

# IDENTIFIKASI INDIVIDU BERDASARKAN SKETSA WAJAH MENGUNAKAN PENDEKATAN DISKRIMINATIF

## SKRIPSI

Untuk memenuhi sebagian persyaratan  
Memperoleh gelar Sarjana Komputer

Disusun Oleh:  
Harits Abdurrohman  
NIM: 145150200111078



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

IDENTIFIKASI INDIVIDU BERDASARKAN SKETSA WAJAH MENGGUNAKAN  
PENDEKATAN DISKRIMINATIF

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh :  
Harits Abdurrohman  
NIM: 145150200111078

Skripsi ini telah diuji dan dinyatakan lulus pada  
13 Juli 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Agus Wahyu Widodo, S.T, M.Cs  
NIP. 19740805 200112 1 001



Dr. Anto Satriyo Nugroho  
NIP. 197010211989111001

Mengetahui,  
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D  
NIP. 19710518 200312 1 001



## PERNYATAAN ORISINALITAS

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

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

Malang, 28 Juni 2018



Harits Abdurrohman

NIM: 145150200111078



## KATA PENGANTAR

Dengan menyebut nam Allah SWT Yang Maha Pengasih Lagi Maha Penyayang. Syukur Alhamdulillah tak henti-hentinya penulis panjatkan atas nikmat dan rahmat yang diberikan olehNya. Berkat pertolongan, kasih sayang dan izin Yang Kuasa, penulis dapat menyelesaikan tugas akhir skripsi dengan judul "Identifikasi Individu Menggunakan Sketsa Wajah dengan Pendekatan Diskriminatif" tepat waktu. Tak luput pula shalawat serta salam penulis haturkan kepada Nabi Muhammad SAW, keluarga beserta para sahabat karena tanpa adanya ajaran Islam, ilmu pengetahuan tidak akan berkembang seperti sekarang ini.

Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan dukungan baik materi maupun moral serta bantuan dalam menyelesaikan skripsi ini. Pihak-pihak tersebut antara lain:

1. Bapak Agus Wahyu Widodo, S.T, M. Cs selaku dosen pembimbing I yang telah memberikan bimbingan, dukungan moril, ilmu dan masukan dalam penyelesaian skripsi ini.
2. Bapak Dr. Anto Satriyo Nugroho selaku dosen pembimbing II dan pakar biometrik dari PUI Biometrik yang juga telah banyak memberikan bimbingan, dukungan moril, ilmu dan masukan dalam penyusunan skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D., selaku ketua jurusan Teknik Informatika.
4. Bapak Mochammad Hannats Hanafi Ichsan, S.ST, M.T dan Ibu Ari Kusyanti, S.T, M.Sc selaku Dosen Pembimbing Akademik yang telah memberikan arahan dan motivasi selama perkuliahan.
5. Bapak dan Ibu dosen yang telah mendidik dan memberikan ilmu selama penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Segenap karyawan di Fakultas Ilmu Komputer Universitas Brawijaya yang membantu Penulis dalam pelaksanaan skripsi ini.
7. Keluarga dari penulis terutama kedua orang tua, Dwi Handoko dan Emmi Ruslianti, serta adik-adik penulis, Hasan Abdullah, M. Hafizh Mustaqim, M. Husaini Rahmatullah dan Hanifah Fadilah Handianti yang telah membantu penulis dari semua do'anya, motivasi, kasih sayang maupun dukungan materil semata-mata untuk keberhasilan penulis.
8. Seluruh teman-teman angkatan 2014, terkhusus teman-teman dari kelas MABA F yang tidak bisa tuliskan satu per satu namanya telah

- mengisi hari-hari di masa perkuliahan dengan belajar bersama baik dalam kelas maupun organisasi
9. Kelompok belajar F-Zoo: Egi Muliandri, Rayza Arfian, Rizky Haryandi Rahman, S. Kom, Chandra Yogi Adhitama, Rahmat Yani, Imada Nur Afifah, S. Kom, Ratna Tri Utami, Silvia Aprilla Lubis dan Wanda Athira Luqyana, S. Kom yang telah mau menemani belajar bersama selama perkuliahan dan berbagi cerita
  10. Teman-teman dari komunitas *Basic Computing Community* yang telah bersama mengembangkan *softskill* dan pengalaman belajar yang luar biasa selama perkuliahan.
  11. Teman-teman bimbingan skripsi, Restu Widodo dan Talitha Raissa yang telah menemani mengerjakan skripsi bersama serta praktikan penulis Ummi Kalsum A. Jusuf dan Chairunnisa Yusuf yang telah mau memberi kepercayaan untuk membantu di sela pengerjaan skripsi.
  12. Teman-teman dari ruang H2.2: Doni Putra Purbawa, Yulius Firantoko, Imam Ghozali, S. Kom, Andi Hafidh, Ivan Putra Pratama, S. Kom, Irnayanti Dwi Kusuma dan teman-teman lain yang tidak bisa saya tuliskan satu per satu yang telah menjadi teman diskusi dan berbagi ilmu
  13. Kepada ketua laboratorium *Computer Vision* Fakultas Ilmu Komputer Universitas Brawijaya, Ibu Fitri Utaminingrum, Dr. Eng., S.T, M.T beserta anggotanya, Bapak Randy Cahya Wihandika, S.ST., M.Kom, Bapak Putra Putra Pandu Adikara, S.Kom, M.Kom, Bapak Moch Ali Fauzi, S.Kom, M.Kom, Bapak Dahniel Syauqy, S.T., M.T., M.Sc. ,Bapak Sigit Adinugroho, S.Kom., M.Sc, Ibu Yuita Arum Sari, Ibu Anggi Gustiningsih Hapsani, S.Kom, Tahajuda Mandariansah, S.Kom, Tusty Nadia Maghfira, S. Kom dan Widya Cahyaningrum, S. Kom yang telah memberikan motivasi, ilmu serta dukungan selama pengerjaan skripsi

Semoga jasa dan amal baik kita semua mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran senantiasa penulis harapkan dari berbagai pihak untuk kesempurnaan skripsi ini.

Malang, 28 Juni 2018

Penulis  
Harits.abdr@gmail.com

## ABSTRAK

Mengidentifikasi seseorang berdasarkan cirinya merupakan salah satu ranah pada ruang lingkup biometrik. Pada ruang lingkup biometrik, setiap individu mempunyai karakteristik unik yang menjadi ciri khasnya seperti sidik jari, mata, bentuk telinga maupun wajah. Fungsi dari biometrik ini dapat digunakan untuk melakukan autentikasi terhadap sistem maupun mengenali seseorang berdasarkan salah satu cirinya. Biometrik ini kemudian dibagi menjadi kategori fisik dan kategori perilaku. Pada bidang forensik, ciri biometrik yang dapat digunakan salah satunya adalah ciri wajah dalam bentuk sketsa. Ada berbagai cara untuk mengekstraksi fitur pada citra dengan menggunakan tekstur analisis namun dalam penelitian ini, metode tekstur analisis yang digunakan adalah menggunakan SIFT (*Scale Invariant Feature Transform*) yang telah terbukti mampu digunakan pada ruang lingkup *heterogeneous* dan MCWLD (*Multiscale Circular Weber Local Descriptor*) yang berbasis pada hukum Weber, yakni melihat dari ratio suatu perubahan intensitas *pixel* dengan *pixel* ketetanggannya. Dari penelitian, dilakukan pengujian parameter baik pada metode SIFT maupun MCWLD. Pada SIFT diuji nilai ukuran *window* sedangkan pada MCWLD dilakukan pengujian nilai *T-dominant orientation*, *M-Segment*, dan *S-bin*. Penelitian ini juga melakukan fusi antara kedua metode dari tingkat fitur dan *score*. Hasil terbaik pada penelitian ini adalah untuk metode SIFT, konfigurasi terbaik adalah ukuran *window* kecil 8 dan *sliding window* 32 *overlapping* dengan nilai *identification rate* 79.79% sedangkan untuk MCWLD dengan konfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$  dengan nilai *identification rate* 82.45% dan Fusi SIFT dengan MCWLD terbaik pada tingkat *score* dengan nilai *identification rate* 82.45%. Meski nilai MCWLD lebih baik dari SIFT pada *top-match*, namun pada keseluruhan hasil atau *top-rank* metode SIFT melampaui dari MCWLD.

Kata kunci: biometrik, *Scale Invariant Feature Transform*, *Multiscale Circular Weber Local Descriptor*, *heterogenous matching*, *sketch recognition*, *multimodal biometric*

## ABSTRACT

Identifying a person based on his or her character is one of the domains in the biometrics. In the scope of biometrics, each individual has unique characteristics that characterize such as fingerprints, eyes, ear or face shape. The function of this biometric can be used to authenticate the system or recognize a person based on one of its characteristics. These biometrics are then divided into physical categories and behavioral categories. In the field of forensics, the characteristics of biometrics that can be used one of them is the characteristic of the face in the form of sketches. There are various ways to extract the features in the image by using the texture of the analysis but in this study, the analytical texture method used is to use the SIFT (Scale Invariant Feature Transform) which has proven to be used in the heterogeneous and MCWLD (Multiscale Circular Weber Local Descriptor) based on Weber's law, for example seeing from the ratio of a change in pixel intensity or its neighborhood. From the research, we do a parameter test both on SIFT method and MCWLD. In SIFT tested the value of window size while at the MCWLD testing the value of T-dominant orientation, M-segment, and S-bin. This research also performs fusion algorithm between both method of feature level and score. The best result in this research is for SIFT method, best configuration is small window size 8 and sliding window 32 overlapping with 79.79% identification rate while for MCWLD with sliding window configuration with window size 16 and overlapping and parameter used is  $T = 6$ ,  $M = 4$  and  $S = 3$  with 82.45% identification rate and SIFT Fusion with the best MCWLD at the score level with an identification rate of 82.45%. Although MCWLD's identification rate are better than SIFT on top-match, but on the overall result or top-rank, SIFT's identification over-perform MCWLD.

**Keywords:** *biometrics, Scale Invariant Feature Transform, Multiscale Circular Weber Local Descriptor, heterogeneous matching, sketch recognition, multimodal biometric*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR KODE SUMBER.....	xiv
DAFTAR LAMPIRAN .....	xv
Bab 1 Pendahuluan .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian .....	4
1.5 Batasan Masalah .....	4
1.6 Sistematika Pembahasan.....	5
Bab 2 Tinjauan Kepustakaan .....	6
2.1 Kajian pustaka.....	6
2.2 Landasan pustaka .....	9
2.2.1 Citra Digital .....	9
2.2.2 Citra Abu-abu .....	9
2.2.3 Sistem Biometrik.....	9
2.2.4 Pengukuran Performa Pengenalan Wajah Biometrik.....	11
2.2.5 <i>Close-set Identification</i> .....	12
2.2.6 <i>Within-between class variance ratio</i> .....	12
2.2.7 <i>Face Identification</i> .....	13
2.2.8 Tingkatan Fitur Wajah.....	14
2.2.9 <i>Face Sketch Recognition</i> .....	15
2.2.10 <i>Heterogeneous Matching</i> .....	16
2.2.11 <i>Local Invariant Features Method</i> .....	16
2.2.12 <i>Sliding window</i> .....	17

2.2.13 <i>Min-Max Normalization</i> .....	18
2.2.14 <i>Scale Invariant Feature Transform (SIFT)</i> .....	19
2.2.15 <i>Multiscale Circular Weber Local Descriptor (MCWLD)</i> .....	23
Bab 3 Metodologi Penelitian.....	28
3.1 Alur Metode Penelitian .....	28
3.2 Studi Literatur .....	29
3.3 Pengumpulan Data .....	29
3.4 Perancangan Algoritme .....	29
3.5 Implementasi Algoritme .....	30
3.6 Pengujian Algoritme dan analisis .....	31
3.7 Kesimpulan dan Saran .....	31
Bab 4 Perancangan.....	32
4.1 Perancangan Algoritme .....	32
4.1.1 <i>Preprocessing</i> .....	33
4.1.2 Ekstraksi Fitur.....	37
4.1.3 Pemadanan dan n-rank hasil .....	60
4.2 Perhitungan Manual.....	61
4.2.1 <i>Pre-Processing: Grayscaleing</i> .....	61
4.2.2 SIFT.....	66
4.2.3 MCWLD .....	74
4.3 Perancangan Skenario Pengujian .....	85
Bab 5 Implementasi .....	87
5.1 Lingkungan Implementasi .....	87
5.1.1 Lingkungan Hardware .....	87
5.1.2 Lingkungan Software .....	87
5.2 Batasan Implementasi .....	88
5.3 Implementasi Aplikasi .....	88
5.3.1 Pre-Processing .....	88
5.3.2 Implementasi Ekstraksi Fitur .....	90
Bab 6 Pengujian Dan Analisis .....	101
6.1 Pengujian .....	101
6.1.1 Pengujian Pengaruh <i>Overlapping</i> dan <i>Non-overlapping</i> .....	102
6.1.2 Pengujian Parameter pada SIFT dan MCWLD.....	103



6.1.3 Pengujian Fusi SIFT dan MCWLD .....	111
6.1.4 Analisis Hasil Pengujian.....	122
Bab 7 Kesimpulan Dan Saran .....	136
7.1 kesimpulan .....	136
7.2 Saran.....	137
DAFTAR PUSTAKA.....	138



## DAFTAR TABEL

Tabel 2.1 Pendekatan pada Heterogeneous Matching .....	6
Tabel 2.2 Kajian Pustaka .....	7
Tabel 2.3 Jenis Sketsa .....	15
Tabel 4.1 Nilai biru dari Gambar 4.19 .....	62
Tabel 4.2 Nilai hijau dari Gambar 4.19 .....	63
Tabel 4.3 Nilai merah dari Gambar 4.19 .....	64
Tabel 4.4 Hasil perhitungan <i>grayscale</i> .....	65
Tabel 4.5 Potongan <i>window</i> dari indeks 0 hingga 3 .....	66
Tabel 4.6 Penambahan kolom setelah kolom terakhir .....	67
Tabel 4.7 Penambahan kolom sebelum kolom pertama .....	67
Tabel 4.8 Penambahan baris setelah baris terakhir .....	68
Tabel 4.9 Penambahan baris sebelum baris pertama .....	68
Tabel 4.10 Hasil perhitungan besaran gradien dan pembobotan .....	70
Tabel 4.11 Hasil perhitungan ekstraksi orientasi dalam derajat .....	71
Tabel 4.12 Hasil ekstraksi orientasi dan pembobotan .....	72
Tabel 4.13 Hasil <i>histogram binning</i> .....	73
Tabel 4.14 Nilai histogram MCWLD .....	81
Tabel 6.1 Pengaruh <i>Overlapping</i> dan <i>non-overlapping</i> pada metode MCWLD ..	102
Tabel 6.2 Pengaruh <i>overlapping</i> dan <i>non-overlapping</i> pada metode SIFT .....	102
Tabel 6.3 Pengujian Parameter SIFT pada konfigurasi <i>window</i> 16 dan <i>overlapping</i> (Baseline) .....	103
Tabel 6.4 Pengujian Parameter SIFT pada semua konfigurasi <i>sliding window</i> ..	103
Tabel 6.5 Pengujian Parameter MCWLD pada konfigurasi <i>window</i> 16 dan <i>overlapping</i> (Baseline) .....	105
Tabel 6.6 Pengujian Parameter MCWLD pada semua konfigurasi <i>sliding window</i> .....	106
Tabel 6.7 Pengujian Fusi Tingkat Fitur SIFT dan MCWLD pada semua konfigurasi <i>sliding window</i> dan parameter .....	111
Tabel 6.8 Nilai <i>pixel</i> pada potongan Foto .....	126
Tabel 6.9 Nilai <i>pixel</i> pada potongan Sketsa .....	126

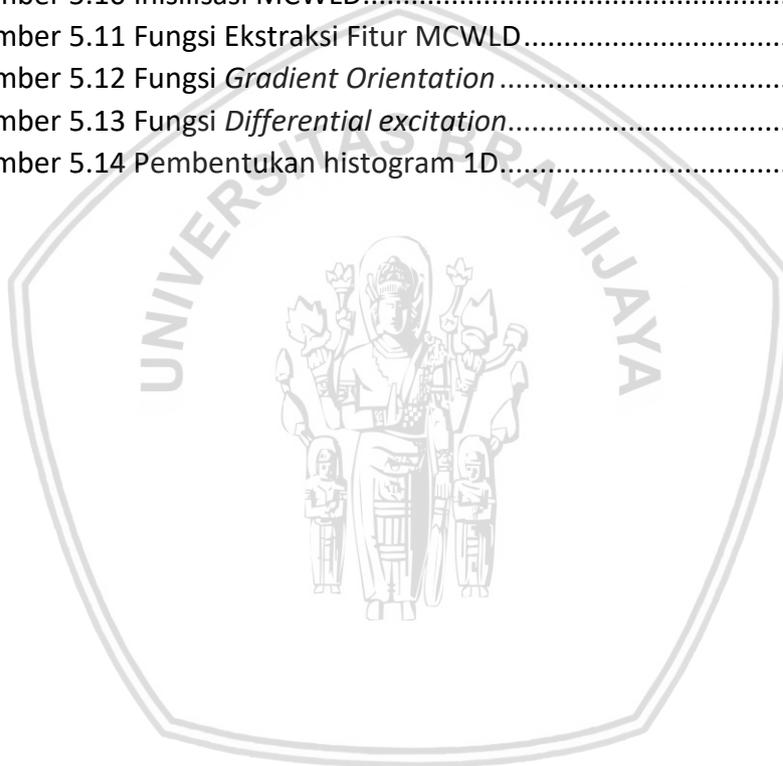
## DAFTAR GAMBAR

Gambar 2.1 Tahapan enrollment dan pengenalan pada sistem biometrik. sumber (Jain, et al., 2007) .....	11
Gambar 2.2 Grafik <i>similarity frequency distribution</i> .....	13
Gambar 2.3 Ilustrasi <i>Sliding window</i> .....	17
Gambar 2.4 Ilustrasi <i>overlapping Window</i> .....	18
Gambar 2.5 <i>Min-max normalization</i> pada fusi tingkat fitur (Jain, et al., 2007).....	19
Gambar 2.6 Pembobotan dengan <i>gaussian weighting</i> .....	21
Gambar 2.7 <i>Histogram Binning</i> .....	21
Gambar 2.8 Representasi perhitungan orientasi.....	22
Gambar 2.9 Ilustrasi fitur SIFT.....	22
Gambar 2.10 Histogram 2-D dari <i>Differential excitation</i> dan orientasi dari citra wajah. Sumber (Bhatt, et al., 2012) .....	25
Gambar 2.11 Sub-histogram dari differential excitaion di orientasi T. Sumber (Bhatt, et al., 2012) .....	25
Gambar 2.12 Matriks histogram. Sumber: (Bhatt, et al., 2012) .....	26
Gambar 3.1 Diagram alir metode penelitian .....	28
Gambar 3.2 Arsitektur Sistem .....	30
Gambar 4.1 Diagram alir Algoritme .....	32
Gambar 4.2 Diagram alir <i>preprocessing</i> .....	33
Gambar 4.3 Diagram Alir <i>grayscaleing</i> .....	34
Gambar 4.4 Diagram Alir <i>Sliding window</i> dengan <i>Overlapping Window</i> .....	36
Gambar 4.5 Diagram Alir Ekstraksi Fitur .....	37
Gambar 4.6 Diagram Alir Ekstraksi Fitur SIFT .....	38
Gambar 4.7 Diagram alir mengambil <i>window</i> kecil .....	40
Gambar 4.8 Diagram Alir Ekstraksi Fitur .....	41
Gambar 4.9 Diagram alir ekstraksi nilai orientasi .....	43
Gambar 4.10 Diagram alir ekstraksi nilai besaran gradien dan pembobotan .....	44
Gambar 4.11 Diagram Alir menambah Padding .....	46
Gambar 4.12 Diagram alir <i>Histogram Binning</i> .....	49
Gambar 4.13 Diagram Alir WLD .....	51
Gambar 4.14 Diagram alir membagi <i>window</i> .....	53
Gambar 4.15 Diagram alir <i>differential excitation</i> .....	55
Gambar 4.16 Diagram Alir <i>Gradient Orientation</i> .....	57
Gambar 4.17 Diagram Alir Pembentukan Histogram WLD.....	59
Gambar 4.18 Diagram alir pmdanan dan <i>n-rank</i> hasil.....	61
Gambar 4.19 Potongan Citra atau <i>Window</i> .....	61
Gambar 4.20 <i>Window</i> dipotong dengan ukuran 3 x 3 <i>pixel</i> atau radius 1 <i>pixel</i> ....	75
Gambar 4.21 Hasil perpotongan 3x3 <i>pixel</i> atau radius 1 <i>pixel</i> .....	75
Gambar 6.1 Grafik CMC SIFT pengujian parameter dan <i>sliding window</i> .....	104
Gambar 6.2 Grafik CMC MCWLD <i>Sliding window Overlapping</i> 16 .....	107
Gambar 6.3 Grafik CMC MCWLD <i>Sliding window Non-overlapping</i> 16.....	108

Gambar 6.4 Grafik CMC MCWLD <i>Sliding window Overlapping</i> 32 .....	109
Gambar 6.5 Grafik CMC MCWLD <i>Sliding window Non-overlapping</i> 32 .....	110
Gambar 6.6 Grafik CMC SIFT+MCWLD <i>Sliding window 16 Overlapping</i> SIFT 4 ...	113
Gambar 6.7 Grafik CMC SIFT+MCWLD <i>Sliding window 16 Overlapping</i> SIFT 8 ...	114
Gambar 6.8 Grafik CMC SIFT+MCWLD <i>Sliding window 16 Non-overlapping</i> SIFT 4 .....	115
Gambar 6.9 Grafik CMC SIFT+MCWLD <i>Sliding window 16 Non-overlapping</i> SIFT 8 .....	116
Gambar 6.10 Grafik CMC SIFT+MCWLD <i>Sliding window 32 Overlapping</i> SIFT 4 .....	117
Gambar 6.11 Grafik CMC SIFT+MCWLD <i>Sliding window 32 Overlapping</i> SIFT 8 .....	118
Gambar 6.12 Grafik CMC SIFT+MCWLD <i>Sliding window 32 Non-overlapping</i> SIFT 4 .....	119
Gambar 6.13 Grafik CMC SIFT+MCWLD <i>Sliding window 32 Non-overlapping</i> SIFT 8 .....	120
Gambar 6.14 Grafik CMC SIFT+MCWLD dengan konfigurasi terbaik pada <i>top-match</i> .....	121
Gambar 6.15 Citra sketsa 16x16 <i>pixel</i> .....	122
Gambar 6.16 Citra sketsa 32 x 32 <i>pixel</i> .....	123
Gambar 6.17 Grafik Distance Frequency Distribution metode SIFT dengan <i>sliding window 16 overlapping</i> dan ukuran <i>window 8</i> .....	124
Gambar 6.18 Grafik Distance Frequency Distribution metode SIFT dengan <i>sliding window 32 overlapping</i> dan ukuran <i>window 8</i> .....	124
Gambar 6.19 Visualisasi <i>sliding window</i> .....	125
Gambar 6.20 Potongan <i>Sliding window</i> .....	125
Gambar 6.21 Perbandingan histogram foto dan sketsa pada potongan citra ....	127
Gambar 6.22 Visualisasi <i>sliding window</i> .....	128
Gambar 6.23 Potongan <i>Sliding window</i> .....	128
Gambar 6.24 Perbandingan histogram foto dan sketsa pada potongan citra pada individu yang berbeda .....	128
Gambar 6.25 Grafik Distance Frequency Distribution metode MCWLD <i>sliding window 16 overlapping</i> dan parameter $T=8, M=4, S=3$ .....	130
Gambar 6.26 Grafik Distance Frequency Distribution metode MCWLD <i>sliding window 16 overlapping</i> dan parameter $T=6, M=4, S=3$ .....	130
Gambar 6.27 Histogram persebaran nilai <i>differential excitation</i> ( $M=6$ ) .....	132
Gambar 6.28 Histogram persebaran nilai <i>differential excitation</i> ( $M=4$ ) .....	132
Gambar 6.29 Grafik <i>distance frequency distribution</i> pada individu $m$ (78) .....	133

## DAFTAR KODE SUMBER

Kode Sumber 5.1 Proses <i>grayscale</i> .....	89
Kode Sumber 5.2 Fungsi <i>Sliding window</i> SIFT .....	89
Kode Sumber 5.3 Inisialisasi SIFT .....	90
Kode Sumber 5.4 Fungsi pemotongan <i>Smaller Window</i> .....	91
Kode Sumber 5.5 Fungsi Ekstraksi Fitur SIFT .....	92
Kode Sumber 5.6 Fungsi <i>Zero Padding</i> .....	92
Kode Sumber 5.7 Fungsi Ekstraksi Orientasi.....	93
Kode Sumber 5.8 Fungsi Besaran Gradien dan pembobotan.....	94
Kode Sumber 5.9 Fungsi <i>Histogram binning</i> .....	95
Kode Sumber 5.10 Inisialisasi MCWLD.....	96
Kode Sumber 5.11 Fungsi Ekstraksi Fitur MCWLD.....	97
Kode Sumber 5.12 Fungsi <i>Gradient Orientation</i> .....	98
Kode Sumber 5.13 Fungsi <i>Differential excitation</i> .....	98
Kode Sumber 5.14 Pembentukan histogram 1D.....	99



## DAFTAR LAMPIRAN

LAMPIRAN A	Dataset.....	142
A.1	Galeri Foto Wajah .....	142
A.2	Galeri Sketsa Wajah .....	149
LAMPIRAN B	Hasil Pemasahan .....	156
B.1	Akurasi Dari Score Level Fusion .....	156
B.2	Akurasi dari <i>Neyman-Pearson Decision Rule</i> .....	172



## BAB 1 PENDAHULUAN

Pada bab ini, penulis memuat latar belakang dari topik yang diangkat dan terkait penelitian meliputi rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian dan sistematika pembahasan.

### 1.1 Latar Belakang

Mengidentifikasi seseorang berdasarkan cirinya merupakan salah satu ranah pada ruang lingkup biometrik (Jain, 2008). Pada ruang lingkup biometrik, setiap individu mempunyai karakteristik unik yang menjadi ciri khasnya seperti sidik jari, mata, bentuk telinga maupun wajah. Fungsi dari biometrik ini dapat digunakan untuk melakukan autentikasi terhadap sistem maupun mengenali seseorang berdasarkan salah satu cirinya. Biometrik ini kemudian dibagi menjadi kategori fisik dan kategori perilaku (Jain, et al., 2011). Kategori fisik adalah ciri yang tampak seperti sidik jari, wajah atau bahkan iris mata. Sedangkan kategori perilaku mencakup gaya berjalan atau suara seseorang. Ada berbagai bidang yang mengimplementasikan biometrik seperti bidang keamanan, kependudukan hingga forensik.

Pada kategori fisik, tiga ciri yang sering digunakan adalah sidik jari, wajah dan iris mata. Meski sidik jari dan iris mata merupakan ciri yang lebih *robust* dibanding wajah, namun pengenalan wajah masih menjadi pusat perhatian dalam penelitian biometrik. Tidak seperti sidik jari dan iris mata, wajah dapat diperoleh dengan mudah sehingga dengan menggunakan CCTV yang beredar dilingkungan tempat tinggal atau publik, gambar wajah tersebut dapat digunakan untuk mengidentifikasi pelaku kejahatan. Meski tidak ada CCTV pada lokasi kejahatan, ciri wajah masih mudah didapatkan dari saksi mata yakni dalam bentuk sketsa. Selain itu memecahkan pengenalan wajah yang tidak terbatas membutuhkan sejumlah besar penelitian dalam pemodelan wajah, ekstraksi fitur dan pencocokan (Klare & Jain, 2010).

Pada bidang forensik, ciri biometrik yang dapat digunakan salah satunya adalah ciri wajah dalam bentuk sketsa (McQuiston-Surrett, et al., 2006). Pada tahun 1880, Alphonse Bertillon yang biasa dijuluki sebagai "*Father of scientific detection*" mengembangkan sebuah sistem yang menyimpan beberapa fitur wajah yang diambil dari foto untuk digunakan sebagai alat identifikasi wajah pelaku penjahat. Sistem ini disebut dengan *Portrait Parle* atau *speaking likeness*. Saksi mata menyebutkan karakteristik dari pelaku yang kemudian dapat dicari dari indeks. Hingga saat ini, sketsa forensik masih digunakan dengan menggunakan karakteristik dari saksi mata (Klare, et al., 2011) karena tidak semua tempat memiliki CCTV. Sistem yang diciptakan oleh Bertillon pada saat itu merupakan cikal bakal dari sketsa forensik modern yang digunakan sekarang. Seiring bertambahnya jumlah penduduk, keragaman (variasi) wajah semakin bertambah juga. Kesulitan yang terus dialami adalah memadankan karakteristik wajah seseorang yang digambarkan pada sketsa dengan foto penduduk membutuhkan

waktu yang lama dan perbedaan informasi karakteristik pelaku kejahatan yang dijelaskan oleh saksi mata yang berbeda juga mempersulit pencarian.

Seiring berjalan waktu, munculah gagasan ide *sketch recognition*, proses pengenalan sketsa secara otomatis yang dilakukan oleh komputer. *Sketch recognition* merupakan metode untuk memadankan sketsa forensik dengan foto penduduk. Sistem *sketch recognition* akan memberikan kandidat pelaku kejahatan yang memiliki kemiripan dengan sketsa forensik yang diberikan. Perbedaan antara *face recognition* dengan *sketch recognition* adalah pada proses pemadannya (*matching mechanism*) antara *query* atau *probe* dengan dataset atau *gallery* yang dimiliki. Sketsa dan foto berada pada modalitas yang berbeda. Perbedaan antara foto (citra digital) dengan sketsa adalah pada foto terdapat informasi makro (yang terlihat jelas seperti bentuk mata, hidung, dagu maupun bentuk wajah) dan mikro (tekstur atau pigmen), sedangkan pada sketsa hanya terdapat informasi makro (secara garis besar), yakni hanya fitur unik saja yang ditekankan dan memiliki sedikit informasi tekstur (Nagpal, et al., 2016). Pada *face recognition*, berbagai pendekatan dapat dilakukan seperti menggunakan pendekatan berdasarkan tampak (*appearance-based*) menggunakan *eigenface*, *fisherface* atau *sirface* (Ling, et al., 2003) atau berdasarkan tekstur hingga kedekatan bentuk wajah berdasarkan rasio menggunakan *fiducial points* (Arca, et al., 2006). Berbeda halnya dengan *sketch recognition* yang perlu mengekstraksi fitur sedemikian rupa sehingga fitur yang diperoleh dapat dipadankan dengan *gallery* yang berupa foto untuk dapat menentukan kandidat pemilik identitas tersebut (*sketch-to-photo identification*). Proses pemadanan pada ruang lingkup *sketch recognition* disebut juga dengan *heterogeneous matching*.

Dari penelitian tentang *heterogeneous matching* khususnya foto dan sketsa, metode-metode yang digunakan dikelompokkan menjadi dua pendekatan yakni dengan menggunakan pendekatan *generative* dan pendekatan *discriminative* (Bhatt, et al., 2012). Pendekatan *generative* merupakan pendekatan dimana foto wajah diubah menjadi sebuah sketsa dengan metode tertentu atau sebaliknya sehingga dapat saling dipadankan. Sedangkan pendekatan *discriminative* melakukan ekstraksi fitur dari sketsa dan foto untuk dipadankan. Pendekatan diskriminatif lebih baik dalam hal komputasi dibandingkan dengan pendekatan generatif. Penelitian berfokus pada pendekatan *discriminative*.

Salah satu pendekatan *discriminative* yang dapat digunakan adalah analisis tekstur. Tekstur adalah sebuah properti yang merepresentasikan permukaan dan struktur dari citra (Srinivasan & G, 2008). Istilah analisis tekstur merujuk pada model matematika yang mencirikan variasi spasial sebagai cara mengekstraksi informasi (Srinivasan & G, 2008). Analisis tekstur yang dapat dilakukan pada citra adalah mengambil nilai orientasi pada setiap *pixel*-nya, perubahan warna (*gradient*) antar *pixel*, jumlah nilai keabuan pada suatu *window* atau *region* (*Gray-level Co-occurrence Matrices*) atau berdasarkan intensitas ketetanggannya (*Random Field Models*) (Tuceryan & Jain, 1993). Pada ruang lingkup *heterogeneous matching* ini, informasi tekstur untuk sketsa berasal dari goresan karbon yang membentuk struktur dari wajah sedangkan pada foto wajah berasal dari

permukaan yang tampak serta perubahan warna. Dengan mengambil nilai gradien serta orientasi pada goresan tersebut kemudian mengelompokkannya dengan membentuk sebuah histogram sehingga menjadi sebuah fitur, maka nilai tersebut dapat digunakan untuk melakukan proses pemadanan antara foto dan sketsa. Hal yang sama dilakukan pada foto. Tujuan dari mengekstraksi fitur ini adalah untuk menyamakan modalitas antara sketsa dan foto.

Ada berbagai cara untuk mengekstraksi fitur pada citra dengan menggunakan tekstur analisis namun dalam penelitian ini, metode analisis tekstur yang digunakan adalah menggunakan SIFT (*Scale Invariant Feature Transform*) yang telah terbukti mampu digunakan pada ruang lingkup *heterogeneous* (Klare & Jain, 2010). SIFT mengekstraksi nilai gradien dari suatu *pixel* kemudian mengelompokkannya berdasarkan orientasi pada *pixel* tersebut ke delapan rentang arah orientasi. Proses ekstraksi pada SIFT dilakukan pada ukuran *window* tertentu. Metode lainnya adalah MCWLD (*Multiscale Circular Weber Local Descriptor*) (Bhatt, et al., 2012) yang berbasis pada hukum Weber, yakni melihat dari ratio suatu perubahan intensitas *pixel* dengan *pixel* ketetanggannya (Chen, et al., 2010). MCWLD menggunakan nilai *differential excitation* yang melihat ratio *pixel* tersebut dengan tetanggannya dan mengelompokkannya pada histogram berdasarkan rentang *differential excitation* itu sendiri pada orientasi tertentu. Adapun fusi diantara kedua metode tersebut juga dapat menentukan metode terbaik yang dapat bekerja dengan baik pada ruang lingkup *heterogeneous matching* khususnya pada *sketch recognition* dalam bidang biometrik ini.

## 1.2 Rumusan Masalah

Meninjau dari latar belakang diatas, dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana tahapan mengimplementasikan identifikasi individu berdasarkan sketsa wajah menggunakan pendekatan diskriminatif?
2. Bagaimana cara terbaik untuk melakukan fusi terhadap dua metode?
3. Bagaimana perbandingan tingkat akurasi dan performa pada pendekatan diskriminatif menggunakan metode SIFT, multiscale circular Weber local descriptor (MCLWD) dan SIFT+MCLWD ?

## 1.3 Tujuan Penelitian

Tujuan umum dari penelitian ini adalah:

Pendekatan diskriminatif mampu mengidentifikasi individu berdasarkan sketsa wajah.

Tujuan khusus dari penelitian ini adalah:

1. Mengimplementasikan metode SIFT, *multiscale circular Weber local descriptor* (MCWLD), serta SIFT+MCWLD

2. Menguji tingkat akurasi terhadap metode SIFT, *multiscale circular Weber local descriptor* (MCWLD), serta SIFT+MCWLD (Fusi)

#### 1.4 Manfaat Penelitian

Melalui penelitian ini, penulis dapat mengimplementasikan ilmu mengenai visi komputer, pengolahan citra digital dan pengenalan pola dengan topik identifikasi individu berdasarkan sketsa wajah menggunakan pendekatan diskriminatif.

Bagi pembaca dan masyarakat:

Melalui penelitian ini dapat memberikan kontribusi yang baik sehingga ilmunya berguna dan dapat dikembangkan seiring perubahan zaman.

Bagi fakultas:

Melalui penelitian ini diharapkan fakultas dapat menambah referensi pembelajaran bagi mahasiswanya mengenai visi komputer

Bagi Pusat Unggulan Iptek Biometrik:

Melalui penelitian ini diharapkan hasil yang diberikan dapat menjadi membantu salah satu studi riset untuk kebutuhan yang akan datang.

#### 1.5 Batasan Masalah

Pada penelitian ini, permasalahan yang diangkat dibatasi oleh hal-hal sebagai berikut:

1. Metode yang digunakan untuk mengambil fitur pada foto dan sketsa adalah SIFT, *Multiscale Circular Weber Local Descriptor* (MCLWD) serta SIFT+MCLWD
2. Metode SIFT yang digunakan pada penelitian ini tidak melakukan proses *keypoint detection* untuk mendapatkan lokasi lokal tetapi menggunakan metode *sliding window* yang *overlapping* sehingga proses pada metode SIFT yang digunakan adalah proses *feature descriptors*. Pada metode SIFT ini, proses pembobotan menggunakan nilai dari *gradient magnitude* dan *gaussian weighing*
3. Hasil yang diberikan dari penelitian ini adalah analisis akurasi terhadap metode yang diujikan.
4. Sketsa yang digunakan adalah *viewed sketches* dan pada penelitian ini tidak mencari pengaruh akurasi terhadap perbedaan jenis sketsa
5. Data yang digunakan adalah data sketsa wajah *open source* dari *Chinese University of Hong Kong* karena sudah menjadi acuan pada pengembangan metode *heterogeneous face recognition*.

## 1.6 Sistematika Pembahasan

### BAB I PENDAHULUAN

Memuat latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika pembahasan/laporan yang terkait dengan penelitian yang dilakukan.

### BAB II TINJAUAN KEPUSTAKAAN

Memuat kajian pustaka dan dasar teori yang berkaitan dengan penelitian dan menunjang penelitian. Pada landasan kepusatakaan ini ruang lingkup dasar teori mencakup pada pengenalan wajah, sistem biometrik, pengenalan seksta, *heterogeneous matching*, serta metode-metode yang diusulkan yakni SIFT, MCWLD dan fusi.

### BAB III METODOLOGI

Memuat uraian metode penelitian yang digunakan dalam menyelesaikan permasalahan dalam penelitian yang terdiri dari alur metode penelitian, studi literatur, pengumpulan data, perancangan algoritme, implementasi algoritme, pengujian algoritme, serta analisis hasil pengujian dan penarikan kesimpulan.

### BAB IV PERANCANGAN

Memuat penjelasan dan detail perancangan yang digunakan untuk mengimplementasikan metode dalam penelitian ini. Perancangan ini meliputi *flowchart* kerja sistem, perancangan akuisisi data, perancangan pemadanan, perancangan antarmuka serta perancangan pengujian algoritme serta memuat implementasi dari perancangan baik dari proses akuisisi data hingga proses pengujian serta memulai potongan Kode Sumber dalam mengimplementasikan metode.

### BAB V IMPLEMENTASI

Memuat implementasi yang didasari dari perancangan yang telah dirangkum pada bab IV Perancangan. Pada bab ini terdapat poin-poin yang menjadi batasan implementasi dan beberapa potongan Kode Sumber yang digunakan untuk mengimplementasikan metode yang diusulkan pada penelitian ini.

### BAB VI PENGUJIAN DAN ANALISIS

Memuat pengujian yang dilakukan terhadap metode yang ditawarkan melalui skenario yang telah dirancang pada bab IV Perancangan. Pada bab ini berisi hasil pengujian algoritme serta melakukan analisis terhadapnya.

### BAB VII KESIMPULAN DAN SARAN

Memuat kesimpulan dari hasil penelitian yang dilakukan serta saran yang diberikan untuk mengembangkan penelitian yang telah dilakukan.

## BAB 2 TINJAUAN KEPUSTAKAAN

Bab ini terdiri dari kajian pustaka dan dasar teori. Kajian pustaka membahas penelitian yang telah dilakukan sebelumnya yang memiliki topik yang sama namun menggunakan metode yang berbeda dengan penelitian yang akan dilakukan ini. Dasar teori membahas mengenai dasar-dasar teori yang akan digunakan untuk menunjang penulisan skripsi yang berjudul “*Identifikasi Individu Berdasarkan Sketsa Wajah Menggunakan Pendekatan Diskriminatif*”.

### 2.1 Kajian pustaka

Sebagai kajian pustaka yang digunakan pada penelitian ini, ruang lingkup penelitian sebelumnya dikhususkan pada ranah biometrik dengan objeknya adalah sketsa wajah. Penelitian-penelitian sebelumnya ini kemudian dikelompokkan berdasarkan pendekatannya terhadap merubah modalitas dari sketsa. Adapun pendekatan tersebut dibagi menjadi dua, yakni:

**Tabel 2.1 Pendekatan pada Heterogeneous Matching**

No	Pendekatan	Keterangan
1	<i>Generative</i>	Memodelkan citra digital (foto) menjadi sebuah sketsa sehingga memiliki modal yang sama dengan <i>query</i> untuk dilakukan proses pemadanan. Begitu juga sebaliknya, yakni membuat sketsa menjadi sebuah foto. Citra sketsa yang dihasilkan ( <i>sketch synthesis</i> ) memiliki pola atau tekstur yang bergantung dengan data yang dimiliki. Pendekatan ini dilakukan pada kasus dimana <i>query</i> yang diberikan tidak terdapat pada data (karena faktor <i>artist</i> , keterangan saksi mata yang berbeda sehingga informasi tidak lengkap) atau data berupa galeri foto tanpa ada sketsa (Bhatt, et al., 2012).
2	<i>Discriminative</i>	Melakukan ekstraksi fitur pada sketsa maupun foto pada galeri untuk di cocokan. Fitur yang digunakan beragam dengan pendekatannya masing-masing. Fitur yang perlu di perhatikan adalah baik pada foto galeri dan sketsa, fitur yang di ekstraksi merupakan modal yang sama sehingga dapat dilakukan proses pemadanan. Pada pendekatan ini, tidak ada proses merubah dari foto ke sketsa maupun sebaliknya (Bhatt, et al., 2012).

Baik pendekatan *generative* maupun *diskriminatif*, masing-masing memiliki kekurangan dan kelebihan masing-masing. Penelitian yang dilakukan oleh (Yung-hui, et al., 2006) menggunakan pendekatan *generative* dengan menggunakan metode *eigenface* dimana sketsa wajah diubah menjadi suatu dimensi tertentu.

**Tabel 2.2 Kajian Pustaka**

No	Judul	Metode	Pendekatan	Keterangan	Hasil
1	<i>Face Photo-Sketch Synthesis and Recognition</i>	<i>Multiscale Markov Random Fields</i> (Wang & Tang, 2009)	<i>Generative</i>	Melakukan sintesis sketsa berdasarkan galeri foto wajah dengan penambahan ekstraksi fitur pada multiskala	Penelitian yang dilakukan menggunakan dataset CUHK Public Data set mendapat akurasi (pada <i>rank</i> sama dengan 1) hingga 96.3%
2	<i>Face Sketch Recognition</i>	<i>Geometrical Measures + Direct Eigenface Method</i> (Tang & Wang, 2004)	<i>Generative</i>	Menggunakan <i>geometrical features</i> karena setiap orang mempunyai bentuk wajah yang berbeda	Hasil akurasi yang diberikan 73%
3	<i>Sketch-to-photo matching: A Feature-based approach</i>	SIFT (Klare & Jain, 2010)	<i>Discriminative</i>	Pada training, mengekstraksi nilai berdasarkan besaran gradien dan orientasi berdasarkan region lokal menggunakan metode SIFT. Penelitian ini menguji <i>common representation</i> dengan asumsi modalitas antara sketsa dan foto tersebut sama.	Hasil yang diberikan pada nilai <i>rank</i> sama dengan 1 mendapatkan nilai akurasi 97.87% untuk Fusi, sedangkan dengan metode SIFT mempunyai akurasi 97% dan menggunakan <i>common representation</i> dengan



				Penelitian juga mencari hasil akurasi fusi antara metode SIFT dengan SIFT yang telah diberi <i>common representation</i> serta konfigurasi <i>multiscale</i> pada metode SIFT	akurasi 96.47%
4	<i>Matching Forensic Sketches to Mugshot Photos</i>	LFDA (Klare, et al., 2011), (Klare & Jain, 2011)	<i>Discriminative</i>	Hasil dari SIFT+MLBP di ekstraksi dengan mengambil nilai yang paling menonjol dengan menggunakan <i>framework</i> LFDA dimana metode ini mengalisis wilayah lokal pada citra.	Penelitian yang dilakukan menggunakan dataset CUHK Public Data set mendapat akurasi (pada <i>rank</i> sama dengan 1) hingga 97.33% tanpa LFDA dan 99.47% dengan LFDA
5	<i>Memetically Optimized MCWLD for Matching Sketches With Digital Face Images</i>	<i>Multiscale Weber Local Descriptor</i> (Bhatt, et al., 2012)	<i>Discriminative</i>	<i>MCWLD</i> atau <i>Multiscale Circular Weber Local Descriptor</i> merupakan salah satu metode deskriptor yang di cetuskan oleh Bhatt (Bhatt, et al., 2012) yang berbasis pada hukum Weber yang dapat mengekstraksi fitur berdasarkan	Penelitian ini menggunakan dataset CUHK Public Data set. Nilai akurasi yang diberikan pada <i>rank</i> sama dengan 1 mencapai 97.28%



				ketetanggaannya a. WLD ini mampu mengekstraksi informasi mikro.	
--	--	--	--	---	--

Pada Tabel 2.2 Kajian Pustaka telah disebutkan beberapa jurnal dari hasil studi literatur yang berkaitan dengan penelitian ini. Dari jurnal yang diperoleh terdapat jurnal yang menggunakan pendekatan diskriminatif pada kasus yang serupa dengan penelitian ini. Adapun kesimpulan dari tabel tersebut adalah menggunakan metode SIFT dan MCWLD sebagai metode yang digunakan pada penelitian ini

## 2.2 Landasan pustaka

### 2.2.1 Citra Digital

Citra bisa di definisikan sebagai fungsi 2 dimensi ( $f(x, y)$ ) dimana koordinat  $x$  dan  $y$  merupakan koordinat spasial dan fungsi tersebut memiliki amplitudo dari setiap sumbu  $x$  dan  $y$  yang menjadi tingkat abu-abu atau intensitas dari citra (Gonzalez, 2010). Sebuah citra digital terdiri dari sejumlah satuan element yang memiliki lokasinya tersendiri dan mengandung sebuah nilai. Element tersebut adalah *pixel*.

