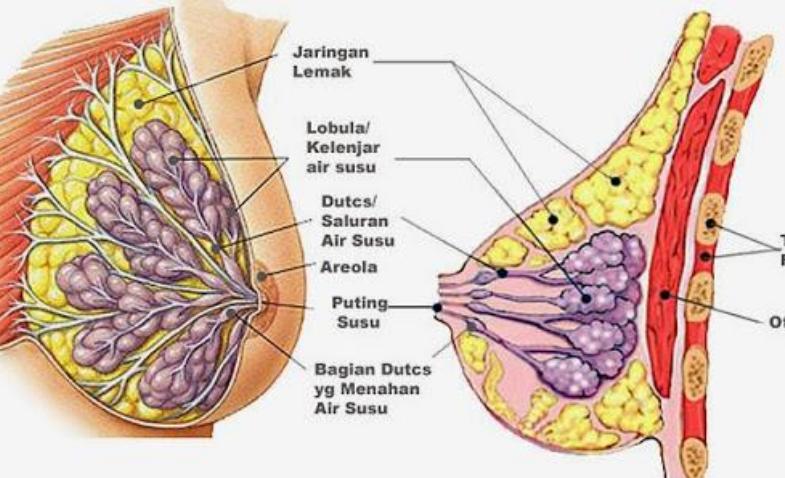
Desripsi	Kombinasi dari <i>Gray-level Co-occurrence Matrix</i> (GLCM) dan <i>Radial Basis Function Neural Network</i> (RBFNN) untuk mengklasifikasikan kasus normal-abnormal digunakan oleh (Pratiwi, Alexander, Harefa, & Nanda, 2015). Dataset MIAS digunakan untuk menguji sistem yang diusulkan. Pada langkah preprocessing, gambar mammogram dipotong menjadi ROI. GLCM digunakan untuk mengekstraksi empat fitur dari ROI, yaitu <i>ASM</i> , <i>Correlation</i> , <i>Sum Entropy</i> dan <i>Sum Variance</i> . Terakhir, perbandingan antara BPNN dan RBFNN digunakan dalam tahap klasifikasi.
Hasil	Normal-Abnormal
<i>Accurature</i>	93,98
<i>Sensitivity</i>	97,22%
<i>Specificity</i>	91,49%
Referensi	(Setiawan, Elysia, Wesley, & Purnama, 2015)
Judul	<i>Mammogram Classification using Law's Texture Energy Measure and Neural Networks</i>
Studi Kasus	Klasifikasi kanker payudara (normal-abnormal) pada MIAS dataset
Deskripsi	Laws' texture energy measures (LAWS) dan SVM digunakan oleh (Setiawan, Elysia, Wesley, & Purnama, 2015) untuk mengklasifikasikan kasus normal-abnormal dari dataset MIAS. Langkah pertama adalah memangkas ROI dari gambar mammogram. Kemudian, LAWS digunakan untuk mengekstrak fitur dari gambar mammogram. Sebagai perbandingan, GLCM digunakan. Fitur yang diekstraksi dari GLCM adalah <i>ASM</i> , <i>Correlation</i> , <i>Sum Variance</i> dan <i>Difference Entropy</i> . Terakhir, Artificial Neural Network (ANN) digunakan untuk menemukan metode <i>feature extraction</i> terbaik.
Hasil	Normal-Abnormal
<i>Accurature</i>	93,90%
<i>Sensitivity</i>	91%
<i>Specificity</i>	100%

2.2 Tumor Payudara

2.2.1 Pengertian

Sel-sel baru akan diproduksi untuk membantu pertumbuhan, menggantikan jaringan yang rusak dan menyembuhkan luka. Biasanya, sel berkembang biak dan mati secara teratur. Ketika proses ini berjalan tidak teratur, darah atau getah bening menjadi tidak normal. Hal ini menyebabkan benjolan yang disebut tumor. Jadi, bisa disebut jika tumor adalah penyakit pada sel (Groove & Marchant, 2016).

Tumor terbagi menjadi dua jenis, yaitu jinak dan ganas. Tumor jinak terjadi ketika sel terbatas pada satu area dan tidak menyebar ke bagian lain dari tubuh. Sedangkan tumor ganas terjadi ketika sel-sel menyebar melalui aliran darah atau sistem limfatis. Kondisi ini disebut kanker.



Gambar 2.1 Kelenjar payudara (sumber: (Dang, 2018))

Jaringan payudara terdiri dari kelenjar susu. Kelenjar susu terdiri dari *lobulus* (dimana susu diproduksi) dan saluran air susu (tabung yang membawa susu ke puting). Gambar jaringan payudara bisa di lihat di **Gambar 2.1**. Tumor payudara adalah pertumbuhan sel-sel abnormal yang melapisi saluran air susu. Meski jarang, tumor payudara juga bisa menyerang pria.

Repository Universitas Brawijaya
Repository Universitas Brawijaya

2.2.2 Faktor yang Mempengaruhi Tumor Payudara

Menurut (Colditz & Bohlke, 2014) dan (ACS, 2018), gaya hidup sehat seperti menjaga berat badan ideal, aktivitas fisik secara teratur, mengurangi atau bahkan menghindari alkohol dapat mencegah kanker payudara. Untuk wanita pascamenopause, setiap peningkatan 5 kg/m² BMI (Body Mass Index), meningkatkan risiko kanker payudara sekitar 12% (Renehan, Tyson, Egger, Heller, & Zwahlen, 2008). Konsumsi alkohol 10-g-per-hari meningkatkan risiko kanker payudara sekitar 7% -10% (Chen, Rosner, Hankinson, Colditz, & Willett, 2011) (Smith-Warner, Spiegelman, & Yaun, 1998) (Hamajima, Hirose, & Tajima, 2002). Untuk tingkat kematian, deteksi dini dan pengobatan mengurangi angka kematian kanker payudara sebesar 39% dari tahun 1989 hingga 2015 di Amerika Serikat (Siegel, Miller, & Jemal, 2018). Salah satu cara deteksi dini kanker payudara adalah dengan menggunakan mammogram (Tabár & Dean, 2003).

2.2.3 Deteksi Dini Tumor Payudara

Deteksi dini tumor payudara bisa dilakukan melalui beberapa cara, antara lain SADARI dan mammogram.

1. SADARI

SADARI adalah pengembangan kepedulian seorang perempuan terhadap kondisi payudaranya sendiri. Tindakan ini dilengkapi dengan langkah-langkah khusus untuk medeteksi tumor payudara dengan cara menganalisis perubahan pada payudara. SADARI dilakukan antara 7 hari sampai 10 hari setelah hari pertama menstruasi/setelah menstruasi. Perubahan yang perlu diperhatikan ketika melakukan SADARI adalah perubahan bentuk dan ukuran payudara, teraba benjolan, nyeri, penebalan kulit, terdapat cekungan kulit seperti lesung pipit, pengerutan kulit payudara, keluar cairan dari puting susu, penarikan puting susu kedalam dan luka pada payudara yang tidak sembuh sembah.

Perlu dicatat jika SADARI tidak menggantikan peran dokter atau tenaga medis terlatih. Pemeriksaan payudara secara rutin oleh para ahli sebaiknya dilakukan untuk perempuan usia 20-40 tahun minimal setiap 3 tahun dan perempuan dengan usia diatas 40 tahun minimal setahun sekali.

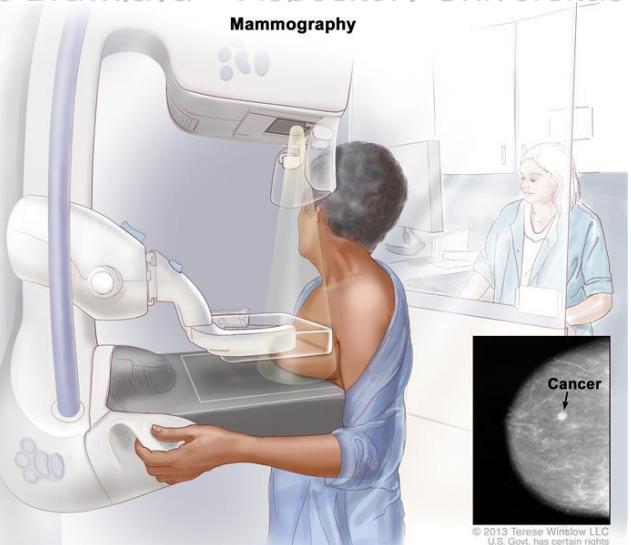
2. Mammogram

Selain melalui SADARI, deteksi dini juga bisa dilakukan dengan mammografi. Mammografi dianggap alat deteksi dini kanker payudara yang paling efektif karena mammografi mampu medeteksi 80% - 90% kasus kanker payudara. Gambar payudara yang dihasilkan dari proses mammografi disebut mammogram. Mammogram adalah Sinar-X dosis rendah dari jaringan payudara (Groove & Marchant, 2016). Dengan menggunakan mammogram, benjolan atau benjolan yang terlalu kecil dan tidak bisa dirasakan bisa dilihat. Mammogram akan dilakukan pada kedua payudara.

Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Selama pembuatan mammogram, payudara akan ditekan di antara dua pelat

Sinar-X, yang menyebar jaringan payudara sehingga gambar yang jelas dapat diambil. Proses ini memakan waktu sekitar 20 detik. Ilustrasi pembuatan gambar mammogram bisa dilihat di Gambar 2.2.



Gambar 2.2 Ilustrasi pembuatan gambar mammogram (Sumber: (Institute, 2019))

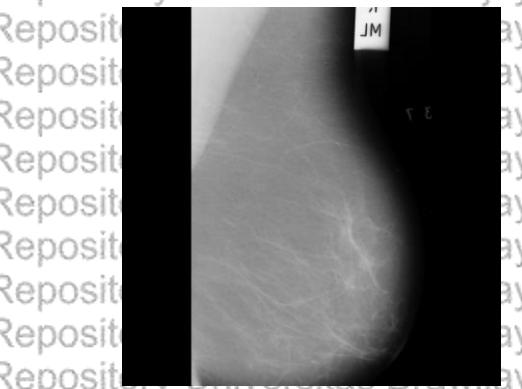
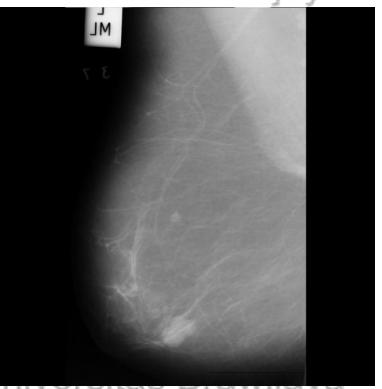
Mammogram terdiri dari area hitam, abu-abu dan putih. Area hitam menggambarkan background gambar. Sedangkan area abu-abu menggambarkan jaringan lemak normal. Sedangkan area putih pada mammogram menggambarkan ducts (saluran air susu) dan lobula (kelenjar air susu). Contoh gambar mammogram ada di Gambar 2.3.



Gambar 2.3 Contoh gambar mammogram

2.3 MIAS

MIAS adalah *database digital mammogram* yang dibuat oleh organisasi UK *research group*. MIAS mengandung 161 *Mediolateral Oblique Pairs* (MLO) dengan ukuran 1024 x 1024 pixel. Gambar diambil dari *screen-film mammographic* payudara nasional Inggris. Setiap gambar mammogram memiliki latar belakang jaringan (*fatty*, *fatty-glandular* dan *dense-glandular*) dan kelas kelainan (*calcification*, *well defined or circumscribed masses*, *spiculated masses*, *other or ill-defined masses*, *architectural distortion*, *asymmetry*, dan *normal*). Kelas lain selain kelas normal termasuk kedalam kelas abnormal. Khusus untuk kelas abnormal, ada informasi tambahan seperti tingkat keparahan tumor (jinak atau ganas), pusat koordinat tumor dan radius tumor. Koordinat 0,0 gambar berada di sudut kiri bawah gambar. Gambar mammogram berada di bagian tengah gambar. Gambar 2.4 (a) menunjukkan kelas abnormal dan Gambar 2.4 (b) menunjukkan kelas normal. Karena alasan kelengkapan data, data nomor 216, 233, 245 tidak digunakan. Tabel 2.2 menampilkan detail data MIAS.



Gambar 2.4 Gambar MIAS dataset (a) Kelas Abnormal, (b) Kelas Normal

Tabel 2.2 Detail data MIAS

Data	319
NORMAL	213
ABNORMAL	107

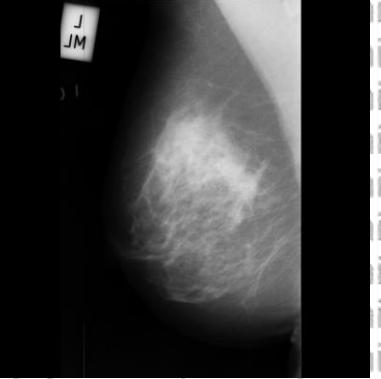
2.3.1 Kelas Mammogram

MIAS membagi mammogram kedalam 7 kelas, yaitu:

1. *Calcification*

Calcification adalah deposit mineral (Kalsium) dalam jaringan payudara. *Calcification* terlihat seperti bintik terang pada gambar mammogram. *Calcification* bisa disebabkan oleh sekresi sel atau dari tumpukan sel mati. *Calcification* terbagi menjadi dua jenis, yaitu *macrocalcification* dan *microcalcification*. *Macrocalcification* adalah kondisi dimana muncul endapan

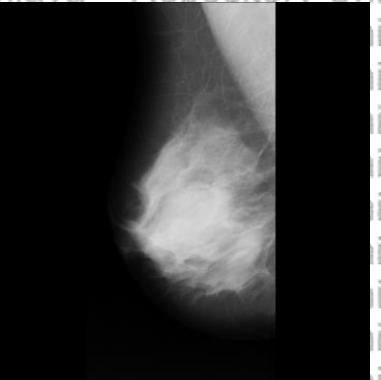
Kalsium besar dalam jaringan payudara. Sementara *microcalcification* adalah bintik kalsium kecil dan terisolasi, biasanya muncul dalam kelompok atau ditemukan tertanam dalam massa payudara. Secara umum, *microcalcification* individu berukuran 0,1-1,0 mm dengan diameter rata-rata sekitar 0,5 mm (Verma, McLeod, & Klevansky, 2010). Gambar 2.5 menunjukkan contoh kasus *calcification*.



Gambar 2.5 Mammogram *calcification*

2. *Well-defined/circumscribed masses*

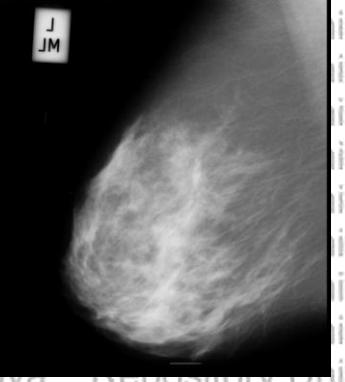
Circumscribed masses adalah tumor dimana setidaknya 75% permukaannya terlihat. 25% sisanya (paling banyak) ditutupi kelenjar disekitarnya (Berment, Becette, Mohallem, Ferreira, & Chérel, 2014). Gambar 2.6 menunjukkan contoh kasus *circumscribed masses*.



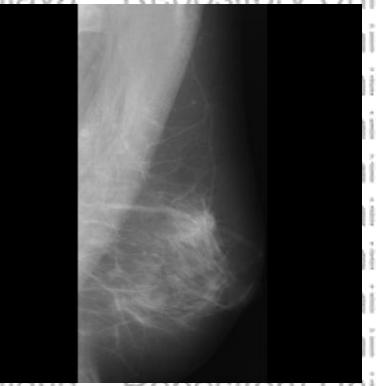
Gambar 2.6 Mammogram *well-defined/circumscribed masses*

3. *Spiculated masses*

Spiculated masses dalam mammografi berhubungan dengan *opacity* yang dibentuk oleh pusatnya yang padat. Dari pusat tersebut muncul beberapa pemanjangan radial linear yang disebut *spicules*. Pusatnya yang padat menjadi pembeda antara *spiculated masses* dan *architectural distortion* (Berment, Becette, Mohallem, Ferreira, & Chérel, 2014). Gambar 2.7 menunjukkan contoh kasus *spiculated masses*.

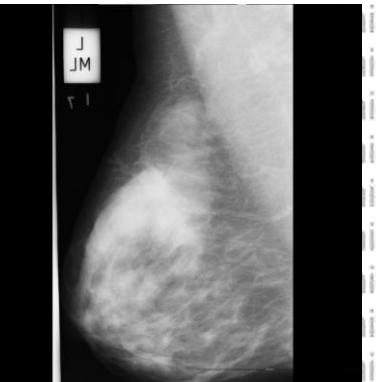


Gambar 2.7 Mammogram *spiculated masses*



Gambar 2.8 Mammogram *ill-defined masses*

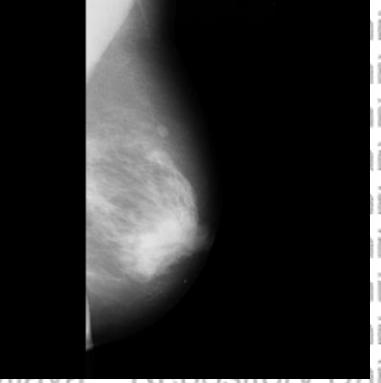
Architectural distortion dalam mammogram biasanya merujuk ke gangguan terhadap “pola” acak yang biasanya terlihat di mammogram. *Architectural distortion* biasanya berbentuk menyerupai bintang dan berukuran jauh lebih besar dibandingkan tumor yang muncul karena kelainan lain (Halls, 2019). Gambar 2.9 menunjukkan contoh kasus *architectural distortion*.



Gambar 2.9 Mammogram *architectural distortion*

6. Asymmetry

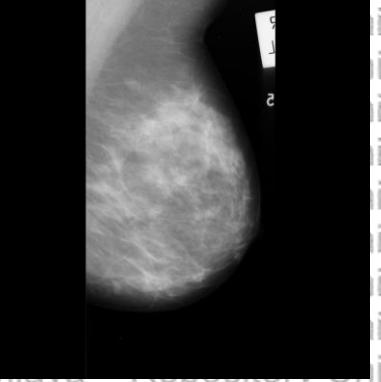
Asymmetry terbagi menjadi *asymmetrical breast tissue* dan *asymmetric density*. *Asymmetrical breast tissue* adalah perbedaan jaringan payudara di area yang sama pada payudara yang berlawanan. Sedangkan *asymmetric density* biasanya berkaitan dengan *opacity*. *Asymmetric density* hanya terlihat di satu sudut X-ray saja (Halls, 2019). Gambar 2.10 menunjukkan contoh kasus *asymmetry*.



Gambar 2.10 Mammogram asymmetry

7. Normal

Mammogram normal (sehat) biasanya sebagian besar berwarna abu-abu. Pada beberapa bagian, ada yang berwarna putih menunjukkan jaringan yang lebih padat dan sehat. Perlu dicatat jika lebih banyak warna putih di mammogram belum tentu mengindikasikan tumor. Akan tetapi, wanita dengan jaringan payudara padat memiliki kemungkinan sedikit lebih besar untuk terserang kanker payudara. Jaringan payudara yang padat juga bisa menutupi abnormalitas yang ada di payudara (Sissons, 2018). Gambar 2.11 menunjukkan contoh kasus normal.

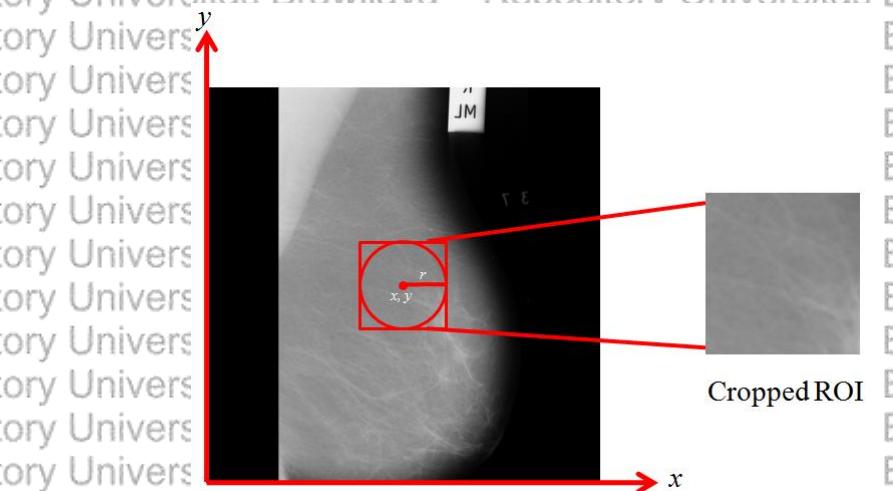


Gambar 2.11 Mammogram normal



2.4 Region of Interest (ROI)

ROI adalah persegi yang dibuat oleh pengguna dengan tujuan untuk mempersingkat waktu komputasi (Brinkmann, 1999). Konsep ROI biasa digunakan dalam berbagai area aplikasi. Sebagai contoh, dalam gambar medis, batas-batas tumor dapat didefinisikan pada gambar. Dalam *Geographical Information Systems* (GIS), ROI bisa diambil sebagai seleksi poligonal dari peta 2 dimensi (2D). Dalam visi komputer dan *Optical Character Recognition* (OCR), ROI mendefinisikan batas objek yang akan dikenali. Dalam banyak aplikasi, label simbolis (tekstual) ditambahkan ke ROI, untuk menggambarkan kontennya secara ringkas. Gambar 2.12 menampilkan ilustrasi pengambilan ROI dari gambar mammogram.

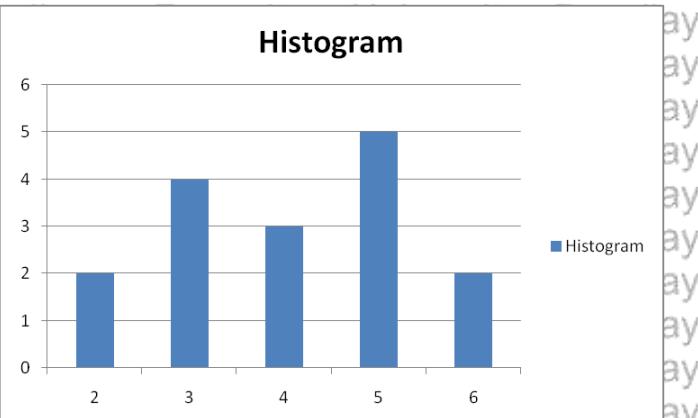


Gambar 2.12 Contoh pengambilan ROI dari gambar mammogram

2.5 Histogram Equalization

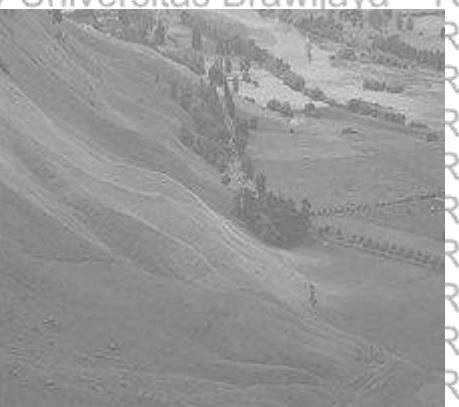
4	5	5	5
3	6	3	4
2	4	2	6
3	5	3	5

(a)



(b)

Gambar 2.13. Mengubah gambar jadi histogram (a) Gambar grayscale, (b) Histogram



(a)



(b)

Gambar 2.14 Histogram Equalization (a) Sebelum proses **histogram equalization**, (b) Setelah proses **histogram equalization** (Sumber : (doxygen.org, 2015))

Histogram equalization digunakan untuk mempertajam kontras. *Histogram equalization* adalah redistribusi nonlinier dari nilai piksel gambar (Zhihong & Xiaohong, 2011). **Gambar 2.13** menunjukkan contoh bagaimana mengubah gambar menjadi histogram. **Gambar 2.13** (a) adalah contoh gambar *grayscale*. **Gambar 2.13** (b) adalah histogram yang dibuat dari **Gambar 2.13** (a). Dari **Gambar 2.13** (b), sumbu x adalah tingkat keabuan dan sumbu y adalah jumlah piksel. Pada **Gambar 2.13** (a), ada 2 piksel dengan nilai 2. *Histogram Equalization* dilakukan menggunakan Persamaan 2.1.

$$f_{i,j}(n) = \frac{(N-1)}{M} \sum_{k=0}^n h_{i,j}(k) \quad (2.1)$$

$$n = 1, 2, 3, \dots, N-1$$

M adalah jumlah pixel, N adalah nilai *graylevel* maksimum, $h_{i,j}(n)$ untuk $n = 1, 2, 3, \dots, N-1$ adalah nilai *graylevel* pada bagian (i,j) . **Gambar 2.14** menampilkan perbedaan gambar sebelum dan sesudah proses *histogram equalization* dilakukan.

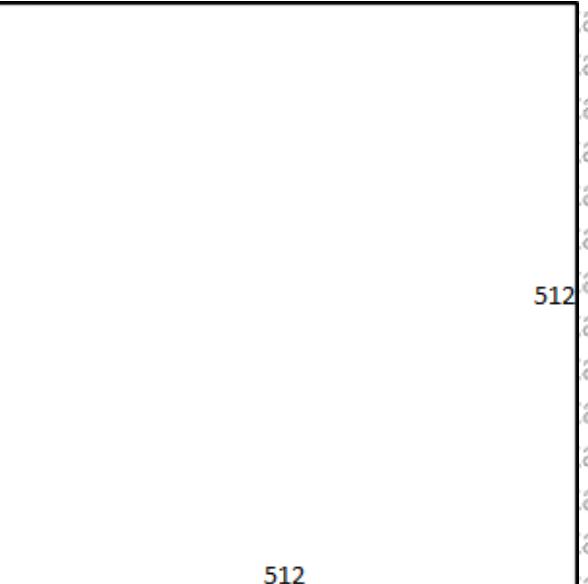
2.5.1 Contrast Limited Adaptive Histogram Equalization (CLAHE)

Histogram equalization menyetarakan nilai *grayscale* gambar secara global. Hal ini disebabkan karena *histogram equalization* menganggap semua bagian gambar memiliki kualitas yang sama. Penyetaraan jenis ini memiliki kelemahan ketika kontras *background* dan *foreground* gambar tidak berbeda jauh. **Gambar 2.15** mengilustrasikan efek *histogram equalization* pada gambar dengan tingkat kontras *background* dan *foreground* yang mirip. Pada **Gambar 2.15**, kontras *background* dan *foreground* ditingkatkan secara keseluruhan. Akibatnya, *background* menjadi lebih terang, sementara *foreground* mengalami *over-brightness*. Salah satu solusi untuk kelemahan *histogram equalization* ini adalah *Adaptive Histogram Equalization* (AHE).



Gambar 2.15 Histogram equalization (a) Sebelum proses histogram

AHE membagi gambar menjadi blok-blok berukuran hampir sama. Blok ini disebut *tile*. Umumnya, setiap *tile* berukuran 8x8. Diberikan gambar F yang ditunjukkan pada Gambar 2.16. Gambar 2.16 memiliki dimensi 512×512 . Secara statistik, untuk gambar dengan dimensi 512×512 , hasil terbaik diperoleh ketika gambar tersebut dibagi kedalam 64 region berukuran sama (Reza, 2004). Proses pembagian gambar F menjadi 64 region ditunjukkan di Gambar 2.17. Pembagian gambar menjadi region menghasilkan tiga jenis region. Jenis region yang pertama disebut *corner region* (CR). Jumlah CR hanya ada 4. Jenis region yang kedua adalah *border region* (BR). BR adalah semua region yang berada di tepi gambar selain CR. Pada Gambar 2.17, ada 24 region yang termasuk BR. Jenis ketiga adalah *inner region* (IR). IR adalah semua region gambar yang tidak termasuk BR dan CR. Pada Gambar 2.17, ada 36 region yang termasuk IR.



