SISTEM KLASIFIKASI KANKER KULIT BERDASARKAN DATA CITRA *DERMOSCOPIC* DENGAN MENGGUNAKAN METODE *DEEP EXTREME LEARNING MACHINE*

SKRIPSI

UIN SUNAN AMPEL
S U R A B A Y A

Disusun Oleh

AHMAD ZOEBAD FOEADY

H72216048

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA

2019

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini,

Nama	:	Ahmad Zoebad Foeady
NIM	:	H72216048
Program Studi	:	Matematika
Angkatan	:	2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul " Sistem Klasifikasi Kanker Kulit Berdasarkan Data Citra *Dermoscopic* dengan Menggunakan Metode *Deep Extreme Learning Machine* ". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, Desember 2019

Yang menyatakan,

METERAI TEMPEL

6000

Ahmad Zoebad Foeady

NIM. H72216048

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh

Nama	:	Ahmad Zoebad Foeady
NIM	:	H72216048
Judul Skripsi	:	Sistem Klasifikasi Kanker Kulit Berdasarkan Data Citra *Dermoscopic* dengan Menggunakan Metode *Deep Extreme Learning Machine*

telah diperiksa dan disetujui untuk diujikan.

Surabaya, Desember 2019

Pembimbing

Dian C. Rini Novitasari, M.Kom
NIP. 198511242014032001

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi oleh

Nama	:	Ahmad Zoebad Foeady
NIM	:	H72216048
Judul Skripsi	:	Sistem Klasifikasi Kanker Kulit Berdasarkan Data Citra *Dermoscopic* dengan Menggunakan Metode *Deep Extreme Learning Machine*

Telah dipertahankan di depan Tim Penguji
pada tanggal Desember 2019

Mengesahkan,
Tim Penguji

Penguji I

Dian C. Rini Novitasari, M.Kom
NIP. 198511242014032001

Penguji II

Aris Fanani, M.Kom
NIP. 198701272014031002

Penguji III

Wika Dianita Utami, M.Sc
NIP. 199206102018012003

Penguji IV

Nurissaidah Ulinnuha, M.Kom
NIP. 199901102014032004

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Sunan Ampel Surabaya

Dr. Eni Purwati, M.Ag
NIP. 196512221990022001

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN
Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : AHMAD ZOEBAD FOEADY
NIM : H72216048
Fakultas/Jurusan : SAINTEK / MATEMATIKA
E-mail address : DEDDY ZOEBAD 711 @ GMAIL.COM

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :
☑ Skripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)
yang berjudul :

SISTEM KLASIFIKASI KANKER KULIT BERDASARKAN DATA CITRA DERMOSCOPIC DENGAN MENGGUNAKAN METODE DEEP EXTREME LEARNING MACHINE

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara ***fulltext*** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya,

Penulis

(AHMAD ZOEBAD FOEADY)
nama terang dan tanda tangan

ABSTRAK

Sistem Klasifikasi Kanker Kulit Berdasarkan Data Citra *Dermoscopic* dengan Menggunakan Metode *Deep Extreme Learning Machine*

Kapasitas ozon yang semakin menipis menyebabkan sinar *Ultra Violet*(UV) yang menuju bumi semakin besar dan dapat menyebabkan penyakit kanker kulit. World Health Organization(WHO) menyebutkan bahwa penderita kanker kulit terus meningkat dari tahun ke tahun pada tiap dekadenya. Terdapat sekitar 2-3 juta penderita kanker kulit non melanoma dan 132.000 pasien penderita penyakit kanker kulit melanoma. Oleh sebab itu perlu dilakukan deteksi dini kanker kulit untuk mengurangi jumlah penderita kanker kulit. Deteksi dini kanker kulit bisa dilakukan oleh dokter dengan melihat secara langsung berdasarkan fitur ABCDE, namun cara ini kurang efisien karena penglihatan seseorang bisa saja berbeda sehingga dibuatlah *Computer Aided Diagnosys* (CAD) untuk membantu pihak medis dalam mendeteksi kanker kulit. Proses yang dilakukan yaitu *pre-processing* dengan menggunakan *dull razor filtering*, *noise removal* dengan menggunakan *median filtering*, dan *contrast enhancemet* dengan menggunakan *histogram equalization*. Selanjutnya dilakukan proses klasifikasi menggunakan *Deep Extreme Learning Machine* (DELM) untuk mengklasifikasikan kanker kulit dan non kanker. Penelitian dilakukan pada 1000 data kanker kulit *Malignant* dan data non kanker atau *Benign* berjumlah 1000 data. Selanjutnya, data tersebut akan dipecah dengan pembagian data menggunakan *K-Fold cross validatian* pada k=5 sehingga terbagi menjadi 80% data *training* dan 20% data *testing*. Hasil Terbaik yang didapatkan oleh model DELM dengan 1000 hidden layer, 300 hidden nodes, dan *polynomial kernel* adalah berupa nilai akurasi sebesar 91.5%, sensitifitasnya sebesar 92%, dan nilai spesifisitasnya sebesar 91% dengan waktu training 27 menit 23 detik.
Kata kunci: Skin Cancer, HOG, DELM

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR ISI

HALAMAN JUDUL . . . . . **i**
LEMBAR PERSETUJUAN PEMBIMBING . . . . . **ii**
PENGESAHAN TIM PENGUJI SKRIPSI . . . . . **iii**
HALAMAN PERNYATAAN KEASLIAN . . . . . **iv**
MOTTO . . . . . **v**
HALAMAN PERSEMBAHAN . . . . . **vi**
KATA PENGANTAR . . . . . **vii**
DAFTAR ISI . . . . . **ix**
DAFTAR TABEL . . . . . **xii**
DAFTAR GAMBAR . . . . . **xiii**
DAFTAR LAMBANG . . . . . **xvi**
ABSTRAK . . . . . **xvii**
ABSTRACT . . . . . **xviii**
I PENDAHULUAN **1**
1.1. Latar Belakang Masalah 1
1.2. Rumusan Masalah 9
1.3. Tujuan Penelitian 9
1.4. Manfaat Penelitian 10
1.5. Batasan Masalah 10
1.6. Sistematika Penulisan 11
II TINJAUAN PUSTAKA **13**
2.1. Kanker 13
2.2. Kanker Kulit 14
2.3. Citra Digital 17
2.4. *Dermoscopic Image* 19
2.5. *Dull Razor Filtering* 20
2.6. *Grayscale Image* 22

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.7. *Median Filter* 23
2.8. *Histogram Equalization* 24
2.9. *Histogram of Oriented Gradient (HOG) Descriptor* 25
2.10. *Extreme Learning Machine*(ELM) 28
2.10.1. *Feedforward Algorithm* 29
2.10.2. *Single Hidden Layer Feedforward Neural Network*(SLFNs) 31
2.10.3. Algoritma ELM 34
2.10.4. Struktur Jaringan Syaraf Tiruan ELM 34
2.11. *Modified ELM* 35
2.11.1. *Kernel Extreme Learning Machine*(KELM) 35
2.11.2. *Extreme Learning Machine Auto Encoder*(ELM-AE) 38
2.11.3. *Multilayer Extreme Learning Machine*(MLELM) 43
2.12. *Deep Extreme Learning Machine* (DELM) 46
2.13. *Confusion Matrix* 49
III METODE PENELITIAN 52
3.1. Jenis Penelitian 52
3.2. Jenis Dan Sumber Data 52
3.3. Rancangan Penelitian 53
IV HASIL DAN PEMBAHASAN 57
4.1. *Pre-processing* Citra 57
4.1.1. *Dull Razor Filtering* 59
4.1.2. *Grayscale Image* 72
4.1.3. *Median Filtering* 75
4.1.4. *Histogram Equalization* 77
4.2. *HOG Descriptor* 82
4.3. Deep Extreme Learning Machine 93
4.3.1. Mendapatkan *Learned Parameter* 94
4.3.2. *K-Fold Cross Validation* 100
4.3.3. Proses *Training* 102
4.3.4. Proses *Testing* 105

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.4. Evaluasi dan Validasi Model 106
4.4.1. Hasil Evaluasi $n_i > n_h$ 108
4.4.2. Hasil Evaluasi $n_i = n_h$ 110
4.4.3. Hasil Evaluasi $n_i < n_h$ 111
4.5. Tampilan Aplikasi 113
4.5.1. *Loading Screen* 113
4.5.2. Layar Utama 114
V PENUTUP 117
5.1. Simpulan 117
5.2. Saran 118
DAFTAR PUSTAKA 119

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR TABEL

2.1 *Confusion Matrix* 50

4.1 Hasil Akhir *HOG Descriptor* Suatu Citra 92

4.2 Hasil Akhir *HOG Descriptor* Semua Citra Citra 93

4.3 Sampel Data *Training* 101

4.4 Sampel Data *Testing* 101

4.5 Sampel Hasil *Testing* 106

4.6 *Confusion Matrix* pada Kernel linear dengan 10 nodes 107

4.7 Sampel Hasil *Testing* dengan nodes 10 108

4.8 Sampel Hasil *Testing* dengan nodes 50 109

4.9 Sampel Hasil *Testing* dengan nodes 81 110

4.10 Sampel Hasil *Testing* dengan nodes 100 111

4.11 Sampel Hasil *Testing* dengan nodes 300 112

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR GAMBAR

2.1 Kanker Kulit 14
2.2 Basal Cell Carcinoma 15
2.3 Squamous Cell Carcinoma 16
2.4 Melanoma 17
2.5 Citra Digital RGB, Gray Scale, dan Biner 19
2.6 Hasil Pengambian Gambar Pada Dermoscopic Camera 20
2.7 Dull Razor Filtering 21
2.8 Konversi RGB ke Grayscale 22
2.9 Median Filtering 24
2.10 Histogram Equalization 25
2.11 Pengambilan Matriks Gradien 27
2.12 HOG Descriptor 28
2.13 Arsitektur *Feedforward* 30
2.14 Arsitektur ELM 35
2.15 Arsitektur KELM 38
2.16 Ilustrasi dari Struktur ELM-AE 39
2.17 Ilustrasi dari Struktur MLELM 44
2.18 Fungsi Aktifasi 46
2.19 Arsitektur DELM 47

3.1 Sampel Data Kanker(Malignant) 53
3.2 Sampel Data Non-Kanker(Benign) 53
3.3 Diagram Alur Deteksi Kanker Kulit 54
3.4 Arsitektur DELM pada Penelitian 56

4.1 (a) Sampel Data *Malignant* (b) Sampel Data *Benign* 58
4.2 Pengambilan Matriks untuk proses *Dull Razor Filtering* 59
4.3 (a) Red Layer (b) Green Layer (c) Blue Layer 60

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.4 (a) Original Image (b) Hasil Dilasi 180° 61
4.5 Sampel Hasil Akhir Dilasi 180° 62
4.6 (a) Original Image (b) Hasil Dilasi 90° 63
4.7 Sampel Hasil Akhir Dilasi 90° 64
4.8 (a) Original image (b) Hasil Erosi 180° 65
4.9 Sampel Hasil Akhir Erosi 180° 65
4.10 (a) Original Image (b) Hasil Erosi 90° 66
4.11 Sampel Hasil Akhir Erosi 90° 66
4.12 Sampel Pengambilan Data untuk *Adaptive Median Filter* 67
4.13 (a) Pengambilan Windowing 3 x 3 Pada t=1 (b) Pengurutan Nilai dan Pencarian Nilai Median, Minimum, dan Maksimum 68
4.14 Hasil Perhitungan *Adaptive Median Filter* dengan t=1 68
4.15 (a) Pengambilan Windowing 3 x 3 pada t=2 (b) Pengurutan Nilai dan Pencarian Nilai Median, Minimum, dan Maksimum 69
4.16 Hasil Perhitungan *Adaptive Median Filter* dengan t=2 69
4.17 (a) Pengambilan Windowing 3 x 3 pada t=2 (b) Pengurutan Nilai dan Pencarian Nilai Median, Minimum, dan Maksimum 70
4.18 Hasil Penambahan *Windowing* Menjadi 5 x 5 71
4.19 Hasil Akhir Perhitungan *Adaptive Median Filter* dengan Data Sampel 71
4.20 (a) Citra Awal (b) Hasil Akhir Proses *Dull Razor Filtering* 72
4.21 (a) Red Layer (b) Green Layer (c) Blue Layer 73
4.22 Simulasi Perhitungan Konversi RGB ke Grayscale 74
4.23 (a) Citra RGB (b) Citra *Grayscale* 74
4.24 (a) Windowing 3 x 3 (b) Nilai Median (c) Hasil Median Filter pada t=1 75
4.25 (a) Windowing 3 x 3 (b) Nilai Median (c) Hasil Median Filter pada t=2 75
4.26 (a) Windowing 3 x 3 (b) Nilai Median (c) Hasil Median Filter pada t=3 76
4.27 (a) Citra Awal (b) Citra Hasil *Median Filter* 77
4.28 Perhitungan Jumlah Tingkat Keabuan yang Ada Pada Suatu Citra . . 78

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.29 Perhitungan Probabilitas pada *Histogram Equalization* 79
4.30 Perhitungan Probabilitas Kumulatif pada *Histogram Equalization* . 80
4.31 Perhitungan Nilai Piksel Baru pada *Histogram Equalization* 81
4.32 (a) Citra Awal (b) Hasil *Histogram Equalization* 82
4.33 (a) Simulasi Pembagian *Block* (b) Pengambilan Sampel *Block* (c) Nilai Piksel pada *Block* Pertama 83
4.34 Penambahan Zero Padding 84
4.35 *Masking* terhadap sumbu x pada t=1 85
4.36 *Masking* terhadap sumbu x pada t=2 85
4.37 Hasil *Masking* terhadap sumbu x (I_x) 86
4.38 *Masking* terhadap sumbu y pada t=1 86
4.39 *Masking* terhadap sumbu y pada t=2 86
4.40 Hasil *Masking* terhadap sumbu y (I_y) 87
4.41 *Gradient Magnitude* dan *Gradient Orientation* pada *Block* Pertama . 89
4.42 Memasukkan Nilai Bin pada t=1 90
4.43 Memasukkan Nilai Bin pada t=2 90
4.44 Hasil Pemasukan Bin pada Block Pertama 91
4.45 Hasil Akhir *HOG Descriptor* pada *Block* Pertama 92
4.46 *Loading Screen* Aplikasi 113
4.47 Layar Utama Aplikasi 114
4.48 *Input Image* 115
4.49 Hasil Pre-processing pada Aplikasi 115
4.50 Hasil Akhir pada Aplikasi 116

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR LAMBANG

$x \in A$	: x anggota A
$\mathbb{R}$	: Himpunan semua bilangan real
$\sum_{i=0}^{x} a_i$	: Penjumlahan semua elemen a
$\|a\|$	: Norm dari a
e	: epsilon
$\forall f(x)$	: untuk setiap f(x)
$\bigtriangledown_{ELM-AE}$	: Nilai gradien dari ELM-AE

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Semakin berkembangnya zaman, teknologi serta inovasi manusia juga akan semakin berkembang mengikuti tren yang ada. Kemajuan teknologi merupakan sesuatu yang tidak bisa dihindari dari kehidupan ini karena kemajuan teknologi akan terus bertambah sesuai dengan meluasnya ilmu pengetahuan manusia. Inovasi terus dibentuk untuk menunjang kemudahan bagi manusia serta memberikan manfaat yang lebih terhadap kehidupan (Setiawan , 2018). Namun, tidak semua perkembangan teknologi akan membawa kebaikan bagi manusia dan lingkungannya. Kadang kala perkembangan zaman akan memperburuk alam dan akan berdampak buruk pada manusia itu sendiri. Beberapa perkembangan teknologi yang memiliki dampak buruk adalah efek rumah kaca, kendaraan bermotor, perkembangan dalam bidang industri, perkembangan dalam bidang sistem jaringan internet dan lain-lain. Sebagai contoh, perkembangan dalam bidang jaringan internet memang sangat memberikan manfaat dan kemudahan bagi umat manusia tapi kebaikan itu menyimpan beberapa hal buruk yang dapat mempengaruhi mental manusia. Manusia akan lebih diperbudak oleh internet dan lebih menyukai berselancar dalam dunia maya daripada bersosialisasi membangun sebuah relasi antar kehidupan (Ngafifi , 2014).

Selain memberikan keburukan terhadap manusia, perkembangan teknologi juga dapat menjadi pemicu rusaknya alam yang ada di bumi. Seperti halnya era

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

globalisasi, era ini dipicu oleh teknologi yang berkembang pesat tanpa melihat dampak buruknya bagi lingkungan. Contoh kerugian akibat perkembangan teknologi tersebut adalah efek rumah kaca. Efek rumah kaca memiliki peran besar dalam pembentukan era globalisasi dengan merusak ozon perlahan-lahan (Cahyono , 2002). Proses terjadinya efek rumah kaca berawal dari radiasi matahari yang mengenai bumi namun gagal. Radiasi yang seharusnya diserap kembali oleh bumi, tapi terhalang oleh rumah kaca yang akan memantulkan kembali panas dari radiasi matahari menuju ozon sehingga dapat menipiskan lapisan tersebut. Penipisan ozon yang secara berkala ini dapat menyebabkan globalisasi (Meivana, Sulistiowati, dan Soejachmoen , 2004).

Pada era globalisasi, panas dan radiasi matahari langsung mengenai bumi yang seharusnya diserap oleh lapisan ozon sebanyak 93-99%. Namun, karena kapasitas ozon yang semakin menipis menyebabkan sinar *Ultra Violet*(UV) yang menuju bumi semakin besar dan dapat menyebabkan berbagai masalah bagi manusia salah satunya adalah penyakit kanker kulit(Thangavel , 2018). Kanker sendiri merupakan tumbuhnya sel yang ada pada tubuh secara tidak normal dan dapat mengganggu kinerja sel yang ada pada tubuh. Sedangkan kanker kulit merupakan pertumbuhan sel kulit yang tidak normal yang biasanya disebabkan oleh paparan sinar UV yang berlebihan (Cancer Council , 2016).

Pertumbuhan sel yang tidak normal memiliki dua macam yaitu *benign* (tumor jinak) dan *malignant* (tumor ganas). Pertumbuhan *benign* lebih lambat daripada *malignant* karena infeksi *malignant* lebih cepat merambat keseluruh tubuh. *Benign* sendiri tidak mengancam nyawa manusia sehingga untuk kasus *benign* sangat jarang yang menimbulkan kematian. *Benign* juga tidak menyerang jaringan yang ada pada kulit dan tidak menyebar ke bagian lain dari manusia.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Benign bisa dihapus dan tidak bisa tumbuh lagi di tempat yang sama. Salah satu contoh *benign* adalah tahi lalat, sedangkan *malignant* dapat memicu kematian dengan kasus kematian yang bisa dikatakan tinggi. *Malignant* bisa menyerang jaringan lain dan dapat menyebar menuju jaringan yang lain. *Malignant* bisa saja dihapus namun akan terus tumbuh. Contoh *malignant* pada kulit adalah *Basal Cell Cancer*, *Squamous Cell Cancer*, dan *melanoma*(National Cancer Institute , 2006). Kanker kulit memiliki berbagai macam jenis kanker sel basal yang merupakan kanker kulit paling umum dan hanya menyebar di salah satu bagian tertentu. Kanker sel basal dimulai pada lapisan sel basal kulit. kanker ini biasanya terjadi pada kulit yang sering terpapar sinar matahari. Salah satu contoh tempat yang sering terjadi kanker sel basal adalah pada wajah karena sering kali terpapar sinar matahari (National Cancer Institute , 2006).(Cancer Council , 2016).

Selain kanker sel basal ada juga kanker sel skuamosa. Kanker sel skuamosa memiliki ciri-ciri berwarna merah, tidak menonjol, dan dapat menyebar jika tidak ditangani. Kanker sel skuamosa dimulai pada sel skuamosa. Kanker ini sering kali terjadi pada kaum berkulit hitam. Kanker ini kerap menyerap kulit yang jarang terpapar cahaya matahari seperti betis dan paha(Cancer Council , 2016)(National Cancer Institute , 2006). Jenis kanker kulit yang terakhir adalah melanoma yang merupakan kanker kulit paling ganas yang dapat menyebar kedalam organ dalam manusia seperti jaringan limfa dan jaringan hati. Melanoma dimulai pada melanosit (sel pigmen). Melanoma dapat terjadi pada permukaan kulit mana saja baik pria maupun wanita. Pada pria sering ditemukan pada kulit kepala dan leher atau antara bahu dan pinggul. Pada wanita sering ditemukan pada kulit kaki bagian bawah dan antara bahu dan pinggul. Melanoma jarang terjadi pada orang dengan kulit gelap dan bila terjadi pada orang dengan kulit gelap, biasanya ditemukan dibawah kaki

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

atau telapak tangan(Cancer Council , 2016)(National Cancer Institute , 2006).

World Health Organization(WHO) menyebutkan bahwa penderita kanker kulit terus meningkat dari tahun ke tahun pada tiap dekadenya. Terdapat sekitar 2-3 juta penderita kanker kulit non melanoma dan 132.000 pasien penderita penyakit kanker kulit melanoma(World Health Organization , 2016). Hal ini disebabkan oleh radiasi matahari yang semakin hari semakin besar dikarenakan penipisan lapisan ozon pada bumi. Berdasarkan data WHO tersebut perlu dilakukan deteksi dini sehingga pengobatan dapat dilakukan secepat mungkin tanpa ada jatuhnya korban jiwa. Penyakit apapun jika diketahui sebabnya secara dini akan dapat dilakukan pengobatan. Hal ini sudah diatur pada Al-Qur'an surat Asy-Syu'ara ayat 80 yang diperkuat oleh hadits riwayat bukhari yang berisi "Tidaklah Allah menurunkan penyakit kecuali Dia juga menurunkan penawarnya" yang berarti semua penyakit pasti terdapat obat ketika dapat ditangani secara benar dan dapat diidentifikasi secara dini. Berdasarkan hadits dan ayat Al-Qur'an tersebut, maka untuk menangani penyakit kanker kulit dapat dilakukan dengan deteksi dini penyakit ini.

Kanker kulit dapat diidentifikasi berdasarkan gejala-gejala yang disebabkan. Para dokter menyebut gejala kanker kulit ini sebagai "ABCDE" yang merupakan *Asymmetry*, *Border that is irregular*, *Color that is uneven*, *Diameter*, dan *Evolving*. *Asymmetry* atau tidak simetris menjelaskan bahwa bentuk yang muncul pada kulit tidak simetris atau lebih tepatnya abstrak. *Border that is irregular* menjelaskan bahwa tepi dari kanker tidak rata dan biasanya memiliki tekstur yang kasar dan kabur. *Color that is uneven* menjelaskan bahwa warna yang ada pada kanker tidak rata, dengan kata lain banyak perpaduan warna antara hitam, coklat, abu-abu, dan merah. *Diameter* menjelaskan bahwa diameter dari kanker

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sekitar 6 milimeter sampai 0.25 inchi. Cara terakhir untuk mendeteksi kanker kulit adalah *Evolving* yang menjelaskan bahwa akan terjadi perubahan pada kanker setelah beberapa bulan terjangkit (American Cancer Society , 2007). Deteksi dini kanker kulit bisa dilakukan oleh dokter dengan melihat secara langsung berdasarkan fitur ABCDE, namun cara ini kurang efisien karena penglihatan seseorang bisa saja berbeda sehingga dibuatlah CAD untuk membantu pihak medis dalam mendeteksi kanker kulit. Pengidentifikasian kanker kulit bisa dilakukan melalui *Computer Aided Diagnosys*(CAD) *system* dengan melalui beberapa tahap seperti pengolahan citra digital, ekstraksi fitur, dan proses pengklasifikasian(Adyanti, Asyhar, Novitasari, et. al. , 2017). Pada penelitian ini sistem CAD yang digunakan adalah *preprocessing* yang berisi tentang pengolahan citra digital, proses ekstraksi fitur tuk mendapatkan parameter hasil *preprocessing*, dan klasifikasi untuk menentukan pasien yang terjangkit.

Tahap *preprocessing* digunakan untuk memperbaiki citra sehingga mempermudah proses pengidentifikasian fitur dari gambar yang digunakan. *Preprocessing* yang digunakan adalah *hair removal*, *noise removal*, *contrast enhancement*, *lesion segmentation*. Langkah-langkah ini berdasarkan penelitian yang pernah dilakukan oleh Dr. Abbas Hanon dkk (Albadra dan Tiun , 2017) Pada penelitian tersebut digunakanlah proses *hair removal* dengan menggunakan *dull razor filtering*, *noise removal* dengan menggunakan *median filtering*, *contrast enhancemet* dengan menggunakan *histogram equalization*, dan segmentasi dengan menggunakan *otsu thresholding*. Berdasarkan nilai hasil akurasi yang didapatkan yaitu sebesar 93%, maka proses yang digunakan sudah cukup baik sehingga *preprocessing* ini akan dipakai pada penelitian ini(Alasadi dan Alsafy , 2015).

Setelah fitur dari sebuah gambar didapatkan maka diperlukan sebuah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

metode klasifikasi untuk menentukan gambar tersebut tergolong data kanker ataupun tidak. Metode yang akan digunakan adalah *Deep Extreme Learning Machine* (DELM). DELM merupakan pengembangan dari metode *Extreme Learning Machine*. DELM dihasilkan dari penggabungan metode *Multilayer Extreme Learning Machine* (MLELM), *Kernel Extreme Learning Machine* (KELM), dan *Extreme Learning Machine Auto Encoding* (ELM-AE)(Ding, Zhang, Xu, Guo, dan Zhang , 2015). *Extreme Learning Machine* merupakan metode jaringan saraf tiruan dengan propagasi maju tanpa dilakukan propagasi mundur dan juga hanya memiliki satu *hidden layer* saja. ELM terkenal sebagai metode jaringan saraf tiruan yang terbilang cepat daripada metode jaringan saraf tiruan lainnya dikarenakan berbasis *single hidden layer*(Huang, Zhu, dan Siew , 2006). Ide awal dari ELM adalah perhitungan nilai parameter pada *hidden layer* seperti bias ataupun bobot yang tidak dibutuhkan dalam proses *learning*. Parameter tersebut dibangkitkan dengan cara melakukan *random* sehingga pada proses *learning* tidak memakan banyak waktu. Meskipun parameter dibentuk secara *random*, namun hasil klasifikasinya dapat dikatakan akurat dengan memiliki akurasi yang tinggi dan banyak peneliti yang memakai ini karena efisiensi waktu penggunaannya. Salah satu peneliti yang menggunakan metode ini adalah M. S. Sreekanth, R. Rajesh, dan J. Satheeshkumar dalam penelitaannya tentang mengklasifikasikan curah hujan dan badai petir dengan membandingkan metode SVM, ANN, dan ELM dengan akurasi terbaik didapat dengan menggunakan metode ELM sebesar 87.69% (Sreekanth, Rajesh dan Satheeshkumar , 2015).

Metode ELM memiliki beberapa kekurangan dalam mendeteksi data tertentu sehingga ELM dimodifikasi dengan menggunakan sebuah kernel agar ELM bisa mengklasifikasikan data dengan baik dengan kata lain memaksimalkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

peluang kebenaran dalam klasifikasi. Metode ELM yang digabungkan dengan kernel disebut *Kernel Extreme Learning Machine* (KELM). KELM pertama kali digagas oleh Huang dkk pada tahun 2010 dengan menerapkan beberapa fungsi kernel kedalam ELM dengan cara mensubtitusikan data yang sudah dilakukan proses kernel untuk diproses kedalam *hidden layer* ELM(Li, Rong, dan Li , 2014). Ide awal dari gagasan KELM adalah metode SVM. Huang dkk meneliti sisi baik SVM yang dilakukan proses kernel terlebih dahulu sebelum menuju proses learning SVM sehingga pembentukan garis hyperplane semakin akurat dalam membagi kedua kelas(Deng, Ong, dan Zheng , 2016). Berdasarkan hal tersebut dilakukan hal yang sama dengan melakukan proses kernel terlebih dahulu yang menghasilkan generalisasi parameter ELM yang lebih optimal. Hal ini menyebabkan proses *learning* pada ELM memiliki waktu yang lebih singkat karena lebih cepat mendapatkan bobot yang optimal. Penelitian mengenai KELM pernah dilakukan oleh Chen Chen dkk tentang klasifikasi spectral-spatial dari gambar hyperspectral. Penelitian tersebut membandingkan metode KELM dan SVM yang menghasilkan akurasi terbaik pada metode KELM. Pada penelitian tersebut juga membandingkan waktu *learning* antara kedua metode dengan perbandingan waktu antara SVM dengan KELM adalah 0.94s dan 0.23s. Berdasarkan hasil tersebut dapat disimpulkan bahwa KELM lebih cepat dan lebih akurat daripada SVM(Chen, Li, Su, dan Liu , 2014).

ELM pernah dilakukan modifikasi oleh Hinton dan Salakhutdinov pada tahun 2006 yang awalnya ELM hanya metode klasifikasi *single hidden layer* kemudian dimodifikasi sehingga menjadi *Multilayer Extreme Learning Machine* (ML-ELM). ML-ELM memiliki 2 proses yaitu *unsupervised learning* dan *supervised learning*. Berdasarkan pernyataan tersebut, metode *unsupervised*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

learning yang dipakai adalah *Extreme learning Machine Auto Encoder* (ELM-AE). Proses pertama dalam metode ini adalah *unsupervised learning* dengan menggunakan metode ELM-AE untuk mendapatkan parameter ELM terbaik di setiap layer. Setelah didapatkan parameter ELM terbaik, maka dilakukan proses *fine-tuning* dengan menggunakan *supervised learning* yaitu metode ELM(Wen, Liu, Yan, dan Sun , 2018). Penelitian mengenai ML-ELM pernah diteliti oleh Lijuan Duan dkk mengenai klasifikasi sinyal EEG. Pada penelitian ini dibandingkan metode klasifikasi ML-ELM dengan metode lain seperti KNN dan bayes. Perbandingan hasil akurasi antara KNN, bayes, dan ML-ELM adalah 92.15%, 90.44%, dan 94.20% sehingga dapat disimpulkan bahwa berdasarkan paper tersebut metode klasifikasi yang paling baik untuk dipakai adalah ML-ELM(Duan,Bao,Miao Xu, dan Chen , 2016).

ML-ELM dan KELM merupakan bibit dari terbentuknya *Deep Extreme Learning Machine* (DELM). Penggabungan metode ML-ELM dan KELM ditujukan untuk mengoptimalkan proses *learning* pada ELM. Metode DELM merupakan salah satu metode *deep learning* yang cepat dalam melakukan proses *learning*. *Deep learning* merupakan *machine learning* yang digunakan pada kecerdasan buatan yang memberikan data input secara detail dengan meningkatkan ukuran neuron dan jumlah hidden layer pada proses *learning*. Peningkatan neuron dan jumlah *hidden layer* mempengaruhi proses *learning* sehingga *deep learning* memiliki proses *learning* yang lambat. Untuk mengatasi masalah ini, *deep learning* digabungkan dengan metode ELM sehingga mendapatkan metode DELM yang memiliki proses learning yang cepat dan efektif(Kutlu , 2018). Penelitian mengenai DELM pernah dilakukan oleh Shifei Ding dkk dengan membandingkan antara basic ELM, KELM, ML-ELM, dan DELM menggunakan data EEG,

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Diabetes, dan ionosphere. Pada penelitian tersebut dihasilkan akurasi terbaik didapatkan dengan menggunakan metode DELM sehingga terbukti bahwa metode DELM dapat mengklasifikasikan lebih baik daripada *basic ELM* dan *modified ELM* lainnya(Ding, Zhang, Xu, Guo, dan Zhang , 2015). Penelitian tersebut dapat digunakan sebagai acuan dalam memilih metode klasifikasi. Berdasarkan rangkaian penelitian terdahulu yang sudah dipaparkan, maka dilakukan penelitian mengenai deteksi kanker kulit dengan menggunakan DELM sebagai metode klasifikasi yang akurat dan cepat. Penelitian ini ditujukan kepada pihak medis maupun masyarakat umum untuk mempermudah dalam mengenali penderita penyakit kanker kulit.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang ada, maka penenliti merumuskan permasalahan sebagai berikut :

1. Bagaimana hasil kinerja *Deep Extreme Learning Machine* dalam mengklasifikasikan kanker kulit terhadap banyaknya *hidden nodes*?
2. Bagaimana model terbaik yang didapatkan untuk proses klasifikasi kanker kulit menggunakan metode *Deep Extreme Learning Machine*?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang ada, maka tujuan penelitian ini adalah :

1. Dapat mengetahui hasil kinerja *Deep Extreme Learning Machine* dalam mengklasifikasikan kanker kulit terhadap banyaknya *hidden layer*.
2. Dapat mengetahui model terbaik yang didapatkan untuk proses klasifikasi kanker kulit menggunakan metode *Deep Extreme Learning Machine*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1.4. Manfaat Penelitian

1. Manfaat Teoritis

   Berdasarkan tujuan penelitian yang akan dicapai, maka diharapkan memberikan manfaat dalam bidang akademik secara langsung ataupun secara tidak langsung. Adapun manfaat penelitian ini adalah:

   a. Dapat memberikan sumbangan ilmiah dalam ilmu kesehatan dengan cara memberikan inovasi dalam menggunakan metode untuk mengidentifikasi kanker kulit.

   b. Dapat digunakan sebagai referensi pada penelitian berikutnya yang berhubungan dengan kanker kulit.