### 2.2.2 Citra Abu-abu

Citra abu-abu merupakan citra yang hanya memiliki satu *channel* warna yang memiliki rentang nilai 0-255. Nilai RGB tersebut dikonversi menjadi nilai abu-abu menggunakan rumus sebagai berikut:

$$Gray\ Value = 0.299 * red + 0.587 * green + 0.114 * blue \quad (2.1)$$

Citra abu-abu sering digunakan untuk pengolahan citra digital karena nilai abu-abu ini dapat merepresentasikan nilai tekstur pada suatu objek. Baik pada sketsa maupun foto, tekstur yang didapat pada satu *channel* lebih diutamakan.

### 2.2.3 Sistem Biometrik

Pada sistem biometrik yang diterangkan oleh Jain A K pada buku *Handbook of Biometric* (Jain, et al., 2007) terdapat beberapa bagian yakni:

1. Modul sensor: merupakan bagian yang mengambil data biometrik langsung dari individu yang bersangkutan. Pengambilan data biometrik bergantung pada *traits* yang ingin diambil. Sebagai contoh, sidik jari dapat diperoleh datanya dengan menggunakan *fingerprnt scanner*.
2. Modul penilaian kualitas dan ekstraksi fitur: bagian ini akan mengecek kualitas dari data biometrik yang di peroleh. Terkadang data yang di ambil, kualitas pada data tersebut sangat buruk sehingga tidak dapat diproses, oleh karena itu data yang di peroleh perlu dicek dahulu



kualitasnya. Pada bagian ini, proses ekstraksi fitur juga terjadi, dimana pada sistem biometrik hanya menggunakan kumpulan fitur diskriminatif yang menonjol. Kumpulan fitur ini merepresentasikan pemilik individu dari data biometrik tersebut.

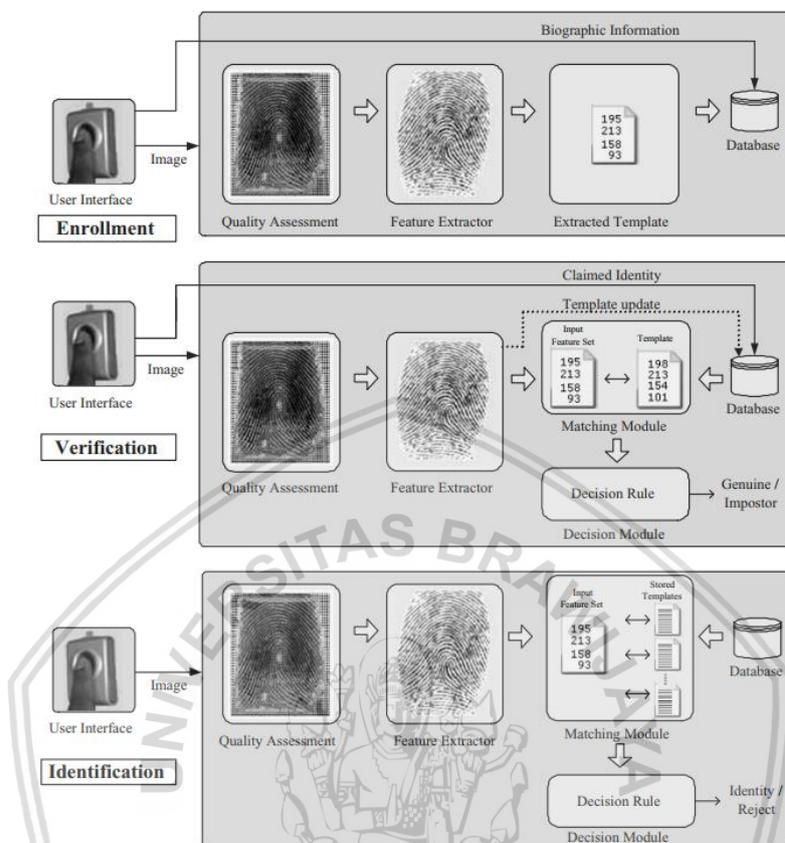
3. Modul pepadanan dan pengambilan keputusan: proses pepadanan dilakukan terhadap fitur yang diperoleh dari citra *probe* dengan apa yang sudah tersimpan basisdata. Pada bagian pepadanan ini akan menghasilkan nilai *match score*. Nilai tersebut akan digunakan untuk memvalidasi identitas untuk identifikasi atau verifikasi.
4. Modul basisdata sistem: bagian ini digunakan untuk menyimpan informasi biometrik. Fitur yang diekstraksi dari data biometrik disimpan berkorespondensi dengan informasi biografik seperti nama, alamat dan lain-lain yang merupakan informasi penting.

Dalam pengoperasiannya, sistem biometrik dapat dioperasikan sebagai sistem untuk verifikasi atau identifikasi. Pada verifikasi, sistem akan memadankan bahwa yang mengklaim dirinya adalah pemilik identitas ini adalah orang yang sama dengan yang terdaftar dalam basis data. Verifikasi melakukan perbandingan *one-to-one*. Sedangkan identifikasi, sistem akan mengenali individu dengan mencari dari seluruh data yang terdaftar. Hal ini disebut dengan perbandingan *one-to-many*. Verifikasi disebut juga pengenalan positif (*positive recognition*) dimana bertujuan untuk mencegah adanya sejumlah individu memiliki identitas yang sama. Contohnya adalah PIN, kata sandi atau token. Identifikasi disebut pengenalan negatif (*negative recognition*) karena bertujuan untuk mencegah adanya individu yang memiliki banyak identitas. Pengenalan negatif hanya bisa dilakukan menggunakan atribut biometrik.

Pada Gambar 2.1 dijelaskan alur *enrollment* atau registrasi identitas terhadap sistem biometrik. Hal yang akan dilakukan pada proses ini adalah mengambil sampel biometrik dari pendaftar serta informasi biografisnya, seperti nama, jenis kelamin, umur, tempat tinggal, tanggal lahir atau informasi penting lainnya tentang pendaftar. Sampel biometrik yang diterima akan diperiksa dahulu. Jika kualitas yang diberikan tidak sesuai dengan standar pada sistem biometrik tersebut, maka pendaftar harus mengulang pengambilan sampel biometrik. Jika sampel yang diberikan lolos dari pemeriksaan kualitas, maka akan diekstraksi fitur dari sampel tersebut yang kemudian menjadi sebuah *template* atau cetakan yang berisi informasi biometrik dari pendaftar. *Template* ini akan disimpan bersamaan dengan informasi biografis pendaftar.

Proses verifikasi seperti pada Gambar 2.1 tersebut, sistem biometrik akan mengambil sampel dari pengguna lalu memeriksa kualitas dari sampel tersebut. Sama seperti proses *enrollment*, jika tidak lolos pada tahap pemeriksaan kualitas, maka pengguna harus mengulang proses pengambilan sampel biometrik. Pada proses verifikasi ini, pengguna adalah orang yang mengklaim dirinya adalah individu yang diklaimnya. Sehingga proses pepadanan hanya diberlangsungkan hanya pada individu yang diklaimnya tersebut. Hasil dari proses pepadanan akan diproses pada modul pengambilan keputusan. Hasil dari verifikasi adalah

pengguna dinyatakan benar sebagai pemilik identitas yang diklaimnya (*genuine*) atau sebagai penyemu (*impostor*).



**Gambar 2.1 Tahapan enrollment dan pengenalan pada sistem biometrik. sumber (Jain, et al., 2007)**

Lain halnya dengan identifikasi, sistem biometrik perlu mengenali pengguna yang mengklaim dirinya adalah orang yang dikenal atau terdaftar pada sistem biometrik atau tidak. Pada identifikasi ini, pengguna tidak memberikan informasi identitas yang diklaimnya seperti verifikasi, tapi hanya memberikan sampel biometrik yang akan dikenali dan diperiksa oleh sistem. Hasil dari identifikasi adalah dikenali atau teridentifikasi atau tertolak.

### 2.2.4 Pengukuran Performa Pengenalan Wajah Biometrik

Mengacu pada buku *handbook of face recognition* (Jain & Li, 2011), pengukuran performa biometrik ada tiga, yakni:

1. *Closed-set identification*: identifikasi terhadap *probe* yang benar-benar ada di galeri. Pertanyaan yang mendasari identifikasi adalah, *wajah siapakah ini?*
2. *Open-set identification*: sistem melakukan identifikasi dan menentukan sampel tersebut terdapat pada galeri. *Probe* pada *open-set*

*identification* terdapat dua, yakni yang benar ada pada galeri dan yang benar tidak ada pada galeri. Pertanyaan yang mendasarinya adalah, *apakah kita mengenali wajah ini?*

3. *Verification* atau *authentication*: sistem melakukan perbandingan 1:1. Pertanyaan yang mendasarinya adalah, *apakah orang tersebut adalah orang yang diklaimnya?*

Dalam mengukur performa pengenalan wajah dan biometrik, hal yang perlu disiapkan adalah galeri yang berisi sampel biometrik yang dikenali oleh sistem dan dua buah kumpulan *probe* (sample biometrik yang akan dikenali oleh sistem). *Probe* yang pertama terdapat sampel biometrik dimana sampel tersebut ada pada galeri. Sedangkan *probe* yang kedua berisi sampel biometrik yang tidak terdapat pada galeri.

### 2.2.5 Close-set Identification

*Close-set identification* seperti yang dijelaskan pada subbab 2.2.4, merupakan salah satu metode pengukuran performa pada ruang lingkup pengenalan wajah dan biometrik. Tugas utama pada *close-set identification* adalah sistem atau metode yang digunakan dapat menentukan bahwa wajah atau *probe* yang diberikan ada pada *rank* pertama atau pada *n-rank* (Jain & Li, 2011). Pengukuran performa ini dilakukan dengan mengurutkan nilai *similarity* (*dissimilarity* untuk *distance*) antara *probe* ke-*j* dengan galeri. *Identification rate* didapat berdasarkan pembagian dari jumlah orang dengan label yang sama dengan *probe* ke-*j* dibagi dengan jumlah data pada galeri.

$$P_I(n) = \frac{|C(n)|}{|Gallery|} \quad (2.2)$$

Dimana

$$C(n) = |\{p_j: rank(p_j) \leq n\}| \quad (2.3)$$

### 2.2.6 Within-between class variance ratio

*Within-between class variance ratio* merupakan sebuah pengukuran yang dilakukan untuk melihat suatu metode dapat memisahkan antara *genuine* dan *impostor* dengan baik. Nilai rasio tersebut didapatkan dari membandingkan nilai variansi dari kelas yang sama atau *within-class variance* dengan kelas yang berbeda atau *between-class variance*. Semakin kecil nilai rasio yang diberikan, maka metode tersebut dapat dikatakan baik untuk memisahkan antara kelas *genuine* dengan kelas *impostor*. Pada implementasinya, nilai rasio tersebut tidak mungkin akan bernilai 0. Adapun rumusnya ditunjukkan pada persamaan (2.4) serta pada Gambar 2.2.

$$\sigma_{within}^2 = \left( \frac{1}{X_{impostor}} \times \sum_{x \in X_{impostor}} (x - \mu_{impostor})^2 \right) + \left( \frac{1}{X_{genuine}} \times \sum_{x \in X_{genuine}} (x - \mu_{genuine})^2 \right) \tag{2.4}$$

$$\sigma_{between}^2 = \frac{N_{genuine} \times N_{impostor}}{N_{genuine} + N_{impostor}} (\mu_{impostor} - \mu_{genuine})^2$$

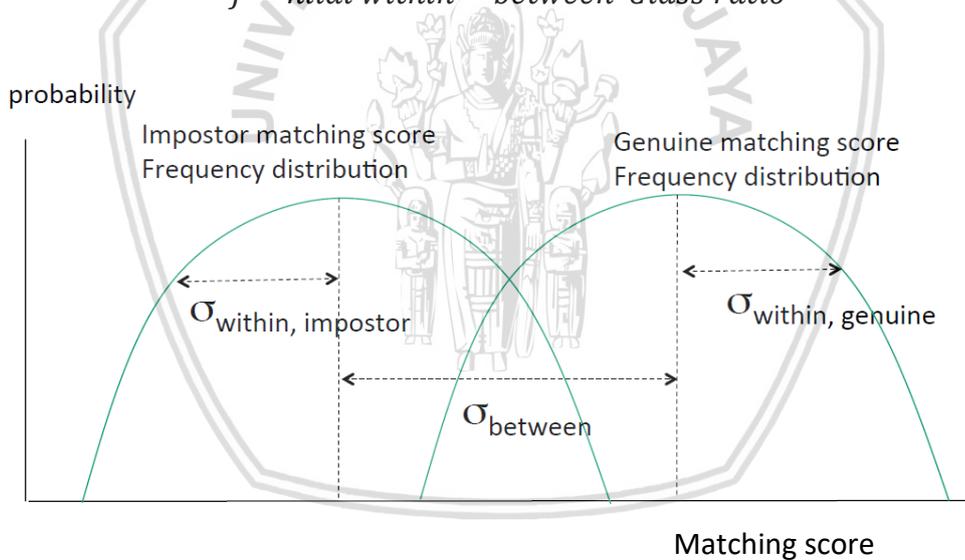
$$J = \frac{\sigma_{between}^2}{\sigma_{within}^2}$$

Dimana

$\mu$  = nilai rata – rata dari suatu kelas

$N$  = jumlah hasil pemadanan

$J$  = nilai within – between Class ratio



Gambar 2.2 Grafik similarity frequency distribution

### 2.2.7 Face Identification

Face identification digunakan untuk mengenali individu oleh komputer berdasarkan query citra wajah. Face identification ini sering kali tertukar dengan face verification. Pada dasarnya, face verification merupakan proses validasi query citra dengan identitas yang telah diklaim. Proses ini membandingkan identitas sebenarnya dengan citra wajah pemilik identitas. Face verification menggunakan kesamaan secara one-to-one, sedangkan face identification menggunakan

kesamaan atas *one-to-many*. Jika jumlah citra wajah terdaftar dalam database bertambah banyak, akurasi dan efisiensi dari *face identification* berkurang. Sedangkan performa pada *face verification* tidak terlalu terpengaruh dengan besarnya data pada basis data (Ikeuchi, 2016). *Face identification* mempunyai tantangan besar seperti variasi intrapersonal dalam jumlah banyak maupun sedikit dan permasalahan jumlah sampel yang sedikit untuk dipecahkan.

Dalam mengekstraksi fitur, ada 4 fitur yang dapat digunakan untuk *face identification*, antara lain adalah fitur geometri, fitur *holistic*, fitur lokal dan fitur semantik. Fitur geometrik bergantung pada ukuran komponen wajah dan jarak antar komponen tersebut. Sebagai contoh mata dan hidung memiliki jarak yang berbeda-beda pada setiap orang. Fitur *holistic* menyederhanakan vektor dimensi-tinggi dengan menjumlahkan semua *pixel* pada region di wajah. Fitur local mengekstraksi berdasarkan tekstur pada suatu regions. Fitur holistik dan fitur local sangat sering digunakan. Fitur semantik memiliki dua tipe, antara lain adalah atribut dan *similarity*. Masing-masing tipe tersebut mempunyai caranya tersendiri untuk di ekstraksi dari citra wajah. Pada atribut, sistem mempunyai hitungan khusus untuk dapat mengenali apakah wajah tersebut mempunyai atribut tertentu atau tidak. Sedang pada *similarity*, yang dicari adalah kesamaannya.

### 2.2.8 Tingkatan Fitur Wajah

Pada wajah, terdapat beberapa tingkatan fitur yang dapat diperoleh yang bergantung pada kualitas citra wajah yang diberikan. Sesuai dengan penjelasan Anil K Jain pada buku *Introduction to Biometrics* (Jain, et al., 2011) tingkatan tersebut terbagi menjadi:

1. *Level 1 details*: pada tingkat ini, jarak antar pupil (*interocular pixel distance* atau *interpupillary distance (IPD)*) adalah kurang dari ( $<$ ) 30 *pixel*. Informasi yang didapatkan pada tingkat ini seperti bentuk wajah dan warna kulit secara global. Informasi lainnya seperti karakteristik laki-laki atau perempuan dan perbedaan ras.
2. *Level 2 details*: tingkat ini jauh lebih baik dibanding sebelumnya dimana informasi yang didapatkan seperti struktur wajah seperti mata, hidung atau mulut, hubungan antar komponen wajah dan bentuk wajah yang lebih presisi. Pada tingkat ini, sudah bisa diaplikasikan ekstraksi fitur seperti geometrik atau tesktur. Rentang skor IPD berada pada nilai 30 hingga 75 *pixel*.
3. *Level 3 details*: tingkat ini memiliki keragaman informasi yang lebih banyak hingga fitur tingkat mikro seperti luka, bintik matahari, jerawat, hingga belang pada wajah dapat terlihat. Nilai skor IPD pada tingkat ini lebih besar dari ( $>$ ) 75 IPD.

### 2.2.9 Face Sketch Recognition

Face sketch recognition adalah pengenalan sketsa wajah secara otomatis oleh komputer untuk mengetahui identitas yang serupa atau mirip dengan sketsa wajah. Dari segi pendekatan atau penggambarannya (Bhatt, et al., 2012), sketsa wajah terbagi menjadi:

Tabel 2.3 Jenis Sketsa

No	Jenis	Keterangan	Kelebihan	Kekurangan
1	<i>Composite sketches</i>	merupakan sketsa wajah yang dihasilkan oleh komputer atau disebut juga dengan <i>software generated sketches</i> dimana pengguna dapat memilih pilihan komponen wajah	Dilakukan oleh komputer sehingga hasilnya baik. Dapat menghasilkan atau <i>generate</i> foto.	Bergantung pada variasi galeri yang dimiliki oleh aplikasi ( <i>inter-software variation</i> ). Masing-masing aplikasi memiliki kekurangannya sendiri
2	<i>Viewed sketches</i>	merupakan sketsa wajah yang di gambar oleh <i>artist</i> dengan melihat ke objeknya atau fotonya	Sketsa yang dihasilkan lebih baik dari <i>semi-forensic sketches</i> .	Tidak handal digunakan pada ranah forensik
3	<i>Semi-forensic sketches</i>	merupakan sketsa wajah yang di gambar berdasarkan ingatan <i>artist</i>	Sketsa yang dihasilkan lebih baik dari <i>forensic sketches</i> .	Tidak handal digunakan pada ranah forensik serta <i>noise</i> atau informasi yang hilang tergantung pada ingatan <i>artist</i> .
4	<i>Forensic sketches</i>	merupakan sketsa wajah yang di gambar berdasarkan keterangan yang diberitahu oleh saksi mata. Sketsa forensik ini telah di pakai sejak abad 19 (McQuiston-	Tidak dikerjakan oleh komputer. <i>Artist</i> fokus menggambar pada ciri khas dari saksi mata	Sketsa yang dihasilkan oleh masing-masing <i>artist</i> berbeda ( <i>inter-artist variation</i> ) karena bergantung pada keahlian dari <i>artist</i> dalam menerjemahkan keterangan saksi mata menjadi



		Surrett, et al., 2006)		sketsa (Nagpal, et al., 2016).
--	--	------------------------	--	--------------------------------

### 2.2.10 Heterogeneous Matching

Secara definisi pada biometrik, *heterogeneous matching* merupakan istilah digunakan untuk memadankan dua buah citra yang diperoleh dari sensor yang berbeda atau berdasarkan perbedaan *spectral band* atau dapat dikatakan demikian jika kedua citra tersebut mempunyai karakteristik informasi yang berbeda (Li & Jain, 2015) (Peng, et al., 2017). Sebagai contohnya adalah citra foto (visual, VIS) dengan *near infrared image* (NIR) atau *thermal infrared* (TIR), sketsa atau citra 3-dimensi.

Seperti yang diterangkan oleh Zhu *et al* (Zhu, et al., 2014) dan Wu *et al* (Wu, et al., 2017), ada tiga metode pada *heterogeneous matching*, yakni:

1. **Data synthesis:** metode ini merubah data pada suatu modal menjadi modal yang lain sehingga similaritas dari data yang berbeda domain tersebut dapat diukur. Contohnya adalah Wang *et al* (Wang & Tang, 2009) menggunakan *multiscale Markov Random Fields* untuk mensintesis sebuah sketsa dari foto yang diberikan dan juga sebaliknya. Kekurangan dari metode ini adalah membutuhkan pasangan sketsa-foto yang cukup banyak sehingga dapat melakukan *mapping* dengan baik.
2. **Latent subspace learning:** memproyeksikan kedua modal menjadi suatu *common latent space* yang relevan terhadap keduanya sehingga dapat di ukur. Metode ini akan mencari *subspace* dimana kedua data berbeda modal memiliki representasi yang sama. Metode yang dapat digunakan antara lain PCA, *Canonical Correlation Analysis* (CCA), *Common Discriminant Feature Extraction* (CDFE) dan *Partial Least Square* (PLS) (Zhu, et al., 2014).
3. **Modal-invariant features based methods:** menggunakan fitur lokal *modality-invariant* yang hanya berkaitan dengan identitas wajah. Metode ini menggunakan ekstraksi fitur seperti LBP, DoG Filtering atau LFDA (Klare, et al., 2011) (Zhu, et al., 2014) (Wu, et al., 2017).

### 2.2.11 Local Invariant Features Method

*Local invariant features method* merupakan sebuah metode yang digunakan untuk mengambil kumpulan fitur yang merepresentasikan keseluruhan dari citra tersebut. Fitur ini disebut juga dengan *keypoint*. Metode ini terbagi menjadi dua kategori (Chen, et al., 2010), yakni:

1. *Sparse descriptor:* merupakan deskriptor yang melakukan pencarian titik minat (*interest point*) lalu membuat *local patch* dan melakukan

ekstraksi fitur serta mendeskripsikannya. Contoh *sparse descriptor* adalah SIFT (Lowe, 2004)

2. *Dense descriptor*: merupakan deskriptor yang mengekstraksi fitur lokal dari setiap *pixel*. Contoh *dense descriptor* adalah *local binary patter* dan *Weber Local Descriptor*.

### 2.2.12 Sliding window

*Sliding window* merupakan fungsi yang memotong citra menjadi  $M \cdot N$  *window*. Nilai yang digunakan pada *sliding window* memiliki dua parameter yang digunakan dalam operasinya, yakni ukuran *window*  $s$  dan ukuran perpindahan (*displacement size*)  $\delta$ . Ukuran *window* merupakan ukuran dari region atau wilayah yang akan dipotong pada citra. Sedangkan ukuran perpindahan (dalam satuan *pixel*) merupakan besaran perpindahan dari satu sampel ke sampel yang lain. Jika besaran ukuran perpindahan kurang dari ukuran *window*, maka yang terjadi adalah *overlapping window* yakni masing-masing sampel memiliki nilai yang sebagiannya merupakan nilai dari sampel sebelumnya.

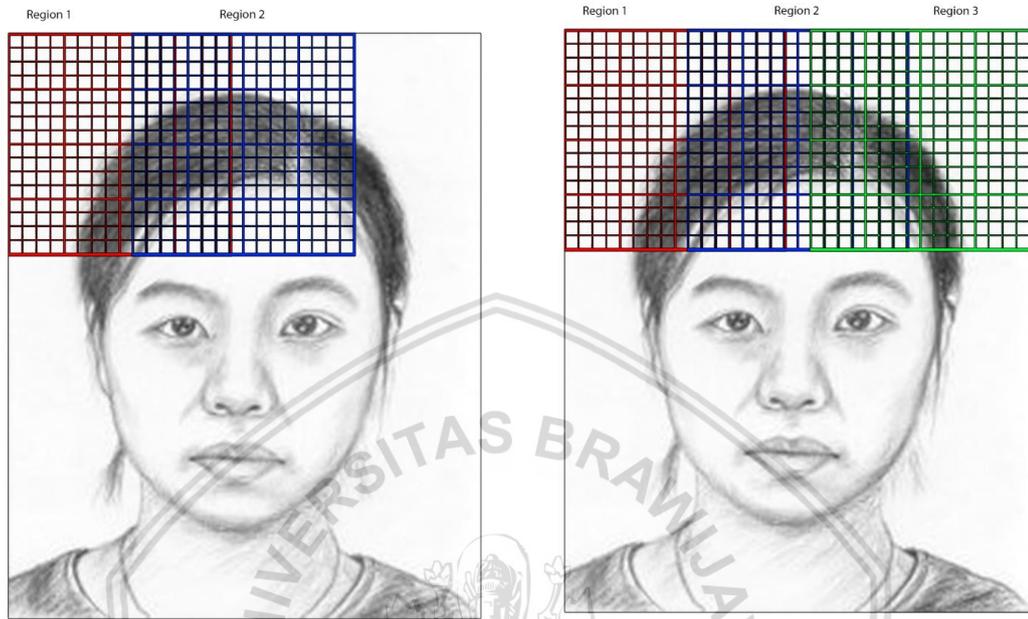
Dari citra ukuran  $H \cdot W$  menjadi  $M \cdot N$  *window*, dimana  $M = \frac{W-s}{\delta+1}$  dan  $N = \frac{H-s}{\delta+1}$ . Ukuran vektor fitur yang didapat adalah  $M \times N$  patches yang digabungkan menjadi satu  $M \times N$  dimensi vektor. Fitur vektor ini disimbolkan dengan (simbol sigma). Berikut adalah contoh dari implementasi *sliding window*.



Gambar 2.3 Ilustrasi *Sliding window*

Ukuran *window* yang digunakan adalah 16x16 *pixel* dan ukuran perpindahan yang digunakan sebesar 9 *pixel*, sehingga *region* kedua dan ketiga akan tampak sebagai berikut. *Sliding window* ini dilakukan sebanyak  $M \cdot N$  kali.

Hasil dari *sliding window* ini akan digunakan untuk ekstraksi fitur oleh SIFT, sehingga jumlah fitur SIFT ada  $128$  (*vektor fitur SIFT*)  $\times M \cdot N$  fitur, sedangkan MCWLD memiliki  $432$  (*vektor fitur MCWLD*)  $\times M \cdot N$  fitur.



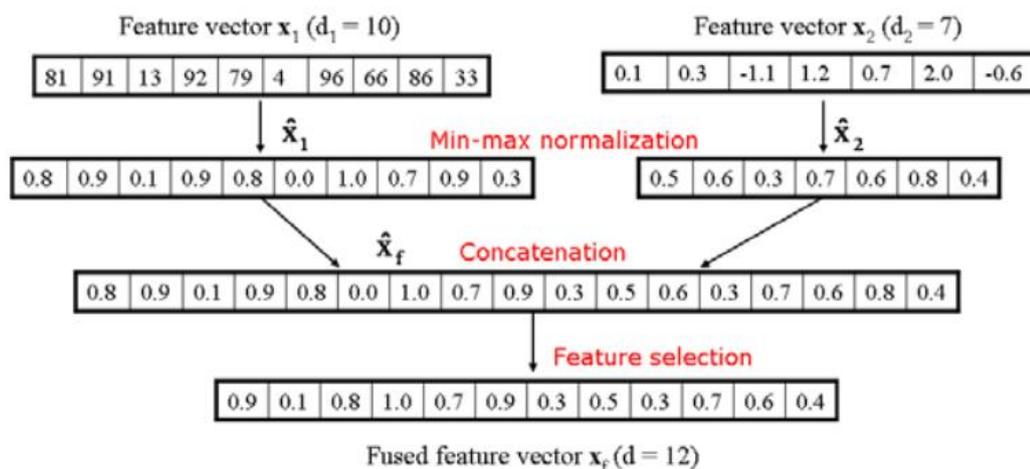
Gambar 2.4 Ilustrasi *overlapping Window*

### 2.2.13 Min-Max Normalization

*Min-max normalization* adalah suatu metode yang digunakan untuk merubah nilai pada suatu data menjadi satu tertentu. Pada *min-max normalization*, yang digunakan adalah nilai minimal pada data tersebut sebagai batas paling bawah dari serta nilai maksimalnya sebagai acuan paling atas. *Min-max normalization* merubah rentang nilai pada suatu data menjadi 0 hingga 1 dimana 0 adalah nilai minimal dari data tersebut dan 1 adalah nilai paling tinggi dari data.

$$n'_i = \frac{(n_i - (\min\{N\}))}{(\max\{N\} - \min\{N\})} \quad (2.5)$$

*Min-max normalization* dalam penelitian ini digunakan untuk melakukan fusi pada tingkat fitur maupun pada tingkat *score*. Untuk melakukan fusi pada tingkat fitur menggunakan *min-max normalization*, dilakukan operasi pada Gambar 2.5. Dari gambar tersebut, kedua fitur yang telah dinormalisasi dapat disatukan menjadi suatu fitur baru.



**Gambar 2.5** *Min-max normalization* pada fusi tingkat fitur (Jain, et al., 2007)

Sedangkan pada tingkat *score* yang dilakukan adalah melakukan normalisasi nilai jarak antara *probe* dengan galeri (Klare & Jain, 2010). Seperti pada persamaan dibawah ini, yang dilakukan adalah melakukan normalisasi terlebih dahulu baru kemudian dijumlahkan menjadi nilai jarak yang baru.

$$D'^i = \frac{(D^i - \min\{D\})}{(\max\{D\} - \min\{D\})} \tag{2.6}$$

$$D^{fusi} = D^{SIFT} + D^{MCWLD} \tag{2.7}$$

### 2.2.14 Scale Invariant Feature Transform (SIFT)

Metode SIFT merupakan metode yang dicetuskan oleh David Lowe (Lowe, 2004) untuk mengenali sebuah objek pada suatu foto dimana objek tersebut tertutupi oleh objek lain, adanya pencahayaan yang berbeda, perbedaan skala atau orientasinya berbeda. Metode SIFT akan menyeleksi banyak titik pada citra menggunakan DOG (*Difference of Gaussian*) sehingga yang tersisa hanyalah titik yang invariant terhadap perubahan skala. Kumpulan titik yang terseleksi ini menjadi kandidat *keypoint* yang akan diseleksi kembali, dimana letak titik tersebut tidak boleh berada pada tepi. Untuk mendeteksinya menggunakan *harris corner*. Pada penelitian ini, SIFT yang digunakan hanya pada ekstraksi fitur dan pembentukan deskriptor. Dari  $16 \times 16$  *window*, dibagi kembali menjadi  $4 \times 4$  *region* yang lebih kecil. Pada *region* yang lebih kecil inilah, proses ekstraksi fitur SIFT dilakukan. Fitur pada SIFT terbentuk dari orientasi dan besaran gradiennya, dengan rumus sebagai berikut:

$$M_{i,j} = \sqrt{(I_{i,j} - I_{i+1,j})^2 + (I_{i,j} - I_{i,j+1})^2} \tag{2.8}$$

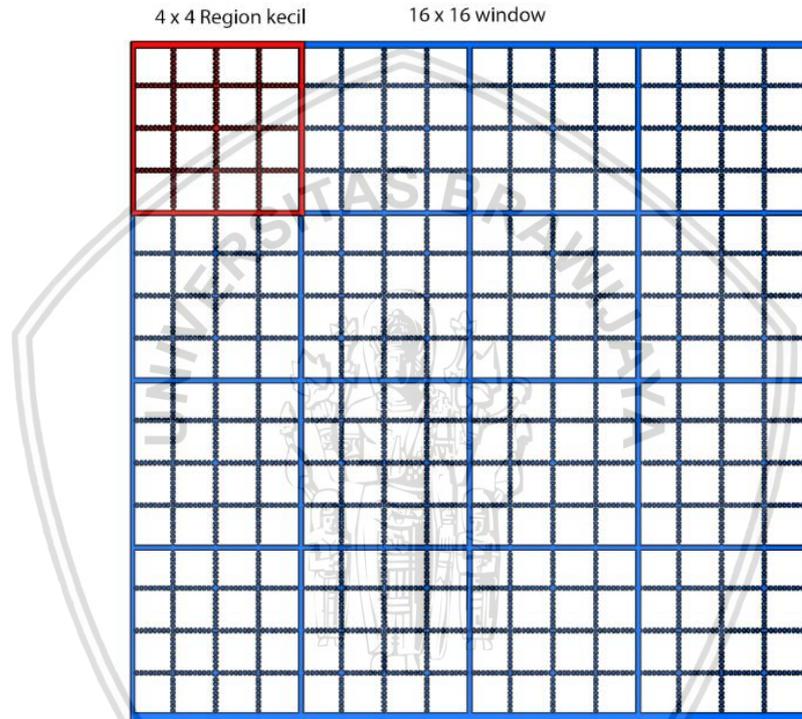


$$R_{i,j} = \text{atan2}(I_{i,j} - I_{i+1,j}, I_{i,j} - I_{i,j+1}) \quad (2.9)$$

Dimana  $I$  adalah *window* dari *sliding window*,  $M$  adalah *magnitude* dari gradien,  $R$  adalah radian yang akan diubah menjadi derajat,  $i$  dan  $j$  adalah indeks baris dan kolom.

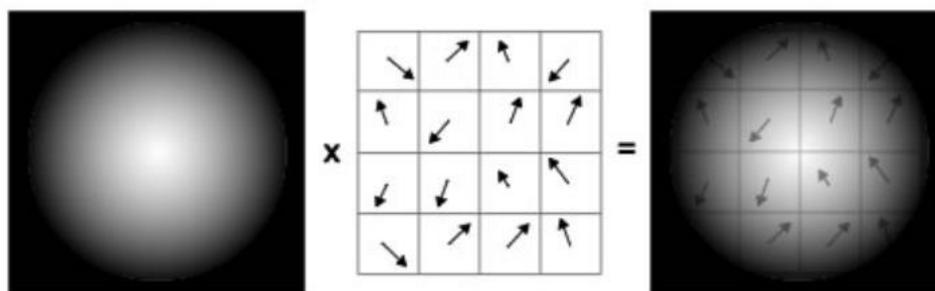
Tahapan pembentukan deskriptor SIFT adalah sebagai berikut:

1. Operasi ini dilakukan pada *window* berukuran 16 x 16 hasil dari *sliding window*. Masing-masing *window* ini akan dibagi menjadi 16 *region* kecil. Dari *region* kecil tersebutlah, operasi mengekstraksi besaran gradien dan orientasi dilakukan.



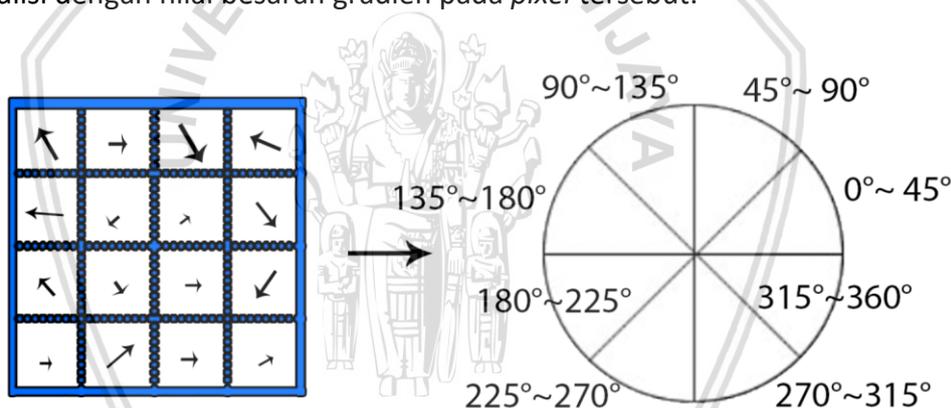
2. *Orientation extraction*: Proses ini hanya menghitung orientasi dari setiap *pixel* menggunakan rumus 2.2. Nilai dari rumus tersebut menghasilkan nilai radian yang perlu diubah dahulu menjadi nilai derajat. Nilai dari orientasi ini nantinya digunakan untuk mengelompokkan nilai besaran gradien yang didapat pada *pixel* yang bersangkutan.
3. *Magnitude gradient extraction*: Proses ini mengekstraksi nilai besaran gradien
4. *Gaussian weighting*. Nilai besaran gradien dikalikan dengan fungsi gaussian sehingga nilai yang jauh dari titik tengah memiliki nilai yang lebih kecil dibanding yang dekat. Proses ini hanya mengalikan nilai fungsi gaussian dengan nilai yang besaran gradien.





Gambar 2.6 Pembobotan dengan *gaussian weighting*

5. *Histogram binning*: Proses selanjutnya adalah *histogram binning*. Histogram yang dibuat terdiri dari 8 bin dengan masing-masing bin memiliki rentang nilai sebesar  $45^\circ$ . Sebagai contoh, jika nilai orientasi pada suatu *pixel* bernilai  $35^\circ$  maka nilai tersebut berada pada rentang  $0^\circ - 45^\circ$ . Nilai pada masing-masing bin tersebut diisi dengan nilai besaran gradiennya. Jadi pada *pixel* dengan nilai  $35^\circ$  tadi, pada bin  $0^\circ - 45^\circ$  akan diisi dengan nilai besaran gradien pada *pixel* tersebut.



Gambar 2.7 *Histogram Binning*

Nilai orientasi dibagi menjadi 8 arah, dengan rentang sebagai berikut:

1.  $0^\circ - 45^\circ$
2.  $45^\circ - 90^\circ$
3.  $90^\circ - 135^\circ$
4.  $135^\circ - 180^\circ$
5.  $180^\circ - 225^\circ$
6.  $225^\circ - 270^\circ$
7.  $270^\circ - 315^\circ$
8.  $315^\circ - 360^\circ$

Pada Gambar 2.7 terdapat ilustrasi sebuah region kecil dengan masing-masing *pixel*-nya terdapat sebuah orientasi dan panjang-pendek dari panah tersebut menggambarkan besaran gradiennya. Dari region tersebut, nilai besaran gradien dikelompokkan terhadap 8-bin yang sesuai dengan nilai

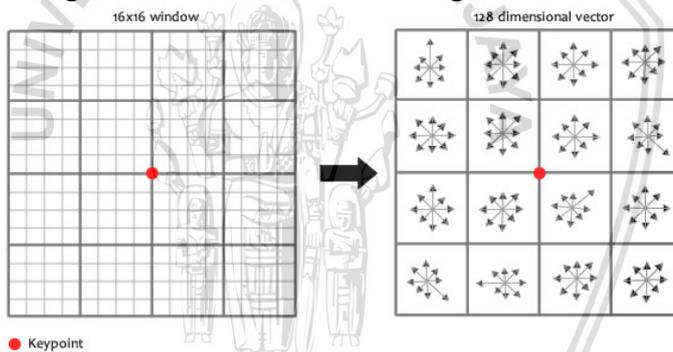


orientasinya. Nilai besaran gradien yang dimasukkan dalam bin juga bergantung pada jauh dekatnya dengan titik tengah pada *window*. Hal ini dilakukan dengan melakukan pembobotan dengan *gaussian-weighted circular window*. Sehingga nilai yang jauh dari titik tengah *window* akan mempunyai nilai besaran magnitude yang lebih rendah dibanding yang lebih dekat. Pembobotan ini dilakukan dengan mengalikan nilai besaran gradien dengan bobot atau nilai *gaussian*. Berikut adalah rumus *gaussian* yang digunakan:

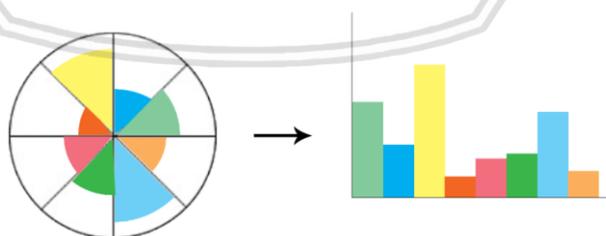
$$f(y, x) = \frac{1}{2 * \pi} e^{-\frac{x^2+y^2}{std^2}} \tag{2.10}$$

6. *Keypoint descriptor*:

Proses selanjutnya adalah melakukan komputasi descriptor berdasarkan posisi lokal *keypoint* pada citra. Salah satu pendekatan untuk mendapatkan sampel dari posisi lokal *keypoint* citra adalah membuat sebuah *windows* berukuran 16 x 16 disekitar *keypoint* lalu dibagi menjadi 4 x 4 region (1 region sama dengan 4 x 4 *window*). Masing-masing region mempunyai 8 besaran gradien, sehingga dapat dibentuk vektor 128-dimensi. Pada setiap region, besaran gradien dan orientasi di hitung.



Gambar 2.8 Representasi perhitungan orientasi



Gambar 2.9 Ilustrasi fitur SIFT

Operasi ini dilakukan terhadap 16 *pixel* yang mempunyai 16 arah orientasi yang acak dan dikelompokkan pada histogram 8-bin. Operasi ini dilakukan pada region ukurn 4x4, sehingga didapatkan 4x4x8 = 128 nilai yang disimpan dalam bentuk vektor. Nilai tersebut di normalisasi dengan *difference sum of squares*.



### 2.2.15 Multiscale Circular Weber Local Descriptor (MCWLD)

MCWLD merupakan deskriptor kedua yang digunakan pada penelitian ini. MCWLD bersifat *sparse descriptor*, namun pada penelitian ini ekstraksi fiturnya tetap dilakukan menggunakan *sliding window*-nya. Ada dua ekstraksi nilai yang dilakukan oleh MCWLD, yakni *differential excitation* dan *Gradient Orientation*. Baik *differential excitation* maupun *Gradient Orientation*, keduanya dilakukan pada *window* dari *sliding window*.

Untuk mendapatkan nilai *differrection excitation*, langkah yang perlu dilakukan adalah sebagai berikut:

1. Menentukan titik tengah dari *window*, sebagai  $x_c$  dan radius, yang merupakan jarak antara titik tengah tersebut dengan tetangganya.
2. Untuk setiap *pixel* tetangganya, nilai tersebut dikurangi dengan titik tengah dan dibagi dengan nilai titik tengah tersebut. Hasil dari pengurangan tersebut dijumlahkan.
3. Penjumlahan dari pengurangan yang telah dilakukan sebelumnya, dicari nilai radiannya menggunakan fungsi *arctan*.

*Multiscale Circular Weber Local Descriptor* membagi citra menjadi beberapa bagian menggunakan *sliding window*. Metode ini memiliki dua komponen utama pembentuk fiturnya, yakni *differential excitation* dan *Gradient Orientation*. MCWLD ini memiliki dua parameter yakni  $P$  dan  $R$ , dimana  $P$  adalah jumlah *pixel* tetangga yang terpisah secara merata pada lingkaran dengan radius  $R$  yang berpusat pada *pixel* saat ini atau *pixel* ke- $i$ . *Multiscale* pada MCWLD dimaksudkan pada variasi parameter  $P$  dan  $R$ . Berikut adalah representasi fitur dari MCWLD:

1. *differential excitation*: merupakan fungsi *arctan* dari penjumlahan rasio yang dihasilkan dari pengurangan *pixel* yang berada di tengah dengan *pixel* tetangganya kemudian dibagi dengan nilai *pixel* tengah itu sendiri

$$\xi(x_c) = \arctan \left\{ \sum_{i=0}^{p-1} \left( \frac{x_i - x_c}{x_c} \right) \right\} \tag{2.11}$$

Jika nilai  $\xi(x_c)$  positif maka *pixel* tetangganya lebih terang dibanding dengan *pixel* di tengah. Jika hasilnya negatif maka *pixel* tetangganya lebih gelap dibanding dengan *pixel* di tengah (Chen, et al., 2010).