Gambar 2.16 Gambar F dengan dimensi 512×512

CR	BR	BR	BR	BR	BR	BR	CR
BR	IR	IR	IR	IR	IR	IR	BR
BR	IR	IR	IR	IR	IR	IR	BR
BR	IR	IR	IR	IR	IR	IR	BR
BR	IR	IR	IR	IR	IR	IR	BR
BR	IR	IR	IR	IR	IR	IR	BR
BR	IR	IR	IR	IR	IR	IR	BR
BR	IR	IR	IR	IR	IR	IR	BR
CR	BR	BR	BR	BR	BR	BR	CR

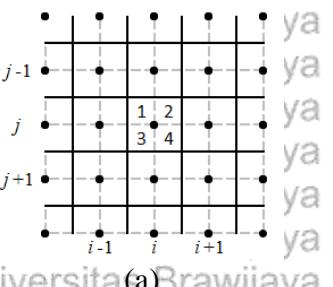
Gambar 2.17 Struktur region gambar F yang dibagi kedalam 64 region berukuran sama

Menggunakan sistem *tile*, AHE mengubah sifat global dari *histogram equalization* menjadi *local*. Hal ini memiliki kekurangan ketika didalam region tersebut terdapat *noise*. *Noise* didalam region tersebut juga akan ikut dipertajam ke arah nilai maksimal. CLAHE dibuat untuk menyelesaikan permasalahan ini. CLAHE menerapkan sistem pembatasan kontras. Proses ini biasa disebut *clipping*. Batas maksimal yang ditentukan CLAHE disebut *clip limit*. Salah satu cara menentukan *clip limit* adalah dengan menggunakan *clip limit* β (Reza, 2004).

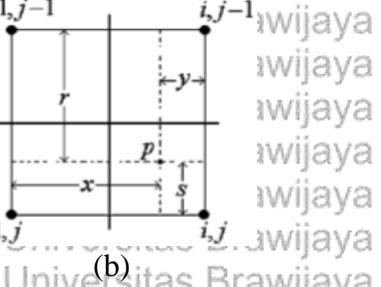
Persamaan 2.2 menampilkan persamaan *clip limit* β .

$$\beta = \frac{M}{N} \left(1 + \frac{\alpha}{100} (s_{max} - 1) \right) \quad (2.2)$$

M adalah jumlah pixel, N adalah nilai *graylevel* maksimum, α adalah *clip factor*. Ketika α bernilai 100, tidak ada batas untuk setiap nilai histogram. s_{max} adalah nilai *slope* maksimal yang diperbolehkan. Untuk gambar X-ray, nilai *slope* adalah 4 (Reza, 2004).

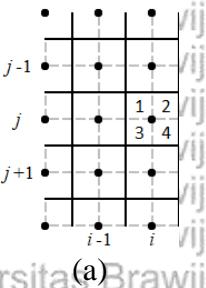


(a)



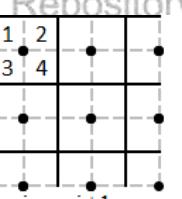
(b)

Gambar 2.18 (a) IR dengan semua region tetangganya. (b) Pixel p dari kuadran 1 dari (i,j) region dan relasinya dengan titik tengah empat tetangga terdekatnya. (Sumber: (Reza, 2004))



Gambar 2.19. (a) BR dengan semua region tetangganya. (b) Pixel p dari kuadran 2 dari (i,j) region dan relasinya dengan titik tengah dua tetangga terdekatnya. (Sumber: (Reza, 2004))

Sumber: (Reza, 2004))



Gambar 2.20. CR pada posisi kiri atas dan struktur ketetanggaannya (Sumber: (Reza, 2004)).

Setelah $clip_limit$ ditentukan, tahap selanjutnya adalah perhitungan nilai baru untuk setiap pixel (p_{new}). Perhitungan (p_{new}) pada setiap region ditentukan oleh letak pixel tersebut. Jika letak pixel seperti Gambar 2.18, p_{new} dihitung dengan menggunakan Persamaan 2.3. Jika letak pixel seperti Gambar 2.19, p_{new} dihitung dengan menggunakan Persamaan 2.4. Jika letak pixel seperti Gambar 2.20, p_{new} dihitung dengan menggunakan Persamaan 2.5. Gambar 2.21 menampilkan hasil penerapan algoritme CLAHE pada Gambar 2.15 (a).

$$p_{new} = \frac{s}{r+s} \left(\frac{y}{x+y} f_{i-1,j-1}(p_{old}) + \frac{x}{x+y} f_{i,j-1}(p_{old}) \right) + \frac{r}{r+s} \left(\frac{y}{x+y} f_{i-1,j}(p_{old}) + \frac{x}{x+y} f_{i,j}(p_{old}) \right) \quad (2.3)$$

Keterangan:

s, r, x, y diambil dari Gambar 2.18.

$f_{i-1,j-1}(p_{old})$ nilai pixel lama di kuadran $f_{i-1,j-1}$ diambil dari Gambar 2.18

$f_{i-1}(p_{old})$ nilai pixel lama di kuadran f_{i-1} diambil dari Gambar 2.18

$f_{i-1} i(p_{old})$ nilai pixel lama di kuadran $f_{i-1} i$ diambil dari Gambar 2.18

$f_{i,j}(n_{old})$ nilai pixel lama $f_{i,j}$ diambil dari Gambar 2.18

Universitas Brawijaya Repository | Repository Universitas Brawijaya

$$p_{new} = \frac{s}{r+s} f_{i,j-1}(p_{old}) + \frac{r}{r+s} f_{i,j}(p_{old}) \quad (2.4)$$

Keterangan:

s, r diambil dari Gambar 2.19

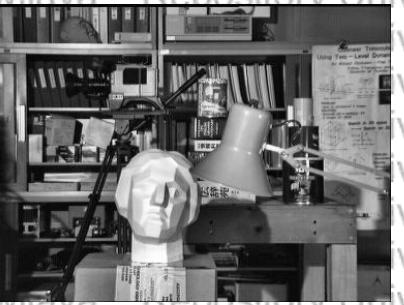
$f_{i,i-1}(p_{old})$ nilai pixel lama di kuadran $f_{i,i-1}$ diambil dari Gambar 2.19

$f_{i,i}(p_{old})$ nilai pixel lama $f_{i,i}$ diambil dari Gambar 2.19

$$p_{new} = f_{i,j}(p_{old})$$

Keterangan:

$f_{i,j}(p_{old})$ nilai pixel lama $f_{i,j}$ diambil dari Gambar 2.20



Gambar 2.21 Hasil CLAHE pada Gambar 2.15 (a) (Sumber: (doxygen.org, 2015))

2.6 Median Filter

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	2	3	3	3	3	4	4	4	5	5	5	5	5	6	6

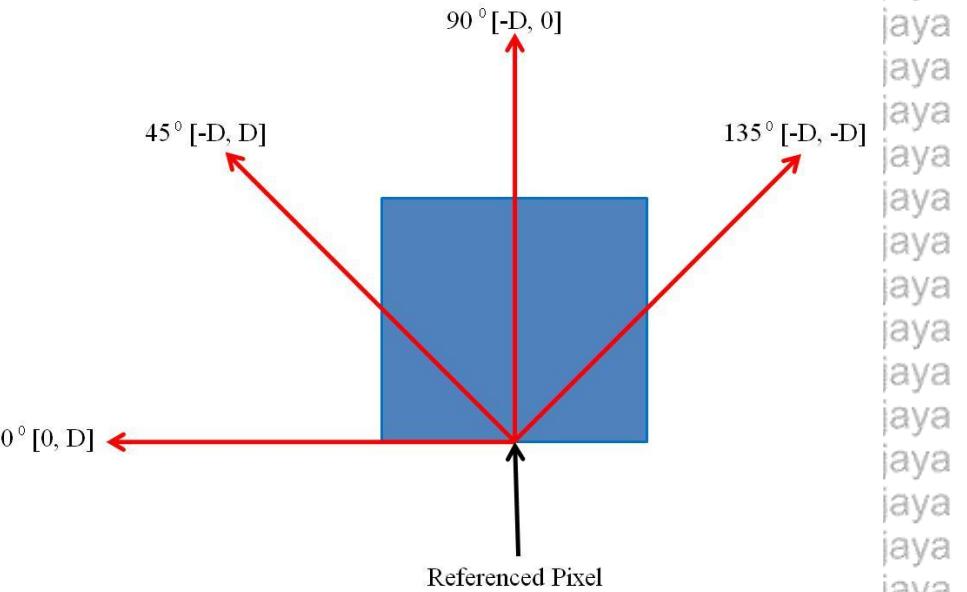
Gambar 2.22 Urutan nilai grayscale Gambar 2.13 (a)

Median Filter termasuk kedalam filter jenis non-linear. Non-linear filter adalah filter yang sifatnya adaptif. Non-linear filter memiliki kemampuan untuk mempertahankan tepi, tapi tetap mengurangi/menghilangkan noise. Median filter dimulai dengan mengambil kernel berukuran tertentu. Setelah itu, nilai didalam kernel akan diurutkan dan diambil nilai tengahnya. Gambar 2.22 adalah contoh urutan nilai grayscale Gambar 2.13 (a). Pada Gambar 2.22, nilai median didapat pada indeks ke-8 dan ke-9, dengan nilai mediannya adalah 4.

2.7 Gray-level Co-occurrence Matrix (GLCM)

GLCM digunakan untuk mengekstrak tekstur dalam gambar berdasarkan transisi tingkat abu-abu antara dua piksel. GLCM memberikan distribusi tingkat abu-abu berpasangan dari piksel tetangga dalam suatu gambar (Haralick, Shanmugam, & Dinstein, 1973). Tahap komputasi GLCM dimulai dengan menentukan satu piksel primer dan satu piksel tetangga. Tahap komputasi GLCM diakhiri dengan pembentukan matriks ketetanggaan GLCM.

GLCM memiliki dua parameter, yaitu d dan θ , di mana d adalah jarak antara piksel utama dan piksel tetangga dan θ adalah arah ketetanggaan antara piksel utama dan piksel tetangga. Gambar 2.23 menggambarkan kombinasi d dan θ pada GLCM.



Gambar 2.23 Kombinasi D dan Θ dalam GLCM

	0	1	2	3
0	1	3	0	3
1	3	2	2	0
2	1	2	3	0
3	2	0	2	3

Gambar 2.24 Co-occurrence Matrix dengan 4 level keabuan

	0	1	2	3		0	1	2	3		0	1	2	3	
0	0	0	3	3	(a)	0	2	1	1	(b)	0	2	0	2	(c)
1	0	0	1	1		1	1	0	1		1	0	1	2	
2	3	1	2	3		2	1	1	2		2	1	4	1	
3	3	1	3	0		3	0	0	3		3	2	0	1	

**Gambar 2.25 Pembentukan matriks GLCM. $d = 1$ dan (a) horizontal ($\theta = 0^\circ$)
(b) diagonal kiri ($\theta = 45^\circ$) (c) vertikal ($\theta = 90^\circ$) (d) diagonal kanan
($\theta = 135^\circ$)**

Diberikan gambar G dengan intensitas tingkat abu-abu yang ditunjukkan pada Gambar 2.24. Dengan $d = 1$ dan ($\theta = 0^\circ$), 1 sel dengan nilai 0 yang memiliki tetangga dengan nilai 2 telah ditemukan dalam Gambar 2.24. Dengan menggunakan aturan simetrik, 2 sel dengan nilai 2 yang memiliki tetangga dengan nilai 0 telah ditemukan pada Gambar 2.24. Oleh karena itu, matriks pada indeks (0,2) pada Gambar 2.25 (a) memiliki nilai 3. Ulangi proses ini untuk semua sel pada Gambar 2.25 (a). Dengan menggunakan metode ini, Gambar 2.25 (b) dengan diberikan $d = 1$ dan ($\theta = 45^\circ$), Gambar 2.25 (c) dengan diberikan $d = 1$ dan ($\theta = 90^\circ$), dan Gambar 2.25 (d) dengan diberikan $d = 1$ dan ($\theta = 135^\circ$) juga dibuat.

Diberikan matriks $M_{(i,j)}^d$ yang merupakan matriks GLCM, sepuluh fitur dari setiap gambar akan diekstraksi. Fitur yang diekstrak adalah *mean*, *standard deviation*, *variance*, *contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy*, *maximum probability* dan *sum average*. Kumpulan fitur ini disebut *feature descriptor*.

Persamaan *feature descriptor* ditampilkan pada Persamaan 2.6 sampai 2.15.

Perhatikan bahwa i dan j adalah tingkat keabuan matriks GLCM. Dengan menerapkan kombinasi d dan θ , setiap gambar akan memiliki 160 fitur.

$$\text{Mean} = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{(i,j)}}{mn} \quad (2.6)$$

m adalah jumlah baris matriks GLCM, n adalah jumlah kolom matriks GLCM, i, j adalah grayscale level matriks GLCM, G adalah 2^8 .

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (M_{(i,j)} - \text{Mean})^2}{mn}} \quad (2.7)$$

$$\text{Variance} = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (M_{(i,j)} - \text{Mean})^2}{mn} \quad (2.8)$$

$$\text{Contrast} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{(i,j)} |i - j|^2 \quad (2.9)$$

$$\text{Dissimilarity} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{(i,j)} |i - j| \quad (2.10)$$

$$\text{Homogeneity} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{M_{(i,j)}}{1 + (i - j)^2} \quad (2.11)$$

$$\text{ASM} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (M_{(i,j)})^2 \quad (2.12)$$

$$\text{Energy} = \sqrt{\text{ASM}} \quad (2.13)$$

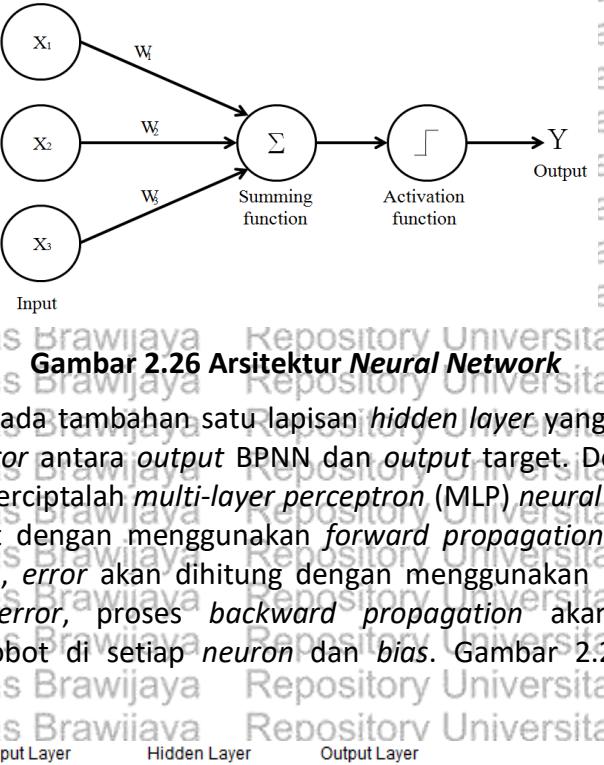
$$\text{Maximum Probability} = \max(M_{(i,j)}) \quad (2.14)$$

$$\text{Sum Average} = \frac{1}{2} \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{(i,j)} (i + j) \quad (2.15)$$

2.8 Back Propagation Neural Network (BPNN)

BPNN adalah salah satu contoh algoritma dalam bidang jaringan syaraf tiruan. Gambar 2.26 menunjukkan arsitektur *Neural Network* (NN). Pada dasarnya, NN terdiri dari beberapa elemen pemrosesan, yang disebut *neuron*. Beberapa *neuron* pada lapisan/kedalaman yang sama disebut *layer*. Setiap *neuron* di *layer* yang berbeda dihubungkan oleh *synapses*. NN dasar terdiri dari dua *layer*, yaitu *input layer* dan *output layer*. *Output* dari *output neuron* didapatkan dengan cara menerapkan fungsi *dot* antara sinyal input dan bobot *synapses*. Setelah itu, hasil

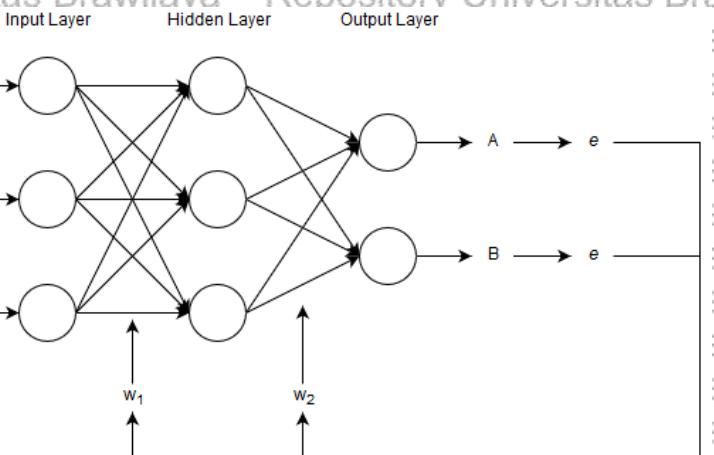
fungsi *dot* akan melewati fungsi aktivasi untuk membatasi *output* kedalam interval pra-spesifik yang ditentukan (Rouhi, Jafari, Kasaei, & Keshavarzian, 2015). Proses ini disebut *forward propagation*. Bobot *synapses* awal pada NN ditentukan secara acak. Lama waktu pelatihan yang dimiliki NN disebut *epoch*. x_i adalah data input dan w_i adalah data bobot *synapses*.



Gambar 2.26 Arsitektur Neural Network

Dalam BPNN, ada tambahan satu lapisan *hidden layer* yang berfungsi untuk meminimalisir *error* antara *output* BPNN dan *output target*. Dengan tambahan *hidden layer* ini, terciptalah *multi-layer perceptron (MLP) neural network*. *Output* dari MLP didapat dengan menggunakan *forward propagation*. Setelah *output* BPNN didapatkan, *error* akan dihitung dengan menggunakan Persamaan 2.18.

Ketika muncul *error*, proses *backward propagation* akan terjadi untuk menyesuaikan bobot di setiap *neuron* dan *bias*. Gambar 2.27 menunjukkan arsitektur BPNN.



Gambar 2.27 Arsitektur BPNN

Sebelum dijadikan masukan untuk BPNN, data masukan (P_1 , P_2 dan P_3 di Gambar 2.27) harus dinormalisasi terlebih dahulu. Salah satu cara normalisasi adalah menggunakan *Z-Score Normalization*. Persamaan *Z-Score Normalization* ditunjukkan di Persamaan 2.16.

$$Z = \frac{(X_i - \mu)}{\sigma} \quad (2.16)$$

Z adalah Z-Score normalization, X adalah data asli, μ adalah rata-rata, σ adalah standard deviation.

Setelah data selesai dinormalisasi, data siap dijadikan masukan untuk BPNN.

Tahapan Algoritme BPNN adalah sebagai berikut:

1. Menentukan jumlah *Hidden Neuron* menggunakan Persamaan 2.17 (Heaton, 2008).

$$HN = \left(\frac{2}{3} n_i \right) + n_o \quad (2.17)$$

HN adalah jumlah *Hidden Neuron*, n_i adalah jumlah *Input Neuron*, n_o adalah jumlah *Output Neuron*.

2. Tentukan *bias* dan *learning rate*

3. Menginisialisasi bobot semua *synapses* secara acak.

4. Untuk semua sampel data training X , lakukan:

3.1 Forward Propagation

Hitung luaran prediksi z untuk setiap sampel data training X menggunakan Persamaan 2.18.

$$z = bias + (X \times w_{ih}) \quad (2.18)$$

w_{ih} adalah bobot *synapses input layer* ke *hidden layer*. Setelah itu, terapkan fungsi aktivasi terhadap z . Asumsikan fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid yang ditampilkan di Persamaan 2.19.

$$z_2 = \frac{1}{1 + e^{-z}} \quad (2.19)$$

Setelah itu, hitung luaran prediksi z_3 menggunakan Persamaan 2.20.

$$z_3 = bias + (z_2 \times w_{ho}) \quad (2.20)$$

w_{ho} adalah bobot *synapses hidden layer* ke *output layer*. Terapkan fungsi aktivasi sigmoid terhadap z_3 . Setelah luaran didapat, hitung *error* menggunakan Persamaan 2.21.

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N (o - t)^2 \quad (2.21)$$

o adalah luaran prediksi, t adalah target yang diinginkan, M adalah jumlah data training X , N adalah jumlah *Output Neuron*.

3.2 Backward Propagation

Hitung *error* di setiap *output neuron* menggunakan Persamaan 2.22.

$$\sigma_k = o(1 - o)(t - o) \quad (2.22)$$

Hitung *error* di setiap *Hidden Neuron* menggunakan Persamaan 2.23.

$$\delta_h = O_h(1 - O_h) \sum_k w_{kh} \sigma_k \quad (2.23)$$

Keterangan:

O_h adalah luaran dari *Hidden Neuron* ke- h

w_{kh} adalah bobot *synapses* yang menghubungkan *Hidden Neuron* ke- h dengan *Output Neuron* ke- k

Update bobot *synapses* menggunakan Persamaan 2.24.

$$w_{ji} = w_{ji} + \Delta w_{ji} \quad (2.24)$$

Δw_{ji} didapat dari Persamaan 2.25.

$$\Delta w_{ji} = \eta \delta_j X_{ij} \quad (2.25)$$

Keterangan:

η adalah *Learning Rate*

5. Jika minimum *error* atau maksimum *epoch* belum tercapai, ulangi langkah 3.

2.9 Pengujian

Pembagian dataset menjadi data uji dan data latih dilakukan menggunakan metode *k-Fold Cross-Validation*. *Cross-Validation* adalah prosedur resampling yang digunakan untuk mengevaluasi model *machine learning* pada sampel data terbatas. *Cross-Validation* mempunyai parameter tunggal k yang menentukan banyaknya kombinasi sampel. Oleh karena itu, metode ini disebut *k-Fold Cross-Validation*. Tidak ada cara formal untuk menentukan jumlah k , tapi secara umum $k = 5$ atau $k = 10$ (Kuhn & Johnson, 2013). Menggunakan *k-Fold Cross-Validation*, setiap data akan menjadi anggota data uji dan data latih. Pada setiap kombinasi, tidak ada duplikasi data.

Metode yang diusulkan akan diuji terhadap masing-masing kombinasi data latih dan data uji. Pengujian yang dilakukan ada tiga, yaitu pengujian *Accururation*, *Sensitivity* dan *Specificity*.

2.9.1 Accururation

Accururation adalah ukuran kemampuan pengklasifikasi untuk menghasilkan diagnosis yang akurat. Formula untuk *Accururation* dapat dilihat di Persamaan 2.26.

$$Accururation = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.26)$$

TP adalah *True Positive*, TN adalah *True Negative*, FP adalah *False Positive*

FN adalah *False Negative*.

Setelah *Accururation* untuk masing-masing kombinasi data uji dan data latih didapat, *Accururation* akan dirata-rata untuk mendapatkan *Global Accururation*. Formula untuk menghitung *Global Accururation* ditampilkan di Persamaan 2.27.

Repository Universitas Brawijaya
Repository Universitas Brawijaya $\sum_{t=0}^n Acc_t$
Global Assumption

$$Global\ Accuracy = \frac{\sum_{i=0}^n Accurration_i}{n} \quad (2.27)$$

n adalah banyaknya kombinasi data uji dan data latih (k dalam *k-Fold Cross-Validation*)

2.9.2 Sensitivity

Sensitivity adalah ukuran kemampuan classifier untuk mengidentifikasi kemunculan kelas target secara akurat. Pada penelitian ini, *Sensitivity* mengukur kemampuan *metode yang diusulkan* untuk mengidentifikasi kelas Normal. Persamaan *Sensitivity* bisa dilihat di Persamaan 2.28.

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (2.28)$$

Setelah *Sensitivity* untuk masing masing kombinasi data uji dan data latih didapat, *Sensitivity* akan dirata-rata untuk mendapatkan *Global Sensitivity*. Formula untuk menghitung *Global Sensitivity* ditampilkan di Persamaan 2.29.

$$Global\ Sensitivity = \frac{\sum_{i=0}^n Sensitivity_i}{n} \quad (2.29)$$

2.9.3 Specificity

Specificity adalah ukuran kemampuan classifier untuk memisahkan kelas target. Pada penelitian ini, *Specificity* mengukur kemampuan metode yang diusulkan untuk mengidentifikasi kelas Abnormal. Persamaan *Specificity* bisa dilihat di Persamaan 2.30.

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (2.30)$$

Setelah *Specificity* untuk masing masing kombinasi data uji dan data latih didapat, *Specificity* akan dirata-rata untuk mendapatkan *Global Specificity*. Formula untuk menghitung *Global Specificity* ditampilkan di Persamaan 2.31.

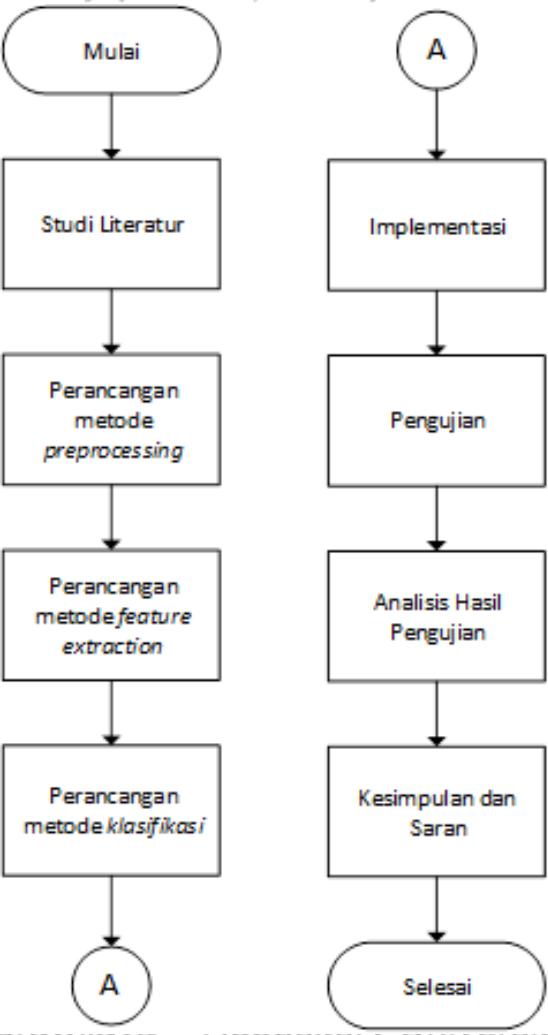
$$Global\ Specificity = \frac{\sum_{i=0}^n Specificity_i}{n} \quad (2.31)$$

Repository Universitas Brawijaya
Repository Universitas Brawijaya

BAB 3 METODOLOGI

3.1 Tahapan Penelitian

Tahapan pada penelitian ini disusun agar proses penelitian dapat dilakukan sesuai skema, berurutan, sistematis dan terencana. Langkah-langkah yang dilakukan dalam penelitian ini digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram alur penelitian

3.1.1 Studi Literatur

Tahapan penelitian ini dimulai dengan studi literatur terhadap berbagai referensi dari sumber-sumber seperti jurnal ilmiah nasional maupun internasional, buku, artikel ataupun buletin yang diterbitkan oleh lembaga terpercaya pada bidangnya. Referensi ini digunakan sebagai dasar teori yang dapat menunjang

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

penelitian. Berdasarkan studi literatur yang telah digali, penelitian ini tersusun menjadi:

1. Penentuan dataset yang digunakan dalam penelitian.
2. ROI Extraction dari gambar mammogram.
3. Peningkatan kontras pada mammogram dengan menggunakan CLAHE.
4. Mengurangi noise pada mammogram dengan menggunakan metode median filter.
5. Feature extraction mammogram dengan menggunakan metode GLCM.
6. Klasifikasi mammogram kedalam kelas normal-abnormal dengan menggunakan metode BPNN.
7. Metode dan teori atau pendekatan lain yang mendukung penelitian ini, digunakan sebagai pembanding ataupun digunakan dan diterapkan dalam penelitian ini.