2. Manfaat Praktis

   Manfaat praktis dapat diambil dari:

   a. Bagi penulis, penulis dapat menerapkan metode yang digunakan untuk mengidentifikasi kanker kulit serta penulis mendapatkan wawasan baru tentang metode yang baru dipelajari.

   b. Bagi khalayak umum, Penulis berharap dapat memberikan manfaat melalui hasil tentang cara mengidentifikasi kanker kulit yang dipaparkan sehingga dapat mengurangi adanya penyebaran penyakit yang besar.

1.5. Batasan Masalah

Mengingat ruang lingkup penelitian yang begitu luas, maka penulis memberi batasan pada penelitian ini, yaitu:

1. Data yang digunakan dalam penelitian ini menggunakan data citra kanker(SCC,

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BCC, dan Melanoma) dan tumor jinak(Tahi lalat, Bekas Luka, Tanda Lahir).

2. Klasifikasi kanker kulit di bagi dalam dua kategori yaitu Tumor Jinak dan Kanker.

3. Data yang digunakan adalah data citra dermoscopic.

4. Keluaran dari sistem ini adalah berupa informasi hasil klasifikasi *benign* dan *malignant* dengan menggunakan metode DELM.

5. *Hidden layer* yang digunakan sebanyak 1000.

1.6. Sistematika Penulisan

Penulisan ini disusun dengan sistematika sebagai berikut:

Bab I Pendahuluan, Bab ini berisi tentang penjelasan mengenai permasalahan apa saja yang menjadi latar belakang dilakukannya penelitian, yaitu banyaknya kematian di dunia akibat penyakit kanker kulit yang terus bertambah. Selain itu pada bagian ini juga menjelaskan mengenai pemilihan dan gambaran singkat metode *preprocessing* dan metode klasifikasi DELM yang digunakan untuk pengolahan citra medis dalam mendeteksi penyakit kanker kulit, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

Bab II Dasar Teori, Bab ini berisi tentang teori-teori yang memiliki keterkaitan dan mendukung penyelesaian masalah dalam penelitian ini. Tinjauan pustaka terkait topik yang digunakan yaitu teori tentang kanker kulit, metode dalam *preprocessing*, dan metode klasifikasi *Deep Extreme Learning Machine*

Bab III Metode Penelitian, Bab ini berisi tentang cara melaksanakan penelitian yang dimulai dari bagaimana proses mendapatkan data dan mengolah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

data sedemikian sehingga rumusan masalah dapat terselesaikan. Bab ini biasanya terdiri dari jenis penelitian, subjek penelitian, data penelitian, teknik analisis data, dan lainnya sesuai dengan kebutuhan penelitian.

Bab IV Hasil Dan Pembahasan, Bab ini berisi tentang uraian hasil penelitian dan pembahasan dalam rangka untuk menjawab rumusan masalah yang telah diuraikan pada bab pendahuluan. Dalam penelitian ini diuraikan mengenai hasil dari *preprocessing*, hasil dari proses klasifikasi menggunakan DELM, dan analisis dari hasil yang diperoleh.

Bab V Penutup, Bab ini berisi tentang kesimpulan dari keseluruhan isi yang telah dibahas dan saran perluasan mengenai hasil penelitian dari analisis deteksi kanker kulit berdasarkan citra dengan menggunakan metode *Deep Extreme Learning Machine*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB II

TINJAUAN PUSTAKA

2.1. Kanker

Kanker adalah salah satu penyakit mematikan di dunia. Kata kanker berasal dari kata Yunani "karkinos" yang berarti udang karang dan merupakan istilah umum untuk ratusan tumor ganas yang masing-masing sangat berbeda satu sama lain (Jong , 2005). Sel yang terdapat di tubuh kehilangan pengendalian sehingga mengalami pertumbuhan yang tidak normal, cepat, serta tidak terkendali. Pertumbuhan sel kanker tersebut disebabkan oleh kerusakan *Deoxyribose Nucleic Acid* (DNA). Selain itu sel kanker dapat menyebar melalui dinding pembuluh darah dan limfe kemudian terlepas dan memaksa masuk ke tempat lain di dalam tubuh, sehingga akan tumbuh kanker di wilayah lain yang nantinya menjadi suatu yang sulit untuk ditangani(Amin, Miah, dan Mia , 2015).

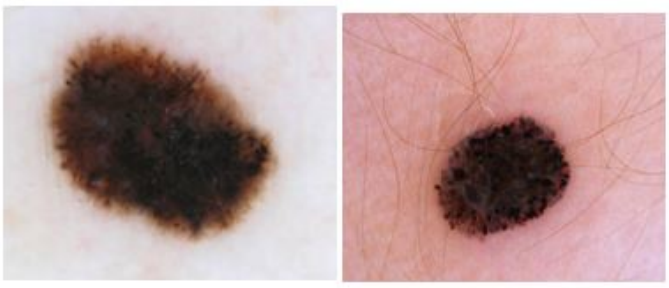
Sel kanker dapat menyerang pria, wanita, anak-anak, maupun dewasa. Kanker biasa juga disebut dengan tumor ganas karena penyembuhannya yang susah dan penyebarannya yang cepat menuju sel yang lain. Cara mendeteksi kanker pada tubuh bisa dilakukan dengan beberapa cara antara lain, *Biopsy, Endoscopy, Magnetic Resonance Image* (MRI), *Computed Topography* (CT), *mammogram*, dan tes darah. Kanker sering terjadi akibat narkoba, polusi yang terlalu banyak, ataupun dari gen. Kanker sendiri memiliki banyak macam yaitu, kanker paru-paru, kanker payudara, kanker prostat, dan masih banyak lainnya. Salah satu kanker yang perlu diwaspadai keganasannya adalah kanker kulit

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dikarenakan tingkat kematian di Indonesia bahkan di Dunia sangat tinggi(Radha dan Rajendiran , 2013).

2.2. Kanker Kulit

Kulit adalah organ yang memisahkan tubuh dan lingkungan manusia. Kulit bertindak sebagai penghalang yang melindungi tubuh terhadap radiasi UV, zat beracun, infeksi. Epidermis adalah lapisan kulit terluar. Sel-sel keratinosit, melanosit dendritik, Merkel, dan Langerhans adalah berbagai jenis sel yang ada dalam epidermis. Dermis yang mendasarinya mengandung jaringan ikat dengan antigen yang menghadirkan sel-sel dendritik dermal, sel mast dan sel-T memori(Fuchs dan Raghayan , 2002). Contoh gambar kanker kulit bisa dilihat pada Gambar 2.1.



Gambar 2.1 Kanker Kulit

(ISIC Archive , 2018)

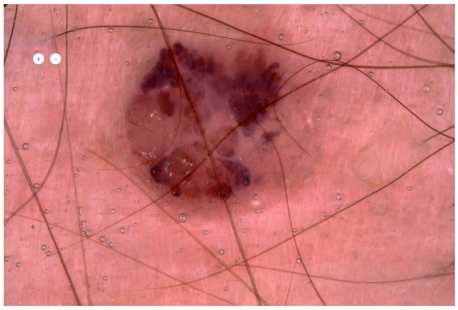
Salah satu penyakit pada kulit yang berbahaya yaitu kanker kulit. Insiden kanker kulit pada manusia telah meningkat dari hari ke hari. Alasan utama untuk kanker kulit adalah karena paparan UV dalam jumlah besar. Sejumlah besar radiasi UV mencapai permukaan bumi karena menipisnya lapisan ozon. Kanker kulit terdiri dari dua jenis, yaitu *Melanoma Maligna*(MM) dan *Melanoma non maligna*(Scherer dan Rajiv , 2010). *Melanoma maligna* terjadi karena paparan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sinar matahari yang intens dan riwayat terbakar matahari. Kanker kulit pada tahap awal dapat disembuhkan dengan mudah dengan prosedur atau teknik sederhana tetapi kanker kulit lanjut tidak dapat diobati secara efektif dengan obat apa pun. Jadi ada kebutuhan untuk mendeteksi dan mengobati penyakit pada tahap awal(Johnson, Dolan, Hamilton, et. al. , 2010). Kanker kulit memiliki 3 jenis yaitu *Basal Cell Carcinoma(BCC), Squamous Cell Carcinoma(SCC)*, dan *melanoma*.

1. Basal Cell Carcinoma (BCC)

Jenis kanker kulit yang paling umum adalah BCC. Mereka umumnya terjadi di daerah kepala dan leher diikuti oleh batang dan ekstremitas. Mereka biasanya muncul dari lapisan basal epidermis. BCC diklasifikasikan menjadi 3 jenis; *Superfisial, nodular,* dan *sclerosing / morpheaform*. Bentuk superfisial BCC dapat dilihat pada batang tubuh dan ekstremitas sebagai wabah eritematosa. Lesi BCC nodular umumnya terlihat di kepala dan leher dan tampak seperti *papula telangiectatic* dengan perbatasan bergulung. Lesi morfeaform sering menyerupai bekas luka dan biasanya yang paling sulit diidentifikasi pada inspeksi visual saja, sering memiliki karakteristik seperti mutiara pada BCC superfisial dan nodular.



Gambar 2.2 Basal Cell Carcinoma

(ISIC Archive , 2018)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pasien sindrom gorlin sering dikaitkan dengan BCC. Orang-orang ini biasanya memiliki BCC di pusat wajah atau di situs anatomi mana pun. BCC jarang mengalami metastasis tetapi cenderung menyebabkan lebih banyak morbiditas. Squamous Cell Carcinoma mengalami metastasis yang cepat. Penyakit tersebut dapat terjadi karena paparan sinar matahari kronis dan dapat dilihat pada berbagai bagian tubuh yang terpapar sinar matahari(Moller, Reymann, dan Jensen , 1979). Contoh BCC dapat dilihat pada Gambar 2.2.

2. Squamous Cell Carcinoma (SCC)

Squamous Cell Carcinoma (SCC) dianggap sebagai jenis kanker paling umum kedua di AS dengan 2.500 kasus didiagnosis setiap tahun. Biasanya terlihat pada orang Indian kulit hitam dan Asia yang mewakili 30% hingga 65% kanker kulit pada kedua ras. Ini terjadi pada bagian kepala dan leher yang terkena sinar matahari. Secara umum, hasil jangka panjang adalah positif, karena kurang dari 4% kasus karsinoma sel skuamosa berisiko metastasis (dan karenanya mengancam jiwa)(Emmons, Geller, dan Puleo , 2010). Contoh *Squamous Cell Carcinoma* dapat dilihat pada Gambar 2.3.

Gambar 2.3 Squamous Cell Carcinoma

(ISIC Archive , 2018)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3. Melanoma

Melanoma lebih jarang terjadi dibandingkan kanker kulit lainnya. Namun, jauh lebih berbahaya jika tidak ditemukan lebih awal. Hal ini menyebabkan sebagian besar (75%) kematian terkait dengan kanker kulit. Terdapat 36 Jenis kanker kulit yang dikaitkan dengan melanosit dari lapisan epidermis. Penyakit tersebut mensintesis pigmen melanin yang menghasilkan sel-sel kulit dengan perlindungan foto dari sinar UV mutagenik. *Melanoma Maligna* lebih jarang dibandingkan dengan BCC dan SCC. Perawatan untuk melanoma sepenuhnya tidak dikenal. Tingkat melanoma kurang dapat disembuhkan. Pencegahan adalah metode terbaik untuk melanoma. Ini adalah satu-satunya kanker yang dapat dicegah agar tidak terus meningkat (Brantsch,Meisner, dan Schönfisch , 2008). Contoh kanker melanoma dapat dilihat pada Gambar 2.4.

Gambar 2.4 Melanoma

(ISIC Archive , 2018)

2.3. Citra Digital

Citra merupakan gambar dua dimensi yang berisi piksel-piksel dengan nilai tertentu yang dihasilkan dari gambar analog. Gambar analog merupakan kumpulan matriks yang terdiri dari n baris dan m kolom. Sebuah piksel suatu gambar

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

direpresentasikan sebagai baris dan kolom pada matriks gambar analog. Sebagai contoh titik diskrit pada baris n dan kolom m disebut dengan piksel $[n, m]$ (Tim Dosen , 2016).

Citra digital merupakan citra dapat diproses ataupun diolah dengan suatu sistem komputer yang yang hasilnya berupa sebuah gambar analog. Proses pengolahan citra digital pada suatu sistem komputer dinamakan *Image Processing*. Citra digital tersusun atas elemen-elemen yang disebut piksel. Piksel tersusun dari kumpulan matriks dua dimensi dengan indeks baris dan kolom direpresentasikan oleh (x, y) yang merupakan bilangan bulat. Koordinat spasial x dan y dari f dalam fungsi $f(x, y)$. Nilai f menunjukkan nilai tingkat kecerahan atau tingkat keabuan dari suatu citra pada titik tersebut. Suatu citra digital dapat direpresentasikan dalam format $f(x, y) = f(N, M)$ dengan $0 \leq x \leq N - 1$ dan $0 \leq y \leq M - 1$. Kemudian kedua faktor tersebut dinyatakan dengan nilai L yang merupakan nilai maksimal warna intensitas dengan ketentuan $L = 0 \leq f(x, y) \leq L - 1$ Citra digital dibagi dalam tiga jenis, yaitu citra warna, citra *grayscale*, dan citra biner(Tim Dosen , 2016).

Citra *gray scale* merupakan citra skala keabuan dengan nilai intensitas maksimal 255 berwarna putih hingga warna hitam dengan nilai intensitas minimal 0. Pada nilai mendekati 0 tingkat keabuan sebuah citra semakin gelap, sedangkan pada nilai mendekati 255 memiliki intensitas keabuan yang semakin cerah. Citra biner merupakan citra yang hanya memiliki dua nilai derajat keabuan, yaitu 0 dan 1 sebagai representasi dari putih dan hitam. Piksel yang bernilai 1 merupakan piksel objek atau fitur dengan memiliki warna putih, dan piksel yang bernilai 0 adalah latar belakang yang memiliki warna hitam. Sedangkan citra warna atau yang biasa disebut citra RGB merupakan jenis citra yang menyajikan warna dalam

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

komponen R (merah), G (hijau), dan B (biru). Setiap warna menggunakan 8 bit (nilainya berkisar dari 0 hingga 255). Sehingga kemungkinan warna yang dapat disajikan adalah 255 x 255 x 255 (Wahyudi, Triyanto, dan Ruslianto , 2017). Citra digital dapat direpresentasikan dalam suatu fungsi $f(x, y)$. Koordinat x adalah koordinat posisi dengan $0 \leq x \leq M$, dan y adalah koordinat posisi dengan $0 \leq y \leq N$, dimana M merupakan lebar citra, dan N adalah tinggi citra. Sedangkan f merupakan fungsi citra digital, dimana $0 \leq f \leq L$, dengan L sebagai derajat keabuan citra (Tim Dosen , 2016). Jenis-jenis citra dapat dilihat melalui Gambar 2.5.

Gambar 2.5 Citra Digital RGB, Gray Scale, dan Biner

(Wahyudi, Triyanto, dan Ruslianto , 2017)

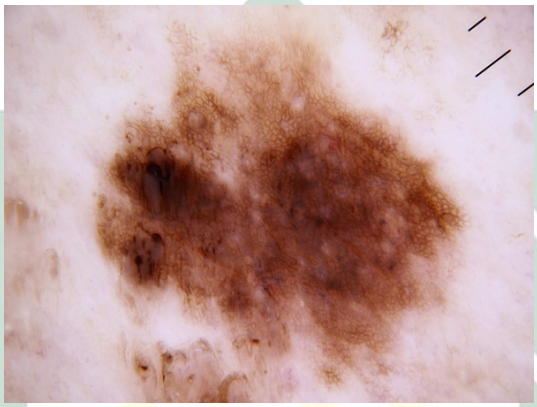
2.4. *Dermoscopic Image*

Dermoscopic image adalah gambar medis yang berbasis mikroskop untuk meningkatkan diagnosa pada *skin lession* berdasarkan analisis warna dan strukturnya. Warna pada hasil gambar dermoscopic memiliki indikasi masing-masing yang merepresentasikan elemen yang ada pada kulit seperti warna kuning merupakan keratin, warna merah merupakan darah, dan warna putih merupakan kalogen. Warna juga merepresentasikan kedalaman pigmen pada kulit. Semakin dalam pigmen kulit dan mendekati epidermis maka semakin gelap warna yang dihasilkan dari gambar dermoscopic(Zaqout , 2016).

Sistem klasifikasi pada gambar dermoscopic biasanya menggunakan fitur

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

visual tingkat rendah seperti bentuk, tekstur, warna, dll. Proses ekstraksi fitur pada gambar ini juga biasanya menggunakan diameter atau luas jaringan kanker, fitur yang didapatkan dari matriks *co-occurrance* pada gambar keabuan, fitur ragam warna yang terdapat pada gambar, dll (Bino , 2012). Contoh gambar dermoscopic dapat dilihat pada Gambar 2.6



Gambar 2.6 Hasil Pengambian Gambar Pada Dermoscopic Camera

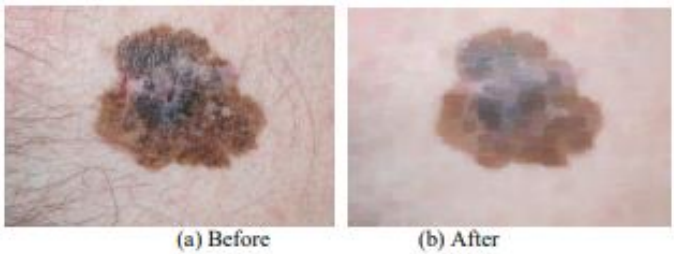
(ISIC Archive , 2018)

2.5. *Dull Razor Filtering*

Dull Razor merupakan metode yang ada pada *preprocessing* yang digunakan untuk menghilangkan piksel rambut. *Dull Razor* pertama kali diperkenalkan pada tahun 1997 untuk menghilangkan rambut gelap secara digital dari *dermoscopic image* (Satheesha, Satyanarayana, dan Giriprasad , 2014). Teknik utama yang digunakan *Dull Razor* untuk menemukan piksel rambut hitam tebal adalah *grayscale morphological closing operation* dengan menggunakan *structuring element curve linear* agar lebih spesifik dalam menghilangkan lekungan rambut. Dari sudut pandang geometris, rambut hitam tampak seperti sebuah garis lengkung berwarna hitam (Lee, Gallagher, Coldman, dan Mclean , 1997).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Dull razor sendiri mengidentifikasi piksel rambut dari gambar *gray scale* yang kemudian di identifikasi bagian yang memiliki kurva lengsung dengan intensitas cahaya yang rendah. Umumnya pada metode ini digunakan proses *smoothing* yang berfungsi dalam penghalusan citra rambut dan disesuaikan dengan warna citra kulit yang meiliki intensitas cahaya yang tinggi (Hosyar dan Al-Jumaily , 2014). Pengaruh terbesar dalam menggunakan metode ini adalah warna rambut dan warna kulit. Jika warna rambut dan kulit memiliki yang relatif sama, maka ketepatan dalam metode ini akan menurun. Hal ini menyebabkan rambut tidak dapat dihilangkan secara maksimal (Huang, kwan, Chang, Liu, Chi, dan Chen , 2013).



Gambar 2.7 Dull Razor Filtering

(Huang, kwan, Chang, Liu, Chi, dan Chen , 2013)

Dull Razor berfokus pada menghilangkan hanya rambut hitam tebal yang terdiri dari tiga tahap dasar:

1. Mengidentifikasi lokasi rambut gelap dengan membuat gambar menjadi *gray scale* melihat tingkat pencahayaannya.
2. Mengganti piksel rambut dengan piksel kulit terdekat.
3. Memperhalus hasil akhir dengan menggunakan *adaptive median filter*.

Gambar yang dihasilkan memiliki bekas berupa garis tipis dari rambut yang dihilangkan, tetapi secara keseluruhan proses ini bekerja cukup efektif untuk

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

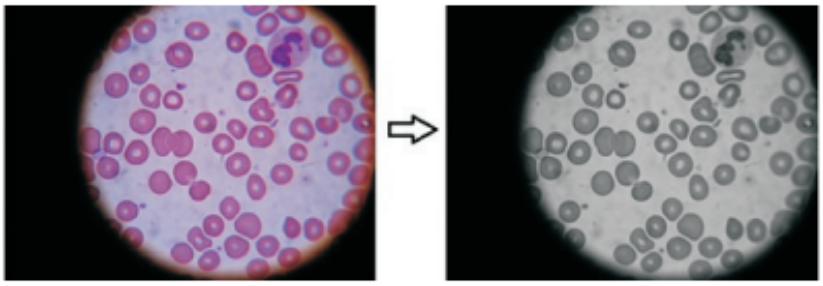
menghasilkan hasil yang memuaskan ketika melakukan segmentasi sel kanker (Choudhari dan Biday , 2014). Hasil dari proses *Dull Razor Filtering* dapat dilihat pada Gambar 2.7.

2.6. *Grayscale Image*

Umumnya gambar memiliki berbagai macam warna seperti pada tingkat warna RGB (*Red, Green,* dan *Blue*). Beberapa kasus pengelolahan citra digital membutuhkan gambar dalam skala *grayscale* untuk mendapatkan beberapa informasi yang lebih baik. Gambar *grayscale* merupakan suatu gambar yang hanya memiliki warna putih sampai hitam atau dalam range 0-255. Gambar *grayscale* memiliki tingkat *noise* yang sedikit sehingga gambar *grayscale* kerap dijadikan sebagai data pada berbagai proses *CAD system* (Santi , 2011). Cara merubah gambar RGB ke dalam grayscale dapat menggunakan Persamaan 2.1.

$$s = (0.299(r)) + (0.587(g)) + (0.144(b)) \tag{2.1}$$

Dengan s merupakan nilai hasil konversi RGB ke *grayscale*. Nilai r, g, dan b berturut-turut merupakan nilai piksel RGB pada *Red Channel*, *Green Channel*, dan *Blue Channel*. Hasil dari proses *grayscale* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Konversi RGB ke Grayscale

(Biswas dan Ghoshal , 2016)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.7. *Median Filter*

Median Filter adalah filter non-linear yang dapat digunakan untuk menghilangkan *noise*. *Median filter* ini adalah semacam teknik *preprocessing* yang hasilnya digunakan untuk proses selanjutnya. Metode ini digunakan untuk menghilangkan *"salt and pepper" noise*. Setiap piksel di gambar mempunyai probabilitas $\frac{p}{2}(0 < p < 1)$ yang terkontaminasi oleh titik hitam dan titik putih(Gopinath , 2011). *Median filter* mempunyai dua keunggulan dibandingkan dengan *mean filter*, yaitu:

1. *Median filter* lebih baik dari *mean filter* karena satu piksel yang sangat tidak representatif pada gambar tidak akan mempengaruhi nilai *median* secara signifikan.
2. Nilai median sebenarnya adalah salah satu nilai piksel suatu gambar sehingga nilai yang dihasilkan lebih realistis daripada *mean filter*.

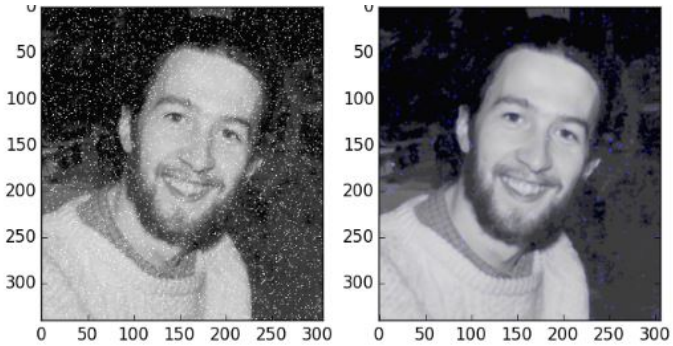
Median filter merupakan metode pengelolaan citra digital yang berbasis data statistik nilai keabuan suatu citra. Tahapan dalam *median filter* adalah mengganti nilai *noise* pada citra dengan nilai hasil median yang didapatkan. nilai keabuan pada nilai ketetanggan akan diurutkan dan diambil nilai mediannya. Umumnya ketetanggaan dalam median filter memiiki nilai sebesar 3 x 3 (Zhu dan Huang , 2012). Persamaan median filter dapat dilihat pada Persamaan 2.2.

$$g(x, y) = median\ \{f(x - i, y - j), i, j \in W\} \tag{2.2}$$

Dimana $f(x, y)$ merupakan gambar asli dari citra, $g(x, y)$ merupakan *output* atau hasil dari *median filter*. W merupakan *2D mask* yang harus berupa matriks *square* yang memiliki ukuran $n\ x\ n$ seperti $3\ x\ 3$ atau $5\ x\ 5$. Bentuk dari *mask* biasanya

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

berupa *linear, square, circular, cross*, dan sebagainya. Hasil dari proses median filtering dapat dilihat pada Gambar 2.9.



Gambar 2.9 Median Filtering

(Zhu dan Huang , 2012)

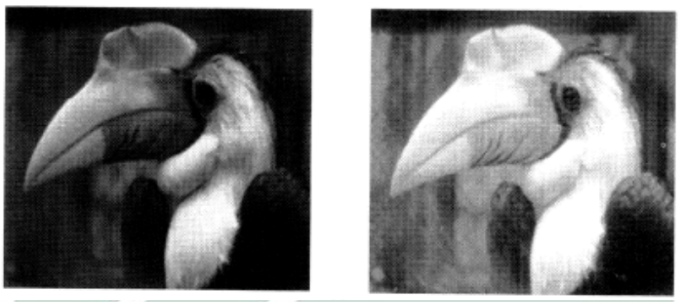
2.8. *Histogram Equalization*

Histogram Equalization(HE) merupakan proses untuk meningkatkan kecerahan atau kontras dan memperbaiki kualitas histogram citra sehingga menghasilkan citra yang lebih mudah untuk dianalisis. HE sangat berguna untuk gambar yang memiliki *background* dan fitur dengan intensitas yang sama(keduanya gelap atau keduanya terang). Metode ini juga dapat meningkatkan kualitas citra sehingga detail informasi pada suatu gambar lebih mudah teridentifikasi (Guveren, 2007). Gambar histogram menyediakan informasi terkait dengan penyebaran intensitas piksel ke dalam gambar. Gambar yang terlalu terang atau terlalu gelap akan memiliki histogram yang sempit. Oleh karena itu, maka diperlukan pengubahan penyebaran nilai intensitas gambar dengan menggunakan peningkatan gambar(Zulpe dan Pawar , 2012). Persamaan *histogram equalization* dapat didefinisikan pada Persamaan 2.3.

$$X' = T(x) = \sum_{i=0}^{x} n_i \frac{\text{MaxIntensity}}{N} \tag{2.3}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Dengan X' merupakan nilai intensitas baru, n_i merupakan jumlah piksel dengan intensitas i, dan N adalah jumlah total dari piksel yang ada pada citra. Hasil dari proses Histogram Equalization dapat dilihat pada Gambar 2.10.



Gambar 2.10 Histogram Equalization

(Zulpe dan Pawar , 2012)

2.9. *Histogram of Oriented Gradient (HOG) Descriptor*

Content-based image retrieval merupakan sebuah metode pengambilan fitur yang menarik pada sistem CAD. Proses ini memungkinkan pengguna untuk menemukan sebuah fitur berguna dari gambar dengan ukuran piksel yang besar. Proses CAD seperti ini sering disebut sebagai *feature descriptor*. Dimana *feature descriptor* mengelola gambar dengan menemukan matriks piksel pada gambar yang dianggap memiliki peran penting dalam proses klasifikasi (Cerna dan Menotti , 2010). Salah satu metode *feature descriptor* adalah *Histogram of Oriented Gradient* (HOG). Metode ini memiliki tujuan untuk mengenali gambar sebagai kelompok histogram lokal. Histogram ini menggunakan angka pada bidang arah gradien lokal gambar (Dalal dan Triggs , 2012).

Ide dasar dari HOG adalah bentuk dan warna dari sebuah objek yang memiliki karakteristik tersendiri berdasarkan distribusi dari gradien pada intensitas cahaya dan juga arah dari bentuk tepinya (Hamdi , 2018). HOG menghitung tiap grid pada suatu sel yang memiliki nilai normalisasi dari *local contrast histogram*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

untuk meningkatkan performa dalam mendeteksi fitur sehingga HOG dapat mendeteksi sebuah fitur bentuk dari gambar dengan baik dikarenakan pada *local intensity gradient* tidak akan pernah sama terhadap gradien pada piksel yang lain(Churchill , 2013). Tahapan HOG yaitu(Atika et. al. , 2016)

a. Menghitung nilai gradien dari setiap piksel.

b. Menentukan jumlah bin orientasi yang akan digunakan dalam pembuatan *histogram* simulasi untuk bin dapat dilihat pada Gambar 2.12.

c. Normalisasi *histogram* dalam kelompok lokasi yang ditentukan.

Gambar dibagi menjadi daerah terhubung kecil yang disebut sel, dan untuk setiap sel dihitung histogram arah gradien atau orientasi tepi setiap piksel dalam sel. Dalam menghitung nilai gradien diterapkan 1D titik tengah *discrete derivative mask* baik secara vertikal maupun horizontal. Pendekatan ini melibatkan penyaringan skala abu-abu dengan kernel filter yang dapat dilihat pada Persamaan 2.4(Berkant , 2015)

$$D_x = [-1\ 0\ 1] \qquad D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \tag{2.4}$$

Nilai D_x dan D_y dapat diubah sesuai dengan panjang masking. Pada rumus diatas, masking yang digunakan sebesar 3 sehingga maskingnya dapat dilihat pada Persamaan 2.4. Proses selanjutnya adalah menghitung hasil masking dari citra terhadap x dan $y(I_x, I_y)$

$$I_x = Im \cdot D_x \qquad I_y = Im \cdot D_y$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

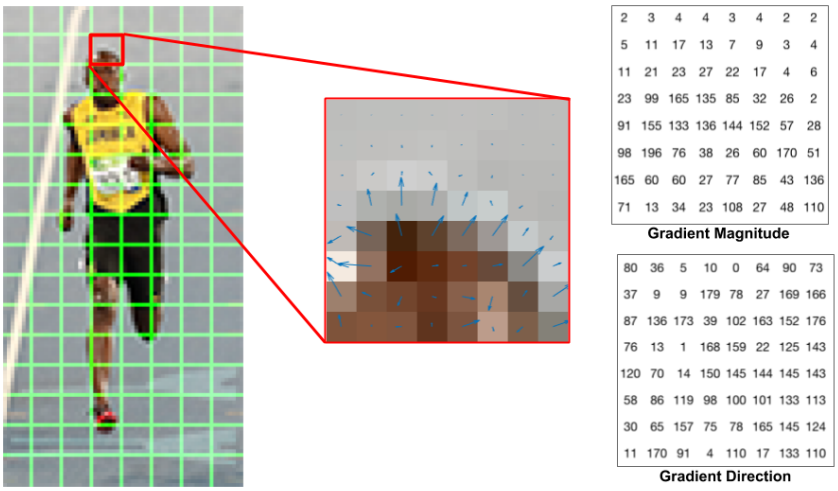
Nilai Im merupakan matriks dari suatu gambar. Selanjutnya menghitung besar gradiennya ($|G|$) dengan menggunakan Persamaan 2.5.

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{2.5}$$

Dan, orientasi gradiennya (λ) dapat dilihat pada Persamaan 2.6.

$$\lambda = tan_2^{-1}\left(\frac{I_y}{I_x}\right) \tag{2.6}$$

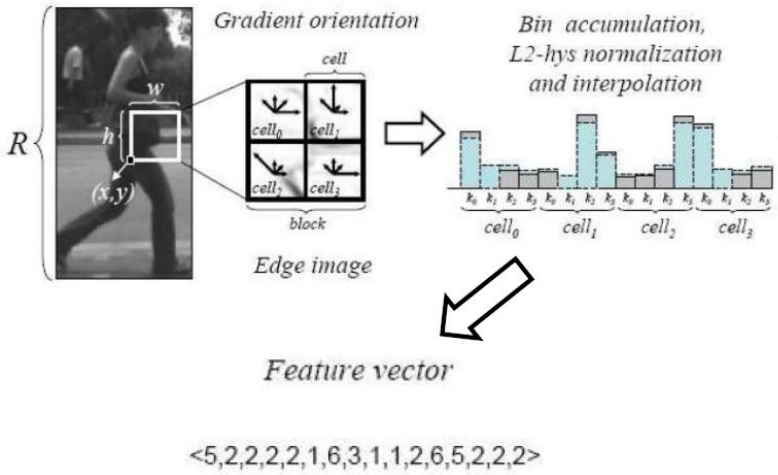
Selanjutnya menghitung histogram dari orientasi gradient tiap blok. Ukuran dari blok tersebut ditentukan oleh pengguna dengan nilai ukuran defaultnya adalah 3 x 3 dimana blok merupakan sebuah potongan gambar yang dapat dilihat pada Gambar 2.11 dengan blok didefinisikan dengan kotak berwarna hijau. Tiap blok akan dilakukan suatu perhitungan *Gradient Magnitude* ($|G|$) dan *Gradient Orientation* (λ). Selanjutnya dilakukan normalisasi pada blok untuk mengurangi efek perubahan kecerahan objek pada suatu blok (Berkant , 2015).