2. *Gradient Orientation*: Untuk mendapatkan orientasi dari  $x_c$ , menggunakan rumus berikut:

$$v_s^{10} = x_{\left(\frac{p}{2}+R\right)} - x_{(R)} \tag{2.12}$$

$$v_s^{11} = x_{(P-R)} - x_{\left(\frac{p}{2}-R\right)}$$



$$\theta(x_c) = \arctan \left\{ \frac{v_s^{11}}{v_s^{10}} \right\}$$

Sama seperti SIFT, hasil dari orientasi ini diubah menjadi sebuah histogram, hanya saja berdasarkan  $T$ -orientasi yang dominan ( $T$ -dominant orientation. Nilai  $T$  yang diujikan oleh penulis adalah 8). Dalam proses ini, nilai dari orientasi tersebut akan di *quantization* menjadi sebuah nilai yang seragam dengan jumlah  $T$ -dominant orientation. Proses *quantization* ini merubah  $\theta$  yang mempunyai rentang nilai  $[-\frac{\pi}{2}, \frac{\pi}{2}]$  menjadi  $\theta'$  dengan rentang nilai  $[0, 2\pi]$ . Untuk melakukan *quantization* nilai  $\theta$  tersebut maka dilakukan pemetaan nilai  $\theta$  menjadi  $\theta'$  rumus sebagai berikut:

$$\theta' = \arctan2(v_s^{11}, v_s^{10}) + \pi \tag{2.13}$$

dimana

$$\arctan2(v_s^{11}, v_s^{10}) = \begin{cases} \theta, & v_s^{11} > 0 \text{ dan } v_s^{10} > 0 \\ \pi + \theta, & v_s^{11} > 0 \text{ dan } v_s^{10} < 0 \\ \theta - \pi, & v_s^{11} < 0 \text{ dan } v_s^{10} < 0 \\ \theta, & v_s^{11} < 0 \text{ dan } v_s^{10} > 0 \end{cases}$$

Kemudian untuk rumus untuk *quantization* adalah berikut:

$$\Phi_t = f_q(\theta') = \frac{2t}{T} \pi \tag{2.14}$$

dimana

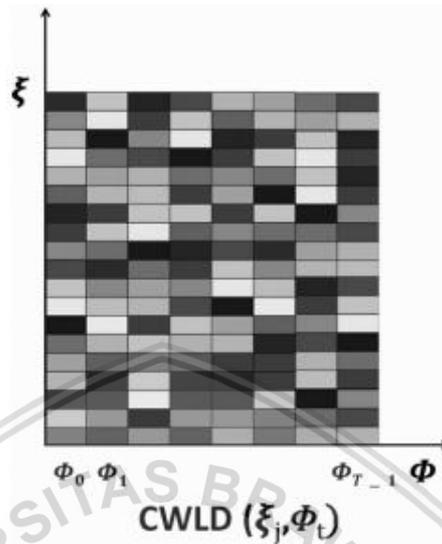
$$t = \text{mod} \left( \left\lfloor \frac{\theta'}{2\pi} + \frac{1}{2} \right\rfloor, T \right)$$

3. *Circular WLD Histogram*: Pada setiap *window* kecil pada potongan *window* dilakukan perhitungan *differential excitation* dan *Gradient Orientation*, dan dikelompokkan ke dalam sebuah histogram 2-D dimana kolomnya adalah nilai dari orientasi yang dominan,  $\Phi_t$ , dan barisnya merupakan nilai *differential excitation* tertentu. Dengan demikian, nilai pada masing-masing sel adalah nilai frekuensi dari *differential excitation* pada *dominant orientation*.

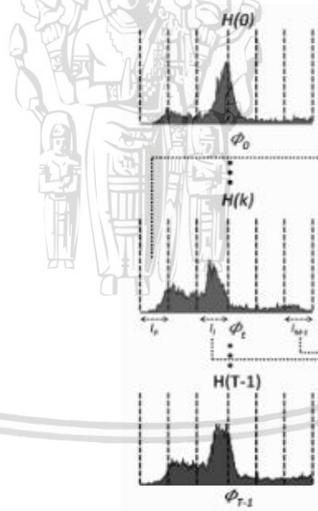
Dimana nilai  $j = 0, 1, \dots, N-1$  dan  $t=0, 1, \dots, T-1$  dan  $N$  merupakan ukuran dimensi dari citra. Berikut adalah empat langkah untuk menghitung CWLD berdasarkan Chen (Chen, et al., 2010):



- a. Dari histogram 2-D yang dihasilkan, dibagi menjadi beberapa histogram 1-D berdasarkan *T-dominant orientation*. Histogram ini di simbolkan dengan  $H(t)$ , dimana  $t = 0, 1, \dots, T-1$ .



Gambar 2.10 Histogram 2-D dari *Differential excitation* dan orientasi dari citra wajah. Sumber (Bhatt, et al., 2012)



Gambar 2.11 Sub-histogram dari differential excitaion di orientasi T. Sumber (Bhatt, et al., 2012)

- b. Pada masing-masing histogram, nilai rentang *differential excitation* dibagi menjadi *M-Segment* yang nantinya menjadi histogram sendiri. *M-Segment* terdiri dari  $l_m$  interval dimana  $l_m$  terdiri dari batas atas dan batas bawah (sehingga dapat dituliskan dengan  $l_m = [\eta_{m,l}, \eta_{m,u}]$ ). Masing-masing segmen, mempunyai rentang yang didapat dengan rumus berikut:



$$\text{batas atas} = \eta_{m,u} = \left( \frac{(m+1)}{M} - \frac{1}{2} \right) \pi \quad (2.15)$$

$$\text{batas bawah} = \eta_{m,l} = \left( \frac{m}{M} - \frac{1}{2} \right) \pi \quad (2.16)$$

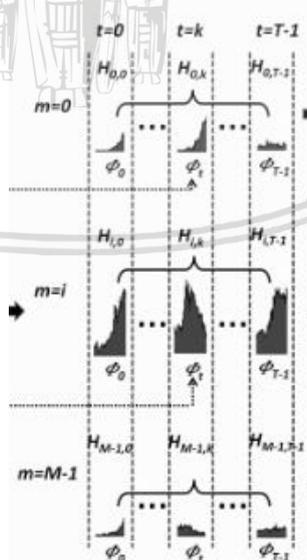
- c. Untuk membagi  $H(t)$ , jumlah  $M$  di tentukan oleh penulis adalah 6. Pada setiap rentang atau subhistogram, pembagian bin di representasikan sebagai berikut:

$$H_{m,t} = h_{m,t,s} \quad (2.17)$$

dimana  $s = 0, 1, \dots, S-1$  dengan nilai  $S$ -bin di tentukan oleh penulis adalah 3 dan  $h_{m,t,s}$  di representasikan

$$h_{m,t,s} = \sum_j \delta(S_j == s), \left( S_j = \left\lfloor \frac{\xi_j - \eta_{m,l}}{(\eta_{m,u} - \eta_{m,l})/S} + \frac{1}{2} \right\rfloor \right) \quad (2.18)$$

dimana  $j = 0, 1, \dots, N-1$ ,  $m$  adalah interval pada *differential excitation*  $\delta_j$ . Untuk  $\eta_{m,l} = \left( \frac{m}{M} - \frac{1}{2} \right) \pi$  merupakan batas bawah dan  $\eta_{m,u} = \left( \frac{m+1}{M} - \frac{1}{2} \right) \pi$  adalah batas atas dan proses pada  $\delta(\cdot)$  merupakan fungsi *Kronecker delta* yang menghasilkan nilai 1 jika kondisi tersebut benar dan 0 jika kondisi tersebut salah.



**Gambar 2.12** Matriks histogram. Sumber: (Bhatt, et al., 2012)

- d. Subhistogram yang telah didapatkan diubah menjadi  $M$ -Segment histogram 1 dimensi



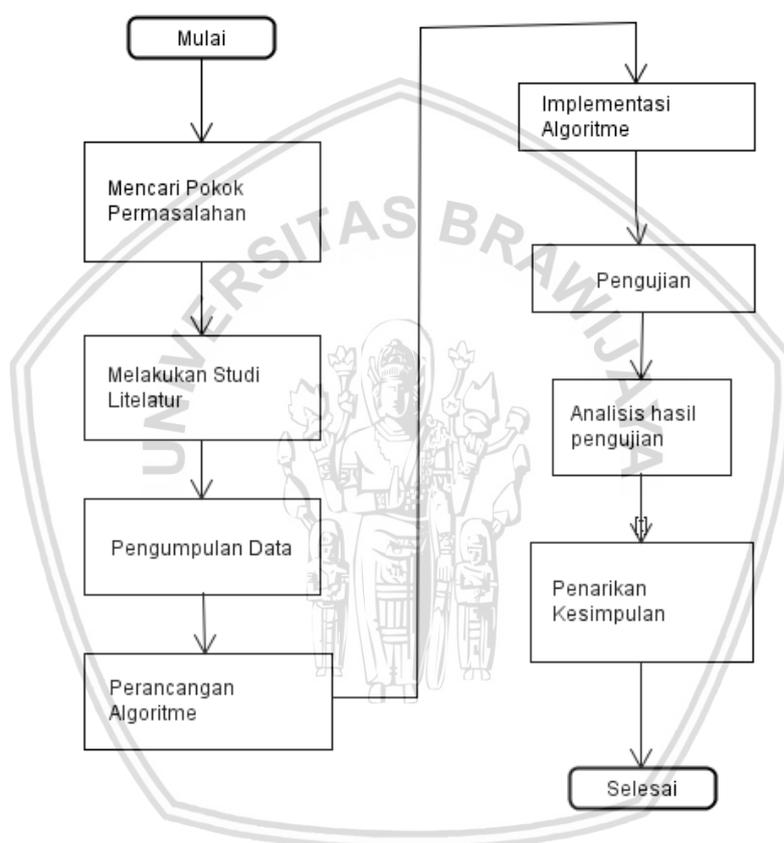
- e.  $M$ -jumlah subhistogram kemudian dijadikan satu histogram yang merepresentasikan  $6 \times 8 \times 3$  ( $M \times T \times S$ ) histogram *circular WLD*. Alasan membagi *differential excitation* ini dimaksudkan untuk menghitung variasi pada citra wajah tertentu dan digunakan untuk menetapkan bobot optimal pada histogram  $H_m$  untuk memperbaiki kinerja dari deskriptor CWLD
4. *Multiscale Circular WLD: multiscale* yang dimaksudkan oleh penulis adalah menggunakan variasi parameter  $P$  dan  $R$ . Nilai parameter  $R$  yang diujikan adalah 1, 2, 3 sedangkan untuk nilai  $P$  adalah 8, 16 dan 24. Citra wajah dibagi menjadi  $M \cdot N$  jumlah *window* menggunakan *sliding window*.



## BAB 3 METODOLOGI PENELITIAN

Dalam perancangan algoritme terdapat proses-proses yang saling berikatan sehingga menciptakan suatu perancangan yang terstruktur dengan baik. Pada bab ini akan dijelaskan langkah-langkah yang dilakukan dalam perancangan algoritme dimulai dari mencari pokok permasalahan hingga menghasilkan kesimpulan pada penarikan kesimpulan.

### 3.1 Alur Metode Penelitian



**Gambar 3.1 Diagram alir metode penelitian**

Alur penelitian diawali dengan mencari pokok permasalahan yang akan diangkat menjadi sebuah penelitian ini. Pokok Permasalahan yang diangkat ini nantinya akan dipecah menjadi beberapa sub-masalah yang harus di selesaikan. Setelah itu proses selanjutnya yakni mencari literatur yang mendukung untuk menjadi acuan dalam melakukan penelitian. Studi literatur menjadi bahan dasar atas teori dari metode yang diterapkan. Setelah mendapatkan bahan dasar penelitian, proses berikutnya adalah mengumpulkan data yang akan dibagi menjadi data latih, evaluasi serta data uji.

### 3.2 Studi Literatur

Pada tahap ini dilakukan pengumpulan literatur berasal dari jurnal, buku, pakar, informasi dari internet dan atau penelitian yang telah dilakukan sebelumnya. Studi literatur dilakukan untuk mendapatkan informasi tentang memecahkan masalah pada ranah *heterogeneous matching*, sistem biometrik dan pengenalan wajah, baik pada metode maupun representasi fiturnya.

### 3.3 Pengumpulan Data

Setelah studi literatur, langkah selanjutnya adalah menentukan data apa yang harus digunakan dalam penelitian ini. Pengumpulan data disesuaikan dengan hasil studi literatur dengan parameter yang telah ditentukan. Pengumpulan data berupa galeri foto serta sketsa yang didapatkan dari *Chinese University of Hong Kong*. Data tersebut telah menjadi acuan dalam pengembangan metode *heterogeneous matching* yang berjumlah 188 individu terdiri 54 relawan perempuan dan 134 relawan laki-laki.

### 3.4 Perancangan Algoritme

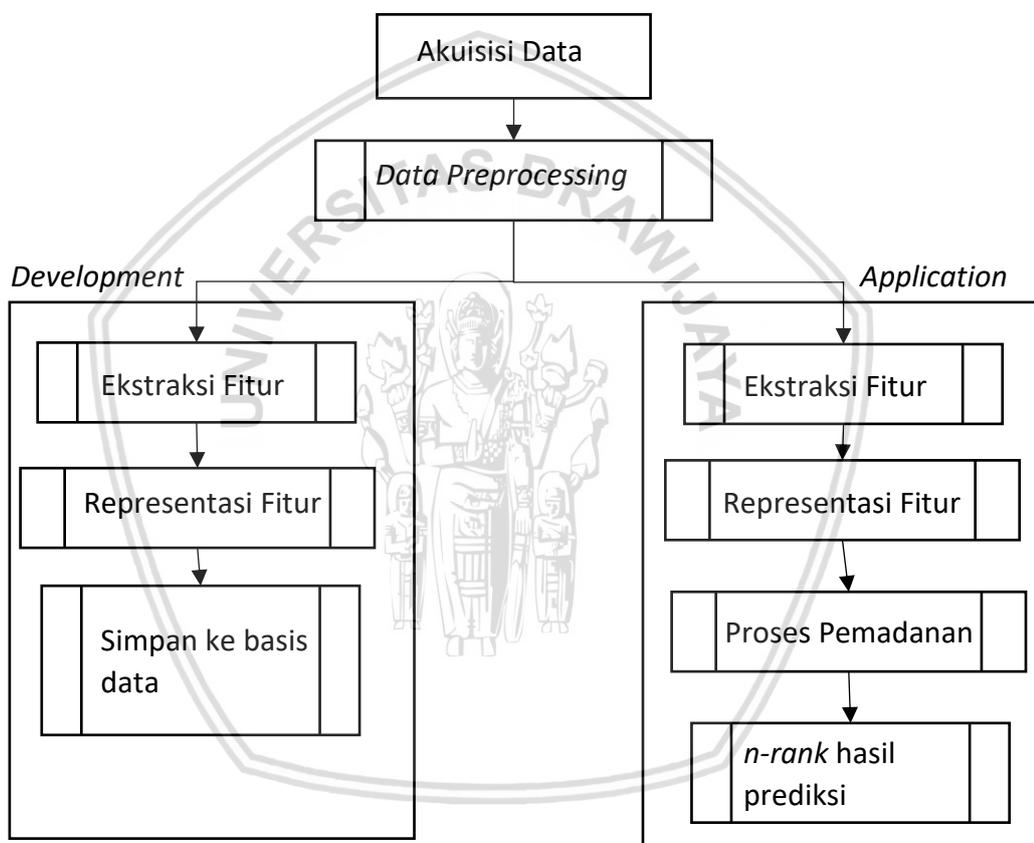
Perancangan algoritme adalah tahapan yang dilakukan untuk merancang tahapan algoritme untuk mencapai tujuan. Perancangan algoritme ini dibangun berdasarkan hasil pengumpulan data dan analisis kebutuhan yang telah dilakukan pada tahap sebelumnya. Menjabarkan mengenai bagaimana alur yang dilakukan dari *input* hingga menjadi *output* yang dihasilkan sesuai dengan tujuan. Pada perancangan algoritme ini, perancangan dilakukan untuk mempermudah implementasi, pengujian dan analisis. Adapun tahapan pada perancangan ini terdiri dari bagian *development* dan *application*.

Mengacu pada Gambar 3.2, alur kerja dari sistem yang dirancang adalah sebagai berikut:

1. Akuisisi data: Proses pengumpulan data yang diperoleh dari jurnal dengan tema yang sama (biometrik, *sketch recogniton*, *heterogeneous matching*) yang dianggap sebagai protokol atau *baseline* yang digunakan pada *sketch recognition*.
2. *Data Preprocessing*: merupakan proses untuk mengolah data sehingga dapat diproses pada proses berikutnya. Proses yang dilakukan pada bagian ini adalah melakukan *grayscale* dan *sliding window*.
3. **Bagian *development*:**
  - a. Ekstraksi fitur: proses melakukan ekstraksi fitur dari citra yang diberikan dengan metode SIFT atau MCWLD
  - b. Representasi fitur: fitur yang telah terkestraksi dikelompokkan sesuai dengan masing-masing individu dan atau melakukan fusi pada tingkat *score* atau fitur.
  - c. Simpan ke basis data: fitur tersebut disimpan pada basis sehingga pada proses *application* tidak perlu melakukan ekstraksi fitur pada data latih.

**4. Bagian *application*:**

- a. Ekstraksi fitur: proses melakukan ekstraksi fitur dari citra yang diberikan dengan metode SIFT atau MCWLD
- b. Representasi fitur: fitur yang telah terkestraksi disusun seperti representasi fitur dari data latih
- c. Proses pemadanan: melakukan proses pemadanan dengan fitur dari data latih dengan menghitung nilai jaraknya.
- d. *n-rank* hasil prediksi: melakukan pengurutan dari proses pemadanan dan mengembalikan sebanyak *n-rank* identitas yang menjadi kadidat dari individu yang dianggap sama.



**Gambar 3.2 Arsitektur Sistem**

**3.5 Implementasi Algoritme**

Tahapan implementasi pada skripsi ini mengacu pada perancangan yang telah dibuat. Sistem yang akan diimplementasikan melakukan ekstraksi fitur menggunakan metode SIFT atau MCWLD kemudian melakukan proses pemadanan antara fitur dari foto dengan sketsa. Adapun implementasi yang dilakukan pada penelitian memiliki batasan sistem sebagai berikut:



1. Pengolahan citra digital pada penelitian menggunakan *histogram binning OpenCV* dengan bahasa pemrograman Python 3 serta menggunakan *histogram binning NumPy* untuk mengolah matriks.
2. Implementasi ekstraksi fitur seperti SIFT dan MCWLD tidak menggunakan *histogram binning OpenCV* yang mengacu dari jurnal yang telah disitasi.
3. Masukan dari sistem ini berupa citra wajah dalam bentuk foto maupun sketsa.
4. Sistem ini mengembalikan  $n$ -jumlah kandidat yang diidentifikasi sebagai individu.
5. Pengujian dilakukan pada bagian *preprocessing*, pengujian parameter, pengujian fusi SIFT dan MCWLD pada tingkat *score* dan fitur.

### 3.6 Pengujian Algoritme dan analisis

Pada tahap pengujian algoritme ini menggunakan parameter yang disesuaikan dengan perancangan algoritme. Beberapa rancangan skenario pengujian algoritme adalah sebagai berikut:

1. Pengujian dilakukan pada bagian *preprocessing*,
2. pengujian parameter,
3. pengujian fusi SIFT dan MCWLD pada tingkat *score* dan fitur.

Dari rancangan skenario tersebut yang dilihat pertama kali adalah kemampuan algoritme tersebut untuk menghasilkan nilai *identification rate* yang baik. Tahap juga ini dilakukan untuk menganalisis hasil dari pengujian yang telah dilakukan dengan tujuan mengetahui kekurangan pada implementasi dari metode yang dilakukan. Hasil dari analisis ini diharapkan dapat menjadi bahan pertimbangan untuk penelitian selanjutnya.

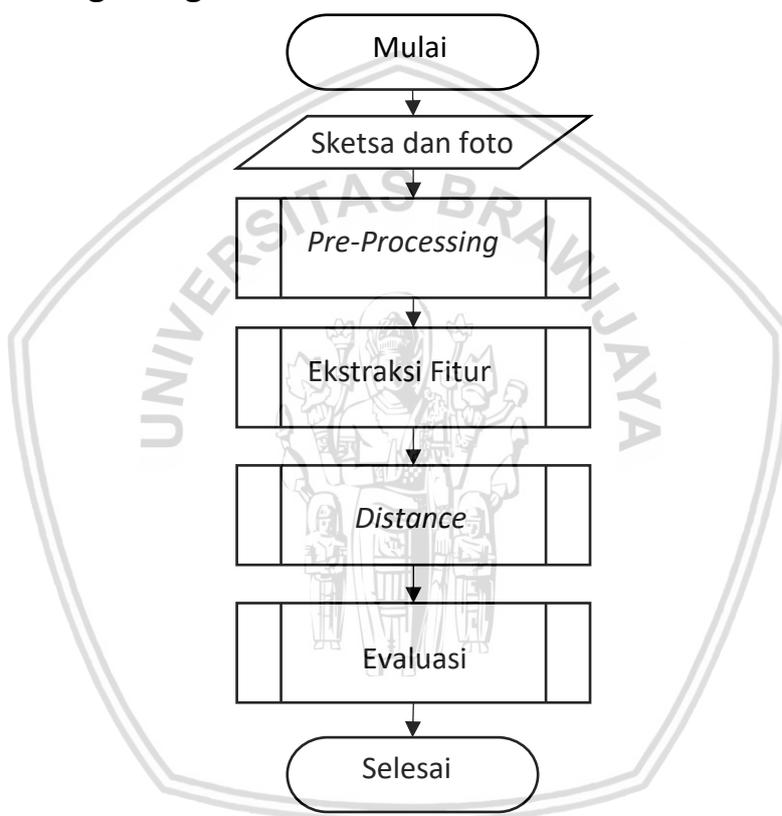
### 3.7 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah tahap perancangan, implementasi dan pengujian selesai dilakukan. Kesimpulan yang didapat berasal dari pengujian yang menjawab dari perumusan masalah. Tahap akhir dari penulisan laporan ini adalah adanya saran yang dimaksudkan untuk memperbaiki kekurangan dalam penelitian ini serta memberikan pertimbangan atas pengembangan metode selanjutnya.

## BAB 4 PERANCANGAN

Bab ini membahas perancangan yang dilakukan pada penelitian ini mencakup perancangan algoritme, skenario pengujian serta manualisasi dari algoritme atau metode yang digunakan. Bab perancangan ini digunakan sebagai acuan yang akan digunakan pada proses bab implementasi identifikasi individu berdasarkan sketsa wajah menggunakan pendekatan diskriminatif.

### 4.1 Perancangan Algoritme

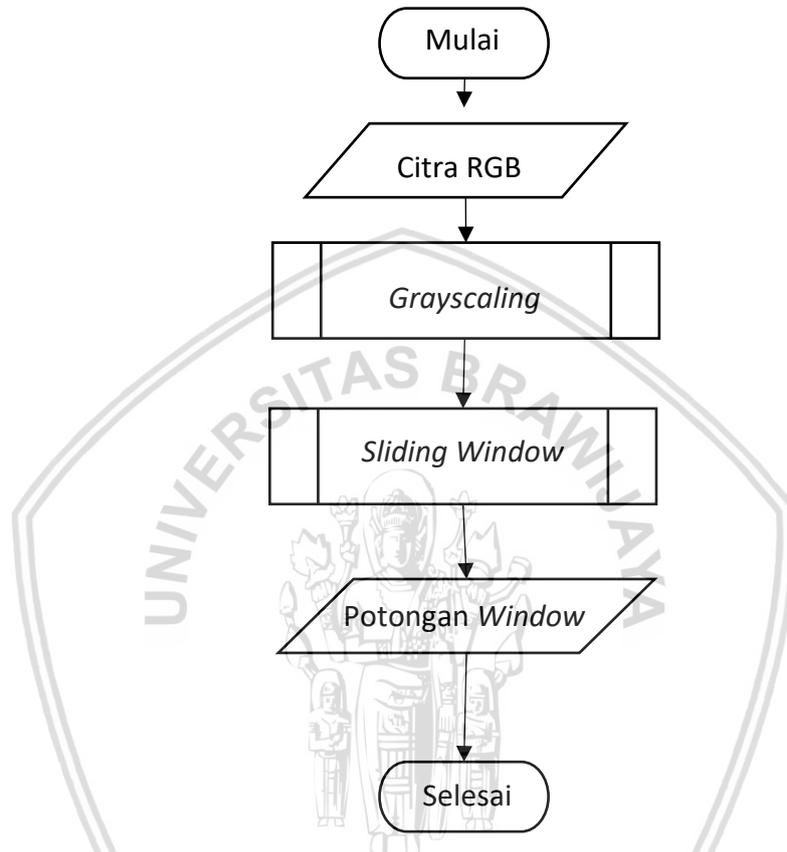


Gambar 4.1 Diagram alir Algoritme

Perancangan algoritme yang dilakukan pada penelitian ditunjukkan pada gambar 4.1. Perancangan algoritme ini menjadi acuan untuk mempermudah memahami alur proses yang dilakukan pada penelitian ini. Adapun langkah-langkah yang dilakukan pada algoritme ini adalah membaca berkas sketsa yang berkorespondensi dengan fotonya dan melakukan *preprocessing* terhadap citra sketsa maupun foto tersebut yakni *grayscale*. Kemudian dari citra tersebut (sketsa dan foto) diimplementasikan *sliding window* dan melakukan ekstraksi fitur menggunakan metode SIFT dan MCWLD pada masing-masing *window*nya. Fitur vektor ini akan disimpan pada basis data. Pada evaluasi ini, akan dibuatkan skenario pengujian yang dibahas pada subbab berikutnya.

### 4.1.1 Preprocessing

*Preprocessing* yang digunakan pada penelitian ini antara lain adalah *grayscale* dan *sliding window*. Tujuan dari *preprocessing* ini adalah untuk merubah citra RGB menjadi citra abu-abu. Citra abu-abu digunakan untuk mengekstraksi tekstur. Alur *preprocessing* dijelaskan pada Gambar 4.2.



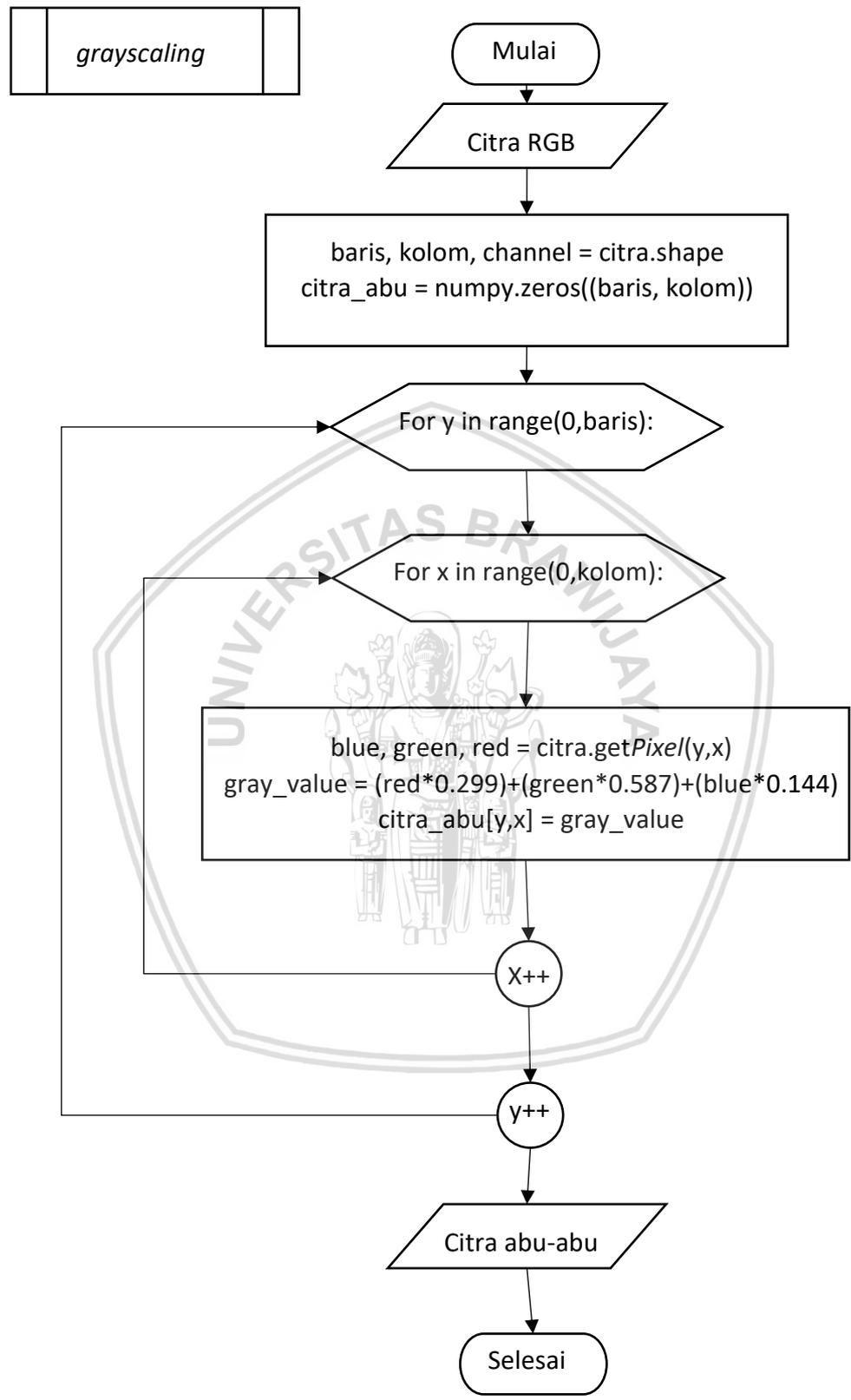
Gambar 4.2 Diagram alir *preprocessing*

Untuk alur proses *grayscale*, digambarkan pada Gambar 4.3. Nilai masing-masing *red*, *green*, dan *blue* didapat dari masing-masing *pixel* yang diakses melalui dua buah perulangan, yakni perulangan untuk baris dan perulangan untuk kolom. Adapun rumus yang digunakan pada perubahan nilai abu-abu oleh *histogram binning OpenCV* adalah sebagai berikut:

$$Gray Value = 0.299 * red\ value + 0.587 * green\ value + 0.144 * blue\ value$$

Nilai abu-abu yang didapat dari rumus tersebut disimpan pada nilai indeks baris dan kolom yang sama dengan saat mengakses nilai *red*, *green* dan *blue*. Seperti pada gambar 4.3, hasil dari *preprocessing* ini menghasilkan citra abu-abu yang siap digunakan untuk proses ekstraksi fitur.





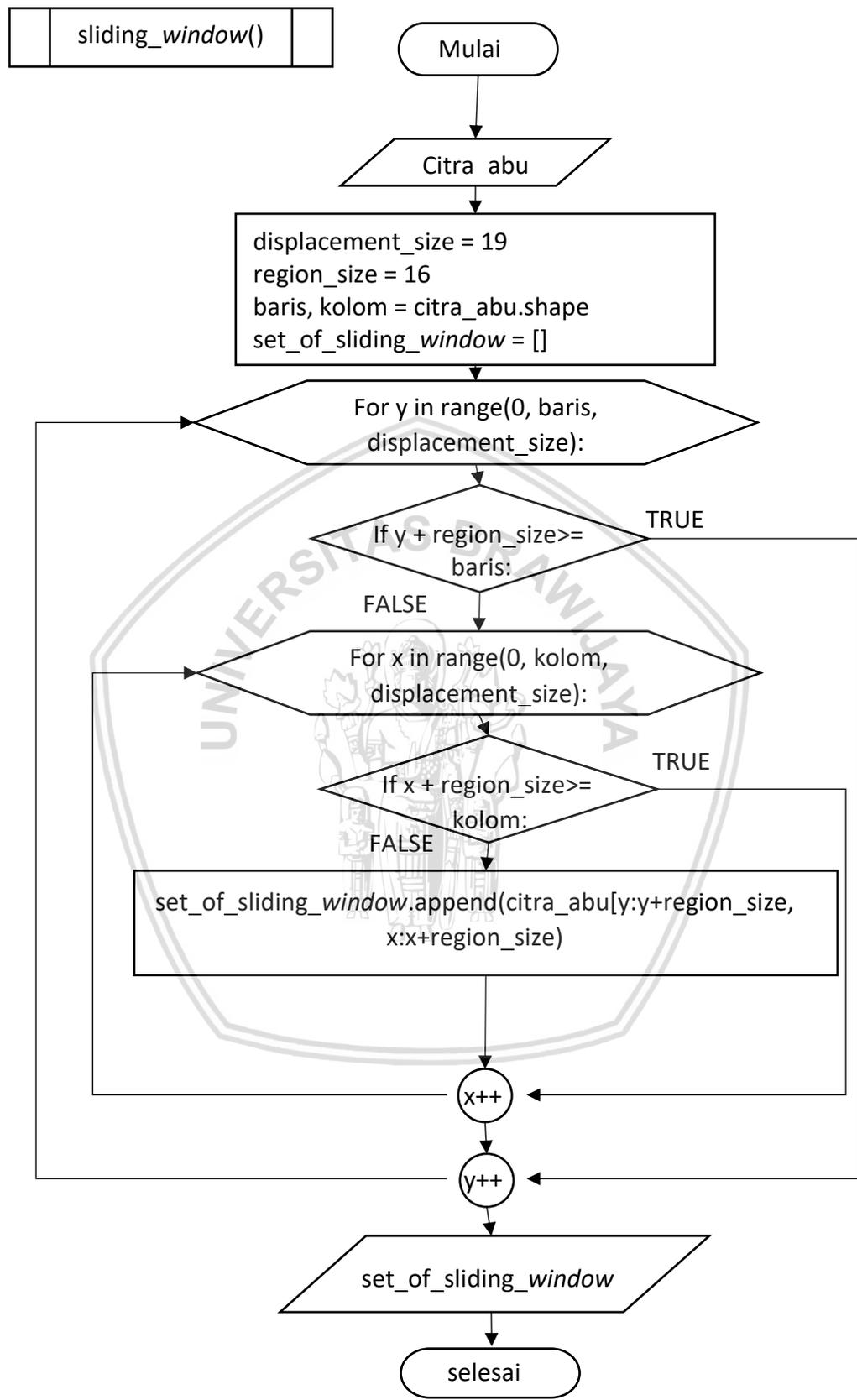
Gambar 4.3 Diagram Alir *grayscale*

Berikut merupakan penjelasan dari diagram alir 4.3:

1. Fungsi ini menerima *input* berupa citra yang memiliki nilai RGB atau 3 *channel* warna.
2. Selanjutnya, program akan mengambil nilai ukuran dari baris dan kolom
3. Untuk setiap nilai indeks baris dan kolom pada citra tersebut, diambil nilai *red*, *green*, *blue* untuk kemudian diproses menjadi nilai abu-abu menggunakan rumus 2.1. Nilai yang didapat akan disimpan pada citra baru di indeks baris dan kolom yang sama dengan saat mengambil nilai RGB.
4. Fungsi ini akan menghasilkan sebuah citra abu-abu.

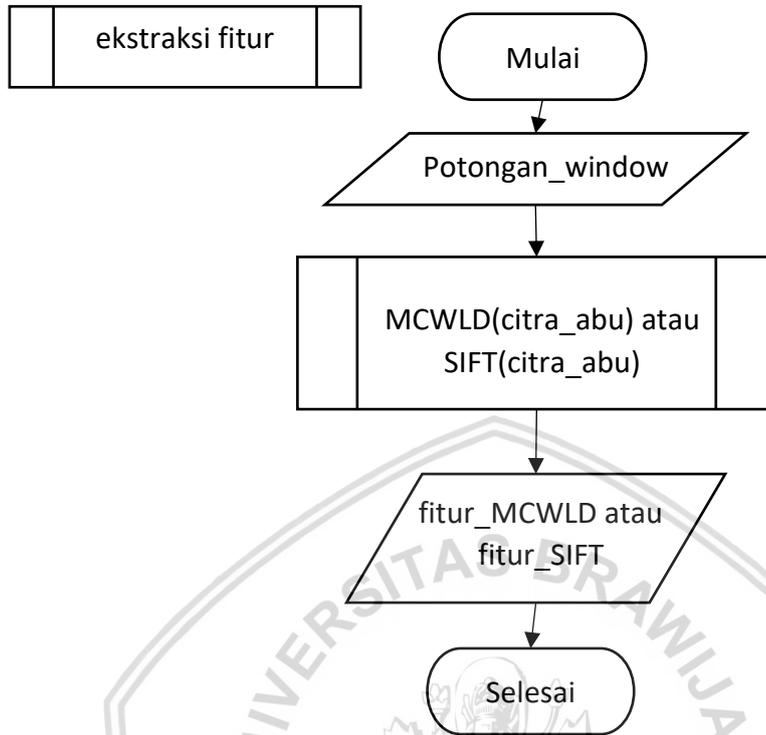
Proses selanjutnya yang dilakukan pada *preprocessing* adalah mengolah citra abu-abu yang dihasilkan dari proses *grayscale* menjadi potongan *window* yang akan diolah oleh proses ekstraksi fitur. Berikut adalah penjelasan dari Gambar 4.4, yakni menjelaskan tentang *sliding window* dengan *overlapping window* yang digunakan penelitian ini sebagai *baseline*:

1. *Input* yang diterima oleh fungsi ini adalah citra wajah baik foto atau sketsa dalam bentuk citra abu-abu
2. Yang perlu diperhatikan saat menggunakan *sliding window* adalah menentukan ukuran *region* dan besaran perpindahan yang dilakukan. Pada penelitian ini, ukuran dari *region* yang digunakan adalah 16 *pixel* dan besaran perpindahan yang dilakukan adalah 9 *pixel*.
3. Untuk setiap indeks baris, diperiksa apakah indeks tersebut jika ditambah dengan ukuran *region* akan melebihi dari ukuran citranya atau tidak. Jika iya, maka operasi dibawahnya tidak akan dikerjakan. Sebaliknya jika tidak nilai indeks baris dijumlahkan dengan besaran *region* masih kurang dari ukuran citra, proses dibawahnya akan dikerjakan
4. Sama seperti baris, pada indeks kolom dilakukan pemeriksaan yang sama yakni memeriksa apakah indeks tersebut jika ditambah dengan ukuran *region* akan melebihi dari ukuran citranya atau tidak. Jika iya, maka operasi dibawahnya tidak akan dikerjakan. Sebaliknya jika tidak nilai indeks kolom dijumlahkan dengan besaran *region* masih kurang dari ukuran citra, proses dibawahnya akan dikerjakan
5. Proses selanjutnya adalah memotong citra dimana titiknya dimulai dari indeks baris dan kolom saat itu hingga indeks baris + ukuran *region* dan indeks kolom + ukuran *region*. Ukuran dari citra yang dipotong tersebut adalah 16 x 16 *pixel*.
6. Citra yang dipotong tadi, akan diproses selanjutnya oleh metode SIFT untuk membentuk deskriptor SIFT dan MCWLD untuk membentuk deskriptor MCWLD.



Gambar 4.4 Diagram Alir Sliding window dengan Overlapping Window

### 4.1.2 Ekstraksi Fitur



Gambar 4.5 Diagram Alir Ekstraksi Fitur

Pada penelitian ini, untuk ekstraksi fitur dilakukan oleh metode SIFT dan MCWLD yang dilakukan secara terpisah. Keduanya menghasilkan nilai fiturnya sendiri yang nanti akan digunakan pada tahap pemadanan. Fitur dari masing-masing metode ini akan digunakan pada proses pemadanan selanjutnya. Fitur dari proses ini tidak disatukan atau disimpan terpisah sehingga tidak menyulitkan saat melakukan proses pemadanan. Pada perancangan ini, perancangan metode SIFT dan MCWLD memiliki proses yang berbeda sehingga subproses pada subbab ini akan dipisah sesuai dengan metode yang digunakan serta menjadi acuan untuk bab implementasi nantinya. Pada subbab berikutnya akan dijelaskan lebih rinci rancangan tahapan untuk metode SIFT maupun MCWLD.

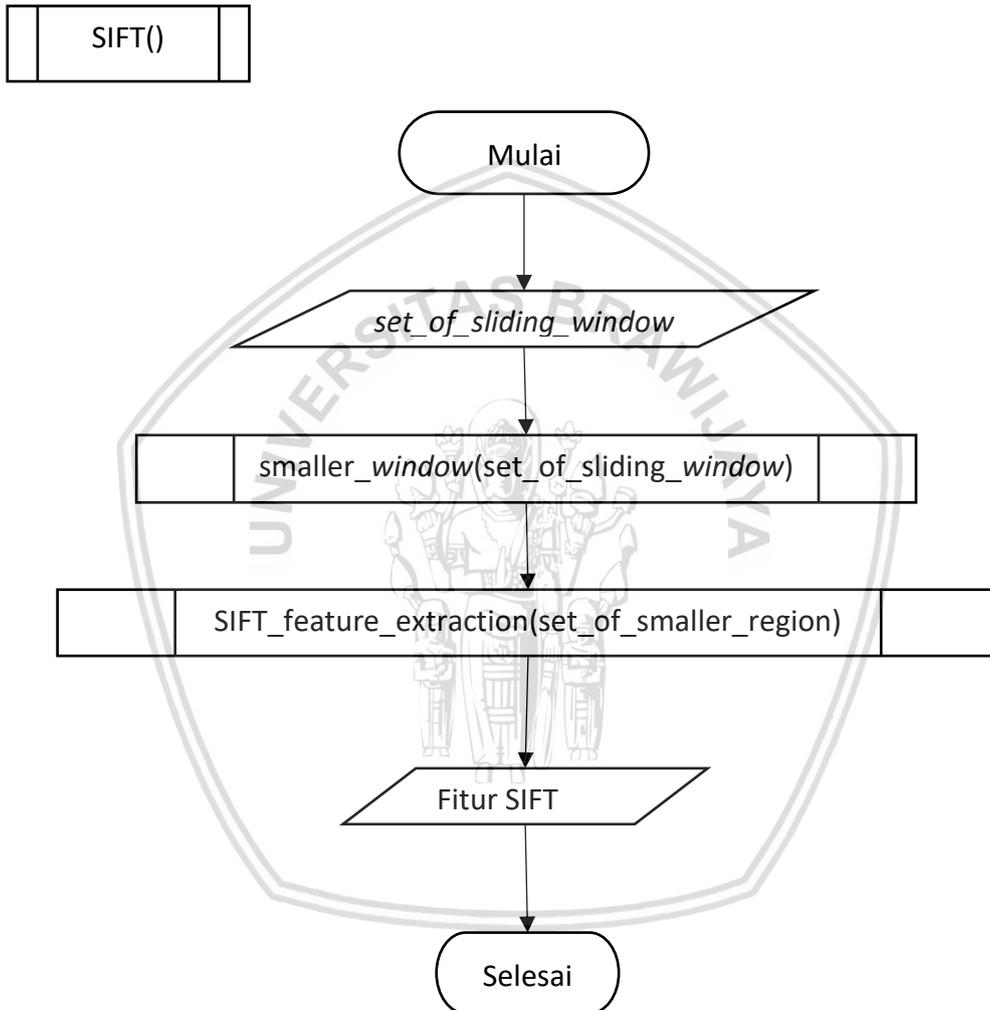
#### 4.1.2.1 Ekstraksi Fitur SIFT (*Scale Invariant Feature Transform*)

Gambar 4.6 menjelaskan alur dari ekstraksi fitur menggunakan metode SIFT. Berikut adalah penjelasan dari tahap-tahapnya:

1. Masukan dari metode SIFT ini adalah citra yang sudah dalam bentuk *grayscale* atau hanya memiliki satu *channel* warna, yakni abu-abu
2. Proses selanjutnya adalah membuat potongan-potongan *window* menggunakan *sliding window* yang *overlapping*.