3.1.2 Perancangan

Setelah studi literatur selesai dilakukan, tahap selanjutnya adalah perancangan. Perancangan pada penelitian ini terbagi menjadi dua, yaitu:

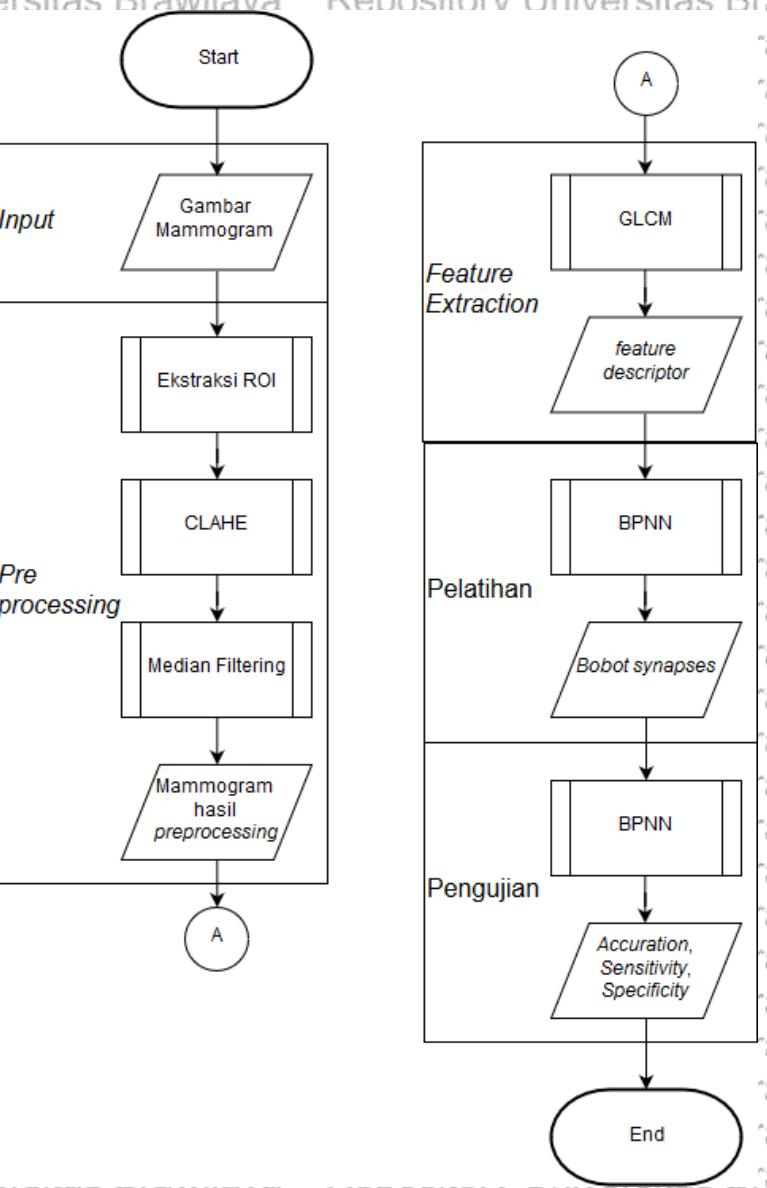
1. Perancangan Program

Setelah melakukan studi literatur, selanjutnya dilakukan perancangan program untuk mengenali tumor payudara menggunakan gambar mammogram. *Flowchart* perancangan program bisa dilihat di Gambar 3.2.

Program yang dirancang dimulai dengan mendapatkan masukan mammogram dalam format PNG berukuran 1024×1024 pixel. Tahap selanjutnya adalah *Preprocessing*. *Preprocessing* dimulai dengan ROI extraction. ROI extraction terdiri dari dua tahap. Pertama, cari pusat koordinat kanker. Setelah itu, dengan menggunakan informasi radius kanker, batas terluar sel kanker didapatkan. Pusat koordinat kanker dan informasi radius kanker didapat dari MIAS dataset. MIAS dataset tidak menyediakan informasi pusat koordinat kanker dan informasi radius kanker untuk mammogram dari kelas normal. Oleh karena itu, ROI kasus normal adalah potongan mammogram yang dibuat dengan asumsi "jika mammogram adalah mammogram sehat, ROI dimanapun akan tetap dianggap sehat". Pusat koordinat untuk kelas normal adalah (512,512) dan radius adalah 256×256 . Tahap preprocessing yang kedua adalah peningkatan kontras ROI menggunakan metode CLAHE. Terakhir, ROI yang sudah ditingkatkan kontrasnya akan menjalani proses noiseremoval menggunakan metode Median Filter.

Tahap selanjutnya adalah *feature extraction*. *Feature extraction* dilakukan dengan menggunakan metode GLCM. Luaran dari GLCM adalah matriks GLCM. Matriks ini akan diubah menjadi fitur dengan menggunakan Persamaan 2.7 sampai Persamaan 2.15. Dengan menggunakan kombinasi jarak dan sudut ketetanggan yang sudah dijelaskan di sub bab 2.8, 16 kumpulan fitur dibuat. Setiap kumpulan fitur terdiri dari 10 *feature descriptor*.

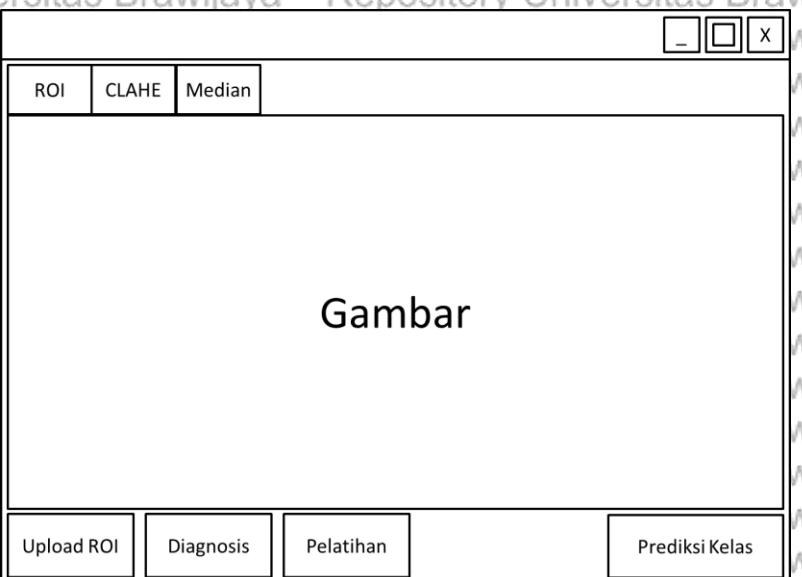
Kumpulan fitur yang dihasilkan pada proses *feature extraction* akan dijadikan masukkan pada *input layer* BPNN. Setiap kumpulan fitur akan diproses secara terpisah. Walaupun begitu, pada tahap pelatihan BPNN, bobot awal *synapses*, jumlah *neuron*, jumlah *epoch* yang digunakan oleh setiap kumpulan fitur dibuat sama.



Gambar 3.2 *Flowchart* perancangan program

2. Perancangan *Interface Layar*

Interface layar program dirancang sedemikian rupa agar pengguna mudah mengoperasikan dan mempertimbangkan hasil yang diberikan. Perancangan *interface* layar bisa dilihat di **Gambar 3.3**.



Gambar

Gambar 3.3 Perancangan interface

3.2 Analisa Kebutuhan

3.2.1 Kebutuhan Software

Konfigurasi *software* yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Sistem Operasi Windows 10 Home Single Language 64-bit
2. Bahasa pemrograman yang digunakan adalah Python versi 3.7
3. Aplikasi dikembangkan dengan menggunakan PyCharm dan Anaconda

3.2.2 Kebutuhan Hardware

Hardware yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Prosessor Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8CPUs), 1.8GHz
2. RAM 8152MB
3. Harddisk 1TB

3.3 Data yang digunakan

Dalam penelitian ini, MIAS dataset akan dibagi menjadi 5 kombinasi data uji dan data latih. Pembagian dataset dilakukan menggunakan metode *k-Fold Cross-Validation*. Implementasi metode *k-Fold Cross-Validation* dilakukan dengan menggunakan *package StratifiedKFold* yang disediakan oleh *Scikit-learn* (Fabian, et al., 2011). *Package StratifiedKFold* mempunyai 3 parameter sebagai berikut:

1. *n_split* : (int), default = 3. Parameter *n_split* menentukan *k* dari *k-Fold Cross-Validation*. Dalam penelitian ini, *n_split* bernilai 5.
2. *shuffle* : (boolean), optional. Parameter *shuffle* menentukan apakah dataset akan diacak terlebih dahulu sebelum dibagi sesuai jumlah *k*. Dalam penelitian ini, *shuffle* bernilai *False* untuk mempermudah pengamatan. Ketika *shuffle* bernilai *False*, *StratifiedKFold* hanya akan membagi dataset tanpa mengubah urutan data.

3. `random_state : (int, RandomState instance or None, optional, default=None)`

Parameter ini hanya digunakan ketika `shuffle` bernilai `True`.

Kombinasi data latih dan data uji dalam penelitian ini adalah sebagai berikut:

1. Kombinasi 1

Tabel 3.1 menampilkan kombinasi data latih dan data uji nomor 1.

Tabel 3.1 Kombinasi 1

Data Latih ROI	
<i>Normal</i>	254
<i>Abnormal</i>	85
Data Uji ROI	
<i>Normal</i>	65
<i>Abnormal</i>	43
<i>Normal</i>	22

2. Kombinasi 2

Tabel 3.2 menampilkan kombinasi data latih dan data uji nomor 2.

Tabel 3.2 Kombinasi 2

Data Latih ROI	
<i>Normal</i>	254
<i>Abnormal</i>	85
Data Uji ROI	
<i>Normal</i>	65
<i>Abnormal</i>	43
<i>Normal</i>	22

3. Kombinasi 3

Tabel 3.3 menampilkan kombinasi data latih dan data uji nomor 3.

Tabel 3.3 Kombinasi 3

Data Latih ROI	
<i>Normal</i>	256
<i>Abnormal</i>	86
Data Uji ROI	
<i>Normal</i>	63
<i>Abnormal</i>	42
<i>Normal</i>	21

4. Kombinasi 4

Tabel 3.4 menampilkan kombinasi data latih dan data uji nomor 4.

Tabel 3.4 Kombinasi 4

Data Latih ROI	
<i>Normal</i>	256
<i>Abnormal</i>	86
Data Uji ROI	
<i>Normal</i>	63
<i>Abnormal</i>	42
<i>Normal</i>	21

5. Kombinasi 5

Tabel 3.5 menampilkan kombinasi data latih dan data uji nomor 5.

Tabel 3.5 Kombinasi 5

Data Latih ROI	256
<i>Normal</i>	170
<i>Abnormal</i>	86
Data Uji ROI	63
<i>Normal</i>	42
<i>Abnormal</i>	21

3.4 Skenario Pengujian

Pengujian yang dilakukan pada penelitian ini ada tiga, yaitu pengujian parameter BPNN yang menghasilkan nilai MSE terkecil, pengujian kombinasi d dan θ yang menghasilkan *Accurcation*, *Sensitivity* dan *Specificity* terbesar serta perbandingan performa metode yang diusulkan dengan metode pembanding.

3.4.1 Pengujian Parameter BPNN

Pada pengujian ini ada 2 parameter BPNN yang diuji, yaitu *Learning Rate* dan jumlah *Epoch*. Pengujian diukur menggunakan Persamaan 2.17. Jumlah *Hidden Neuron* didapat menggunakan Persamaan 2.16. Tabel 3.6 menunjukkan tabel pengujian *Learning Rate*. Tabel 3.7 menunjukkan tabel pengujian jumlah *Epoch* (*Epoch* dalam ribu). MSE yang ditampilkan pada Tabel 3.6 dan Tabel 3.7 adalah rata-rata MSE dari 5 kombinasi data latih dan data uji.

Tabel 3.6 Tabel pengujian Learning Rate

Learning Rate	d	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	
$0,1$	0	45	90	135	0	45												
$0,2$																		
$0,3$																		
$0,4$																		
$0,5$																		
$0,6$																		
$0,7$																		
$0,8$																		
$0,9$																		
1																		

Repository Universitas Brawijaya
Repository Universitas Brawijaya

Tabel 3.7 Tabel pengujian jumlah Epoch

Epoch	d	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
h	θ	0	45	90	135												
5																	
10																	
15																	
20																	
25																	
30																	
35																	
40																	
45																	
50																	
55																	
60																	
65																	
70																	
75																	
80																	

3.4.2 Pengujian Kombinasi d dan θ

Setelah parameter BPNN ditentukan, kemampuan BPNN akan diukur berdasarkan *Accuration*, *Sensitivity* dan *Specificity*. Persamaan untuk menguji *Accuration* ada di Persamaan 2.26, *Sensitivity* ada di Persamaan 2.28 dan *Specificity* ada di Persamaan 2.30. *Accuration*, *Sensitivity* dan *Specificity* yang ditampilkan pada Tabel 3.8 adalah rata-rata *Accuration*, *Sensitivity* dan *Specificity* dari 5 kombinasi data latih dan data uji. Rata-rata *Accuration*, *Sensitivity* dan *Specificity* dihitung menggunakan Persamaan 2.27, Persamaan 2.29 dan Persamaan 2.31 secara berturut-turut.

Tabel 3.8 Tabel pengujian kombinasi d dan θ

d	θ	Accuration	Sensitivity	Specificity
1	0°			
2	0°			
3	0°			
4	0°			
1	45°			
2	45°			
3	45°			
4	45°			
1	90°			
2	90°			
3	90°			
4	90°			
1	135°			
2	135°			
3	135°			
4	135°			

3.4.3 Perbandingan Metode yang Diusulkan dengan Metode Pembanding

Pengujian ini dilakukan untuk mengukur seberapa handal sistem CADx yang dirancang dibandingkan metode pembanding. Tabel pengujian ini bisa dilihat di Tabel 3.9.

Tabel 3.9 Perbandingan metode yang diusulkan dengan metode pembanding

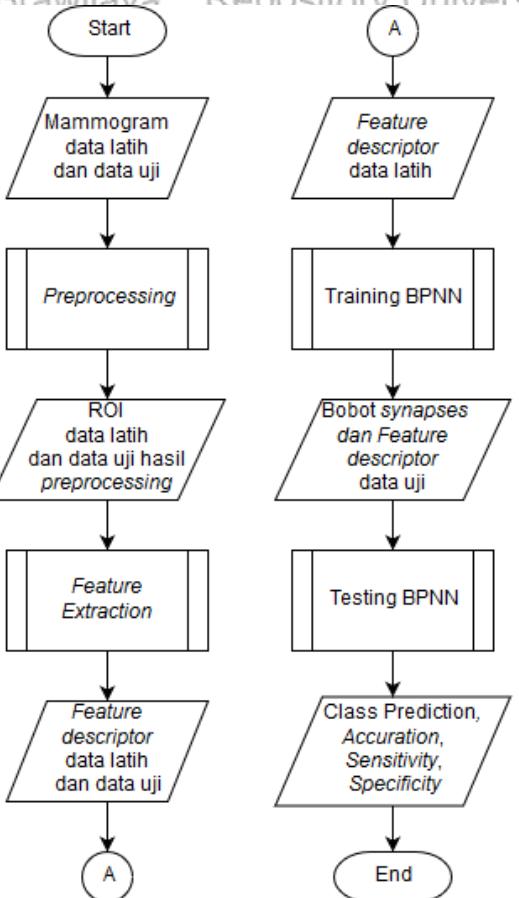
Referensi	Teknik	Performa (normal-abnormal)
(Kral & Lenc, 2016; Kral & Lenc, 2016)	LBP, SVM	Accuratu = 84%
(Pratiwi, Alexander, Harefa, & Nanda, 2015)	GLCM, RBFNN	Accuratu = 93,98% Sensitivity = 97,22%, Specificity = 91,49%
(Setiawan, Elyria, Wesley, & Purnama, 2015)	LAWS, ANN	Accuratu = 93,90%, Sensitivity = 100%, Specificity = 91%
Metode yang diusulkan	GLCM-BPNN	Accuratu =, Sensitivity =, Specificity =

Repository Universitas Brawijaya
Repository Univ

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan diuraikan perancangan, contoh perhitungan rumus matematik yang sudah diuraikan pada Bab 2 dan implementasi metode yang diusulkan.

4.1 Perancangan



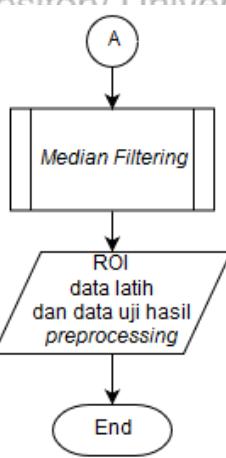
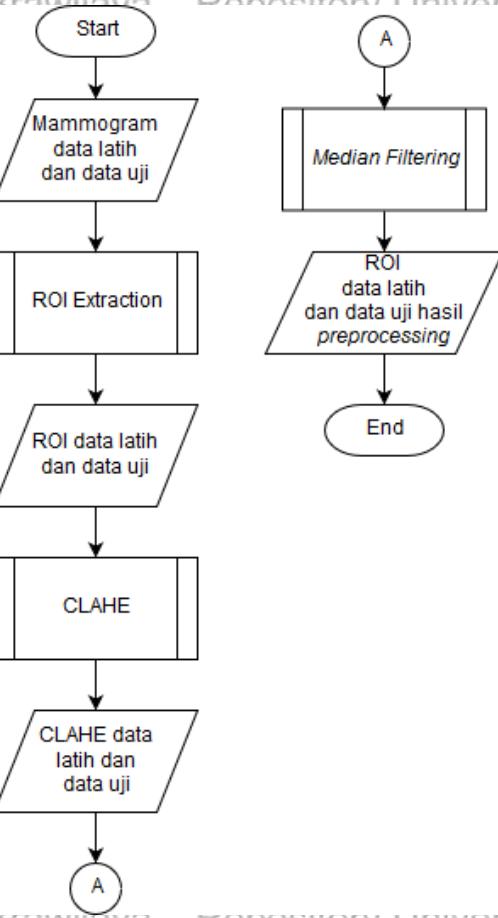
Gambar 4.1 Flowchart metode yang diusulkan

Flowchart metode yang diusulkan diuraikan pada Gambar 4.1. Penjelasan dari Gambar 4.1 adalah sebagai berikut:

1. Metode yang diusulkan dimulai dengan mengambil gambar mammogram data latih dan data uji sebagai *input* sistem.
2. Setelah itu, *preprocessing* akan dilakukan terhadap masing-masing Gambar mammogram data latih dan data uji. *Output* dari tahap *preprocessing* ada 2, yaitu ROI data latih hasil *preprocessing* dan ROI data uji hasil *preprocessing*.
3. Selanjutnya, pada tahap *Feature Extraction*, ROI data latih dan data uji hasil *preprocessing* akan diubah menjadi matriks GLCM. Dari matriks GLCM, kumpulan *feature descriptor* data latih dan data uji dihasilkan. Perhitungan *feature descriptor* dilakukan dengan menggunakan Persamaan 2.6 sampai Persamaan 2.15.

4. Proses selanjutnya adalah pelatihan BPNN. Proses ini membutuhkan *input array feature descriptor* data latih. *Output* dari proses ini adalah *bobot synapses* BPNN.
5. Terakhir, pengujian BPNN. Proses ini membutuhkan *input* *bobot synapses* BPNN dan *feature descriptor* data uji. *Output* dari proses ini adalah *Class Prediction, Accurartion, Sensitivity* dan *Specificity*.

4.1.1 Preprocessing

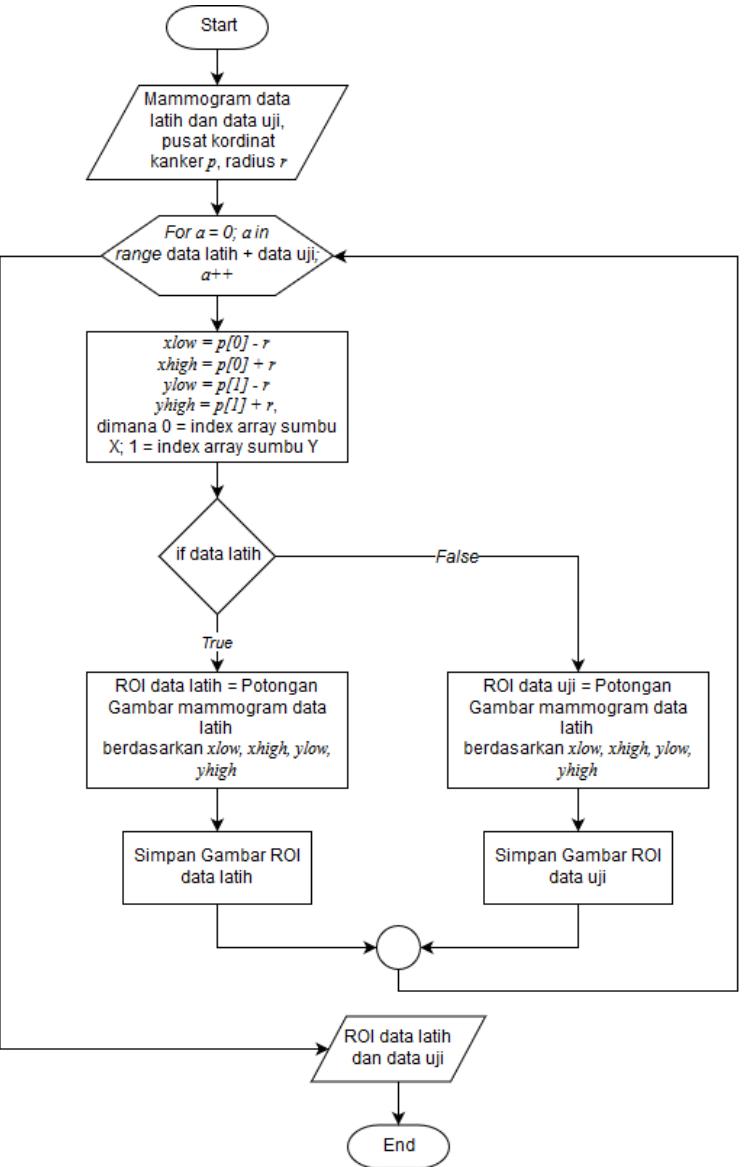


Gambar 4.2 Flowchart Preprocessing

Flowchart Preprocessing pada metode yang diusulkan diuraikan pada Gambar 4.2.

4.2. Tahap preprocessing adalah sebagai berikut:

1. *Preprocessing* dimulai dengan mengekstrak ROI dari mammogram data latih dan data uji. Pengambilan ROI ini dilakukan pada proses *ROI Extraction*. Luaran dari proses *ROI Extraction* adalah *ROI data latih dan data uji*.
2. Setelah *ROI data latih dan data uji* didapat, kontras *ROI data latih dan data uji* akan ditingkatkan pada proses *CLAHE*. *Output* dari proses *CLAHE* adalah *CLAHE data latih dan data uji*.
3. Terakhir, *noise* di *CLAHE data latih dan data uji* akan dikurangi/dihilangkan pada proses *Median Filter*. *Output* dari proses *Median Filter* adalah *ROI data latih dan data uji hasil preprocessing*.

4.1.1.1 ROI Extraction**Gambar 4.3 Flowchart ROI Extraction**

Flowchart ROI Extraction ditunjukkan pada Gambar 4.3. ROI Extraction membutuhkan *input f_train, f_test*, pusat kordinat kanker *p*, dan radius *r*. Setelah semua *input* didapatkan, tahap selanjutnya adalah perhitungan *x_low, x_high, y_low, y_high*. Terakhir, bagian mammogram akan diambil sesuai *x_low, x_high, y_low, y_high*. Proses ini dilakukan untuk semua *f_train* dan *f_test*. *Output* dari proses ini adalah *g_train, g_test*.

Diberikan $p = [100, 200]$ untuk sumbu-X dan sumbu-Y dan $r = 50$, maka didapatkan batas-batas ROI sebagai berikut:

$$x_{low} = 100 - 50 = 50$$

$$x_{high} = 100 + 50 = 150$$

$$y_{low} = 200 - 50 = 150$$

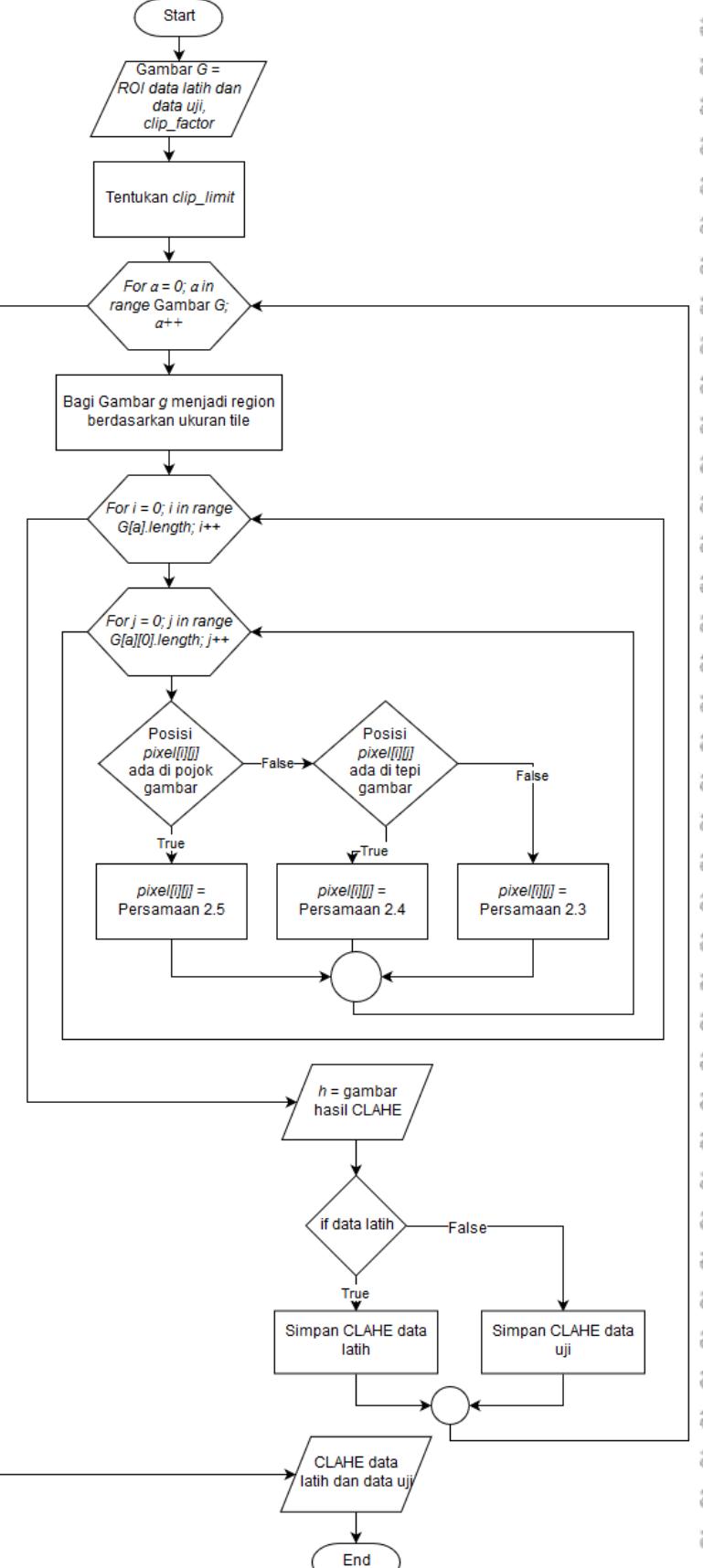
$$y_{high} = 200 + 50 = 250$$

Menggunakan $x_low, x_high, y_low, y_high$ yang sudah didapat, ROI akan diambil mulai dari kordinat [50, 150] sampai [150, 250].