Gambar 2.11 Pengambilan Matriks Gradien

(Vázquez , 2014)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 2.12 HOG Descriptor

(Mallick , 2016)

Sebuah block pada HOG descriptor memiliki 2 x 2 cell untuk dilakukan perhitungannya. Jika sebuah cell berisi suatu matriks 8x8, maka jumlah matriks pada block adalah 16 x 16. dalam satu block akan dihitung vektor bin yang jika satu block berisi 2 x 2 cell, akan menghasilkan matriks vektor bin sebanyak 16 nilai dengan menggunakan 4 orientasi yaitu 0°, 45°, 90^0, dan 135°. Simulasi perhitungan vektor bin dapat dilihat pada Gambar 2.12. Nilai HOG harus dilakukan normalisasi dengan menggunakan Persamaan 2.7.

$$No = \frac{Nilai\ bin}{||nilai\ bin|| + 0.01}$$

2.10. *Extreme Learning Machine*(ELM)

ELM merupakan metode klasifikasi yang diperkenalkan oleh Huang pada tahun 2004 dan diterbitkan pada 2006. Metode ini terdiri dari 3 layer neural network dengan fungsi aktivasi sigmoid. ELM memiliki banyak kelebihan dibandingkan dengan metode berbasis gradien konvensional. Kelebihan tersebut

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

diantaranya ELM dapat meminimalisirkan jumlah iterasi, megoptimalkan nilai *learning rate*, proses *learning* yang cepat, memiliki hasil klasifikasi yang lebih baik dari *backpropagation* dalam banyak kasus, dan cocok untuk banyak fungsi aktivasi nonlinear (Ismaeel, Miri, dan Chourishi , 2015). Proses ELM adalah dengan menggunakan jaringan *feed-fordward neural network* dengan *single hidden layer* (SLFNs). Lapisan atau *layer* pada ELM terdiri dari layer *input, hidden layer,* dan layer *output* (Albadra dan Tiun , 2017).

2.10.1. *Feedforward Algorithm*

Algoritma *feedforward* merupakan algoritma yang memiliki beberapa jaringan(*multilayer network*) dengan proses yang hanya melalui propagaasi maju tanpa menggunakan propagasi mundur melalui beberapa *neuron* sebelum menuju perhitungan akhir dari suatu *output*. Algoritma *feedforward* bertujuan untuk mengenali pola suatu data sehingga dalam proses identfikasi atau klasifikasi memiliki hasil yang akurat. Proses perhitungan pada *feedforwad* akan mencari bobot terbaik dengan menggunakan pola dari suatu fungsi aktifasi. Setiap lapisan atau hidden layer akan dilakukan penjumlahan bobot dan menerapkan fungsi aktifasi. Fungsi aktifasi yang digunakan adalah *sigmoid*. Proses pada tiap layer akan digunakan untuk mengoptimalkan suatu bobot sehingga hasil keluaran dari algoritma *feedforward* adalah untuk mendaptkan bobot paling optimal dalam mengidentifikasi kedua kelas. Dalam perhitungan penjumlahan bobot pada tiap layer dapat digunakan Persamaan 2.7(Pandjaitan , 2007).

$$S_j = \Sigma_{i=0}^{n} a_i w_{j,i} \tag{2.7}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengan:

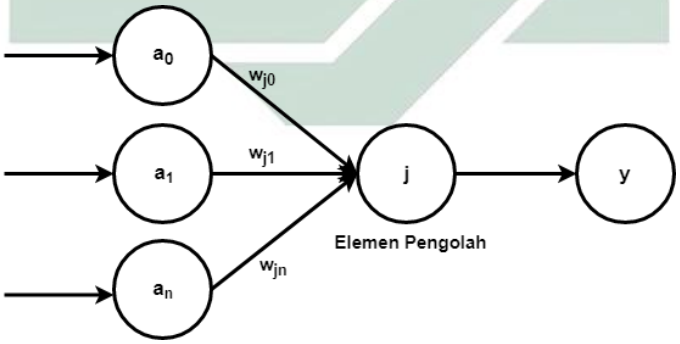
a_i = input yang berasal dari layer ke-i

$w_{j,i}$ = Bobot yang menghubugkan antara layer ke-i mrnuju layer ke-j

Setelah mendapatkan nilai S_j, maka dilakukan perhitugan untuk memperoleh $f(S_j)$ dengan menggunakan fungsi *sigmoid* dengan persamaan *sigmoid* yang dapat dilihat pada Persamaan 2.8.

$$f(S_j) = \frac{1}{1 + e^{-S_j}} \tag{2.8}$$

Sebuah arsitektur pembentukan algoritma *feedforward* dapat dilihat melalui Gambar2.13. Dimana pada gambar tersebut terlihat bahwa suatu input pada a_i akan diproses pada j yang didalamnya berisi tentang penjumlahan bobot dan pembentukan fungsi dengan menggunakan fungsi aktifasi sigmoid. Setelah proses pada j terselesaikan, maka akan menghasilkan suatu output y_i dimana berisi tentang informasi hasil dari proses klasifikasi.



Gambar 2.13 Arsitektur *Feedforward*

Pada algotima *feed forward*, semakin banyak jumlah *hidden layer* akan memperbaiki tingakt keberhasilan metode ini. Namun, jika *hidden layer* terlalu banyak akan menimbulkan permasalahan yakni pembaharuan bobot yang sudah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

optimal sehingga membuat bobot tersebut menjadi tidak optimal kembali(Pandjaitan , 2007).

2.10.2. *Single Hidden Layer Feedforward Neural Network*(SLFNs)

Fungsi keluaran SLFN dengan L merupakan banyak *hidden nodes* dimana x vektor input dan dituliskan pada Persamaan 2.9.

$$f_L(x) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x), x \in \mathbb{R}^d, \beta_i \in \mathbb{R}^m \tag{2.9}$$

Dimana $G(a_i, b_i, x)$ dinotasikan sebagai fungsi *output* dari *hidden nodes* ke-i, dimana (a_i, b_i) merupakan parameter *hidden nodes* dan β_i merupakan vektor bobot yang menghubungkan *hidden node* ke-i dengan *output node*(Huang, Zhu, dan Siew , 2006). Untuk node aditif dengan fungsi aktivasi g dapat dinyatakan pada Persamaan 2.10.

$$G(a_i, b_i, x) = g(a_i \cdot x + b_i), a_i \in \mathbb{R}^d, b_i \in \mathbb{R} \tag{2.10}$$

Jika jumlah *input* data sebanyak *n* dengan jumlah *hidden layer* yang diinisialisasi sebanyak *m*, maka dapat disusun menjadi sebuah matriks *H* yang berisi *output* dari hasil perhitungan pada *hidden layer* yang berukuran m x n sehingga matriks yang terbentuk dapat dilihat pada Persamaan 2.11.

$$H = \begin{bmatrix} g(x_1 \cdot w_1, 1 + b_1) & \cdots & g(x_1 \cdot w_1, m + b_m) \\ \vdots & \ddots & \vdots \\ g(x_n \cdot w_n, 1 + b_1) & \cdots & g(x_n \cdot w_n, m + b_m) \end{bmatrix} \tag{2.11}$$

SLFN dengan L *hidden node* dapat mendekati N sampel yang berbeda

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

secara acak dan target $(x_i, t_i) \in \mathbb{R}^d, x \in \mathbb{R}^m$ dapat diselesaikan dengan Persamaan 2.12.

$$L_p = min\ \Sigma\|\beta \cdot h(x_i) - t_i\|^2 \text{ dan } min\ \|\beta\| \tag{2.12}$$

Dimana β merupakan bobot *output*, $h(x_i)$ adalah *hidden layer* dan t_i adalah target label. Berdasarkan teori dasar dai optimasi, masalah optimasi tersebut dapat diselesaikan dengan *simple constrained optimization* sehingga menjadi:

$$\begin{aligned} minL_p &= \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\sum_{i=1}^{N}\xi_i^2\ s.\ t.\ h(x)\beta = t_i - \xi_i \\ i &= 1, 2, \ldots, N \end{aligned}$$

dimana ξ_i mendefinisikan *training error*, *C* merupakan *penalty parameter*, dan *N* adalah banyak sampel dari data. Berdasarkan teori KKT(Krush-Kuhn-Tucker), fungsi tersebut ekuivalen untuk diselesaikan dengan menggunakan permasalahan optimalisasi dual sehingga dihasilkan sebuah Persamaan 2.13

$$L_{p(ELM)} = \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i(h(x_i)^T\beta - t_i + \xi_i) \tag{2.13}$$

Dimana operator lagrangian dari α_i korespondensi dengan data sampel ke-*i*. Kondisi optimal dengan menggunaan teorema KKT didapatkan:

$$\frac{\partial L_{p(ELM)}}{\partial \beta} = 0 \Rightarrow \beta = \sum_{i=1}^{N}\alpha_i(h(x_i))^T = H^T\ \alpha \tag{2.14}$$

$$\frac{\partial L_{p(ELM)}}{\partial \xi_i} = 0 \Rightarrow C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i = 0\ \ i = 1, 2, \ldots, N \tag{2.15}$$

$$\frac{\partial L_{p(ELM)}}{\partial \alpha_i} = 0 \Rightarrow \sum_{i=1}^{N}h(x_i)\beta - t_i + \xi_i = 0\ \ i = 1, 2, \ldots, N \tag{2.16}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Berdasarkan Persamaan 2.9 dan $f_L(x)$ dapat ditulis sebagai y dan $G(a_i, b_i, x)$ dapat ditulis sebagai H sehingga diperoleh:

$$y = H\,\beta \tag{2.17}$$

Subtitusikan hasil differensial pada Persamaan 2.14 pada Persamaan 2.17 sehingga diperoleh:

$$\begin{aligned} y &= H(H^T\,\alpha) && (2.18)\\ y &= (H^T\,H)\,\alpha && (2.19)\\ (H^T\,H)^{-1}\,T &= (H^T\,H)\,(H^T\,H)^{-1}\,\alpha && (2.20)\\ (H^T\,H)^{-1}\,T &= I\,\alpha && (2.21)\\ (H^T\,H)^{-1}\,T &= \alpha && (2.22) \end{aligned}$$

Kemudian masukkan kembali nilai α yang didapat pada Persamaan 2.22 kedalam Persamaan 2.14 dengan variabel y diganti dengan T sebagai target kelas sehingga diperoleh:

$$\begin{aligned} \beta &= H^T\,\alpha && (2.23)\\ \beta &= H^T\,(H^T\,H)^{-1}\,T && (2.24)\\ \beta &= H^T\,(H^T\,H)^{-1}\,T && (2.25)\\ \beta &= H^\dagger\,T && (2.26) \end{aligned}$$

Dimana $H^\dagger$ merupakan invers dari matriks *Moore-Penrose* H.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.10.3. Algoritma ELM

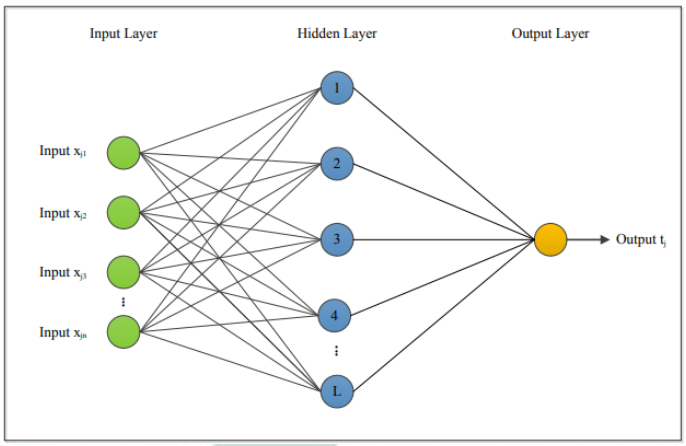
Huang et. al. (Huang, Zhu, dan Siew , 2006) mengatakan bahwa terdapat beberapa langkah-langkah untuk menyelesaikan metode ELM. *Input* dari metode ELM sendiri merupakan data *training*, data *testing*, fungsi aktifasi *g(x)*, dan jumlah *hidden layer*. sehingga proses ELM dapat dijabarkan sebagai berikut:

1. Menentukan vektor bobot w_j secara random dan bias b_j sebagai faktor pengaruh pada *hidden layer* ke-j.
2. Menghitung atau mendefinisikan matriks keluaran yang berupa H_{nXm} pada data *training* dengan menggunakan Persamaan 2.11.
3. Menghitung bobot keluaran β pada data *training* dengan menggunakan Persamaan 2.26.
4. Menghitung keluaran klasifikasi dengan menggunakan nilai β yang dihasilkan pada proses *training* dengan menggunakan data *testing*.

2.10.4. Struktur Jaringan Syaraf Tiruan ELM

Diberikan sebuah jaringan syaraf tiruan ELM dengan jumlah variabel *input* *x* sebanyak *n* dan jumlah *neuron* pada *hidden layer* sebanyak *m*. Misalkan $x = [x_1, x_2, x_3, \ldots, x_n]$ dengan x_i merupakan *input* data untuk dilakukan proses ELM, sehingga dapat dilakukan proses penjumlahan bobot dan fungsi aktifasi *g(x)* dengan menghasilkan matriks *H* dengan ukuran $n\ x\ m$. Dari hasil pada *hidden layer* akan dilakukan perhitungan yang sama untuk menghasilkan matriks *output T* dengan ukuran matriks *H* adalah $m\ x\ 1$ karena nilai *output* hanya memiliki satu *nodes* saja(Pandjaitan , 2007)(Huang, Zhu, dan Siew , 2006). Arsitektur ELM dapat dilihat pada Gambar 2.14.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 2.14 Arsitektur ELM

(Huang, Zhu, dan Siew , 2006)

2.11. *Modified ELM*

Pada dasarnya metode ELM merupakan sebuah metode sederhana dari *neural network* sehingga masih memiliki beberapa kelemahan dalam keakuratan pengklasifikasian suatu sistem. Untuk mengatasi masalah tersebut, maka dikembangkan metode ELM berdasarkan beberapa kebutuhan dalam perkembangan zaman. Pengembangan metode ELM diharapkan memiliki peran penting dalam dunia pendidikan untuk menghindari hilangnya suatu metode lama karena penambahan suatu metode baru. Beberapa metode pengembangan ELM adalah *Kernel Extreme Learning Machine*(KELM), *Extreme Learning Machine Auto Encoder*(ELM-AE), dan *Multilayer Extreme Learning Machine*(MLELM).

2.11.1. *Kernel Extreme Learning Machine*(KELM)

Pada dasarnya kernel berfungsi untuk memetakan data menuju ke dimensi yang lebih tinggi menggunakan fungsi pemetaan $\theta(x)$, dengan perkalian fungsinya

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

yaitu:

$$(\theta(x_i) \cdot \theta(x_j)) = K(x_i, x_j)$$

Kernel $K(x_i, x_j)$ merupakan fungsi yang memenuhi:

$$\int\int K(x_i, x_j) f(x_i) f(x_j) dx_i dx_j \geq 0, \forall f(x) \in \mathbb{R}^2$$

Contoh sebuah kernel dasar yang biasa digunakan pada metode kecerdasan buatan SVM adalah: (Ahuja dan Kumar Yadav , 2012)(Adarsh dan Jeyakumari , 2013)

$$\begin{aligned} \text{Polinomial} &: K(x_i, x_j) = (x_i \cdot x_j + 1)^p \\ \text{Gaussian} &: K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{(2\sigma^2)}) \\ \text{Linear} &: K(x_i, x_j) = x_i \cdot x_j \end{aligned}$$

Berdasarkan metode SVM yang mendapatkan hasil klasifikasi yang lebih bagus ketika menerapkan metode kernel, maka ELM mencoba mengkombinasikan dengan menggunakan metode kernel. Metode penggabungan ini disebut dengan metode Kernel Extreme Learning Machine(KELM). Diasumsikan HH^T tidak singular, sehingga penambahan koefisien $\frac{I}{C}$ dimana I merupakan matriks identitas pada persamaan HH^T didalam perhitungan bobot β. Sehingga didapapatkan perubahan pada persamaan T menjadi $\dfrac{I}{C} + HH^T$ dan menjadikan perubahan persamaan ELM menjadi(Wang, dan Han , 2014)(Deng, Ong, dan Zheng , 2016)

$$(\frac{I}{C} + HH^T)\alpha = T$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sehingga didapatkan nilai α adalah

$$\alpha = (\frac{1}{C} + HH^T)^{-1}\, T$$

sehingga persamaan bobot β berubah yang awalnya $\beta = H^T\, \alpha$ menjadi Persamaan 2.27

$$\beta = (\frac{1}{C} + HH^T)^{-1}\, H^T\, T \tag{2.27}$$

sehingga KELM dapat ditulis menjadi:

$$\begin{aligned} \Omega_{ELM} &= H^T H \\ \Omega_{ELM_{i,j}} &= h(x_i) \cdot h(x_j) = K(x_i, x_j) \end{aligned}$$

Lambang Ω tanda kernel pada ELM. Fungsi keluaran dari KELM juga dapat ditulis menjadi Persamaan 2.28.

$$\beta = h(x)H^T(\frac{I}{C} + H^T H)^{-1}T \tag{2.28}$$

$$= \begin{bmatrix} K(x \cdot x_1) \\ \vdots \\ K(x \cdot x_N) \end{bmatrix} (\frac{I}{C} + H^T H)^{-1}T \tag{2.29}$$

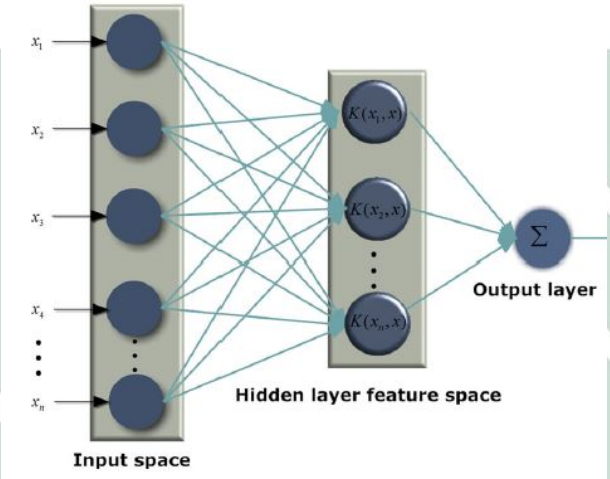
Pada KELM fungsi kernel yang biasa digunakan adalah kernel yang sama seperti digunakan pada metode SVM, Namun yang lebih sering dipakai pada KELM adalah *Gaussian kernel*. Terdapat kernel lain yang dapat dipakai yaitu wavelet kernel dapat

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dilihat pada Persamaan 2.30(Li, Rong, dan Li , 2014)

$$K(x_i, x_j) = cos(a\frac{\|x_i - x_j\|}{b})exp(-\frac{\|x_i - x_j\|^2}{c}) \tag{2.30}$$

dengan a, b, c adalah suatu parameter yang dapat diubah dan diperbaiki berdasarkan kondisi yang diinginkan. Arsitektur dari KELM dapat dilihat pada Gambar 2.15.



Gambar 2.15 Arsitektur KELM

(Zhao, Huang, Wei, et. al. , 2016)

Pada gambar tersebut terlihat bahwa proses penambahan kernel berada pada *hidden layer* ELM(Huang,Li,Lin,Huang, Zong, et. al. , 2013).

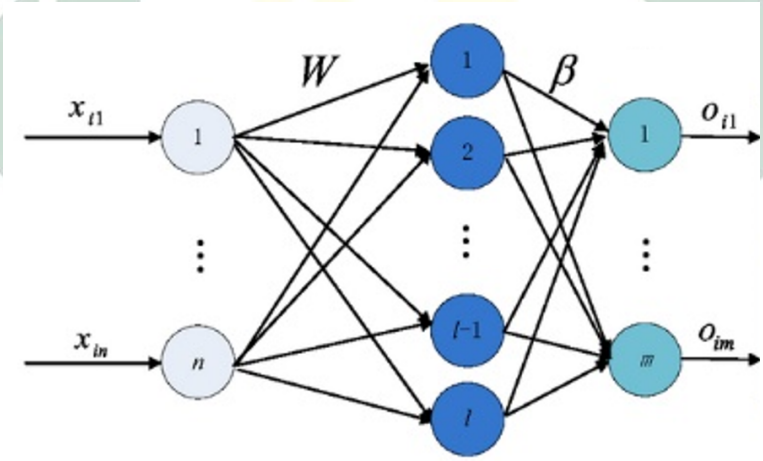
2.11.2. *Extreme Learning Machine Auto Encoder*(ELM-AE)

Auto Encoder (AE) merupakan sirkuit pembelajaran yang didasarkan pada paradigma *encoder-decoder*, dimana *encoder* mengubah *input* menjadi representasi dimensi lebih rendah dan *decoder* merekonstruksi input awal dari representasi *encoder* melalui fungsi minimalisasi. AE dilatih tanpa pengawasan yang memungkinkan penggalian fitur-fitur penting dari data yang tak berlabel. AE dan metode pembelajaran tanpa pengawasan telah banyak digunakan dalam

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

penyelsaian masalah pra-pelatihan jaringan, ekstraksi fitur, pengurangan dimensi, dan pengelompokkan (Wang, Zhang, Chai, et. al. , 2018).

AE klasik hanya memiliki satu lapisan tersembunyi sebagai proses representasi dimensi yang lebih rendah dari inputan. Dalam dekade terakhir, AE banyak dikombinasikan dalam mesin pembelajaran cerdas maupun mendalam karena keunggulannya dalam hal kompleksitas komputasi serta akurasi klasifikasi (Mao, He, Li, dan Yan, 2016). Kombinasi ELM-AE merupakan algoritma pembelajaran baru tanpa pengawasan yang terdiri dari satu lapisan tersembunyi dengan n_h simpul, n_i simpul lapisan *input*, dan n simpul lapisan *output* seperti pada Gambar 2.16(Sun, Zhang, Zhang, dan Hu , 2016).



Gambar 2.16 Ilustrasi dari Struktur ELM-AE

(Chen, Bin, Lu, Zhang dan Liang , 2016)

ELM-AE dan ELM memiliki banyak kemiripan, namun terdapat dua perbedaan diantara keduanya, yaitu :

1. ELM adalah jaringan saraf terawasi dengan output label kelas, sedangkan ELM-AE adalah jaringan tanpa pengawasan dengan output yang sama seperti inputannya.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2. Dalam kasus ELM, *input* dari bobot dan bias dari lapisan tersembunyi merupakan hasil dari pengacakan nilai, sedangkan dalam ELM-AE keduanya adalah ortogonal (Roul, Rohan, dan Gaurav , 2016).

Seperti yang ditunjukkan pada Gambar 2.16, ide dasar pelatihan ELM-AE merupakan proses dua tahap. Tahap pertama sejumlah *neuron* tersembunyi memetakan data asli ke dalam fitur ruang dimensi n_h. Menurut ukuran n_i dan n_h ELM-AE memiliki tiga representasi berikut :

1. *Compressed*, menampilkan fitur dari ruang data *input* dimensi yang lebih tinggi ke dimensi yang lebih rendah ($n_i > n_h$).
2. *Sparse*, menampilkkan fitur dari ruang data *input* dimensi yang rendah ke dimensi yang lebih tinggi ($n_i < n_h$).
3. *Equal*, menampilkan fitur dari dimensi nruang data *input* sama dengan dimensi ruang fitur ($n_i = n_h$) (Roul, Rohan, dan Gaurav , 2016).

Dengan n_i adalah jumlah unit *input* dan n_h adalah jumlah unit lapisan tersembunyi.

Dalam ELM-AE, berlaku lemma Johnson-Lindenstrauss yang berbunyi suatu nilai acak yang bersifat orthogonal dapat diproyeksikan ke ruang dimensi yang sama atau lebih tinggi sehingga bobot acak dan bias ortogonal dari node tersembunyi memproyeksikan data *input* ke ruang dimensi yang berbeda atau sama, seperti yang ditunjukkan oleh lemma Johnson-Lindenstrauss dan dihitung menggunakan Persamaan 2.31.

$$h(x_i) = g\left(x_i a + b\right) \tag{2.31}$$

dengan $a^T a = I, b^T b = I$. Dimana,x_i adalah data *input*, $h(x_i) \in \mathbb{R}^{n_h}$ adalah vektor

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

keluaran dari lapisan tersembunyi, $g(.)$ adalah fungsi aktivasi yang dapat berupa fungsi sigmoid, gaussian, dan sebagainya. I adalah matriks identitas urutan n_h, a adalah bobot *input* antara lapisan *input* dan lapisan tersembunyi dari matriks $n_i \times n_h$, dan b adalah bias dari unit tersembunyi (Chen, Bin, Lu, Zhang dan Liang , 2016).

Sedangkan untuk arsitektur *spare* ELM-AE, parameter acak ortogonal dapat dihitung menggunakan Persamaan 2.31 dengan syarat $aa^T = I, b^T b = 1$. Hasil keluaran dari jaringan diberikan pada Persamaan 2.32.

$$f(x_i) = h(x_i)^T \beta, i = 1, ..., N \tag{2.32}$$

dengan β adalah bobot keluaran antara lapisan tersembunyi dengan lapisan keluaran yang merupakan matriks $n_h \times n_i$ (Sun, Zhang, Zhang, dan Hu , 2016).

Pada tahap kedua, ELM-AE perlu memperbarui bobot keluaran β dengan mengadopsi *squared loss of the prediction error* dengan model seperti pada Persamaan 2.33.

$$\min_{\beta \in \mathbb{R}^{n_h \times n_i}} L_{ELM-AE} = \min_{\beta \in \mathbb{R}^{n_h \times n_i}} \frac{1}{2}\|\beta\|^2 + \frac{C}{2}\|X - H\beta\|^2 \tag{2.33}$$

dimana istilah pertama adalah istilah regulasi yang mengontrol kompleksitas model, dan C adalah koefisian penalti pada *error* pelatihan. Nilai X diasumsikan $X = \left[x_1^T, x_2^T, ..., x_N^T\right]^T \in \mathbb{R}^{N \times n_i}$, $H = \left[h(x_1)^T, h(x_2)^T, ..., h(x_N)^T\right]^T \in \mathbb{R}^{N \times n_h}$ (Mao, He, Li, dan Yan, 2016).

Dengan mengatur gradien L_{ELM-AE} terhadap β menjadi nol, maka didapat Persamaan 2.34.

$$\bigtriangledown_{ELM-AE} = \beta - CH^T (X - H\beta) = 0 \tag{2.34}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Persamaan gradien 2.34 mudah dipecahkan dan solusinya dibahas dalam dua selanjutnya (Sun, Zhang, Zhang, dan Hu , 2016). Ketika jumlah sampel pelatihan N lebih besar dari jumlah *neuron* tersembunyi n_h, solusi yang didapat seperti pada Persamaan 2.35.

$$\beta^* = \left(H^T H + \frac{I_{n_h}}{C}\right)^{-1} H^T X \tag{2.35}$$

dimana I_{n_h} adalah matriks identitas dengan dimensi n_h (elm-ae3).

Jika jumlah sampel pelatihan N lebih kecil dari jumlah *neuron* tersembunyi n_h, H akan memiliki lebih banyak kolom daripada baris. Dalam hal ini, diperkenalkan batasan tambahan untuk $\beta = H^T \alpha \left(\alpha \in \mathbb{R}^{N \times n_i}\right)$. Maka solusi yang didapat seperti pada Persamaan 2.36.

$$\beta^* = H^T \left(H H^T + \frac{I_N}{C}\right)^{-1} X \tag{2.36}$$

dimana I_N adalah matriks identitas dengan dimensi N (Sun, Zhang, Zhang, dan Hu , 2016). Untuk representasi ELM-AE dengan dimensi yang smaa, bobot keluaran dihitung menggunakan Persamaan 2.37.

$$\beta^* = H^{\dagger} X \tag{2.37}$$

dimana $\beta^T \beta = I$ (Mao, He, Li, dan Yan, 2016). Diberikan data input X, dapat diperoleh representasi dalam ruang dimensi n_h sebagai $X_{new} = X\beta^T$. Selanjutnya X_{new} dapat digunakan untuk mengganti data asli dalam proses pengelompokkan atau klasifikasi (Sun, Zhang, Zhang, dan Hu , 2016). Berdasarkan penjelasan di atas, langkah-langkah utama ELM-AE adalah sebagai berikut :

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1. *Input* data $X = \{x_i\}_{i=1}^{N}$, jumlah *neuron* tersembunyi n_h, dan koefisien penalti C.

2. *Output* berupa bobot β dan *output* lapisan tersembunyi.

3. Langkah 1, bangkitkan bobot a dan bias b masukan secara acak dengan metode ortogonal pada Persamaan 2.31.

4. Hitung *output* lapisan tersembunyi H.

5. Hitung bobot keluaran β, jika $n_i \geq n_h$ digunakan Persamaan 2.35, jika $n_i \leq n_h$ digunakan Persamaan 2.36, sedangkan jika $n_i = n_h$ digunakan Persamaan 2.37.

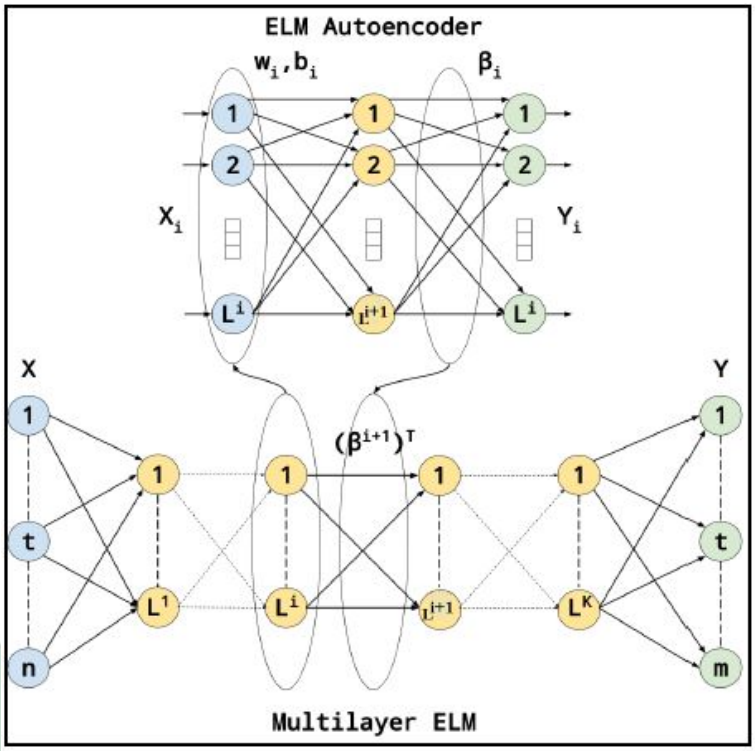
6. Output berupa X_{new} dengan Persamaan $X_{new} = X\beta^T$ (Roul, Rohan, dan Gaurav , 2016).

2.11.3. *Multilayer Extreme Learning Machine*(MLELM)

Extreme Learning Machine merupakan sebuah metode *neural network* dengan *single hidden layer* untuk proses klasifikasinya. Hinton dan Salakhutdinov melakukan modifikasi terhadap ELM sehingga ELM kini dapat digunakan dalam beberapa *hidden layer* (Duan,Bao,Miao Xu, dan Chen , 2016). Metode ini diberikan sebuah nama Multilayer Extreme Learning Machine(MLELM). MLELM membentuk suatu arsitektur dengan mengkombinasikan metode ELM dengan metode *Extreme Learning Machine Auto Encoder*(ELM-AE). Dengan memanfaatkan ELM sebagai metode dasar, maka diperoleh sebuah keuntungan dari keakuratan suatu metode *multilayer perceptron* dan dari kecepatan learning ELM sehingga MLELM menjadi suatu metode *multilayer perceptron* yang memiliki kecepatan learning yang paling cepat. MLELM memiliki suatu struktur atau biasa dikenl dengan sebutan arsitektur yang dapat dilihat pada Gambar

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.17(Le dan Xiao , 2018).



Gambar 2.17 Ilustrasi dari Struktur MLELM

(Roul , 2005)

Multilayer Extreme Learning Machine memiliki beberapa karakteristik yang tidak dimiliki oleh metode klasifikasi lainnya yaitu(Roul, Rohan, dan Gaurav , 2016)

1. Memiliki proses *unsupervised learning* dan *supervised learning* didalam proses trainingnya.
2. Tidak memiliki iterasi atau perulangan dalam proses trainingnya.
3. Memiliki waktu yang relatif cepat pada proses *training* sehingga dikatakan sebagai metode *multilayer neural network* tercepat.
4. Berbasis dari beberapa metode ELM-AE yang ditumpuk untuk menjadi suatu *multilayer neural network*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

5. Parameter ELM dilakukan proses *training* dengan menggunakan metode ELM-AE.

Beberapa modifikasi yang dilakukan pada MLELM sehingga memiliki perbedaan yang mencolok dengan *Basic ELM* yaitu memiliki hidden layer yang lebih dari satu dengan jumlah node yang sama di semua hidden layer untuk dilakukan prose ELM-AE. Persamaan umum yang dipakai dalam *multilayer ELM* adalah bagaimana dari *input* akan dilakukan proses menuju ke *hidden layer* dan dari *hidden layer* menuju *hidden layer* lainnya sampai pada *hidden layer* ke-$(n-1)$(Wen, Liu, Yan, dan Sun , 2018). Persamaan umum dari MLELM dapat dilihat pada Persamaan 2.38.

$$H^n = g((\beta^n)^T H^{n-1}) \tag{2.38}$$

Sehingga dari persamaan tersebut dapat disimpulkan bahwa langkah-langkah pengerjaan MLELM dapat dideskripsikan sebagai berikut:

1. Definisikan sebuah input X_i.
2. Bangkitkan bobot a dan bias b masukan secara acak dengan metode ortogonal pada Persamaan 2.31.
3. Hitung *output* lapisan tersembunyi H^n pada Persamaan 2.38.
4. Hitung bobot keluaran β, jika $n_i \geq n_h$ digunakan Persamaan 2.35, jika $n_i \leq n_h$ digunakan Persamaan 2.36, sedangkan jika $n_i = n_h$ digunakan Persamaan 2.37.
5. Output berupa X_{new} dengan Persamaan $X_{new} = X\beta^T$.
6. Ulangi langkah 3 sampai mencapai *hidden layer* ke-$(n-1)$.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

7. Hasil output H^{n-1} akan dihitung dengan menggunakan metode *basic ELM* untuk mendapatkan nilai y yang merupakan hasil klasifikasi.