3. Setelah membuat potongan-potongan, proses selanjutnya akan membuat potongan yang lebih kecil lagi dimana potongan ini akan diekstraksi nilainya.
4. Proses ini akan mengekstraksi dan merepresentasikannya sbegai fitur dari SIFT
5. Fitur yang dihasilkan oleh metode SIFT ini sejumlah *sliding window* x vektor berdimensi-128

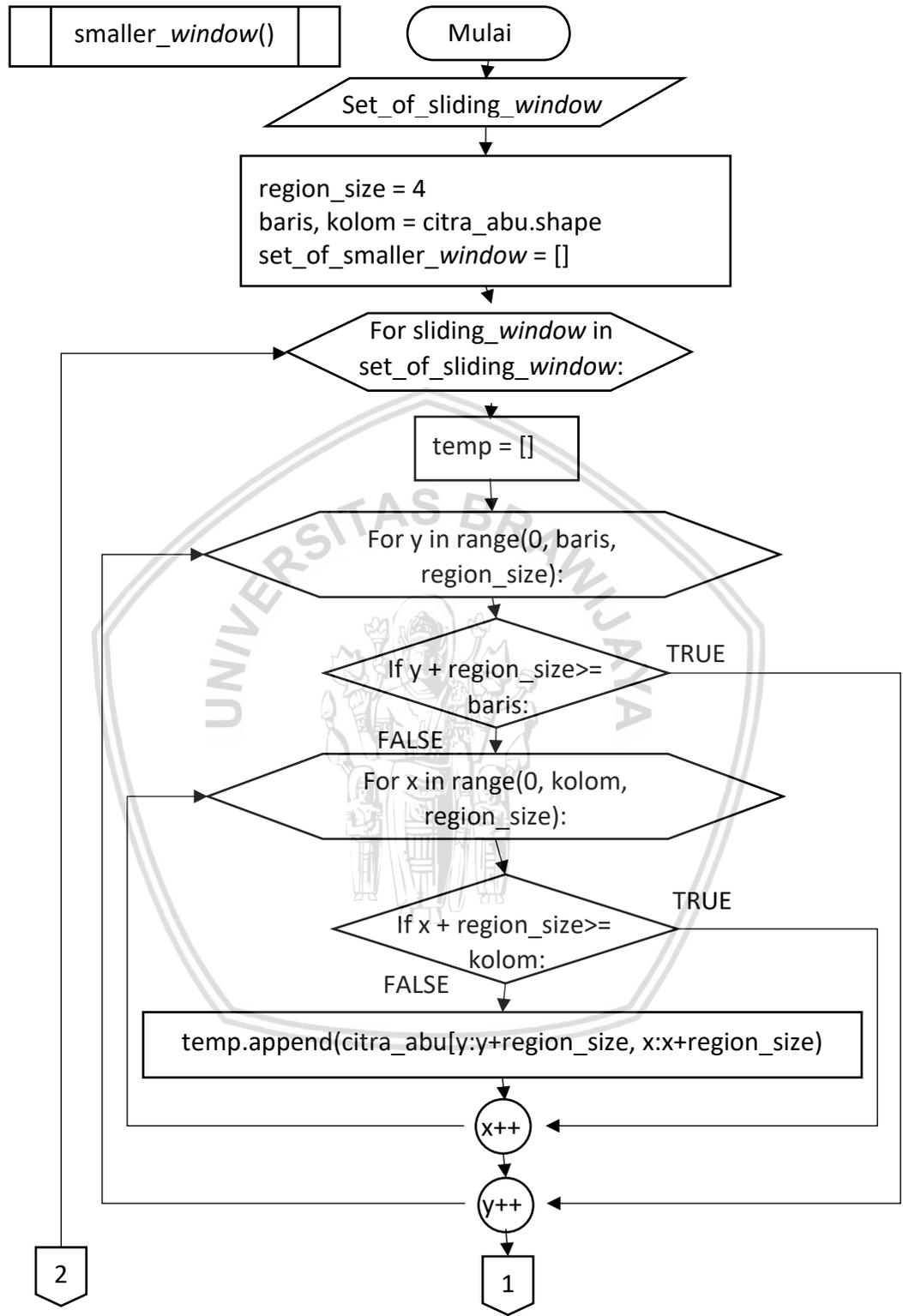


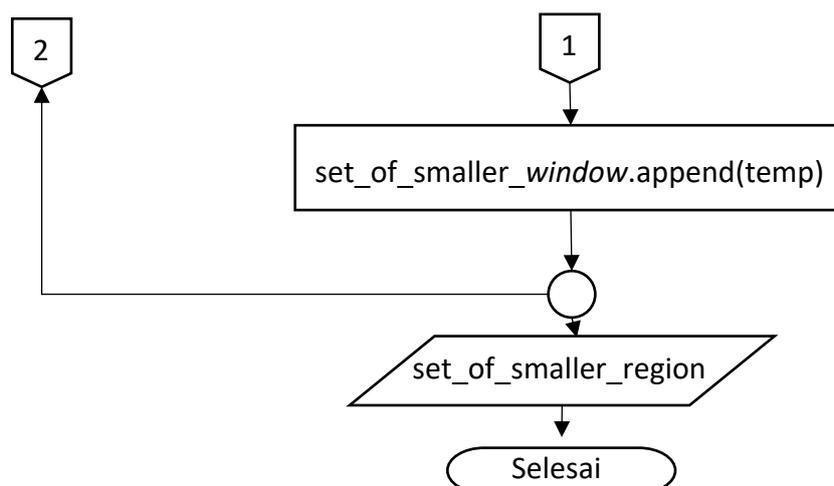
**Gambar 4.6 Diagram Alir Ekstraksi Fitur SIFT**

Pada proses selanjutnya, sub proses dijelaskan oleh masing-masing diagram seperti yang dijelaskan sebelumnya mengikuti rancangan yang telah dijabarkan pada proses sebelumnya. Adapun sub proses dijelaskan dengan diagram alirnya sendiri. Sub proses menjadi dasar dari implementasi pada masing-masing metode.



Membuat potongan kecil pada *window* 16 x 16





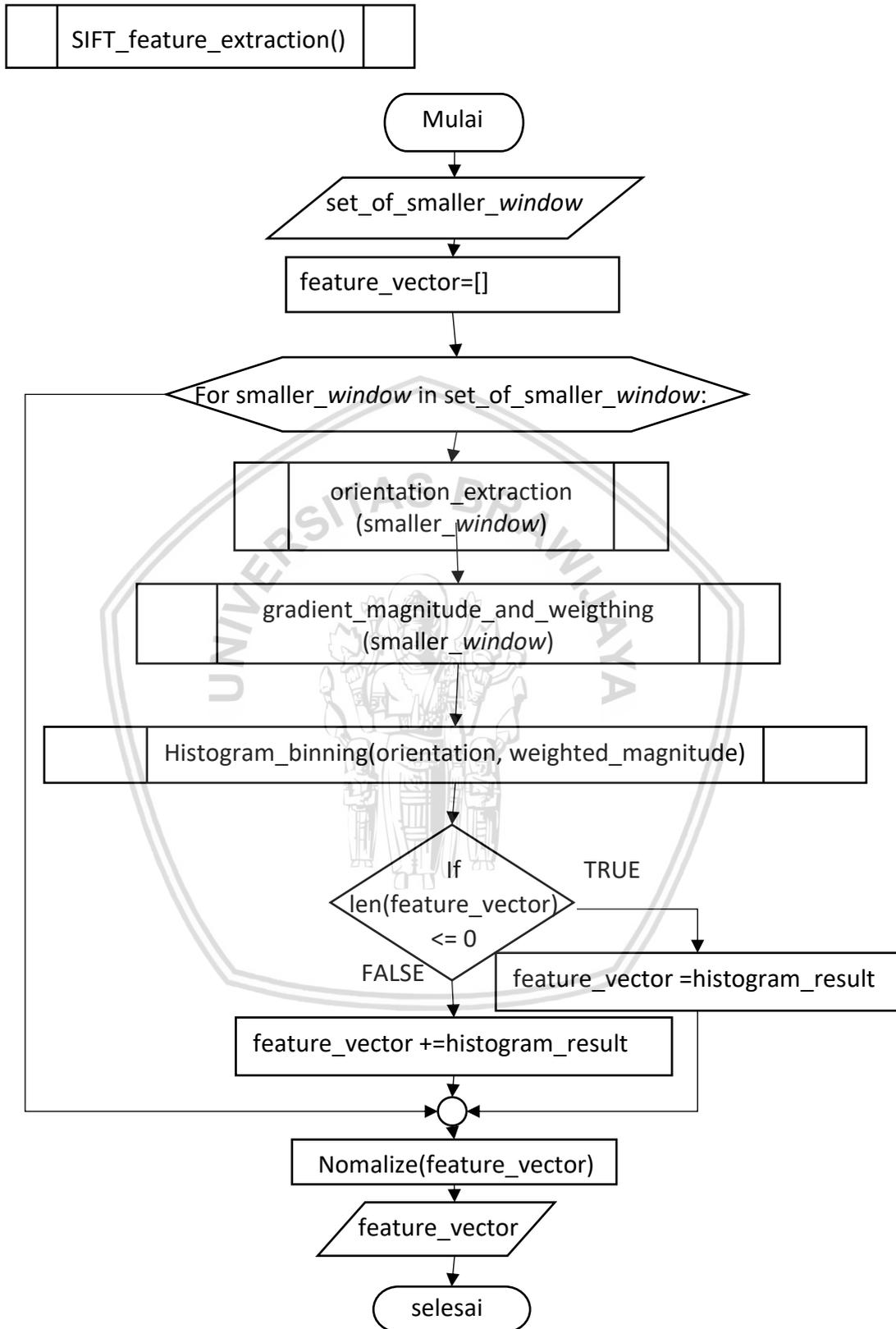
**Gambar 4.7 Diagram alir mengambil *window* kecil**

Berikut adalah penjelasan dari gambar 4.6 yakni proses membagi *window* menjadi 16 *window* kecil:

1. *Input* dari fungsi ini adalah *window* yang sudah diperoleh dari *sliding window*
2. Proses yang dilakukan pada saat membagi region ini sama dengan *sliding window* yang tidak *overlapping*. Yang perlu diperhatikan adalah ukuran region dan besaran perpindahannya memiliki nilai yang sama. Ini yang membuatnya tidak *overlapping*.
3. Untuk setiap indeks baris kolom, diperiksa jika nilai indeks tersebut dijumlahkan dengan ukuran region apakah melebihi dari ukuran citra atau tidak. Jika iya, proses dibawahnya tidak akan dijalankan.
4. Proses ini membagi citra yang dimulai dari indeks baris dan kolom pada saat itu hingga indeks baris + ukuran region dan indeks kolom + ukuran region. Citra yang dipotong tersebut memiliki ukuran 4 x 4 *pixel*. Hasil perpotongan ini disimpan pada sebuah variabel.
5. Keluaran dari proses ini adalah potongan-potongan region yang selanjutnya akan digunakan untuk mengekstraksi nilai orientasi dan besaran gradien.

Hasil akhir dari proses ini diberikan dalam bentuk larik yang setiap elemen pada larik tersebut merupakan potongan kecil dari *window* yang diberikan. Proses ini dilakukan untuk mempermudah pada proses selanjutnya sehingga tidak perlu melakukan proses pembagian *window* karena sudah dipetakan untuk setiap masing-masing *local region* pada larik.

Ekstraksi fitur SIFT



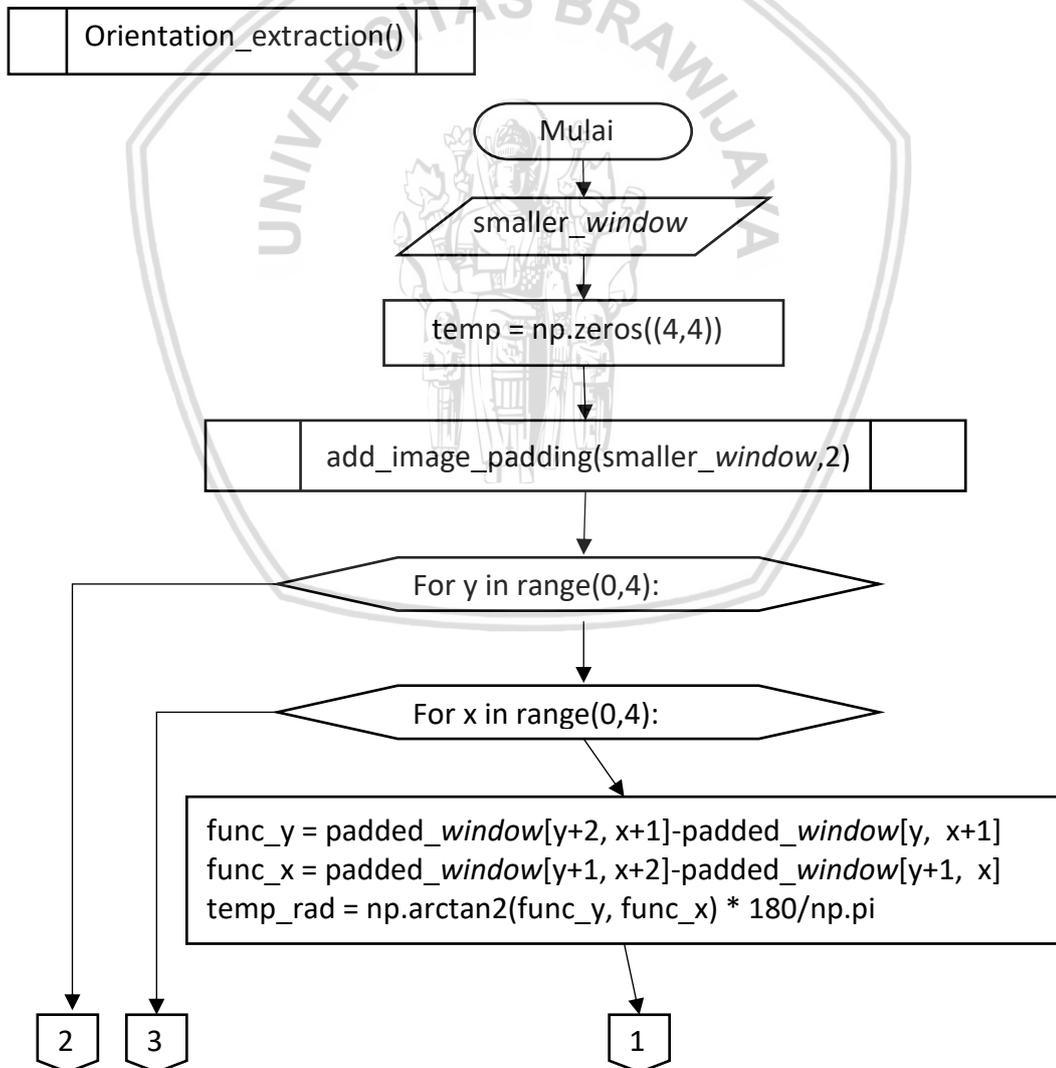
Gambar 4.8 Diagram Alir Ekstraksi Fitur

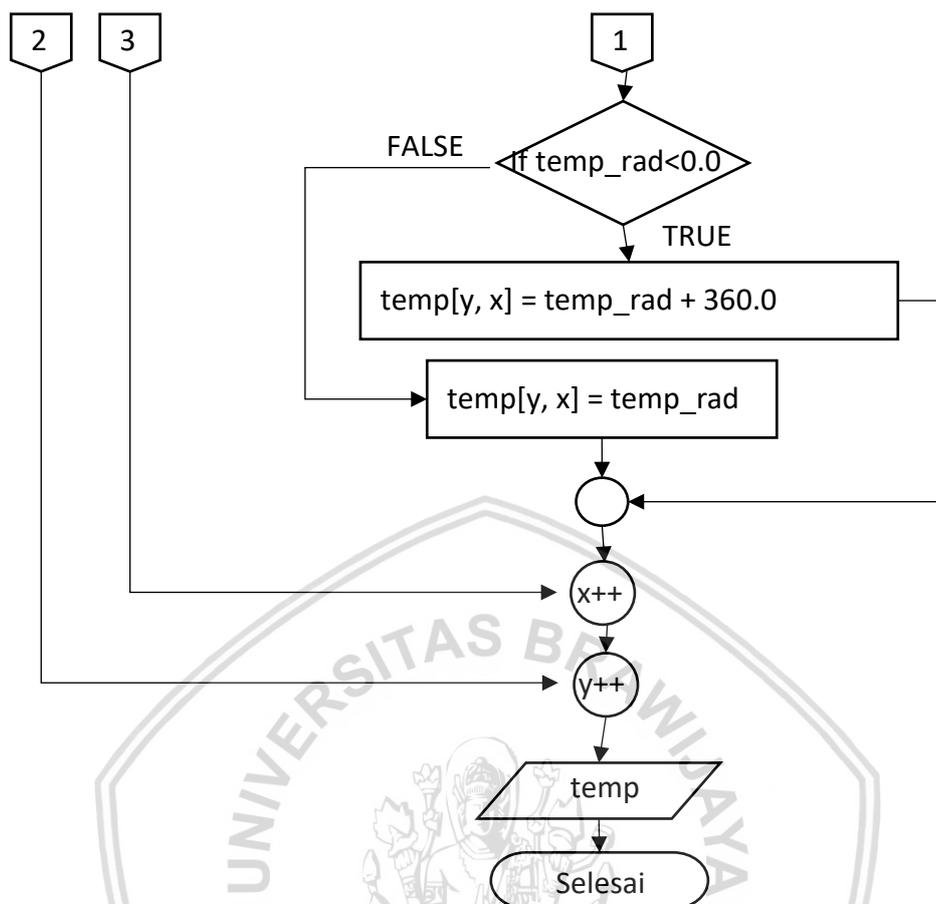


Berikut adalah penjelasan dari diagram alir pada gambar 4.8 yang menjelaskan proses ekstraksi fitur oleh SIFT pada masing-masing *window*. Adapun proses SIFT ini hanya pada bagian untuk memproses pembentukan deskriptor:

1. *Input* yang diterima dari fungsi ini adalah *window* yang sebelumnya didapat pada proses *sliding window*.
2. Proses selanjutnya adalah membagi *window* berukuran 16 x 16 *pixel* tersebut menjadi 16 *window* berukuran 4 x 4 *pixel*.
3. Untuk setiap region, dilakukan ekstraksi nilai fitur (besaran gradien dan orientasi) SIFT. Nilai fitur pada masing-masing region tersebut disimpan pada sebuah variabel.
4. Nilai fitur yang didapat dari masing-masing region digabung menjadi 1 vektor dan dinormalisasi menjadi unit vektor.
5. Hasil akhirnya berupa vektor berdimensi 128

Ekstraksi nilai orientasi



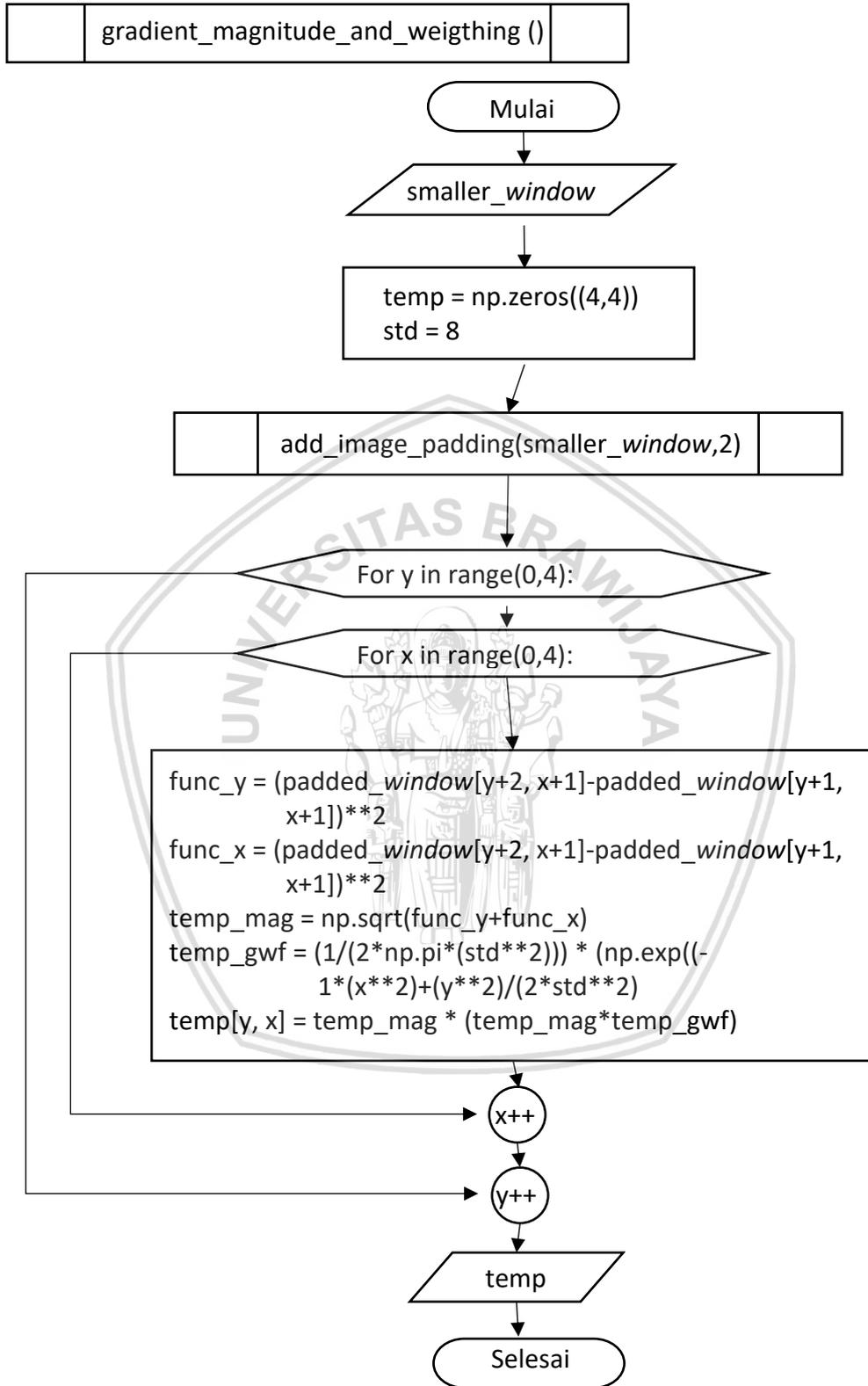


**Gambar 4.9 Diagram alir ekstraksi nilai orientasi**

Diagram 4.9 menerangkan proses yang terjadi pada masing-masing region. Adapun langkah-langkahnya sebagai berikut:

1. *Input* dari fungsi ini adalah kumpulan *window* dari operasi sebelumnya.
2. Dalam fungsi ini, memanggil sebuah fungsi untuk menambahkan *padding* pada *window* kecil dengan ukuran *padding* sebesar 2 *pixel*.
3. Untuk setiap indeks baris dan kolom pada region, dilakukan ekstraksi orientasi dengan cara membentuk dahulu nilai *x* dan nilai *y*. Nilai fungsi *x* didapat dari pengurangan *pixel* pada titik *x* sebelum *x* saat itu dengan setelahnya. Nilai fungsi *y* berasal dari pengurangan *pixel* pada titik *y* sebelum *y* saat itu dengan titik *y* setelahnya. Nilai fungsi *x* dan *y* akan membentuk nilai radian dengan fungsi *arctan2* yang diubah menjad nilai derajat. Nilai penambahan 2 berasal dari ukuran *padding*.
4. Hasil dari fungsi adalah sebuah matriks dengan masing-masing selnya adalah nilai orientasi dalam derajat yang menentukan arah dari *pixel*nya.

Ekstraksi nilai besaran gradien dan pembobotan



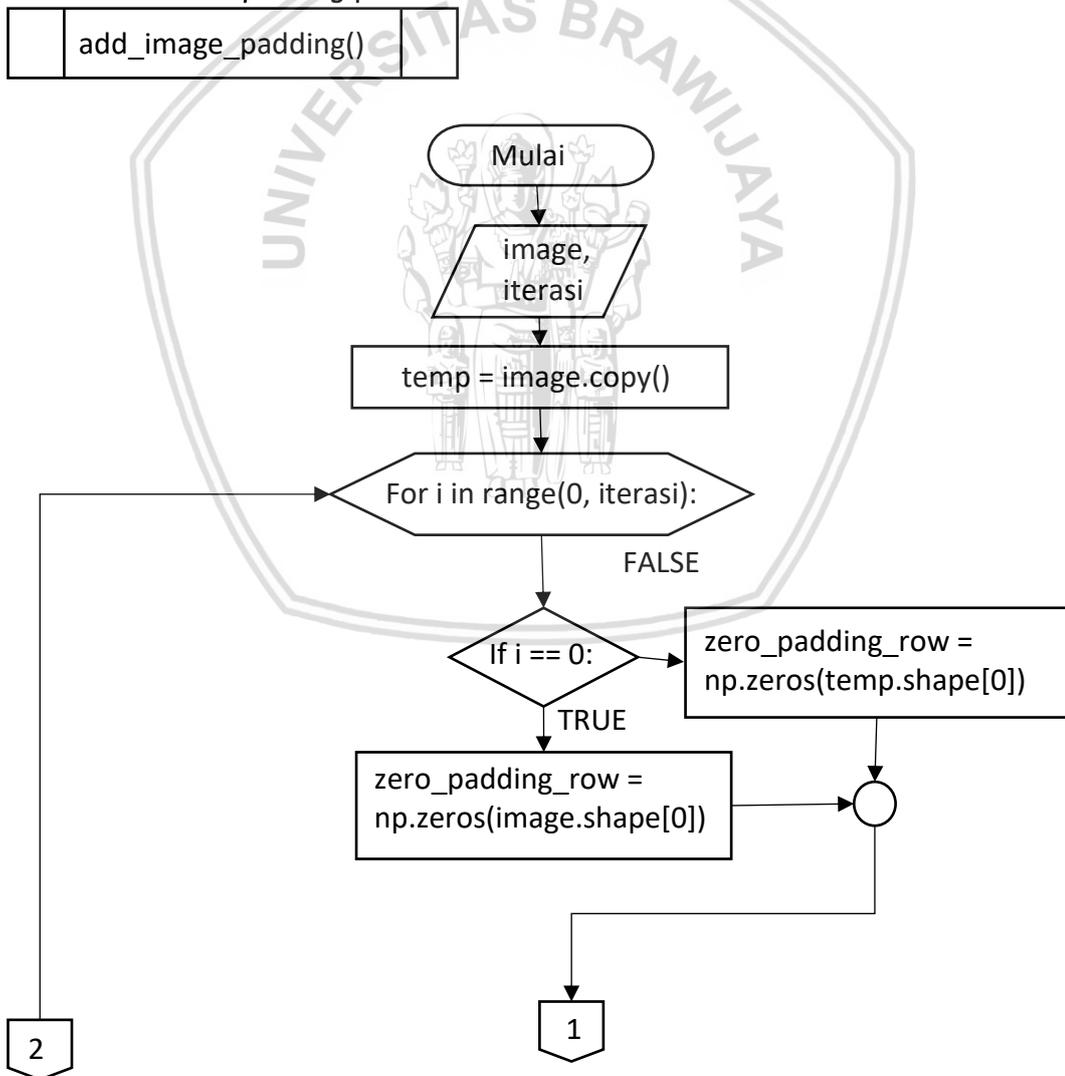
Gambar 4.10 Diagram alir ekstraksi nilai besaran gradien dan pembobotan

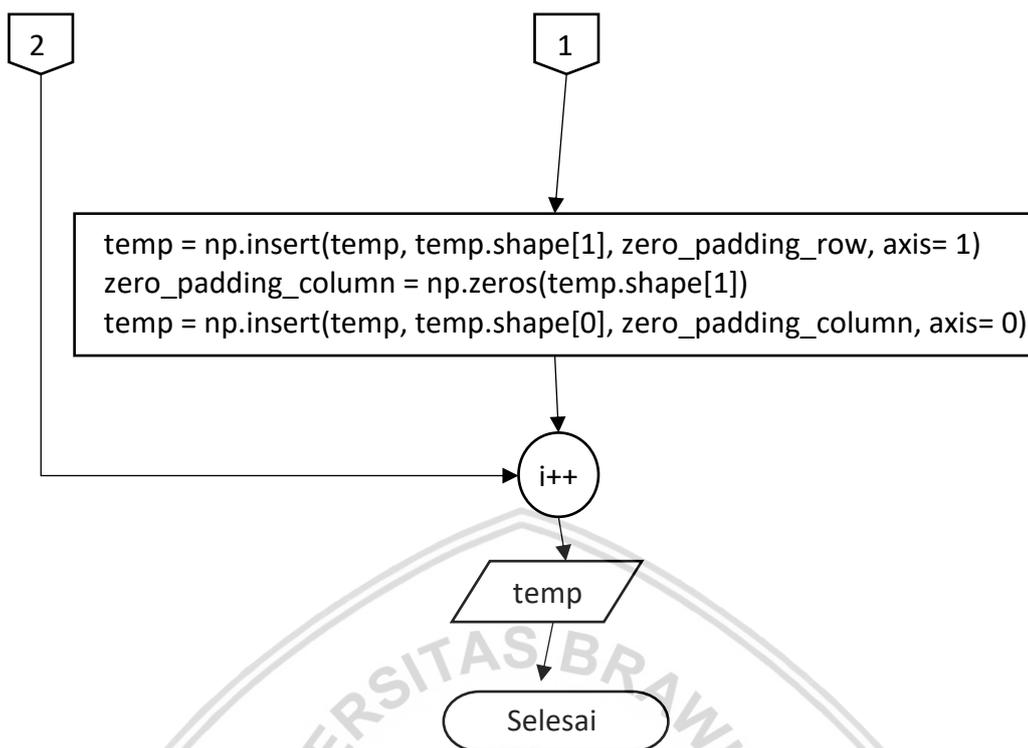


Pada gambar 4.10, diagram alir tersebut menjelaskan tentang bagaimana mengambil nilai besaran gradien dan memberikan nilai bobot terhadap besaran tersebut. berikut adalah langkah-langkahnya:

1. Untuk *window* kecil dilakukan padding sebesar 2 *pixel*
2. Untuk setiap baris dan kolom, operasi yang dilakukan ini sama seperti pada operasi pembentukan fungsi pada ekstraksi orientasi namun dikuadratkan
3. Fungsi x dan y dijumlahkan kemudian diakarkan sehingga didapatkan nilai besaran gradien. Untuk pembobotan, nilai dari bobot didapat dari fungsi *gaussian weighting function*. Nilai deviasi standar yang digunakan adalah setengah dari ukuran *window*, yakni 8. Nilai besaran gradien kemudian dijumlahkan dengan nilai besaran gradien yang dikalikan dengan nilai fungsi *gaussian weighting function*.
4. Hasil dari fungsi adalah sebuah matriks dengan masing-masing selnya adalah nilai besaran gradien yang telah diberi bobot.

Proses melakukan *padding* pada citra





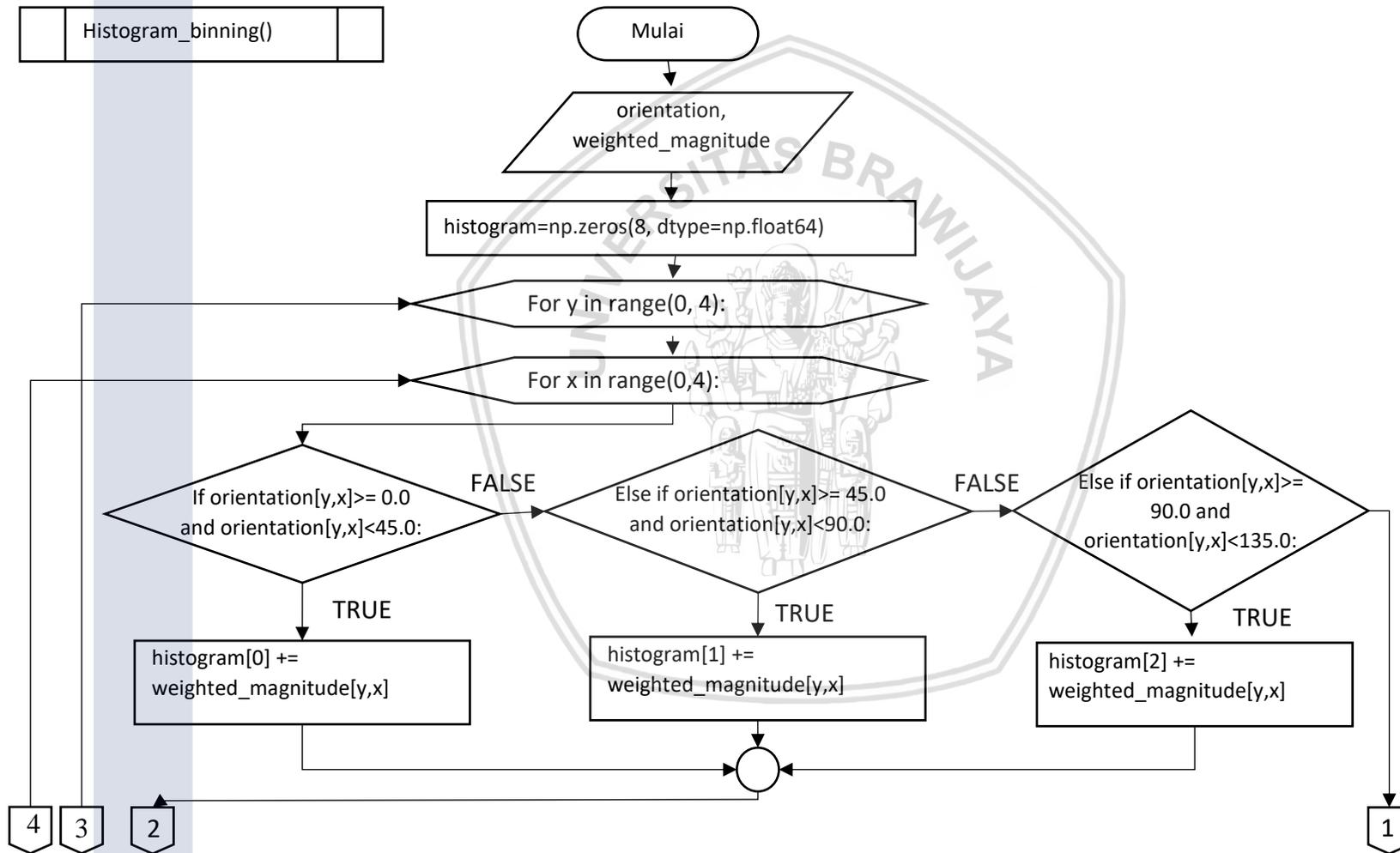
**Gambar 4.11 Diagram Alir menambah Padding**

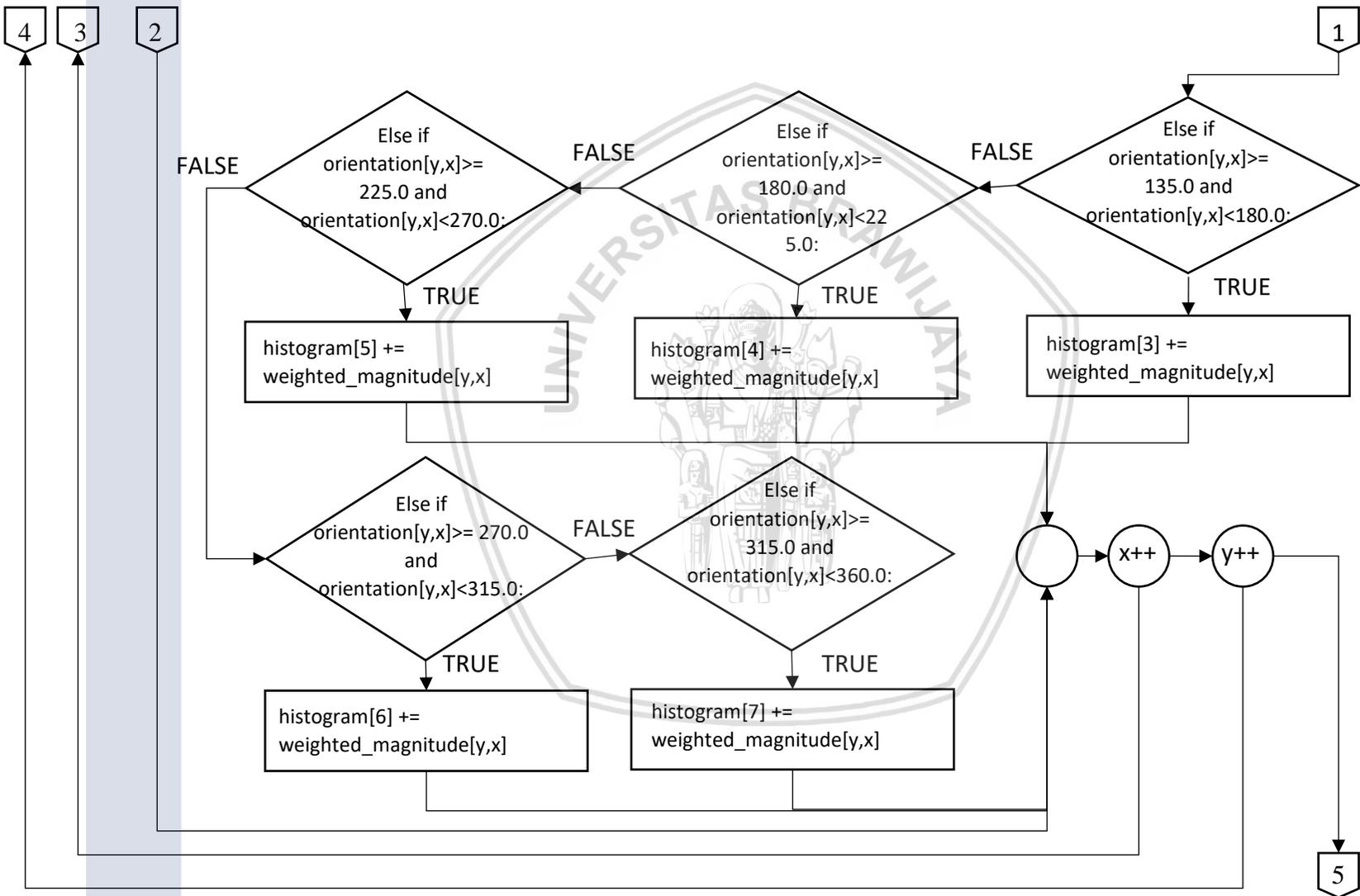
Gambar 4.11 menjelaskan tentang alur memberi *padding* pada citra. Fungsi *padding* ini memberi baris dan kolom tambahan bernilai 0 pada awal dan akhir dari citra. Adapun langkah-langkahnya adalah sebagai berikut:

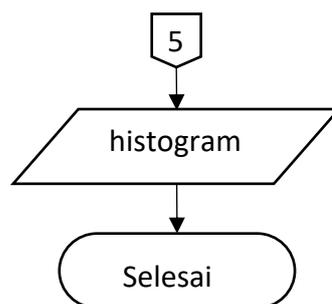
1. *Input* dari fungsi ini adalah citra dan jumlah iterasi yang dilakukan.
2. Yang pertama dilakukan adalah membuat sebuah variabel *temporary* yang akan menjadi citra yang diberi *padding*.
3. Untuk setiap perulangan sebanyak nilai iterasi, membuat sebuah array bernilai 0 dengan ukuran sama dengan nilai sumbu y pada citra, kemudian ditambahkan pada variabel *temporary* di sisi paling bawah dan atas. Pada perulangan pertama, nilai yang digunakan adalah lebar sumbu y pada citra asli, namun pada iterasi berikutnya menggunakan citra dari variabel *temporary*.
4. Begitu juga pada sumbu x, dilakukan hal yang sama untuk membentuk kolom baru di sisi paling kanan dan kiri. Berbeda dengan sumbu y, pada sumbu x tidak ada pengecekan karena nilai jumlah baris atau sumbu y tidak mempengaruhi hasil pada penambahan pada sumbu x.
5. Hasil dari operasi ini adalah citra yang telah diberi *padding*.

Proses melakukan *histogram binning*

Histogram\_binning()







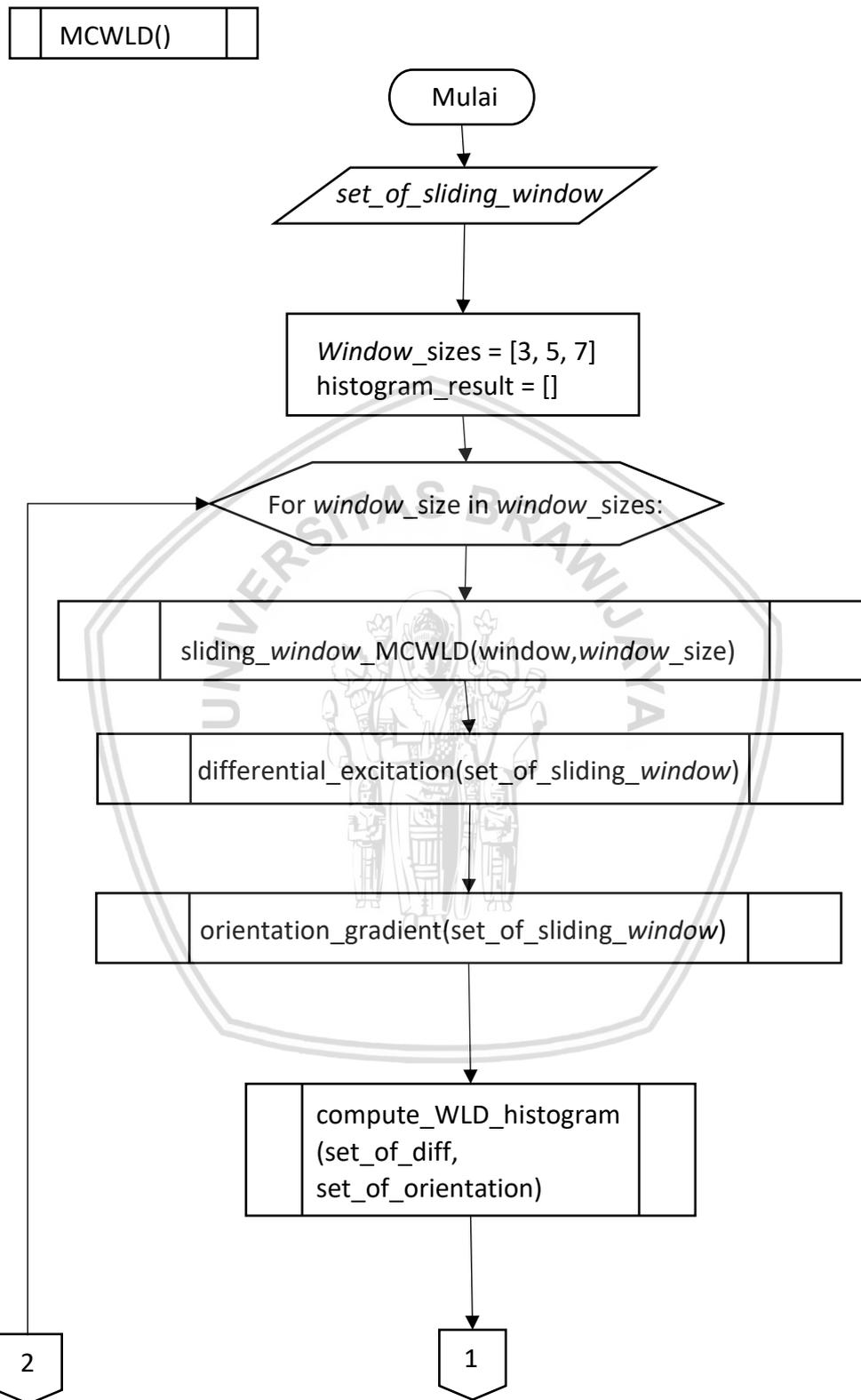
**Gambar 4.12 Diagram alir *Histogram Binning***

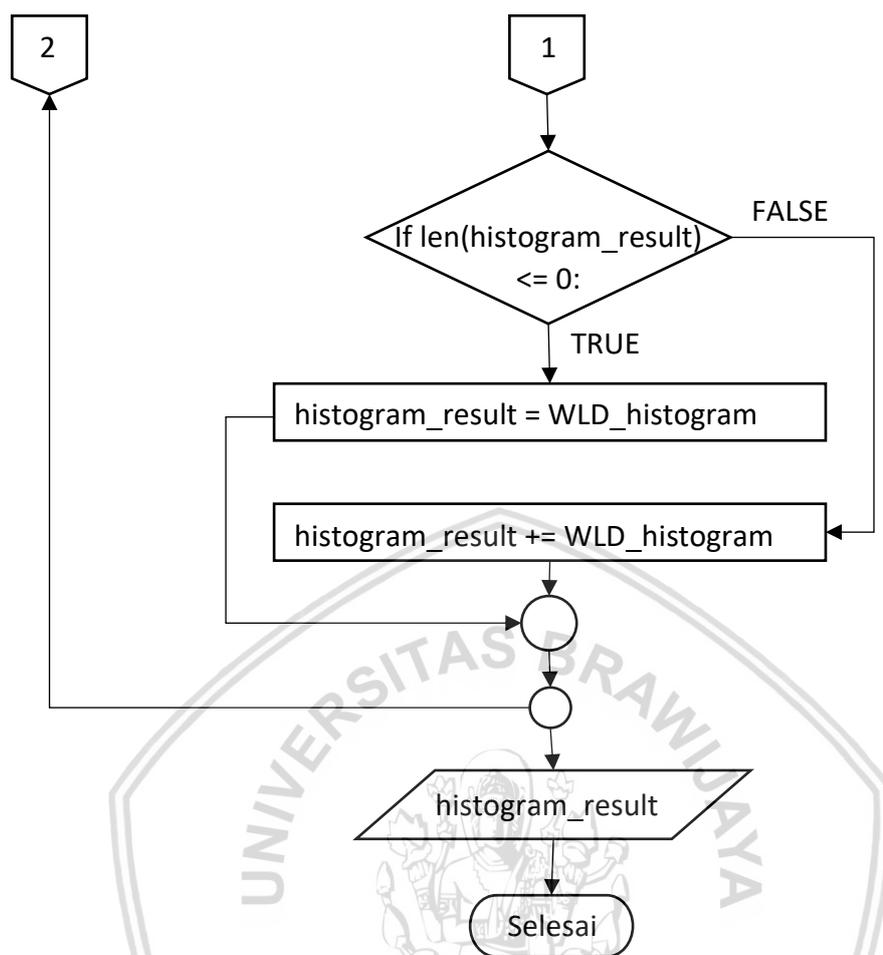
Gambar 4.12 menjelaskan proses pembobotan dan pembentukan histogram 8-bin. Adapun langkah-langkahnya sebagai berikut:

1. *Input* dari proses adalah matriks yang berisi nilai besaran gradien yang telah diberi bobot dan matriks yang berisi nilai orientasinya.
2. Fungsi ini melakukan inisialisasi variabel histogram yang akan menyimpan nilai pada operasi selanjutnya. Nilai awal masing-masing binnya berisi nilai 0.
3. Selanjutnya untuk matriks orientasi dan matriks besaran gradien yang telah diberi bobot pada masing-masing indeks baris dan kolomnya dicek pada bin berapa nilai orientasi tersebut berada. Semisal nilai orientasi tersebut adalah  $49^\circ$  maka nilai besaran gradien yang telah diberi bobot tersebut akan berada pada bin kedua. Setiap nilai akan ditambahkan terus ke masing-masing bin.
4. Hasil akhirnya berupa histogram 8-bin yang akan diproses di tahap terakhir.