4.1.1.2 CLAHE

Flowchart CLAHE ditunjukkan pada Gambar 4.4. Penjelasan Gambar 4.4 adalah sebagai berikut:

1. CLAHE dimulai dengan mengambil gambar ROI data latih dan data uji.
2. Menentukan *clip limit histogram* akan menggunakan Persamaan 2.2.
3. Bagi ROI akan region sesuai jumlah yang sudah ditentukan.
4. Hitung nilai pixel baru dengan syarat
 - a. Jika pixel berada di pojok gambar, hitung nilai baru pixel dengan menggunakan Persamaan 2.3.
 - b. Jika pixel berada di tepi gambar, hitung nilai baru pixel dengan menggunakan Persamaan 2.4.
 - c. Selain pixel pada posisi poin a atau b, hitung nilai baru pixel dengan menggunakan Persamaan 2.5.
5. Jika ada nilai bin histogram yang melebihi *clip limit histogram*, sebarkan nilai di bin tersebut ke bin lain yang memiliki jumlah kurang dari *clip limit histogram*.
6. Setelah CLAHE selesai, simpan gambar hasil CLAHE sesuai jenisnya. Jika data adalah data latih, simpan sebagai CLAHE data latih. Jika data adalah data uji, simpan sebagai CLAHE data uji.
7. Ulangi langkah 3 sampai 6 untuk semua gambar ROI data latih dan data uji.



Gambar 4.4 Flowchart CLAHE

Diberikan gambar f seperti yang tertera di Gambar 4.5. Gambar f adalah gambar grayscale dengan dimensi 16×16 .

4	13	1	1	10	0	8	12	11	2	4	4	13	7	12	11
3	7	4	3	15	9	4	4	6	12	0	9	0	7	8	7
11	14	8	12	3	11	11	13	3	9	4	12	6	4	15	11
12	7	14	5	4	11	4	12	8	0	1	9	2	4	2	2
13	7	4	10	8	14	6	0	4	12	11	0	10	1	4	6
2	12	2	1	15	13	13	6	12	10	11	12	0	5	7	13
1	15	15	1	15	13	8	2	15	0	0	7	8	7	1	9
4	5	13	5	15	6	10	5	1	15	10	3	4	10	5	15
3	5	15	2	10	1	6	15	11	10	13	15	7	12	5	4
4	12	14	2	13	13	12	1	5	5	13	1	5	5	13	15
11	1	12	5	7	2	6	9	13	6	6	1	8	15	8	3
1	9	13	10	5	1	11	5	14	12	14	7	1	8	7	4
7	10	9	11	6	2	12	12	3	15	1	6	1	3	12	15
13	0	14	3	13	3	8	15	8	14	5	14	3	10	7	1
3	14	5	13	0	4	13	9	9	9	14	14	13	12	14	13
0	9	5	9	8	1	14	4	5	2	8	1	12	7	5	15

Gambar 4.5 Gambar f

4	13	1	1	10	0	8	12	11	2	4	4	13	7	12	11
3	7	4	3	15	9	4	4	6	12	0	9	0	7	8	7
11	14	8	12	3	11	11	13	3	9	4	12	6	4	15	11
12	7	14	5	4	11	4	12	8	0	1	9	2	4	2	2
13	7	4	10	8	14	6	0	4	12	11	0	10	1	4	6
2	12	2	1	15	13	13	6	12	10	11	12	0	5	7	13
1	15	15	1	15	13	8	2	15	0	0	7	8	7	1	9
4	5	13	5	15	6	10	5	1	15	10	3	4	10	5	15
3	5	15	2	10	1	6	15	11	10	13	15	7	12	5	4
4	12	14	2	13	13	12	1	5	5	13	1	5	5	13	15
11	1	12	5	7	2	6	9	13	6	6	1	8	15	8	3
1	9	13	10	5	1	11	5	14	12	14	7	1	8	7	4
7	10	9	11	6	2	12	12	3	15	1	6	1	3	12	15
13	0	14	3	13	3	8	15	8	14	5	14	3	10	7	1
3	14	5	13	0	4	13	9	9	9	14	14	13	12	14	13
0	9	5	9	8	1	14	4	5	2	8	1	12	7	5	15

Gambar 4.6 Region Gambar f

Tahap *preprocessing* yang pertama adalah CLAHE. Langkah pertama pada CLAHE adalah membagi gambar f menjadi region. Gambar 4.6 menampilkan pembagian Gambar f menjadi region. Setiap region berukuran 2×2 . Warna merah adalah CR, biru adalah BR dan hitam adalah IR.

Setelah Gambar f dibagi kedalam region, langkah selanjutnya adalah menentukan *clip limit*. Rumus perhitungan *clip limit* ditampilkan pada Persamaan 2.2. Ditetapkan $\alpha = 20$, $s_{max} = 4$ dan gambar yang digunakan adalah gambar F , maka *clip limit* yang didapat adalah

$$\beta = \frac{M}{N} \left(1 + \frac{\alpha}{100} (s_{max} - 1) \right) = \frac{256}{16} \left(1 + \frac{20}{100} (4 - 1) \right) = 25,6 \approx 26$$

Langkah selanjutnya adalah meng dalam gambar *f*. Gambar 4.7 me Gambar 4.6.

Langkah selanjutnya adalah menghitung nilai grayscale baru untuk setiap pixel dalam gambar f . Gambar 4.7 menampilkan empat region pada posisi kiri atas Gambar 4.6.

Repository Universitas Brawijaya
Repository Universitas Brawijaya 0
Repository Universitas Brawijaya 1
Repository Universitas Brawijaya 2
Repository Universitas Brawijaya 3
Gambar 4.1

0	1	2	3
4	13	1	1
3	7	4	3
11	14	8	12
12	7	14	5

Gambar 4.7 Potongan Gambar 4.6

Nilai pixel baru yang akan dicari adalah pixel di posisi [2,2] pada Gambar 4.7. Rumus perhitungan nilai pixel yang digunakan ada pada Persamaan 2.3.

$$p_{new} = \frac{s}{r+s} \left(\frac{y}{x+y} f_{i-1,j-1}(p_{old}) + \frac{x}{x+y} f_{i,j-1}(p_{old}) \right) + \frac{r}{r+s} \left(\frac{y}{x+y} f_{i-1,j}(p_{old}) + \frac{x}{x+y} f_{i,j}(p_{old}) \right)$$

Untuk mempermudah, tentukan nilai dari masing-masing variabel yang dibutuhkan.

$s = 5$; pixel posisi [3,3]

r = 7 ± 14 = 21: pixel posisi [1,1], [2,1]

$x \equiv 7 \pm 14 \equiv 21$: pixel posisi [3,1], [3,2]

v ≡ 3 + 12 ≡ 15: pixel posisi [1 3] [2 3]

Repository Universitas Brawijaya

Repository Universitas Brawijaya
Berpengaruh Pada Perilaku Penggunaan

Repository Universitas Brawijaya

Repository | Universitas Brawijaya

Repository Universitas Brawijaya

$$p_{new} = \frac{3}{21+5} \left(\frac{13}{21+15} 7 + \frac{z_1}{21+15} 4 \right) + \frac{z_1}{21+5} \left(\frac{13}{21+15} 14 + \frac{z_1}{21+15} 8 \right)$$

Universitas Brawijaya Repository Universitas Brawijaya

26(36) 36' 26(36¹) 36²) Repository Universitas Brawijaya
Repository Universitas Brawijaya

Cara perhitungan yang sama diterapkan untuk semua pivil yang memiliki tetapan

Cara perhitungan yang sama diterapkan untuk semua pixel yang memiliki tetangga di arah $(i-1, j+1)$, $(i, j+1)$, $(i+1, j)$. Dengan menggunakan Persamaan 2.4, dicari nilai pixel baru untuk pixel di posisi [2,0] dari Gambar 4.7.

$$p_{new} = \frac{s}{r+s} f_{i,j-1}(p_{old}) + \frac{r}{r+s} f_{i,j}(p_{old})$$

Untuk mempermudah, tentukan nilai dari masing masing variabel yang dibutuhkan.

$s = 12$; pixel posisi [3,0]

$r = 4 + 3 = 7$; pixel posisi [1,0], [2,0]

$f_{i,j-1}(p_{old}) = 3$; pixel posisi [1,0]

Repository Universitas Brawijaya
Brawijaya University Repository

$$p_{new} = \frac{12}{7+12} 3 + \frac{7}{7+12} 11$$
$$= \frac{12}{19} 3 + \frac{7}{19} 11$$
$$= 1,89 + 4,05 = 5,94 \approx 6$$

Perhitungan menggunakan Persamaan 2.4 dilakukan untuk semua pixel yang memiliki tetangga di posisi $f_{i,j-1}$ dan $f_{i,j}$. Terakhir, hitung nilai baru untuk pixel di posisi [0,0] dari Gambar 4.7. Perhitungan pixel di posisi [0,0] menggunakan Persamaan 2.5.

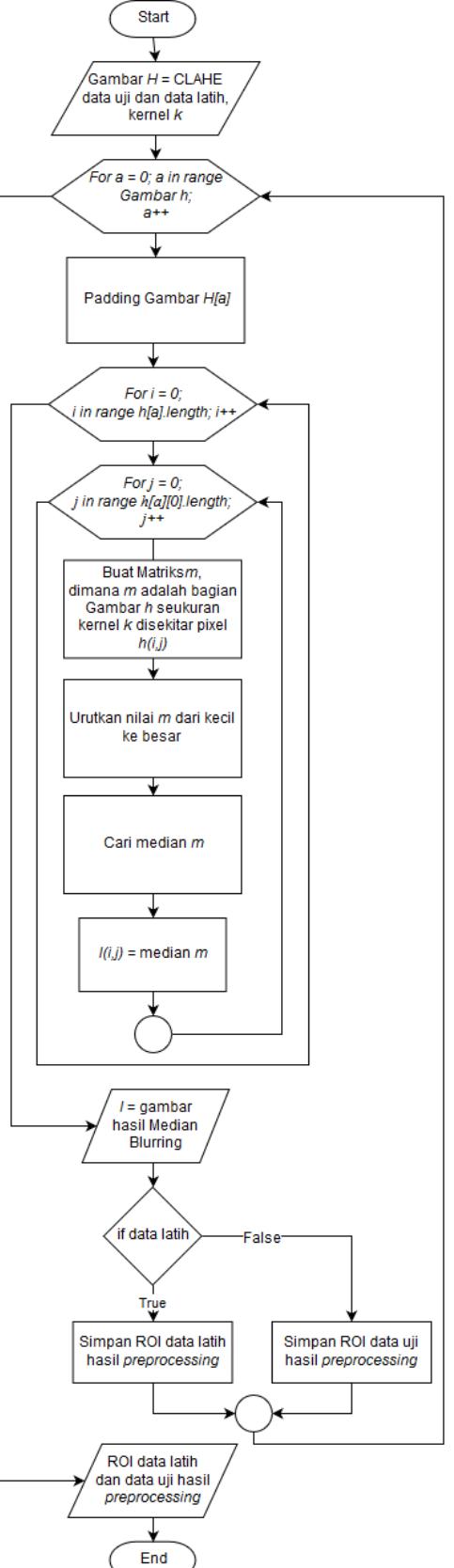
$$p_{new} = f_{i,j}(p_{old}) = 4$$

Gunakan Persamaan 2.5 untuk menghitung pixel pada posisi yang menyerupai pixel di posisi [0,0]. Setiap nilai pixel baru hasil CLAHE akan dimasukkan kedalam bin histogram. Selama tinggi di salah satu bin belum mencapai β , nilai di bin tersebut bisa terus bertambah. Jika tinggi salah satu bin sudah melebihi nilai β , tinggi bin tersebut akan dipotong dan disebarluaskan ke bin lain yang jumlahnya kurang dari β .

4.1.1.3 Median Filter

Tahap *preprocessing* yang terakhir adalah *median filter*. Algoritme median filter ditampilkan di Gambar 4.8. Alur kerja *median filter* adalah sebagai berikut:

1. Mengambil gambar CLAHE data latih dan data uji sebagai *input*.
2. Menentukan ukuran kernel k .
3. Menambahkan *padding* pada masing masing gambar ROI data latih dan data uji.
4. Menentukan pixel yang akan dirubah nilainya ($p[i][j]$).
5. Membuat matriks m , dimana m berisi pixel $p[i][j]$ dan tetangganya seukuran k .
6. Mengurutkan nilai dalam matriks m mulai dari besar ke kecil.
7. Mencari median matriks m .
8. Mengulangi langkah 4 sampai 7 hingga semua pixel dalam gambar mengalami proses *median filter*.
9. Setelah *median filter* selesai, simpan gambar hasil *median filter* sesuai jenisnya. Jika data adalah data latih, simpan sebagai ROI data latih hasil *preprocessing*. Jika data adalah data uji, simpan sebagai ROI data uji hasil *preprocessing*.
10. Mengulangi langkah 4 sampai 9 dilakukan untuk semua gambar CLAHE data uji dan data latih.



Gambar 4.8 Flowchart Median Filter

Repository Universitas Brawijaya
Dalam ilustrasi *median filter*, digunakan Gambar 4.9. Pada Gambar 4.9 ada penambahan baris dan kolom terluar dengan nilai 0. Penambahan kolom dan baris ini disebut *padding*. Penambahan padding dilakukan untuk menjaga agar ukuran gambar sebelum dan sesudah mengalami proses *Median Filter* tetap sama.

0	1	2	3	4	5
0	0	0	0	0	0
1	0	4	13	1	1
2	0	3	7	4	3
3	0	11	14	8	12
4	0	12	7	14	5
5	0	0	0	0	0

Gambar 4.9 Penambahan *Padding* pada Gambar 4.7

Langkah pertama dalam *median filter* adalah menentukan ukuran kernel. Dalam ilustrasi ini, kernel yang digunakan berukuran 3×3 (warna hijau pada Gambar 4.7). Kernel digunakan untuk menandai bagian gambar yang sedang mengalami proses *median filter*. Kernel akan digerakkan dengan pergeseran 1 pixel ke arah kanan. Pada ilustrasi ini, akan dicari median untuk menggantikan nilai pixel di posisi [1,1].

Langkah kedua dalam *median filter* adalah mengurutkan nilai pixel didalam kernel. Gambar 4.10 menampilkan hasil pengurutan nilai pixel didalam kernel. Median dari Gambar 4.10 adalah 0. Oleh karena itu, nilai pixel di [1,1] diganti dengan 0.

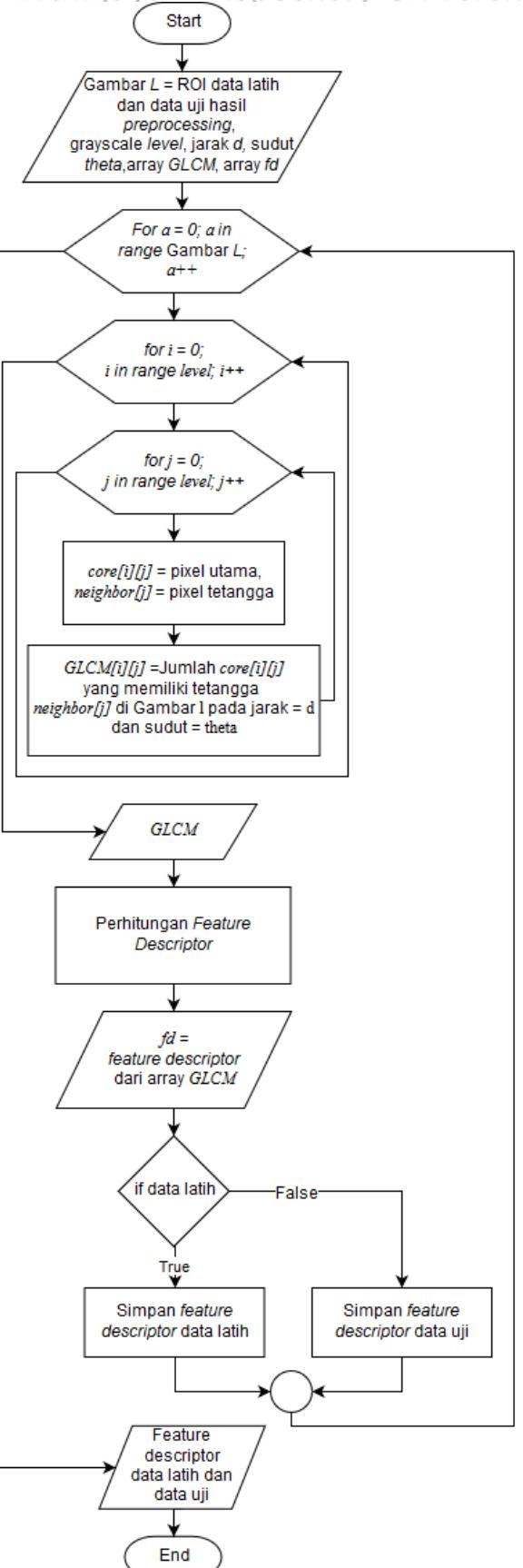
0	0	0	0	0	3	4	7	13
---	---	---	---	---	---	---	---	----

Gambar 4.10 Urutan nilai didalam kernel pada Gambar 4.9

4.1.2 Feature Extraction

Feature extraction dilakukan dengan menggunakan metode GLCM. Algoritme GLCM ditampilkan pada Gambar 4.11. Alur kerja GLCM adalah sebagai berikut:

1. Mengambil gambar ROI data latih dan data uji *preprocessing* sebagai input.
2. Menentukan maksimal nilai *grayscale level*, jarak *d* dan sudut *theta*.
3. Menentukan *core*[*i*][*j*] dan *neighbor*[*j*], dimana *core*[*i*][*j*] adalah pixel utama dan *neighbor*[*j*] adalah pixel tetangga.
4. Menghitung *GLCM*[*i*][*j*], dimana *GLCM*[*i*][*j*] adalah jumlah *core*[*i*][*j*] yang memiliki tetangga *neighbor*[*j*] pada jarak *d* dan sudut *theta* dalam gambar *i*.
5. Mengulangi langkah 3 dan 4 hingga matriks *GLCM* selesai terbentuk.
6. Menghitung *feature descriptor* dari matriks *GLCM*.
7. Setelah perhitungan *feature descriptor* selesai, simpan *feature descriptor* sesuai jenisnya. Jika data adalah data latih, simpan sebagai *feature descriptor* data latih. Jika data adalah data uji, simpan sebagai *feature descriptor* data uji.
8. Mengulangi langkah 3 sampai 7 hingga semua gambar *median filter l_train* dan *l_test* mengalami proses *feature extraction*.



Gambar 4.11 Flowchart Feature Extraction



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	1	0	0	2	1	1	2	2	3	2	1	2	2	1	1
1	1	2	1	1	4	4	3	4	3	3	2	3	2	1	6
2	0	1	2	0	3	1	2	1	2	1	1	3	1	1	2
3	0	1	0	0	2	1	0	1	2	1	2	1	3	3	2
4	2	1	3	2	4	4	3	3	0	2	2	2	4	3	1
5	1	4	1	1	4	4	0	3	0	2	3	1	2	5	5
6	1	4	2	0	3	0	2	0	0	1	1	1	3	2	1
7	2	3	1	1	3	3	0	0	5	0	2	0	4	3	2
8	2	4	2	2	0	0	0	5	0	1	1	0	3	1	3
9	3	3	1	1	2	2	1	0	1	4	1	1	0	3	1
10	2	3	1	2	2	3	1	2	1	1	0	2	1	2	0
11	1	2	1	1	2	1	1	0	0	1	2	2	4	1	2
12	2	3	3	3	4	2	3	4	3	0	1	4	2	2	4
13	2	2	1	3	3	5	2	3	1	3	2	1	2	4	2
14	1	1	1	3	1	5	1	2	3	1	0	1	4	2	2
15	1	6	2	2	1	4	2	1	4	1	1	2	1	4	0

Gambar 4.12 Matriks GLCM $M_{i,j}$

4.1.2.2 GLCM

Contoh perhitungan *feature descriptor* GLCM akan dilakukan dengan menggunakan matriks GLCM yang dihasilkan dari Gambar 4.5. Pembentukan matriks GLCM dilakukan dengan menggunakan aturan simetrik, $d = 1$ dan $\theta = 0^{\circ}$. Matriks GLCM $M_{i,j}$ ditampilkan di Gambar 4.12. Contoh perhitungan *feature descriptor* GLCM adalah sebagai berikut:

1. Mean

Mean dihitung dengan menggunakan Persamaan 2.6.

$$\text{Mean} = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{i,j}}{mn} = \frac{2+1+\dots+0+2}{16 \times 16} = \frac{480}{256} = 1,875$$

2. Standard Deviation

Standard deviation dihitung dengan menggunakan Persamaan 2.7.

$$\begin{aligned} \text{Std. Deviation} &= \sqrt{\frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (M_{i,j} - \text{Mean})^2}{mn}} \\ &= \sqrt{\frac{(2-1,875)^2 + \dots + (0-1,875)^2 + (2-1,875)^2}{256}} \\ &= 1,317 \end{aligned}$$

3. Variance

Variance dihitung dengan menggunakan Persamaan 2.8

$$\text{Variance} = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (M_{(i,j)} - \text{Mean})^2}{mn}$$
$$= \frac{(2 - 1,875)^2 + (1 - 1,875)^2 + \dots + (0 - 1,875)^2 + (2 - 1,875)^2}{256}$$

$$= 1.734$$

4. Contrast

Contrast dihitung dengan menggunakan Persamaan 2.9

$$\text{Contrast} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{(i,j)} (i - j)^2$$
$$= (2 \times (0 - 0))^2 + (1 \times (0 - 1))^2 + \dots + (0 \times (15 - 14))^2 + (2 \times (15 - 15))^2$$
$$= 23.260$$

5. Dissimilarity

Dissimilarity dihitung dengan menggunakan Persamaan 2.10

$$\text{Dissimilarity} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{(i,j)} |i - j|$$
$$= (2 \times |0 - 0|) + \dots + (0 \times |15 - 14|) + (2 \times |15 - 14|)$$
$$= 2.788$$

6. Homogeneity

Homogeneity dihitung dengan menggunakan persamaan 2.11

$$\text{Homogeneity} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \left[\frac{M_{(i,j)}}{1 + (i - j)^2} \right]$$
$$= \left[\frac{2}{1 + (0 - 0)^2} \right] + \dots + \left[\frac{0}{1 + (15 - 14)^2} \right] + \left[\frac{2}{1 + (15 - 15)^2} \right]$$
$$= 76.211$$

7. ASM

ASM dihitung dengan menggunakan Persamaan 2.12

$$\text{ASM} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (M_{(i,j)})^2 = 2^2 + 6^2 + \dots + 0^2 + 2^2 = 1344$$

8. Energy

Energy dihitung dengan menggunakan Persamaan 2.13

$$\text{Energy} = \sqrt{\text{ASM}} = \sqrt{1344} = 36.661$$

9. Maximum Probability

Maximum Probability dihitung dengan menggunakan Persamaan 2.14

$$\text{Maximum Probability} = \max(M_{(i,j)}) = 6$$

10. Sum Average

Sum Average dihitung dengan menggunakan Persamaan 2.15

$$\text{Sum Average} = \frac{1}{2} \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} M_{(i,j)}(i+j)$$
$$= \frac{1}{2} ((2 * (0 + 0)) + \dots + (0 * (15 + 14)) + (2 * (15 + 15)))$$
$$= 3692$$

4.1.3 BPNN

Arsitektur BPNN metode yang diusulkan ditampilkan dalam Gambar 4.13.

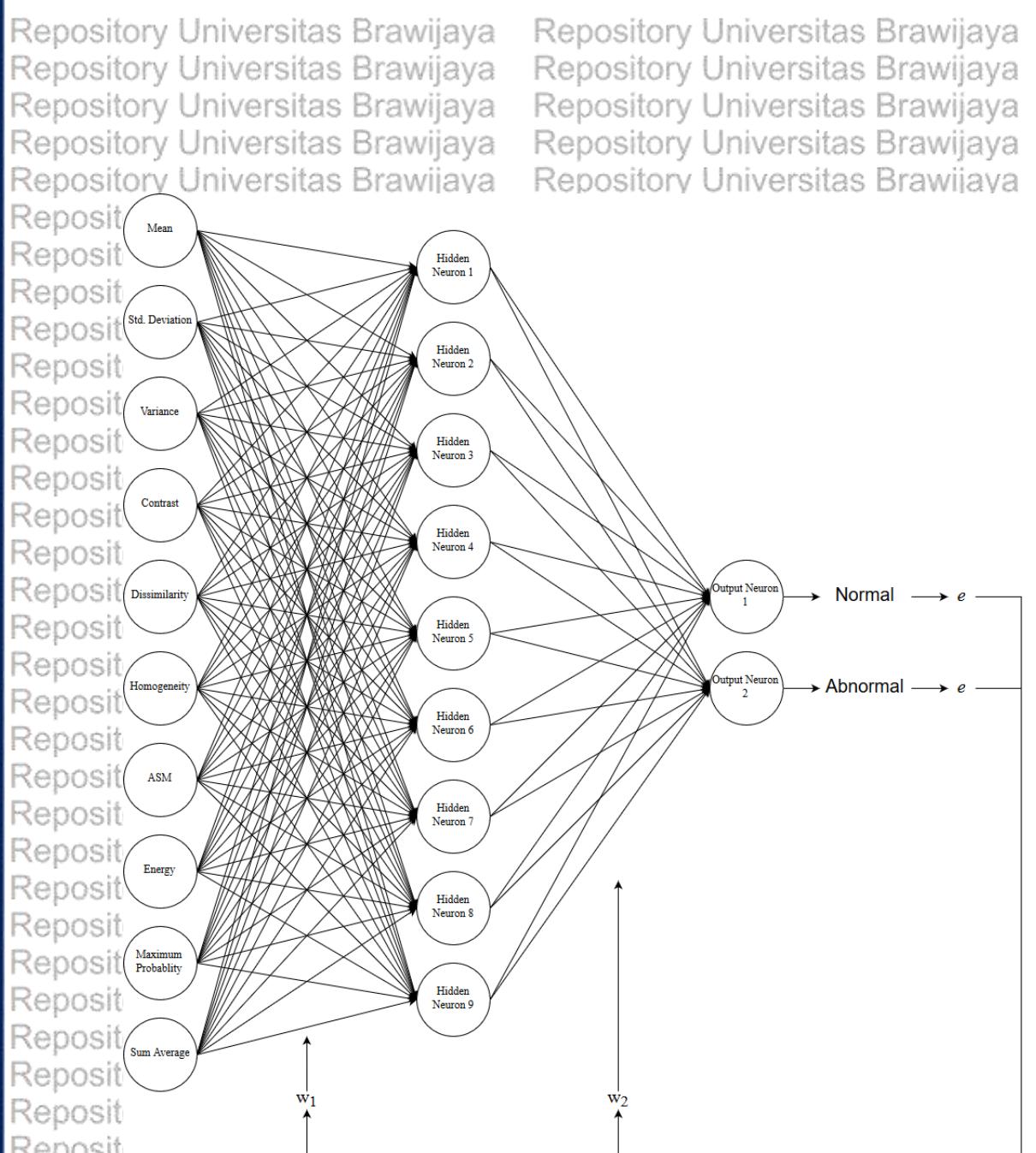
BPNN yang dirancang terdiri dari 3 layer, yaitu *input layer*, *hidden layer* dan *output layer*. *Feature descriptor* dijadikan *input* pada *input layer* BPNN. 1 *input neuron* memiliki 9 *output synapses* yang mengarah ke *Hidden Neuron* yang berbeda beda. Pada *hidden layer*, ada 9 *Hidden Neuron*. Masing masing *Hidden Neuron* memiliki 1 *synapses input* dan 2 *synapses output*. Terakhir, pada *output layer*, ada 2 *output neuron*. *Output neuron* yang pertama adalah Normal dan *output neuron* yang kedua adalah Abnormal.