2.12. *Deep Extreme Learning Machine* (DELM)

Deep Extreme Learning Machine(DELM) merupakan suatu metode pengembangan dari ELM yang digabungkan dengan *Deep Learning*. Metode DELM sendiri merupakan metode penggabungan dari *basic ELM, ELM Auto Encoder, Multilayer ELM,* dan juga *Kernel ELM*. Pada DELM fungsi aktifasi pada *hidden layer* dapat berupa sebuah fungsi linear maupun non-linear. Fungsi aktifasi yang dapat dipakai pada DELM dapat dilihat pada Gambar 2.18(Dongping , 2018)(Fayaz dan Kim , 2018).

Function names	Formulations
Sigmoid	$f(x) = 1/\left(1 + e^{-x}\right)$
Gaussian (Radial basis function)	$f(x) = e^{-x^2}$
Rectified linear unit (ReLu)	$f(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases}$
TanH	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Hardlim	$f(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$
Sine	$f(x) = \sin(x)$
Triangular basis function	$f(x) = \begin{cases} 1 - \lvert x \rvert, -1 \leq x \leq 1 \\ 0, otherwise \end{cases}$

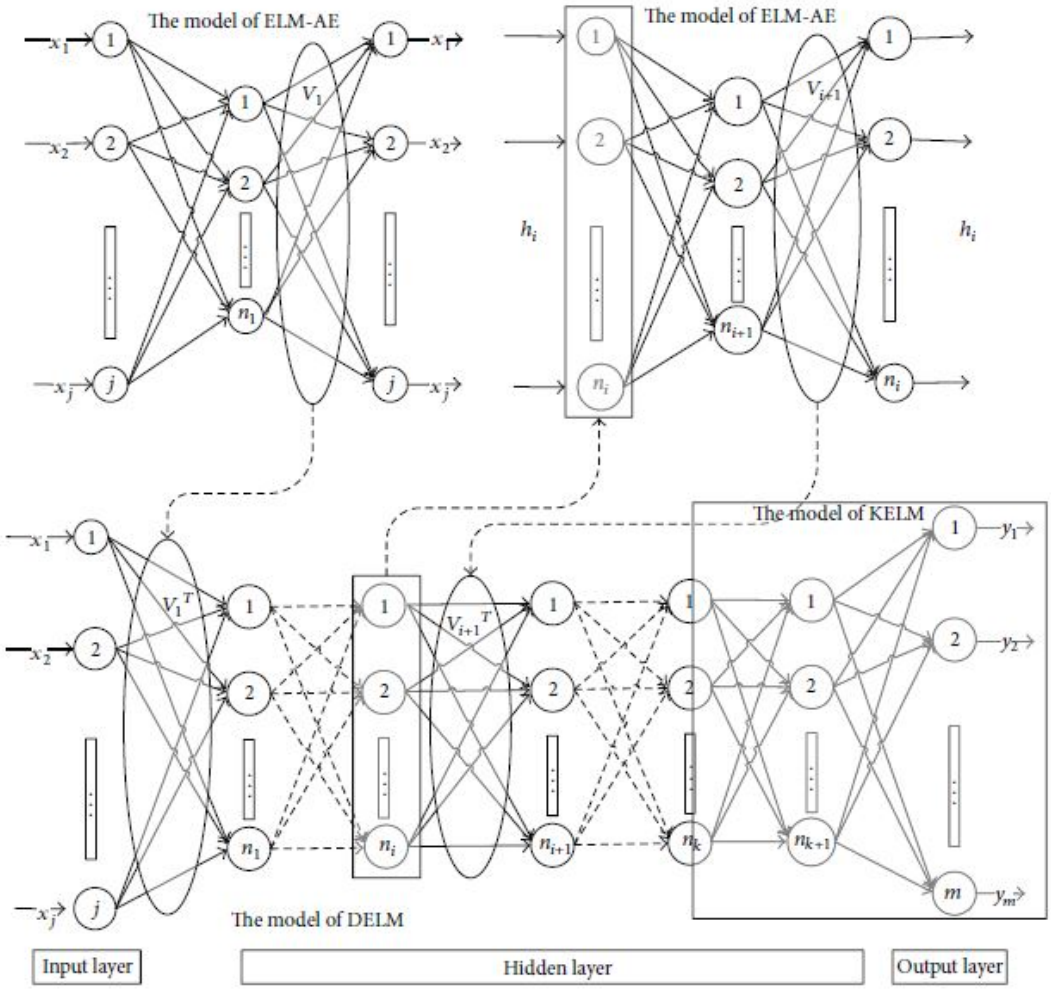
Gambar 2.18 Fungsi Aktifasi

(Gao, Ma, She, dan Dong , 2018)

Dalam DELM ddapat ditambahkan satu *hidden layer* yang merupakan proses KELM. Berdasarkan hal tersebut, dapat disimpulkan bahwa *output* terakhir dari MLELM berupa metode KELM. Metode KELM akan terletak pada *hidden layer* H_k (dengan ukuran matriks n_k x *N*) dan dapat dibentuk suatu fungsi kernel terhadap persamaan $H_{k+1}H_{k+1}^T$. Berdasarkan proses kombinasi dari ketiga metode

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tersebut(KELM,MLELM,ELM-AE) yang membentuk suatu metode DELM diharapkan dapat menjadi suatu metode *Deep Learning* yang memiliki proses *learning* paling cepat(Guo, Pang, Yan, dan Qiao , 2017)(Ibrahim an Abadeh , 2017). Struktur dari DELM dapat dilihat pada Gambar 2.19.



Gambar 2.19 Arsitektur DELM

(Ding, Zhang, Xu, Guo, dan Zhang , 2015)

Seperti yang dilihat pada Gambar 2.19, bahwa pada *hidden layer* pertama akan dilakukan proses basic ELM untuk menghasilkan nilai dari *hidden layer* ke-1. Selanjutnya, dilakukan proses MLELM dimana dalam MLELM tersebut terdapat proses ELM-AE pada setiap perhitungan menuju hidden layer ke-2 sampai menuju

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ke-(i-1). Pada *hidden layer* ke-(i-1) menuju ke *layer output* akan dilakukan suatu proses perhitungn dengan menggunakan metode KELM. Kernel pada DELM dapat didefinisikan dengan Persamaan 2.39(Ding, Zhang, Xu, Guo, dan Zhang , 2015).

$$\Omega_{DELM} = H_{k+1}H_{k+1}^T \tag{2.39}$$

$$\Omega_{DELM} = \begin{bmatrix} K(H_k(1,1), H_k(1,1)) & \cdots & K(H_k(1,1), H_k(1,N)) \\ \vdots & \ddots & \vdots \\ K(H_k(M,N), H_k(N,1)) & \cdots & K(H_k(M,N), H_k(M,N)) \end{bmatrix} \tag{2.40}$$

$$h_{k+1}(h_k(x))H_{k+1}^T = \begin{bmatrix} K(h_k(x), H_k(:,1)) \\ \vdots \\ K(h_k(x), H_k(:,N)) \end{bmatrix}^T \tag{2.41}$$

Sehingga persamaan basic ELM dapat berubah menjadi Persamaan 2.42.

$$\beta = h_{k+1}(h_k(x))H_{k+1}^T(\frac{I}{C} + H_{k+1}H_{k+1}^T)^{-1}T \tag{2.42}$$

Kemudian dimasukkan Persamaan 2.39 menuju Persamaan 2.42 sehingga menjadi Persamaan 2.43.

$$\beta = h_{k+1}(h_k(x))H_{k+1}^T(\frac{I}{C} + \Omega_{DELM})^{-1}T \tag{2.43}$$

Atau dapat ditulis dengan Persamaan 2.44.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$f(x) = \begin{bmatrix} K(h_k(x), H_k(:,1)) \\ \vdots \\ K(h_k(x), H_k(:,N)) \end{bmatrix}^T \cdot (\frac{I}{C} + \Omega_{DELM})^{-1} T \tag{2.44}$$

Berdasarkan perubahan ataupun modifikasi suatu persamaan ELM, maka dapat disimpulkan langkah-langkah metode DELM adalah:

1. Definisikan sebuah input X_i.
2. Bangkitkan bobot a dan bias b masukan secara acak dengan metode ortogonal pada Persamaan 2.31.
3. Hitung *output* lapisan tersembunyi H^n pada Persamaan 2.38.
4. Hitung bobot keluaran β, jika $n_i \geq n_h$ digunakan Persamaan 2.35, jika $n_i \leq n_h$ digunakan Persamaan 2.36, sedangkan jika $n_i = n_h$ digunakan Persamaan 2.37.
5. Output berupa X_{new} dengan Persamaan $X_{new} = X\beta^T$.
6. Ulangi langkah 3 sampai mencapai *hidden layer* ke-$(n-1)$.
7. Hasil output H^{n-1} akan dihitung dengan menggunakan persamaan KELM yang telah dimodifikasi menjadi Persamaan 2.44 untuk mendapatkan nilai y yang merupakan hasil klasifikasi.

2.13. *Confusion Matrix*

Confusion matrix merupakan metode evaluasi untuk sistem klasifikasi dimana dapat mengetahui jumlah data yang memiliki kesalahan klasifikasi. Klasifikasi dikatakan baik ketika sistem dapat mengklasifikasikan dengan baik dan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

memiliki error yang kecil. Metode ini membantu agar dapat mengetahui ketepatan dari suatu klasifikasi dan mengetahui error dari suatu sistem klasifikasi (Visa, Ramsay, Ralescu, dan Van Der Knaap , 2011). *Confusion matrix* menghasilkan suatu matriks yang berisi nilai TN (*True Negatif*), TP (*True Positif*), FN (*False Negatif*), dan FP (*False Positif*) yang ilustrasinya dapat dilihat pada Tabel 2.1.

Tabel 2.1 *Confusion Matrix*

		Data Prediksi	
		Positif	Negatif
Data	Positif	TP	FN
Aktual	Negatif	FP	TN

Dengan keterangan:

(a.) TP (*True Positif*) = merupakan sebuah data positif yang benar terklasifikasikan positif

(b.) FP (*False Positif*) = merupakan sebuah data negatif yang salah terklasifikasikan menjadi data positif

(c.) FN (*False Negatif*) = merupakan sebuah data positif yang salah terklasifikasikan menjadi data negatif

(d.) TN (*True Negatif*) = merupakan sebuah data negatif yang benar terklasifikasikan menjadi data negatif

Pada confusion matrix nilai yang didapatkan sebagai taraf evaluasi pengukuran keberhasilan klasifikasi adalah akurasi, sensitifitas, dan spesifisitas. Penjelasan tiap-tiap evaluasi dapat dilihat dibawah ini:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1. Akurasi

Akurasi merupakan nilai yang dapat menunjukkan ketepatan dari suatu klasifikasi dengan menghitung banyak data yang terklasifikasi dengan benar. Semakin besar nilai akurasinya maka terbukti semakin banyak data yang terklasifikasikan dengan benar (Adyanti , 2018). Persamaan akurasi dapat dilihat pada Persamaan 2.45.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.45}$$

2. Spesifisitas

Spesifisitas merupakan suatu hasil yang merepresentasikan banyak data yang benar terklasifikasikan negatif. Semakin besar nilai spesifisitasnya maka semakin besar sistem klasifikasi dapat mengklasifikassikan kelas negatif dengan baik (Adyanti , 2018). Persamaan spesifisitas dapat dilihat pada Persamaan 2.46.

$$Spesifisitas = \frac{TN}{FP + TN} \tag{2.46}$$

3. Sensitifitas

Sensitifitas merupakan suatu hasil yang merepresentasikan banyak data yang benar terklasifikasikan pada kelas positif. Jadi, semakin besar nilai sensitifitasnya maka semakin besar data yang digunakan dalam sistem klasifikasi dapat mengklasifikasikan data kelas positif dengan baik (Adyanti , 2018). Persamaan Sensitifitas dapat dilihat pada Persamaan 2.47.

$$Sensitifitas = \frac{TP}{TP + FN} \tag{2.47}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB III

METODE PENELITIAN

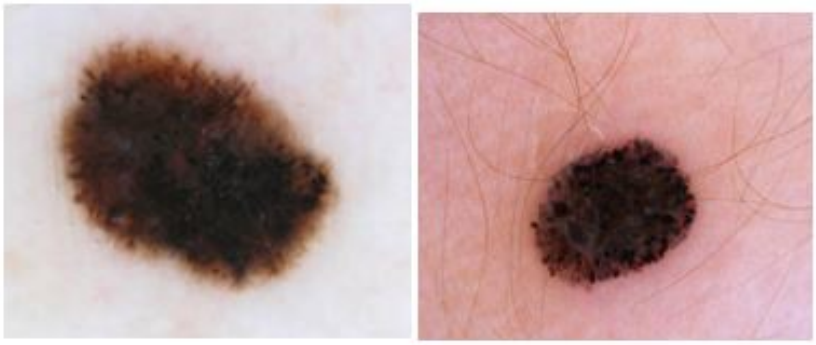
3.1. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian deskriptif kuantitatif karena dalam penelitian ini mengandung unsur perhitungan dan analisis dari hasil yang didapatkan. Penelitian ini juga termasuk dalam kategori penelitian terapan berdasarkan aspek fungsinya. Hasil penelitian memiliki tujuan sebagai alternatif untuk deteksi secara tepat dan mempercepat pemeriksaan kanker kulit. Penelitian terapan juga dapat diartikan sebagai suatu tindakan aplikatif untuk pemecahan masalah tertentu.

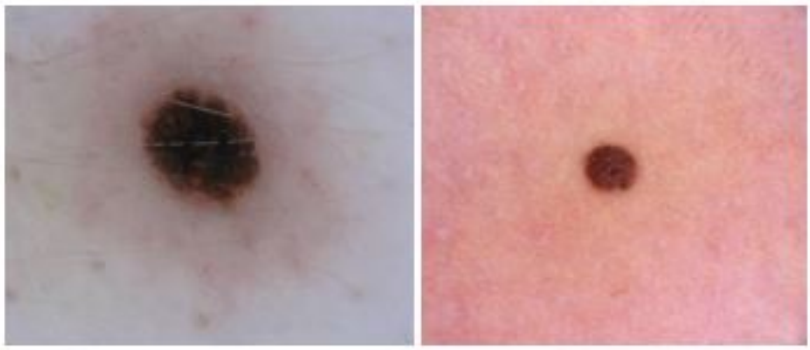
3.2. Jenis Dan Sumber Data

Jenis data yang digunakan pada penelitian ini adalah data citra kanker kulit. Data yang didapat berasal dari *International Skin Imaging Collaboration Archive*. Terdapat 2 macam data yang diambil yaitu data kanker kulit melanoma (Malignant) dan data non kanker(Benign). Data citra kanker kulit berjumlah 1000 data kanker kulit Malignant dan 1000 data non kanker atau Benign yang nantinya akan digunakan sebagai percobaan dalam pengidentifikasian penyakit kanker kulit.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 3.1 Sampel Data Kanker(Malignant)



Gambar 3.2 Sampel Data Non-Kanker(Benign)

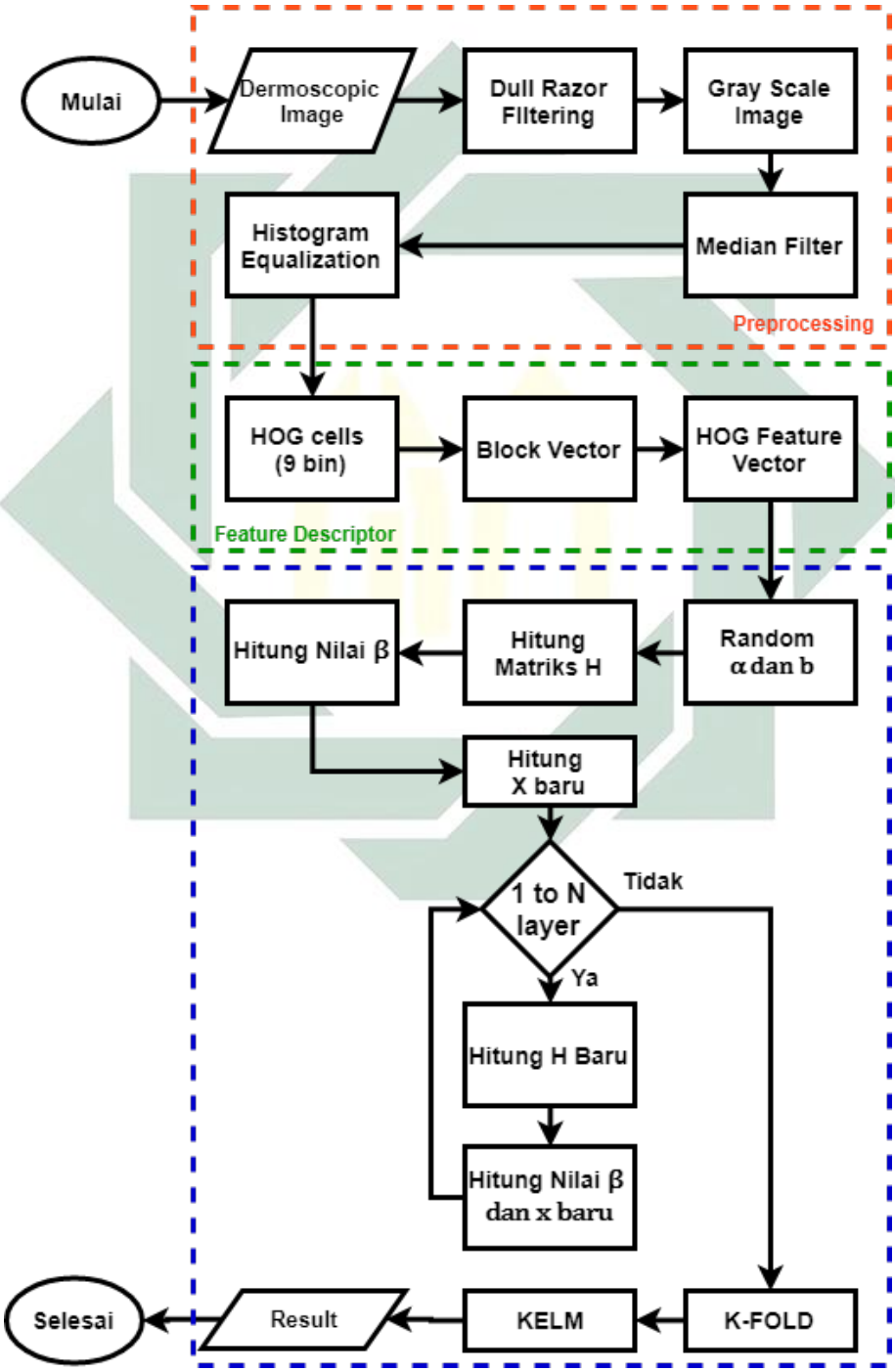
3.3. Rancangan Penelitian

Alur penelitian ini dimulai dengan mengkaji dan mencari beberapa kajian literatur mengenai deteksi kanker kulit menggunakan beberapa metode yang digunakan dalam mengklasifikasikan jenis kanker kulit maupun membedakan kanker kulit(Maignant) dengan non-kanker(Benign) menggunakan metode klasifikasi. Selanjutnya, setelah referensi dan kajian literature telah didapatkan maka proses selanjutnya merupakan pencarian data yang dibutuhkan dalam penelitian.

Data yang digunakan dapat ditemukan di website resmi *International Skin Imaging Collaboration Archive* yang berisi tentang data kanker kulit beserta labelnya. Pada data yang akan digunakan merupakan data primer sehingga perlu

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dilakukan *preprocessing* untuk mendapatkan citra yang bagus. Setelah dilakukan *preprocessing* selanjutnya adalah mengekstrasi fitur pada hasil *preprocessing* untuk mendapatkan fitur statistik dari gambar. Pada tahapan selanjutnya, fitur statistik akan digunakan sebagai input untuk proses klasifikasi.



Gambar 3.3 Diagram Alur Deteksi Kanker Kulit

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Proses klasifikasi akan menentukan seberapa besar keberhasilan dari penelitian ini. Tujuan utama dari proses klasifikasi asalah membagi data menjadi dua kelas untuk menentukan data tersebut terjangkit kanker kulit atau tidak. Setelah hasil klasifikasi didapatkan maka diperlukan proses perhitungan untuk menentukan akurasi yang didapat sehingga bisa diketahui berapa persen penelitian ini berhasil.

Alur penelitian tersebut memiliki tujuan yang akan dicapai yaitu mendapatkan hasil klasifikasi kanker kulit yang baik. Untuk mencapai tujuan tersebut maka diperlukan langkah-langkah pengerjaan yang terstruktur. Berdasarkan diagram alir tersebut bisa disimpulkan bahwa tahapan-tahapan penelitian tersebut adalah:

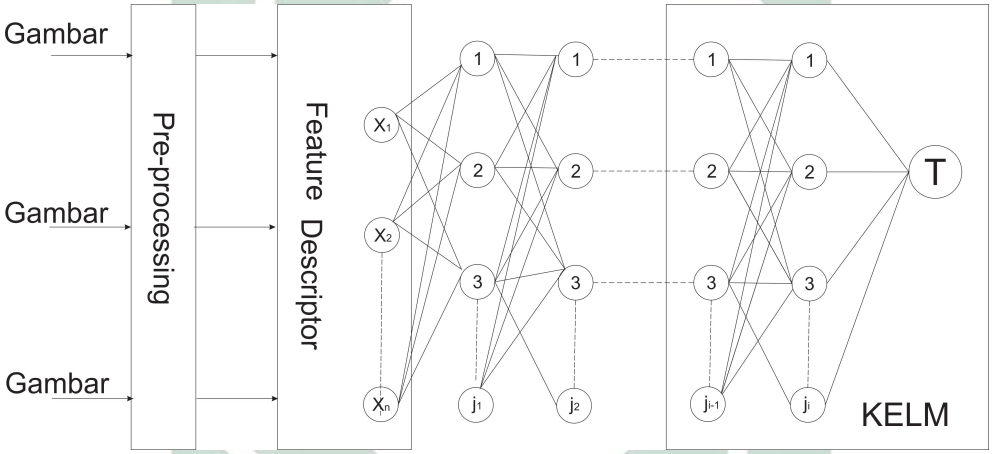
1. Mencari data kanker kulit dan non-kanker pada citra *dermoscopic*.
2. Mengubah data citra RGB menjadi grayscale pada Persamaan 2.1.
3. Menghapus fitur rambut dengan menggunakan *Dull Razor Filtering*.
4. Menghapus noise yang ada pada citra dengan menggunakan *median filter* dengan menggunakan persamaan 2.2.
5. Meningkatkan kualitas citra dengan menggunakan *histogram equalization* dengan Persamaan 2.3.
6. Melakukan *feature descriptor* dengan menggunakan metode HOG dengan menggunakan Persamaan 2.6.
7. Melakukan proses training untuk mendapatkan β output dan parameter DELM lainnya.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

8. Melakukan proses testing dengan menggunakan β output dan parameter DELM yang sudah tersedia untuk menentukan tingkat keberhasilan penelitian.

9. Melakukan proses perhitungan *confusion matrix* dengan menggunakan Persamaan 2.45, 2.46, dan 2.47.

Pada penggunaan DELM, arsitektur yang dipakai dapat dilihat pada Gambar 3.4. Percobaan yang dilakukan pada arsitektur tersebut adalah banyaknya nodes dan banyaknya hidden layer yang digunakan.



Gambar 3.4 Arsitektur DELM pada Penelitian

Percobaan pertama ada pada banyaknya nodes *j* pada suatu hidden layer yang akan dilakukan uji coba dengan jumlah nodes sama dengan jumlah variabel input, jumlah nodes lebih banyak daripada jumlah input, dan jumlah nodes lebih sedikit dari jumlah hidden nodes. Jumlah hidden layer *i* juga akan disusun dengan uji coba sebanyak 10, 25, dan 50 hidden layer. Seperti yang terlihat pada arsitektur tersebut bahwa keluaran dari DELM akan merupakan satu nilai T yang bernilai 0 jika merupakan tumor jinak dan 1 jika merupakan kanker.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB IV

HASIL DAN PEMBAHASAN

4.1. *Pre-processing* Citra

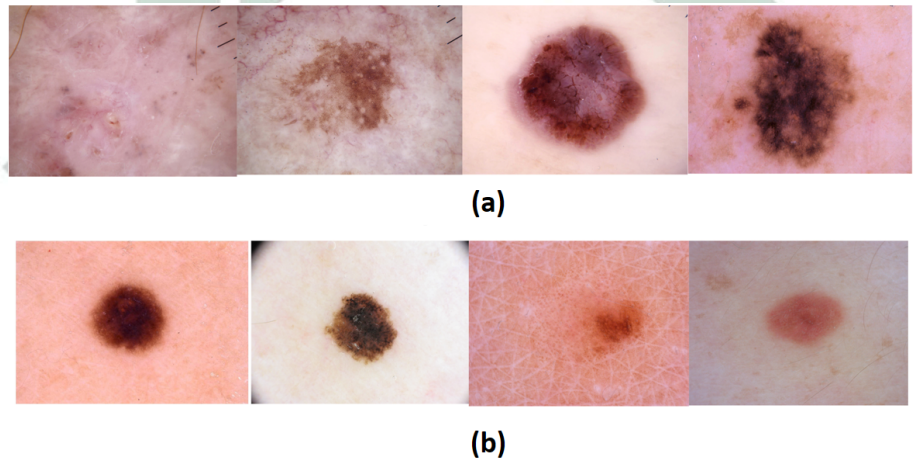
Deteksi kanker kulit dengan menggunakan CAD system harus melalui proses *pre-processing* untuk memperbaiki citra sehingga citra tersebut dapat memuat banyak informasi. Citra yang dipakai dalam klasifikasi penyakit kanker kulit adalah citra dermoscopic yang memiliki format *Portable Network Graphics* (.png) sebanyak 1000 data benign dan 1000 data malignant. Data malignant kanker memuat *melanoma*, *basal cell carcinoma*, dan *squamous cell carcinoma* sedangkan pada data benign kanker memuat tahi lalat, tanda lahir, dan bekas luka. data tersebut memiliki tipe gambar RGB (*Red, Green, Blue*). Data tersebut akan di proses dengan menggunakan 2 tahap yaitu perbaikan kualitas citra (*pre-processing*) dan klasifikasi dengan menggunakan Deep Learning. Metode Deep Learning sudah memuat proses ekstraksi fitur sehingga proses ekstraksi fitur sudah tidak diperlukan. Proses pertama yang dilakukan adalah perbaikan kualitas citra.

Perbaikan kualitas citra yang dipakai dalam penelitian ini adalah proses *Dull Razor filtering* untuk menghilangkan fitur rambut pada citra dermoscopic, mengubah gambar menjadi *grayscale* sehingga beberapa *noise* yang ada pada gambar dapat dikurangi, *median filter* untuk menghilangkan *noise-noise* yang masih tertinggal, dan yang terakhir adalah *histogram equalization* untuk meningkatkan ketajaman gambar sehingga beberapa informasi dalam gambar lebih

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

terlihat. Tahap selanjutnya adalah klasifikasi dengan menggunakan Deep Learning. Metode Deep Learning yang digunakan adalah Deep Extreme Learning Machine (DELM). DELM menggunakan metode *feature descriptor* untuk mendapatkan fitur dari sebuah gambar. Metode feature descriptor yang digunakan adalah HOG descriptor dan selanjutnya data hasil ekstraksi akan digunakan untuk input dalam proses klasifikasi.

Klasifikasi pada penelitian ini dilakukan ke dalam 2 kelas yaitu malignant kanker dan benign kanker. Gambar 4.1 merupakan contoh sampel data yang dipakai dalam proses klasifikasi dari tiap-tiap kelas.



Gambar 4.1 (a) Sampel Data *Malignant* (b) Sampel Data *Benign*

Citra dermoscopic tidak dapat langsung diproses pada tahap klasifikasi karena pada citra dermoscopic memiliki banyak *noise* yang nantinya mempengaruhi hasil pada proses klasifikasi. Selain memiliki noise yang banyak, pada citra dermoscopic juga memiliki nilai intensitas cahaya yang tidak seragam sehingga diperlukan perbaikan citra dengan tujuan pemerataan nilai intensitas cahaya. Berdasarkan permasalahan diatas, maka diperlukan proses *pre-processing* untuk memperbaiki permasalah yang ada pada citra digital. Berikut merupakan penjelasan lebih rinci mengenai *pre-processing* yang digunakan pada penelitian

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

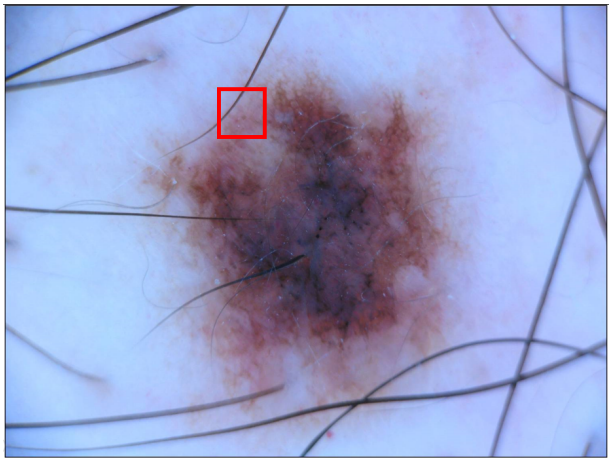
mengenai klasifikasi kanker kulit.

4.1.1. *Dull Razor Filtering*

Citra dermoscopic yang dipakai dalam penelitian kadang kala memiliki citra rambut sehingga untuk menyamakan citra untuk memaksimalkan proses klasifikasi, maka diperlukan sebuah tahapan untuk menghilangkan fitur rambut tersebut. Salah satu metode yang dapat digunakan adalah Dull Razor Filtering. Proses pada metode ini bekerja pada citra RGB dengan tahapan metode:

1. Mengidentifikasi citra rambut dan menghilangkannya dengan menggunakan metode *morphological closing operation*. *Morphological closing* yang digunakan membutuhkan *line structuring element* dengan derajat yang dipakai adalah 90° dan 180° dengan radius 10.
2. Menghaluskan gambar dan mengganti piksel rambut dengan menggunakan metode *adaptive median filter*.

Diberikan sebuah citra dan diambil sampel matriks yang ada pada citra tersebut. Posisi pada pengambilan sampel matriks sebesar 12 x 12 piksel dapat dilihat pada Gambar 4.2.



Gambar 4.2 Pengambilan Matriks untuk proses *Dull Razor Filtering*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada sampel tersebut menghasilkan 3 kali matriks pada tiap *layer* yaitu *red* (R), *green* (G), dan *blue* (B) dengan ukuran masing-masing sebesar 12 x 12 yang dapat dilihat pada Gambar 4.3.