Proses pada SIFT diakhiri dengan menyambung seluruh histogram yang jumlah 16 histogram dengan masing-masing bin yang jumlah 8. Sehingga nilai akhirnya berupa vektor berdimensi 128. Nilai vektor ini kemudian dinormalisasi menjadi unit vektor. Proses ini dilakukan pada masing-masing *window* sehingga menghasilkan  $128 \times M \times N$ . Hasil akhir ini kemudian disimpan dengan fitur dan label yang saling berkorespondensi pada suatu basis data telah disediakan oleh bahasa pemrograman Python sehingga tidak perlu melakukan proses ekstraksi fitur pada saat pengujian. Penyimpanan pada basis data ini mempermudah dalam pengerjaan dan tidak memakan waktu saat melakukan pengujian. Baik fitur dari foto dan sketsa disimpan dengan diurutkan sesuai dengan identitas dari data yang tersedia. Dengan melakukan pengurutan seperti ini, proses pemadanan akan lebih mudah dilakukan pada proses evaluasi.

### 4.1.2.2 Ekstraksi Fitur MCWLD (Multiscale Circular Weber Local Descriptor)





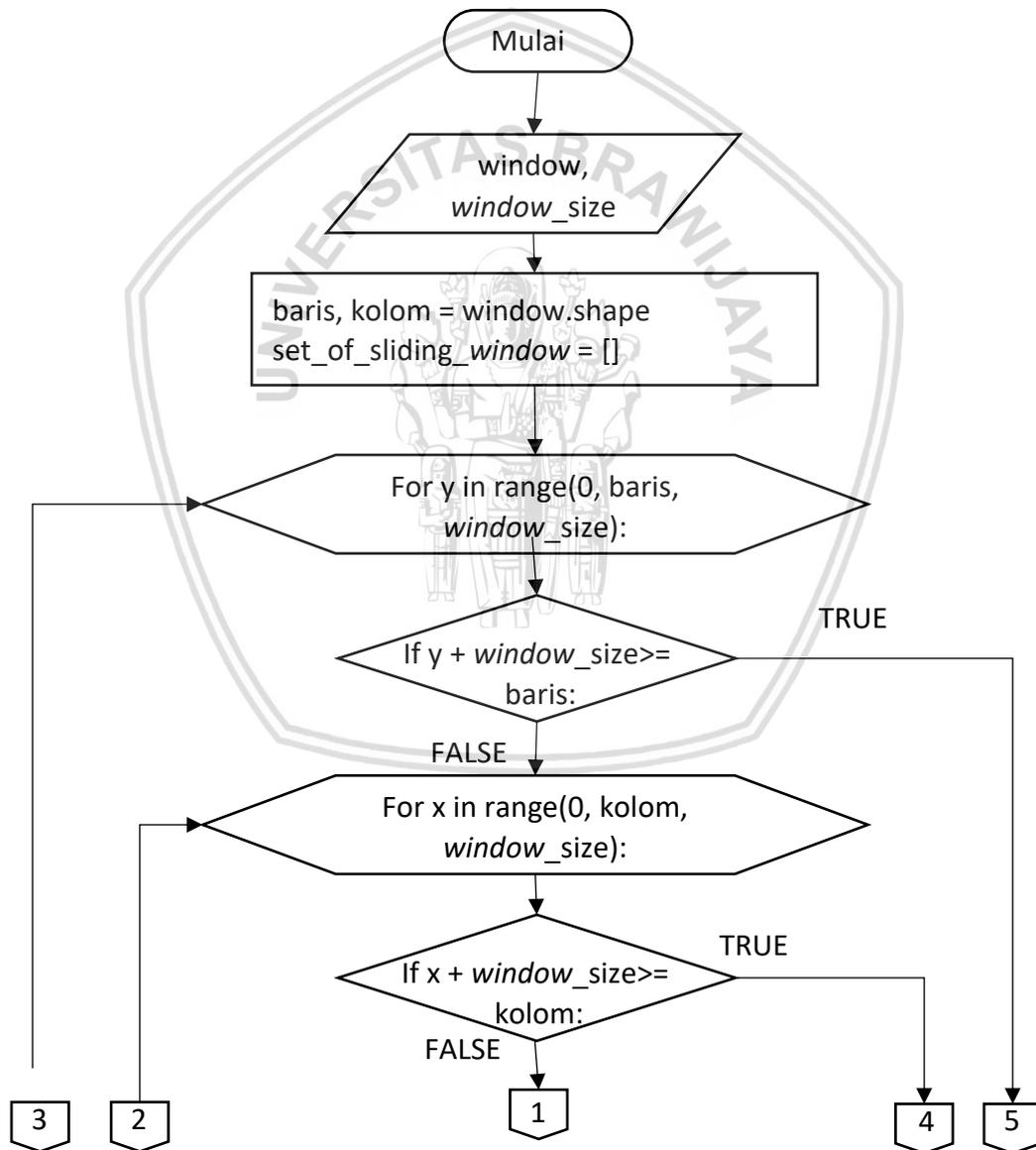
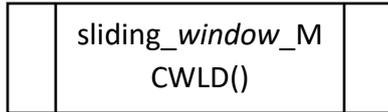
Gambar 4.13 Diagram Alir WLD

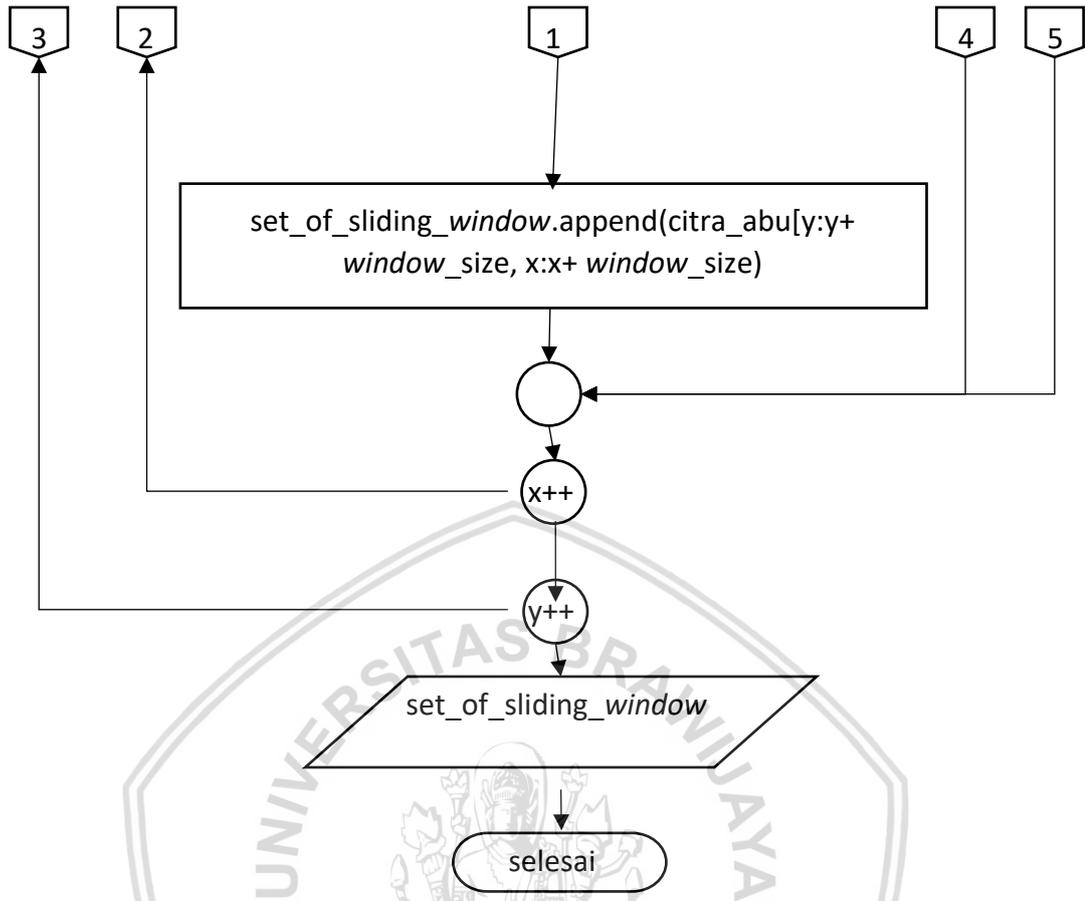
Metode MCWLD dirancang sebagaimana pada diagram alir pada gambar 4.13. adapun penjelasannya adalah sebagai berikut:

1. Input dari proses ini adalah potongan *window* dari *preprocessing*.
2. Menentukan ukuran *window* atau region sesuai dengan jarak radiusnya dari titik tengah. Radius sama dengan 1 memiliki ukuran *window* sama dengan 3, radius sama dengan 2 memiliki ukuran *window* sama dengan 5 dan radius sama dengan 3 memiliki ukuran *window* sama dengan 7.
3. Tahap selanjutnya ialah membagi citra tersebut menjadi N *window*. Hasil dari proses ini menghasilkan 1 kumpulan *window* dengan ukuran *window* 3, 1 kumpulan *window* dengan ukuran *window* 5 dan 1 kumpulan *window* dengan ukuran *window* 7.
4. Setelah membagi *window*, pada masing-masing *window* tersebut diekstraksi nilai *differential excitation*nya. Nilai ini disimpan dalam sebuah array
5. Selain *differential excitation*, nilai yang ekstraksi adalah orientasi dari *window* tersebut. Hasil dari proses ini kemudian disimpan dalam sebuah array

6. Proses selanjutnya adalah membentuk histogram 2D hingga menghasilkan histogram 1 dimensi dari masing-masing ukuran *window*. Hasil histogram dari masing-masing ukuran *window* yang kemudian digabung menjadi 1 histogram sehingga bersifat multiscale.
7. Hasil dari proses MCWLD ini merupakan vektor berdimensi 432.

Membagi *window* sesuai ukuran radius



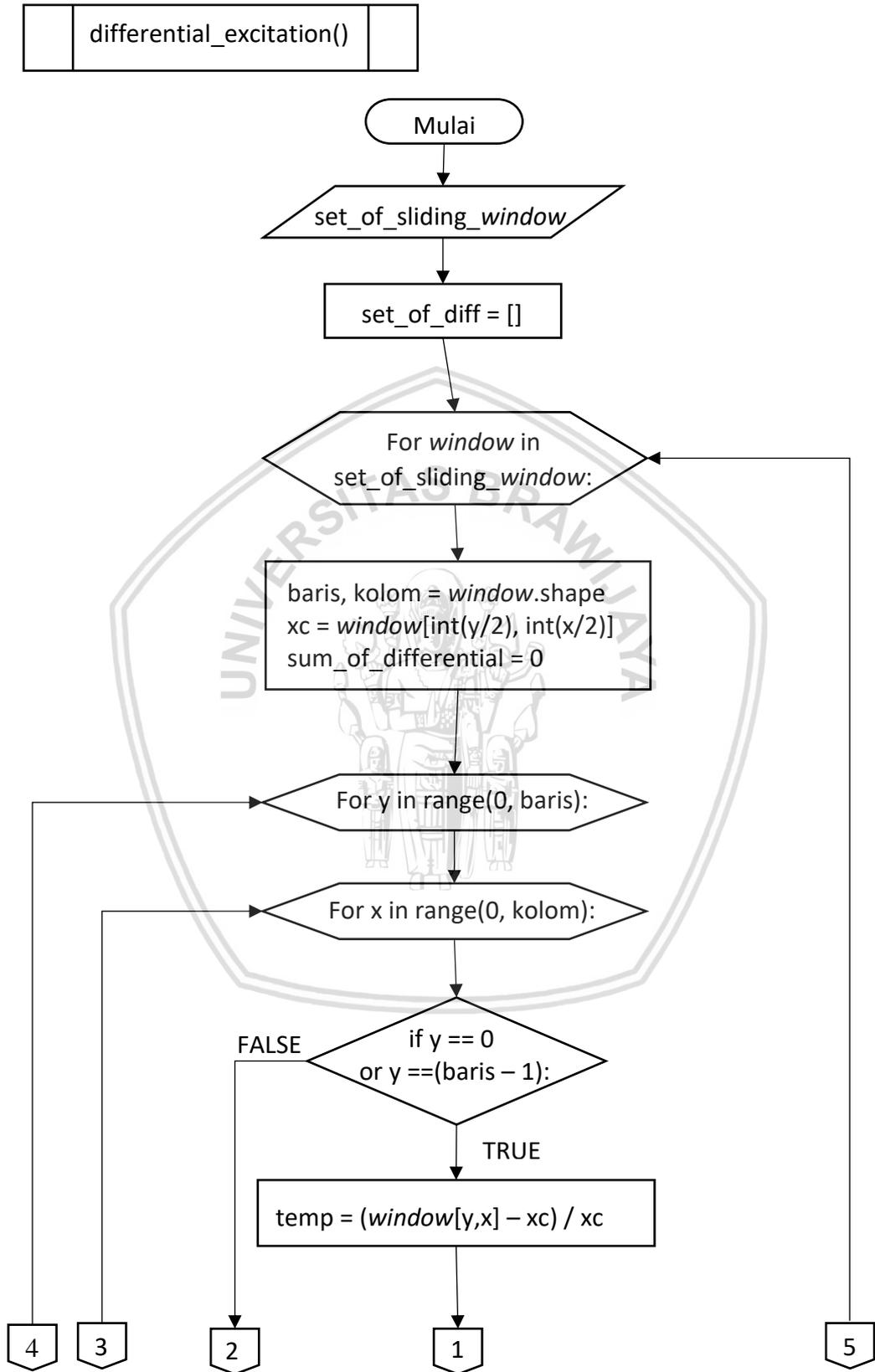


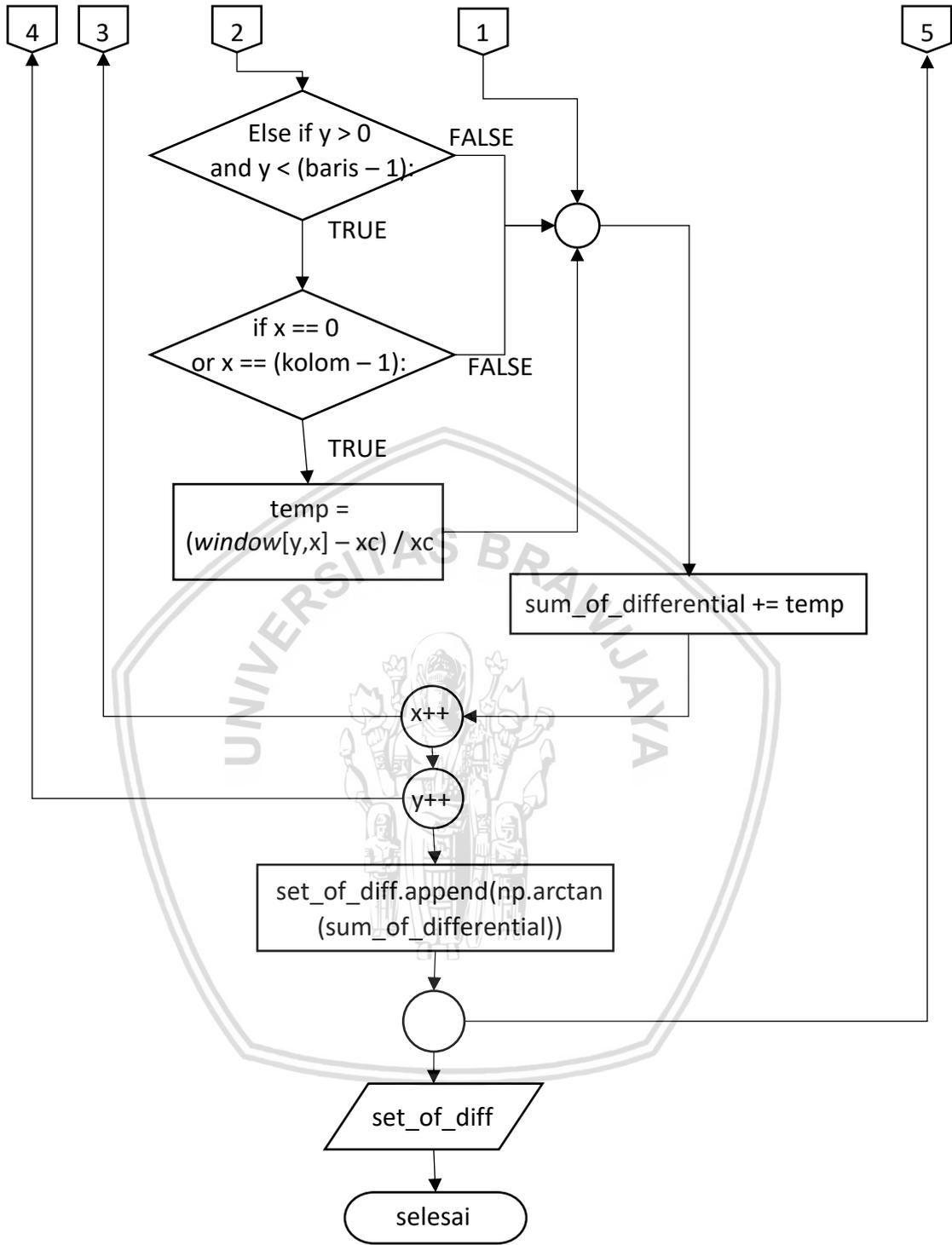
**Gambar 4.14 Diagram alir membagi *window***

Gambar 4.14 menjelaskan alur untuk membagi *window*. *window* yang dihasilkan memiliki ukuran dari satu nilai *window\_size*. Adapun tahapnya adalah sebagai berikut:

1. *Input* dari proses ini adalah sebuah citra dan ukuran *window*nya. Yang diproses pada fungsi ini hanya satu nilai ukuran *window*.
2. Melakukan inisialisasi sebuah array yang akan menyimpan matriks *window* nantinya.
3. Untuk setiap indeks baris dan kolom, proses yang dilakukan adalah membagi citra dari indeks baris dan kolom saat itu hingga indeks baris ditambah ukuran *window* dan indeks kolom ditambah ukuran *window*. Operasi ini sama dengan diagram pada Gambar 4.6 hanya saja tidak melakukan proses *overlapping window*.
4. Hasil citra yang dipotong tersebut disimpan pada variabel *set\_of\_sliding\_window*.
5. Hasil dari fungsi ini akan diteruskan ke fungsi selanjutnya.

### Differential excitation





Gambar 4.15 Diagram alir *differential excitation*

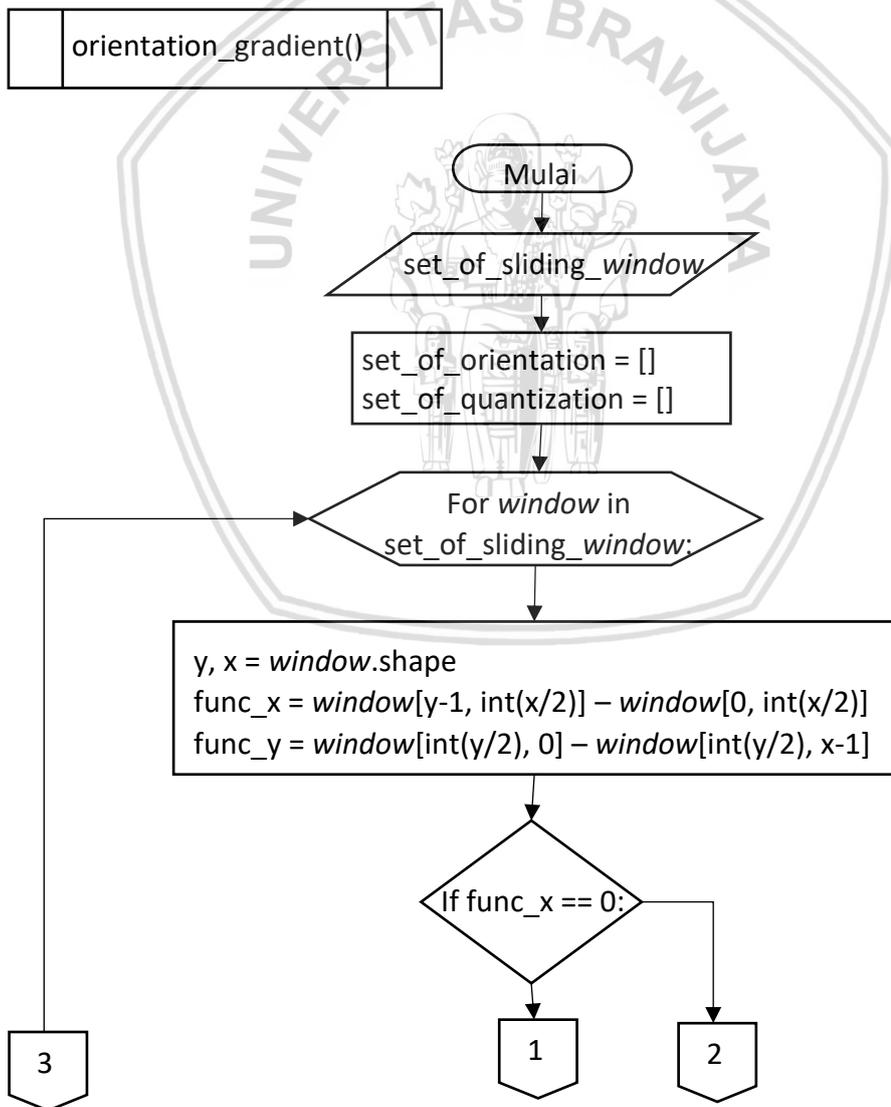
Gambar 4.15 menggambarkan alur proses *differential excitation*. Berikut adalah penjelasan dari setiap tahapnya:

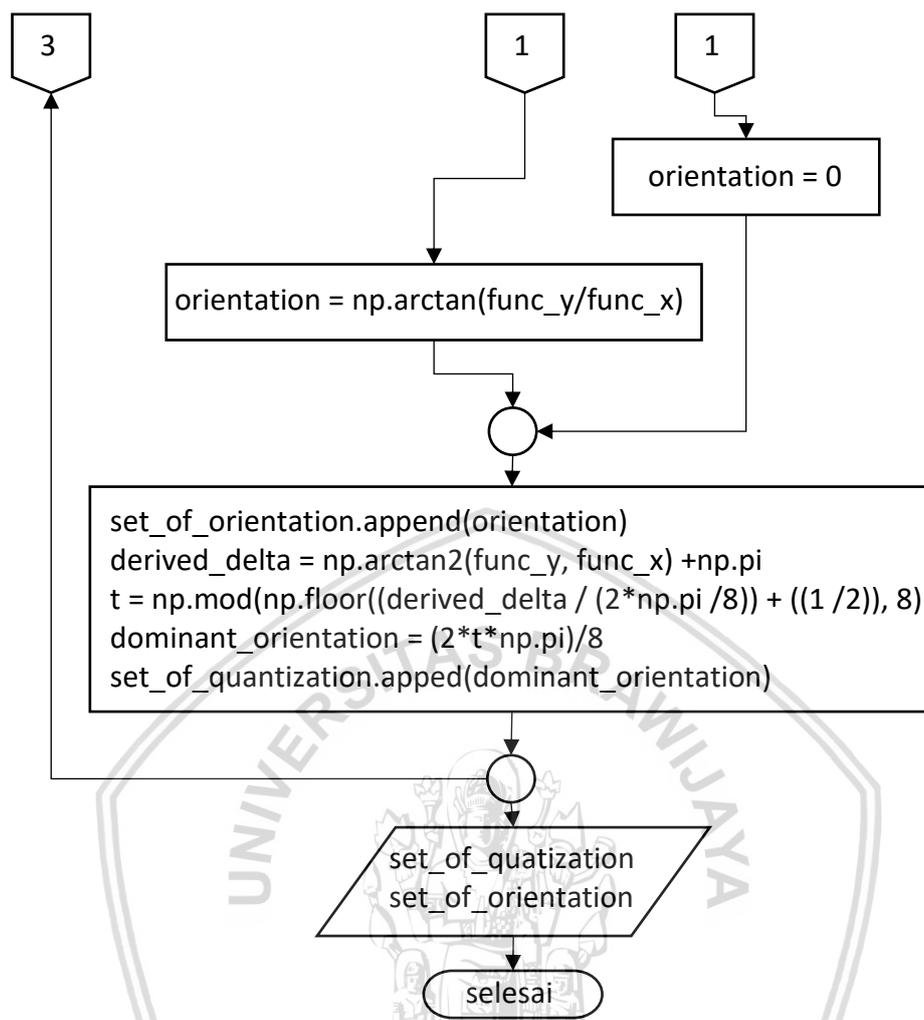
1. *Input* dari fungsi ini adalah sebuah *window* dari kumpulan *window*. Operasi diawal akan mengambil ukuran tinggi dan lebar dari *window*.



2. Untuk setiap indeks baris dan kolom, jika nilai indeks baris sama dengan 0, maka operasi pengurangan antara *pixel* pada baris ke 0 dengan *pixel* yang di tengah dilakukan. Jika nilai indeks baris lebih besar dari 0 dan bukan merupakan baris terakhir, maka pengurangan hanya dilakukan pada *pixel* di kolom pertama dan kolom terakhir. Jika indeks baris merupakan baris terakhir, operasi pengurangan dilakukan seperti pada baris pertama.
3. Hasil dari perhitungan ini dijumlahkan dengan nilai-nilai sebelumnya. Hasil dari setiap perulangan adalah jumlah nilai hasil pengurangan dengan titik tengahnya kemudian dilakukan operasi arctan yang merubahnya menjadi nilai radian.
4. Hasil dari fungsi ini disimpan menjadi satu kumpulan nilai *differential excitation* dan digunakan saat pembentukan histogram 2D bersamaan dengan nilai orientasi.

**Gradient Orientation**





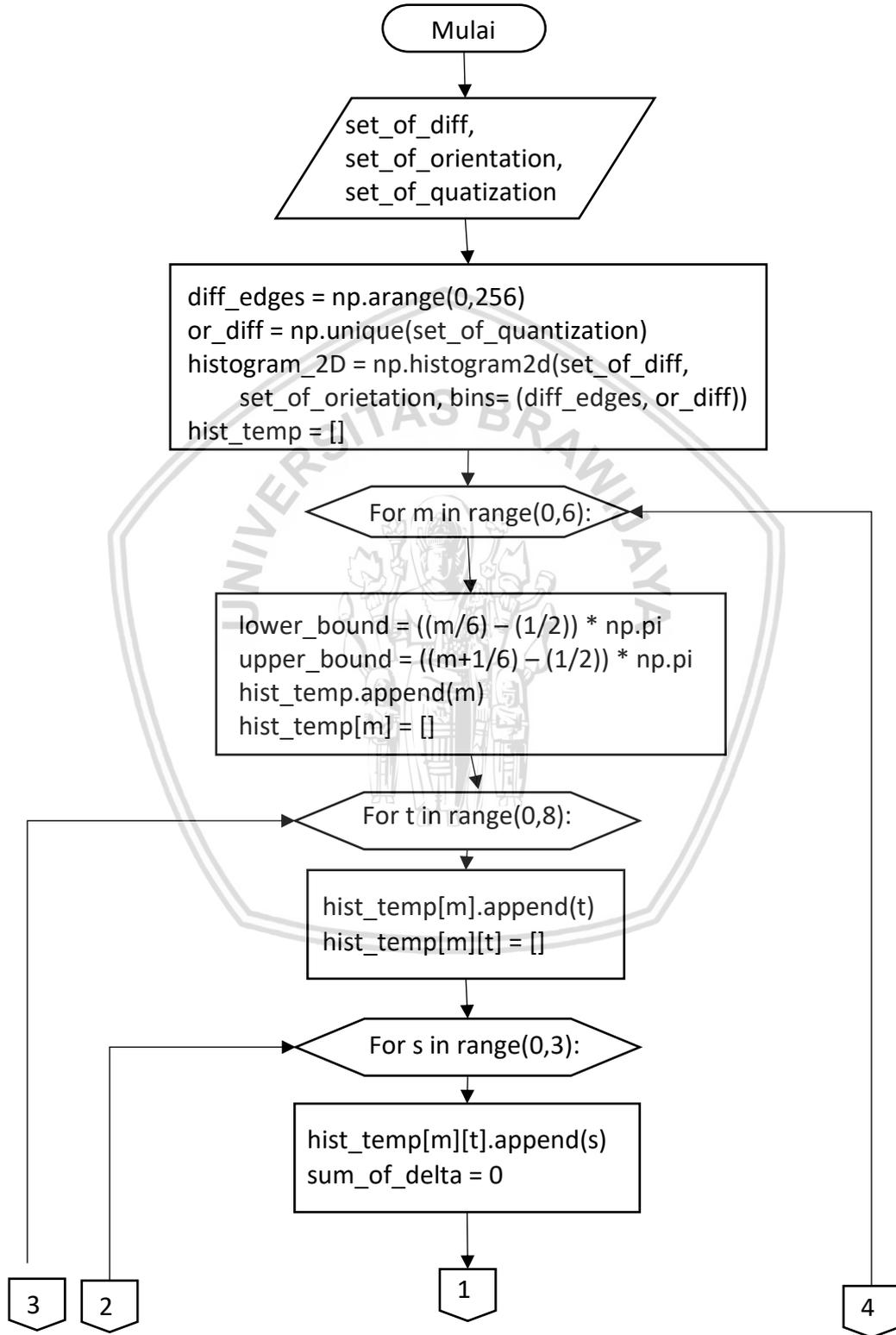
**Gambar 4.16 Diagram Alir Gradient Orientation**

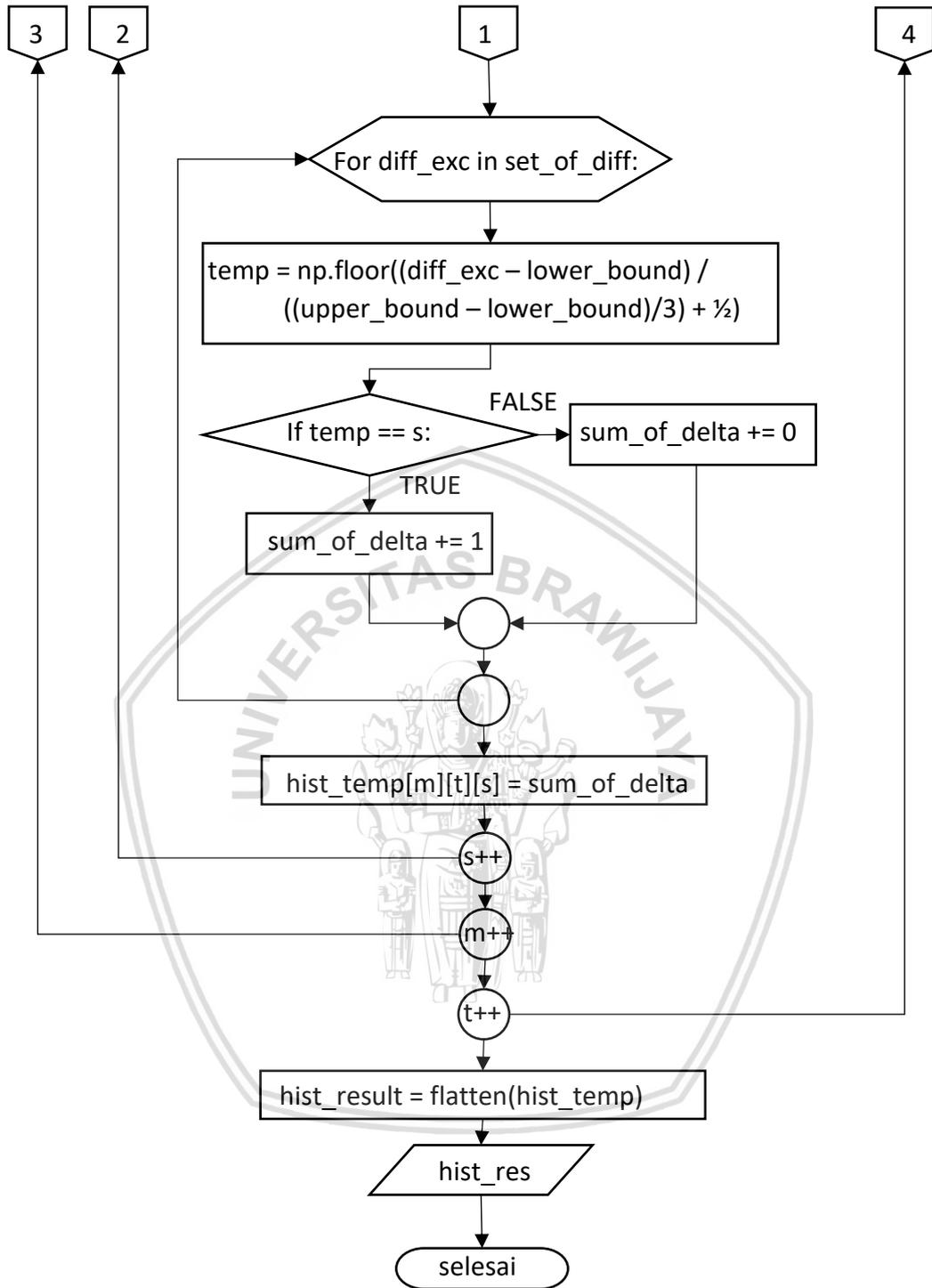
Gambar 4.16 menggambarkan alur proses ekstraksi *gradient orientation*. Berikut adalah penjelasan dari setiap tahapnya:

1. *Input* dari fungsi ini adalah sebuah *window* dari kumpulan *window*. Operasi diawal akan mengambil ukuran tinggi dan lebar dari *window*.
2. Nilai yang diambil dari matriks (*Window*) yang diberikan akan diambil nilai dari elemen ke p (urutan sesuai dengan jumlah tetangga) dan ke R (urutan sesuai dengan nilai radius) untuk membentuk fungsi x dan y
3. Kemudian menggunakan rumus arctan untuk membentuk nilai radian dan melakukan *quantization* terhadap nilai radian tersebut dengan menurunkan fungsi dari arctan tersebut sehingga menghasilkan rentang nilai yang berbeda dengan arctan.
4. Hasil dari fungsi ini disimpan menjadi satu kumpulan nilai *set of quationzation* dan digunakan saat pembentukan histogram 2D bersamaan dengan nilai *differential excitation*

Pembentukan Histogram WLD

```
compute_WLD_histogram()
```





**Gambar 4.17 Diagram Alir Pembentukan Histogram WLD**

Diagram alir pada Gambar 4.17 menjelaskan tentang alur pembuatan histogram pada MCWLD dimana semua nilai orientasi dan *differential excitation* terkumpul. Berikut adalah langkah-langkah yang dilakukan:

1. *Input* dari fungsi ini adalah kumpulan dari *differential excitation*, orientasi dan *quantization* dari orientasi.

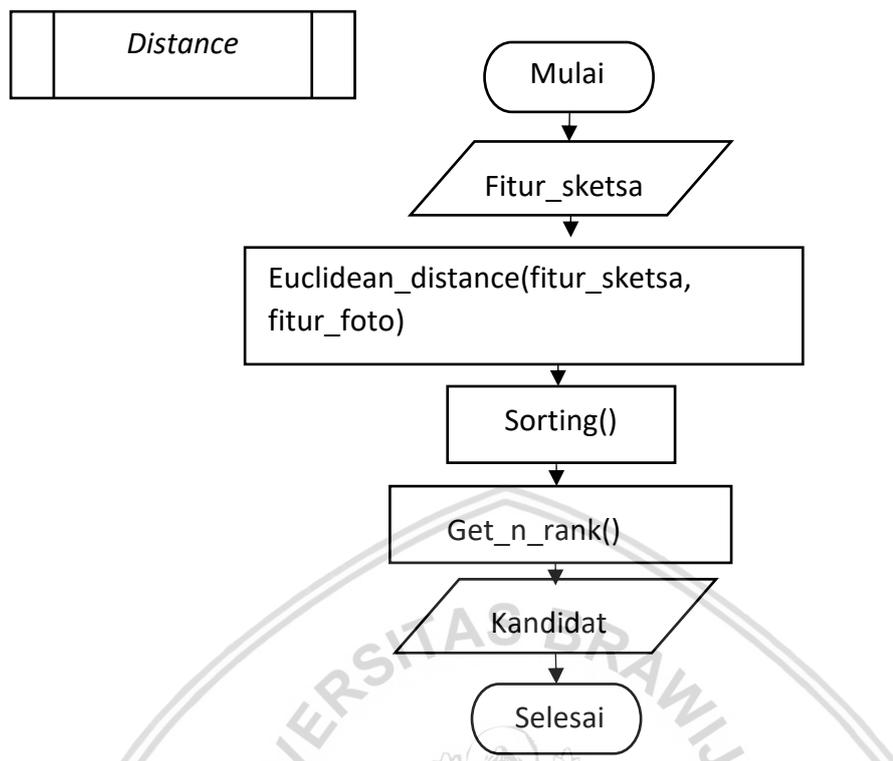


2. Proses selanjutnya adalah membuat histogram 2D dengan fungsi `histogram2d`. Bin yang digunakan untuk orientasi adalah nilai-nilai unik kumpulan *quantization*. Sedangkan bin untuk *differential excitation* di dapat berdasarkan jumlah *pixel* dari satu *window* (ukuran *window* = 16 x 16 *pixel*, total *pixel* 256).
3. Setelah membuat histogram 2 dimensi, langkah selanjutnya adalah membagi histogram 2 dimensi tersebut menjadi kumpulan histogram 1 dimensi. Langkah yang dilakukan adalah membuat batas bawah dan atas untuk masing-masing orientasi. Untuk setiap  $m$ , dibuat nilai batas bawah dan batas atas.
4. Untuk setiap nilai  $t$  atau jumlah orientasi dominan, dilakukan proses perulangan lagi. Untuk setiap  $s$  atau rentang pada bin, dilakukan eksekusi membagi nilai *differential excitation* ke dalam bin yang telah ditentukan.
5. Jika nilai temp sama dengan nilai  $s$  pada saat itu, maka `sum_of_delta` akan ditambah 1. Sebaliknya maka 0. Perulangan ini terus dilakukan hingga nilai histogram 2 dimensi berubah menjadi histogram 1 dimensi.
6. Diakhir, nilai histogram dari masing-masing potongan disambung menjadi satu histogram. Untuk satu *window size*, hasil yang diberikan adalah satu vektor berdimensi 144. Namun karena menggunakan tiga jenis ukuran *window*, maka hasil akhirnya adalah 432. Sama seperti metode SIFT, fitur diubah menjadi *unit length* atau unit vektor.

#### 4.1.3 Pemadanan dan $n$ -rank hasil

Proses ini mengembalikan  $n$ -rank hasil kandidat yang mengurutkan dari jarak terdekat dari hasil pemadanan. Adapun alurnya seperti pada Gambar 4.18. Berikut adalah langkah-langkahnya:

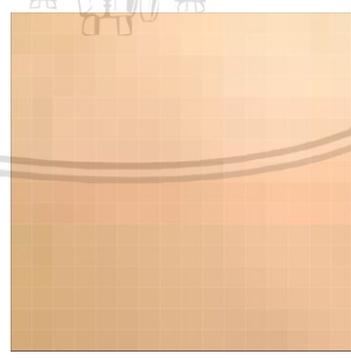
1. *Input* dari proses ini adalah fitur dari sketsa yang akan dicocokkan dengan fitur dari foto yang telah disimpan.
2. Fitur dari sketsa dihitung jaraknya dengan fitur dari seluruh foto menggunakan perhitungan jarak *euclidean distance*. Kemudian selanjutnya akan diurutkan dari yang terdekat atau memiliki nilai yang kecil.
3. Dari nilai jarak yang telah diurutkan untuk setiap masing-masing sketsa akan diambil  $n$ -rank hasil pemadanan untuk diproses pada perhitungan nilai *identification rate* pada proses evaluasi.



Gambar 4.18 Diagram alir pemdanan dan *n-rank* hasil

### 4.2 Perhitungan Manual

Perhitungan manual atau manualisasi pada laporan penelitian ini dilakukan pada citra berukuran 16 x 16 *pixel*. Ukuran tersebut merupakan ukuran dari *window* yang digunakan untuk melakukan ekstraksi fitur baik pada SIFT maupun pada MCWLD.