Asumsikan *output* dari BPNN adalah array 1 dimensi, yang memiliki panjang 2. Pada tahap pelatihan, *output* BPNN akan diarahkan sedemikian rupa untuk mencapai target yang diberikan. Jika data latih berasal dari kelas Normal, target yang harus dicapai BPNN adalah [1,0]. Jika data latih berasal dari kelas Abnormal, target yang harus dicapai BPNN adalah [0,1].

Setelah BPNN selesai dilatih, BPNN akan memasuki tahap pengujian. Pada tahap pengujian, BPNN akan memprediksi *feature descriptor* yang dimasukkan termasuk kelas Normal atau Abnormal. Asumsikan output dari BPNN adalah array *o*, maka prediksi yang dikeluarkan BPNN harus memenuhi 2 syarat, yaitu:

1. $o[0] > o[1]$, maka data uji berasal dari kelas Normal. Syarat ini berlaku karena pada tahap pelatihan, kelas Normal memiliki target [1,0]. Dimana indeks ke-0 dari array *o* memiliki nilai 1 dan indeks ke-1 dari array *o* memiliki nilai 0.

2. $o[1] \geq o[0]$, maka data uji berasal dari kelas Abnormal. Syarat ini berlaku karena pada tahap pelatihan, kelas Abnormal memiliki target [0,1]. Dimana indeks ke-1 dari array *o* memiliki nilai 1 dan indeks ke-0 dari array *o* memiliki nilai 0.

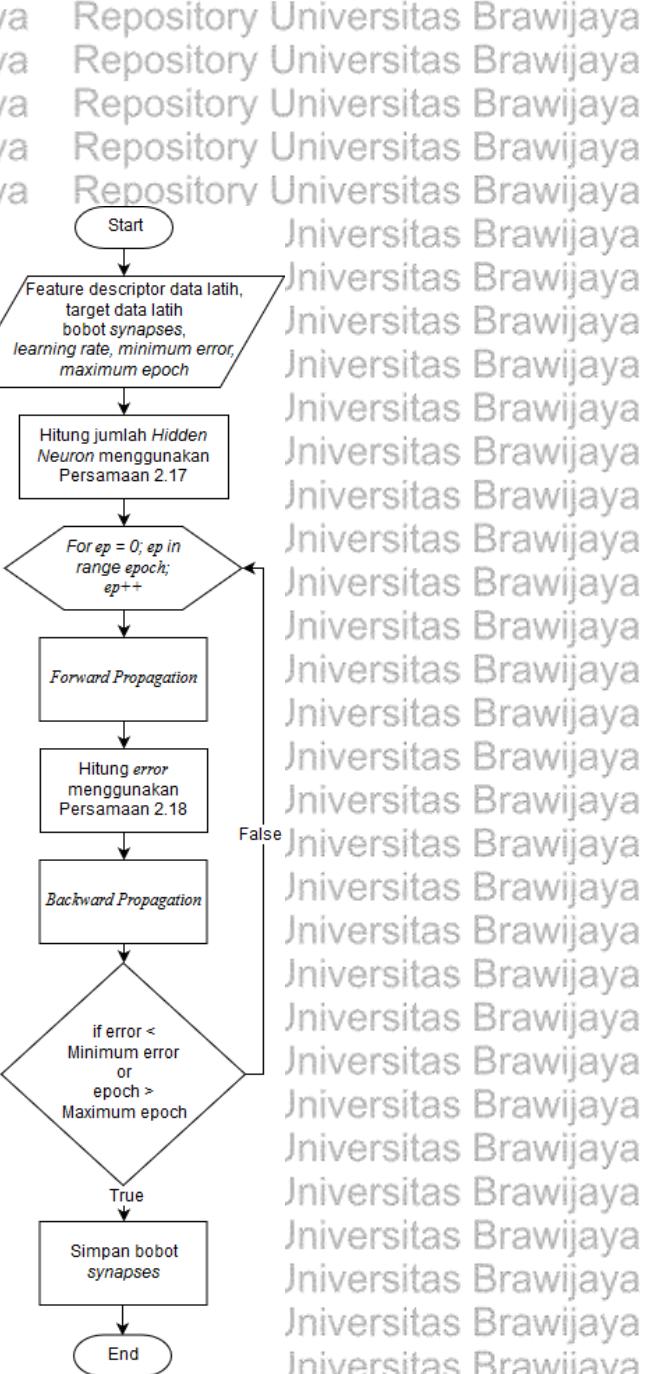


Gambar 4.13 Arsitektur BPNN metode yang diusulkan

4.1.3.1 Training BPNN

Gambar 4.14 menampilkan *flowchart training* BPNN. Penjelasan Gambar 4.14 adalah sebagai berikut:

1. Mengambil *feature descriptor* data latih dan target data latih
2. Menginisialisasi *bias*, bobot *synapses*, *learning rate*, *minimum error* dan *maximum epoch*.
3. *Forward Propagation*
4. Menghitung *error* menggunakan Persamaan 2.21.
5. *Backward Propagation*
6. Ulangi langkah 3-5 sampai *error* kurang dari *minimum error* atau *epoch* lebih dari *maximum epoch*.

**Gambar 4.14 Flowchart training BPNN**

Diberikan matriks *feature descriptor* data latih sebagai berikut:

$$fd_{train} = \begin{bmatrix} 47,254 & 436,202 & 19027,227 & 34179920,000 & 6702840,000 \\ 5,770 & 77,809 & 605,431 & 3161080,000 & 706160,000 \\ 19,922 & 226,302 & 5121,274 & 15811780,000 & 2849700,000 \\ 1,080 & 9,007 & 8,113 & 2690540,000 & 295660,000 \\ 19,922 & 230,037 & 5291,718 & 15062440,000 & 2959160,000 \end{bmatrix}$$

Untuk mempermudah ilustrasi, *feature descriptor* yang digunakan hanya 5, yaitu *Mean*, *Standard Deviation*, *Variance*, *Contrast* dan *Dissimilarity*. Dikarenakan jarak antara setiap elemen *feature descriptor* data latih dan *target* yang terlalu jauh, *feature descriptor* data latih harus dinormalisasi terlebih dahulu. Normalisasi dilakukan menggunakan normalisasi Z – Score. Normalisasi Z-Score dihitung menggunakan Persamaan 2.16:

Target sebagai berikut:

$$f_{d_train} = \begin{bmatrix} 1,768 \\ -0,809 \\ 0,07 \\ -1,1 \\ 0,07 \end{bmatrix}, \begin{bmatrix} 1,63 \\ -0,801 \\ 0,206 \\ -1,267 \\ 0,232 \end{bmatrix}, \begin{bmatrix} 1,895 \\ -0,787 \\ -0,129 \\ -0,874 \\ -0,105 \end{bmatrix}, \begin{bmatrix} 1,745 \\ -0,961 \\ 0,142 \\ -1,003 \\ 0,077 \end{bmatrix}, \begin{bmatrix} 1,759 \\ -0,878 \\ 0,065 \\ -1,058 \\ 0,113 \end{bmatrix}$$

$$target = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Jumlah epoch 1000. Jumlah hidden layer BPNN ditentukan dengan menggunakan Persamaan 2.17.

$$n_h = \left(\frac{2}{3} * n_i\right) + n_o = \left(\frac{2}{3} * 5\right) + 2 = 5,333 \approx 5$$

Inisialisasi bobot synapses input layer ke hidden layer sebagai berikut:

$$w_{ih} = \begin{bmatrix} -1,316 & 0,577 & -1,2 & -0,352 & -0,023 \\ 1,473 & -0,184 & -1,224 & 1,916 & -0,116 \\ 0,122 & 1,15 & -0,057 & 1,525 & 0,278 \\ -0,478 & 0,295 & 0,399 & 1,39 & -0,84 \\ -1,77 & -0,188 & 1,389 & 1,539 & -2,089 \end{bmatrix}$$

Inisialisasi bobot hidden layer ke output layer sebagai berikut:

$$w_{ho} = \begin{bmatrix} 0,157 & -1,429 \\ 1,372 & -0,445 \\ -0,675 & 0,473 \\ -0,666 & 1,47 \\ 0,981 & 0,214 \end{bmatrix}$$

Pelatihan dalam BPNN dibagi menjadi dua, yaitu *forward propagation* dan *backward propagation*. Pada *forward propagation*, BPNN hanya akan menentukan *output* yang dihasilkan. Ketika *output* yang dihasilkan tidak sesuai target, BPNN akan melakukan perbaikan. Perbaikan ini terjadi pada tahap *backward propagation*.

1. Forward Propagation

Forward propagation terbagi menjadi dua tahap perhitungan. Pertama, *forward propagation* antara f_d dan w_{ih} menggunakan Persamaan 2.18. bias yang digunakan adalah 1.

$$z = \begin{bmatrix} -2,642 & 4,083 & -0,086 & 11,552 & -3,842 \\ 2,802 & -0,343 & 1,393 & -4,137 & 3,534 \\ 1,013 & 0,884 & 0,817 & 1,47 & 0,684 \\ 2,827 & -0,503 & 2,051 & -5,396 & 3,982 \\ 1 & 0,879 & 0,825 & 1,54 & 0,642 \end{bmatrix}$$

Setelah z didapat, terapkan fungsi aktivasi sigmoid yang tertera di Persamaan 2.19 terhadap z.

Kedua, *forward propagation* antara z_2 dan w_{ho} menggunakan Persamaan 2.20. *bias* yang digunakan adalah 1.

$$z_2 = \begin{bmatrix} 0,067 & 0,982 & 0,479 & 1 & 0,021 \\ 0,943 & 0,415 & 0,801 & 0,016 & 0,972 \\ 0,734 & 0,708 & 0,694 & 0,813 & 0,665 \\ 0,944 & 0,377 & 0,886 & 0,005 & 0,982 \\ 0,731 & 0,707 & 0,695 & 0,823 & 0,655 \end{bmatrix}$$

Fungsi aktivasi sigmoid juga diterapkan pada tahap *forward propagation* yang kedua.

$$z_3 = \begin{bmatrix} 1,391 & 2,1681 \\ 2,12 & 0,078 \\ 1,728 & 1,302 \\ 2,027 & 0,119 \\ 1,709 & 1,321 \end{bmatrix}$$

Terakhir, *forward propagation* akan dievaluasi dengan menggunakan Persamaan 2.21.

$$o = \begin{bmatrix} 0,801 & 0,897 \\ 0,893 & 0,519 \\ 0,849 & 0,786 \\ 0,884 & 0,53 \\ 0,847 & 0,789 \end{bmatrix}$$

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N (f(x,y) - f'(x,y))^2 = 0,397$$

2. Backward Propagation

Selisih antara target dan o menentukan synapses mana yang akan dirubah.

Semakin kecil selisih antara target dan o , semakin sedikit perubahan bobot synapses. Sama seperti *forward propagation*, *backward propagation* juga terbagi kedalam dua tahap perhitungan. Pertama, *error backward propagation* antara *output layer* dan *hidden layer*. Perhitungan *error backward propagation* ini dilakukan dengan menggunakan Persamaan 2.22.

$$\sigma_k = \begin{bmatrix} -0,128 & 0,009 \\ -0,085 & 0,12 \\ 0,019 & -0,132 \\ -0,091 & 0,117 \\ 0,02 & 0,131 \end{bmatrix}$$

Kedua, *error backward propagation* antara *hidden layer* dan *input layer*. Perhitungan *backward propagation* ini dilakukan dengan menggunakan Persamaan 2.23.

$$\delta_h = \begin{bmatrix} -0,002 & -0,003 & 0,023 & 0 & -0,003 \\ -0,01 & -0,041 & 0,018 & 0,004 & -0,002 \\ 0,037 & 0,018 & -0,016 & -0,031 & -0,002 \\ -0,01 & -0,042 & 0,012 & 0,001 & -0,001 \\ 0,038 & 0,018 & -0,016 & -0,03 & -0,002 \end{bmatrix}$$

Terakhir, bobot *synapses* akan diperbaiki berdasarkan tempat *synapses* tersebut berada. Perbaikan bobot *synapses* dilakukan dengan menggunakan Persamaan 2.24.

$$w_{ih} = \begin{bmatrix} -1,314 & 0,585 & -1,199 & -0,353 & -0,023 \\ 1,476 & -0,175 & -1,224 & 1,914 & -0,116 \\ 0,122 & 1,156 & -0,055 & 1,525 & 0,278 \\ -0,476 & 0,303 & 0,4 & 1,389 & -0,84 \\ -1,768 & -0,18 & 1,397 & 1,538 & -2,089 \end{bmatrix}$$

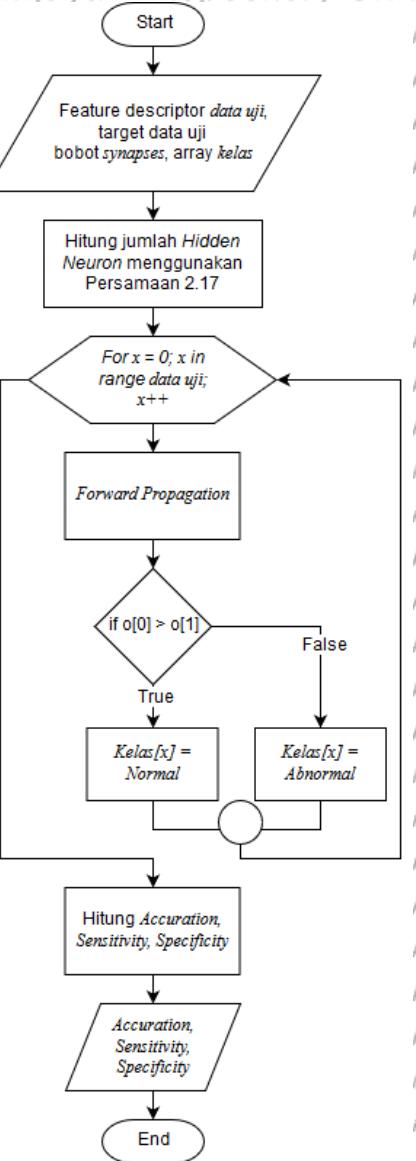
$$w_{ho} = \begin{bmatrix} 0,142 & -1,4261 \\ 1,355 & -0,453 \\ -0,693 & 0,475 \\ -0,676 & 1,45 \\ -0,966 & 0,22 \end{bmatrix}$$

Pelatihan BPNN dilakukan hingga *Epoch* maksimal tercapai atau *error* minimum tercapai.

4.1.3.2 Testing BPNN

Gambar 4.15 menampilkan *flowchart testing* BPNN. Penjelasan Gambar 4.15 adalah sebagai berikut:

1. Mengambil *feature descriptor* data uji dan target data uji.
2. Mengambil bobot *synapses* hasil *Training* BPNN.
3. Inisialisasi array *kelas*.
4. Lakukan forward propagation untuk semua data uji.
5. Bandingkan luaran di *output layer* BPNN. Jika *output neuron* ke-0 ($o[0]$) lebih besar dari *output neuron* ke-1 ($o[1]$), data ke-x termasuk kelas Normal. Jika tidak, data ke-x termasuk kelas Abnormal.
6. Simpan prediksi kelas didalam array *kelas*.
7. Hitung *Accuratuon*, *Sensitivity* dan *Specificity* menggunakan Persamaan 2.26, Persamaan 2.28 dan Persamaan 2.30 beruturut-turut.

Gambar 4.15 *Testing BPNN*

Setelah pelatihan BPNN selesai dilakukan, tahap selanjutnya adalah pengujian BPNN. Pengujian BPNN dilakukan dengan menggunakan bobot synapses terbaik yang menghasilkan *error* terkecil. Perhitungan yang terjadi pada tahap pengujian sama seperti *forward propagation*, hanya saja tidak ada perhitungan MSE. Diberikan *fd_test* sebagai berikut:

1,992	20,332	413,379	1749460,000	306128,000
1,992	32,061	1027,976	1238562,000	265014,000
1,992	13,037	169,957	2942240,000	430956,000
1,992	16,278	264,966	2470112,000	392104,000
1,992	18,869	356,052	2289952,000	365648,000
0,115	1,188	1,41	145656,000	22732,000
1,992	20,379	415,297	1492940,000	298904,000
0,560	6,191	38,333	403394,000	82222,000
0,278	2,805	7,868	277860,000	49548,000
1,992	61,746	3812,589	1943902,000	312178,000

Sama seperti saat f_d_train , f_d_test juga harus dinormalisasi terlebih dahulu.

$$f_d_{test} = \begin{bmatrix} 0,649 & 0,062 & -0,217 & 0,276 & 0,383 \\ 0,649 & 0,765 & 0,345 & -0,279 & 0,089 \\ 0,649 & -0,374 & -0,44 & 1,568 & 1,275 \\ 0,649 & -0,18 & -0,353 & 1,057 & 0,997 \\ 0,649 & -0,025 & -0,27 & 0,862 & 0,808 \\ -1,776 & -1,084 & -0,595 & -1,464 & -1,642 \\ 0,649 & 0,065 & -0,216 & -0,003 & 0,331 \\ -1,201 & -0,785 & -0,561 & -1,184 & -1,217 \\ -1,566 & -0,987 & -0,589 & -1,32 & -1,45 \\ 0,649 & 2,543 & 2,895 & 0,486 & 0,426 \end{bmatrix}$$

Menggunakan $Epoch = 1000$, didapatkan w_{ih} dan w_{ho} sebagai berikut:

$$w_{ih} = \begin{bmatrix} -1,316 & 0,577 & -1,2 & -0,352 & -0,023 \\ 1,473 & -0,184 & -1,224 & 1,916 & -0,116 \\ 0,122 & 1,15 & -0,057 & 1,525 & 0,278 \\ -0,478 & 0,295 & 0,399 & 1,39 & -0,84 \\ -1,77 & -0,188 & 1,389 & 1,539 & -2,089 \end{bmatrix}$$

$$w_{ho} = \begin{bmatrix} 0,157 & -1,429 \\ 0,893 & -1,143 \\ -1,154 & -0,225 \\ -1,145 & 0,772 \\ 0,981 & 0,214 \end{bmatrix}$$

Setelah mendapatkan w_{ih} dan w_{ho} , tahap selanjutnya adalah *forward propagation*. Hasil forward propagation terhadap f_test menggunakan w_{ih} dan w_{ho} adalah sebagai berikut:

$$f_d_{test} = \begin{bmatrix} 0,612 & 0,553 \\ 0,759 & 0,435 \\ 0,38 & 0,653 \\ 0,417 & 0,64 \\ 0,457 & 0,621 \\ 0,814 & 0,312 \\ 0,662 & 0,529 \\ 0,83 & 0,301 \\ 0,818 & 0,307 \\ 0,789 & 0,346 \end{bmatrix}$$

Tabel 4.1 Hasil BPNN

NO	PREDIKSI	TARGET	STATUS
1	Normal	Normal	T
2	Normal	Normal	T
3	Abnormal	Normal	F
4	Abnormal	Normal	F
5	Abnormal	Normal	F
6	Normal	Abnormal	F
7	Normal	Normal	T
8	Normal	Abnormal	F
9	Normal	Abnormal	F
10	Normal	Normal	T

Setelah klasifikasi selesai, tahap selanjutnya adalah menghitung *Accurcation*, *Sensitivity* dan *Specificity*. Perhitungan *Accurcation*, *Sensitivity* dan *Specificity* dilakukan dengan menggunakan data yang ditampilkan di Tabel 4.1. Persamaan *Accurcation*, *Sensitivity* dan *Specificity* terdapat pada Persamaan 2.26, Persamaan 2.28 dan Persamaan 2.30.

Berdasarkan Tabel 4.1, didapatkan

$$\text{Accurcation} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$\frac{4 + 0}{4 + 0 + 3 + 3} \times 100\% = 40\%$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100\% = \frac{4}{4 + 3} \times 100\% = 57\%$$

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100\% = \frac{0}{0 + 3} \times 100\% = 0\%$$

4.2 Implementasi

4.2.1 Preprocessing

4.2.1.1 ROI Extraction

Source Code 1. Source Code ROI Extraction

```
1 def grayImg(self, info):
2     path = []
3     for img in glob.glob(self.dir+"\\*."+self.format):
4         path.append(img)
5
6     for row in range(0, len(info)):
7         im = cv2.imread(path[row])
8         flip = cv2.flip(im, 0)
9         xlow = int(info[row][4]) - int(info[row][6])
10        xhigh = int(info[row][4]) + int(info[row][6])
11        ylow = int(info[row][5]) - int(info[row][6])
12        yhigh = int(info[row][5]) + int(info[row][6])
13        crop = flip[ylow:yhigh, xlow:xhigh]
14        flip2 = crop[::-1, ::-1]
15        cv2.imwrite("crop/training/cropped-" + info[row][0] + ".png", flip2)
```

Implementasi *ROI Extraction* ditunjukkan pada *Source Code 1*. Penjelasan *Source Code 1* adalah sebagai berikut:

1. Baris ke-3, perulangan untuk membaca semua file dengan format yang sudah ditentukan di direktori yang sudah dispesifikasikan. Dalam penelitian ini, format yang dibaca adalah .PNG.
2. Baris ke-4, menyimpan alamat dan nama file yang .PNG di direktori yang sudah dispesifikasikan.

3. Baris ke-7, membaca gambar
4. Baris ke-8, memutar gambar
5. Baris ke-9 sampai Baris ke-12, menghitung $x_low, x_high, y_low, y_high$
6. Baris ke-13, memotong gambar sesuai wilayah ROI
7. Baris ke-14, menyimpan ROI di ditektori yang sudah ditentukan

4.2.1.2 CLAHE

Source Code 2. Source Code CLAHE

```
1 clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
2 histogram_equalization = clahe.apply(im)
3 histogram_dest = "1. CLAHE\\\"+self.destination[1]+"\\"
4 cv2.imwrite(os.path.join(self.destination[0], histogram_dest,
"histEqual "+name[1]), histogram_equalization)
```

Dalam penelitian ini, implementasi CLAHE diterapkan dengan menggunakan *package* yang disediakan oleh cv2. *Source code* CLAHE ditampilkan dalam *Source code 2*.

Penjelasan *Source Code 2* adalah sebagai berikut:

1. Baris ke 1, inisialisasi CLAHE.
2. Baris ke 2, terapkan clahe pada gambar *im*. *im* adalah matriks gambar.
3. Baris ke 3, alamat folder penyimpanan gambar hasil CLAHE.
4. Baris ke 4, simpan gambar CLAHE di folder yang sudah ditentukan. cv2.imwrite memiliki 2 parameter. Parameter pertama adalah alamat lengkap folder. Parameter kedua adalah nama gambar yang disimpan.

4.2.1.3 Median Filter

Implementasi *Median Filter* pada penelitian ini ditampilkan di *Source Code 3*. Dalam penelitian ini, implementasi *Median Filter* dilakukan menggunakan *package* yang disediakan oleh cv2.

Source Code 3. Source Code Median Blurring

```
1 median_blurring = cv2.medianBlur(histogram_equalization, 3)
2 median_dest = "2. medianBlurring\\\"+self.destination[1]+"\\"
3 cv2.imwrite(os.path.join(self.destination[0], median_dest,
"medianBlurring "+name[1]), median blurring)
```

Penjelasan *Source Code 3* adalah sebagai berikut:

1. Baris ke 1, inisialisasi *median blurring*. *Median filter* mengambil masukkan matriks *histogram_equalization*. *histogram_equalization* adalah hasil CLAHE pada matriks gambar *im*.
2. Baris ke 2, alamat folder penyimpanan gambar hasil *median blurring*.
3. Baris ke 3, simpan gambar hasil *median blurring* di folder yang sudah ditentukan.