215	213	213	213	213	213	213	213	212	212	212	211
215	213	213	213	213	213	213	213	211	211	211	211
215	213	213	213	213	213	213	213	210	210	210	210
215	213	215	213	215	213	215	213	212	210	212	210
213	212	213	212	213	212	213	212	215	214	215	214
212	212	212	212	212	212	212	212	214	214	214	214
211	211	211	211	211	211	211	211	214	214	214	214
210	210	210	210	210	210	210	210	213	213	213	213
209	209	209	209	209	209	209	209	211	212	212	213
210	210	210	210	210	210	210	210	210	211	211	212
210	210	210	210	210	210	210	210	210	210	211	212
211	211	211	211	211	211	211	211	209	210	210	211

(a)

221	221	221	221	221	221	221	221	220	220	220	219
221	221	221	221	221	221	221	221	219	219	219	219
221	221	221	221	221	221	221	221	218	218	218	218
219	219	221	221	221	221	221	221	218	218	218	218
214	214	216	216	216	216	218	218	220	220	222	222
213	213	213	213	215	215	217	217	219	219	221	221
212	212	212	212	214	214	216	216	219	219	221	221
211	211	211	211	213	213	215	215	218	218	220	220
210	210	210	210	212	212	214	214	216	217	219	220
211	211	211	211	213	213	215	215	215	216	218	219
211	211	211	211	213	213	215	215	215	215	218	219
212	212	212	212	214	214	216	216	214	215	217	218

(c)

209	210	210	210	210	210	210	210	209	209	209	208
209	210	210	210	210	210	210	210	208	208	208	208
209	210	210	210	210	210	210	210	207	207	207	207
209	210	209	210	209	210	209	210	206	207	206	207
206	207	205	206	205	206	205	206	207	208	207	208
205	205	205	205	204	204	204	204	206	206	206	206
204	204	204	204	203	203	203	203	206	206	206	206
203	203	203	203	202	202	202	202	205	205	205	205
202	202	202	202	201	201	201	201	203	204	204	205
203	203	203	203	202	202	202	202	202	203	203	204
203	203	203	203	202	202	202	202	202	202	203	204
204	204	204	204	203	203	203	203	201	202	202	203

(b)

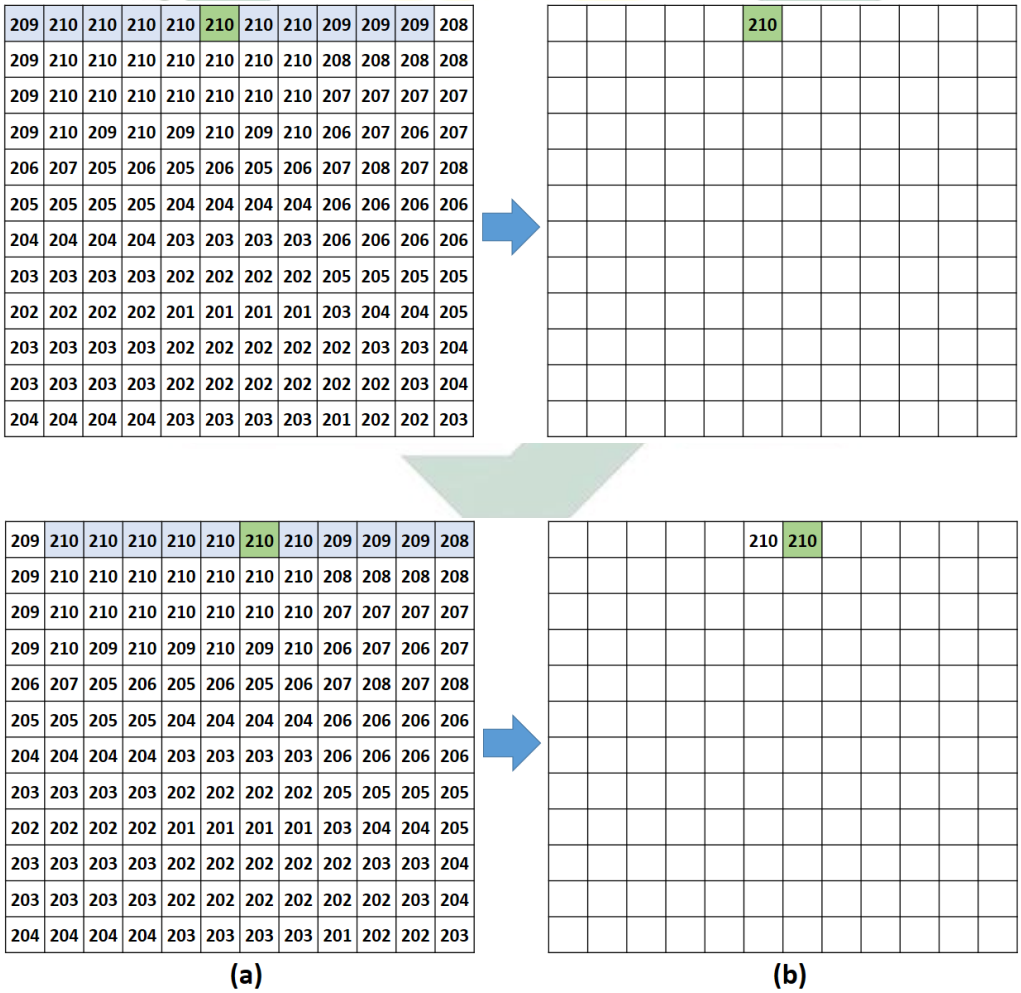
Gambar 4.3 (a) Red Layer (b) Green Layer (c) Blue Layer

Sebagai contoh perhitungan digunakan *green layer* sebagai sampel. *Structuring element* (SE) yang dipakai adalah line dengan derajat 90° dan 180° dengan panjang 10 piksel tiap proses. Dikarenakan panjang pikselnya genap, maka panjang piksel SE diubah menjadi 11. Proses yang akan dilakukan adalah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

morphological closing operation sehingga prosesnya adalah dilasi yang kemudian dilakukan proses erosi. Proses dilasi adalah mencari nilai maksimal sesuai dengan *mask* dari SE. Contoh perhitungan dapat dilihat dibawah ini dan untuk lebih jelasnya dapat dilihat ilustrasi pada Gambar 4.4

$$
\begin{aligned}
I(1,6) &= max([209\ 210\ 210\ 210\ 210\ 210\ 210\ 210\ 209\ 209\ 209]) \\
&= 210 \\
I(1,7) &= max([210\ 210\ 210\ 210\ 210\ 210\ 210\ 209\ 209\ 209\ 208]) \\
&= 210
\end{aligned}
$$



Gambar 4.4 (a) Original Image (b) Hasil Dilasi 180°

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Proses perhitungan tersebut dilakukan sampai seluruh piksel diubah sehingga hasil akhir dari proses dilasi pada derajat 180° dapat dilihat pada Gambar 4.5.

210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
207	207	207	207	208	208	208	208	208	208	208	208
205	205	205	206	206	206	206	206	206	206	206	208
204	204	204	206	206	206	206	206	206	206	206	208
204	204	204	205	205	205	205	205	205	205	206	208
203	203	203	203	204	204	205	205	205	205	207	208
203	203	203	203	203	203	204	204	205	205	207	208
203	203	203	203	203	203	204	204	205	205	207	208
204	204	204	204	204	204	204	204	205	205	207	208

Gambar 4.5 Sampel Hasil Akhir Dilasi 180°

Pada Gambar 4.6 terlihat nilai matriksnya sudah banyak yang berubah dibandingkan dengan matriks awalnya. Proses selanjutnya adalah perhitungan dilasi dengan sudut 90°. Perhiutngan dilasi dengan sudut 90° menggunakan nilai dari hasil perhitungan dilasi 180°. Contoh perhitungan dilasi dengan sudut 90° dan simulasi yang dapat dilihat pada Gambar 4.6

$$
\begin{aligned}
I(6,1) &= max([210\ 210\ 210\ 210\ 207\ 205\ 204\ 204\ 203\ 203\ 203]) \\
&= 210 \\
I(7,1) &= max([210\ 210\ 210\ 207\ 205\ 204\ 204\ 203\ 203\ 203\ 204]) \\
&= 210
\end{aligned}
$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
207	207	207	207	208	208	208	208	208	208	208	208
205	205	205	206	206	206	206	206	206	206	206	208
204	204	204	206	206	206	206	206	206	206	206	208
204	204	204	205	205	205	205	205	205	205	206	208
203	203	203	203	204	204	205	205	205	205	207	208
203	203	203	203	203	203	204	204	205	205	207	208
203	203	203	203	203	203	204	204	205	205	207	208
204	204	204	204	204	204	204	204	205	205	207	208

210											

210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
207	207	207	207	208	208	208	208	208	208	208	208
205	205	205	206	206	206	206	206	206	206	206	208
204	204	204	206	206	206	206	206	206	206	206	208
204	204	204	205	205	205	205	205	205	205	206	208
203	203	203	203	204	204	205	205	205	205	207	208
203	203	203	203	203	203	204	204	205	205	207	208
203	203	203	203	203	203	204	204	205	205	207	208
204	204	204	204	204	204	204	204	205	205	207	208

210											
210											

(a) **(b)**

Gambar 4.6 (a) Original Image (b) Hasil Dilasi 90°

Perhitungan dilasi 90° dilakukan sama seperti dilasi 180° dimana prosesnya dilakukan sampai seluruh jumlah piksel telah dihitung. Perbedaan dilasi 180° dengan 900° terletak pada arah matriks *masking*-nya. Dilasi 180° memiliki arah matriks yang horizontal sedangkan dilasi 90° memiliki arah matriks yang vertikal. Hasil akhir keseluruhan proses dilasi dapat dilihat pada Gambar 4.7

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

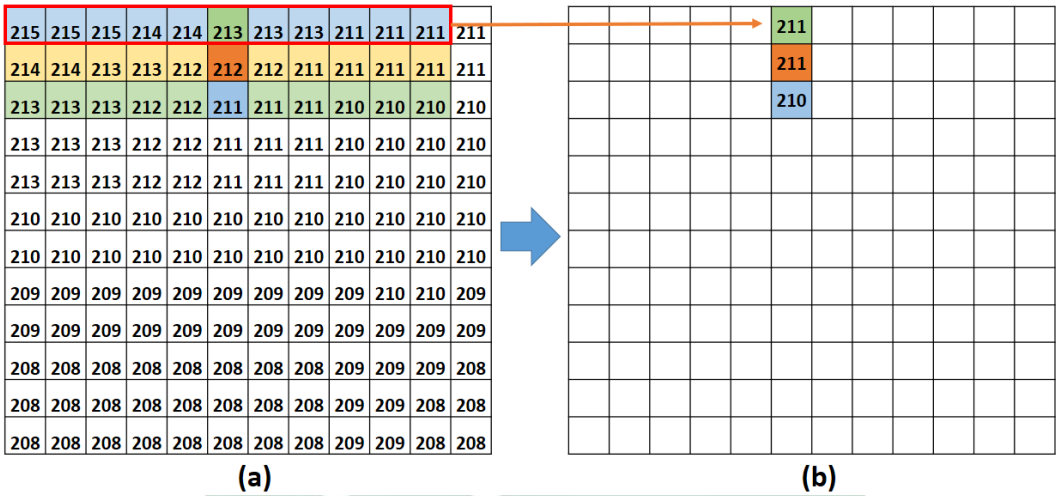
215	215	215	214	214	213	213	213	211	211	211	211
214	214	213	213	212	212	212	211	211	211	211	211
213	213	213	212	212	211	211	211	210	210	210	210
213	213	213	212	212	211	211	211	210	210	210	210
213	213	213	212	212	211	211	211	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
210	210	210	210	210	210	210	210	210	210	210	210
209	209	209	209	209	209	209	209	209	210	210	209
209	209	209	209	209	209	209	209	209	209	209	209
208	208	208	208	208	208	208	208	209	209	209	208
208	208	208	208	208	208	208	208	209	209	208	208
208	208	208	208	208	208	208	208	209	209	208	208

Gambar 4.7 Sampel Hasil Akhir Dilasi 90°

Proses selanjutnya untuk mendapatkan hasil *closing morphological operation* adalah dengan melakukan erosi menggunakan *structuring element* yang sama seperti proses dilasi yaitu pada tipe "*line*" dengan sudut 180° dan 90°. Proses erosi merupakan suatu proses untuk mencari nilai minimun sesuai dengan mask yang digunakan. Contoh perhitungan erosi dengan sudut 180° dengan panjang masking 10 dapat dilihat dibawah ini dan untuk simulasinya dapat dilihat pada Gambar 4.8.

$$
\begin{aligned}
I(1,6) &= min([215\ 215\ 215\ 214\ 214\ 213\ 213\ 213\ 211\ 211\ 211]) \\
&= 211 \\
I(2,6) &= min([214\ 214\ 213\ 213\ 212\ 212\ 212\ 211\ 211\ 211\ 211]) \\
&= 211 \\
I(3,6) &= min([213\ 213\ 213\ 212\ 212\ 211\ 211\ 211\ 210\ 210\ 210]) \\
&= 210
\end{aligned}
$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.8 (a) Original image (b) Hasil Erosi 180°

Proses perhitungan seperti pada Gambar 4.8 dilakukan terus-menerus sampai seluruh piksel mendapatkan hasil erosi 180°. Hasil keseluruhan piksel setelah dilakukan erosi 180° dapat dilihat pada Gambar 4.9.

212	212	212	211	211	211	211	211	211	211	211	211
212	212	211	211	211	211	211	211	211	211	211	211
211	211	211	210	210	210	210	209	209	209	209	209
210	210	210	210	210	210	210	209	208	208	208	207
210	210	210	210	210	210	210	209	207	207	207	205
210	210	210	210	210	210	210	209	207	206	206	205
209	209	209	209	209	210	210	209	207	207	206	205
209	209	209	209	209	209	209	208	207	207	206	205
208	208	208	208	208	209	209	208	207	207	206	205
207	207	207	207	207	208	208	207	207	207	206	205
206	206	206	206	207	208	208	207	207	207	206	205
206	206	206	206	207	208	208	207	207	207	206	205

Gambar 4.9 Sampel Hasil Akhir Erosi 180°

Hasil tersebut akan digunakan dalam proses perhitungan dilasi dengan sudut sebesar 90°. Contoh Perhitungan erosi dengan sudut 90° dapat dilihat pada Gambar 4.10.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

212	212	212	211	211	211	211	211	211	211	211	211
212	212	211	211	211	211	211	211	211	211	211	211
211	211	211	210	210	210	210	209	209	209	209	209
210	210	210	210	210	210	210	209	208	208	208	207
210	210	210	210	210	210	210	209	207	207	207	205
210	210	210	210	210	210	210	209	207	206	206	205
209	209	209	209	209	210	210	209	207	207	206	205
209	209	209	209	209	209	209	208	207	207	206	205
208	208	208	208	208	209	209	208	207	207	206	205
207	207	207	207	207	208	208	207	207	207	206	205
206	206	206	206	207	208	208	207	207	207	206	205
206	206	206	206	207	208	208	207	207	207	206	205

(a)

206	206	206									

(b)

Gambar 4.10 (a) Original Image (b) Hasil Erosi 90°

Sama halnya seperti pada erosi dengan sudut 180°, erosi pada sudut 90° juga dilakukan perhitungan sampai mendapatkan hasil akhir seperti pada Gambar 4.11.

210	210	210	210	210	210	210	209	207	206	206	205
209	209	209	209	209	210	210	209	207	206	206	205
209	209	209	209	209	209	209	208	207	206	206	205
208	208	208	208	208	209	209	208	207	206	206	205
207	207	207	207	207	208	208	207	207	206	206	205
206	206	206	206	207	208	208	207	207	206	206	205
206	206	206	206	207	208	208	207	207	206	206	205
206	206	206	206	207	208	208	206	207	206	206	205
206	206	206	206	207	208	208	207	205	206	206	205
206	206	206	206	207	207	208	207	207	206	206	205
206	206	206	206	207	207	208	207	207	206	206	205
206	206	206	206	207	207	208	207	207	207	206	205

Gambar 4.11 Sampel Hasil Akhir Erosi 90°

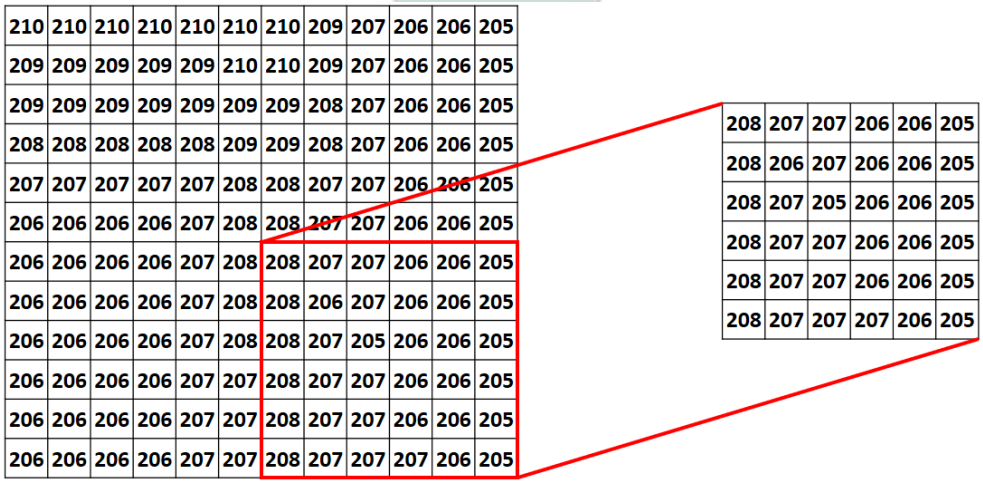
Pada proses dull razor filtering diperlukan proses *adaptive median filter* untuk menghaluskan gambar sehingga didapatkan gambar yang baik. Proses *adaptive median filter* menggunakan nilai matriks dari hasil *closing morphological operation* yang dapat dilihat pada Gambar 4.11. Pada proses *adaptive median filter*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

digunakan ukuran *window* 3 x 3 dan *max window* dengan ukuran 5 x 5 sehingga data sampel yang akan digunakan sebesar 6 x 6 dari data sampel yang ada pada Gambar 4.11. *Adaptive median filter* memiliki 3 kondisi yang harus dipenuhi dengan matriks X sebagai input. Kondisi tersebut yaitu:

1. Ketika memenuhi kondisi $X_{max} < X_{mean} < X_{min}$ dan $X_{max} < X_{xy} < X_{min}$ dengan X_{xy} adalah nilai tengah dari piksel yang sudah dilakukan proses *window*, maka nilai X_{xy} tetap tanpa diubah.
2. Ketika memenuhi kondisi $X_{max} < X_{mean} < X_{min}$ dan tidak memenuhi kondisi $X_{max} < X_{xy} < X_{min}$, maka nilai X_{xy} diubah menjadi nilai X_{mean}.
3. Ketika tidak memenuhi kondisi $X_{max} < X_{mean} < X_{min}$ dan $X_{max} < X_{xy} < X_{min}$, maka ukuran *window* diganti dengan *max window* dan dilakukan proses *median filter* seperti biasa.

Selanjutnya diambil sampel data dari hasil *closing morphological operation* dengan ukuran 6 x 6. Contoh pengambilan sampel data untuk proses adaptive median filter dapat dilihat pada Gambar 4.12.



Gambar 4.12 Sampel Pengambilan Data untuk *Adaptive Median Filter*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Sampel data yang digunakan telah didefinisikan pada Gambar 4.12 dengan ukuran 6 x 6. Selanjutnya sampel tersebut dilakukan perhitungan dengan menggunakan *adaptive median filter*. Proses *adaptive median filter* hampir sama dengan *median filter* dengan perbedaannya ada pada ketentuan maupun syarat perhitungan. Proses perhitungan dapat dilihat dibawah ini:

208	207	207	206	206	205
208	206	207	206	206	205
208	207	205	206	206	205
208	207	207	206	206	205
208	207	207	206	206	205
208	207	207	207	206	205

(a)

205	206	207	207	207	207	208	208	208

(b)

Gambar 4.13 (a) Pengambilan Windowing 3 x 3 Pada t=1 (b) Pengurutan Nilai dan Pencarian Nilai Median, Minimum, dan Maksimum

Pada Gambar 4.13 terlihat bahwa $X_{xy} = 206$, $X_{max} = 208$, $X_{min} = 205$, dan $X_{med} = 207$ sehingga berdasarkan hasil diatas dapat disimpulkan perhitungan tersebut kondisi atau syarat pertama dimana $X_{max} < X_{mean} < X_{min}$ dan $X_{max} < X_{xy} < X_{min}$. Dikarenakan memenuhi kondisi 1, maka nilai X_{xy} tetap tanpa diganti dengan X_{med}. Hasil perhitungan pada $t = 1$ dapat dilihat pada Gambar 4.14.

	206				

Gambar 4.14 Hasil Perhitungan *Adaptive Median Filter* dengan t=1

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Perhitungan akan dilanjutkan menuju ke t=2. Perhitungan pada t=2 sama seperti t=1 dengan mengurutkan data dan kemudian mencari nilai maksimum, minimum, dan median. Hasil perhitungan pada t=2 dapat dapat dilihat pada Gambar 4.15.

208	207	207	206	206	205
208	206	207	206	206	205
208	207	205	206	206	205
208	207	207	206	206	205
208	207	207	206	206	205
208	207	207	207	206	205

(a)

205	206	206	206	206	207	207	207	207

(b)

Gambar 4.15 (a) Pengambilan Windowing 3 x 3 pada t=2 (b) Pengurutan Nilai dan Pencarian Nilai Median, Minimum, dan Maksimum

Pada Gambar 4.15 terlihat bahwa $X_{xy} = 207$, $X_{max} = 207$, $X_{min} = 205$, dan $X_{med} = 206$ sehingga berdasarkan hasil diatas dapat disimpulkan perhitungan tersebut kondisi atau syarat kedua dimana memenuhi syarat $X_{max} < X_{mean} < X_{min}$ dan tidak memnuhi syarat $X_{max} < X_{xy} < X_{min}$. Berdasarkan ketentuan dari *adaptive median filter*, maka nilai X_{xy} diganti dengan niali X_{med} sehingga nilai X_{xy} menjadi 206. Hasil perhitungan pada t=2 dapat dapat dilihat pada Gambar 4.16.

	206	206			

Gambar 4.16 Hasil Perhitungan *Adaptive Median Filter* dengan t=2

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Selanjutnya akan diberikan sebuah contoh ketika proses *windowing* pada *adaptive median filter* ada pada kondisi 3 dengan ukuran *max window* sebesar 5 x 5. Contoh tersebut dapat dilihat pada Gambar 4.17.

208	207	207	206	206	205
208	206	207	206	206	205
208	207	205	206	206	205
208	207	207	206	206	205
208	207	207	206	206	205
208	207	207	207	206	205

(a)

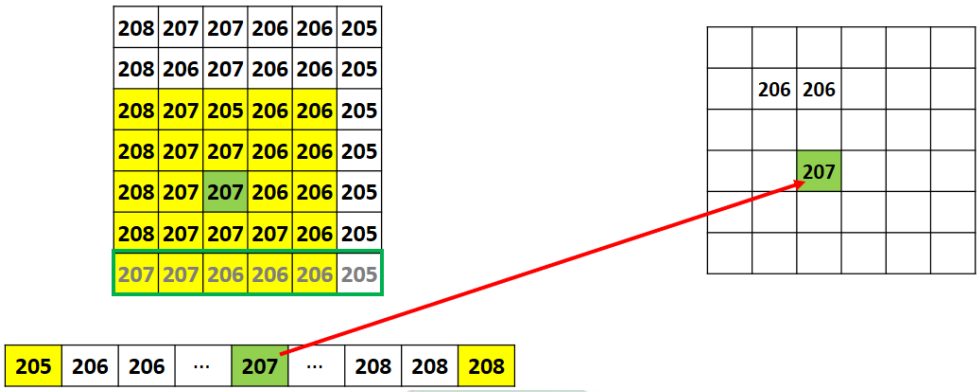
206	206	207	207	207	207	207	207	207

(b)

Gambar 4.17 (a) Pengambilan Windowing 3 x 3 pada t=2 (b) Pengurutan Nilai dan Pencarian Nilai Median, Minimum, dan Maksimum

Terlihat pada Gambar 4.17 bahwa kondisi tersebut memenuhi kondisi ke 3 yaitu tidak memenuhi kondisi $X_{max} < X_{mean} < X_{min}$ dan $X_{max} < X_{xy} < X_{min}$. Rincian data yang didapat adalah $X_{max} = 207$, $X_{xy} = 207$, $X_{med} = 207$, dan $X_{min} = 206$. Pada kasus Gambar 4.17 nilai X_{max}, X_{xy}, X_{med} sama yaitu 207 sehingga kondisi yang memenuhi hanya pada konsidi ketiga. Selanjutnya dilakukan penambahan *windowing* sehingga menjadi 5 x 5 dan dilanjutkan dengan menghitung nilai mediannya. Hasil penambahan *windowing* dapat dilihat pada Gambar 4.18. Pada gambar tersebut terdapat nilai angka yang berwarna abu-abu yang memiliki border berwarna hijau. Data tersebut merupakan data tambahan karena kurangnya data sampel yang diambil sehingga mengambil tambahan data yang ada pada citra aslinya agar dapat dilakukan proses *median filter* dengan *window* 5 x 5.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.18 Hasil Penambahan *Windowing* Menjadi 5 x 5

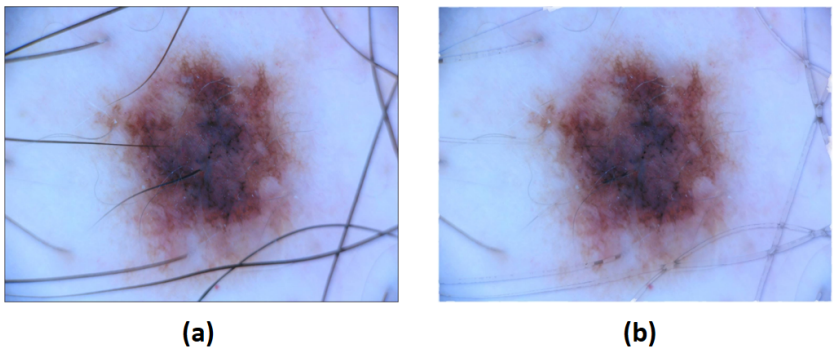
Perhitungan akan terus dilakukan sampai mendapatkan kesuluran nilai pada *adaptive median filter*. Hasil akhir dari sampel perhitungan dapat dilihat pada Gambar 4.19.

206	206	206	206	206	205
206	206	206	206	206	205
207	207	207	206	206	206
207	207	207	206	206	206
207	207	207	206	206	206
207	207	207	206	206	206

Gambar 4.19 Hasil Akhir Perhitungan *Adaptive Median Filter* dengan Data Sampel

Berikut adalah hasil akhir citra setelah dilakukan proses *adaptive median filter* yang disajikan pada Gambar 4.20. Hasil citra tersebut sudah membersihkan fitur rambut yang ada pada kulit. Fitur kulit yang sudah dilakukan proses *dull razor filtering* akan lebih memiliki noise yang sedikit dikarenakan noise yang berupa fitur rambut telah dihilangkan. Dull razor filtering juga menghaluskan gambar sehingga hasil akhir dari tahap ini memiliki gambar yang terlihat lebih bersih.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

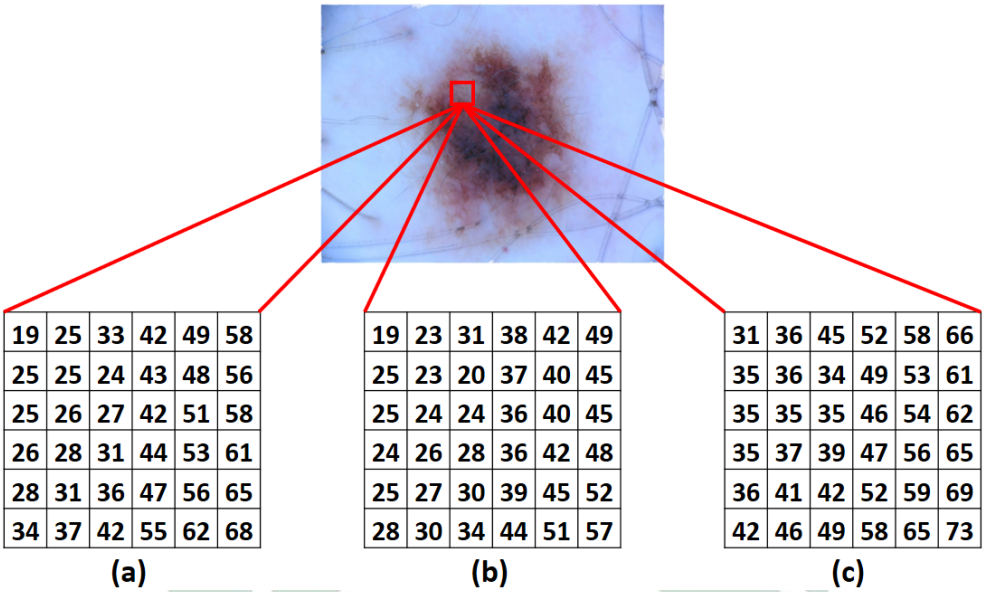


Gambar 4.20 (a) Citra Awal (b) Hasil Akhir Proses *Dull Razor Filtering*

4.1.2. *Grayscale Image*

Kebanyakan suatu tahap *image processing* menggunakan citra dengan dimensi *grayscale*, sedangkan citra dermoscopic ada pada dimensi RGB. Pengubahan citra RGB menjadi *grayscale* juga dimaksudkan untuk mengurangi *noise* dan hanya mengambil informasi yang penting pada citra. Pengubahan citra ini juga dapat berfungsi untuk menyederhanakan citra dan proses perhitungan. Matriks yang dihasilkan dari proses konversi RGB ke *Grayscale* memiliki skala nilai 0-255. Semakin kecil nilai yang dihasilkan, maka warna dari gambar akan semakin gelap begitu pula sebaliknya. Perhitungan konversi RGB ke *grayscale* akan menggunakan sampel data sebesar 6 x 6 untuk mempermudah perhitungan. Pada sampel data yang diambil akan dipisahkan berdasarkan channel warna pada tiap layer yaitu *red layer*, *green channel*, dan *blue layer*. Tiap *layer* pada RGB memiliki nilai yang berbeda-beda sehingga memperlukan sebuah persamaan untuk melakukan konversi menuju *grayscale*. Sampel data yang diambil dapat dilihat pada Gambar 4.21.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.21 (a) Red Layer (b) Green Layer (c) Blue Layer

Nilai yang dihasilkan pada tiap tiap layer yang sudah disajikan pada Gambar 4.21 akan dihitung menggunakan Persamaan 2.1. Simulasi untuk perhitungan konversi RGB ke *grayscale* dapat dilihat pada Gambar 4.22. Contoh perhitungan konversi secara matematis disajikan dibawah ini:

$$
\begin{aligned}
G(1,1) &= (0.299(r)) + (0.587(g)) + (0.144(b)) \\
&= (0.299(19)) + (0.587(19)) + (0.144(31)) \\
&= (5.681) + (11.153) + (4.464) \Rightarrow \text{ pembulatan kebawah} \\
&= 5 + 11 + 4 \\
&= 20 \\
G(1,2) &= (0.299(25)) + (0.587(23)) + (0.144(36)) \\
&= (7.475) + (13.501) + (5.184) \Rightarrow \text{ pembulatan kebawah} \\
&= 7 + 13 + 5 \\
&= 25
\end{aligned}
$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
G(1,2) &= (0.299(33)) + (0.587(31)) + (0.144(45)) \\
&= (9.867) + (18.197) + (6.48) \Rightarrow \text{ pembulatan kebawah} \\
&= 9 + 18 + 6 \\
&= 33
\end{aligned}
$$

19	25	33	42	49	58
25	25	24	43	48	56
25	26	27	42	51	58
26	28	31	44	53	61
28	31	36	47	56	65
34	37	42	55	62	68

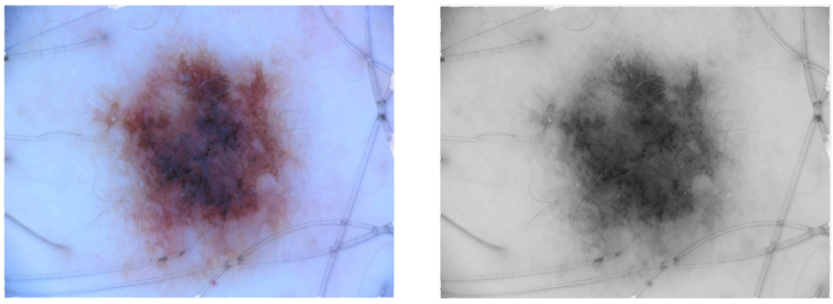
19	23	31	38	42	49
25	23	20	37	40	45
25	24	24	36	40	45
24	26	28	36	42	48
25	27	30	39	45	52
28	30	34	44	51	57

31	36	45	52	58	66
35	36	34	49	53	61
35	35	35	46	54	62
35	37	39	47	56	65
36	41	42	52	59	69
42	46	49	58	65	73

20	25	33	41	46	54
26	25	23	40	44	50
26	26	26	39	45	51
26	28	30	40	47	54
27	30	33	43	50	58
31	34	38	49	56	62

Gambar 4.22 Simulasi Perhitungan Konversi RGB ke Grayscale

Perhitungan terus dilakukan sampai mendapatkan semua hasilnya seperti pada Gambar 4.22. Berdasarkan hasil tersebut dapat divisualisasikan hasil dari konversi RGB ke *grayscale* berdasarkan data asli. Hasil akhir proses konversi ke *grayscale* dapat dilihat pada Gambar 4.23.



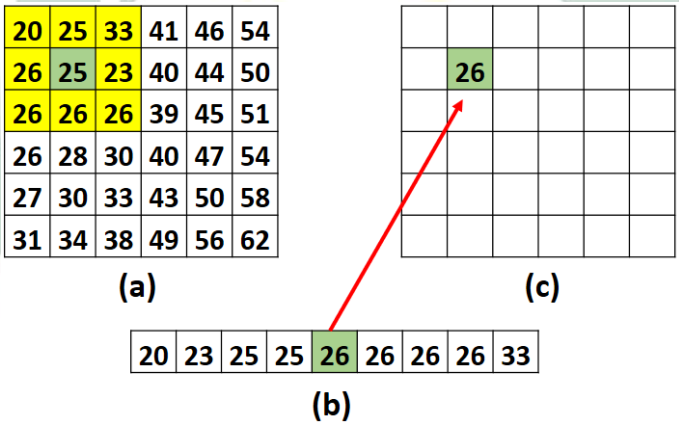
(a) (b)

Gambar 4.23 (a) Citra RGB (b) Citra *Grayscale*

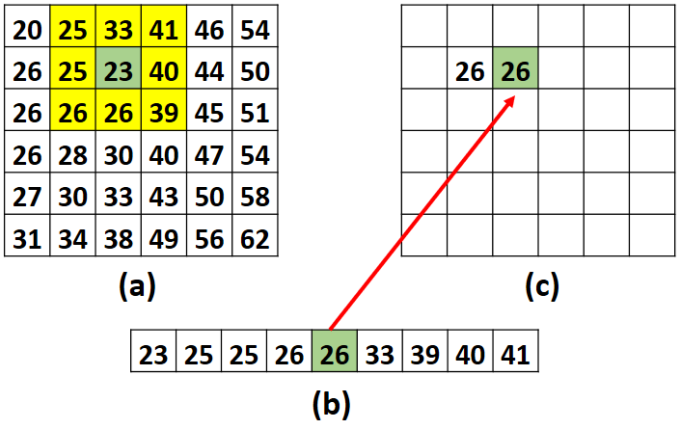
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.1.3. *Median Filtering*

Proses *median filter* merupakan proses *filtering* untuk menghaluskan gambar. *Median filtering* juga dapat berguna untuk mengurangi *noise* kecil seperti *salt and pepper noise* yang sering kali ada pada citra. Proses *median filter* akan menggunakan *window* 3 x 3 yang kemudian dilakukan proses *windowing* untuk mencari nilai median dengan menggunakan Persamaan 2.2. Setelah nilai median didapatkan, selanjutnya adalah mengganti nilai piksel yang ada pada tengah-tengah matriks menjadi nilai median yang telah diperoleh. Simulasi perhitungan *median filter* dapat dilihat pada Gambar 4.24, 4.25, dan 4.26.