Gambar 4.19 Potongan Citra atau *Window*

#### 4.2.1 Pre-Processing: Grayscaleing

Proses *grayscaleing* merubah citra RGB menjadi nilai abu-abu. Berikut adalah nilai tabel yang menunjukkan nilai RGB dari citra 16 x 16 *pixel* pada Gambar 4.19:

Tabel 4.1 Nilai biru dari Gambar 4.19

$y \backslash x$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	134	137	140	144	151	158	161	166	167	171	176	177	177	176	173	177
1	133	135	135	141	147	153	159	163	167	171	177	179	180	179	174	58
2	142	141	139	142	149	155	156	158	163	166	176	178	180	181	176	179
3	140	137	140	143	147	150	152	156	161	167	177	179	181	181	177	179
4	139	136	144	147	148	151	152	155	159	164	169	172	175	174	172	58
5	138	134	144	144	147	147	149	152	157	161	166	168	171	170	166	168
6	139	139	145	145	146	146	147	150	153	158	165	168	171	170	168	168
7	137	138	140	142	145	145	145	147	152	153	163	165	167	168	165	165
8	135	136	139	140	144	143	146	146	149	151	156	157	160	159	158	158
9	133	132	135	136	143	145	145	145	149	152	156	158	161	161	160	160
10	126	125	134	133	137	138	142	144	147	148	153	156	158	159	155	154
11	127	124	131	132	136	137	141	143	146	147	152	155	157	158	157	155
12	125	124	130	131	133	135	140	142	144	145	149	152	155	156	156	156
13	126	125	128	129	132	134	138	140	142	143	146	148	151	153	154	155
14	123	122	126	127	130	131	136	138	140	141	143	144	146	149	151	154
15	121	120	125	125	128	129	134	136	138	139	140	140	141	145	148	150

Tabel 4.2 Nilai hijau dari Gambar 4.19

$y \backslash x$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	186	189	192	193	200	202	206	208	212	215	216	217	219	218	215	218
1	185	187	187	190	196	199	203	208	212	215	217	220	220	220	216	218
2	191	189	187	190	195	199	200	202	205	208	213	215	217	219	218	219
3	189	185	188	191	193	196	198	200	203	209	214	216	218	218	219	218
4	186	182	190	190	191	192	193	197	201	204	206	209	212	212	211	211
5	185	183	190	190	190	190	190	194	199	203	203	205	208	208	207	207
6	184	183	188	188	186	186	187	190	194	199	203	206	209	209	208	208
7	182	181	183	184	185	185	185	188	192	194	201	203	205	207	205	205
8	180	180	178	180	183	183	185	186	191	192	195	196	199	199	198	198
9	177	175	175	178	182	185	185	185	190	191	195	196	199	199	198	198
10	177	176	179	181	180	183	185	187	189	190	192	195	197	198	196	195
11	177	174	179	180	180	183	185	187	189	190	192	195	197	198	196	195
12	177	176	178	179	178	180	183	185	186	187	188	191	194	195	195	195
13	177	175	176	177	177	179	181	183	184	185	185	187	190	192	193	194
14	176	172	174	175	175	176	179	181	182	183	182	183	185	188	190	193
15	171	170	173	173	173	174	177	179	180	181	179	179	180	184	187	190

Tabel 4.3 Nilai merah dari Gambar 4.19

y\x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	232	236	239	241	246	249	250	253	255	255	251	252	254	253	250	251
1	231	234	234	234	238	242	246	250	252	255	255	253	255	253	251	251
2	237	237	235	238	242	246	247	249	250	251	251	253	255	254	255	254
3	235	233	236	239	241	244	246	247	248	252	252	254	255	255	255	255
4	230	229	238	239	240	241	242	244	248	252	250	253	255	254	255	253
5	229	229	238	238	239	239	239	241	246	250	247	249	252	250	252	251
6	228	230	239	239	238	238	236	238	243	248	251	254	255	254	255	255
7	226	230	234	237	238	237	237	237	240	243	249	251	253	252	254	253
8	224	227	230	232	235	235	237	238	238	241	247	248	251	251	250	250
9	224	224	227	231	234	237	237	238	239	243	247	250	253	253	252	252
10	217	216	222	223	223	226	228	230	232	232	237	240	242	243	241	240
11	219	216	221	222	222	225	227	229	231	232	236	239	241	242	241	241
12	218	217	220	221	221	223	226	228	229	230	232	236	239	240	240	240
13	218	217	218	219	220	222	224	226	227	228	229	232	234	237	238	239
14	215	214	216	217	218	219	222	224	225	226	226	227	229	232	234	237
15	213	212	215	216	218	220	222	223	224	223	223	224	228	224	232	235

Masing-masing dari nilai tersebut kemudian diproses menggunakan persamaan **(2.1)** untuk merubah ketiga nilai warna tersebut menjadi abu-abu. Berikut adalah contoh perhitungan nilai abu-abu yang dilakukan:

$$gray = (0.299 * 232) + (0.587 * 186) + (0.114 * 134)$$

$$gray = 69.368 + 106.182 + 19.296$$

$$gray = 193.846 \approx 193$$

Hasil dari perhitungan tersebut membuang nilai dibelakang koma karena pengaruh dari tipe data yang digunakan, yakni *numpy unsigned integer* yang memiliki rentang dari 0 hingga 255. Hasil dari proses *grayscale* ini selanjutnya digunakan pada operasi ekstraksi nilai fitur baik pada metode SIFT, maupun MCWLD. Hasil perhitungan *grayscale* ditunjukkan pada Tabel 4.4.

**Tabel 4.4 Hasil perhitungan *grayscale***

y\x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	193	197	200	201	208	211	214	216	219	221	221	222	224	223	220	223
1	192	195	195	198	204	207	212	216	219	221	222	225	225	225	221	223
2	199	197	195	198	203	208	209	211	213	216	220	222	224	225	224	224
3	197	193	196	199	202	205	207	209	211	217	221	223	224	224	224	224
4	193	190	199	199	200	201	202	206	210	213	214	217	220	220	219	219
5	192	191	199	199	199	199	199	203	208	212	211	213	216	216	215	215
6	192	192	198	198	196	196	197	199	203	208	213	216	218	218	217	217
7	190	190	193	195	196	195	195	197	201	203	211	213	215	216	215	214
8	188	189	189	190	194	193	196	196	200	201	206	207	210	209	208	208
9	186	184	185	189	193	195	195	196	199	202	206	207	210	210	209	209
10	183	182	186	188	187	190	192	194	197	198	201	204	206	207	204	203
11	183	180	186	187	186	189	191	193	196	197	200	203	205	206	205	203
12	183	182	185	186	185	187	190	192	194	195	196	200	203	204	204	204
13	182	181	183	184	184	186	188	190	192	193	193	196	198	201	202	203
14	179	178	181	182	182	183	186	188	190	191	190	191	193	196	198	201
15	117	176	180	180	180	182	184	186	188	189	187	187	188	192	196	199

## 4.2.2 SIFT

### 4.2.2.1 Padding

Proses *padding* adalah memberi *border* atau baris dan kolom baru yang nilainya dapat berupa nilai yang sama dengan baris dan kolom baik sebelum dan sesudahnya atau dapat juga bernilai nol. Pada sesuai dengan perancangan, nilai *padding* yang ditambahkan adalah nilai nol. Hasil dari *padding* ini digunakan pada metode SIFT untuk mendapatkan nilai besaran gradien dan orientasi. Fungsi ini digunakan pada citra yang telah dipotong menjadi  $4 \times 4$  *pixel* atau *window* kecil yang digunakan untuk mengekstraksi nilai gradien dan orientasi.

Tabel 4.5 Potongan *window* dari indeks 0 hingga 3

y\x	0	1	2	3
0	193	197	200	201
1	192	195	195	198
2	199	197	195	198
3	197	193	196	199

**Tabel 4.6 Penambahan kolom setelah kolom terakhir**

$y \setminus x$	0	1	2	3	4
0	193	197	200	201	0
1	192	195	195	198	0
2	199	197	195	198	0
3	197	193	196	199	0

**Tabel 4.7 Penambahan kolom sebelum kolom pertama**

$y \setminus x$	0	1	2	3	4	5
0	0	193	197	200	201	0
1	0	192	195	195	198	0
2	0	199	197	195	198	0
3	0	197	193	196	199	0

**Tabel 4.8 Penambahan baris setelah baris terakhir**

$y \backslash x$	0	1	2	3	4	5
0	0	193	197	200	201	0
1	0	192	195	195	198	0
2	0	199	197	195	198	0
3	0	197	193	196	199	0
4	0	0	0	0	0	0

**Tabel 4.9 Penambahan baris sebelum baris pertama**

$y \backslash x$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	193	197	200	201	0
2	0	192	195	195	198	0
3	0	199	197	195	198	0
4	0	197	193	196	199	0
5	0	0	0	0	0	0

#### 4.2.2.2 Ekstraksi Berasan Gradien dan pembobotan

Proses ini dilakukan setelah *window* diberi *padding*, karena proses ini membutuhkan nilai yang berasal dari *pixel* sebelum dan setelah *pixel* itu sendiri. Rumus yang digunakan ada pada rumus (2.8). Berikut adalah contoh perhitungan mendapatkan nilai besaran gradien dan pembobotannya:

- Besaran gradien

$$M_{y,x} = \sqrt{(I_{y+1,x+2} - I_{y+1,x})^2 + (I_{y+2,x+1} - I_{y,x+1})^2}$$

$$M_{0,0} = \sqrt{(I_{1,2} - I_{1,0})^2 + (I_{2,1} - I_{0,1})^2}$$

$$M_{0,0} = \sqrt{(197 - 0)^2 + (192 - 0)^2}$$

$$M_{0,0} = \sqrt{38809 + 36864}$$

$$M_{0,0} = 275.087$$

- Bobot *gaussian*

$$f_{y,x} = \frac{1}{2 * \pi} e^{-\frac{x^2+y^2}{std^2}}$$

$$f_{0,0} = \frac{1}{2 * \pi} e^{-\frac{(0-7)^2+(0-7)^2}{std^2}}$$

$$f_{0,0} = \frac{1}{2 * \pi} e^{-\frac{49+49}{std^2}}$$

$$f_{0,0} = \frac{1}{2 * 3.141592653589793} 2.7182818245^{-\frac{98}{24^2}}$$

$$f_{0,0} = \frac{1}{6.283185307179586} 2.7182818245^{-\frac{98}{576}}$$

$$f_{0,0} = 0.1342$$

- Weighted gradient

$$w_{y,x} = M_{y,x} * f_{y,x}$$

$$w_{0,0} = M_{0,0} * f_{0,0}$$

$$w_{0,0} = 275.087 * 0.1342$$

$$w_{0,0} = 36.9166754$$

Tabel 4.10 Hasil perhitungan besaran gradien dan pembobotan

y\x	0	1	2	3
0	<b>36.91</b>	26.79	27.29	40.01
1	26.78	0.42	0.83	28.35
2	27.58	0.64	0.20	28.90
3	39.41	28.64	28.91	41.94

### 4.2.2.3 Ekstraksi Orientasi

Sama seperti proses ekstraksi besaran gradien dan pembobotan, ekstraksi orientasi dilakukan setelah *window* kecil diberi padding. Proses ini terpisah dengan proses ekstraksi besaran gradien dan pembobotan. Rumus yang digunakan pada ekstraksi orientasi adalah rumus (2.9). Berikut adalah contoh dari perhitungan ekstraksi orientasi:

$$\theta_{y,x} = \arctan2 \left( (I_{y+2,x+1} - I_{y,x+1}), (I_{y+1,x+2} - I_{y+1,x}) \right) * \frac{180}{\pi}$$

$$\theta_{0,0} = \arctan2 \left( (I_{2,1} - I_{0,1}), (I_{1,2} - I_{1,0}) \right) * \frac{180}{\pi}$$

$$\theta_{0,0} = \arctan2((192 - 0), (197 - 0)) * \frac{180}{3.141592653589793}$$

$$\theta_{0,0} = 0.77254540073818212 * \frac{180}{3.141592653589793}$$

$$\theta_{0,0} = 44.26359094454071$$

Tabel 4.11 Hasil perhitungan ekstraksi orientasi dalam derajat

y\x	0	1	2	3
0	44.26°	87.94°	88.82°	135.28°
1	1.76°	0°	300.96°	180.88°
2	1.45°	206.56°	45°	179.70°
3	314.12°	269.70°	271.76°	225.29°

#### 4.2.2.4 Perhitungan *histogram binning*

Perhitungan *histogram binning* yang dilakukan pada SIFT adalah melakukan pengelompokan nilai *weighted gradient* sesuai dengan nilai orientasinya. Hasil dari kedua proses sebelumnya diperoleh sebuah matrix dengan masing-masing selnya memiliki nilai orientasi dan nilai *weighted gradient* seperti berikut:

**Tabel 4.12 Hasil ekstraksi orientasi dan pembobotan**

$y \backslash x$	0	1	2	3
0	44.26°, 36.91	87.94°, 26.79	88.82°, 27.29	135.28°, 40.01
1	1.76°, 26.78	0°, 0.42	300.96°, 0.83	180.88°, 28.35
2	1.45°, 27.58	206.56°, 0.64	45°, 0.20	179.70°, 28.90
3	314.12°, 39.41	269.70°, 28.64	271.76°, 28.91	225.29°, 41.94

Orientasi yang dibagi menjadi:

1. 0° – 45°
2. 45° – 90°
3. 90° – 135°
4. 135° – 180°
5. 180° – 225°
6. 225° – 270°
7. 270° – 315°

8.  $315^\circ - 360^\circ$

Sebagai contoh, pada sel indeks (0, 0) memiliki rentang nilai yang berada pada bin pertama, maka nilai *weighted gradient* ditambahkan pada bin tersebut. begitu juga dengan nilai lainnya sehingga hasil akhirnya adalah sebagai berikut:

**Tabel 4.13 Hasil *histogram binning***

$0^\circ - 45^\circ$	$45^\circ - 90^\circ$	$90^\circ - 135^\circ$	$135^\circ - 180^\circ$	$180^\circ - 225^\circ$	$225^\circ - 270^\circ$	$270^\circ - 315^\circ$	$315^\circ - 360^\circ$
91.72	54.30	0	40.01	57.90	41.94	58.39	39.41

Setelah semua nilai diekstraksi, maka hasilnya akan digabungkan menjadi satu nilai atau biasa disebut dengan vektor dengan representasi sebagai berikut:

91.72, 54.30, 0.0, 40.01, 57.90, 41.94, 58.39, 39.41,
39.40, 58.11, 0.0, 43.22, 61.83, 44.50, 62.08, 102.14,
41.77, 61.42, 0.0, 44.83, 64.89, 47.16, 64.88, 106.47,
30.89, 42.53, 62.93, 45.16, 65.33, 47.69, 66.06, 76.53,
1.02, 90.38, 0.0, 40.01, 58.29, 41.62, 57.28, 94.76,
37.97, 55.19, 0.0, 40.71, 58.16, 41.94, 60.64, 94.07,
41.13, 58.65, 0.0, 42.92, 62.25, 45.46, 62.70, 101.20,
41.39, 0.0, 59.75, 43.63, 63.71, 46.62, 63.86, 103.80,
36.48, 51.16, 0.0, 38.0, 54.47, 40.80, 54.77, 88.08,
0.0, 90.71, 0.0, 39.40, 56.81, 40.98, 58.31, 92.03,
37.97, 56.59, 0.0, 41.51, 59.75, 43.01, 59.52, 98.38,
0.0, 39.77, 58.09, 41.92, 60.65, 45.80, 60.34, 99.24,
0.0, 85.03, 0.0, 37.09, 53.45, 38.53, 54.78, 86.20,
35.22, 51.87, 0.0, 38.20, 54.91, 39.60, 54.19, 91.10,
36.73, 53.52, 0.0, 39.40, 56.28, 40.24, 58.92, 91.34,
38.16, 27.60, 28.27, 40.91, 58.73, 42.26, 62.66, 93.80

Nilai yang fitur tersebut kemudian dinormalisasi menjadi unit vektor sehingga hasilnya adalah sebagai berikut:

```
0.145, 0., 0.063, 0.091, 0.066, 0.092, 0.062, 0.091,  
0., 0.068, 0.097, 0.07, 0.098, 0.161, 0.066, 0.097,  
0., 0.070, 0.102, 0.074, 0.102, 0.168, 0.048, 0.067,  
0.099, 0.071, 0.103, 0.075, 0.104, 0.121, 0.001, 0.143,  
0., 0.063, 0.092, 0.065, 0.090, 0.149, 0.06, 0.087,  
0., 0.06, 0.092, 0.066, 0.095, 0.148, 0.065, 0.092,  
0., 0.067, 0.098, 0.071, 0.099, 0.160, 0.065, 0.,  
0.094, 0.069, 0.100, 0.073, 0.101, 0.164, 0.057, 0.080,  
0., 0.060, 0.086, 0.064, 0.086, 0.139, 0., 0.143,  
0., 0.062, 0.089, 0.064, 0.092, 0.145, 0.06, 0.089,  
0., 0.065, 0.094, 0.068, 0.094, 0.155, 0., 0.062,  
0.091, 0.066, 0.095, 0.072, 0.095, 0.157, 0., 0.134,  
0., 0.058, 0.084, 0.06, 0.086, 0.136, 0.055, 0.082,  
0., 0.06, 0.086, 0.062, 0.085, 0.144, 0.058, 0.084,  
0., 0.062, 0.089, 0.063, 0.093, 0.144, 0.06,  
0.043, 0.0447, 0.064, 0.092, 0.066, 0.099, 0.148
```

### 4.2.3 MCWLD

#### 4.2.3.1 Membagi *Window* Sesuai Ukuran Radius

Proses awal yang dilakukan oleh metode MCWLD adalah membagi citra menjadi potongan-potongan kecil yang menjadi luasan area untuk diekstraksi nilainya. *Window* kecil ini dibagi berdasarkan ukuran radiusnya. Pada penelitian ini, ukuran radius yang digunakan adalah 1, 2 dan 3. Namun pada bagian manualisasi ini, yang digunakan adalah radius ukuran 1.



**Gambar 4.20** *Window* dipotong dengan ukuran 3 x 3 *pixel* atau radius 1 *pixel*

**Gambar 4.21** Hasil perpotongan 3x3 *pixel* atau radius 1 *pixel*

### 4.2.3.2 Differential excitation

Proses yang terjadi pada tahap ini adalah mengurangi nilai titik tengah pada *window* kecil dengan tetangganya yang berada di radius yang telah ditentukan (dalam hal ini radius sama dengan 1). Perhitungan yang dilakukan mengikuti pada persamaan (2.11) dan dilakukan pada *window* pertama. Sebelumnya, *window* tersebut diberi filter dimana nilai tengah atau  $x_c$  dikurangi 8 atau sejumlah tetangganya dan tetangganya ditambah dengan satu.

$$\xi(x_c) = \arctan \left\{ \sum_{i=0}^{p-1} \left( \frac{x_i - x_c}{x_c} \right) \right\}$$

$$\xi(x_c) = \arctan \left\{ \sum_{i=0}^{8-1} \left( \frac{x_i - x_c}{x_c} \right) \right\}$$

$$\xi(x_c) = \arctan \left\{ \sum_{i=0}^7 \left( \frac{x_i - (195 - 8)}{(195 - 8)} \right) \right\}$$

$$\xi(x_c) = \arctan \left\{ \frac{((194 + 1) - (195 - 8))}{(195 - 8)} + \frac{((197 + 1) - (195 - 8))}{(195 - 8)} + \frac{((200 + 1) - (195 - 8))}{(195 - 8)} + \frac{((193 + 1) - (195 - 8))}{(195 - 8)} + \frac{((195 + 1) - (195 - 8))}{(195 - 8)} + \frac{((199 + 1) - (195 - 8))}{(195 - 8)} + \frac{((198 + 1) - (195 - 8))}{(195 - 8)} + \frac{((196 + 1) - (195 - 8))}{(195 - 8)} \right\}$$

$$\xi(x_c) = \arctan \left\{ \frac{((195) - (187))}{(187)} + \frac{((198) - (187))}{(187)} + \frac{((201) - (187))}{(187)} + \frac{((194) - (187))}{(187)} + \frac{((196) - (187))}{(187)} + \frac{((200) - (187))}{(187)} + \frac{((199) - (187))}{(187)} + \frac{((197) - (187))}{(187)} \right\}$$

$$\xi(x_c) = \arctan \left\{ \frac{(8)}{(187)} + \frac{(11)}{(187)} + \frac{(14)}{(187)} + \frac{(7)}{(187)} + \frac{(9)}{(187)} + \frac{(13)}{(187)} + \frac{(12)}{(187)} + \frac{(10)}{(187)} \right\}$$

$$\xi(x_c) = \arctan\{0.0427 + 0.0588 + 0.0748 + 0.0374 + 0.0481 + 0.0695 + 0.0641 + 0.0534\}$$

$$\xi(x_c) = \arctan\{0.4491\}$$

$$\xi(x_c) = 0.4221$$

#### 4.2.3.3 Gradient Orientation dan Orientation Quantization

Proses ekstraksi nilai orientasi gradien ini dilakukan pada *window* yang sama dengan operasi *differential excitation* sebelumnya. Perhitungan pada proses *Gradient Orientation* ini menggunakan persamaan (2.12) dan dilakukan pada *window* pertama. Sama seperti yang dilakukan pada operasi *differential excitation*, sebelum diberi dilakukan ekstraksi nilai yang dilakukan pertama kali adalah memberi filter terlebih dahulu seperti yang terlihat pada gambar berikut.

- orientasi

$$\theta(x_c) = \arctan \left\{ \frac{v_s^{11}}{v_s^{10}} \right\}$$

$$\theta(x_c) = \arctan \left\{ \frac{x_{(\frac{p}{2}+R)} - x_{(R)}}{x_{(P-R)} - x_{(\frac{p}{2}-R)}} \right\}$$

$$\theta(x_c) = \arctan \left\{ \frac{x_{(\frac{8}{2}+3)} - x_{(3)}}{x_{(8-3)} - x_{(\frac{8}{2}-3)}} \right\}$$

$$\theta(x_c) = \arctan \left\{ \frac{x_{(7)} - x_{(3)}}{x_{(5)} - x_{(1)}} \right\}$$

$$\theta(x_c) = \arctan \left\{ \frac{((193 + 1) - (195 - 1))}{((198 + 1) - (197 - 1))} \right\}$$

$$\theta(x_c) = \arctan \left\{ \frac{194 - 194}{199 - 196} \right\}$$

$$\theta(x_c) = \arctan \left\{ \frac{0}{3} \right\}$$

$$\theta(x_c) = 0$$

- merubah  $\theta$  menjadi  $\theta'$

$$\theta' = \arctan 2(v_s^{11}, v_s^{10}) + \pi$$

$$\theta' = \arctan 2 \left( \left( x_{(\frac{p}{2}+R)} - x_{(R)} \right), \left( x_{(P-R)} - x_{(\frac{p}{2}-R)} \right) \right) + \pi$$

$$\theta' = \arctan 2 \left( \left( x_{(\frac{8}{2}+3)} - x_{(3)} \right), \left( x_{(8-3)} - x_{(\frac{8}{2}-3)} \right) \right) + \pi$$

$$\theta' = \arctan 2 \left( \left( x_{(7)} - x_{(3)} \right), \left( x_{(5)} - x_{(1)} \right) \right) + \pi$$

$$\theta' = \arctan 2 \left( \left( (193 + 1) - (195 - 1) \right), \left( (198 + 1) - (197 - 1) \right) \right) + \pi$$

$$\theta' = \arctan 2 \left( (194 - 194), (199 - 196) \right) + \pi$$

$$\theta' = \arctan 2(0, 3) + 3.1415$$



$$\theta' = 0 + 3.1415$$

$$\theta' = 3.1415$$

- quantization

$$t = \text{mod} \left( \left\lfloor \frac{\theta'}{\frac{2\pi}{8}} + \frac{1}{2} \right\rfloor, T \right)$$

$$t = \text{mod} \left( \left\lfloor \frac{3.1415}{\frac{3.1415}{4}} + \frac{1}{2} \right\rfloor, 8 \right)$$

$$t = \text{mod} \left( \left\lfloor 4 + \frac{1}{2} \right\rfloor, 8 \right)$$

$$t = \text{mod}(4, 8)$$

$$t = 4$$

$$\Phi_t = f_q(\theta') = \frac{2t}{T} \pi$$

$$\Phi_4 = f_q(3.1415) = \frac{2 * 4}{8} * 3.1415$$

$$\Phi_4 = f_q(3.1415) = \frac{8}{8} * 3.1415$$

$$\Phi_4 = f_q(3.1415) = 3.1415$$

Nilai hasil *quantization* dari orientasi yang didapat dari *window* tersebut adalah 3.1415. Nilai hasil *quantization* ini akan menjadi tempat atau lokasi bin bagi nilai *differential excitation* pada *window* tersebut.

#### 4.2.3.4 Pembentukan Histogram Hasil

Proses pembentukan histogram hasil untuk metode MCWLD ini dilakukan setelah semua nilai pada *window* 16 x 16 *pixel* tersebut telah diekstraksi seluruh nilai orientasi dan *differential excitation*nya. Berikut adalah yang perlu diperhatikan:



1. Rentang nilai pada *M-Segment*. Nilai tersebut dapat dicari dengan persamaan (2.15) dan (2.16). Dengan nilai *M* pada *M-Segment* sama dengan 6, berikut adalah perhitungan pada nilai  $m=0$ :

$$l_m = [\eta_{m,l}, \eta_{m,u}]$$

$$l_0 = [\eta_{0,l}, \eta_{0,u}]$$

$$\eta_{m,u} = \left( \frac{(m+1)}{M} - \frac{1}{2} \right) \pi$$

$$\eta_{0,u} = \left( \frac{(0+1)}{6} - \frac{1}{2} \right) * 3.1415$$

$$\eta_{0,u} = \left( \frac{1}{6} - \frac{1}{2} \right) * 3.1415$$

$$\eta_{0,u} = -0.333 * 3.1415$$

$$\therefore \text{batas atas} = \eta_{0,u} = -1.0461195$$

$$\eta_{m,l} = \left( \frac{m}{M} - \frac{1}{2} \right) \pi$$

$$\eta_{0,l} = \left( \frac{0}{6} - \frac{1}{2} \right) * 3.1415$$

$$\eta_{0,l} = \left( 0 - \frac{1}{2} \right) * 3.1415$$

$$\eta_{0,l} = -\frac{1}{2} * 3.1415$$

$$\therefore \text{batas bawah} = \eta_{0,l} = -1.57079$$

$$\therefore l_0 = [-1.57079, -1.0461195]$$

2. Ketika sudah mendapatkan nilai rentang *M-Segment* tersebut, yang dilakukan berikutnya adalah melakukan pembentukan histogram. Pada penelitian ini, dilakukan alternatif pembuatan histogram 1-D yang dilakukan dengan cara langsung mengelompokkan masing-masing nilai pada binnya. Nilai yang didapat dari Gambar 4.19 adalah sebagai berikut:

Tabel 4.14 Nilai histogram MCWLD

Window ke-	Differential excitation	Gradient Orientation	Quantization orientasi	Rentang (indeks) m ke-	(Indeks) S ke-
1	0.422187	3.141593	0	3	2
2	0.482713	0.785398	0.785398	3	2
3	0.431744	1.570796	1.428899	3	2
4	0.383143	1.570796	1.292497	3	1
5	0.371856	1.570796	1.405648	3	1
6	0.330548	0.785398	1.107149	3	1
7	0.28527	0.785398	0.982794	3	1
8	0.231616	0.785398	0.785398	3	1
9	0.242778	0	0.244979	3	1
10	0.272146	1.570796	-1.5708	3	1
11	0.25687	2.356194	-0.7854	3	1
12	0.255743	3.926991	1.107149	3	1
13	0.282555	3.926991	1.107149	3	1
14	0.382117	3.926991	0.674741	3	1
15	0.276909	4.712389	1.373401	3	1
16	0.406816	3.141593	0.244979	3	2
17	0.383212	2.356194	-1.10715	3	1
18	0.320219	1.570796	-1.5708	3	1
19	0.286863	1.570796	1.249046	3	1
20	0.361494	0	0.321751	3	1
21	0.388127	0.785398	0.54042	3	1
22	0.383015	0.785398	0.463648	3	1
23	0.420452	0.785398	0.927295	3	2
24	0.364959	1.570796	-1.5708	3	1
25	0.369989	1.570796	-1.5708	3	1
26	0.338324	2.356194	-0.7854	3	1
27	0.303672	3.926991	0.785398	3	1
28	0.334198	3.141593	0.380506	3	1
29	0.455009	0	-0.38051	3	2
30	0.368591	2.356194	-0.67474	3	1
31	0.349607	2.356194	-0.7854	3	1
32	0.3689	1.570796	1.249046	3	1
33	0.332372	0.785398	0.643501	3	1
34	0.333762	0.785398	0.463648	3	1
35	0.343969	0.785398	0.785398	3	1
36	0.343645	1.570796	1.249046	3	1
37	0.2707	1.570796	-1.5708	3	1
38	0.226113	0.785398	0.927295	3	1
39	0.246073	0.785398	0.785398	3	1
40	0.226462	0	0	3	1
41	0.252554	5.497787	-0.588	3	1
42	0.248229	5.497787	-0.7854	3	1
43	0.533156	1.570796	1.249046	4	0



44	0.260928	2.356194	-1.05165	3	1
45	0.345002	3.141593	0	3	1
46	0.357066	3.141593		3	1
47	0.364408	0	0	3	1
48	0.389317	0	0.380506	3	1
49	0.366497	0.785398	0.896055	3	1
50	0.346782	1.570796	1.249046	3	1
51	0.314451	0.785398	0.785398	3	1
52	0.381339	0	0.2783	3	1
53	0.347241	0.785398	0.519146	3	1
54	0.283253	0	0	3	1
55	0.348392	5.497787	-0.40489	3	1
56	0.335834	5.497787	-0.40489	3	1
57	0.508489	2.356194	-0.9273	3	2
58	0.24621	1.570796	-1.40565	3	1
59	0.360491	4.712389	1.570796	3	1
60	0.331096	5.497787	-0.46365	3	1
61	0.345002	5.497787	-0.588	3	1
62	0.394791	0	0.244979	3	2
63	0.371633	0.785398	0.982794	3	1
64	0.318748	0.785398	1.05165	3	1
65	0.290557	0.785398	0.588003	3	1
66	0.403116	3.141593		3	2
67	0.411901	1.570796	-1.5708	3	2
68	0.35299	0	0	3	1
69	0.41241	4.712389	1.570796	3	2
70	0.396156	5.497787	-1.10715	3	2
71	0.423757	2.356194	-1.10715	3	2
72	0.237535	0.785398	0.785398	3	1
73	0.329624	5.497787	-0.98279	3	1
74	0.386876	5.497787	-0.98279	3	1
75	0.391405	5.497787	-0.7854	3	1
76	0.427026	0.785398	0.463648	3	2
77	0.394791	0.785398	1.030377	3	2
78	0.369906	0.785398	1.05165	3	1
79	0.317266	0.785398	0.86217	3	1
80	0.280231	1.570796	-1.3734	3	1
81	0.267472	1.570796	-1.24905	3	1
82	0.247218	3.141593		3	1
83	0.269473	3.926991	0.982794	3	1
84	0.301522	4.712389	1.249046	3	1
85	0.388957	0.785398	0.785398	3	1
86	0.366449	0	0.2783	3	1
87	0.348771	0	0.197396	3	1
88	0.337622	5.497787	-0.7854	3	1
89	0.3611	5.497787	-1.10715	3	1
90	0.397809	3.141593	0	3	2



91	0.398522	1.570796	1.325818	3	2
92	0.346286	0.785398	1.107149	3	1
93	0.417751	0.785398	0.950547	3	2
94	0.259989	0.785398	0.950547	3	1
95	0.302607	0	0.2783	3	1
96	0.282124	0	0.165149	3	1
97	0.262995	0	-0.32175	3	1
98	0.295441	5.497787	-0.46365	3	1
99	0.359376	0	-0.24498	3	1
100	0.411541	0	0	3	2
101	0.407527	0.785398	0.643501	3	2
102	0.359949	0.785398	0.785398	3	1
103	0.41972	3.141593	0	3	2
104	0.370378	2.356194	-0.46365	3	1
105	0.400413	1.570796	-1.5708	3	2
106	0.363329	1.570796	-1.5708	3	1
107	0.421567	1.570796	-1.5708	3	2
108	0.370891	0.785398	0.785398	3	1
109	0.449479	0.785398	0.588003	3	2
110	0.38986	0.785398	0.463648	3	1
111	0.415004	5.497787	-0.7854	3	2
112	0.445577	5.497787	-0.7854	3	2
113	0.43384	0	-0.38051	3	2
114	0.440924	1.570796	-1.5708	3	2
115	0.397534	1.570796	1.373401	3	2
116	0.308722	0.785398	0.896055	3	1
117	0.254966	0.785398	0.785398	3	1
118	0.318556	5.497787	-1.10715	3	1
119	0.393279	1.570796	-1.5708	3	2
120	0.298133	1.570796	1.325818	3	1
121	0.327923	0.785398	1.107149	3	1
122	0.256826	0.785398	0.927295	3	1
123	0.286863	1.570796	1.249046	3	1
124	0.227518	0.785398	0.463648	3	1
125	0.250766	4.712389	-1.24905	3	1
126	0.275027	5.497787	-0.98279	3	1
127	0.495311	0.785398	0.785398	3	2
128	0.323426	2.356194	-1.10715	3	1
129	0.375708	4.712389	1.570796	3	1
130	0.459193	0	0.244979	3	2
131	0.407527	0.785398	0.643501	3	2
132	0.399006	0.785398	0.785398	3	2
133	0.386026	1.570796	-1.5708	3	1
134	0.373187	0.785398	0.463648	3	1
135	0.4294	0.785398	0.588003	3	2
136	0.401761	0.785398	0.785398	3	2
137	0.387524	0.785398	0.785398	3	1
138	0.379635	0	0.244979	3	1



139	0.360422	5.497787	-0.7854	3	1
140	0.420107	5.497787	-1.03038	3	2
141	0.506608	3.141593	0	3	2
142	0.335173	1.570796	-1.5708	3	1
143	0.367922	4.712389	1.570796	3	1
144	0.420296	1.570796	-1.5708	3	2
145	0.357566	1.570796	1.249046	3	1
146	0.368266	1.570796	-1.5708	3	1
147	0.364649	1.570796	-1.5708	3	1
148	0.34235	0.785398	0.785398	3	1
149	0.373187	0.785398	1.107149	3	1
150	0.335747	0.785398	1.107149	3	1
151	0.317128	0.785398	0.982794	3	1
152	0.318702	0.785398	0.785398	3	1
153	0.308053	5.497787	-1.10715	3	1
154	0.336901	4.712389	1.373401	3	1
155	0.445088	3.141593	0	3	2
156	0.34195	0.785398	1.107149	3	1
157	0.360001	5.497787	-0.7854	3	1
158	0.413	3.141593		3	2
159	0.394791	1.570796	1.249046	3	2
160	0.355773	1.570796	1.249046	3	1
161	0.347468	0.785398	0.785398	3	1
162	0.364649	0	0	3	1
163	0.381428	0.785398	0.463648	3	1
164	0.373187	0.785398	0.643501	3	1
165	0.345002	0.785398	0.674741	3	1
166	0.307841	0.785398	0.463648	3	1
167	0.320219	0	-0.32175	3	1
168	0.333945	5.497787	-1.10715	3	1
169	0.39234	5.497787	-1.10715	3	1
170	0.375571	0	0	3	1
171	0.383442	0	0	3	1
172	0.386337	0.785398	0.785398	3	1
173	0.367922	0.785398	0.785398	3	1
174	0.369024	0.785398	0.785398	3	1
175	0.360566	0.785398	0.463648	3	1
176	0.372988	0	0	3	1
177	0.371152	0	0	3	1
178	0.383284	0	0.244979	3	1
179	0.374992	0.785398	0.463648	3	1
180	0.337374	0.785398	0.404892	3	1
181	0.329502	0	0.197396	3	1
182	0.346286	0	0	3	1
183	0.456605	0	-0.32175	3	2
184	0.389444	0.785398	0.785398	3	1
185	0.397237	0	0	3	2
186	0.395197	0	0	3	2



187	0.421909	0.785398	0.588003	3	2
188	0.372777	0.785398	0.982794	3	1
189	0.364209	0.785398	0.463648	3	1
190	0.376711	0	0	3	1
191	0.360566	5.497787	-0.46365	3	1
192	0.388957	0	-0.24498	3	1
193	0.387048	0	0.165149	3	1
194	0.383284	0	0.358771	3	1
195	0.350045	0.785398	0.463648	3	1
196	0.365067	0.785398	0.643501	3	1

### 4.3 Perancangan Skenario Pengujian

Pengujian yang dilakukan pada penelitian ini merupakan pengujian untuk melihat dampak dari perubahan sebuah parameter terhadap akurasi yang didapatkan. Evaluasi *close-set identification* yang dilakukan terhadap nilai *top match* atau rank-1, rank-5, rank-10, rank-25, dan rank-50. Selain itu juga dievaluasi berdasarkan sisi fiturnya, nilai EER dan rasio *within-between class variance ratio*. Sebuah nilai dikatakan sebagai benar jika foto dan sketsa merupakan orang yang sama dan berada pada salah satu rank yang ditentukan. Nilai yang ditunjukkan pada pengujian ini adalah *identification rate*. Tujuan dari pengujian ini untuk mengetahui seberapa baik metode ini dapat bekerja pada ruang lingkup *heterogeneous matching*. Berikut adalah rincian dari skenario pengujian:

1. Pengujian *overlapping* dan *non-overlapping*

Pengujian ini dilakukan dengan melihat pengaruh dari *overlapping* dan *non-overlapping* pada dua jenis ukuran *window* yakni 16 dan 32. Untuk ukuran 16x16 *pixel*, besaran perpindahannya adalah 9 *pixel* sedangkan untuk 32x32 *pixel* besaran perpindahannya adalah 18 *pixel*. Sama seperti skenario pertama, nilai yang diamati adalah pengaruhnya pada evaluasi *close-set identification*.

2. Pengujian parameter pada SIFT dan MCWLD

Pengujian ini dilakukan dengan mengubah parameter yang mempengaruhi metode SIFT dan MCWLD dalam proses pengambilan fitur. Adapun parameter yang diubah pada metode SIFT adalah ukuran *window* kecil. Ukuran *window* kecil ini mempengaruhi luasan area dalam mengekstraksi nilai fiturnya. Sedangkan untuk metode MCWLD, parameter yang diubah adalah *T-dominant orientation*, *M-Segment*, dan *S-bin*. Parameter tersebut mempengaruhi ukuran histogram yang dihasilkan serta mempengaruhi proses *histogram binning* untuk menghasilkan sebuah vektor fitur dari setiap *window* dari *sliding window*. Tujuan dari merubah parameter pada

MCWLD ini adalah untuk melihat perubahan fitur yang dihasilkan pada satu *window*. Pengujian ini melanjutkan dari pengujian sebelumnya.

3. Pengujian perbandingan metode SIFT, MCWLD dan SIFT+MCWLD  
Pengujian ini dilakukan dengan membandingkan nilai *identification rate* dari SIFT, MCWLD dan kombinasinya. Tujuan dari pengujian ini untuk melihat metode terbaik dari ketiga metode tersebut.



## BAB 5 IMPLEMENTASI

Pada bab 5 ini akan dibahas proses implementasi berdasarkan apa yang sudah dirancang pada bab 4 Perancangan. Pembahasan pada bab ini mencakup penjelasan dari lingkungan implementasi, batasan implementasi dan implementasi aplikasi. Lingkungan implementasi terdiri dari lingkungan *hardware* dan lingkungan *software*. Batasan implementasi membahas tentang batas-batas yang diterapkan pada implementasi penelitian ini. Implementasi aplikasi melibatkan tentang implementasi pada *code* yang terdiri dari tahap-tahapan yang sudah dirancang.

### 5.1 Lingkungan Implementasi

#### 5.1.1 Lingkungan Hardware

Implementasi penelitian ini dilakukan pada lingkungan *hardware* dengan spesifikasi sebagai berikut:

1. *Processor*: Intel® Core™ i5-5200U @ 2.20 GHz (2 cores 4 Threads)
2. *Memory*: 8 GB
3. *Harddisk*: 1 TB

#### 5.1.2 Lingkungan Software

Berikut adalah lingkungan *software* yang digunakan dalam implementasi penelitian ini:

1. Sistem operasi *Microsoft Windows 10 (Ver. 1709)*
2. *Microsoft Office Word 2016* sebagai penunjang dalam penulisan dan pembuatan dokumen
3. *Microsoft Office Presentation 2016* sebagai penunjang dalam membuat presentasi penelitian
4. *PyCharm (ver 2017.2)* sebagai penunjang dalam menulis *code* pada penelitian ini
5. *Anaconda (ver 4.3.34)* digunakan untuk mengatur *package* atau *histogram binning* pada bahasa pemrograman Python (versi 3)
6. *Histogram binning* yang digunakan antara lain:
  - a. *OpenCV*
  - b. *NumPy*
  - c. *Matplotlib*
  - d. *Numba*
  - e. *Scikit-learn*

## 5.2 Batasan Implementasi

Berikut ini adalah batasan dalam mengimplementasikan Identifikasi Individu Berdasarkan Sketsa Wajah Menggunakan Pendekatan Diskriminatif:

1. Data yang digunakan adalah *Chinese University of Hong Kong Public Data Set*. Data ini terdiri dari 54 foto wajah dan 54 sketsa wajah perempuan, 134 foto wajah dan 134 sketsa wajah laki-laki. Foto dan sketsa saling berkorespondensi.
2. Metode *preprocessing* yang digunakan adalah *grayscale* dan *sliding window*. Ukuran citra yang digunakan berdimensi 250 x 200 *pixel* dengan nilai IPD 50 (*level 2 details*).
3. Ukuran *region size* atau ukuran *window* yang digunakan adalah 16x16 *pixel* dan *displacement size* yang digunakan adalah 9 *pixel* untuk *sliding window* dengan *overlapping* baik pada SIFT dan MCLWD. Konfigurasi ini dijadikan sebagai *baseline* dalam penelitian sebagai pembanding untuk pengujian.
4. Jumlah *patch* yang dihasilkan dari *overlapping sliding window* adalah 567 dan 180 untuk *non-overlapping sliding window* dengan masing-masing *patch* merupakan vektor berdimensi 128 milik SIFT atau vektor berdimensi 432 milik MCWLD.
5. Pada SIFT, operasi dilakukan pada *window* berukuran 16x16 *pixel* yang terbagi menjadi 4x4, sedangkan MCWLD menggunakan radius 1 *pixel*, 2 *pixel* dan 3 *pixel*.
6. Metode pemadanan yang digunakan adalah *euclidean distance*
7. Fusi algoritme dilakukan pada tingkat fitur dengan menggunakan *min-max normalization* dan fitur dirangkai menjadi satu, sedangkan pada tingkat *score* dilakukan dengan *min-max normalization* pada nilai jarak dan dijumlahkan sehingga menghasilkan nilai jarak baru.

## 5.3 Implementasi Aplikasi

Subbab ini menjelaskan tentang implementasi *code* berdasarkan apa yang sudah dijabarkan pada perancangan di Bab 4. Adapun bagian pada subbab ini terbagi menjadi *preprocessing*, ekstraksi fitur, proyeksi diskriminan dan algoritme fusi. Masing-masing bagian dibagi menjadi sub bagian sesuai dengan perancangan.

### 5.3.1 Pre-Processing

<i>grayscale</i>	
1	<code>def grayscale(rgb_img):</code>
2	<code>rows, columns, channels = rgb_img.shape</code>
3	<code>gray_img = np.zeros((rows, columns), dtype=np.uint8)</code>
4	<code>for y in range(rows):</code>
5	<code>for x in range(columns):</code>
6	<code>Blue, Green, Red = rgb_img[y, x]</code>
7	<code>gray = ((Red * 0.299) + (Green * 0.587) + (Blue * 0.114))</code>

```

8         gray_img[y, x] = gray
       return gray_img
    
```

**Kode Sumber 5.1 Proses *grayscale***

Berikut adalah penjelasan dari Kode Sumber 5.1 Proses *grayscale*

:

1. Baris pertama hingga ke 3 merupakan bagian deklarasi fungsi *grayscale* dan inialisasi variabel. Parameter yang digunakan adalah citra RGB. Variabel *gray\_img* merupakan matriks bernilai 0 yang berdimensi sama dengan citra RGB.
2. Untuk baris 4 hingga baris 7, merupakan perulangan terhadap baris dan kolom pada citra. Perulangan ini digunakan untuk mengakses setiap sel pada matriks citra RGB pada setiap kolom untuk setiap barisnya. Konversi nilai RGB menjadi abu-abu ada pada baris ke-6 yang menggunakan rumus (2.1). Baris ke-7 menyimpan nilai abu-abu tersebut pada matriks *gray\_img*.
3. Baris ke-8 mengembalikan nilai berupa matriks *gray\_img*.

```

Sliding window
1 @numba.jit(parallel=True)
2 def __sliding_window_sift(self):
3     for row in range(0, self.__image.shape[0],
4 self.__displacement_size):
5         if row + self.__region_size >=
6 self.__image.shape[0]:
7             continue
8         for column in range(0, self.__image.shape[1],
9 self.__displacement_size):
10            if column + self.__region_size >=
11 self.__image.shape[1]:
12                continue
13            self.__set_of_sliding_windows.append(
14                self.__image[row:row + self.__region_size,
15 column:column + self.__region_size])
16            self.__displacement_smaller_region_y =
17 int(np.sqrt(self.__set_of_sliding_windows[0].shape[0]))
18            self.__displacement_smaller_region_x =
19 int(np.sqrt(self.__set_of_sliding_windows[0].shape[1]))
20            self.__smaller_region_y =
21 self.__set_of_sliding_windows[0].shape[0]
22            self.__smaller_region_x =
23 self.__set_of_sliding_windows[0].shape[1]
    
```

**Kode Sumber 5.2 Fungsi *Sliding window SIFT***

Berikut adalah penjelasan dari Kode Sumber 5.2 Fungsi *Sliding window SIFT*

:

1. Baris pertama merupakan properti atau *pointer* yang menandakan fungsi dibawahnya menggunakan optimalisasi dari *histogram binning numba*.