4.2.2 Feature Extraction

4.2.2.1 GLCM

Implementasi dari metode GLCM

4.2.2 Feature Extraction

4.2.2.1 GLCM

Implementasi dari metode GLCM ditampilkan dalam *Source Code 4*.

```
Source Code 4. GLCM Repository Universitas Brawijaya
1 def glcm(self):
2     for d in range(1, 5):
3         for t in range(4):
4             feature = []
5             teta = t * np.pi / 4
6             folder = "GLCM\\GLCM_" + str(d) + "\\"
7             file_name = self.data_type + "_" + str(t) + ".txt"
8             dest = self.destination + folder + file_name
9             for img in range(len(self.source)):
10                bgr =
cv2.cvtColor(cv2.imread(self.source[int(img)]),
cv2.COLOR_BGR2RGB)
11                img = cv2.cvtColor(bgr, cv2.COLOR_RGB2GRAY)
12                glcm = greycomatrix(img, [d], [teta], levels=256,
symmetric=True, normed=False)
13                feature.append(self.feature_descriptor(glcm))
14                np.savetxt(dest, feature, delimiter=", ")
15
16 def feature_descriptor(self, matrix):
17     fd = []
18     # mean
19     mean = np.mean(matrix)
20     fd.append(mean)
21     # standard deviation
22     std = np.std(matrix)
23     fd.append(std)
24     # variance
25     variance = np.var(matrix)
26     fd.append(variance)
27     # contrast
28     contrast = 0
29     for i in range(len(matrix)):
30         for j in range(len(matrix[0])):
31             contrast += matrix[int(i)][int(j)] * ((int(i) -
int(j)) * (int(i) - int(j)))
32     fd.append(contrast[0][0])
33     # dissimilarity
34     dissimilarity = 0
35     for i in range(len(matrix)):
36         for j in range(len(matrix[0])):
37             dissimilarity += matrix[int(i)][int(j)] *
np.abs(int(i) - int(j))
```

```
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
):
Repository Universitas Brawijaya
Repository Universitas Brawijaya
p.pi / 4
CM\GLCM_" + str(d) + "\\"
self.data_type + "_" + str(t) + ".txt"
destination + folder + file_name
ange(len(self.source)):
    self.source[int(img)],
cv2.cvtColor(bgr, cv2.COLOR_RGB2GRAY)
reycomatrix(img, [d], [teta], levels=256,
    False)
append(self.feature_descriptor(glcm))
est, feature, delimiter=", ")
elf, matrix):
    x)
    trix)
matrix)):
en(matrix[0])):
matrix[int(i)][int(j)] * ((int(i) - j)))
] [0])
matrix)):
en(matrix[0])):
y += matrix[int(i)][int(j)] *

```

Source Code 4. GLCM

```
38     fd.append(dissimilarity[0][0])
39     # homogeneity
40     homogeneity = 0
41     for i in range(len(matrix)):
42         for j in range(len(matrix[0])):
43             homogeneity += matrix[int(i)][int(j)] / (1 + ((int(i)
44 - int(j)) * (int(i) - int(j))))
45             fd.append(homogeneity[0][0])
46             # ASM
47             ASM = 0
48             for i in range(len(matrix)):
49                 for j in range(len(matrix[0])):
50                     ASM += matrix[int(i)][int(j)] *
51                     matrix[int(i)][int(j)]
52                     fd.append(ASM[0][0])
53                     # energy
54                     energy = np.sqrt(ASM)
55                     fd.append(energy[0][0])
56                     # max prob
57                     max_prob = np.amax(matrix)
58                     fd.append(max_prob)
59                     # sum avg
60                     sum_avg = 0
61                     for i in range(0, len(matrix)):
62                         for j in range(0, len(matrix[0])):
63                             sum_avg += matrix[i][j] * (i + j)
64                             sum_avg = sum_avg / 2
65                             fd.append(sum_avg[0][0])
66                             return fd
```

Source Code 3 terdiri dari dua method. Method pertama adalah `glcm(self)` dan method kedua adalah `feature_descriptor(self, matrix)`. Penjelasan dari method `glcm(self)` adalah sebagai berikut:

1. Baris ke 2, perulangan proses glcm sebanyak 4 kali, mulai dari $d = 1$ sampai $d = 4$
2. Baris ke 3, perulangan proses glcm sebanyak 4 kali, mulai dari $t = 0$ sampai $t = 4$
3. Baris ke 3, perulangan proses glcm sebanyak 4 kali, mulai dari $t = 0$ sampai $t = 4$
4. Baris ke 4, inisialisasi array `feature`
5. Baris ke 5, perhitungan teta
6. Baris ke 6-8, penentuan folder tempat penyimpanan fitur GLCM

7. Baris ke 10, membaca file gambar

8. Baris ke 11, pembentukan matriks `GLCM` dengan menggunakan fungsi `greycomatrix` yang disediakan oleh `skimage`

9. Baris ke 12, memanggil method `feature_descriptor(self, matrix)` dan menyimpan hasil kembalian method `feature_descriptor(self, matrix)` di array `feature`

10. Baris ke 13, menulis array `feature` di file `.txt` dan menyimpannya di folder yang sudah ditentukan

Penjelasan dari method `feature_descriptor(self, matrix)` adalah sebagai berikut:

1. Baris ke 17, inisialisasi array `fd`
2. Baris ke 19-20, perhitungan fitur *mean* dan menyimpannya di array `fd`
3. Baris ke 22-23, perhitungan fitur *standard deviation* dan menyimpannya di array `fd`
4. Baris ke 25-26, perhitungan fitur *variance* dan menyimpannya di array `fd`
5. Baris ke 28-32, perhitungan fitur *contrast* dan menyimpannya di array `fd`
6. Baris ke 34-38, perhitungan fitur *dissimilarity* dan menyimpannya di array `fd`
7. Baris ke 40-44, perhitungan fitur *homogeneity* dan menyimpannya di array `fd`
8. Baris ke 46-50, perhitungan fitur *ASM* dan menyimpannya di array `fd`
9. Baris ke 52-53, perhitungan fitur *energy* dan menyimpannya di array `fd`
10. Baris ke 55-56, perhitungan fitur *maximum probability* dan menyimpannya di array `fd`
11. Baris ke 57-63, perhitungan fitur *sum average* dan menyimpannya di array `fd`
12. Baris ke 64, mengembalikan array `fd`

4.2.3 BPNN

4.2.3.1 Pelatihan BPNN

```
Source Code 5. Pelatihan BPNN
1 def __init__(self, input_file, hidden_neuron, target_file,
2 weight_file, type):
3     #parameters
4     self.X = input_file / npamax(input_file, axis=0)
5     self.y = np.loadtxt(target_file, delimiter=",",
6 dtype=float)
7     self.input_size = len(self.X[0])
8     self.hidden_size = hidden_neuron
9     self.output_size = 2
10
11    #weights
```

Source Code 5. Pelatihan BPNN**Repository Universitas Brawijaya**

```
10      # self.W1 = np.random.randn(self.input_size,
11      # self.hidden_size)
12      # self.W2 = np.random.randn(self.hidden_size,
13      # self.output_size)
14      # np.savetxt(weight_file+"\init_w1.txt", self.W1,
15      # delimiter=',')
16      # np.savetxt(weight_file+"\init_w2.txt", self.W2,
17      # delimiter=',')
18      self.W1 = np.loadtxt('params\init_w1.txt',
19      delimiter=",", dtype=float)
20      self.W2 = np.loadtxt('params\init_w2.txt',
21      delimiter=",", dtype=float)

22  def forward(self, X):
23      self.z = np.dot(X, self.W1)
24      self.z2 = self.sigmoid(self.z)
25      self.z3 = np.dot(self.z2, self.W2)
26      o = self.sigmoid(self.z3)
27      return o

28  def sigmoid(self, s):
29      return 1/(1+np.exp(-s))

30  def sigmoidPrime(self, s):
31      return s * (1 - s)

32  def backward(self, X, target, calc_output):
33      self.o_error = target - calc_output
34      self.o_delta = self.o_error *
35      self.sigmoidPrime(calc_output)

36      self.z2_error = self.o_delta.dot(self.W2.T)
37      self.z2_delta = self.z2_error *
38      self.sigmoidPrime(self.z2)

39      self.W1 += X.T.dot(self.z2_delta)
40      self.W2 += self.z2.T.dot(self.o_delta)

41  def train (self):
42      o = self.forward(self.X)
43      self.backward(self.X, self.y, o)
44      w = [self.W1, self.W2]
45      return w

46  def err (self):
47      print("Input: \n" + str(self.X))
48      print("Actual Output: \n" + str(self.y))
```

Source Code 5. Pelatihan BPNN

```

49     print("Predicted Output: \n" +
50         str(self.forward(self.X)))
      print("Loss: \n" + str(np.mean(np.square(self.y -
self.forward(self.X)))))


```

Implementasi pelatihan BPNN dilakukan dengan menggunakan *Source Code 5*.

Penjelasan dari *Source Code 5* adalah sebagai berikut:

1. Baris ke-1, inisialisasi konstruktur.
2. Baris ke-3, normalisasi *fd_train*.
3. Baris ke-4, inisialisasi *target* pelatihan.
4. Baris ke-5 sampai Baris ke-7, penetuan jumlah *input layer*, *hidden layer* dan *output layer*.
5. Baris ke-10 dan Baris ke-11, random bobot *synapses w_ih* dan *w_ho*.
6. Baris ke-12 dan Baris ke-13, menyimpan bobot *synapses* di file yang sudah ditentukan.
7. Baris ke-14 dan Baris ke-15, membaca bobot *synapses* yang sudah disimpan sebelumnya.
8. Baris ke-17 sampai Baris ke-22, *forward propagation*
9. Baris ke-24 dan Baris ke-25, aktivasi sigmoid
10. Baris ke-27 dan Baris ke-28, turunan dari aktivasi sigmoid
11. Baris ke-30 sampai Baris ke-38, *backward propagation*
12. Baris ke-40 sampai Baris ke-44, method inti untuk proses pelatihan
13. Baris ke-46 sampai Baris ke-50, mencetak informasi yang dihasilkan dari tahap pelatihan BPNN

4.2.3.2 Pengujian BPNN**Source Code 6. Pengujian BPNN**

```

1 def __init__(self, w1, w2):
2     self.W1 = w1
3     self.W2 = w2
4
5     def forward(self, X):
6         self.z = np.dot(X, self.W1)
7         self.z2 = self.sigmoid(self.z)
8         self.z3 = np.dot(self.z2, self.W2)
9         o = self.sigmoid(self.z3)
10
11     def sigmoid(self, s):
12         return 1 / (1+np.exp(-s))
13

```

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya

Source Code 6. Pengujian BPNN

```
14 def test (self, X):
15     o = self.forward(X)
16     if o[0] > o[1]:
17         out = "Normal"
18     else:
19         out = "Abnormal"
20     return out;
```

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya

Implementasi pengujian BPNN dilakukan dengan menggunakan Source Code 6.

Penjelasan dari Source Code 6 adalah sebagai berikut:

1. Baris ke-1, inisialisasi konstruktur.
2. Baris ke-2 dan Baris ke-3, inisialisasi bobot *synapses w_{ih}* dan *w_{ho}*.
3. Baris ke-5 sampai Baris ke-10, *forward propagation*
4. Baris ke-12 dan Baris ke-13, aktivasi sigmoid
5. Baris ke-15 dan Baris ke-21, klasifikasi *fd_testing*

4.2.4 User Interface

Penambahan *user interface* pada metode yang diusulkan dilakukan untuk mempermudah pengoperasian sistem yang dibuat. *User interface* yang dibuat mengikuti perancangan *user interface* yang diterangkan di subbab 3.1.2. Sistem yang dirancang terdiri dari dua halaman utama, yaitu halaman Diagnosis dan halaman Pelatihan.

4.2.1.2 Halaman Diagnosis

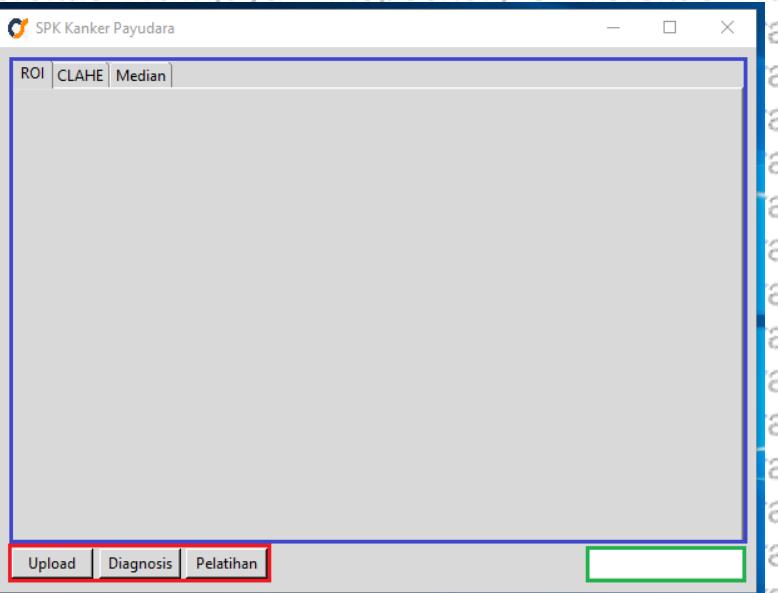
Gambar 4.16 menampilkan halaman diagnosis. Penjelasan dari Gambar 4.16 adalah sebagai berikut:

1. Button

Bagian ini ditandai oleh kota berwarna merah di Gambar 4.14. Bagian ini berisi tombol untuk mengoperasikan sistem yang dibuat. Bagian ini terbagi menjadi 3 tombol, yaitu tombol *Upload*, tombol *Diagnosis* dan tombol *Pelatihan*. Tombol *Upload* berfungsi untuk membuka *file manager*. Setelah itu, pengguna tinggal memilih ROI yang akan diprediksi. Tombol *Diagnosis* berfungsi untuk menerapkan metode yang diusulkan ke ROI yang diupload oleh pengguna. Tombol *Pelatihan* berfungsi untuk menampilkan detail data yang digunakan dalam proses pelatihan dan prediksi sistem.

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya



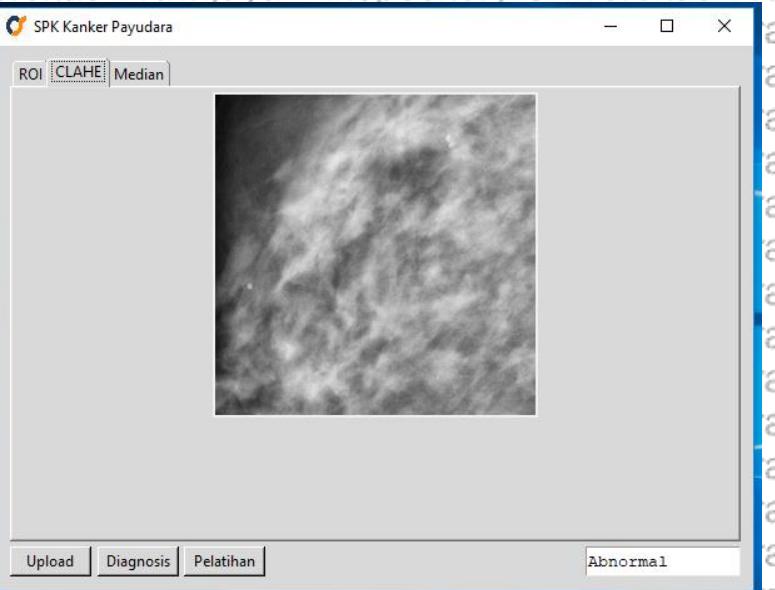
Gambar 4.16 Halaman diagnosis

2. Notebook

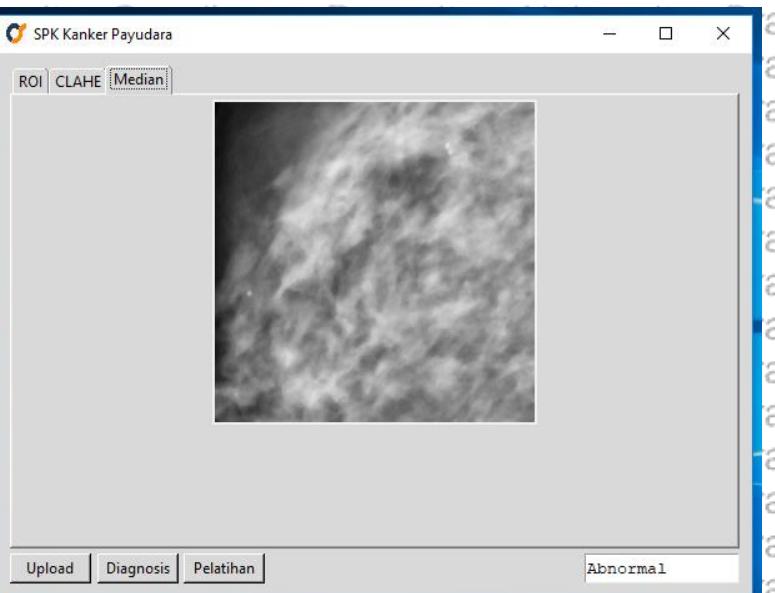
Bagian ini ditandai oleh kotak berwarna biru di Gambar 4.16. Bagian ini terbagi menjadi 3 tab, yaitu ROI, CLAHE dan Median. Tab ROI berfungsi untuk menampilkan gambar ROI yang diupload oleh pengguna (ditampilkan di Gambar 4.17). Tab CLAHE berfungsi untuk menampilkan gambar ROI setelah melewati proses CLAHE (ditampilkan di Gambar 4.18). Tab Median berfungsi untuk menampilkan gambar hasil CLAHE setelah melewati proses Median Filter (ditampilkan di Gambar 4.19). Tab CLAHE dan tab Median akan menampilkan gambar setelah pengguna menekan tombol Diagnosis.



Gambar 4.17 Tab ROI



Gambar 4.18 Tab CLAHE



Gambar 4.19 Tab Median

3. Entry

Bagian ini ditandai oleh kotak berwarna hijau di Gambar 4.16. Bagian ini berfungsi untuk menampilkan hasil prediksi sistem. *Entry* secara otomatis akan terisi ketika pengguna menekan tombol diagnosis.

4.2.1.3 Halaman Pelatihan

Halaman pelatihan berfungsi untuk menampilkan detail data yang digunakan dalam tahap pelatihan dan pengujian. Gambar 4.20 menampilkan halaman pelatihan.

1	0	1	Hitung	
w1	w2	Prediksi	Pengujian	
0	1	2	3	4
-11.77042 50917928 5	-5.715546 62325247 25	2.3165744 58952965	-6.668972 00022204	0.05391529 852616461
-0.147064 77751560 34	3.2704841 63848062			
11.615887 41178649 5	9.1540057 61015764	1.8629682 41848071 5	5.9562036 61494971	-2.746161 29940336 73
-0.454045 65309433 43	-0.772342 84852468 66			
-15.71224 18099371 38	-15.62702 50085160 87	9.1609652 19336155	-13.29486 65470658 23	-16.33288 66546683 07
-14.03590 30462945 05	13.119255 35127217 8			
-2.190205 24833286 25	-1.725853 02443224 1	4.8179259 57081753	-2.002404 78573866 86	-8.109986 19362013 6
-8.058366 93830028 8	6.5761758 26993865			
-3.332586 28072947 1	10.042881 06688209 6	-3.271098 80073644 6	-2.689933 83636088 6	7.1617684 05676528 6
7.5356430 62678778 6	-6.149091 11508860 6			

Gambar 4.20 Halaman pelatihan

Repository Universitas Brawijaya
Brawijaya University **PPID** **UINAS**

BAB 5 HASIL DAN PEMBAHASAN

Pada bab ini, dibahas mengenai penentuan parameter pengujian GLCM-BPNN, hasil pengujian GLCM-BPNN dan pembahasan terkait hasil pengujian yang didapat. Dikarenakan data yang tidak lengkap, data nomor 59, 216, 233 dan 245 dihilangkan dari dataset.

5.1 Pengujian Parameter BPNN

Performa BPNN dipengaruhi oleh jumlah *Hidden Neuron*, *Learning Rate* dan jumlah *Epoch*. Oleh karena itu, sebelum melakukan pengujian pengaruh d dan θ dalam GLCM terhadap *Accurracy*, *Sensitivity* dan *Specificity* BPNN, jumlah *Hidden Neuron*, *Learning Rate* dan jumlah *Epoch* harus ditentukan terlebih dahulu. Dalam penelitian ini, jumlah *Hidden Neuron* ditentukan menggunakan Persamaan 2.17.

Tahapan pengujian parameter BPNN ditentukan berdasarkan kemunculan parameter tersebut dalam algoritme BPNN. Tahap pertama BPNN adalah *Forward Propagation*. Pada tahap ini, parameter BPNN yang berperan adalah jumlah *Hidden Neuron*. Setelah *Forward Propagation* selesai, bobot BPNN akan diperbaiki pada tahap *Backward Propagation*. Perubahan bobot BPNN pada tahap *Backward Propagation* dipengaruhi oleh *Learning Rate*. Terakhir, *Forward Propagation* dan *Backward Propagation* akan dilakukan sejumlah *Epoch* yang ditentukan. Berdasarkan tahapan algoritme BPNN, parameter pertama yang diuji adalah jumlah *Hidden Neuron*, lalu dilanjutkan dengan pengujian *Learning Rate* dan terakhir pengujian jumlah *Epoch*.

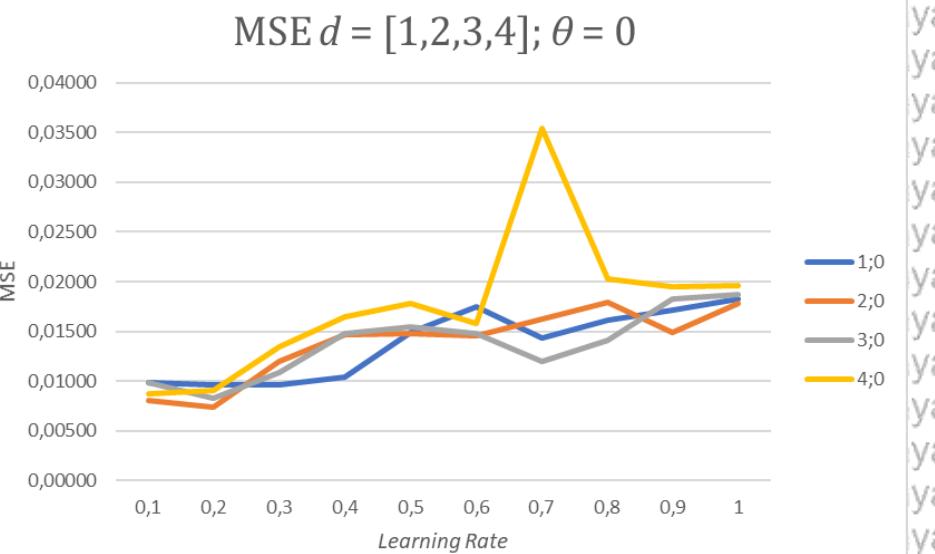
Pada setiap pengujian parameter BPNM, bobot *synapses* awal yang digunakan sama. Hal ini dilakukan untuk memberikan perlakuan yang sama untuk semua kasus uji.

5.1.1 Pengujian *Learning Rate*

Learning Rate yang diuji mulai dari 0,1 sampai 1. Parameter pengujian lain yang digunakan adalah sebagai berikut:

1. Jumlah *Hidden Neuron* yang digunakan didapat dengan menggunakan Persamaan 2.17.
$$HN = \left(\frac{2}{3}n_i\right) + n_o = \left(\frac{2}{3} \times 10\right) + 2 = 8,667 \approx 9$$
 2. Jumlah *Epoch* yang digunakan adalah 5.000.
 3. *Bias* yang digunakan adalah 1.
 4. Pengujian *Learning Rate* dilakukan untuk semua kombinasi d dan θ pada GLCM.
 5. Pengujian *Learning Rate* diukur dengan menggunakan rumus MSE yang ditampilkan di Persamaan 2.21. Nilai MSE yang diambil adalah MSE pada *Epoch* 5.000 untuk masing masing kombinasi d dan θ pada GLCM.
 6. Setiap kombinasi d dan θ akan diuji menggunakan *5-fold cross-validation*.

Hasil pengujian *Learning Rate* ditampilkan pada Gambar 5.1 sampai 5.4.



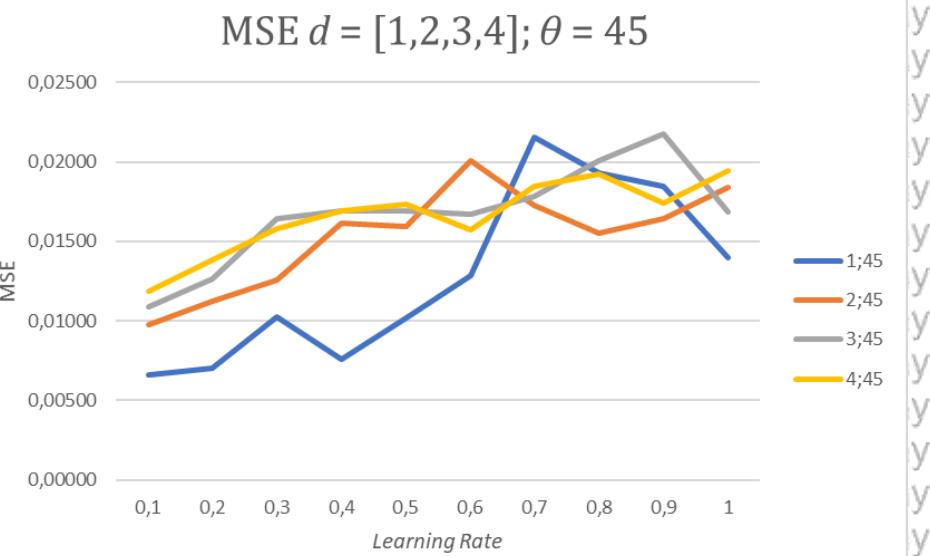
Gambar 5.1 Hasil pengujian $Learning\ Rate$ untuk $d = [1, 2, 3, 4]$ dan $\theta = 0^0$. Sumbu X adalah $Learning\ Rate$ dan sumbu Y adalah MSE.

Gambar 5.1 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing-masing kombinasi $d = [1,2,3,4]$ dan $\theta = 0^0$. Garis biru pada Gambar 5.1 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 0^0$. Garis oranye pada Gambar 5.1 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 0^0$. Garis abu-abu pada Gambar 5.1 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 0^0$. Garis kuning pada Gambar 5.1 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 0^0$.

Pada Gambar 5.1 bisa dilihat jika grafik MSE masih berubah-ubah dan belum stabil. Walaupun begitu, MSE masih ada di kisaran 0,00741 sampai 0,03540. Detail hasil pengujian $Learning\ Rate$ untuk kombinasi $d = [1,2,3,4]$ dan $\theta = 0^0$ bisa dilihat di Tabel 5.1. Cell berwarna merah adalah cell yang memiliki nilai minimal pada kolom tersebut.

Tabel 5.1 Detail hasil pengujian $Learning\ Rate$ untuk $d = [1, 2, 3, 4]$ dan $\theta = 0^0$

Learning Rate	$d = 1; \theta = 0^0$	$d = 2; \theta = 0^0$	$d = 3; \theta = 0^0$	$d = 4; \theta = 0^0$
0,1	0,00985	0,00801	0,00983	0,00877
0,2	0,00963	0,00741	0,00832	0,00905
0,3	0,00956	0,01194	0,01082	0,01342
0,4	0,01042	0,01468	0,01482	0,01649
0,5	0,01485	0,01479	0,01540	0,01784
0,6	0,01746	0,01457	0,01476	0,01579
0,7	0,01429	0,01624	0,01196	0,03540
0,8	0,01612	0,01790	0,01408	0,02027
0,9	0,01709	0,01489	0,01826	0,01944
1	0,01821	0,01780	0,01866	0,01964



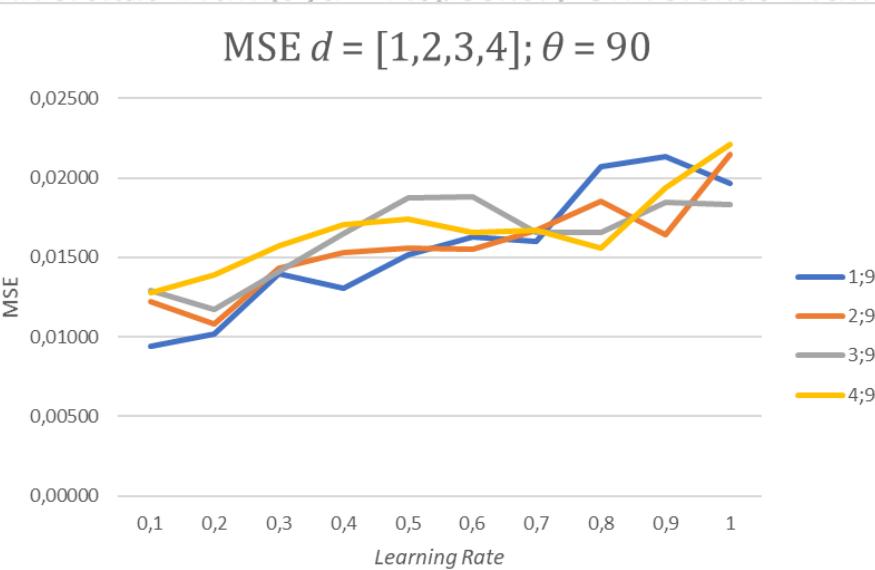
Gambar 5.2 Hasil pengujian *Learning Rate* untuk $d = [1, 2, 3, 4]$ dan $\theta = 45^\circ$. Sumbu X adalah *Learning Rate* dan sumbu Y adalah MSE.

Gambar 5.2 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing-masing kombinasi $d = [1,2,3,4]$ dan $\theta = 45^\circ$. Garis biru pada Gambar 5.2 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 45^\circ$. Garis oranye pada Gambar 5.2 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 45^\circ$. Garis abu-abu pada Gambar 5.2 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 45^\circ$. Garis kuning pada Gambar 5.2 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 45^\circ$.

Pada Gambar 5.2 bisa dilihat jika grafik MSE masih berubah-ubah dan belum stabil. Walaupun begitu, MSE masih ada di kisaran 0,00660 sampai 0,02179. Detail hasil pengujian *Learning Rate* untuk kombinasi $d = [1,2,3,4]$ dan $\theta = 45^\circ$ bisa dilihat di Tabel 5.2. Cell berwarna merah adalah cell yang memiliki nilai minimal pada kolom tersebut.