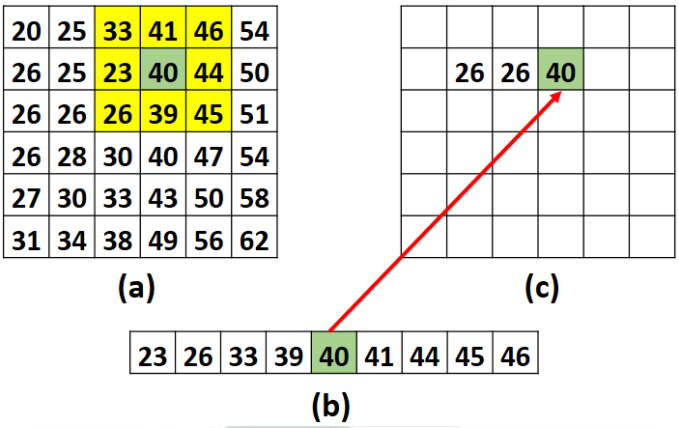


Gambar 4.24 (a) Windowing 3 x 3 (b) Nilai Median (c) Hasil Median Filter pada t=1



Gambar 4.25 (a) Windowing 3 x 3 (b) Nilai Median (c) Hasil Median Filter pada t=2

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.26 (a) Windowing 3 x 3 (b) Nilai Median (c) Hasil Median Filter pada t=3

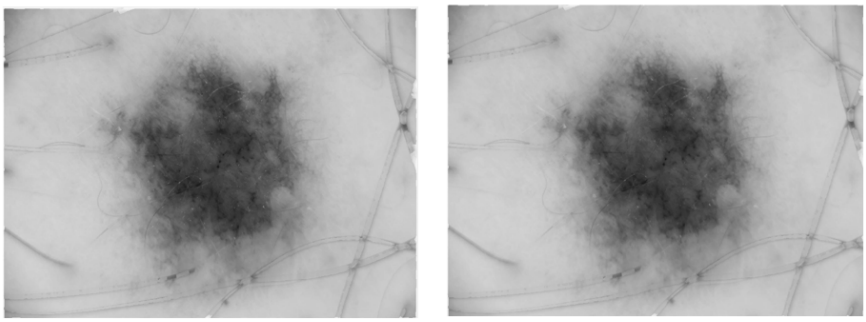
Gambar yang dihasilkan dari proses *median filter* tidak akan terlihat jelas perbedaanya karena *median filter* hanya menghapus noise kecil. Hasil *median filter* akan dirasakan ketika citra hasil *median filter* diolah menjadi data numerik. *Median filter* juga kerap dipakai dalam proses penghalusan gambar dan biasanya menggunakan ukuran *window* yang lebih besar. Pada kasus *image processing* kali ini hanya digunakan untuk membersihkan *noise* yang ada pada Gambar sehingga cukup menggunakan ukuran window 3 x 3.

Pada Gambar 4.24(a), kolom dengan warna kuning merupakan *windowing* dengan ukuran 3 x 3 yang kemudian nilai pada *windowing* akan dijadikan matriks baris berukuran 1 x 9. Nilai yang sudah dijadikan matriks baris akan dilakukan proses pengurutan nilai dari nilai yang lebih kecil menuju nilai yang lebih besar. Hasil dari proses pengurutan $t = 1$ dapat dilihat pada Gambar 4.24(b). Nilai yang telah urut tersebut kemudian diambil nilai mediannya. Pada kasus t=1 didapatkan nilai median 207 sehingga nilai tengah *windowing* berubah dari 25 menjadi 26. Pada kasus t=2 nilai tengah *windowing* berubah dari 23 menjadi 26 dan pada t=3 nilai tengah *windowing* tetap dikarenakan nilai tengahnya sama dengan nilai mediannya. Hasil perhitungan tiap piksel dari *median filter* dapat dilihat pada Gambar 4.27.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Hasil untuk citra keluaran dari proses *median filter* juga dapat dilihat pada Gambar 4.27

25	25	33	41	46	54
25	26	26	40	45	51
26	26	28	40	45	51
26	27	30	40	47	54
28	30	34	43	50	56
31	34	42	49	56	59



(a) (b)

Gambar 4.27 (a) Citra Awal (b) Citra Hasil *Median Filter*

4.1.4. *Histogram Equalization*

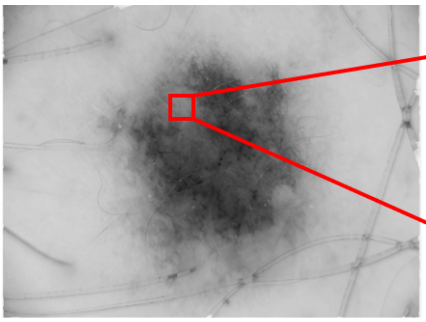
Umumnya suatu citra *grayscale* memiliki tingkat kecerahan yang tidak merata sehingga kasus ini dapat memberikan dampak buruk pada proses klasifikasi. *Histogram equalization* merupakan suatu metode untuk mengatasi masalah tingkat kecerahan yang tidak merata pada suatu citra. *Histogram equalization* tidak hanya dapat meratakan kecerahan suatu citra, namun *histogram equalization* juga dapat memberikan efek gambar yang lebih tajam sehingga lebih mudah untuk mengidentifikasi fitur yang ada didalamnya. Proses *histogram equalization* dapat dijabarkan seperti berikut:

1. **Menghitung jumlah tingkat keabuan yang ada pada suatu citra**

   Pada tahap ini, dicari berapa banyak jumlah kemunculan suatu nilai piksel citra. Sebagai contoh, diasumsikan n merupakan jumlah piksel yang muncul.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Suatu citra memiliki nilai piksel yang bernilai 25 muncul sebanyak 3 kali dalam suatu image sehingga nilai n untuk 25 adalah 3 dan dapat ditulis $n(25) = 3$. Perhitungan lebih lengkap untuk tahap ini dapat dilihat pada Gambar 4.28



25	25	33	41	46	54
25	26	26	40	45	51
26	26	28	40	45	51
26	27	30	40	47	54
28	30	34	43	50	56
31	34	42	49	56	59

Gray	0	⋯	25	26	27	28	29	⋯	55	56	57	58	59	⋯	255
n	4	⋯	3	9	11	8	3	⋯	3129	3429	3451	3618	3542	⋯	0

Gambar 4.28 Perhitungan Jumlah Tingkat Keabuan yang Ada Pada Suatu Citra

2. **Menghitung probabilitas dari nilai jumlah tingkat keabuan**

Pada tahap ini dilakukan suatu perhitungan untuk mencari probabilitas dari kemunculan nilai piksel pada tahap 1. Hal pertama yang perlu dilakukan untuk mencari probabilitas adalah mencari jumlah seluruh nilai yang dapat dilihat dibawah ini:

$$\begin{aligned} jumlah\ keseluruhan &= \sum_{i=0}^{N} n \\ &= (4 + \cdots + 3 + 9 + 11 + 8 + \cdots + 0) \\ &= 1698016 \end{aligned}$$

Setelah didapatkan nilai jumlah dari keseluruhan piksel yang muncul selanjutnya dilakukan perhitungan probabilitas yang dapat dilihat dibawah ini:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$P(n) = \frac{n}{jumlah}$$
$$P(0) = \frac{4}{1698016}$$
$$P(0) = 0.00000236$$

$$P(n) = \frac{n}{jumlah}$$
$$P(58) = \frac{3618}{1698016}$$
$$P(58) = 0.0021$$

$$P(n) = \frac{n}{jumlah}$$
$$P(27) = \frac{8}{1698016}$$
$$P(27) = 0.00000471$$

$$P(n) = \frac{n}{jumlah}$$
$$P(255) = \frac{0}{1698016}$$
$$P(255) = 0$$

Untuk mempersingkat perhitungan dapat dilihat hasil akhir dari perhitungan probabilitas pada Gambar 4.29.

Gray	0	⋯	25	26	27	⋯
n	4	⋯	9	11	8	⋯
P	0.00000023	⋯	0.00000053	0.00000064	0.00000047	⋯

Gray	55	56	57	58	59	⋯	255
n	3129	3429	3451	3618	3542	⋯	0
P	0.0018	0.0020	0.0020	0.0021	0.0021	⋯	0

Gambar 4.29 Perhitungan Probabilitas pada *Histogram Equalization*

3. **Menghitung nilai kumulatif dari probabilitas**

Nilai kumulatif ini akan digunakan untuk menghitung nilai piksel baru yang lebih rata dalam hal kecerahan. Berikut adalah contoh perhitungan nilai kumulatif dari probabilitas:

$$T(n) = \sum_{j=0}^{k} P_j$$
$$T(0) = 0.00000236$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
T(n) &= \sum_{j=0}^{k} P_j \\
T(55) &= 0.00000236 + \cdots + 0.00000471 + \cdots + 0.0018 \\
T(55) &= 0.0148
\end{aligned}
$$

$$
\begin{aligned}
T(n) &= \sum_{j=0}^{k} P_j \\
T(255) &= 0.00000236 + \cdots + 0.00000471 + \cdots + 0.0018 + \cdots + 0 \\
T(255) &= 1.00
\end{aligned}
$$

Untuk hasil lengkap perhitungan nilai kumulatif dapat dilihat pada Gambar 4.30

Gray	0	⋯	25	26	27	⋯
n	4	⋯	9	11	8	⋯
P	0.00000023	⋯	0.00000053	0.00000064	0.00000047	⋯
C	0.00000023	⋯	0.0000013	0.0000019	0.0000024	⋯

Gray	55	56	57	58	59	⋯	255
n	3129	3429	3451	3618	3542	⋯	0
P	0.0018	0.0020	0.0020	0.0021	0.0021	⋯	0
C	0.0148	0.0168	0.0188	0.0209	0.0230	⋯	1

Gambar 4.30 Perhitungan Probabilitas Kumulatif pada *Histogram Equalization*

4. **Menentukan nilai piksel baru**

Tahapan terakhir dalam *histogram equalization* adalah menentukan nilai piksel baru dengan cara melakukan perkalian dari nilai kumulatif probabilitas dengan nilai maksimum piksel yaitu 255. Contoh perhitungan dapat dilihat dibawah ini:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$T(n) = C * 255$$
$$T(0) = 0.00000236 * 255$$
$$T(0) = 0.0005865 \approx 0$$

$$T(n) = C * 255$$
$$T(58) = 0.0209 * 255$$
$$T(58) = 5.3295 \approx 5$$

$$T(n) = C * 255$$
$$T(27) = 0.0000241 * 255$$
$$T(27) = 0.006145 \approx 0$$

$$T(n) = C * 255$$
$$T(255) = 1 * 255$$
$$T(255) = 255$$

Untuk keseluruhan nilai yang didapat dengan menggunakan persamaan diatas dapat dilihat pada Gambar 4.31.

Gray	0	⋯	25	26	27	⋯
n	4	⋯	9	11	8	⋯
P	0.00000023	⋯	0.00000053	0.00000064	0.00000047	⋯
C	0.00000023	⋯	0.0000013	0.0000019	0.0000024	⋯
T	0	⋯	0	0	0	⋯

Gray	55	56	57	58	59	⋯	255
n	3129	3429	3451	3618	3542	⋯	0
P	0.0018	0.0020	0.0020	0.0021	0.0021	⋯	0
C	0.0148	0.0168	0.0188	0.0209	0.0230	⋯	1
T	4	4	5	5	6	⋯	255

Gambar 4.31 Perhitungan Nilai Piksel Baru pada *Histogram Equalization*

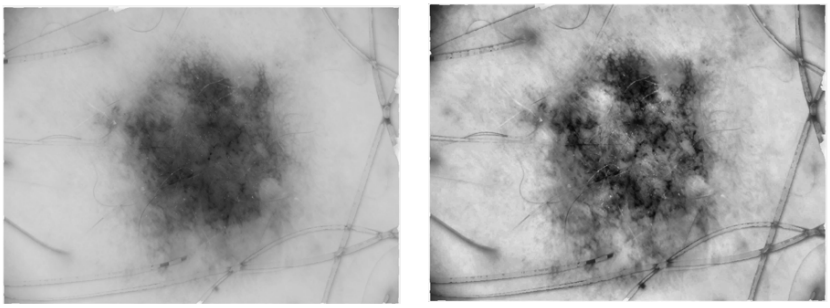
5. **Mengganti piksel lama dengan piksel baru hasil *histogram equalization***

Tahap terakhir dalam *histogram equalization* adalah mengganti piksel lama dengan piksel baru yang sudah didapatkan pada tahap 4. Penggantian piksel ini sesuai dengan posisi semula, jadi tidak mengubah posisi dari piksel awal. Hasil perubahan dan hasil akhir citra dapat dilihat pada Gambar 4.32.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

25	25	33	41	46	54
25	26	26	40	45	51
26	26	28	40	45	51
26	27	30	40	47	54
28	30	34	43	50	56
31	34	42	49	56	59

0	0	0	0	1	3
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	1	3
0	0	0	0	2	4
0	0	0	1	4	6



(a) (b)

Gambar 4.32 (a) Citra Awal (b) Hasil *Histogram Equalization*

Terlihat pada gambar 4.32(b) bahwa hasil hasil dari *histogram equalization* memiliki kecerahan yang seimbang. Perbandingan citra awal dan citra hasil *histogram equalization* ada pada tingkat kecerahannya dan tingkat ketajamannya. Hasil h*istogram equalization* juga memiliki ketajaman yang lebih baik sehingga fitur dari kanker kulit dapat terlihat lebih jelas.

4.2. *HOG Descriptor*

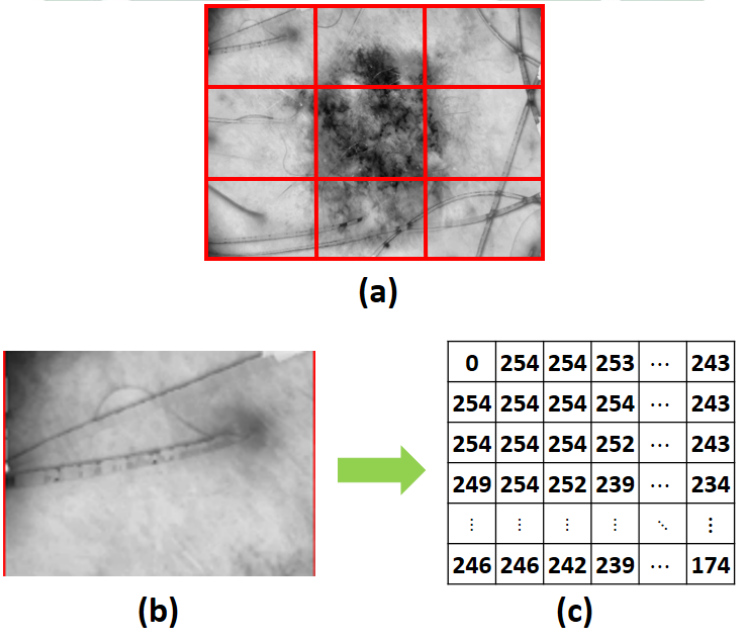
Suatu citra harus diolah menjadi numerik untuk dilakukan proses klasifikasi. Ekstraksi citra menjadi suatu data numerik memiliki beberapa tipe seperti *feature extraction*, *feature descriptor*, *feature selection*, dan lain-lain. Pada *Deep Extreme Learning Machine* yang digunakan adalah ekstraksi citra bertipe *feature descriptor*. *Feature Descriptor* yang digunakan pada penelitian ini adalah *HOG Descriptor* dimana metode ini mengambil nilai gradien dan arah orientasi. Nilai tersebut didapatkan dengan menggunakan Persamaan 2.6 dan Persamaan 2.5 dimana Persamaan 2.6 merupakan persamaan untuk mencari nilai gradien dan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Persamaan 2.5 digunakan untuk mencari arah dari gradien tersebut. Setelah didapatkan nilai keduanya, proses selanjutnya adalah memasukkan nilainya ke dalam *bin* yang sudah tersedia. Langkah-langkah untuk mendapatkan fitur dari *HOG Descriptor* dapat dilihat dibawah ini:

1. **Pembagian block pada citra**

   Langkah pertama dalam melakukan *HOG Descriptor* adalah dengan memecah gambar menjadi beberapa block. Pada penelitian ini digunakan pemisahan sebanyak 9 *block* sehingga gambar akan dipisah menjadi 3 x 3. Pada tiap *block* memiliki nilai fitur *bin* sendiri sehingga nantinya akan menghasilkan fitur sebanyak 81 dikarenakan terdapat 9 *block* dan 9 *bin*. Pembagian *block* untuk satu *image* harus rata dan memiliki ukuran yang sama tiap *block*-nya. Pada penelitian ini, ukuran *image* pada tiap *block* adalah 376 x 501 Simulasi pembagian *block* dan pengambilan sampel data dapat dilihat pada Gambar 4.33.



0	254	254	253	⋯	243
254	254	254	254	⋯	243
254	254	254	252	⋯	243
249	254	252	239	⋯	234
⋮	⋮	⋮	⋮	⋱	⋮
246	246	242	239	⋯	174

Gambar 4.33 (a) Simulasi Pembagian *Block* (b) Pengambilan Sampel *Block* (c) Nilai Piksel pada *Block* Pertama

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2. **Penggunaan *masking* HOG**

Proses selanjutnya adalah penerapan *masking* pada citra yang akan diekstrak menggunakan *HOG Descriptor*. *Masking* yang diberikan dapat dilihat pada Persamaan 2.4. *Masking* dilakukan dengan dua sumbu yaitu sumbu x dan sumbu y sehingga terdapat 2 hasil *masking* yaitu I_x dan I_y. Sebelum dilakukan *masking*, perlu dilakukan penambahan *padding* agar seluruh citra dapat dihitung. Penambahan *padding* ada 2 cara yaitu *zero padding* dan *neighborhood padding*. Pada penelitian ini digunakan *zero padding* yaitu penambahan padding dengan nilai piksel 0. Pada penelitian ini hanya menambahkan padding sebanyak satu piksel di bagian atas, bawah, kanan, dan kiri citra. Hasil penambahan *padding* dapat dilihat pada Gambar 4.34. Pada Gambar tersebut penambahan *padding* akan diberikan warna merah jambu sehingga lebih terlihat penambahan *padding*-nya.

0	0	0	0	0	⋯	0	0
0	0	254	254	253	⋯	243	0
0	254	254	254	254	⋯	243	0
0	254	254	254	252	⋯	243	0
0	249	254	252	239	⋯	234	0
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
0	246	246	242	239	⋯	174	0
0	0	0	0	0	⋯	0	0

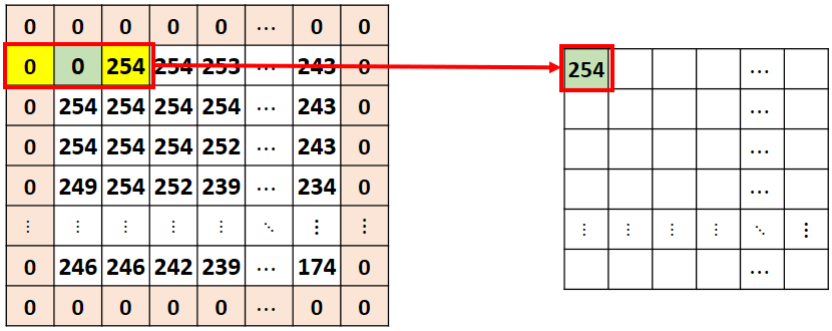
Gambar 4.34 Penambahan Zero Padding

Selanjutnya matriks pada Gambar 4.34 akan dilakukan proses *masking*. *Masking* pertama adalah *masking* terhadap sumbu x dengan hasil I_x. Simulasi perhitungannya dapat dilihat pada Gambar 4.35 dan Gambar 4.36. Contoh

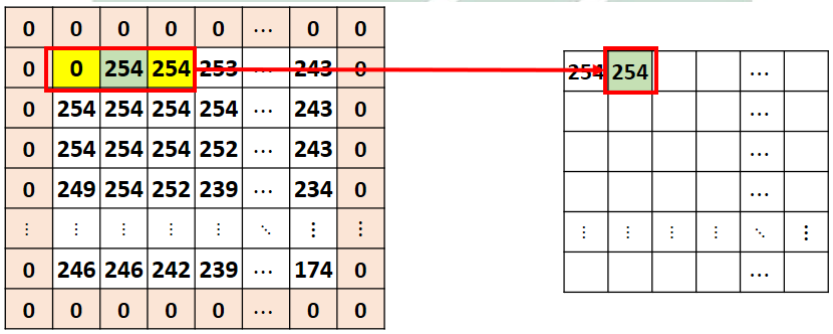
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

perhitungan *masking* terhadap sumbu x dapat dilihat dibawah ini:

$$
\begin{aligned}
I_x(1,1) &= (-1)0 + (0)0 + (1)254 \\
&= 254 \\
I_x(1,2) &= (-1)0 + (0)254 + (1)254 \\
&= 254 \\
I_x(1,3) &= (-1)254 + (0)254 + (1)253 \\
&= -1
\end{aligned}
$$



Gambar 4.35 *Masking* terhadap sumbu x pada t=1



Gambar 4.36 *Masking* terhadap sumbu x pada t=2

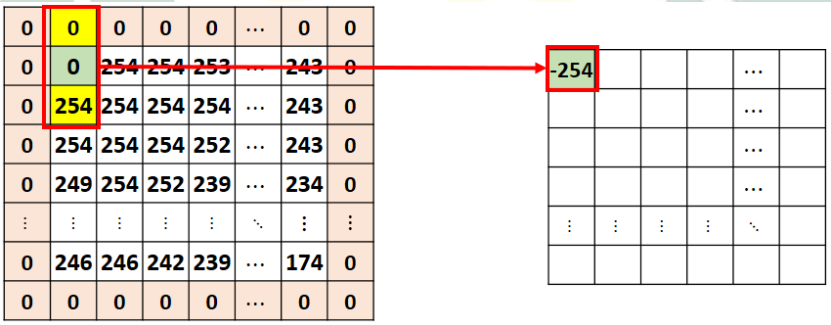
Perhitungan tersebut terus dilakukan hingga didapatkan keseluruhan nilai I_x yang dapat dilihat pada Gambar 4.37.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

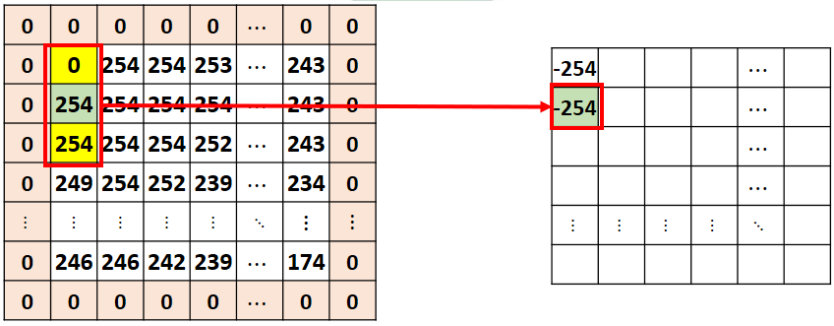
254	254	-1	-3	⋯	2
254	0	0	-2	⋯	4
254	0	-2	-9	⋯	4
254	3	-15	-21	⋯	2
⋮	⋮	⋮	⋮	⋱	1
3	9	15	14	10	0

Gambar 4.37 Hasil *Masking* terhadap sumbu x (I_x)

Untuk masking terhadap sumbu y perhitungannya hampir sama seperti sumbu x, perbedaannya ada pada cara jalan maskingnya. Pada sumbu x jalannya horizontal sedangkan pada sumbu y jalannya vertikal. Simulasi perhitungan *masking* terhadap sumbu y dapat dilihat pada Gambar 4.38 dan Gambar 4.39.



Gambar 4.38 *Masking* terhadap sumbu y pada t=1



Gambar 4.39 *Masking* terhadap sumbu y pada t=2

Sama halnya seperti I_x, perhitungan tersebut terus dilakukan hingga didapatkan keseluruhan nilai I_y yang dapat dilihat pada Gambar 4.40.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

-254	-254	-254	-254	⋯	-243
-254	0	0	1	⋯	0
5	0	2	15	⋯	9
5	5	17	21	⋯	51
⋮	⋮	⋮	⋮	⋱	⋮
-2	-2	-4	1	⋯	16

Gambar 4.40 Hasil *Masking* terhadap sumbu y (I_y)

3. **Perhitungan nilai *Gradien Magnitude* dan *Gradien Orientation***

Nilai I_x dan I_y yang telah didapatkan akan digunakan untuk menghitung nilai *gradient magnitude* dan *gradient orientation* yang ada pada Persamaan 2.6 dan Persamaan 2.5. Pada penelitian ini akan diberikan contoh perhitungan *gradient magnitude* yang dapat dilihat dibawah ini:

$$\begin{aligned} |G(1,1)| &= \sqrt{I_x(1,1)^2 + I_y(1,1)^2} \\ &= \sqrt{254^2 + (-254)^2} \\ &= \sqrt{129032} \\ &= 359.21 \approx 359 \end{aligned}$$

$$\begin{aligned} |G(1,3)| &= \sqrt{I_x(1,3)^2 + I_y(1,3)^2} \\ &= \sqrt{(-1)^2 + (-254)^2} \\ &= \sqrt{64517} \\ &= 254.01 \approx 254 \end{aligned}$$

Selanjutnya dilakukan perhitungan *gradient orientation* dengan contoh

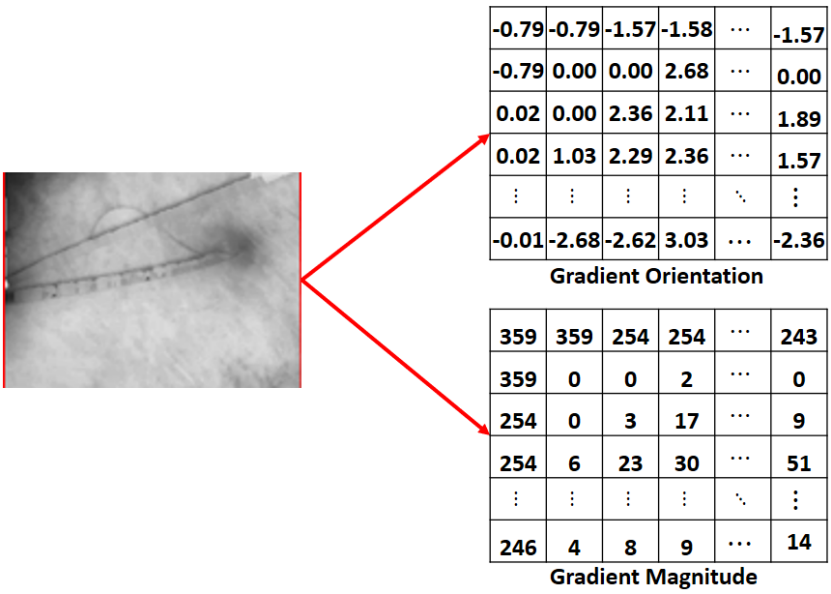
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

perhitungan dibawah ini:

$$
\begin{aligned}
\lambda(1,1) &= tan_2^{-1}\left(\frac{I_y(1,1)}{I_x(1,1)}\right) \\
&= \left[tan^{-1}\left(\frac{I_y(1,1)}{I_x(1,1)}\right)\right]\left(\frac{3.14}{180}\right) \\
&= \left[tan^{-1}\left(\frac{-254}{254}\right)\right]\left(\frac{3.14}{180}\right) \\
&= \left[tan^{-1}(-1)\right]\left(\frac{3.14}{180}\right) \\
&= [-45]\left(\frac{3.14}{180}\right) \\
&= -0.785 \\
\lambda(1,3) &= \left[tan^{-1}\left(\frac{I_y(1,3)}{I_x(1,3)}\right)\right]\left(\frac{3.14}{180}\right) \\
&= \left[tan^{-1}\left(\frac{-254}{-1}\right)\right]\left(\frac{3.14}{180}\right) \\
&= \left[tan^{-1}(254)\right]\left(\frac{3.14}{180}\right) \\
&= [-89.77]\left(\frac{3.14}{180}\right) \\
&= -1.566
\end{aligned}
$$

Perhitungan diatas hanyalah sekedar contoh perhitungan yang harus dilakukan. Perhitungan tersebut terus dilakukan hingga didapatkan seluruh nilai *gradient magnitude* dan *gradient orientation* dari seluruh piksel citra. Hasil perhitungan seluruh piksel dalam satu *block* dapat dilihat pada Gambar 4.41.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

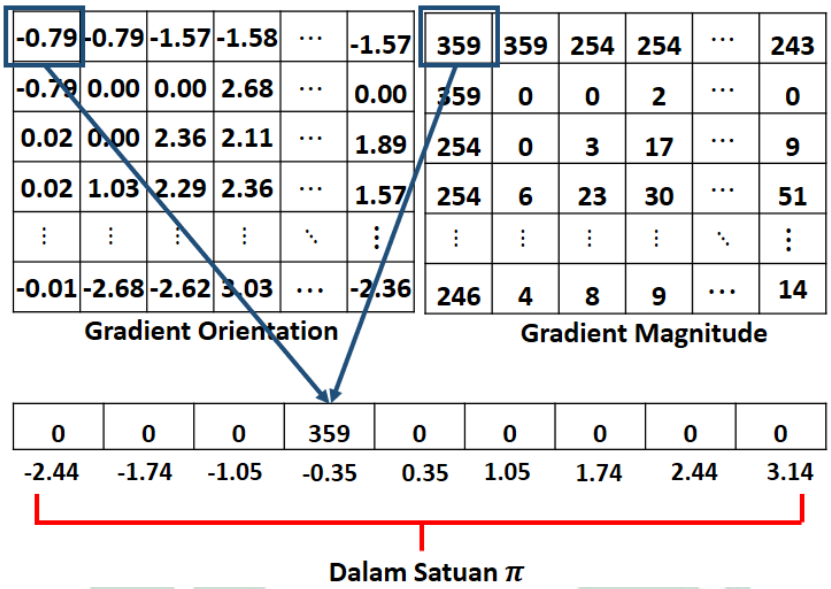


Gambar 4.41 ***Gradient Magnitude*** **dan** ***Gradient Orientation*** **pada** ***Block*** **Pertama**

4. **Memasukkan nilai ke dalam bin**

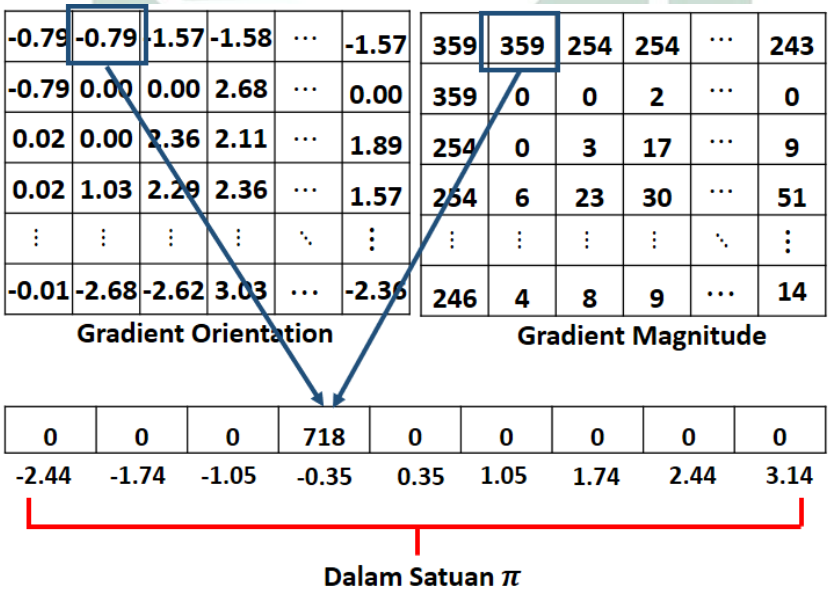
Proses selanjutnya adalah memasukkan nilai *gradient magnitude* kedalam *bin* sesuai dengan nilai *gradient orientation*. Nilai *gradient magnitude* dimasukkan dengan cara berjalan berurutan dari nilai *bin* paling awal. Ketika nilai gradient orientationnya kurang dari nilai *bin*, maka nilai *gradient magnitude*-nya ditambahkan ke dalam *bin* tersebut. Perhitungan lebih jelasnya dapat melihat Gambar 4.42. Pada Gambar tersebut terlihat bahwa nilai *gradient orientation* pada $t = 1$ adalah -0.79. Nilai bin yang memenuhi dengan syarat $gradient\ orientation < nilai\ bin$ adalah pada *bin* ke-4 yaitu -0.35. Hal ini tidak berlaku untuk *bin* ke-5 dengan nilai 0.35 dikarenakan sudah ditemukan nilai *bin* yang memenuhi pada urutan *bin* ke-4. Nilai magnitude pada *t=1* adalah 359 dan nilai tersebut dimasukkan kedalam bin ke-4 karena memenuhi syarat tersebut sehingga nilai *bin* ke-4 yang awalnya 0 ditambahkan dengan nilai *magnitude*-nya yaitu 359 menjadi 359.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.42 Memasukkan Nilai Bin pada t=1

Selanjutnya masuk pada $t = 2$ dengan nilai *magnitude* 359 dan orientasi -0.79. Nilai *bin* yang memenuhi adalah pada bin ke-4 sehingga nilai *bin* ke-4 yang awalnya 359 ditambahkan lagi dengan 359 menghasilkan 718 dan lebih jelasnya dapat dilihat pada Gambar 4.43.