2. Baris ke- 3 hingga baris ke-9 merupakan perulangan untuk mengakses matriks citra dan memotongnya sesuai dengan ukuran yang telah ditentukan.
3. Baris ke-10 hingga baris ke-13 melakukan inialisasi panjang baris dan kolom untuk *window* berukuran yang lebih kecil.

### 5.3.2 Implementasi Ekstraksi Fitur

#### 5.3.2.1 SIFT

Inialisasi SIFT	
1	<code>import numba</code>
2	<code>import numpy as np</code>
3	<code>import cv2</code>
4	<code>import Util as U</code>
5	<code>class SIFT:</code>
6	<code>def __init__(self):</code>
7	<code>self.__set_of_sliding_windows = []</code>
8	<code>self.__set_of_smaller_windows = []</code>
9	<code>def SIFT_feature_extraction(self, image, region_size=16,</code>
	<code>displacement_size=9):</code>
10	<code>if np.ndim(image) == 2:</code>
11	<code>self.__image = image</code>
12	<code>self.__rows, self.__columns = self.__image.shape</code>
13	<code>elif np.ndim(image) == 3:</code>
14	<code>self.__image = U.grayscale(image)</code>
15	<code>self.__rows, self.__columns = self.__image.shape</code>
16	<code>self.__region_size = region_size</code>
17	<code>self.__displacement_size = displacement_size</code>
18	<code>self.__std = 1.5 * self.__region_size</code>
19	<code>self.__sliding_window_sift()</code>
20	<code>self.__smaller_window from sliding_window()</code>

Kode Sumber 5.3 Inialisasi SIFT

Berikut adalah penjelasan dari Kode Sumber 5.3 Inialisasi SIFT

:

1. Baris pertama hingga ke-4 merupakan Kode Sumber untuk mengimpor *histogram binning* yang dibutuhkan pada program antara lain *numba* yang digunakan untuk mempercepat perhitungan dari *histogram binning* *numpy* yang merupakan *histogram binning* yang mengatur perhitungan matriks dan aljabar linear. *Histogram binning cv2* adalah *histogram binning* milik OpenCV yang digunakan untuk memproses segala hal yang berkaitan dengan pemrosesan citra digital. *Util* merupakan package yang dibuat sendiri yang berisi segala fungsi yang menunjang berjalannya program.
2. Baris ke-5 merupakan deklarasi dari kelas SIFT. Ekstraksi Fitur SIFT dibuat dalam bentuk kelas untuk mempermudah implementasi.
3. Baris ke-6 hingga ke-8 merupakan inialisasi dari seluruh variabel yang digunakan pada operasi SIFT, meliputi:

- a. `self.__set_of_sliding_windows` merupakan variabel yang menyimpan seluruh *window* dari hasil *sliding window*.
  - b. `self.__set_of_smaller_windows` merupakan variabel yang menyimpan seluruh *window* kecil dari setiap *window*
4. Baris ke-9 merupakan deklarasi fungsi SIFT. Pada fungsi ini, yang dilakukan hanyalah melakukan inisialisasi variabel yang digunakan pada SIFT. Parameter yang digunakan adalah citra, ukuran region serta ukuran perpindahan. Masing-masing parameter tersebut masukan ke dalam variabel baru.
  5. Baris ke-18 merupakan inisialisasi variabel deviasi standar dengan nilai 1.5 kali dari ukuran *window*.
  4. Baris ke-19 dan 20 menjalankan fungsi *sliding window* serta memotong *window* menjadi *window* yang lebih kecil.

Smaller window	
1	<code>@numba.jit(parallel=True)</code>
2	<code>def __smaller_window from sliding_window(self):</code>
3	<code>    for window in self.__set_of_sliding_windows:</code>
4	<code>        temp = []</code>
5	<code>        for row in range(0, self.__smaller_region_y,</code>
6	<code>self.__displacement_smaller_region_y):</code>
7	<code>            for column in range(0, self.__smaller_region_x,</code>
	<code>self.__displacement_smaller_region_x):</code>
	<code>                temp.append(window[row:row +</code>
	<code>self.__displacement_smaller_region_y,</code>
	<code>                                column:column +</code>
	<code>self.__displacement_smaller_region_x])</code>
	<code>        self.__set_of_smaller_windows.append(temp)</code>

**Kode Sumber 5.4 Fungsi pemotongan *Smaller Window***

Berikut adalah penjelasan Kode Sumber 5.4 Fungsi pemotongan *Smaller Window*

:

1. Fungsi diatas merupakan fungsi untuk memotong *window* menjadi ukuran *window* yang lebih kecil. baris pertama dan kedua merupakan instansiasi `numba.jit` serta deklarasi fungsi.
2. Baris ke-3 sampai dengan baris ke-7 merupakan operasi untuk memotong *window* dan menyimpang hasil potongannya. Adapun yang dilakukan pada baris ke-7 sama dengan *sliding window* tanpa *overlapping window*.

Ekstraksi fitur sift	
1	<code>@numba.jit(parallel=True)</code>
2	<code>def feature_extraction(self):</code>
3	<code>    output_volume = []</code>
4	<code>    for sliding_windows in self.__set_of_smaller_windows:</code>
5	<code>        temp =</code>
6	<code>self.__inner_feature_extraction(sliding_windows)</code>
	<code>        temp =</code>



```

7 self.__normalize_feature_vector(feature_vector=temp)
8     output_volume.append(temp)
9     self.__reset()
    return output_volume

```

**Kode Sumber 5.5 Fungsi Ekstraksi Fitur SIFT**

Berikut adalah penjelasan dari Kode Sumber 5.5 Fungsi Ekstraksi Fitur SIFT

:

1. Baris pertama dan kedua merupakan instansiasi numba.jit serta deklarasi fungsi ekstraksi fitur.
2. Pada baris ke-3 variabel `ouput_volume` menyimpan hasil perhitungan histogram dari setiap *window* kecil.
3. Baris ke-4 merupakan perulangan untuk setiap *window* dari *sliding window*. Variabel `sliding_windows` tersebut sudah terbagi menjadi *window* yang lebih kecil.
4. Baris ke-5 hingga ke-7 merupakan pemanggilan fungsi untuk mengekstraksi nilai fitur SIFT dan menyimpannya pada variabel `temp`. Variabel `temp` ini kemudian dinormalisasi menjadi unit vektor dan disimpan pada variabel `output_volume`
5. Baris ke-9 mengembalikan variabel `output_volume`

```

Zero padding
1 def add_image_padding(image, iterasi=1):
2     temp = image.copy()
3     for i in range(0, int(iterasi)):
4         if i == 0:
5             zero_padding_column = np.zeros(image.shape[0])
6         else:
7             zero_padding_column = np.zeros(temp.shape[0])
8             temp = np.insert(temp, temp.shape[1],
9 zero_padding_column, axis=1)
10            temp = np.insert(temp, 0, zero_padding_column,
11 axis=1)
11            zero_padding_row = np.zeros(temp.shape[1])
12            temp = np.insert(temp, temp.shape[0],
13 zero_padding_row, axis=0)
13            temp = np.insert(temp, 0, zero_padding_row, axis=0)
    return temp

```

**Kode Sumber 5.6 Fungsi Zero Padding**

**Berikut adalah penjelasan dari Kode Sumber 5.6 Fungsi Zero Padding**

:

1. Baris pertam merupakan deklarasi dari fungsi *zero padding* dengan parameternya adalah citra serta jumlah iterasi yang disimpan pada variabel `iterasi`. Secara default nilai variabel `iterasi` adalah 1.
2. Baris kedua merupakan inisialisasi variabel `temp` dengan nilai yang sama dengan citra yang masukkan.



- Baris ketiga hingga ke-12 merupakan operasi penambahan padding bernilai nol yang berurutan dari kolom paling akhir, kolom pada indeks awal, baris pada indeks akhir serta baris pada indeks awal.
- Baris ke-13 mengembalikan matriks temp yang berukuran baris+2 x kolom+2

Ekstraksi orientasi	
1	<code>@numba.jit(parallel=True)</code>
2	<code>def __orientation_extraction(self, smaller_window):</code>
3	<code>    y, x = smaller_window.shape</code>
4	<code>    padded_window =</code>
	<code>    np.float64(U.add_image_padding(smaller_window, 1))</code>
5	<code>    temp = np.zeros((y, x))</code>
6	<code>    for row in numba.prange(0, y):</code>
7	<code>        for column in numba.prange(0, x):</code>
8	<code>            func_y = (padded_window[row + 2, column + 1] -</code>
	<code>            padded_window[row, column + 1])</code>
9	<code>            func_x = (padded_window[row + 1, column + 2] -</code>
	<code>            padded_window[row + 1, column])</code>
10	<code>            temp_rad = np.arctan2(func_y, func_x) * 180 /</code>
11	<code>            np.pi</code>
	<code>            if temp_rad &lt; 0.0:</code>
	<code>                temp[row, column] = temp_rad + 360.0</code>
12	<code>            elif temp_rad &gt; 0.0:</code>
13	<code>                temp[row, column] = temp_rad</code>
	<code>    return temp</code>

#### Kode Sumber 5.7 Fungsi Ekstraksi Orientasi

Berikut adalah penjelasan dari Kode Sumber 5.7 Fungsi Ekstraksi Orientasi

:

- Baris pertama dan kedua merupakan instansiasi numba.jit serta deklarasi fungsi ekstraksi orientasi.
- Baris ke-3 hingga ke-5 merupakan inisialisasi variabel. Nilai variabel y dan x didapat dari mengambil ukuran dari smaller\_window. Padded\_window merupakan variabel yang menyimpan nilai smaller\_window yang telah diberi padding. Serta variabel temp merupakan matriks dengan nilai elementnya 0 dan digunakan untuk menyimpan hasil ekstraksi nilai 0
- Baris ke-6 hingga ke-10 merupakan bagian perulangan baris-kolom untuk mendapatkan nilai orientasi. Pada kasus ini, nilai setiap fungsinya diambil dari padded\_window dengan nilai indeks yang digunakan diberi +1 atau -1 karena letak dari smaller\_window pada padded\_window berada di baris ke-1 dan di kolom ke-1 (tidak dimulai dari 0). Temp\_rad merupakan variabel yang menyimpan hasil dari fungsi arctan2 dengan rentang nilai [-pi, pi].
- Baris ke-11 hingga ke-12 merubah nilai dari rentang tersebut menjadi rentang [0, 360]

5. Baris ke-13 mengembalikan matriks yang setiap selnya berisi nilai orientasi dalam derajat.

```

Gradient magnitude
1  @numba.jit(parallel=True)
2  def __gradient_magnitude(self, smaller_window):
3      y, x = smaller_window.shape
4      padded_window =
np.float64(U.add_image_padding(smaller_window, 1))
5      temp = np.zeros((y, x))
6      for row in numba.prange(0, y):
7          for column in numba.prange(0, x):
8              func_x = (padded_window[row + 1, column + 2] -
padded_window[row + 1, column])
9              func_y = (padded_window[row + 2, column + 1] -
padded_window[row, column + 1])
10             temp_mag = np.sqrt((func_x ** 2) + (func_y **
11 2))
12             temp_gwf = (1 / (2 * np.pi)) * ( np.exp((-1 *
((column - 7) ** 2) + ((row - 7) ** 2)) / (self.__std) /
13 self.__std))
14             temp[row, column] =temp_mag * temp_gwf
15         return temp
    
```

**Kode Sumber 5.8 Fungsi Besaran Gradien dan pembobotan**

Berikut adalah penjelasan dari Kode Sumber 5.8 Fungsi Besaran Gradien dan pembobotan

- :
1. Baris pertama dan kedua merupakan instansiasi numba.jit serta deklarasi fungsi ekstraksi besaran gradien dan pembobotan.
  2. Baris ke-3 hingga ke-5 merupakan inisialisasi variabel. Nilai variabel y dan x didapat dari mengambil ukuran dari *smaller\_window*. *Padded\_window* merupakan variabel yang menyimpan nilai *smaller\_window* yang telah diberi padding. Serta variabel temp merupakan matriks dengan nilai elementnya 0 dan digunakan untuk menyimpan hasil ekstraksi nilai 0
  3. Baris ke-6 hingga ke-11 merupakan perulangan untuk setiap baris dan kolom. Pada baris ke-8 dan ke-9 mengambil nilai fungsi x dan y dengan mengurangi elemen yang berada di depan dan di belakang dari posisi *pixel* saat itu. Baris ke-10 merupakan akhir dari besaran gradien yang didapat dari akar penjumlahan dari fungsi x dan y yang dikuadratkan. Baris ke-11 merupakan fungsi gaussian. Menggunakan nilai - 7 karena titik tengahnya berada di elemen ke 7,7
  4. Baris ke-13 merupakan nilai besaran gradien yang telah diberi bobot dengan cara mengalikan dengan temp\_gwf, fungsi gaussian.
  5. Baris ke-14 mengembalikan matriks yang setiap elemennya merupakan besaran gradien yang telah diberi bobot.

*Histogram\_binning*

```

1 def __histogram_orientation_binning(self, orientation:
  np.ndarray, weight: np.ndarray):
2     hist = np.zeros(8, dtype=np.float64)
3     for row in range(0, orientation.shape[0]):
4         for column in range(0, orientation.shape[1]):
5             if orientation[row, column] >= 0.0 and
orientation[row, column] < 45.0:
6                 hist[0] += weight[row, column]
7             elif orientation[row, column] >= 45.0 and
orientation[row, column] < 90.0:
8                 hist[1] += weight[row, column]
9             elif orientation[row, column] >= 90.0 and
orientation[row, column] < 130.0:
10                hist[2] += weight[row, column]
11             elif orientation[row, column] >= 130.0 and
orientation[row, column] < 175.0:
12                hist[3] += weight[row, column]
13             elif orientation[row, column] >= 175.0 and
orientation[row, column] < 220.0:
14                hist[4] += weight[row, column]
15             elif orientation[row, column] >= 220.0 and
orientation[row, column] < 265.0:
16                hist[5] += weight[row, column]
17             elif orientation[row, column] >= 265.0 and
orientation[row, column] < 310.0:
18                hist[6] += weight[row, column]
19             elif orientation[row, column] > 310.0 and
orientation[row, column] < 360.0:
20                hist[7] += weight[row, column]
21     return hist

```

Kode Sumber 5.9 Fungsi *Histogram binning*

Berikut adalah penjelasan Kode Sumber 5.9 Fungsi *Histogram binning*

:

1. Baris pertama merupakan deklarasi dari fungsi `histogram_binning` dengan parameternya berupa matriks orientasi dan matriks besaran gradien yang telah diberi bobot.
2. Baris ke-2 melakukan inisialisasi variabel `hist` yang merupakan histogram dari SIFT. Variabel ini bernilai 0 dengan panjang 8.
3. Baris ketiga hingga ke-20 merupakan perulangan untuk setiap elemen pada baris dan kolom dari matriks orientasi. Operasi yang berjalan ini memetakan setiap besaran gradien yang telah diberi bobot dengan nilai orientasi pada baris dan kolom tersebut. Nilai percabangan yang digunakan tersebut memberi batas pada nilai orientasi masuk ke dalam bin pada histogram.
4. Baris ke-21 mengembalikan nilai histogram dengan 8-bin.

### 5.3.2.2 MCWLD

Inisialisasi MCWLD

```

1 import numba
2 import numpy as np
3 import cv2
4 import asyncio
5 class WLD:
6     def __init__(self, region_size=16, displacement_size=9,
7 M=6, T=8, S=3):
8         self.__set_of_sliding_windows = []
9         self.__orientation_quantitaion_list = []
10        self.__differential_excitation_list = []
11        self.__T = T
12        self.__S = S
13        self.__M = M
14        self.__region_size = region_size
15        self.__displacement_size = displacement_size
16        self.__set_window_size = [3, 5, 7]
17        self.loop = asyncio.get_event_loop()
18        self.__set_histogram_bin_interval()
19        self.__s = np.arange(self.__S)

```

Kode Sumber 5.10 Inisialisasi MCWLD

Berikut adalah penjelasan dari Kode Sumber 5.10 Inisialisasi MCWLD

:

1. Baris pertama hingga ke-4 merupakan bagian untuk mengimpor *histogram binning*. Adapun *histogram binning* yang digunakan sama seperti yang tertulis pada subbab 5.1.2
2. Baris ke-5 hingga ke-19 merupakan inisialisasi variabel yang digunakan pada MCWLD. Variabel `set_of_sliding_window` merupakan list yang menyimpan kumpulan dari *sliding window*. Variabel ini berfungsi sama dengan milik SIFT. `orientation_quantitaion_list` menyimpan nilai hasil *quantization* orientasi. `differential_excitation_list` menyimpan seluruh nilai dari *differential excitation*. Variabel T merupakan banyaknya orientasi dominan yang diperoleh dari ekstraksi orientasi. Variabel S merupakan banyaknya rentang bin dari satu sub-histogram. Variabel M adalah banyaknya pembagi rentang nilai *differential excitation*. `set_window_size` menandakan ukuran *window* kecil yang digunakan pada MCWLD. Variabel `loop` digunakan agar ekstraksi fitur dapat dilakukan secara asinkronus.

Ekstraksi Fitur

```

1 def WLD_feature_extraction(self, image):
2     if np.ndim(image) == 2:
3         self.__image = image
4         self.__rows, self.__columns = self.__image.shape
5     elif np.ndim(image) == 3:
6         self.__image = U.grayscale(image)
7         self.__rows, self.__columns = self.__image.shape
8     self.__sliding_window_feature()
9     output_volume = []
10    try:
11        for sliding_window in self.__set_of_sliding_windows:
12            hst1, hst2, hst3 = self.loop.run_until_complete(
self.__inner_feature_extractor_with_asyncio(sliding_window))
13            output_volume.append(hst1.result() +
hst2.result() + hst3.result())
14    except Exception as e:
15        print("Error: ", e)
16    finally:
17        self.__set_of_sliding_windows.clear()
18        self.loop.close()
19        self.loop = asyncio.new_event_loop()
20    return output_volume

```

Kode Sumber 5.11 Fungsi Ekstraksi Fitur MCWLD

Berikut adalah penjelasan dari Kode Sumber 5.11 Fungsi Ekstraksi Fitur MCWLD

:

1. Baris pertama merupakan deklarasi dari fungsi MCWLD dengan parameternya adalah citra
2. Baris kedua hingga ke-8 merupakan inialisasi variabel yang akan digunakan pada proses ekstraksi, mulai dari mengubah citra menjadi citra abu-abu, mengambil panjang sumbu x dan y serta melakukan pemotongan *sliding window* sesuai dengan ukuran region atau *window* dan besaran perpindahannya.
3. Baris ke-9 terdapat inialisasi variabel *output\_volume* yang digunakan untuk menyimpan nilai hasil ekstraksi fitur dari ketiga jenis radius.
4. Baris ke-10 hingga ke-20 merupakan sebuah perulangan yang bersifat asynchronous sehingga untuk setiap radius, ekstraksi fiturnya dapat dilakukan secara paralel. Fungsi dari *try-except* digunakan untuk menangkap error yang terjadi saat operasinya dijalankan.

#### Gradient Orientation

```

1 @numba.jit
2 def __circular_gradient_orientation(self, window):
3     y, x = window.shape
4     temp = np.int16(window)
5     filter_v10 = (temp[y - 1, int(x / 2)] - temp[0, int(x /
6     2)])
7     filter_v11 = (temp[int(y / 2), 0] - temp[int(y / 2), x -
8     1])
9     derived_delta = np.arctan2(filter_v11, filter_v10) +
10    np.pi

```

```

9     t = np.mod(np.floor((derived_delta / (2 * np.pi /
10 self.__T)) + (1 / 2)), self.__T)
11     dominant_orientation = (2 * t * np.pi) / self.__T
    self.__orientation_quantitaion_list.append(
        dominant_orientation)

```

**Kode Sumber 5.12 Fungsi *Gradient Orientation***

Berikut adalah penjelasan dari Kode Sumber 5.12 Fungsi *Gradient Orientation*

:

1. Baris pertama dan kedua merupakan instansiasi numba.jit serta deklarasi fungsi ekstraksi orientasi gradien
2. Baris ketiga hingga ke-7 adalah bagian untuk mengekstraksi nilai orientasi. Hal ini bekerja dengan mengambil titik tengah pada sumbu y dibaris pertama yang dikurangi dengan baris terakhir untuk menghasilkan fungsi y, kemudian mengambil titik tengah pada sumbu x dibaris pertama yang dikurangi dengan baris terakhir untuk menghasilkan fungsi x. Kedua fungsi ini akan menghasilkan nilai orientasi dari fungsi arctan.
3. Baris ke-8 hingga ke-11 merupakan proses *quantization* nilai radian dari *Gradient Orientation*. Fungsi dari method ini adalah mengubah nilai radian tersebut menjadi sebuah nilai dengan rentang tertentu. Rentang ini disesuaikan dengan besaran dari variabel T. Proses yang dilakukan ini sama dengan proses melakukan mapping terhadap *Gradient Orientation*.

```

Differential excitation
1 @numba.jit
2 def __differential_excitation(self, window):
3     y, x = window.shape
4     number_of_neighborhood = (y * x) - 1
5     if number_of_neighborhood <= 0:
6         pass
7     else:
8         xc = window[int(y / 2), int(x / 2)]
9         temp1 = (window[0, 0:x].astype(np.float64) + 1) -
              (xc - number_of_neighborhood)
10        temp1 = temp1 / xc
11        temp2 = (window[1:y, 0::x - 1].astype(np.float64) +
12)              - (xc - number_of_neighborhood)
13        temp2 = temp2 / xc
14        temp3 = (window[y - 1, 0:x].astype(np.float64) + 1)
              - \
              (xc - number_of_neighborhood)
14        temp3 = temp3 / xc
15        sum_of_differential = temp1.sum() + temp2.sum() +
              temp3.sum()
16        self.__differential_excitation_list.append(
              np.arctan(sum_of_differential))

```

**Kode Sumber 5.13 Fungsi *Differential excitation***

Berikut adalah penjelasan dari Kode Sumber 5.13 Fungsi *Differential excitation*

:

1. Baris pertama dan kedua merupakan instansiasi numba.jit serta deklarasi fungsi ekstraksi `differential_excitation`.
2. Baris ketiga hingga keempat merupakan inialisasi variabel `y` dan `x` yang didapat dari panjang sumbu `y` dan `x` dari `window` serta `number_of_neighborhood` merupakan jumlah *pixel* tetangga dari titik tengah.
3. Baris kelima hingga ke 16, jika jumlah *pixel* tetangganya kurang dari sama dengan 0, maka operasi ini tidak akan dilakukan. Sebaliknya, maka fungsi ini akan mencari nilai titik tengah dari `window` yang diberikan. Kemudian melakukan pengurangan antara baris pertama dengan titik tengah yang disimpan pada `temp1`, mengurangi baris kedua hingga satu baris sebelum terakhir pada kolom pertama dan kolom terakhir dengan titik tengah, serta mengurangi baris terakhir dengan tengah. Nilai pengurangan tersebut kemudian dijumlahkan dan diubah menjadi radian menggunakan fungsi `arctan`.

Pembentukan histogram 1D	
1	<code>def __histogram_result_build(self):</code>
2	<code>__hist_2D = [[np.zeros(self.__S) for t in</code>
	<code>range(len(self.__T_interval))] for m in</code>
	<code>range(len(self.__M_interval))]</code>
3	<code>for j, (diff_exc, quant_or) in enumerate(</code>
	<code>zip(self.__differential_excitation_list,</code>
	<code>self.__orientation_quantitaion_list)):</code>
4	<code>for m, (lb, ub) in enumerate(self.__M_interval):</code>
5	<code>if diff_exc &gt; lb and diff_exc &lt; ub:</code>
6	<code>t = self.__T_interval.index(quant_or)</code>
7	<code>Sj = np.floor(((diff_exc - lb) / ((ub - lb)</code>
	<code>/ self.__S)) + 0.5)</code>
8	<code>if np.isin(self.__s, int(Sj)).any():</code>
9	<code>__hist_2D[m][t][int(Sj)] += 1</code>
10	<code>res_hist = []</code>
11	<code>for M in range(len(__hist_2D)):</code>
12	<code>h_M = []</code>
13	<code>for T in range(len(__hist_2D[M])):</code>
14	<code>h_M += list(__hist_2D[M][T])</code>
15	<code>res_hist += h_M</code>
16	<code>return res_hist</code>

**Kode Sumber 5.14 Pembentukan histogram 1D**

**Berikut adalah penjelasan dari Kode Sumber 5.14 Pembentukan histogram 1D**

:

1. Baris pertama merupakan deklarasi dari fungsi `histogram_result_build`
2. Baris kedua adalah inialisasi variabel `hist_2D` yang merupakan representasi dari histogram 2D dimana setiap kolomnya merupakan

- interval dari setiap  $T$  dominant orientation dan kolomnya terdiri dari  $M$  segmen yang masing-masing segmennya berisi  $S$ -jumlah bin
3. Baris ke-3 hingga ke-9 merupakan proses *histogram binning*. Proses ini memasukkan setiap nilai *differential excitation* sesuai dengan nilai *quantization* orientasinya. Nilai *differential excitation* akan menemui segmennya dan disesuaikan dengan nilai binnya yang dihitung pada baris ke-7 hingga ke 9.
  4. Baris ke-10 hingga ke-16 ada proses penggabungan histogram 2D menjadi histogram 1D. Langkah yang dilakukan adalah membagi histogram 2D sebanyak  $M$  segmen. Di masing-masing segment, dilakukan penggabungan antar orientasi dari yang paling rendah. Terakhir, setiap segmen digabung menjadi satu. Hasil akhirnya tersimpan pada variabel `res_hist`



## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini yang dibahas adalah pengujian yang dilakukan merujuk pada perancangan skenario pengujian yang dirangkum pada bab 4. Pengujian ini ditujukan sebagai bahan untuk menjawab rumusan masalah yang dilanjut dengan analisisnya. Pengujian ini dilakukan terhadap 188 orang dengan masing-masing orang satu sketsa dan satu foto profil. Hasil dari pengujian ini akan dianalisis untuk mendapatkan sebuah korelasi antara data dan metode yang digunakan dalam penelitian ini.

### 6.1 Pengujian

Skenario pengujian yang digunakan merujuk pada perancangan pada Bab 4, yakni:

1. Pengujian *overlapping* dan *non-overlapping*
2. Pengujian parameter pada SIFT dan MCWLD
3. Pengujian fusi SIFT dan MCWLD

Dari kedua pengujian tersebut, masing-masing mempunyai tujuannya sendiri. Skenario pengujian pertama adalah menguji masing-masing parameter yang digunakan pada SIFT dan MCWLD. Adapun parameter yang dapat diubah pada SIFT adalah ukuran *window* kecilnya dan pada MCWLD adalah kombinasi nilai *T-dominant orientation*, *M-Segment* dan *S-Bins*. Nilai yang diuji pada pengujian ini adalah nilai *true positive rate* dari masing-masing kondisi.

Skenario pada pengujian kedua mencari konfigurasi terbaik antara *overlapping* dan *non-overlapping*, baik pada metode SIFT, MCWLD maupun kombinasinya. Adapun ukuran *window* yang digunakan pada pengujian ini adalah  $16 \times 16$  *pixel* dan besaran perpindahan yang dilakukan sebesar  $9$  *pixel* untuk kondisi *overlapping window*. Nilai yang diuji pada pengujian ini adalah nilai *true positive rate* dari masing-masing kondisi. Hasil pada pengujian ini ditampilkan pada subbab 6.1.2 selanjutnya dalam bentuk tabel dan grafik CMC (*Cumulative Match Count*).

Skenario pada pengujian ketiga adalah mencari metode terbaik yang pada kasus identifikasi pelaku penjahat menggunakan sketsa wajah. Pengujian ketiga ini menggunakan konfigurasi terbaik dari pengujian sebelumnya. Hasil dari pengujian ini ditampilkan dalam bentuk tabel serta grafik CMC (*Cumulative Match Count*).

Masing-masing dari skenario pengujian ini akan dianalisis pada subbab 6.1.4 Analisis Hasil Pengujian yang dibahas per metode yang digunakan mengacu pada hasil pengujian yang dilakukan pada subbab ini. Adapun yang dilakukan pada subbab ini hanya menguji serta memberikan pengamatan awal sebagai bahan untuk analisis kemudian.

### 6.1.1 Pengujian Pengaruh *Overlapping* dan *Non-overlapping*

Pengujian pengaruh *overlapping* dan *non-overlapping* pada penelitian ini dilakukan terhadap hasil dari pengujian parameter yang dilakukan pada pengujian sebelumnya. Pada pengujian ini, *window* yang digunakan sebagai patch berukuran 16 x 16 *pixel* dan 32 x 32 *pixel*. Untuk besaran perpindahan *pixel* pada konfigurasi *overlapping* dilakukan sebesar 9 *pixel* untuk ukuran *window* 16 x 16 *pixel* dan 18 *pixel* untuk ukuran 32 x 32 *pixel*. Dalam pengujian ini, ukuran *window* 16 x 16 memiliki patch atau *window* berjumlah 567 untuk konfigurasi *overlapping* dan 180 untuk *non-overlapping*. Adapun hasil yang diberikan adalah sebagai berikut:

**Tabel 6.1 Pengaruh *Overlapping* dan *non-overlapping* pada metode MCWLD**

Ukuran <i>window</i>	<i>Overlapping</i>	Rank				
		1	5	10	25	50
16	Ya ( <i>baseline</i> )	52.13%	80.32%	87.23%	92.02%	95.74%
	Tidak	51.06%	79.79%	87.23%	92.02%	95.21%
32	Ya	35.11%	63.83%	78.19%	83.51%	87.77%
	Tidak	21.81%	53.19%	64.89%	80.32%	87.23%

Pengamatan awal dari pengaruh konfigurasi *sliding window* pada metode MCWLD menunjukkan bahwa pengaruh dari *non-overlapping* memperburuk nilai *identification rate* yang dihasilkan. Nilai tersebut juga menurun saat ukuran *window* yang digunakan semakin besar. Hal tersebut menunjukkan kondisi *sliding window* dengan ukuran *window* 16 x 16 *pixel* dan *overlapping* lebih baik dari konfigurasi yang lain. Konfigurasi ini sendiri merupakan *baseline* yang digunakan oleh Klare (Klare, et al., 2011). Pengamatan awal ini kemudian akan dijelaskan pada bagian analisis.

**Tabel 6.2 Pengaruh *overlapping* dan *non-overlapping* pada metode SIFT**

Ukuran <i>window</i>	<i>Overlapping</i>	Rank				
		1	5	10	25	50
16	Ya ( <i>baseline</i> )	45.21%	87.77%	90.96%	98.40%	100.00%
	Tidak	39.89%	82.98%	91.49%	94.68%	98.40%
32	Ya	79.79%	98.40%	99.47%	100.00%	100.00%
	Tidak	53.19%	91.49%	94.68%	98.94%	99.47%

Pengamatan awal dari pengaruh konfigurasi *sliding window* pada metode SIFT menunjukkan bahwa pengaruh dari *non-overlapping* juga memperburuk nilai *identification rate* yang dihasilkan seperti pada metode MCWLD. Namun berbeda dengan metode MCWLD bahwa metode SIFT jika ukuran *window* yang digunakan



semakin besar, nilai *identification rate* yang dihasilkan semakin besar juga. Hal tersebut menunjukkan kondisi *sliding window* dengan ukuran *window* 32 x 32 *pixel* dan *overlapping* lebih baik dari konfigurasi yang lain. Konfigurasi yang dijadikan *baseline* adalah pada kondisi *overlapping* di ukuran *window* 16 x 16 *pixel*. Pengamatan awal ini selanjutnya akan dijelaskan pada bagian analisis.

### 6.1.2 Pengujian Parameter pada SIFT dan MCWLD

Pengujian parameter untuk metode SIFT dilakukan pada ukuran *window* kecilnya. Pengujian ini dilakukan dengan menguji beberapa nilai, yakni 4 dan 8. Nilai parameter yang digunakan oleh David Lowe (Lowe, 2004) adalah 4 dan nilai ini dijadikan sebagai *baseline*. Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat *robust* dari metode SIFT pada ruang lingkup *Heterogeneous* dalam mengekstraksi nilai tekstur. Hasil dari pengujian ini dapat dilihat pada Tabel 6.5.

**Tabel 6.3 Pengujian Parameter SIFT pada konfigurasi *window* 16 dan *overlapping* (Baseline)**

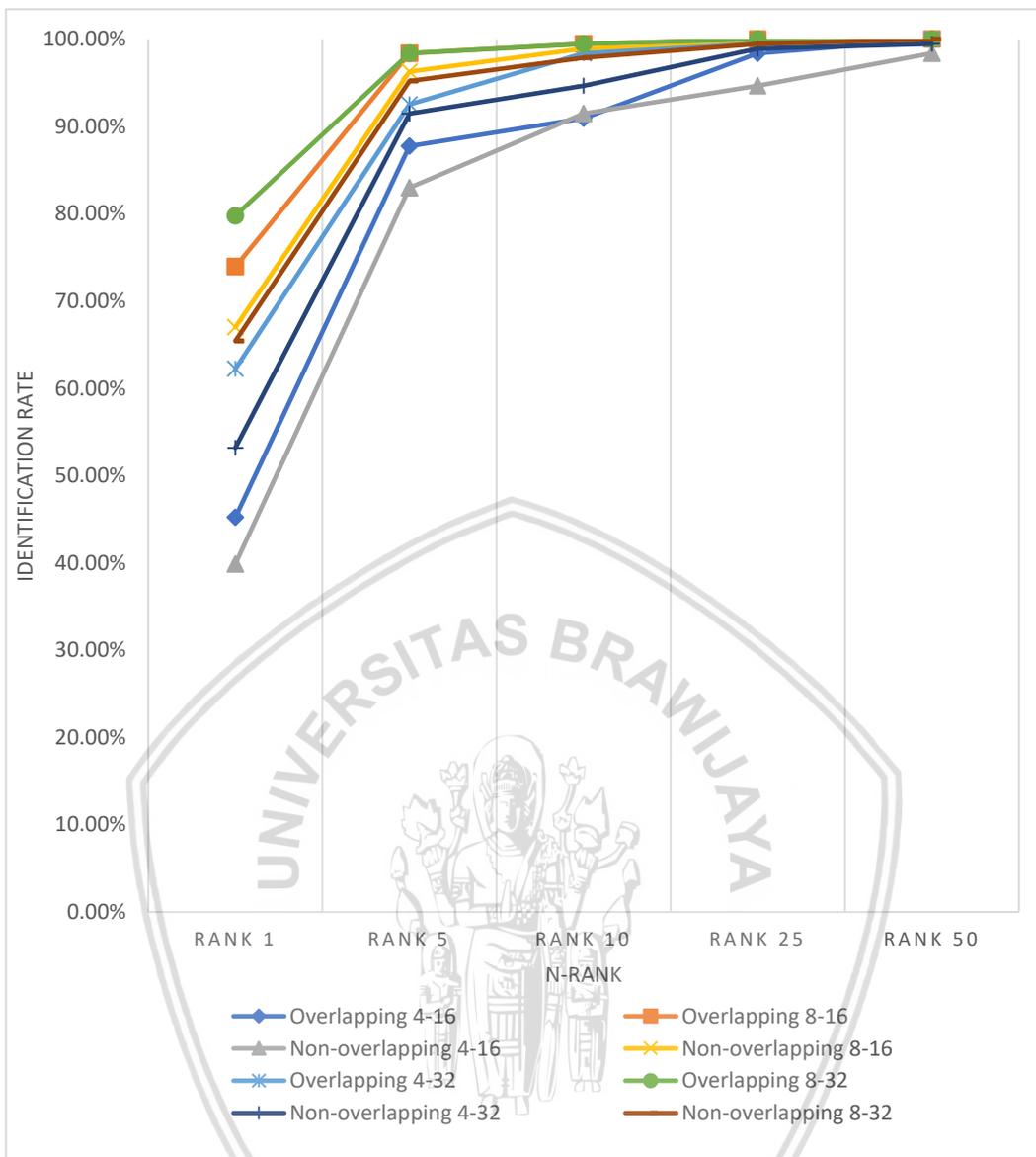
Parameter ( <i>Ukuran window</i> )	Rank				
	1	5	10	25	50
4 ( <i>baseline</i> )	26.60%	74.77%	80.85%	94.14%	100%
8	69.68%	97.34%	98.93%	100%	100%

Pengamatan awal yang dapat dilihat dari Tabel 6.3 menunjukkan bahwa ukuran *window* kecil pada metode SIFT mempengaruhi hasil yang diberikan. Secara keseluruhan, berikut adalah tabel pengujian yang dilakukan pada semua konfigurasi *sliding window*. Keterangan ukuran *window* disimbolkan dengan huruf "K", *overlapping* dengan huruf "O", dan parameter yang digunakan dengan huruf "P".

**Tabel 6.4 Pengujian Parameter SIFT pada semua konfigurasi *sliding window***

K	O	P	Rank				
			1	5	10	25	50
16	Ya	4	45.21%	87.77%	90.96%	98.40%	100.00%
		8	73.94%	98.40%	99.47%	100.00%	100.00%
	Tidak	4	39.89%	82.98%	91.49%	94.68%	98.40%
		8	67.02%	96.28%	98.94%	99.47%	100.00%
32	Ya	4	62.23%	92.55%	98.40%	99.47%	100.00%
		8	79.79%	98.40%	99.47%	100.00%	100.00%
	Tidak	4	53.19%	91.49%	94.68%	98.94%	99.47%
		8	65.43%	95.21%	97.87%	99.47%	100.00%





**Gambar 6.1** Grafik CMC SIFT pengujian parameter dan *sliding window*

Berdasarkan dari Gambar 6.1 dapat diambil kesimpulan bahwa ada kecenderungan bertambahnya nilai *identification rate* saat ukuran *window* kecil yang digunakan semakin besar. Dapat terlihat dengan jelas bahwa konfigurasi *overlapping* dengan ukuran *window* 32 dan ukuran *window* kecil 8 memiliki nilai *identification rate* yang lebih baik dibanding konfigurasi lain pada *rank-1* atau *top-match*. Dalam beberapa kondisi, pada *rank-25* metode SIFT sudah menghasilkan nilai sempurna dalam mengidentifikasi individu berdasarkan sketsa wajah.

Pada metode MCWLD, pengujian parameternya dilakukan terhadap kombinasi nilai *T-dominant orientation*, *M-Segment* dan *S-Bins*. Nilai *T-dominant orientation*, *M-Segment* dan *S-Bins* dari penulisnya adalah  $T=8$ ,  $M=6$  dan  $S=3$ . Sama seperti pada metode SIFT, nilai tersebut dijadikan sebagai *baseline*. Nilai yang digunakan sebagai parameter yang diuji adalah  $T=8$ ,  $M=4$ ,  $S=4$  dan  $T=8$ ,  $M=3$ ,  $S=3$ .



Nilai-nilai tersebut mempengaruhi hasil yang diberikan karena nilai tersebut bertanggung jawab atas pembentukan histogram dari metode MCWLD ini. Tujuan dilakukannya pengujian adalah untuk mengetahui kombinasi dari ketiga parameter tersebut dalam membentuk representasi sebuah fitur ciri. Hasil dari pengujian ini dapat dilihat pada Tabel 6.5.

**Tabel 6.5 Pengujian Parameter MCWLD pada konfigurasi *window* 16 dan *overlapping* (Baseline)**

Parameter (T-M-S)	Rank				
	1	5	10	25	50
8-6-3 (Baseline)	52.13%	80.32%	87.23%	92.02%	95.74%
8-6-4	35.11%	67.55%	76.60%	83.51%	92.02%
8-4-3	82.45%	92.02%	92.02%	94.15%	94.15%
8-4-4	62.77%	87.23%	90.96%	93.62%	97.87%
6-6-3	55.85%	82.45%	87.23%	92.55%	95.21%
6-6-4	44.15%	71.81%	78.72%	85.64%	92.02%
6-4-3	81.91%	92.55%	93.09%	93.62%	94.15%
6-4-4	64.89%	87.77%	90.43%	94.68%	96.28%

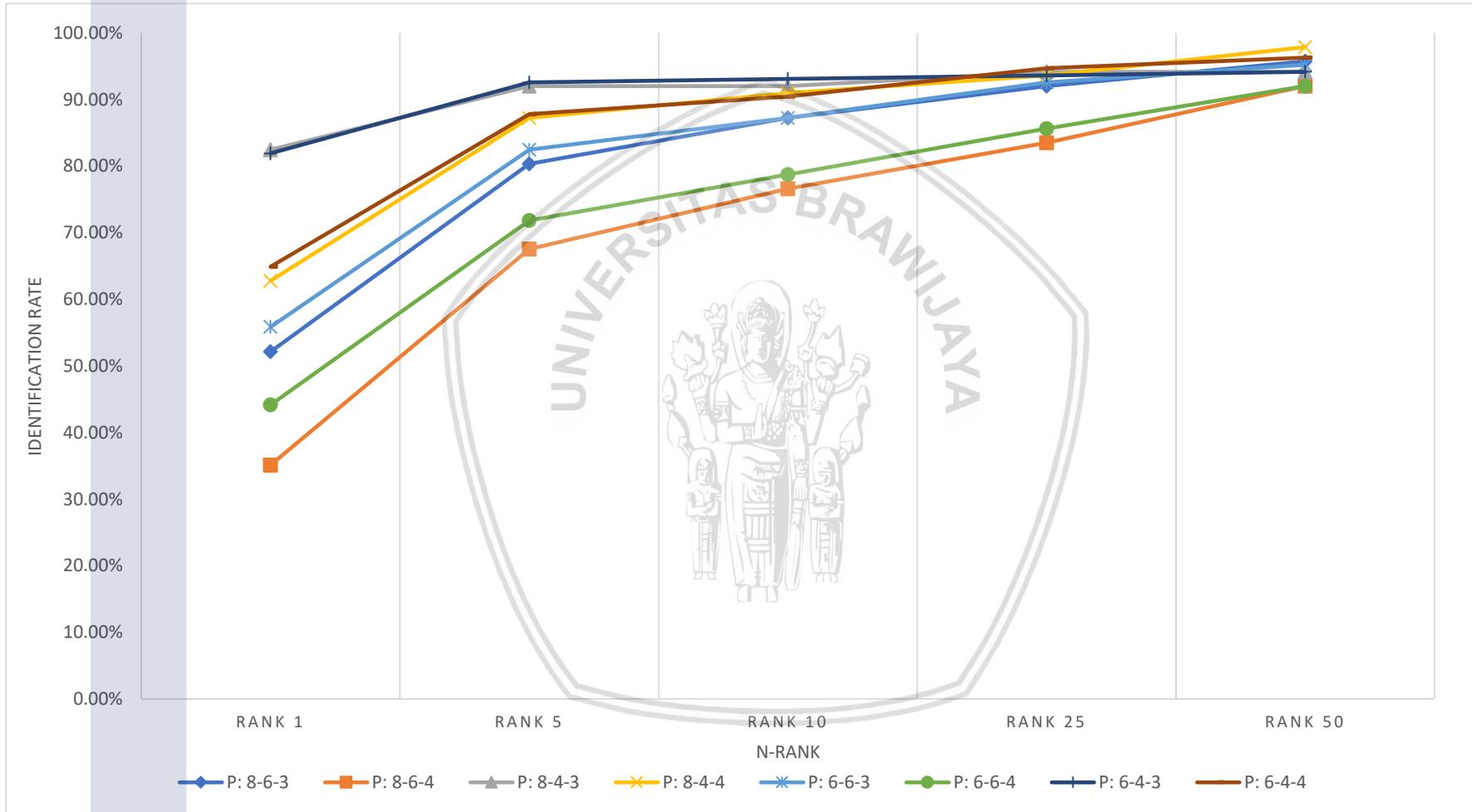
Pengamatan awal dari pengujian parameter tersebut menunjukkan perubahan pada nilai *M-Segment* dapat meningkatkan nilai *identification rate* serta jika merubah nilai *T-dominant orientation* dapat mengakibatkan berturunnya nilai *identification rate* namun tidak signifikan. Tidak seperti merubah nilai *S-bin* yang berdampak sangat buruk pada nilai *identification rate* yang dihasilkan. Dari pengamatan awal ini kemudian akan dianalisis pada subbab selanjutnya. Pada bagian metode MCWLD. Pengamatan awal ini hanya membandingkan dengan kondisi parameter yang dijelaskan oleh Chen (Chen, et al., 2010) sebagai *baseline* dari pengujian ini. Secara keseluruhan tabel pengujian pada kondisi *overlapping* dan *non-overlapping* ditunjukkan pada tabel berikut dengan keterangan sama seperti pada Tabel 6.4.