Tabel 5.2 Detail hasil pengujian *Learning Rate* untuk $d = [1, 2, 3, 4]$ dan $\theta = 45^\circ$

Learning Rate	$d = 1; \theta = 45^\circ$	$d = 2; \theta = 45^\circ$	$d = 3; \theta = 45^\circ$	$d = 4; \theta = 45^\circ$
0,1	0,00660	0,00976	0,01089	0,01188
0,2	0,00707	0,01121	0,01264	0,01387
0,3	0,01029	0,01257	0,01643	0,01577
0,4	0,00761	0,01617	0,01692	0,01689
0,5	0,01021	0,01592	0,01694	0,01733
0,6	0,01284	0,02006	0,01673	0,01571
0,7	0,02157	0,01724	0,01787	0,01850
0,8	0,01928	0,01554	0,02008	0,01925
0,9	0,01850	0,01645	0,02179	0,01738
1	0,01401	0,01840	0,01689	0,01943



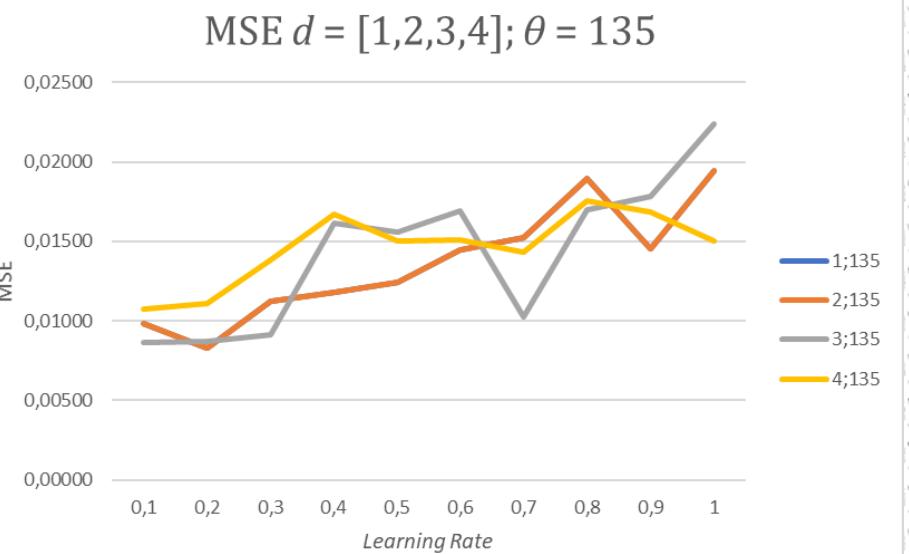
Gambar 5.3 Hasil pengujian *Learning Rate* untuk $d = [1, 2, 3, 4]$ dan $\theta = 90^\circ$. Sumbu X adalah *Learning Rate* dan sumbu Y adalah MSE.

Gambar 5.3 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing-masing kombinasi $d = [1,2,3,4]$ dan $\theta = 90^\circ$. Garis biru pada Gambar 5.3 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 90^\circ$. Garis oranye pada Gambar 5.3 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 90^\circ$. Garis abu-abu pada Gambar 5.3 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 90^\circ$. Garis kuning pada Gambar 5.3 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 90^\circ$.

Pada Gambar 5.3 bisa dilihat jika grafik MSE masih berubah-ubah dan belum stabil. Walaupun begitu, MSE masih ada di kisaran 0,00945 sampai 0,02214. Detail hasil pengujian *Learning Rate* untuk kombinasi $d = [1,2,3,4]$ dan $\theta = 90^\circ$ bisa dilihat di Tabel 5.3. Cell berwarna merah adalah cell yang memiliki nilai minimal pada kolom tersebut.

Tabel 5.3 Detail hasil pengujian *Learning Rate* untuk $d = [1, 2, 3, 4]$ dan $\theta = 90^\circ$

Learning Rate	$d = 1; \theta = 90^\circ$	$d = 2; \theta = 90^\circ$	$d = 3; \theta = 90^\circ$	$d = 4; \theta = 90^\circ$
0,1	0,00945	0,01225	0,01291	0,01276
0,2	0,01021	0,01083	0,01171	0,01393
0,3	0,01401	0,01434	0,01413	0,01575
0,4	0,01307	0,01533	0,01652	0,01705
0,5	0,01516	0,01557	0,01874	0,01740
0,6	0,01626	0,01550	0,01882	0,01657
0,7	0,01599	0,01670	0,01658	0,01673
0,8	0,02073	0,01854	0,01655	0,01562
0,9	0,02133	0,01647	0,01846	0,01936
1	0,01969	0,02146	0,01834	0,02214



Gambar 5.4 Hasil pengujian Learning Rate untuk $d = [1, 2, 3, 4]$ dan $\theta = 135^\circ$.
Sumbu X adalah Learning Rate dan sumbu Y adalah MSE.

Gambar 5.4 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing-masing kombinasi $d = [1,2,3,4]$ dan $\theta = 135^\circ$. Garis biru pada Gambar 5.4 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 135^\circ$. Garis oranye pada Gambar 5.4 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 135^\circ$. Garis abu-abu pada Gambar 5.4 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 135^\circ$. Garis kuning pada Gambar 5.4 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 135^\circ$.

Pada Gambar 5.4 bisa dilihat jika grafik MSE masih berubah-ubah dan belum stabil. Walaupun begitu, MSE masih ada di kisaran 0,0832 sampai 0,02240. Detail hasil pengujian Learning Rate untuk kombinasi $d = [1,2,3,4]$ dan $\theta = 135^\circ$ bisa dilihat di Tabel 5.4. Cell berwarna merah adalah cell yang memiliki nilai minimal pada kolom tersebut.

Tabel 5.4 Detail hasil pengujian Learning Rate untuk $d = [1, 2, 3, 4]$ dan $\theta = 135^\circ$

Learning Rate	$d = 1; \theta = 135^\circ$	$d = 2; \theta = 135^\circ$	$d = 3; \theta = 135^\circ$	$d = 4; \theta = 135^\circ$
0,1	0,00982	0,00982	0,00867	0,01077
0,2	0,00832	0,00832	0,00874	0,01112
0,3	0,01127	0,01127	0,00913	0,01382
0,4	0,01180	0,01180	0,01613	0,01671
0,5	0,01240	0,01240	0,01557	0,01504
0,6	0,01450	0,01450	0,01694	0,01508
0,7	0,01524	0,01524	0,01023	0,01430
0,8	0,01895	0,01895	0,01700	0,01758
0,9	0,01455	0,01455	0,01781	0,01684
1	0,01944	0,01944	0,02240	0,01501

Demi mempermudah pengamatan, Tabel 5.1 sampai Tabel 5.4 digabung menjadi satu dan ditampilkan pada Tabel 5.5. Sel berwarna merah pada Tabel 5.5 menandakan sel tersebut memiliki nilai MSE terkecil di kolom tempat sel tersebut berada. Dikarenakan keterbatasan tempat, nilai MSE di masing-masing sel dibulatkan menjadi 3 angka dibelakang koma. Berdasarkan hasil pengujian, *Learning Rate* yang menghasilkan nilai MSE terkecil paling banyak adalah 0,1. Ketika *Learning Rate* adalah 0,1, 9 kombinasi d dan θ ($d = [2,3]$ dan $\theta = 0^\circ; d = 2$ dan $\theta = 45^\circ; d = [2,4]$ dan $\theta = 90^\circ; d = [1,2,34]$ dan $\theta = 135^\circ$) mencapai nilai MSE terkecil. Oleh karena itu, *Learning Rate* yang digunakan dalam penelitian ini adalah 0,1.

Tabel 5.5 Detail hasil pengujian *Learning Rate*

<i>Learning Rate</i>	d	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	θ	0°	0°	0°	0°	45°	45°	45°	45°	90°	90°	90°	90°	135°	135°	135°	135°
0,1	0,010	0,007	0,009	0,010	0,008	0,010	0,012	0,010	0,010	0,011	0,013	0,009	0,009	0,012	0,013	0,011	0,011
0,2	0,010	0,007	0,010	0,008	0,007	0,011	0,011	0,008	0,008	0,013	0,012	0,009	0,009	0,014	0,014	0,011	0,011
0,3	0,010	0,010	0,014	0,011	0,012	0,013	0,014	0,011	0,011	0,016	0,014	0,009	0,013	0,016	0,016	0,014	0,014
0,4	0,010	0,008	0,013	0,012	0,015	0,016	0,015	0,012	0,015	0,017	0,017	0,016	0,016	0,017	0,017	0,017	0,017
0,5	0,015	0,010	0,015	0,012	0,015	0,016	0,016	0,012	0,015	0,017	0,019	0,016	0,018	0,017	0,017	0,015	0,015
0,6	0,017	0,013	0,016	0,015	0,015	0,020	0,015	0,015	0,015	0,017	0,019	0,017	0,016	0,016	0,017	0,017	0,015
0,7	0,014	0,022	0,016	0,015	0,016	0,017	0,017	0,015	0,015	0,012	0,018	0,017	0,010	0,035	0,018	0,017	0,014
0,8	0,016	0,019	0,021	0,019	0,018	0,016	0,019	0,019	0,014	0,020	0,017	0,017	0,020	0,019	0,016	0,018	0,018
0,9	0,017	0,019	0,021	0,015	0,015	0,016	0,016	0,015	0,018	0,022	0,018	0,018	0,019	0,017	0,019	0,017	0,017
1	0,018	0,014	0,020	0,019	0,018	0,018	0,021	0,019	0,019	0,017	0,018	0,022	0,020	0,019	0,022	0,015	0,015

5.1.2 Pengujian Jumlah Epoch

Jumlah *Epoch* yang diuji mulai dari 5.000 sampai 50.000. Parameter pengujian lain yang digunakan adalah sebagai berikut:

1. Jumlah *Hidden Neuron* yang digunakan didapat dengan menggunakan persamaan 2.17.

$$HN = \left(\frac{2}{3} n_i \right) + n_o = \left(\frac{2}{3} \times 10 \right) + 2 = 8,667 \approx 9$$

2. *Learning Rate* yang digunakan adalah 0,1.

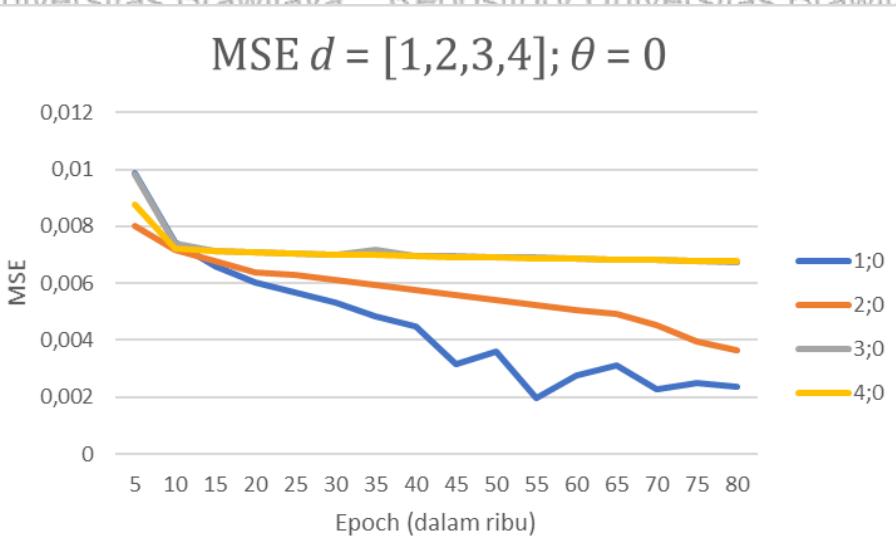
3. *Bias* yang digunakan adalah 1.

4. Pengujian *Epoch* dilakukan untuk semua kombinasi d dan θ pada GLCM.

5. Pengujian *Epoch* diukur dengan menggunakan rumus MSE yang ditampilkan di Persamaan 2.21.

6. Setiap kombinasi d dan θ akan diuji menggunakan *5-fold cross-validation*.

Hasil pengujian *Epoch* ditampilkan pada Gambar 5.5 sampai 5.8.



Gambar 5.5 Hasil pengujian jumlah *Epoch* untuk $d = [1, 2, 3, 4]$ dan $\theta = 0^\circ$. Sumbu X adalah jumlah *Epoch* (dalam ribu) dan sumbu Y adalah MSE

Gambar 5.5 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing-masing kombinasi $d = [1,2,3,4]$ dan $\theta = 0^\circ$. Garis biru pada Gambar 5.5 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 0^\circ$. Garis oranye pada Gambar 5.5 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 0^\circ$. Garis abu-abu pada Gambar 5.5 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 0^\circ$. Garis kuning pada Gambar 5.5 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 0^\circ$.

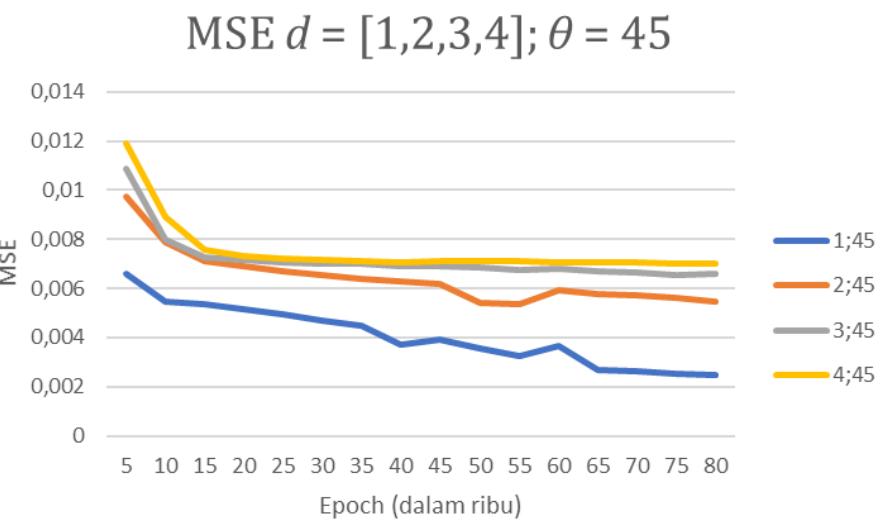
Pada Gambar 5.5 bisa dilihat jika grafik MSE lebih stabil. Penjelasan dari Gambar 5.5 adalah sebagai berikut:

1. Ketika $d = 1$ dan $\theta = 0^\circ$, nilai MSE lebih stabil dan memiliki tren menurun. Nilai MSE terus mengalami penurunan sampai Epoch ke-45.000 (0,003175). Setelah itu di Epoch ke-50.000, nilai MSE naik menjadi 0,0036603.
2. Ketika $d = 2$ dan $\theta = 0^\circ$, nilai MSE terus mengalami penurunan hingga mencapai nilai minimum di Epoch ke-80.000 (0,003629).
3. Ketika $d = 3$ dan $\theta = 0^\circ$, nilai MSE terus mengalami penurunan hingga Epoch ke-30.000 (0,006985). Setelah itu, di Epoch ke-35.000, nilai MSE naik menjadi 0,00714.
4. Ketika $d = 4$ dan $\theta = 0^\circ$, nilai MSE terus mengalami penurunan hingga Epoch ke-40.000 (0,006782).

Detail hasil pengujian Epoch untuk kombinasi $d = [1,2,3,4]$ dan $\theta = 0^\circ$ bisa dilihat di Tabel 5.6. Sel berwarna merah adalah titik terakhir nilai MSE mengalami penuruan sebelum nilainya naik di Epoch selanjutnya.

Tabel 5.6 Detail hasil pengujian jumlah Epoch untuk $d = [1, 2, 3, 4]$ dan $\theta = 0^\circ$

Epoch (dalam ribu)	$d = 1 ; \theta = 0^\circ$	$d = 2 ; \theta = 0^\circ$	$d = 3 ; \theta = 0^\circ$	$d = 4 ; \theta = 0^\circ$
5	0,009849	0,008014	0,009835	0,008765
10	0,007456	0,007179	0,007383	0,007231
15	0,006585	0,006780	0,007128	0,007126
20	0,006048	0,006383	0,007091	0,007084
25	0,005686	0,006307	0,007055	0,007048
30	0,005346	0,006112	0,006985	0,007017
35	0,004855	0,005947	0,007164	0,006987
40	0,004472	0,005768	0,006959	0,006958
45	0,003175	0,005593	0,006938	0,006929
50	0,003603	0,005420	0,006915	0,006901
55	0,001989	0,005248	0,006893	0,006877
60	0,002769	0,005080	0,006869	0,006855
65	0,003137	0,004920	0,006843	0,006835
70	0,002292	0,004541	0,006815	0,006817
75	0,002478	0,003977	0,006784	0,006799
80	0,002356	0,003629	0,006751	0,006782



Gambar 5.6 Hasil pengujian jumlah *Epoch* untuk $d = [1,2,3,4]$ dan $\theta = 45^\circ$.

Sumbu X adalah jumlah *Epoch* (dalam ribu) dan sumbu Y adalah MSE

Gambar 5.6 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing-masing kombinasi $d = [1,2,3,4]$ dan $\theta = 0^\circ$. Garis biru pada Gambar 5.6 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 0^\circ$. Garis oranye pada Gambar 5.6 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 0^\circ$. Garis abu-abu pada Gambar 5.6 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 0^\circ$. Garis kuning pada Gambar 5.6 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 0^\circ$.

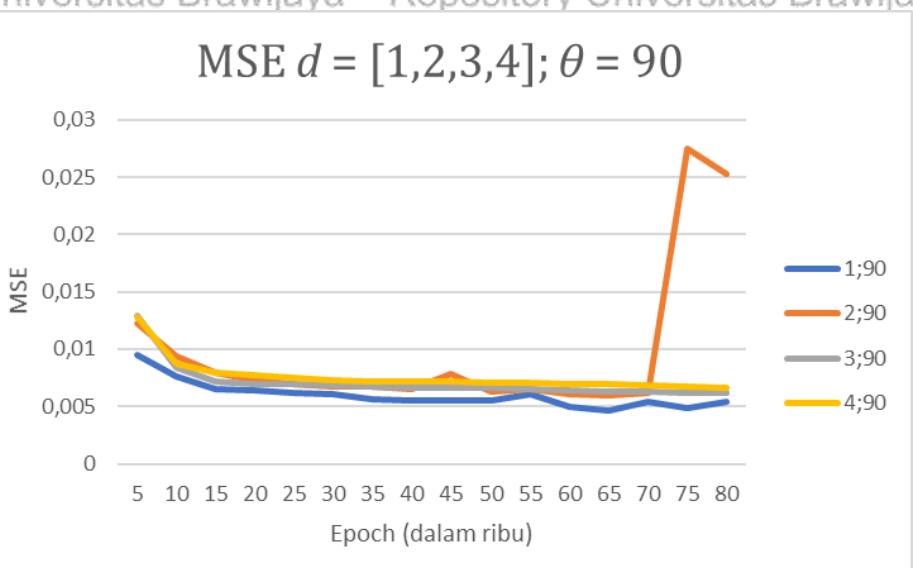
Pada Gambar 5.6 bisa dilihat jika grafik MSE lebih stabil. Penjelasan dari Gambar 5.6 adalah sebagai berikut:

1. Ketika $d = 1$ dan $\theta = 45^\circ$, nilai MSE memiliki tren menurun hingga *Epoch* ke 40.000 (0,003690). Setelah itu, nilai MSE naik menjadi 0,003929 di *Epoch* ke 45.000.
2. Ketika $d = 2$ dan $\theta = 45^\circ$, nilai MSE terus mengalami penurunan hingga *Epoch* ke 55.000 (0,005342). Setelah itu, di *Epoch* ke 60.000, nilai MSE naik menjadi 0,005902.
3. Ketika $d = 3$ dan $\theta = 45^\circ$, nilai MSE terus mengalami penurunan hingga *Epoch* ke 50.000 (0,006750). Setelah itu, nilai MSE naik menjadi 0,006814 di *Epoch* ke 60.000.
4. Ketika $d = 4$ dan $\theta = 45^\circ$, nilai MSE terus mengalami penurunan hingga *Epoch* ke 40.000 (0,007068). Setelah itu, nilai MSE naik menjadi 0,007118 di *Epoch* ke 45.000.

Detail hasil pengujian *Epoch* untuk kombinasi $d = [1,2,3,4]$ dan $\theta = 45^\circ$ bisa dilihat di Tabel 5.7. Sel berwarna merah adalah titik terakhir nilai MSE mengalami penurunan sebelumnya naik di *Epoch* selanjutnya.

Tabel 5.7 Detail hasil pengujian jumlah Epoch untuk $d = [1,2,3,4]$ dan $\theta = 45^\circ$

Epoch (dalam ribu)	$d = 1; \theta = 45^\circ$	$d = 2; \theta = 45^\circ$	$d = 3; \theta = 45^\circ$	$d = 4; \theta = 45^\circ$
5	0,006604	0,009758	0,010894	0,011884
10	0,005467	0,0079	0,007983	0,008936
15	0,005372	0,007114	0,007273	0,007592
20	0,005147	0,006905	0,007138	0,007333
25	0,004953	0,006702	0,007069	0,007221
30	0,004689	0,006547	0,007015	0,007148
35	0,004482	0,006414	0,006989	0,007105
40	0,003690	0,006295	0,006909	0,007068
45	0,003929	0,006186	0,006879	0,007118
50	0,003536	0,005425	0,006837	0,007101
55	0,003236	0,005342	0,006750	0,007084
60	0,003643	0,005902	0,006814	0,007069
65	0,002697	0,005784	0,006709	0,007055
70	0,002610	0,005704	0,006642	0,007044
75	0,002540	0,00562	0,006549	0,007031
80	0,002469	0,005463	0,006591	0,00702



Gambar 5.7 Hasil pengujian jumlah Epoch untuk $d = [1,2,3,4]$ dan $\theta = 90^\circ$. Sumbu X adalah jumlah Epoch (dalam ribu) dan sumbu Y adalah MSE

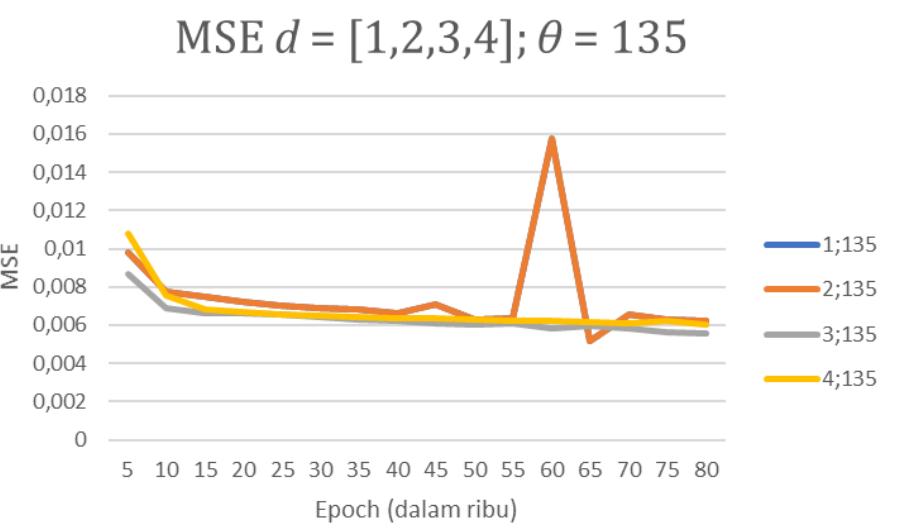
Gambar 5.7 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing masing kombinasi $d = [1,2,3,4]$ dan $\theta = 90^\circ$. Garis biru pada Gambar 5.7 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 90^\circ$. Garis oranye pada Gambar 5.7 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 90^\circ$. Garis abu-abu pada Gambar 5.7 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 90^\circ$. Garis ungu/kuning Gambar 5.7 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 90^\circ$.

Penjelasan dari Gambar 5.7 adalah sebagai berikut:

1. Ketika $d = 1$ dan $\theta = 90^\circ$, nilai MSE terus mengalami penurunan hingga Epoch ke-50.000 (0,005469). Setelah itu, nilai MSE naik menjadi 0,006051. Setelah Epoch-50.000, nilai MSE menjadi tidak stabil. Nilai MSE mencapai nilai minimum di Epoch ke-65.000 (0,004631).
 2. Ketika $d = 2$ dan $\theta = 90^\circ$, nilai MSE terus mengalami penurunan hingga Epoch ke-40.000 (0,006486). Setelah itu, nilai MSE naik menjadi 0,007809. Lonjakan nilai MSE terjadi antara Epoch ke-70.000 (0,006134) dan Epoch ke-75.000 (0,027466).
 3. Ketika $d = 3$ dan $\theta = 90^\circ$, nilai MSE terus mengalami penurunan hingga mencapai nilai minimum di Epoch ke-80.000 (0,006157).
 4. Ketika $d = 4$ dan $\theta = 90^\circ$, nilai MSE terus mengalami penurunan sampai Epoch ke-35.000 (0,007164). Setelah itu, nilai MSE naik menjadi 0,007179.

Detail hasil pengujian *Epoch* untuk kombinasi $d = [1,2,3,4]$ dan $\theta = 90^0$ bisa dilihat di Tabel 5.8. Sel berwarna merah adalah titik terakhir nilai MSE mengalami penuruan sebelumnya naik di *Epoch* selanjutnya.

Tabel 5.8 Detail hasil pengujian jumlah Epoch untuk $d = [1, 2, 3, 4]$ dan $\theta = 90^\circ$



Gambar 5.8 Hasil pengujian jumlah Epoch untuk $d = [1,2,3,4]$ dan $\theta = 135^0$.

Sumbu X adalah jumlah Epoch (dalam ribu) dan sumbu Y adalah MSE.

Gambar 5.8 adalah rata-rata MSE $k = [1,2,3,4,5]$ untuk masing-masing kombinasi $d = [1,2,3,4]$ dan $\theta = 135^0$. Garis biru pada Gambar 5.8 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 1$ dan $\theta = 135^0$. Garis oranye pada Gambar 5.8 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 2$ dan $\theta = 135^0$. Garis abu-abu pada Gambar 5.8 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 3$ dan $\theta = 135^0$. Garis kuning pada Gambar 5.8 mewakili rata-rata MSE untuk $k = [1,2,3,4,5]$ pada kombinasi $d = 4$ dan $\theta = 135^0$.

Pada Gambar 5.8 bisa dilihat jika grafik MSE stabil. Penjelasan dari Gambar 5.8 adalah sebagai berikut:

1. Ketika $d = 1$ dan $\theta = 135^0$, nilai MSE terus mengalami penurunan hingga mencapai titik konvergen ketika Epoch = 40.000 (0,006598). Setelah Epoch ke-40.000, nilai MSE berubah-ubah dan mencapai nilai minimum di Epoch ke-65.000 (0,005172). Lonjakan nilai MSE terjadi di antara Epoch ke-55.000 (0,006362) dan Epoch ke-60.000 (0,015758).
2. Ketika $d = 2$ dan $\theta = 135^0$, nilai MSE terus mengalami penurunan hingga mencapai titik konvergen ketika Epoch = 40.000 (0,006598). Setelah Epoch ke-40.000, nilai MSE berubah-ubah dan mencapai nilai minimum di Epoch ke-65.000 (0,005172). Lonjakan nilai MSE terjadi di antara Epoch ke-55.000 (0,006362) dan Epoch ke-60.000 (0,015758).
3. Ketika $d = 3$ dan $\theta = 135^0$, nilai MSE terus mengalami penurunan hingga Epoch ke-15.000 (0,006624). Setelah itu, nilai MSE naik menjadi 0,006628.
4. Ketika $d = 4$ dan $\theta = 135^0$, nilai MSE terus mengalami penurunan hingga mencapai nilai minimum di Epoch ke-80.000 (0,006044).