Gambar 4.43 Memasukkan Nilai Bin pada t=2

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Perhitungan terus dilakukan sampai mendapatkan hasil seluruh *bin* dalam *block* pertama. Hasil akhir dari proses pemasukan *bin* pada *block* pertama dapat dilihat pada Gambar 4.44.

1294371	328208	75479	135344	177169	354434	71059	152439	135234
-2.44	-1.74	-1.05	-0.35	0.35	1.05	1.74	2.44	3.14

Dalam Satuan π

Gambar 4.44 Hasil Pemasukan Bin pada Block Pertama

5. **Normalisasi nilai pada bin**

Proses selanjutnya adalah normalisasi nilai hog pada tiap *block* dengan menggunakan Persamaan 2.7. Hal pertama yang harus dilakukan adalah mencari nilai norm dari bin pada *block* pertama.

$$\begin{aligned} Norm &= ||nilai\ bin|| \\ &= \sqrt{\sum_{i=1}^{9} nilai\ bin(i)^2} \\ &= \sqrt{1294371^2 + 328208^2 + \cdots + 135234^2} \\ &= \sqrt{2010718869815} \\ &= 1417998.19 \end{aligned}$$

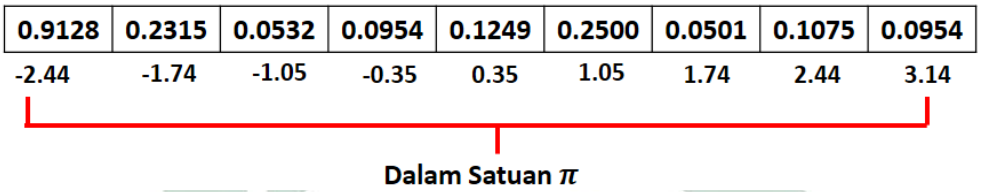
Langkah selanjutnya adalah menghitung nilai normalisasi pada nilai *bin*. Contoh perhitungan yang didapatkan dapat dilihat dibawah ini:

$$\begin{aligned} X(i) &= \frac{bin(i)}{Norm + 0.01} \\ X(1) &= \frac{1294371}{1417998.19 + 0.01} \\ &= 0.9128 \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
X(i) &= \frac{bin(i)}{Norm + 0.01} \\
X(2) &= \frac{328208}{1417998.19 + 0.01} \\
&= 0.2315
\end{aligned}
$$

Hasil akhir untuk ekstraksi fitur HOG pada block pertama dapat dilihat pada Gambar 4.45.



Gambar 4.45 Hasil Akhir *HOG Descriptor* pada *Block* Pertama

Hasil dari HOG suatu citra memiliki fitur sebanyak 81 yang dapat diuraikan untuk fitur tiap block-nya pada Tabel 4.1.

Tabel 4.1 Hasil Akhir *HOG Descriptor* Suatu Citra

Block	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	$\cdots$	Bin 9
Block 1	0.9128	0.2315	0.0532	0.0954	0.1249	$\cdots$	0.0954
Block 2	0.9262	0.1774	0.0886	0.1226	0.1295	$\cdots$	0.1036
Block 3	0.9113	0.2733	0.0718	0.1378	0.1023	$\cdots$	0.1063
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Block 9	0.9260	0.2351	0.0435	0.0969	0.0793	$\cdots$	0.1183

6. **Hasil akhir untuk semua citra**

Pada suatu sistem yang dibuat, digunakan sebuah citra sebanyak 2000 dengan 1000 data normal dan 1000 data kanker. Hasil dari proses ektraksi menggunakan *HOG Descriptor* dapat dilihat pada Tabel 4.2.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.2 Hasil Akhir *HOG Descriptor* Semua Citra Citra

Data	Fitur 1	Fitur 2	Fitur 3	Fitur 4	$\cdots$	Fitur 81	Kelas
Data 1	0.9149	0.2891	0.0287	0.1061	$\cdots$	0.0528	Normal
Data 2	0.8895	0.3169	0.0129	0.1488	$\cdots$	0.0680	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\cdots$	
Data 1000	0.9355	0.2478	0.0230	0.1083	$\cdots$	0.1271	
Data 1001	0.8923	0.2701	0.0476	0.0882	$\cdots$	0.0702	Kanker
Data 1002	0.9396	0.2135	0.0436	0.0848	$\cdots$	0.0874	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
Data 2000	0.9270	0.2806	0.0485	0.0930	$\cdots$	0.2412	

4.3. Deep Extreme Learning Machine

Fitur yang telah didapatkan akan dimasukkan dalam suatu sistem *Deep Extreme Learning Machine*. Proses *Deep Extreme Learning Machine* memiliki beberapa tahap dalam proses *learning*-nya. Tahapan *learning* pada *Deep Extreme Learning* meliputi Mendapatkan *Learned Parameter*, *K-Fold Cross Validation*, *Training*, dan *Testing*. Proses pertama adalah mendapatkan *learned parameter*. Tahap ini bertujuan untuk memperbaiki parameter yang sudah ada sehingga lebih mudah untuk mengenali pola. Metode yang dipakai dalam *learned parameter* adalah *Restricted ELM-AE* yang membentuk MLELM. Tahapan selanjutnya adalah *K-Fold Cross Validation* yang berguna untuk memisahkan data training dan data testing. Proses selanjutnya adalah proses *training* untuk mendapatkan nilai bobot output (β) dan dilanjutkan dengan proses *testing* untuk mendapatkan hasil klasifikasi.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.3.1. Mendapatkan *Learned Parameter*

Proses pertama adalah mendapatkan *learned parameter* dengan menggunakan *Restricted ELM-AE* yang membentuk MLELM. Sebagai contoh dalam perhitungan akan digunakan jumlah layer sebanyak 1000 dan 10 *hidden nodes* sehingga akan menggunakan aturan *compressed auto encoder*. Langkah pertama adalah membentuk nilai bobot awal (w) dan bias (b) dengan cara *random orthogonal* yaitu dengan ketentuan $w^T w = I$ dan $b^T b = 1$. Nilai w memiliki ukuran matriks sebesar 81 x 10 dikarenakan jumlah fitur *input* ada 81 dan jumlah *hidden nodes* yang dipakai adalah 10. Pada nilai b memiliki ukuran sebanyak *hidden nodes* sehingga nilai b memiliki ukuran sebesar 10 x 1.

$$w_{81\ x\ 10} = \begin{bmatrix} 0.1013 & 0.0241 & -0.0767 & 0.0285 & \cdots & 0.1983 \\ 0.0583 & 0.0802 & 0.3105 & -0.0477 & \cdots & 0.1199 \\ -0.1369 & 0.0378 & -0.1027 & 0.0212 & \cdots & 0.0943 \\ 0.0211 & -0.1756 & 0.0047 & -0.1272 & \cdots & 0.0363 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -0.1269 & -0.1000 & 0.0575 & 0.1514 & \cdots & -0.0370 \end{bmatrix} \quad b = \begin{bmatrix} -0.4584 \\ 0.4992 \\ -0.3737 \\ -0.3122 \\ \vdots \\ 0.4718 \end{bmatrix}$$

Setelah didapatkan nilai bobot dan bias, Nilai H akan dihitung dengan menggunakan persamaan 2.11 sehingga didapatkan:

$$\begin{aligned} H(i,j) &= g(x_{i,:} w_{:,j} + b_{j,1}) \\ H(1,1) &= g(x_{1,1:81} w_{1:81,1} + b_{1,1}) \\ &= sigmoid(x_{1,1:81} w_{1:81,1} + b_{1,1}) \\ &= sigmoid((0.9149 * 0.1013 + 0.2891 * 0.0583 + \cdots \\ &\quad +0.0528 * -0.1269) + (-0.4584)) \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
&= sigmoid(0.5523) + (-0.4584)) \\
&= sigmoid(0.0939) \\
&= \frac{1}{1+e^{-0.0939}} \\
&= 0.5235
\end{aligned}
$$

$$
\begin{aligned}
H(1,2) &= sigmoid(x_{1,1:81} w_{1:81,2} + b_{2,1}) \\
&= sigmoid((0.9149 * 0.0241 + 0.2891 * 0.0802 + \cdots \\
&\quad +0.0528 * -0.1000) + (-0.4992)) \\
&= sigmoid(-0.1222) + (-0.4992)) \\
&= sigmoid(-0.6214) \\
&= \frac{1}{1+e^{-(-0.6214)}} \\
&= 0.3495
\end{aligned}
$$

Perhitungan terus dilakukan sehingga mendapatkan matriks H berukuran 2000 x 10 dikarenakan fitur input yang dipakai penelitian ini adalah 2000 data. Matriks H yang didapatkan dapat dilihat dibawah ini:

$$
H_{2000\ x\ 10} = \begin{bmatrix}
0.5235 & 0.3495 & 0.5201 & 1.2947 & \cdots & -0.6071 \\
0.6169 & 0.4333 & 0.4908 & 1.1424 & \cdots & -0.4317 \\
0.5708 & 0.3729 & 0.4317 & 1.0659 & \cdots & -0.4762 \\
0.5992 & 0.2734 & 0.3713 & 0.7955 & \cdots & -0.3408 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0.6862 & 0.3171 & 0.3538 & 0.9553 & \cdots & -0.5388
\end{bmatrix}
$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Setelah nilai H didapatkan, maka proses selanjutnya adalah menghitung nilai output keluaran (β). Pada ELM-AE terdapat 3 persaman β yaitu *compressed*, *sparse*, dan *equal*. Pada penelitian ini tipe yang digunakan adalah *compressed* karena dimensi *input* lebih besar daripada dimensi *hidden nodes*. Pada *compressed ELM-AE* untuk mendapatkan nilai β dapat menggunakan Persamaan 2.35 dengan pemisalan nilai $C = 1$, I_n merupakan matriks identitas dengan ukuran 10 x 10, dan X merupakan hasil dari proses *HOG Descriptor*.

$$\beta = \left(H^T H + \frac{I_n}{C}\right)^{-1} H^T X$$

$$\beta = \left(\begin{bmatrix} 0.5235 & \cdots & 0.5992 \\ \vdots & \ddots & \vdots \\ -0.6071 & \cdots & -0.5388 \end{bmatrix} \begin{bmatrix} 0.5235 & \cdots & -0.6071 \\ \vdots & \ddots & \vdots \\ 0.5992 & \cdots & -0.5388 \end{bmatrix} + \frac{\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}}{1} \right)^{-1}$$

$$\begin{bmatrix} 0.5235 & \cdots & 0.5992 \\ \vdots & \ddots & \vdots \\ -0.6071 & \cdots & -0.5388 \end{bmatrix} \begin{bmatrix} 0.9149 & \cdots & 0.0528 \\ \vdots & \ddots & \vdots \\ 0.9270 & \cdots & 0.2412 \end{bmatrix}$$

$$\beta = \begin{bmatrix} 0.2215 & 0.0145 & \cdots & -0.0161 \\ 0.2208 & 0.0157 & \cdots & -0.0171 \\ \vdots & \cdots & \ddots & \vdots \\ 0.2208 & 0.0145 & \cdots & -0.0173 \end{bmatrix}$$

Nilai β yang didapatkan memiliki ukuran matriks sebesar 10 x 81. Proses selanjutnya adalah mencari nilai X_{new}. Contoh Perhitungan nilai X_{new} dapat dijabarkan dibawah ini:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
X_{new}(1,1) &= X_{1,1:81}\beta^T_{1:81,1} \\
&= (0.9149 * 0.2215 + 0.2891 * 0.0145 + \cdots + 0.0528 * -0.0161) \\
&= 6.2870 \\
X_{new}(1,2) &= X_{1,1:81}\beta^T_{1:81,2} \\
&= (0.9149 * 0.2208 + 0.2891 * 0.0157 + \cdots + 0.0528 * -0.0171) \\
&= 6.2776
\end{aligned}
$$

Perhitungan terus dilakukan sampai mendapatkan nilai X_new dengan ukuran 2000 x 10. Hasil matriks dari X_{new} dapat dilihat dibawah ini:

$$
X_{new} = \begin{bmatrix}
6.2870 & 6.2776 & 6.2798 & \cdots & 6.2496 \\
6.3301 & 6.3213 & 6.3228 & \cdots & 6.2986 \\
6.1553 & 6.1558 & 6.1566 & \cdots & 6.1585 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
6.3264 & 6.3170 & 6.3189 & \cdots & 6.3038
\end{bmatrix}
$$

Setelah mendapatkan nilai X_{new}, maka selanjutnya masuk kedalam layer pertama dalam MLELM. Proses pada MLELM hanya ada 3 yaitu perhitungan H_{new}, β_{new}, dan X_{new}. Hal pertama yang dilakukan adalah menghitung H_{new} dengan menggunakan Persamaan 2.38. Contoh perhitungan H_{new} dapat dilihat dibawah ini:

$$
\begin{aligned}
H_{new} &= g((\beta)^T H_{old}) \\
H_{new}(1,1) &= sigmoid(\beta^T_{1,1:10} H_{1:10,1})
\end{aligned}
$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
H_{new}(1,1) &= sigmoid(0.2215 * 0.5235 + 0.2208 * 0.6169 + \cdots \\
&\quad +0.2208 * 0.6862) \\
&= sigmoid(1.1415) \\
&= \frac{1}{(1+e^{-1.1415})} \\
&= 0.7580
\end{aligned}
$$

Perhitungan dilakukan samai mendapatkan nilai H_{new} dengan ukuran 81 x 2000 seperti dibawah ini:

$$
H_{new(81\ x\ 2000)} = \begin{bmatrix} 0.7580 & 0.7609 & 0.7572 & \cdots & 0.7587 \\ 0.5193 & 0.5195 & 0.5192 & \cdots & 0.5183 \\ 0.4310 & 0.4301 & 0.4313 & \cdots & 0.4308 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.4785 & 0.4782 & 0.4786 & \cdots & 0.4785 \end{bmatrix}
$$

Hasil dari H baru ini akan dipakai dalam perhitungan β_{new}. Persamaan yang dipakai untuk menghitung β_{new} adalah Persamaan 2.37. Nilai β_{new} ini juga nantinya akan dipakai dalam penentuan X_{new} sehingga nilai X selalu diperbaiki agar menjadi lebih baik dan lebih meningkatkan hasil klasifikasi. Contoh perhitungan β_{new} ada dibawah ini:

$$
\begin{aligned}
\beta_{new} &= H^{\dagger}X \\
\beta_{new} &= H^{T}(HH^{T})^{-1}X
\end{aligned}
$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$\beta_{new(10\ x\ 81)} = \begin{bmatrix} 0.7580 & \cdots & 0.4785 \\ \vdots & \ddots & \vdots \\ 0.7587 & \cdots & 0.4785 \end{bmatrix} \left(\begin{bmatrix} 0.7580 & \cdots & 0.7587 \\ \vdots & \ddots & \vdots \\ 0.4785 & \cdots & 0.4785 \end{bmatrix} \begin{bmatrix} 0.7580 & \cdots & 0.4785 \\ \vdots & \ddots & \vdots \\ 0.7587 & \cdots & 0.4785 \end{bmatrix} \right)^{-1}$$

$$\begin{bmatrix} 6.2870 & \cdots & 6.2496 \\ \vdots & \ddots & \vdots \\ 6.3264 & \cdots & 6.3038 \end{bmatrix}$$

$$= \begin{bmatrix} 0.3958 & 0.2710 & 0.2249 & \cdots & 0.2497 \\ 0.3953 & 0.2706 & 0.2246 & \cdots & 0.2494 \\ 0.3953 & 0.2707 & 0.2246 & \cdots & 0.2494 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.3941 & 0.2698 & 0.2239 & \cdots & 0.2486 \end{bmatrix}$$

Nilai β_{new} memiliki ukuran 10 x 81. Langkah terakhir pada layer pertama adalah menghitung nilai X_{new}. Cara perhitungannya sama seperti pada ELM-AE yaitu:

$$\begin{aligned} X_{new}(1,1) &= X_{1,1:81}\beta^T_{1:81,1} \\ &= (6.2870 * 0.3958 + 6.2776 * 0.2710 + \cdots + 6.2496 * 0.2497) \\ &= 6.1344 \\ X_{new}(1,2) &= X_{1,1:81}\beta^T_{1:81,2} \\ &= (6.2870 * 0.3953 + 6.2776 * 0.2706 + \cdots + 6.2496 * 0.2494) \\ &= 6.1338 \end{aligned}$$

Perhitungan terus dilakukan sampai mendapatkan nilai X_new dengan ukuran 2000 x 10. Hasil matriks dari X_{new} dapat dilihat dibawah ini:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$X_{new} = \begin{bmatrix} 6.1344 & 6.1338 & 6.1390 & \cdots & 6.2117 \\ 6.1145 & 6.2134 & 6.1131 & \cdots & 6.2109 \\ 6.1412 & 6.1543 & 6.1244 & \cdots & 6.2044 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 6.1121 & 6.2021 & 6.2110 & \cdots & 6.1921 \end{bmatrix}$$

Proses MLELM terus diulang sampai layer ke-1000. Pada Layer ke-1000 didapatkan hasil dari *learned parameter* adalah:

$$X_{2000\ x\ 10} = \begin{bmatrix} 0.4940 & 1.1539 & 1.0862 & \cdots & 0.4021 \\ 0.4941 & 1.1541 & 1.0863 & \cdots & 0.4021 \\ 0.4939 & 1.1536 & 1.0859 & \cdots & 0.4020 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.5371 & 1.2614 & 1.1772 & \cdots & 0.3112 \end{bmatrix}$$

Matriks hasil akhir dari proses *learned parameter* memiliki ukuran 2000 x 10 dikarenakan terdapat 2000 data yang fitur awalnya 81 dilakukan *compressed auto encoder* menjadi 10 parameter.

4.3.2. *K-Fold Cross Validation*

Data yang didapatkan dari proses *learned parameter* adalah 2000 data dengan 10 parameter. Untuk melakukan pengecekan terhadap evaluasi sistem, maka disarankan memakai *cross validation* agar sistem yang teruji diyakini kebenaran hasilnya. Pada penelitian ini digunakan *K-Fold Cross Validation* untuk memisahkan data *training* dan data *testing*. Nilai k yang digunakan pada penelitian ini adalah 5 sehingga data terbagi menjadi 80% *training* dan 20% *testing*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Penerapan metode ini menghasilkan pembagian data 1600 sebagai *training* dan 400 sebagai *testing* dengan rincian data *testing* 200 normal dan 200 kanker. Hasil pemisahan data *training* dan data *testing* dapat dilihat pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Sampel Data *Training*

Data	Fitur 1	Fitur 2	Fitur 3	Fitur 4	$\cdots$	Fitur 10	Kelas
Train 1	0.4940	1.1539	1.0867	0.5116	$\cdots$	0.4021	Normal
Train 2	0.4941	1.1539	1.0867	0.5118	$\cdots$	0.4021	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\cdots$	
Train 800	0.4937	1.1531	1.0854	1.5727	$\cdots$	0.4018	
Train 801	0.5212	1.2236	1.0087	1.7348	$\cdots$	0.3887	Kanker
Train 802	0.4997	1.1976	1.0773	1.0076	$\cdots$	0.3964	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
Train 1600	0.4908	1.0772	1.1146	1.2377	$\cdots$	0.4001	

Tabel 4.4 Sampel Data *Testing*

Data	Fitur 1	Fitur 2	Fitur 3	Fitur 4	$\cdots$	Fitur 10	Kelas
Test 1	0.4938	1.5332	1.0913	0.3327	$\cdots$	0.4022	Normal
Test 2	0.4996	1.1432	1.0897	0.3177	$\cdots$	0.4021	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\cdots$	
Test 200	0.4876	1.5533	1.0732	1.3269	$\cdots$	0.4020	
Test 201	0.5013	1.2346	1.0076	1.7684	$\cdots$	0.3848	Kanker
Test 202	0.5112	1.2875	1.1247	1.3447	$\cdots$	0.3927	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
Test 400	0.5134	1.3086	1.0587	1.4761	$\cdots$	0.3886	

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.3.3. Proses *Training*

Pada proses training digunakan metode KELM. Metode KELM memiliki 4 kernel yang bisa digunakan yaitu *linear*, *Gaussian*, *Polynomial*, dan *Wavelet*. Data yang digunakan pada proses *training* sebanyak 1600 data. Langkah awal proses *training* adalah dengan melakukan random bobot awal (w) dan random bias (b). Ukuran nilai w adalah 10 x 10 dan ukuran b sebesar 10 x 1 dikarenakan jumlah fitur input sebanyak 10 dan jumlah *hidden nodes*-nya 10. Nilai w dan b yang digunakan adalah:

$$w_{10\ x\ 10} = \begin{bmatrix} 0.2014 & 0.9483 & 0.0978 & \cdots & 0.7162 \\ 0.9361 & 0.9107 & 0.8094 & \cdots & 0.0983 \\ 0.5447 & 0.6278 & 0.5092 & \cdots & 0.5943 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.2798 & 0.6090 & 0.9374 & \cdots & 0.5733 \end{bmatrix} \quad b = \begin{bmatrix} 0.8251 \\ 0.2743 \\ 0.4700 \\ \vdots \\ 0.9579 \end{bmatrix}$$

Sama halnya seperti ELM-AE dan MLELM, setelah didapatkan nilai bobot dan bias maka nilai H akan dihitung dengan menggunakan persamaan 2.11 sehingga didapatkan:

$$\begin{aligned} H(i,j) &= g(x_{i,:}w_{:,j} + b_{j,1}) \\ H(1,1) &= g(x_{1,1:10}w_{1:10,1} + b_{1,1}) \\ &= sigmoid(x_{1,1:10}w_{1:10,1} + b_{1,1}) \\ &= sigmoid((0.4940 * 0.2014 + 1.1539 * 0.9361 + \cdots \\ &\quad +0.4021 * 0.2798) + (0.8251)) \\ &= sigmoid(4.8222) + (0.8251)) \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
&= sigmoid(5.6473) \\
&= \frac{1}{1+e^{-5.6473}} \\
&= 0.9965
\end{aligned}
$$

$$
\begin{aligned}
H(1,2) &= sigmoid(x_{1,1:10} w_{1:10,2} + b_{2,1}) \\
&= sigmoid((0.5212 * 0.9483 + 1.2236 * 0.9107 + \cdots \\
&\quad +0.3887 * 0.6090) + (0.2743)) \\
&= sigmoid(4.9600) + (0.2743)) \\
&= sigmoid(5.2343) \\
&= \frac{1}{1+e^{-5.2343}} \\
&= 0.9947
\end{aligned}
$$

Perhitungan terus dilakukan sehingga mendapatkan matriks H yang memiliki ukuran 1600 x 10 dikarenakan fitur input yang dipakai penelitian ini adalah 1600 data. Matriks H yang didapatkan dapat dilihat dibawah ini:

$$
H_{1600\ x\ 10} = \begin{bmatrix}
0.9965 & 0.9947 & 0.9969 & 0.9820 & \cdots & 0.9982 \\
0.9963 & 0.9938 & 0.9965 & 0.9896 & \cdots & 0.9978 \\
0.9959 & 0.9944 & 0.9958 & 0.9912 & \cdots & 0.9984 \\
0.9967 & 0.9948 & 0.9967 & 0.9850 & \cdots & 0.9974 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0.9964 & 0.9945 & 0.9962 & 0.9857 & \cdots & 0.9985
\end{bmatrix}
$$

Selanjutnya akan dicari nilai kernel dari H. Sebagai contoh akan dilakukan dengan menggunakan kernel *linear*. Diasumsikan $x_i = H^T$ dan $x_j = H$ sehingga jika

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menggunakan kernel linear akan menjadi:

$$\begin{aligned} linear &= x_i x_j \\ &= H^T H \end{aligned}$$

Berdasarkan rumus diatas maka nilai kernel K adalah:

$$\begin{aligned} K_{1600\ x\ 1600} &= H_{1600\ x\ 10} H^T_{10\ x\ 1600} \\ &= \begin{bmatrix} 0.9965 & 0.9947 & \cdots & 0.9982 \\ 0.9963 & 0.9938 & \cdots & 0.9978 \\ \vdots & \vdots & \ddots & \vdots \\ 0.9964 & 0.9945 & \cdots & 0.9985 \end{bmatrix} \begin{bmatrix} 0.9965 & 0.9963 & \cdots & 0.9964 \\ 0.9947 & 0.9938 & \cdots & 0.9945 \\ \vdots & \vdots & \ddots & \vdots \\ 0.9982 & 0.9978 & \cdots & 0.9985 \end{bmatrix} \\ &= \begin{bmatrix} 9.8575 & 9.8579 & \cdots & 9.8527 \\ 9.8664 & 9.8574 & \cdots & 9.8576 \\ \vdots & \vdots & \ddots & \vdots \\ 9.8573 & 9.8644 & \cdots & 9.8577 \end{bmatrix} \end{aligned}$$

Nilai matriks K memiliki ukuran sebesar 1600 x 1600 dikarenakan matriks H berukuran 1600 x 10 dan matriks H^T berukuran 10 x 1600. Langkah terakhir proses training adalah menghitung nilai β_{output} dengan menggunakan Persamaan 2.28. Nilai T merupakan nilai traget klasifikasi yang berupa matriks baris. Pada penelitian ini target data normal adalah 0 dan data kanker adalah 1. Nilai T memiliki ukuran sebesar 1600 x 1. sehingga perhitungannya menjadi:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
\beta &= K\left(\frac{I}{C}H^T H\right)^{-1} T \\
&= \begin{bmatrix} 9.8575 & \cdots & 9.8527 \\ \vdots & \ddots & \vdots \\ 9.8573 & \cdots & 9.8577 \end{bmatrix} \cdots \\
&\quad \left(\frac{\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}}{1} \begin{bmatrix} 0.9965 & \cdots & 0.9964 \\ \vdots & \ddots & \vdots \\ 0.9982 & \cdots & 0.9985 \end{bmatrix} \begin{bmatrix} 0.9965 & \cdots & 0.9982 \\ \vdots & \ddots & \vdots \\ 0.9964 & \cdots & 0.9985 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 0.1450 & 0.6583 & \cdots & 0.1720 \\ 0.9772 & 0.1640 & \cdots & 0.1527 \\ \vdots & \vdots & \ddots & \vdots \\ 0.6979 & 0.1101 & \cdots & 0.8637 \end{bmatrix}
\end{aligned}
$$

4.3.4. Proses *Testing*

Pada proses testing juga dilakukan suatu perhitungan yang sama seperti testing perbedaannya hanya pada Persamaan akhir. Pada training bertujuan untuk mencari nilai β dan proses testing untuk mencari nilai T sehingga persamaannya berubah menjadi:

$$
T = K\left(\frac{I}{C}H^T H\right)\beta
$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Dikarenakan perhitungannya sama, maka tidak akan ditampilkan proses perhitungannya dan untuk hasil beberapa data dapat dilihat pada Tabel 4.5.

Tabel 4.5 Sampel Hasil *Testing*

Data	Target Aktual	Target Hasil
Test 1	Normal	Normal
Test 2	Normal	Normal
⋮	⋮	⋮
Test 186	Normal	Kanker
⋮	⋮	⋮
Test 372	Kanker	Normal
⋮	⋮	⋮
Test 399	Kanker	Kanker
Test 400	Kanker	Kanker

4.4. Evaluasi dan Validasi Model

Hasil dari proses *learning* akan dilakukan proses validasi dengan menggunakan metode *confusion matrix*. Penggunaan *confusion matrix* berguna untuk menentukan keberhasilan suatu sistem klasifikasi. *Confusion matrix* sendiri memiliki 3 parameter keluaran yaitu akurasi, sensitifitas, dan spesifisitas. Semakin tinggi nilai yang dihasilkan maka akan semakin tinggi pula tingkat keberhasilkan dari sistem tersebut. Hasil *confusion matrix* dari hasil klasifikasi kanker kulit dengan metode DELM menggunakan kernel linear dan 10 *hidden nodes* dapat dilihat pada Tabel 4.6. *True positif* adalah data *malignant* teridentifikasi *malignant*, *False Negatif* adalah data *malignant* teridentifikasi *benign*, *False Negatif* adalah data *benign* teridentifikasi *malignant*, *True Negatif* adalah data

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

benign teridentifikasi *benign*,

Tabel 4.6 *Confusion Matrix* pada Kernel linear dengan 10 nodes

		Data Prediksi	
		Positif	False
Data	Positif	181	19
Aktual	Negatif	64	136

Pada tabel tersebut terlihat 136 dari 200 data *benign* terklasifikasi *benign* dan 181 dari 200 data pasien *malignant* terklasifikasi *malignant*. Berdasarkan hasil *confusion matrix* yang dapat dilihat pada Tabel 4.6 dapat dihitung nilai akurasi, sensitifitas, dan spesifisitas dengan perhitungan dibawah ini dengan menggunakan Persamaan 2.45, Persamaan 2.46, dan Persamaan 2.47.

$$
\begin{aligned}
Akurasi &= \frac{TP+TN}{TP+FN+FP+TN} \\
&= \frac{136+181}{136+64+19+181}100\% \\
&= 79.25\% \\
Sensitifitas &= \frac{TN}{FP+TN} \\
&= \frac{181}{10+181}100\% \\
&= 90.50\% \\
Spesifisitas &= \frac{TP}{TP+FN} \\
&= \frac{136}{136+64}100\% \\
&= 68\%
\end{aligned}
$$

Pada proses DELM memiliki rumus *output* yang berbeda jika jumlah input (n_i) dan jumlah *hidden nodes* (n_h) berbeda. Terdapat 3 rumus yaitu ketika $n_i > n_h$, $n_i = n_h$,

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dan $n_i < n_h$ sehingga akan dilakukan uji coba berdasarkan parameter tersebut dan juga berdasarkan fungsi kernel yang dipakai

4.4.1. Hasil Evaluasi $n_i > n_h$

Nilai n_i pada sistem adalah 81 sehingga untuk ujicoba $n_i > n_h$ akan menggunakan jumlah hidden nodes sebesar 10 dan 50. Hasil untuk percobaan dengan hidden nodes 10 dapat dilihat pada Tabel 4.7.

Tabel 4.7 Sampel Hasil *Testing* dengan nodes 10

Kernel	Validasi	Nilai	Waktu Training
Linear	Akurasi	79.25 %	4 menit 18 detik
	Spesifisitas	68 %	
	Sensitifitas	90.5%	
Gaussian	Akurasi	77%	5 menit 36 detik
	Spesifisitas	67.5%	
	Sensitifitas	86.5%	
Polynomial	Akurasi	80.5%	6 menit 12 detik
	Spesifisitas	70%	
	Sensitifitas	91%	
Wavelet	Akurasi	76%	5 menit 06 detik
	Spesifisitas	65.5%	
	Sensitifitas	86.5%	

Pada percobaan dengan menggunakan 10 hidden nodes didapatkan hasil terbaik ada pada kernel polynomial dengan nilai akurasi sebesar 80.5% dengan waktu training 6 menit 12 detik. Pada penerapan kernel polynomial terdapat nilai spesifisitas yang buruk. Hal tersebut menjelaskan bahwa sistem kurang bisa

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

mengenali pasien dengan label benign. Beberapa pasien *benign* teridentifikasi *malignant*, namun nilai sensitifitasnya dapat dikatakan baik sehingga sistem bisa mengenali penderita kanker dengan baik dan pasien dapat segera dilakukan pengobatan secara dini. Uji coba selanjutnya adalah dengan menggunakan hidden nodes sebanyak 50 dengan hasil yang dapat dilihat pada Tabel 4.8.

Tabel 4.8 Sampel Hasil *Testing* dengan nodes 50

Kernel	Validasi	Nilai	Waktu Training
Linear	Akurasi	80.25%	8 menit 59 detik
	Spesifisitas	73%	
	Sensitifitas	87.5%	
Gaussian	Akurasi	83.5%	10 menit 03 detik
	Spesifisitas	81%	
	Sensitifitas	86%	
Polynomial	Akurasi	85.5%	10 menit 49 detik
	Spesifisitas	79%	
	Sensitifitas	92%	
Wavelet	Akurasi	78.25%	9 menit 56 detik
	Spesifisitas	76.5%	
	Sensitifitas	80%	

Pada uji coba ini, hasil yang didapatkan hampir sama seperti pada 10 *hidden nodes*. Perbedaannya ada pada nilai akurasi, sensitifitas, dan spesifikasi. Pada *hidden nodes* 50 didapatkan nilai akurasi sebesar 85.5% dengan waktu training 10 menit 49 detik. Akurasi tersebut didapatkan pada kernel yang sama yaitu polynomial dengan nilai sensitifitas yang juga tinggi. Sehingga dapat disimpulkan bahwa dengan kondisi $n_i > n_h$ didapatkan hasil tertinggi dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menggunakan kernel polynomial.