Tabel 5.9 Detail hasil pengujian jumlah Epoch untuk $d = [1,2,3,4]$ dan $\theta = 135^\circ$

<i>Epoch</i> (dalam ribu)	$d = 1; \theta = 135^\circ$	$d = 2; \theta = 135^\circ$	$d = 3; \theta = 135^\circ$	$d = 4; \theta = 135^\circ$
5	0,009822	0,009822	0,008674	0,010769
10	0,007723	0,007723	0,006892	0,007581
15	0,007486	0,007486	0,006624	0,006846
20	0,007223	0,007223	0,006628	0,006695
25	0,007027	0,007027	0,006538	0,006566
30	0,006882	0,006882	0,006405	0,00652
35	0,006852	0,006852	0,006301	0,006452
40	0,006598	0,006598	0,006211	0,006392
45	0,007088	0,007088	0,006123	0,006344
50	0,006297	0,006297	0,005996	0,006287
55	0,006362	0,006362	0,006063	0,006253
60	0,015758	0,015758	0,00582	0,006208
65	0,005172	0,005172	0,005932	0,006167
70	0,006538	0,006538	0,005862	0,006127
75	0,006283	0,006283	0,005619	0,006195
80	0,006208	0,006208	0,005584	0,006044

Demi mempermudah pengamatan, Tabel 5.6 sampai Tabel 5.9 digabung menjadi menjadi satu dan ditampilkan pada Tabel 5.10. Pada Tabel 5.10, data dibulatkan menjadi 3 angka dibelakang koma. Tabel 5.11 menampilkan rata-rata waktu komputasi pengujian *Epoch* untuk $k = [1,2,3,4,5]$. Sel berwarna merah pada Tabel 5.10 dan Tabel 5.11 menandakan titik terakhir nilai MSE mengalami penurunan sebelum nilainya naik di *Epoch* selanjutnya. Dalam pengujian *Epoch*, nilai MSE terkecil diambil ketika nilai MSE turun dan naik di *Epoch* selanjutnya. Hal ini dilakukan untuk mempersingkat waktu komputasi. Sebagai contoh untuk $d = 1$ dan $\theta = 0^\circ$, nilai MSE di *Epoch* ke-45.000 adalah 0,003175. Nilai tersebut naik menjadi 0,003603 di *Epoch* ke-50.000. Di *Epoch* ke-70.000, nilai MSE mencapai nilai minimum untuk $d = 1$ dan $\theta = 0^\circ$, yaitu sebesar 0,002292. Waktu komputasi dari *Epoch* ke-45.000 adalah 4,470 detik dan *Epoch* ke-70.000 adalah 6,990 detik. Itu artinya, dalam waktu 2,52 detik, nilai MSE hanya turun 0,000883.

Berdasarkan hasil pengujian, jumlah *Epoch* yang menjadi titik terakhir nilai MSE mengalami penurunan sebelum naik di *Epoch* selanjutnya terbanyak adalah 40.000. Ketika *Epoch* adalah 40.000 ada 5 kombinasi d dan θ ($d = [1,4]$ dan $\theta = 45^\circ; d = 2$ dan $\theta = 90^\circ; d = [1,2]$ dan $\theta = 135^\circ;$) yang mencapai titik terakhir nilai MSE mengalami penurunan sebelum naik di *Epoch* selanjutnya. Oleh karena itu, *Epoch* yang digunakan dalam penelitian ini adalah 40.000.



Epoch	$d=1$	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	0	0 ⁰	0 ⁰	0 ⁰	45 ⁰	45 ⁰	45 ⁰	45 ⁰	90 ⁰	90 ⁰	90 ⁰	90 ⁰	135 ⁰	135 ⁰	135 ⁰	135 ⁰
5	0,010	0,008	0,010	0,009	0,007	0,010	0,011	0,012	0,009	0,012	0,013	0,013	0,010	0,010	0,009	0,011
10	0,007	0,007	0,007	0,007	0,005	0,008	0,008	0,009	0,008	0,009	0,008	0,009	0,008	0,008	0,007	0,008
15	0,007	0,007	0,007	0,007	0,005	0,007	0,007	0,008	0,007	0,008	0,007	0,008	0,007	0,007	0,007	0,007
20	0,006	0,006	0,007	0,007	0,005	0,007	0,007	0,007	0,006	0,007	0,007	0,008	0,007	0,007	0,007	0,007
25	0,006	0,006	0,007	0,007	0,005	0,007	0,007	0,007	0,006	0,007	0,007	0,008	0,007	0,007	0,007	0,007
30	0,005	0,006	0,007	0,007	0,005	0,007	0,007	0,007	0,006	0,007	0,007	0,007	0,007	0,007	0,006	0,007
35	0,005	0,006	0,007	0,007	0,004	0,006	0,007	0,007	0,006	0,007	0,007	0,007	0,007	0,007	0,006	0,006
40	0,004	0,006	0,007	0,007	0,004	0,006	0,007	0,007	0,006	0,006	0,007	0,007	0,007	0,007	0,006	0,006
45	0,003	0,006	0,007	0,007	0,004	0,006	0,007	0,007	0,006	0,008	0,007	0,007	0,007	0,007	0,006	0,006
50	0,004	0,005	0,007	0,007	0,004	0,005	0,007	0,007	0,005	0,006	0,007	0,007	0,006	0,006	0,006	0,006
55	0,002	0,005	0,007	0,007	0,003	0,005	0,007	0,007	0,006	0,007	0,007	0,007	0,006	0,006	0,006	0,006
60	0,003	0,005	0,007	0,007	0,004	0,006	0,007	0,007	0,005	0,006	0,006	0,007	0,016	0,016	0,006	0,006
65	0,003	0,005	0,007	0,007	0,003	0,006	0,007	0,007	0,005	0,006	0,006	0,007	0,005	0,005	0,006	0,006
70	0,002	0,005	0,007	0,007	0,003	0,006	0,007	0,007	0,005	0,006	0,006	0,007	0,007	0,007	0,006	0,006
75	0,002	0,004	0,007	0,007	0,003	0,006	0,007	0,007	0,005	0,027	0,006	0,007	0,006	0,006	0,006	0,006
80	0,002	0,004	0,007	0,007	0,002	0,005	0,007	0,007	0,005	0,025	0,006	0,007	0,006	0,006	0,006	0,006

Tabel 5.10 Detail hasil pengujian jumlah Epoch

Tabel 5.11 Detail waktu komputasi pengujian jumlah Epoch

Epoch	$d = 1$	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	0°	0°	0°	0°	45°	45°	45°	45°	90°	90°	90°	90°	135°	135°	135°	135°
5	0,503	0,509	0,497	0,506	0,525	0,512	0,522	0,528	0,506	0,512	0,500	0,508	0,503	0,497	0,509	0,503
10	0,997	0,994	1,000	1,010	1,006	0,994	1,019	1,014	1,003	0,989	0,997	0,996	0,993	1,007	0,990	1,001
15	1,513	1,534	1,501	1,510	1,513	1,506	1,518	1,503	1,497	1,500	1,500	1,490	1,487	1,484	1,497	1,488
20	1,998	1,989	2,015	2,003	1,998	2,003	2,047	2,008	1,991	1,994	1,989	1,999	1,988	1,995	1,990	1,987
25	2,482	2,498	2,505	2,521	2,509	2,482	2,571	2,498	2,501	2,499	2,494	2,486	2,489	2,491	2,488	2,478
30	3,003	2,997	2,995	3,023	3,009	2,995	3,050	3,021	2,986	2,985	2,997	3,009	2,980	3,010	2,974	2,985
35	3,502	3,513	3,533	3,532	3,506	3,504	3,539	3,527	3,487	3,487	3,477	3,492	3,497	3,484	3,485	3,482
40	3,984	3,998	4,021	4,027	4,007	4,009	4,128	4,103	4,010	3,991	4,011	3,990	3,984	3,972	3,982	3,985
45	4,470	4,496	4,517	4,535	4,502	4,523	4,626	4,553	4,498	4,510	4,476	4,500	4,482	4,478	4,473	4,476
50	4,997	5,024	5,029	5,029	4,986	5,047	5,132	4,986	4,978	4,988	4,992	4,962	4,963	4,979	4,971	5,000
55	5,475	5,499	5,552	5,541	5,499	5,530	5,690	5,480	5,470	5,489	5,483	5,472	5,474	5,495	5,462	5,468
60	5,978	5,993	6,063	6,032	5,991	6,019	6,111	6,019	5,981	6,027	5,995	5,954	5,994	5,980	5,959	5,997
65	6,484	6,524	6,541	6,551	6,487	6,526	6,524	6,538	6,512	6,460	6,487	6,467	6,480	6,461	6,454	6,497
70	6,990	7,030	7,053	7,027	7,023	7,057	7,037	7,140	7,000	6,973	6,955	6,972	6,966	6,976	6,985	6,991
75	7,472	7,533	7,604	7,555	7,504	7,678	7,554	7,527	7,476	7,445	7,479	7,463	7,465	7,485	7,464	7,493
80	8,004	7,995	8,064	8,022	7,989	8,163	8,178	8,000	8,018	7,971	7,974	8,053	7,947	7,988	7,970	8,010

Repository Universitas Brawijaya
Repository Universitas Brawijaya

5.2 Pengujian Kombinasi d dan θ

Setelah parameter BPNN ditentukan, tahap selanjutnya adalah menguji performa metode yang diusulkan. Pengujian diukur menggunakan 3 parameter pengujian, yaitu *Accurration*, *Sensitivity* dan *Specificity*. Pengujian kombinasi d dan θ dilakukan untuk semua kombinasi d dan θ dalam GLCM. Parameter pengujian lain yang digunakan adalah sebagai berikut:

1. Jumlah *Hidden Neuron* yang digunakan didapat dengan menggunakan Persamaan 2.17.

$$HN = \left(\frac{2}{3} n_i \right) + n_o = \left(\frac{2}{3} \times 8 \right) + 2 = 8,667 \approx 9$$

2. *Learning Rate* yang digunakan adalah 0,1.
3. Jumlah *Epoch* yang digunakan adalah 40.000.
4. Setiap kombinasi d dan θ akan diuji menggunakan *5-fold cross validation*.

Detail hasil pengujian *Accurration* ditampilkan dalam Tabel 5.12. Kolom $k=1$, $k=2$, $k=5$ menampilkan *Accurration* untuk masing masing kombinasi data latih dan data uji. *Accurration* dihitung menggunakan Persamaan 2.26. Kolom *Global Accurration* menampilkan rata-rata *Accurration* dari *5-fold cross validation*. *Global Accurration* dihitung menggunakan Persamaan 2.27. Sel berwarna biru di Tabel 5.12 menunjukkan nilai *Global Accurration* terbesar. Sel berwarna merah di Tabel 5.12 menunjukkan nilai *Global Accurration* terkecil. *Global Accurration* terbesar didapat ketika $d = [1,2]$ dan $\theta = 135^\circ$ (96,60%). Sedangkan *Global Accurration* terkecil didapat ketika $d = 4$ dan $\theta = 45^\circ$ (94,39%).

Tabel 5.12 Detail hasil pengujian *Accurration*

Parameter GLCM	d	θ	<i>Accurration</i>				<i>Global Accurration</i>	
			$k = 1$	$k = 2$	$k = 3$	$k = 4$		
1	0°	0°	89,23%	92,31%	92,06%	100,00%	100,00%	94,72%
2	0°	0°	93,85%	92,31%	93,65%	100,00%	100,00%	95,96%
3	0°	0°	92,31%	92,31%	93,65%	100,00%	100,00%	95,65%
4	0°	0°	92,31%	92,31%	92,06%	100,00%	100,00%	95,34%
1	45°	45°	93,85%	93,85%	93,65%	95,24%	100,00%	95,32%
2	45°	45°	93,85%	92,31%	93,65%	100,00%	100,00%	95,96%
3	45°	45°	92,31%	92,31%	93,65%	100,00%	100,00%	95,65%
4	45°	45°	92,31%	92,31%	95,24%	100,00%	100,00%	95,97%
1	90°	90°	93,85%	92,31%	93,65%	100,00%	100,00%	95,96%
2	90°	90°	93,85%	90,77%	92,06%	100,00%	100,00%	95,34%
3	90°	90°	93,85%	90,77%	92,06%	100,00%	100,00%	95,34%
4	90°	90°	93,85%	90,77%	92,06%	100,00%	100,00%	95,34%
1	135°	135°	93,85%	92,31%	96,83%	100,00%	100,00%	96,60%
2	135°	135°	93,85%	92,31%	96,83%	100,00%	100,00%	96,60%
3	135°	135°	92,31%	92,31%	92,06%	100,00%	100,00%	95,34%
4	135°	135°	92,31%	90,77%	90,48%	100,00%	98,41%	94,39%

Detail hasil pengujian *Sensitivity* ditampilkan dalam Tabel 5.13. Kolom $k = 1, k = 2, \dots, k = 5$ menampilkan *Sensitivity* untuk masing masing kombinasi data latih dan data uji. *Sensitivity* dihitung menggunakan Persamaan 2.28. Kolom *Global Sensitivity* menampilkan rata-rata *Sensitivity* dari *5-fold cross validation*. Global *Sensitivity* dihitung menggunakan Persamaan 2.29. Sel berwarna biru di Tabel 5.13 menunjukkan nilai *Global Sensitivity* terbesar. Sel berwarna merah di Tabel 5.13 menunjukkan nilai *Global Sensitivity* terkecil. *Global Sensitivity* terbesar didapat ketika $d = [1,2]$ dan $\theta = 135^\circ$ (97,72%). Sedangkan *Global Sensitivity* terkecil didapat ketika $d = 4$ dan $\theta = 135^\circ$ (93,96%).

Tabel 5.13 Detail hasil pengujian *Sensitivity*

Parameter		<i>Sensitivity</i>					<i>Global Sensitivity</i>
GLCM		$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	
d	θ						
1	0°	90,91%	97,67%	92,86%	100,00%	100,00%	96,29%
2	0°	90,91%	97,67%	95,24%	100,00%	100,00%	96,76%
3	0°	86,36%	97,67%	95,24%	100,00%	100,00%	95,86%
4	0°	86,36%	97,67%	92,86%	100,00%	100,00%	95,38%
	45°	90,91%	100,00%	95,24%	92,86%	100,00%	95,80%
1	45°	95,45%	97,67%	95,24%	100,00%	100,00%	97,67%
2	45°	86,36%	97,67%	95,24%	100,00%	100,00%	95,86%
3	45°	86,36%	97,67%	97,62%	100,00%	100,00%	96,33%
	90°	90,91%	97,67%	95,24%	100,00%	100,00%	96,76%
1	90°	90,91%	95,35%	92,86%	100,00%	100,00%	95,82%
2	90°	90,91%	95,35%	92,86%	100,00%	100,00%	95,82%
3	90°	90,91%	95,35%	92,86%	100,00%	100,00%	95,82%
4	90°	90,91%	95,35%	92,86%	100,00%	100,00%	95,82%
	135°	90,91%	97,67%	100,00%	100,00%	100,00%	97,72%
1	135°	90,91%	97,67%	100,00%	100,00%	100,00%	97,72%
2	135°	86,36%	97,67%	92,86%	100,00%	100,00%	95,38%
3	135°	86,36%	97,67%	90,48%	100,00%	97,62%	93,96%

Detail hasil pengujian *Specificity* ditampilkan dalam Tabel 5.14. Kolom $k = 1, k = 2, \dots, k = 5$ menampilkan *Specificity* untuk masing masing kombinasi data latih dan data uji. *Specificity* dihitung menggunakan Persamaan 2.30. Kolom *Global Specificity* menampilkan rata-rata *Specificity* dari *5-fold cross validation*. Global *Specificity* dihitung menggunakan Persamaan 2.31. Sel berwarna biru di Tabel 5.14 menunjukkan nilai *Global Specificity* terbesar. Sel berwarna merah di Tabel 5.14 menunjukkan nilai *Global Specificity* terkecil. *Global Specificity* terbesar adalah 93,53%. Ada 14 kombinasi d dan θ yang memiliki *Global Specificity* 93,53%, yaitu kombinasi $d = [2,3,4]$ dan $\theta = 0^\circ$, $d = [1,3,4]$ dan $\theta = 45^\circ$, $d = [1,2,3,4]$ dan $\theta = 90^\circ$, $d = [1,2,3,4]$ dan $\theta = 135^\circ$. Sedangkan *Global Specificity* terkecil didapat ketika $d = 3$ dan $\theta = 135^\circ$ (92,13%).

Tabel 5.14 Detail hasil pengujian Specificity

Parameter GLCM		Specificity					Global Specificity
d	θ	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	
1	0°	88,37%	81,82%	90,48%	100,00%	100,00%	92,13%
2	0°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
3	0°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
4	0°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
1	45°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
2	45°	93,02%	81,82%	90,48%	100,00%	100,00%	93,06%
3	45°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
4	45°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
1	90°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
2	90°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
3	90°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
4	90°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
1	135°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
2	135°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
3	135°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%
4	135°	95,35%	81,82%	90,48%	100,00%	100,00%	93,53%

Berdasarkan hasil pengujian *Accururation*, *Sensitivity* dan *Specificity* yang ditampilkan dalam Tabel 5.12, Tabel 5.13 dan Tabel 5.14 secara berturut-turut, kombinasi d dan θ yang memberikan hasil terbaik adalah $d = [1,2]$ dan $\theta = 45^\circ$.

5.3 Perbandingan Metode yang Diusulkan dengan Algoritme Pembanding

Selain menguji parameter yang diperlukan, metode yang diusulkan juga dibandingkan dengan algoritme pembanding. Hasil perbandingan ditampilkan dalam Tabel 5.15. Perbandingan dilakukan menggunakan *Accururation*, *Sensitivity* dan *Specificity*. *Accururation*, *Sensitivity* dan *Specificity* yang digunakan adalah *Accururation*, *Sensitivity* dan *Specificity* yang didapat menggunakan $d = [1,2]$ dan $\theta = 45^\circ$.

Tabel 5.15 menampilkan detail perbandingan metode yang diusulkan dengan metode pembanding. *Accururation* terbaik yang didapat metode yang diusulkan adalah 96,60%. *Accururation* ini lebih baik dibanding (Kral & Lenc, 2016) yang memiliki *Accururation* sebesar 84%, (Pratiwi, Alexander, Harefa, & Nanda, 2015) yang memiliki *Accururation* sebesar 93,98% dan (Setiawan, Elysia, Wesley, & Purnama, 2015) yang memiliki *Accururation* sebesar 93,90%. Sedangkan untuk *Sensitivity*, metode yang diusulkan memiliki *Sensitivity* sebesar 97,72%. *Sensitivity* metode yang diusulkan lebih kecil jika dibandingkan dengan penelitian yang dilakukan oleh (Setiawan, Elysia, Wesley, & Purnama, 2015) yang memiliki *Sensitivity* sebesar 100%, tapi lebih baik dibandingkan *Sensitivity* sebesar 97,22% yang dimiliki (Pratiwi, Alexander, Harefa, & Nanda, 2015). Terakhir, *Specificity* yang

Tabel 5.15 Perbandingan metode yang diusulkan dengan metode pembanding

Referensi	Teknik	Performa (normal-abnormal)
(Kral & Lenc, 2016; Kral & Lenc, 2016)	LBP, SVM	<i>Accururation = 84%</i>
(Pratiwi, Alexander, Harefa, & Nanda, 2015)	GLCM, RBFNN	<i>Accururation = 93,98%, Sensitivity = 97,22%, Specificity = 91,49%</i>
(Setiawan, Elysia, Wesley, Purnama, 2015)	& LAWS, ANN	<i>Accururation = 93,90%, Sensitivity = 100%, Specificity = 91%</i>
Metode yang diusulkan	GLCM-BPNN	<i>Accururation = 96,60%, Sensitivity = 97,72%, Specificity = 93,53%</i>

6.1 Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, didapatkan kesimpulan sebagai berikut:

1. Berdasarkan pengujian yang dilakukan, *Learning Rate* yang memberikan hasil terbaik adalah 0,1 dan jumlah *Epoch* yang memberikan hasil terbaik adalah 40.000. Sedangkan untuk jumlah Hidden Neuron adalah 9, didapatkan melalui Persamaan 2.17.
2. Berdasarkan pengujian yang dilakukan, kombinasi d dan θ yang menghasilkan *Accuratuon*, *Sensitivity* dan *Specificity* terbaik adalah $d = [1,2]$ dan $\theta = 135^\circ$.
3. Menggunakan $d = [1,2]$ dan $\theta = 135^\circ$ didapatkan *Accuratuon* sebesar 96,60%, *Sensitivity* sebesar 97,72% dan *Specificity* sebesar 93,53%. Berdasarkan hasil *Accuratuon* dan *Specificity*, metode yang diusulkan memiliki performa yang lebih baik dibanding metode pembanding. Akan tetapi, berdasarkan *Sensitivity*, metode yang diusulkan memiliki performa sedikit lebih buruk jika dibandingkan dengan LAWS dan ANN.

6.2 Saran

Berdasarkan penelitian yang sudah dilakukan, didapatkan saran sebagai berikut:

1. Pada penelitian ini belum dipelajari pengaruh perubahan d dan θ terhadap performa metode yang diusulkan.
2. Pada penelitian ini, setiap gambar mammogram memiliki 160 *feature descriptor*. Pada tahap pengujian, setiap feature descriptor hanya dipasangkan dengan feature descriptor lain dari kombinasi d dan θ yang sama. Pada penelitian selanjutnya, akan diterapkan metode optimasi seperti Algoritma Genetika untuk menentukan kombinasi fitur yang terbaik.
3. Pada penelitian ini, ROI didapat dari MIAS dataset. Untuk penelitian berikutnya, akan difokuskan pada metode segmentasi untuk memisahkan ROI dari gambar mammogram.
4. Pada penelitian selanjutnya, kelas Abnormal akan dispesifikasi menjadi tumor jinak dan tumor ganas.

BAB 6 PENUTUP



DAFTAR RUJUKAN

- ACS. (2018). *American Cancer Society / Information and Resources about for Cancer: Breast, Colon, Lung, Prostate, Skin*. Dipetik August 22, 2018, dari Cancer.org: <https://www.cancer.org/>
- Ball, J., & Bruce, L. (2007). Digital Mammographic Computer Aided Diagnosis (CAD) using Adaptive Level Set Segmentation. *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.
- Berment, H., Becette, V., Mohallem, M., Ferreira, F., & Chérel, P. (2014). Masses in mammography: What are the underlying anatomopathological lesions? *Diagnostic and Interventional Imaging*, 126-135.
- Boyat, A. K., & Joshi, B. K. (2015). A REVIEW PAPER. NOISE MODELS IN DIGITAL IMAGE PROCESSING. *Signal & Image Processing : An International Journal (SIPIJ)* Vol.6, No.2, 63-75.
- Brinkmann, R. (1999). *The Art and Science of Digital Compositing*. Burlington: Morgan Kaufmann.
- Chen, W., Rosner, B., Hankinson, S., Colditz, G., & Willett, W. (2011). Moderate Alcohol Consumption During Adult Life, Drinking Patterns, and Breast Cancer Risk. *JAMA*, vol. 306, no. 17, 1884-1890.
- Colditz, G., & Bohlke, K. (2014). Priorities for the primary prevention of breast cancer. *CA: A Cancer Journal for Clinicians*, vol. 64, no. 3, 186-194.
- Dang, O. (2018, September 20). *Struktur Anatomi Payudara*. Diambil kembali dari Dangstar: <https://dangstars.blogspot.com/2012/10/struktur-anatomi-payudara.html>
- doxygen.org. (2015, December 18). *OpenCV ; Histogram – 2: Histogram Equalization*. Diambil kembali dari OpenCV: https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html
- Elmore, J. G., Jackson, S. L., Abraham, L., Miglioretti, D. L., Carney, P. A., Geller, B. M., . . . al, e. (2009). Variability in interpretive performance at screening mammography and radiologists' characteristics associated with accuracy. *Radiology*, 641-651.
- Fabian, P., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Blondel, M. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825-2830.
- Gco.iarc.fr. (2018). *Cancer today*. Dipetik December 12, 2018, dari http://gco.iarc.fr/today/online-analysis-pie?v=2018&mode=cancer&mode_population=continents&population=900&populations=900&key=total&sex=2&cancer=39&type=1&statistic=5&

- Repository Universitas Brawijaya
prevalence=0&population_group=0&ages_group%5B%5D=0&ages_group%5B%5D=17&nb_items=7&group_
- GRIBKOV, I. V., KOLTSOV, P. P., KOTOVICH, N. V., KRAVCHENKO, A. A., KUTSAEV, A. S., & ZAKHAROV, A. V. (t.thn.). *Performance Evaluation of Texture Segmentation Methods*. Moscow.
- Groove, C., & Marchant, P. (2016). *Understanding Breast Cancer: A guide for people with cancer, their families and friends*. Sydney: Cancer Council Australia.
- Halls, S. (2019, Maret 29). *Common Mammogram Findings - Moose and Doc*. Diambil kembali dari Moose and Doc Breast Cancer: <https://breast-cancer.ca/abnorm-mams/>
- Hamajima, N., Hirose, K., & Tajima, K. (2002). Alcohol, tobacco and breast cancer – collaborative reanalysis of individual data from 53 epidemiological studies, including 58 515 women with breast cancer and 95 067 women without the disease. *British Journal of Cancer*, vol. 87, no. 11, 1234-1245.
- Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural Features For Image Classification. 3(6).
- Heaton, J. (2008). *Introduction to Neural Network with Java, Second Edition*. St. Louis: Heaton Research, Inc.
- Humeau-Heurtier, A. (2019). Texture Feature Extraction Methods: A Survey. *IEEE Access*, 8975-9000.
- Institute, N. C. (2019, 12 5). *Definition of mammogram - NCI Dictionary of Cancer Terms - National Cancer Institute*. Diambil kembali dari National Cancer Institute: <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/mammogram>
- Kral, P., & Lenc, P. (2016). LBP features for breast cancer detection. *IEEE International Conference on Image Processing (ICIP)*.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling, 1st ed.* 2013. New York: Springer.
- Materka, A., & Strzelecky, M. (1998). *Texture Analysis Method - A Review*. Brussels: COST B11 Report.
- Nezhadian, F., & Rashidi, S. (2017). Breast cancer detection without removal of pectoral muscle by extraction turn counts feature. *2017 Artificial Intelligence and Signal Processing Conference (AISP)*.
- Pratiwi, M., Alexander, Harefa, J., & Nanda, S. (2015). Mammograms Classification Using Gray-level Co-occurrence Matrix and Radial Basis Function Neural Network. *Procesia Computer Science*, (hal. 83-91).
- Renehan, A., Tyson, M., Egger, M., Heller, R., & Zwahlen, M. (2008). Body-mass index and incidence of cancer: a systematic review and meta-analysis of prospective observational studies. *The Lancet*, vol. 371, no. 9612, 569-578.

- Repository Universitas Brawijaya
Reza, A. M. (2004). Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement. *Journal of VLSI Signal Processing* 38, 35–44.
- Rouhi, R., Jafari, M., Kasaei, S., & Keshavarzian, P. (2015). Benign and malignant breast tumors classification based on region growing and CNN segmentation. *Expert Systems with Applications* 42, 990-1002.
- Setiawan, A., Elysia, Wesley, J., & Purnama, Y. (2015). Mammogram Classification using Law's Texture Energy Measure and Neural Networks. *Procedia Computer Science*, (hal. 92-97).
- Shirvaikar, M., Huang, N., & Dong, X. N. (2016). THE MEASUREMENT OF BONE QUALITY USING GRAY LEVEL CO-OCCURRENCE MATRIX TEXTURAL FEATURES. *Journal of Medical Imaging and Health Informatics*, 1357–1362.
- Siegel, R. L., Miller, K., & Jemal, A. (2018). Cancer statistics, 2018. *CA: A Cancer Journal for Clinicians*, 7-30.
- Sissons, C. (2018, Juni 7). *Mammogram images: Normal, abnormal, and breast cancer*. Diambil kembali dari Medical News Today: <https://www.medicalnewstoday.com/articles/322068.php>
- Smith-Warner, S., Spiegelman, D., & Yaun, S. e. (1998). Alcohol and breast cancer in women: a pooled analysis of cohort studies. *Southern Medical Journal*, vol. 91, no. 7, 535-540.
- Tabár, L., & Dean, P. (2003). Mammography and breast cancer: the new era. *International Journal of Gynecology & Obstetrics*, 319-326.
- Verma, B., McLeod, P., & Klevansky, A. (2010). Classification of benign and malignant patterns in digital mammograms for the diagnosis of breast cancer. *Expert Systems with Applications* 37, 3344–3351.
- Zihong, W., & Xiaohong, X. (2011). Study on Histogram Equalization. *2011 2nd International Symposium on Intelligence Information Processing and Trusted Computing*, 2. Hubei.
- Zulpe, N., & Pawar, V. (2012). GLCM Textural Features for Brain Tumor Classification. *IJCSI International Journal of Computer Science*, 354-359.