4.4.2. Hasil Evaluasi $n_i = n_h$

Jumlah input yang ada pada sistem adalah 81 sehingga untuk mendapatkan kondisi $n_i = n_h$, maka diperlukan hidden nodes sebanyak 81 juga. Hasil uji coba dengan beberapa kernel terhadap 81 hidden nodes dapat dilihat pada Tabel 4.9.

Tabel 4.9 Sampel Hasil *Testing* dengan nodes 81

Kernel	Validasi	Nilai	Waktu Training
Linear	Akurasi	80.5%	12 menit 27 detik
	Spesifisitas	77%	
	Sensitifitas	84%	
Gaussian	Akurasi	80.5%	13 menit 48 detik
	Spesifisitas	77.5%	
	Sensitifitas	83.5%	
Polynomial	Akurasi	73.75%	15 menit 38 detik
	Spesifisitas	67.5%	
	Sensitifitas	80%	
Wavelet	Akurasi	58.5%	14 menit 17 detik
	Spesifisitas	52.5%	
	Sensitifitas	64.5%	

Pada uji coba dengan menggunakan $n_i = n_h$ dengan jumlah *hidden nodes* sebanyak 81 didapatkan hasil akurasi terbaik didapatkan pada kernel linear dan gaussian dengan nilai akurasi sebesar 80.5% dan waktu training 12 menit 27 detik dan 13 menit 48 detik. Perbedaan antara keduanya ada pada nilai sensitifitas dan spesifisitasnya. Berdasarkan pertimbangan nilai spesifisitas dan sensitifitas, maka

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

didapatka hasil terbaik ada pada kernel linear dikarenakan nilai dari sensitifitasnya tinggi sehingga dapat mengenali pasien kanker kulit dengan baik.

4.4.3. Hasil Evaluasi $n_i < n_h$

Uji coba dengan kondisi $n_i < n_h$ akan menggunakan hidden nodes sebanyak 100 dan 300. Hal ini dilakukan untuk menguji kernel dan jumlah *hidden nodes* yang tepat untuk melakukan klasifikasi. Uji coba dengan menggunakan 100 hidden nodes dapat dilihat pada Tabel 4.10

Tabel 4.10 Sampel Hasil *Testing* dengan nodes 100

Kernel	Validasi	Nilai	Waktu Training
Linear	Akurasi	78.75%	15 menit 48 detik
	Spesifisitas	70.5%	
	Sensitifitas	87%	
Gaussian	Akurasi	82%	17 menit 11 detik
	Spesifisitas	75%	
	Sensitifitas	89%	
Polynomial	Akurasi	86.5%	18 menit 56 detik
	Spesifisitas	82.5%	
	Sensitifitas	90.5%	
Wavelet	Akurasi	70.75%	17 menit 03 detik
	Spesifisitas	66.5%	
	Sensitifitas	75%	

Hasil uji coba dengan menggunakan 100 *hidden nodes* mendapatkan hasil akurasi terbesar yaitu 86.5% pada kernel polynomial dengan waktu training 18 menit 56 detik. Hasil ini lebih bagus jika dibandingkan dengan uji coba

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sebelumnya. Nilai sensitifitas dan spesifisitas juga memiliki nilai yang baik, namun pada spesifisitas masih memiliki nilai dibawah 90% sehingga akan dilakukan uji coba kembali untuk mendapatkan hasil yang lebih baik. Uji coba selanjutnya adalah menggunakan *hidden nodes* sebanyak 300. Hasil uji coba tersebut dapat dilihat pada Tabel 4.11

Tabel 4.11 Sampel Hasil *Testing* dengan nodes 300

Kernel	Validasi	Nilai	Waktu Training
Linear	Akurasi	78.75%	22 menit 32 detik
	Spesifisitas	67.5%	
	Sensitifitas	90%	
Gaussian	Akurasi	85%	25 menit 18 detik
	Spesifisitas	84%	
	Sensitifitas	86%	
Polynomial	Akurasi	91.5 %	27 menit 23 detik
	Spesifisitas	91%	
	Sensitifitas	92%	
Wavelet	Akurasi	72.25%	25 menit 41 detik
	Spesifisitas	72.5%	
	Sensitifitas	72%	

Uji coba dengan 300 hidden nodes memiliki akurasi terbaik sebesar 91.5% dengan waktu training 27 menit 23 detik. Nilai akurasi ini sudah dalam kategori sangat baik karena sudah melebihi 90% sehingga sudah dapat mengklasifikasikan dengan baik. Nilai sensitifitas dan spesifitasnya juga memiliki nilai diatas 90%. Oleh karena itu, berdasarkan nilai-nilai uji coba hasil terbaik ada pada kernel polynomial dengan jumlah *hidden nodes* sebanyak 300. Uji coba juga dilakukan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengan nilai *hidden nodes* sebanyak 500, namun karena hasil yang jelek maka hanya ditampilkan sampai uji coba *hidden nodes* ke 300.

4.5. Tampilan Aplikasi

Aplikasi ini merupakan aplikasi yang digunakan untuk mendeteksi kanker kulit berdasarkan keganasannya yaitu *benign* dan *malignant*. Hasil keluaran dari aplikasi ini berupa keputusan mengenai pasien tersebut berpeluang memiliki kanker kulit atau tidak.

4.5.1. *Loading Screen*

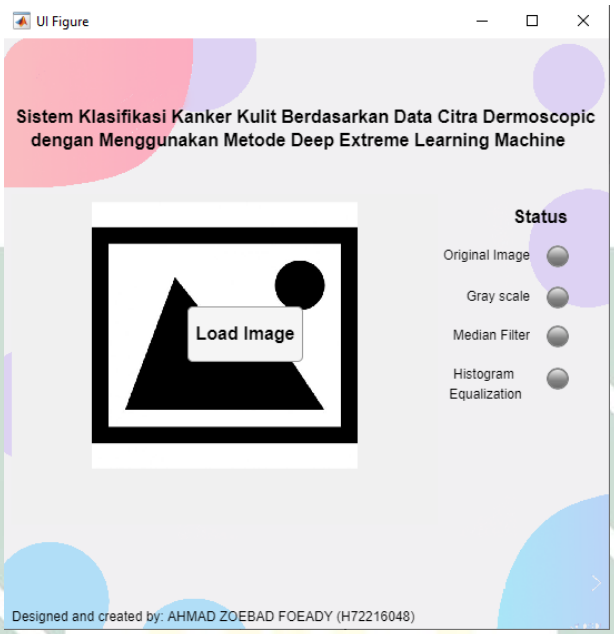
Gambar 4.46 *Loading Screen* Aplikasi

Loading screen pada Gambar 4.46 merupakan tampilan awal ketika *user* membuka aplikasi ini. Pada layar ini berisikan informasi mengenai judul yang digunakan. Proses *loading* membutuhkan waktu sekitar 10 menit untuk masuk ke halaman utama aplikasi. *Loading screen* diberikan untuk memberikan efek jeda dan memperkenalkan aplikasi kepada *user* terlebih dahulu sebelum masuk ke

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dalam layar utama aplikasi.

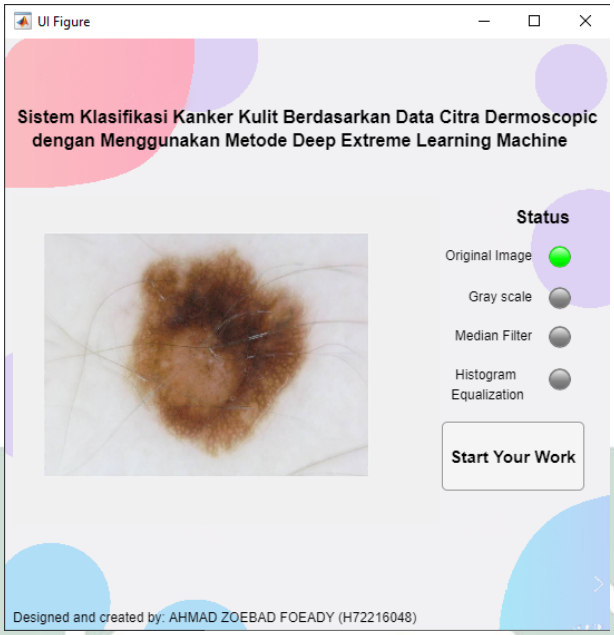
4.5.2. Layar Utama



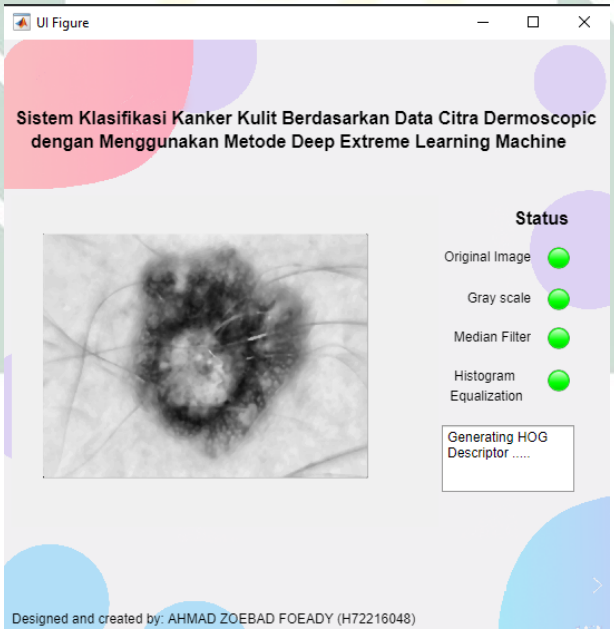
Gambar 4.47 Layar Utama Aplikasi

Layar utama pada Gambar 4.47 hanya berisikan satu buah tombol yaitu tombol *upload file*. Proses upload akan dilakukan dengan mengambil gambar yang ada pada komputer. Setelah memasukkan gambar, maka lampu pada *original image* akan menyala berwarna hijau dan terdapat suatu tombol yang muncul untuk memulai proses. Contoh aplikasi setelah di input gambar dapat dilihat pada 4.48. Ketika tombol tersebut ditekan, maka aplikasi akan menjalankan tugasnya untuk menghasilkan *pre-processing* yang dipakai. Hasil akhir dari *pre-processing* merupakan suatu citra grayscale yang memiliki ketajaman yang diperbaiki dengan menggunakan *histogram equalization*. Contoh hasil akhir pre-processing pada aplikasi dapat dilihat pada Gambar 4.49.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



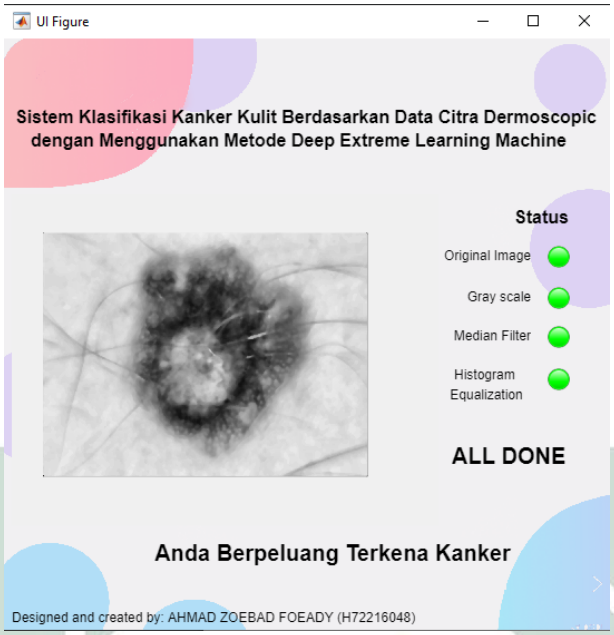
Gambar 4.48 ***Input Image***



Gambar 4.49 Hasil Pre-processing pada Aplikasi

Setelah semua proses selesai maka aplikasi akan memunculkan jawaban berupa suatu keputusan untuk mengetahui penyakit kanker kulit. Pembentukan aplikasi ini berguna bagi beberapa orang yang tidak bisa pergi ke dokter. Hasil akhir dari aplikasi dapat dilihat pada Gambar 4.50.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.50 Hasil Akhir pada Aplikasi

Terlihat bahwa pada Gambar 4.50 bahwa hasil yang dikeluarkan hanya berupa suatu keputusan. Ketika teridentifikasi kanker kulit maka akan menyatakan pesan ”Anda berpeluang terkena kanker kulit” dan ketika teridentifikasi *benign* maka akan muncul pesan ”Anda tidak terkena kanker kulit”.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB V

PENUTUP

5.1. Simpulan

Berdasarkan hasil penelitian mengenai klasifikasi kanker kulit dengan menggunakan metode *Deep Extreme Learning Machine* (DELM) dapat disimpulkan bahwa:

1. Banyaknya *hidden nodes* mempengaruhi hasil dari klasifikasi citra. Hasil akurasi terbaik berturut-turut pada jumlah *hidden nodes* $n_i > n_h$, $n_i = n_h$, dan $n_i > n_h$ adalah 85.5%, 80.5%, dan 91.5%. Semakin banyak *hidden nodes* juga tidak berarti bahwa hasil yang didapatkan baik seperi halnya pada kasus $n_i > n_h$ dengan $n_i = n_h$. Pada kasus tersebut lebih bagus hasil akurasi dengan *hidden nodes* yang lebih rendah, namun pada kasus $n_i < n_h$ dengan menaikkan jumlah *hidden nodes* didapatkan hasil yang lebih baik. Pada penelitian ini dapat disimpulkan untuk menentukan hasil yang baik dibutuhkan penentuan jumlah *hidden nodes* yang sesuai. Pada penelitian ini penentuan *hidden nodes* yang paling baik adalah dengan menggunakan kondisi $n_i < n_h$.

2. Model terbaik yang didapatkan untuk mengklasifikasikan kanker kulit dengan menggunakan *Deep Extreme Learning Machine* adalah pada kernel polynomial. Hampir pada tiap jumlah *hidden nodes* yang berbeda nilai terbaik selalu ada pada kernel polynomial kecuali pada kasus $n_i = n_h$. Jumlah hidden nodes yang sesuai untuk klasifikasi kanker kulit pada uji coba

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

yang telah dilakukan adalah sebanyak 300, sehingga dapat disimpulkan bahwa model terbaik adalah dengan menggunakan kernel polynomial dengan jumlah *hidden layer* sebanyak 1000 dan *hidden nodes* sebanyak 300 menghasilkan nilai akurasi sebesar 91.5%, spesifisitas sebesar 92%, dan nlai sensitifitas sebesar 91% dengan waktu training 27 menit 23 detik.

5.2. Saran

Penelitian mengenai klasifikasi citra dermoscopic unruk mendeteksi kanker kulit dengan menggunakan metode DELM masih banyak kekurangan. Berdasarkan beberapa kekurangan yang ada, peneliti menyarankan beberapa hal sebagai bahan perbaikan dan pengembangan kedepannya, diantaranya adalah:

1. Melakukan uji coba terhadap fungsi aktifasi pada DELM.
2. Menemukan pre-processing yang baik sehingga mendapatkan nilai akurasi yang baik.
3. Menggunakan metode feature descriptor yang lain.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR PUSTAKA

A, Suresh;K, S. (2012). An Efficient Texture Classification System Based On Gray Level Co- Occurrence Matrix. *International Journal Of Computer Science And Information Technology dan Security*, 2(4), 793–798.

Albadra, M. A. A., dan Tiuna, S. (2017). Extreme learning machine: a review. *International Journal of Applied Engineering Research*, 12(14), 4610-4623.

Adarsh, P., dan Jeyakumari, D. (2013). Multiclass Svm-Based Automated Diagnosis Of Diabetic Retinopathy. *International Conference On Communications And Signal Processing*, 206–210. Https://Doi.Org/10.1109/Iccsp.2013.6577044

Adyanti, D. A., Asyhar, A. H., Novitasari, D. C. R., Lubab, A., dan Hafiyusholeh, M. (2017). Forecasts Marine Weather On Java Sea Using Hybrid Methods: Ts-Anfis. *International Conference On Electrical Engineering, Computer Science And Informatics (EECSI)*, 4(September), 492–497. Https://Doi.Org/10.11591/Eecsi.4.1114

Adyanti, D. A (2018). *Analisis Citra Dental Panoramic Radiograph (Dpr) Pada Tulang Mandibula Untuk Deteksi Osteoporosis Menggunakan Metode Glcm – Svm Multiclass (Gray Level Co- Occurrence Matrix – Support Vector Machine Multiclass)*. UIN Sunan Ampel Surabaya.

Ahuja, Y., dan Kumar Yadav, S. (2012). Multiclass Classification And Support Vector Machine. *Global Journal Of Computer Science*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

And Technology Interdisciplinary, 12(11), 14–19. Retrieved From Https://Globaljournals.Org/GjcstVolume12/2-Multiclass-Classification-And.Pdf

AlAsadi, A. H. H., dan AL-Safy, B. M. R. (2016). *Early detection and classification of melanoma skin cancer*. LAP LAMBERT Academic Publishing.

Ali, S. K. (2018). Diagnosis Retinopathy Disease Using GLCM And ANN. *Journal of Theoretical and Applied Information Technology (JATIT)*, 6028-6040.

American Cancer Society. (2007). *Melanoma Skin Cancer Early Detection, Diagnosis, And Staging*, 1–21. American Cancer Society Report.

Al-Amin, M., Alam, M. B., dan Mia, M. R. (2015). Detection of Cancerous and Non-cancerous Skin by using GLCM Matrix and Neural Network Classifier. *International Journal of Computer Applications*, 132(8), 44.

Anggoro, W. (2016). Implementasi Ekstraksi Fitur Tekstur Gray Level Co-Occurrence Matrices (GLCM) untuk Pengelompokkan Citra Tenun Menggunakan Algoritma K-Means. *Semarang: Jurusan Teknik Informatika, FIK UDINUS*.

Atika, R., Suciat, F. N., dan Navastara, D. A. (2016). Implementasi Metode Kombinasi Histogram of Oriented Gradient Dan Hierarchical Centroid Untuk Sketch Based Image Retrieval. *Jurnal Teknis ITS*.

Berkant, B. (2015). Implementation Of HOG Edge Detector Algorithm Onfpga's. *Procedia - Social and Behavioral Science*, 1567–1575.

Bino, S. V. (2012). Grey Level Co-Occurrence Matrices: Generalisation And Some New Features. *International Journal Of Computer*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Science, Engineering And Information Technology, 2(2), 151–157. Https://Doi.Org/10.5121/Ijcseit.2012.2213

Biswas, S., dan Ghoshal, D. (2016). Blood Cell Detection Using Thresholding Estimation Based Watershed Transformation with Sobel Filter in Frequency Domain. *Procedia Computer Science*, 89, 651–657.https://doi.org/10.1016/j.procs.2016.06.029

Brantsch, K. D., Meisner, C., Schönfisch, B., Trilling, B., Wehner-Caroli, J., Röcken, M., dan Breuninger, H. (2008). Analysis of risk factors determining prognosis of cutaneous squamous-cell carcinoma: a prospective study. *The lancet oncology*, 9(8), 713-720.

Cahyono, W. E. (2010). Urgensi Menjaga Lapisan Ozon Bagi Penghuni Bumi. *Berita Dirgantara*, 8(2).

Cancer Council. (2016). Understanding Skin Cancer. *SOS Print and Media Group*.

Cerna, L. R., dan Menotti, D. (2010). Cerna, L. R., Cámara-Chávez, G., dan Menotti, D. (2013). Face detection: Histogram of oriented gradients and bag of feature method. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Chen, C., Li, W., Su, H., dan Liu, K. (2014). Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine. *Remote Sensing*, 6(6), 5795-5814.

Chen, W. R., Bin, J., Lu, H. M., Zhang, Z. M., dan Liang, Y. Z. (2016). Calibration

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

transfer via an extreme learning machine auto-encoder. *Analyst*, 141(6), 1973-1980.

Choudhari, S., dan Biday, S. (2014). Artificial Neural Network for SkinCancer Detection. *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, 3(5), 147–153.

Churchill, M. (2013). Histogram of Oriented Gradients for Detection of Multiple Scene. *International Conference on Computer Vision and Pattern Recognition*.

Dalal, N., dan Triggs, B. (2012). Histograms of Oriented Gradients for Human Detection.*International Conference on Computer Vision and Pattern Recognition*, 886–893.

Deng, W. Y., Ong, Y. S., dan Zheng, Q. H. (2016). A fast reduced kernel extreme learning machine. *Neural Networks Journals*, 76, 29-38.

Ding, S., Zhang, N., Xu, X., Guo, L., dan Zhang, J. (2015). Deep extreme learning machine and its application in EEG classification. *Mathematical Problems in Engineering*.

Dongping, L. (2018). Automatic Detection Of Cardiovascular Disease Using Deep Kernel Extreme, 30(4), 1–10. https://doi.org/10.4015/S1016237218500382

Duan, L., Bao, M., Miao, J., Xu, Y., dan Chen, J. (2016). Classification Based on Multilayer Extreme Learning Machine for Motor Imagery Task from EEG signals. *Procedia - Procedia Computer Science*, 88, 176–184. https://doi.org/10.1016/j.procs.2016.07.422

Emmons, K. M., Geller, A. C., Puleo, E., Savadatti, S. S., Hu, S. W., Gorham, S., ... dan Group, D. F. S. C. S. (2011). Skin cancer education and early detection at the

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

beach: a randomized trial of dermatologist examination and biometric feedback. *Journal of the American Academy of Dermatology*, 64(2), 282-289.

Fayaz, M., dan Kim, D. (2018). A prediction methodology of energy consumption based on deep extreme learning machine and comparative analysis in residential buildings. *Electronics Journal*, 7(10), 222.

Fradkin, D., dan Muchnik, I. (2006). Support vector machines for classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 70, 13-20.

Fuchs, E., dan Raghavan, S. (2002). Getting under the skin of epidermal morphogenesis. *Nature Reviews Genetics*, 3(3), 199.

Gao, Z., Ma, C., She, Z., dan Dong, X. U. (2018). An Enhanced Deep Extreme Learning Machine for Integrated Modular Avionics Health State Estimation. *IEEE Access*, 6, 65813–65823. https://doi.org/10.1109/ACCESS.2018.2878813

Gopinath, N. (2011). Graph Based Image Segmentation Method for Identification of Cancer in Prostate MRI Image. *Journal of Computer Applications (JCA)*, 4(4), 2011.

Guo, X., Pang, Y., Yan, G., dan Qiao, T. (2017, May). Time series forecasting based on deep extreme learning machine. In *2017 29th Chinese Control And Decision Conference (CCDC)* (pp. 6151-6156). IEEE.

Hamdi, S. (2018). Object Detection Using Histogram, *Al Akhawayn University*. 1–52.

Hoshyar, A. N. dan Al-Jumaily, A. (2014). The beneficial techniques in

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

preprocessing step of skin cancer detection system comparing. *Procedia Computer Science*, 42(C), 25–31. https://doi.org/10.1016/j.procs.2014.11.029

Huang, A., Kwan, S. Y., Chang, W. Y., Liu, M. Y., Chi, M. H., dan Chen, G. S. (2013). A robust hair segmentation and removal approach for clinical images of skin lesions. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, (July), 3315–3318. https://doi.org/10.1109/EMBC.2013.6610250

Huang, G. B., Zhu, Q., dan Siew C. K. (2006). Extreme Learning Machine: theory and Applications. *Journal Neurocomputing*, 70(1): 489-501

Huang, W., Li, N., Lin, Z., Huang, G. B., Zong, W., Zhou, J., dan Duan, Y. (2013, July). Liver tumor detection and segmentation using kernel-based extreme learning machine. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 3662-3665). IEEE.

Ibrahim, W., dan Abadeh, M. S. (2017). Extracting features from protein sequences to improve deep extreme learning machine for protein fold recognition. *Journal of theoretical biology*, 421, 1-15.

ISIC Archive. (2018). The International Skin Imaging Archive. Retrieved 17 August 2019, from isic-archive.com/!/topWithHeader/wideContentTop/main

Ismaeel, S., Miri, A., dan Chourishi, D. (2015, May). Using the extreme learning machine (ELM) technique for heart disease diagnosis. In *2015 IEEE Canada International Humanitarian Technology Conference (IHTC2015)* (pp. 1-3). IEEE.

Johnson, T. M., Dolan, O. M., Hamilton, T. A., Lu, M. C., Swanson, N. A., dan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Lowe, L. (1998). Clinical and histologic trends of melanoma. *Journal of the American Academy of Dermatology*, 38(5), 681-686.

Jong, W. D. (2005). Kanker, Apakah Itu? Pengobatan, Harapan Hidup, dan Dukungan Keluarga, *Jakarta: Arcan*.

Kutlu, Y. (2018). Superiorities of Deep Extreme Learning Machines against Convolutional Neural Networks. *Natural and Engineering Sciences*, 103-109.

Le, B. T., dan Xiao, D. (2018). Coal Exploration Based on a Multilayer Extreme Learning Machine and Satellite Images. *IEEE Access*, PP(c), 1. https://doi.org/10.1109/ACCESS.2018.2860278

Li, B., Rong, X., dan Li, Y. (2014). An improved kernel based extreme learning machine for robot execution failures. *The Scientific World Journal*.

Liu, D., dan Yu, J. (2009). Otsu method and K-means. *Proceedings - 2009 9th International Conference on Hybrid Intelligent Systems, HIS 2009,* 1(2), 344–349. https://doi.org/10.1109/HIS.2009.74

Mallick, S. (2016). Histogram Of Oriented Gradients. Retrieved 9 September 2019, from https://www.learnopencv.com/histogram-of-oriented-gradients/

Mao, W., He, J., Li, Y., dan Yan, Y. (2017). Bearing fault diagnosis with auto-encoder extreme learning machine: A comparative study. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(8), 1560-1578.

Meiviana, A., Sulistiowati, D. R., dan Soejachmoen, M. H. (2004). Bumi Makin Panas Ancaman Perubahan Iklim Di Indonesia. *Jakarta: Kementrian Lingkungan Hidup Republik Indonesia*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Mohanaiah, P., Sathyanarayana, P., dan Gurukumar, L. (2013). Image Texture Feature Extraction Using Glcm Approach. *International Journal Of Scientific dan Research Publication*, 3(5), 1–5. Https://Doi.Org/10.1.1.414.96981

Møller, R., Reymann, F., dan Hou-Jensen, K. (1979). Metastases in dermatological patients with squamous cell carcinoma. *Archives of dermatology*, 115(6), 703-705.

Mustafa, S., Dauda, A. B., dan Dauda, M. (2017, October). Image Processing and SVM classification for melanoma detection. In *2017 International Conference on Computing Networking and Informatics (ICCNI)* (pp. 1-5). IEEE.

National Cancer Institute (2006). What You Need To Know About Melanoma And Other Skin Cancer. *National Cancer Institute*.

Ngafifi, M., dan Ngafifi, M. (2014). Advances in Technology and Patterns of Human Life in Socio-Cultural Perspective. *J. Pembang. Pendidik. Fondasi dan Apl*, 2(1), 33-47.

Öztürk, Ş., dan Akdemir, B. (2018). Application Of Feature Extraction And Classification Methods For Histopathological Image Using Glcm, Lbp, Lbglcm, Glrlm And Sfta. *Procedia Computer Science*, 132(June), 40–46. Https://Doi.Org/10.1016/J.Procs.2018.05.057

Pandjaitan, L. W. (2007). Dasar-Dasar Komputasi Cerdas. Yogyakarta:Andi

Pathak, B., dan Barooah, D. (2013). Texture Analysis Based On The Gray-Level Co-Occurence Martix Considering Possible Orientations. *International Journal Of Advanced Research In Electrical, Electronics*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

And Instrumentation Engineering, 2(9), 4206–4212. Retrieved From Http://Www.Ijareeie.Com/Upload/2013/September/7-Texture.Pdf

Radha R, Rajendiran P (2013). Using K-Means Clustering Technique To Study Of Breast Cancer, *Congress On Computing and Communication Technologies*.

Rejani, Y., dan Selvi, S. T. (2009). Early detection of breast cancer using SVM classifier technique. *arXiv preprint arXiv*:0912.2314.

Roul, R. K. (2018). Detecting spam web pages using multilayer extreme learning machine. *International Journal of Big Data Intelligence*, 5(1-2), 49-61.

Roul, R. K., Rohan, S., dan Gaurav, A. (2016). Study on suitability and importance of multilayer extreme learning machine for classification of text data. *Soft Computing*. https://doi.org/10.1007/s00500-016-2189-8

Santi, C. N. (2011). Mengubah Citra Berwarna Menjadi GrayScale dan Citra biner. *Jurnal Teknologi Informasi Dinamik*, 16(1), 14-19.

Satheesha, T. Y., Satyanarayana, D., dan Giriprasad, M. N. (2014). A Pixel Interpolation Technique for curved hair removal in skin images to support melanoma detection. *Journal of Theoretical and Applied Information Technology*, 70(3), 559–565.

Scherer, D., dan Kumar, R. (2010). Genetics of pigmentation in skin cancer—a review. *Mutation Research/Reviews in Mutation Research*, 705(2), 141-153.

Setiawan, D. (2018). Dampak Perkembangan Teknologi Informasi dan Komunikasi Terhadap Budaya. *JURNAL SIMBOLIKA: Research and Learning in Communication Study*, 4(1), 62-72.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Sreekanth, M. S., Rajesh, R., dan Satheeshkumar, J. (2015). Extreme learning machine for the classification of rainfall and thunderstorm. *Journal of Applied Sciences*, 15(1), 153–156. https://doi.org/10.3923/jas.2015.153.156

Sun, K., Zhang, J., Zhang, C., dan Hu, J. (2016). Generalized Extreme Learning Machine Autoencoder And A New Deep Neural Network. *Neurocomputing*, 1–22.https://doi.org/10.1016/j.neucom.2016.12.027

Lee, T., Ng, V., Gallagher, R., Coldman, A., dan McLean, D. (1997). Dullrazor®: A software approach to hair removal from images. *Computers in biology and medicine*, 27(6), 533-543.

Inasawa, S., Suzuki, R., Qian, E. W., Kitajima, T., dan Yamashita, Y. (2011). Ozone layer depletion and its effects: a review. *International Journal of Environmental Science and Development*, 2, 30-37.

Tim Dosen. (2016). *Pengolahan Citra*. Jakarta: Universitas Mercu Buana.

Vázquez, D., Lopez, A., dan Ponsa, D. (2013). *Domain adaptation of virtual and real worlds for pedestrian detection* (Doctoral dissertation, Dissertation, Universitat de Barcelona, Spain).

Visa, S., Ramsay, B., Ralescu, A., dan Van Der Knaap, E. (2011). Confusion Matrix-Based Feature Selection. *Ceur Workshop Proceedings*, 710, 120–127. Https://Doi.Org/10.1.1.666.8961

Wahyudi, E., Triyanto, D., dan Ruslianto, I. (2017). Identifikasi Teks Dokumen Menggunakan Metode Profile Projection dan Template Matching. *Jurnal Coding, Sistem Komputer Untan*, 03(2), 1-10.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Wang, F., Zhang, B., Chai, S., dan Xia, Y. (2018). Community detection in complex networks using deep auto-encoded extreme learning machine. *Modern Physics Letters B*, 32(16), 1850180.

Wang, X., dan Han, M. (2014, July). Multivariate time series prediction based on multiple kernel extreme learning machine. In *2014 international joint conference on neural networks (IJCNN)* (pp. 198-201). IEEE.

Wen, X., Liu, H., Yan, G., dan Sun, F. (2018). Weakly paired multimodal fusion using multilayer extreme learning machine. *Soft Computing*, 22(11), 3533-3544.

World Health Organization. (2016). Skin Cancers. Retrieved 13 August 2019, from https://www.who.int/uv/faq/skincancer/en/index1.html

Wulandini, F., dan Nugroho, A. N. (2009). Text Classification Using Support Vector Machine for Web mining Based Spation Temporal Analysis of the Spread of Tropical Diseases. In *International Conference on Rural Information and Communication Technology* (pp. 189-192).

Yousefi, J. (2011). Image Binarization using Otsu Thresholding Algorithm. *University of Guelph, Ontario, Canada.*

Zaqout, I. S. (2016). Diagnosis of Skin Lesions Based on Dermoscopic Images Using Image Processing Techniques. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 9(9), 189–204. https://doi.org/10.14257/ijsip.2016.9.9.18

Zhao, D., Huang, C., Wei, Y., Yu, F., Wang, M., dan Chen, H. (2016). An Effective Computational Model for Bankruptcy Prediction Using Kernel Extreme Learning

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Machine Approach. *Computational Economics*. https://doi.org/10.1007/s10614-016-9562-7

Zhu, Y., dan Huang, C. (2012). An Improved Median Filtering Algorithm for Image Noise Reduction. *Physics Procedia*, 25, 609–616. https://doi.org/10.1016/j.phpro.2012.03.133

Zulpe, N., dan Pawar, V. (2012). Glcm Textural Features For Brain Tumor Classification. *International Journal Of Computer Science*, 9(3), 354–359. Retrieved From Http://Www.Doaj.Org/Doaj?Func=AbstractId=1158398

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id