Berdasarkan pada Gambar 6.2, Gambar 6.3, Gambar 6.4 dan Gambar 6.5 menunjukkan bahwa terdapat kecenderungan perubahan *identification rate* jika terdapat perubahan pada nilai *M-Segment*. Jika dibandingkan dengan kondisi *baseline* yang digunakan, nilai *identification rate* yang diberikan lebih kecil dibandingkan dengan perubahan nilai *M-Segment*. Pengamatan awal dari gambar tersebut menunjukkan pula bahwa perubahan dari jumlah *T-dominant orientation* memiliki dampak menaiknya nilai *identification rate*. Dari pengamatan awal ini juga bahwa mengubah nilai *S-bins* tidak dianjurkan karena akan menurunkan nilai *identification rate* secara signifikan.

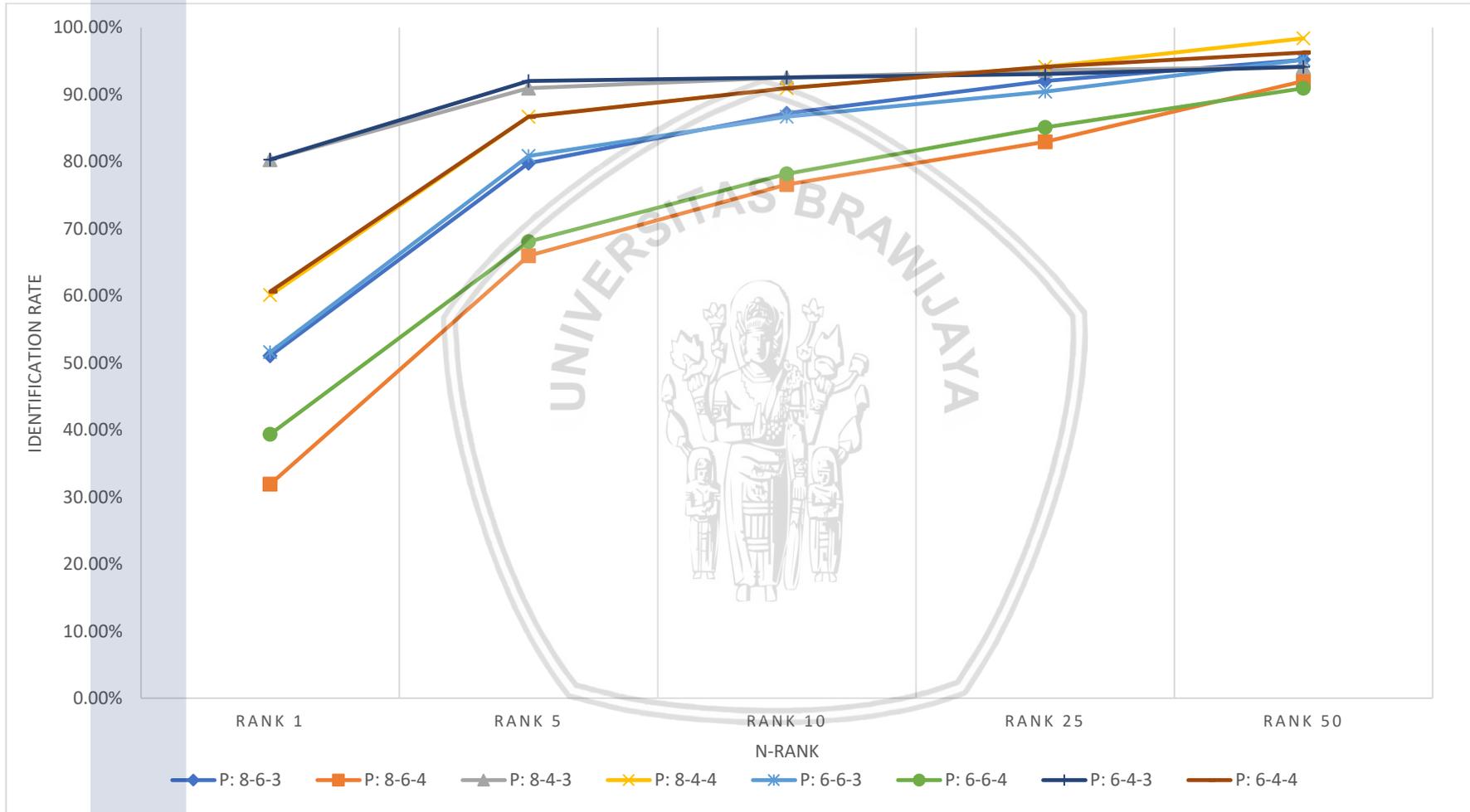
Tabel 6.6 Pengujian Parameter MCWLD pada semua konfigurasi *sliding window*

K	O	P (T-M-S)	Rank				
			1	5	10	25	50
16	Ya	8-6-3	52.13%	80.32%	87.23%	92.02%	95.74%
		8-6-4	35.11%	67.55%	76.60%	83.51%	92.02%
		8-4-3	82.45%	92.02%	92.02%	94.15%	94.15%
		8-4-4	62.77%	87.23%	90.96%	93.62%	97.87%
		6-6-3	55.85%	82.45%	87.23%	92.55%	95.21%
		6-6-4	44.15%	71.81%	78.72%	85.64%	92.02%
		6-4-3	81.91%	92.55%	93.09%	93.62%	94.15%
		6-4-4	64.89%	87.77%	90.43%	94.68%	96.28%
	Tidak	8-6-3	51.06%	79.79%	87.23%	92.02%	95.21%
		8-6-4	31.91%	65.96%	76.60%	82.98%	92.02%
		8-4-3	80.32%	90.96%	92.55%	93.62%	94.15%
		8-4-4	60.11%	86.70%	90.96%	94.15%	98.40%
		6-6-3	51.60%	80.85%	86.70%	90.43%	95.21%
		6-6-4	39.36%	68.09%	78.19%	85.11%	90.96%
		6-4-3	80.32%	92.02%	92.55%	93.09%	94.15%
		6-4-4	60.64%	86.70%	90.96%	94.15%	96.28%
32	Ya	8-6-3	35.11%	63.83%	78.19%	83.51%	87.77%
		8-6-4	30.32%	56.91%	68.09%	81.91%	85.11%
		8-4-3	59.57%	80.32%	86.70%	89.89%	93.09%
		8-4-4	40.43%	70.74%	78.72%	86.70%	92.02%
		6-6-3	32.45%	65.43%	77.66%	84.04%	87.77%
		6-6-4	31.91%	56.91%	66.49%	80.32%	85.11%
		6-4-3	60.11%	79.26%	87.23%	91.49%	93.09%
		6-4-4	35.64%	68.62%	77.66%	86.70%	90.43%
	Tidak	8-6-3	21.81%	53.19%	64.89%	80.32%	87.23%
		8-6-4	17.55%	45.74%	56.91%	73.40%	82.98%
		8-4-3	51.06%	73.40%	81.38%	88.30%	92.02%
		8-4-4	25.53%	60.11%	69.15%	84.04%	89.89%
		6-6-3	21.28%	52.66%	63.30%	80.32%	87.23%
		6-6-4	18.09%	47.34%	56.38%	72.34%	83.51%
		6-4-3	50.00%	73.94%	81.91%	89.89%	92.02%
		6-4-4	21.81%	56.91%	68.09%	84.04%	89.36%

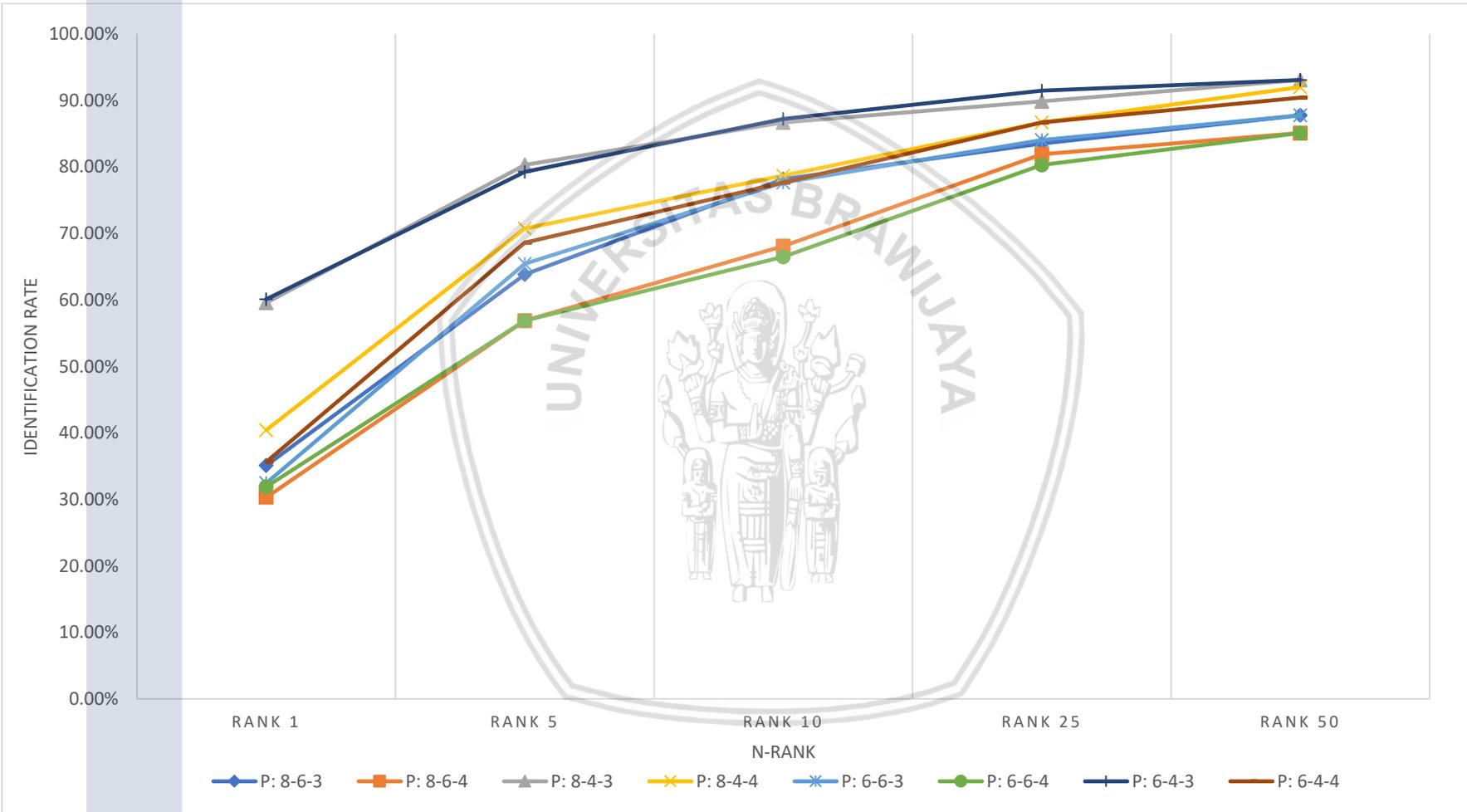




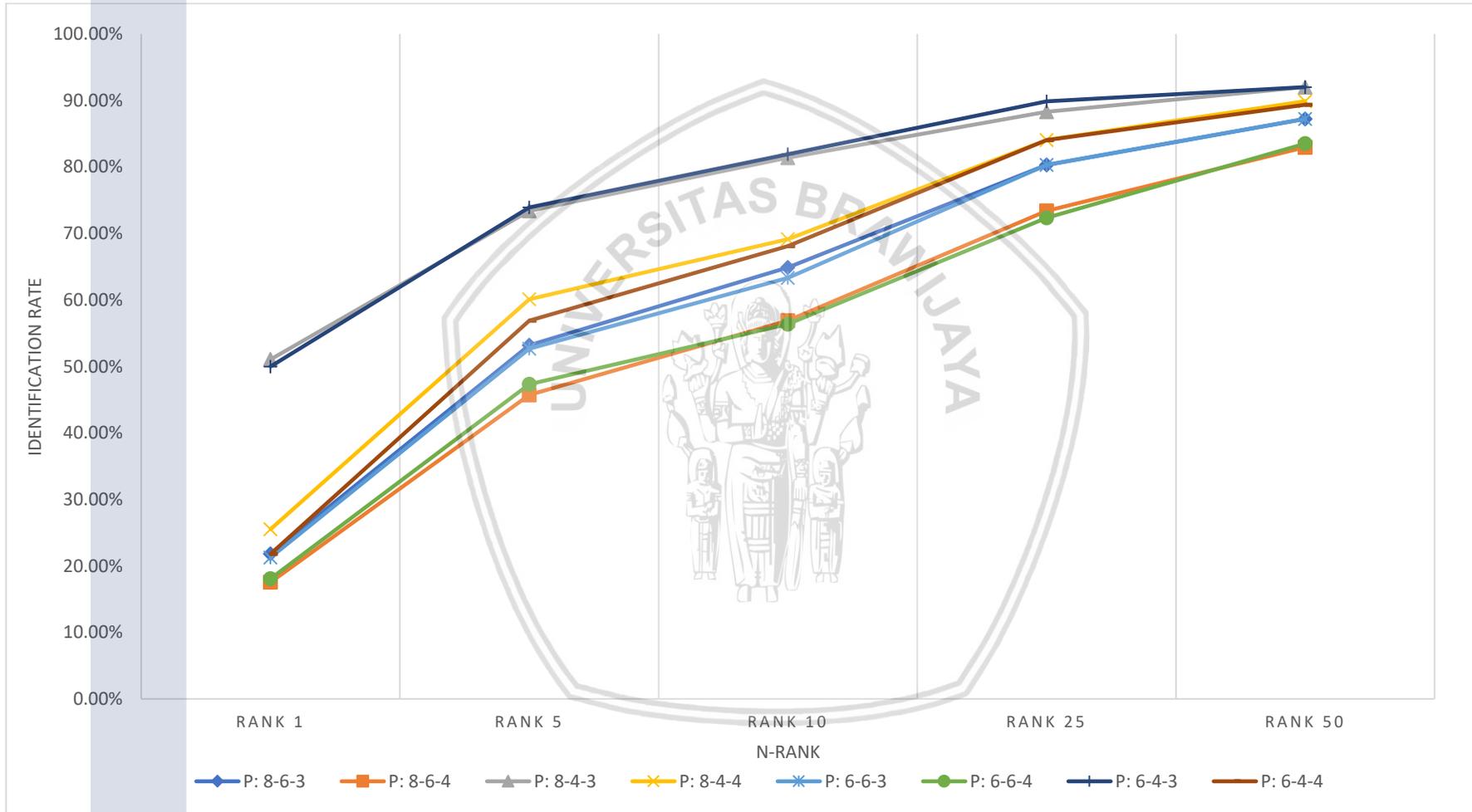
Gambar 6.2 Grafik CMC MCWLD Sliding window Overlapping 16



Gambar 6.3 Grafik CMC MCWLD Sliding window Non-overlapping 16



Gambar 6.4 Grafik CMC MCWLD Sliding window Overlapping 32



Gambar 6.5 Grafik CMC MCWLD Sliding window Non-overlapping 32

### 6.1.3 Pengujian Fusi SIFT dan MCWLD

Pengujian ketiga ini melihat nilai *identification rate* dari kedua metode yang diusulkan. Nilai yang dibandingkan berdasarkan konfigurasi *sliding window* dan parameter dari pengujian sebelumnya. Kemudian untuk metode SIFT+MCWLD dilakukan dengan mengombinasikan nilai fitur SIFT dan MCWLD pada ukuran dan konfigurasi *sliding window* yang sama. Fusi pada tingkat fitur dilakukan pada konfigurasi *sliding window* yang berbeda tidak dapat dilakukan karena jumlah dari *window* yang dihasilkan berbeda. Sedangkan pada tingkat *score* dapat dilakukan pada semua konfigurasi. Keterangan pada Tabel 6.7 Pengujian Fusi Tingkat Fitur SIFT dan MCWLD pada semua konfigurasi *sliding window* dan parameter antara lain adalah: K (Konfigurasi ukuran *sliding window*), O (konfigurasi *overlapping* maupun tidak), Ps (Parameter untuk SIFT) dan Pm (Parameter MCWLD). Untuk tingkat *score* ada pada lampiran.

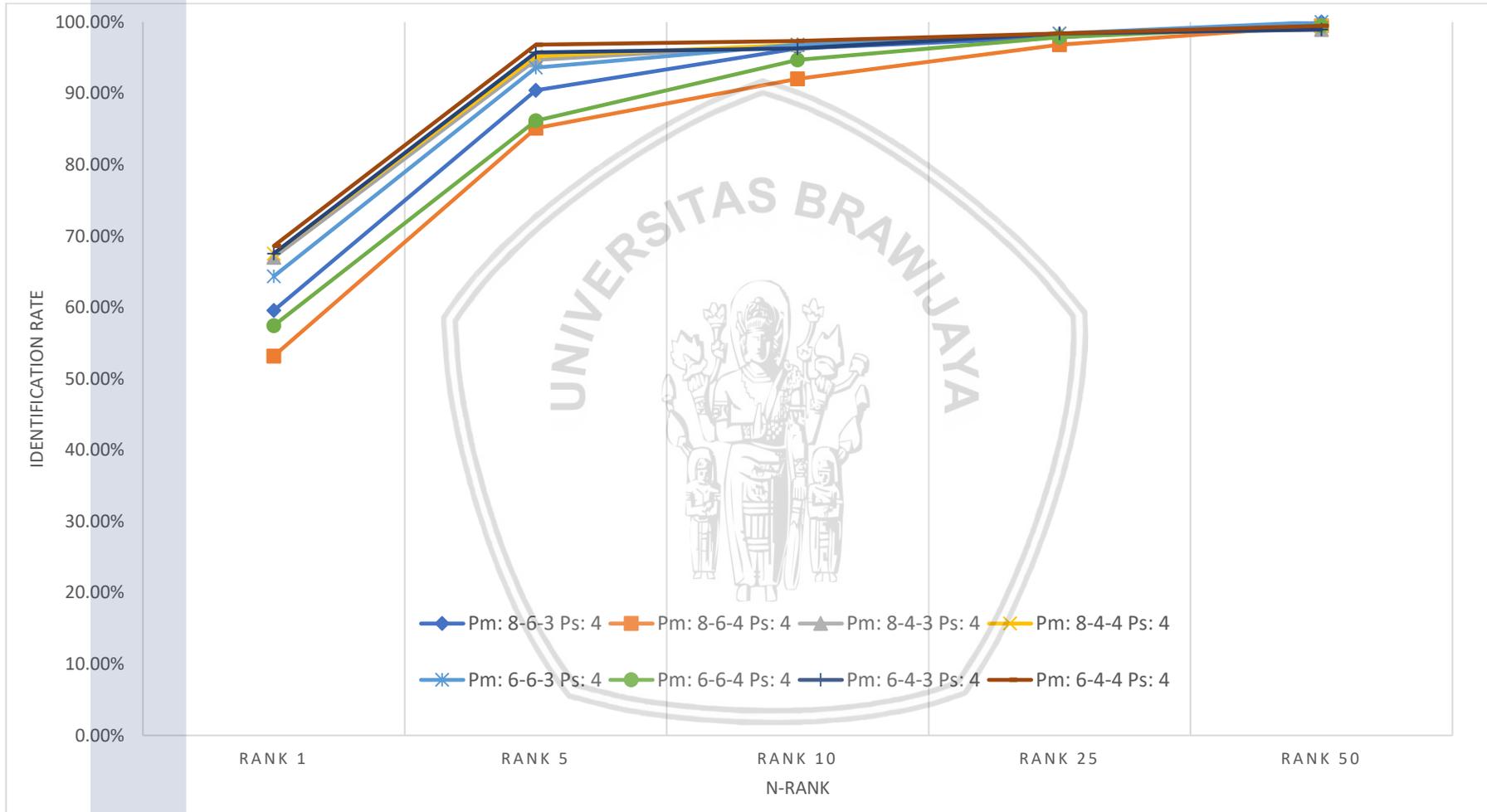
**Tabel 6.7 Pengujian Fusi Tingkat Fitur SIFT dan MCWLD pada semua konfigurasi *sliding window* dan parameter**

K	O	Ps	Pm	Rank				
				1	5	10	25	50
16	Ya	4	8-6-3	59.57%	90.43%	96.28%	97.87%	100.00%
			8-6-4	53.19%	85.11%	92.02%	96.81%	99.47%
			8-4-3	67.02%	94.68%	96.81%	98.40%	98.94%
			8-4-4	67.55%	95.21%	96.81%	98.40%	99.47%
			6-6-3	64.36%	93.62%	96.81%	98.40%	100.00%
			6-6-4	57.45%	86.17%	94.68%	97.87%	99.47%
			6-4-3	67.55%	95.74%	96.28%	98.40%	98.94%
			6-4-4	68.62%	96.81%	97.34%	98.40%	99.47%
		8	8-6-3	61.17%	88.83%	96.28%	98.40%	99.47%
			8-6-4	51.60%	79.26%	87.77%	96.28%	98.94%
			8-4-3	73.40%	92.02%	94.15%	97.34%	99.47%
			8-4-4	71.28%	94.68%	96.81%	98.40%	100.00%
			6-6-3	60.11%	90.96%	95.74%	98.94%	100.00%
			6-6-4	57.45%	81.91%	87.23%	96.28%	98.94%
			6-4-3	74.47%	93.09%	94.15%	95.21%	98.94%
			6-4-4	70.21%	96.28%	98.40%	99.47%	100.00%
	Tidak	4	8-6-3	54.26%	89.89%	93.09%	98.40%	100.00%
			8-6-4	45.21%	80.85%	90.96%	96.81%	98.40%
			8-4-3	55.85%	90.96%	94.15%	98.40%	99.47%
			8-4-4	57.45%	93.09%	95.74%	98.94%	100.00%
			6-6-3	53.72%	89.89%	95.21%	98.94%	99.47%
			6-6-4	48.40%	84.57%	92.02%	97.34%	98.94%
			6-4-3	55.85%	93.09%	95.21%	98.94%	99.47%
			6-4-4	55.85%	92.55%	96.28%	98.94%	100.00%
8		8-6-3	60.11%	87.23%	93.62%	98.40%	99.47%	

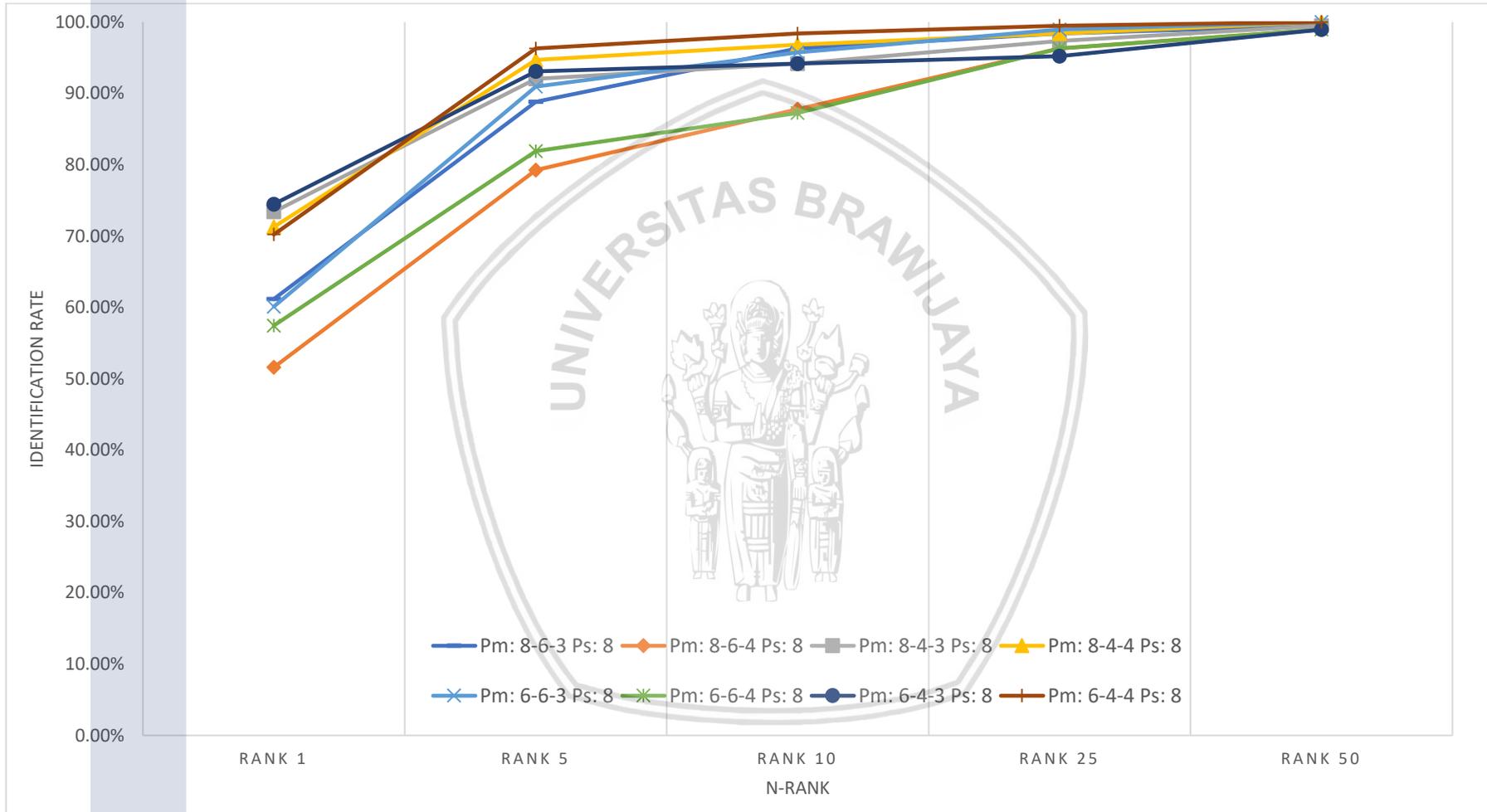


32	Ya	4	8-6-4	50.53%	77.13%	85.11%	94.68%	98.40%
			8-4-3	71.28%	91.49%	93.09%	96.28%	98.94%
			8-4-4	69.15%	95.74%	97.34%	98.40%	100.00%
			6-6-3	59.57%	88.83%	96.81%	98.40%	99.47%
			6-6-4	55.32%	80.32%	87.23%	96.28%	98.94%
			6-4-3	69.15%	93.09%	93.62%	95.74%	98.40%
			6-4-4	67.55%	96.28%	98.40%	98.94%	100.00%
		8-6-3	69.68%	91.49%	95.74%	99.47%	100.00%	
		8-6-4	65.43%	89.36%	94.15%	97.34%	100.00%	
		8-4-3	74.47%	94.15%	96.81%	99.47%	100.00%	
		8-4-4	72.34%	93.62%	97.34%	99.47%	100.00%	
		6-6-3	71.28%	92.02%	96.28%	99.47%	100.00%	
		6-6-4	66.49%	90.43%	94.68%	98.40%	100.00%	
		6-4-3	75.00%	95.21%	96.81%	99.47%	100.00%	
	6-4-4	73.40%	94.68%	97.34%	99.47%	100.00%		
	8	8-6-3	67.02%	87.77%	94.15%	100.00%	100.00%	
	8-6-4	58.51%	82.98%	88.30%	96.28%	99.47%		
	8-4-3	73.94%	93.09%	97.34%	99.47%	100.00%		
	8-4-4	73.40%	94.15%	99.47%	100.00%	100.00%		
	6-6-3	68.09%	90.43%	97.87%	100.00%	100.00%		
	6-6-4	59.57%	85.64%	92.02%	97.34%	100.00%		
	6-4-3	75.00%	94.15%	97.34%	99.47%	100.00%		
	6-4-4	71.28%	95.74%	98.94%	100.00%	100.00%		
	Tidak	4	8-6-3	59.57%	86.17%	93.62%	96.81%	98.94%
	8-6-4		55.85%	84.57%	91.49%	96.28%	98.94%	
	8-4-3		67.02%	90.43%	94.15%	97.34%	98.94%	
	8-4-4		63.30%	93.09%	94.68%	97.87%	99.47%	
	6-6-3		60.11%	87.23%	93.62%	96.81%	99.47%	
6-6-4	57.98%		86.17%	92.02%	96.28%	98.94%		
6-4-3	67.02%		92.02%	94.68%	97.34%	98.40%		
6-4-4	65.43%		92.55%	94.68%	97.87%	99.47%		
8	8-6-3		51.60%	78.19%	88.83%	96.28%	98.94%	
8-6-4	47.34%		71.81%	82.45%	93.09%	97.87%		
8-4-3	59.04%		89.36%	93.09%	96.28%	99.47%		
8-4-4	57.45%		86.17%	94.68%	98.94%	99.47%		
6-6-3	51.06%		80.85%	90.43%	96.81%	100.00%		
6-6-4	48.94%		73.94%	85.64%	93.62%	98.40%		
6-4-3	60.11%	89.36%	93.09%	96.28%	99.47%			
6-4-4	59.57%	87.77%	94.15%	98.94%	100.00%			

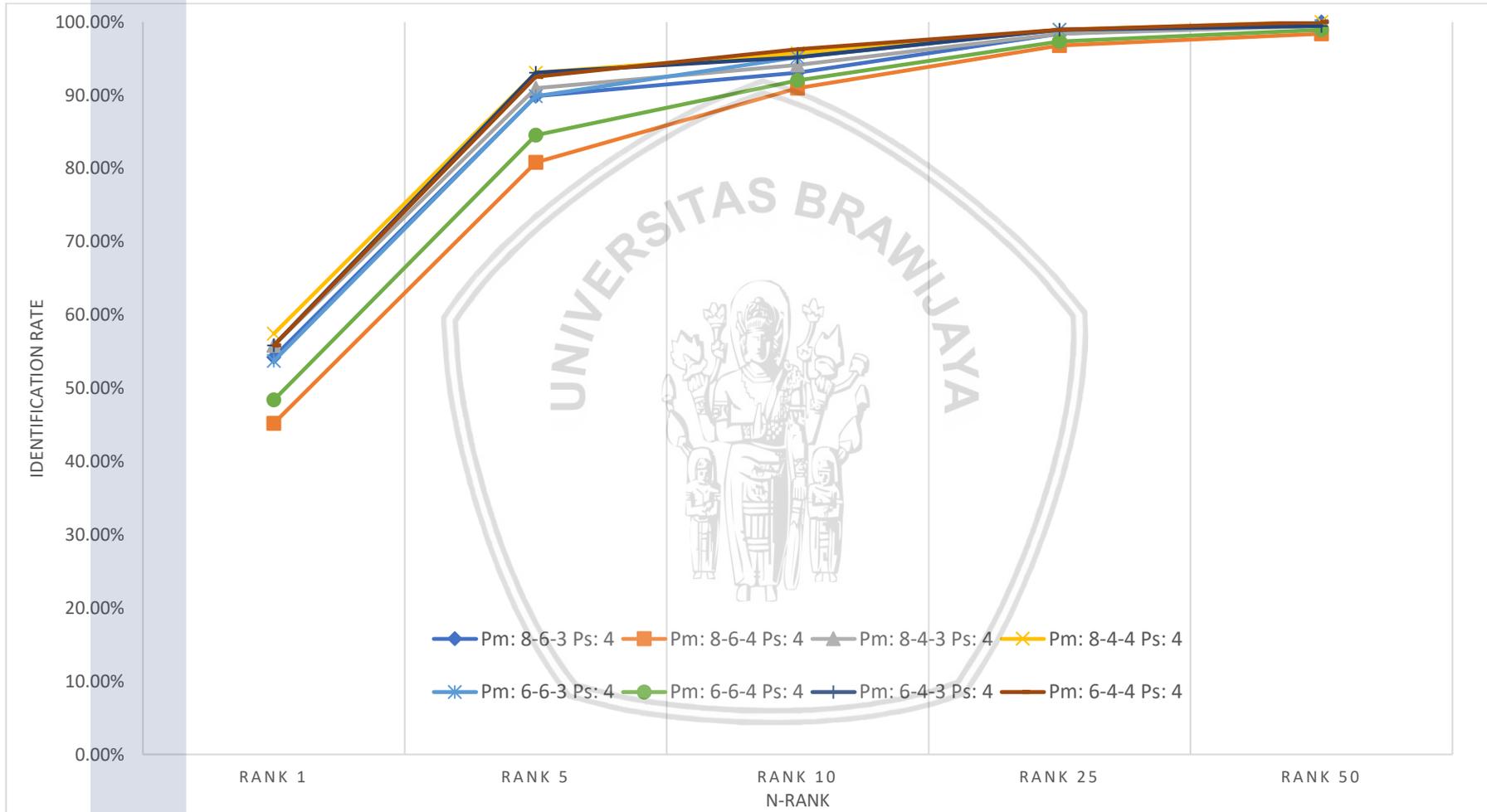




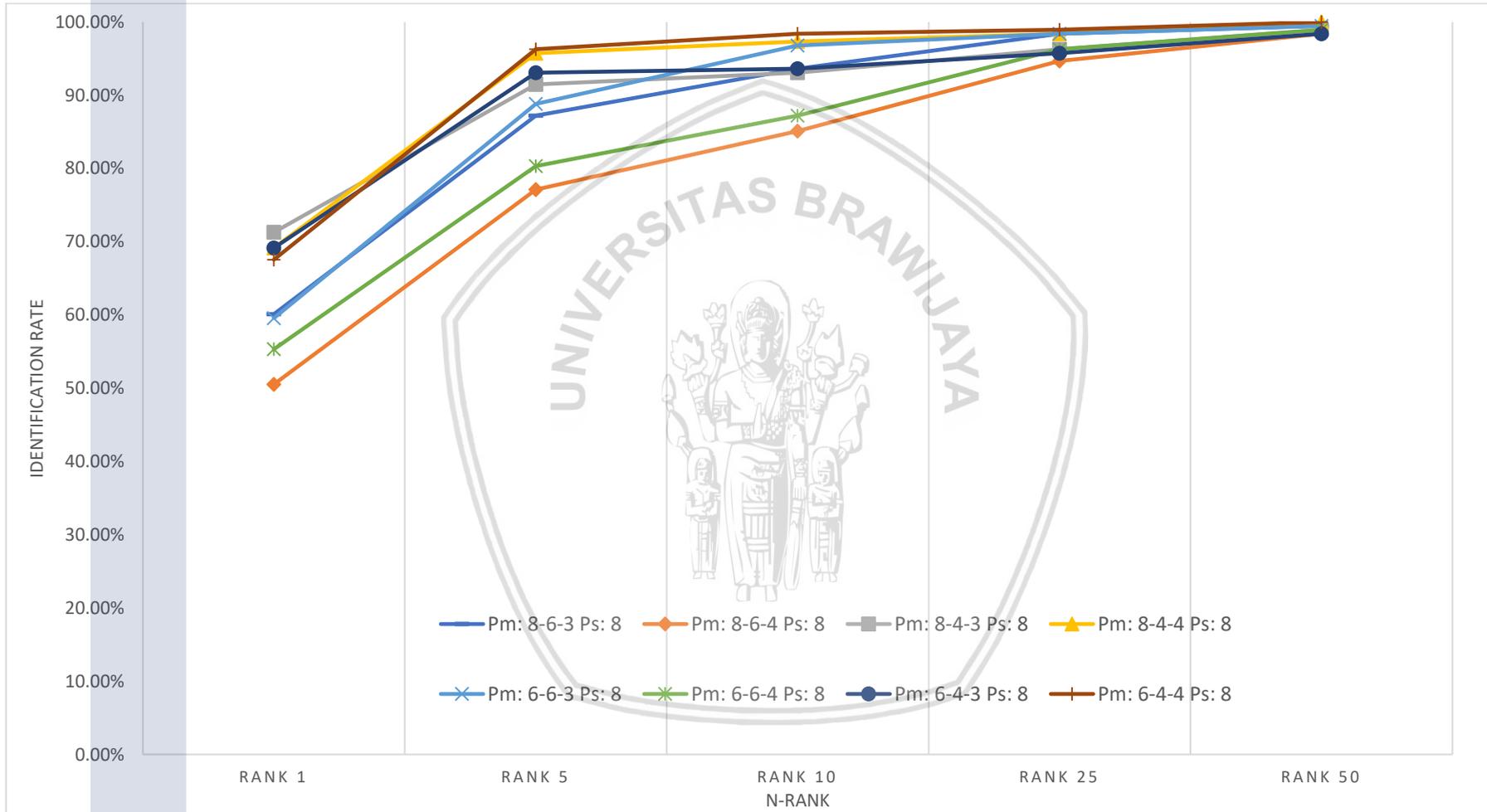
Gambar 6.6 Grafik CMC SIFT+MCWLD Sliding window 16 Overlapping SIFT 4



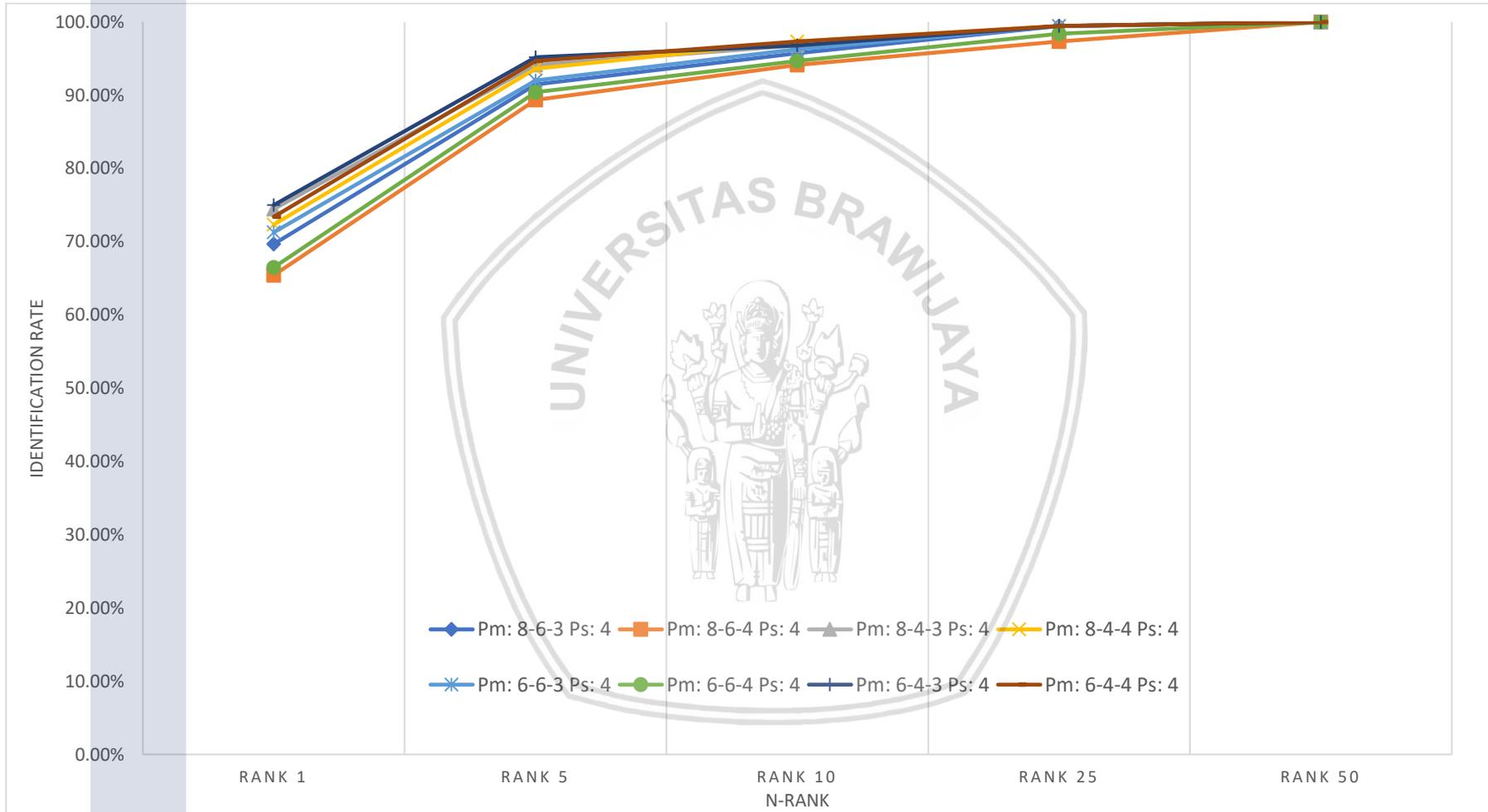
Gambar 6.7 Grafik CMC SIFT+MCWLD Sliding window 16 Overlapping SIFT 8



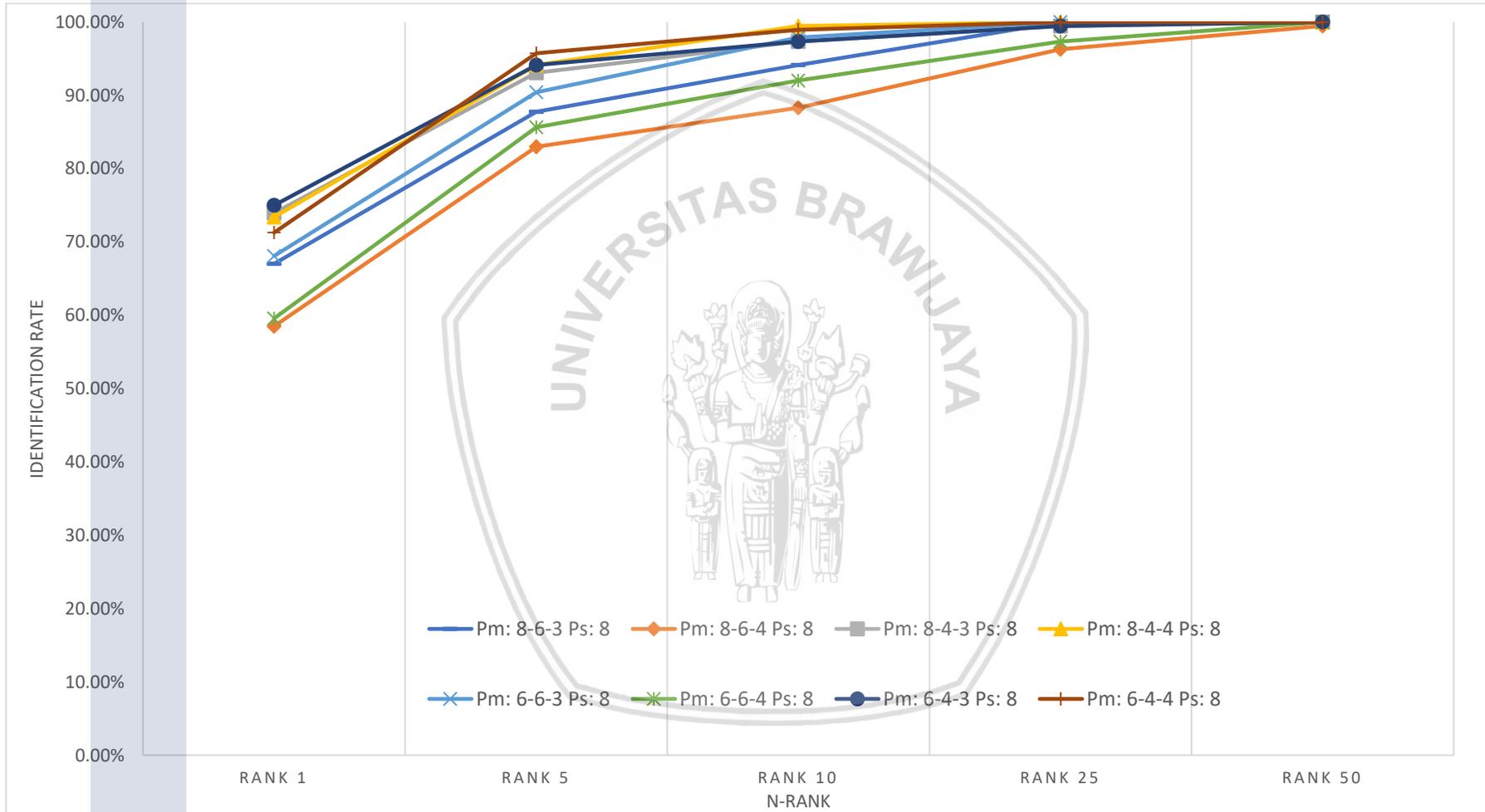
Gambar 6.8 Grafik CMC SIFT+MCWLD Sliding window 16 Non-overlapping SIFT 4



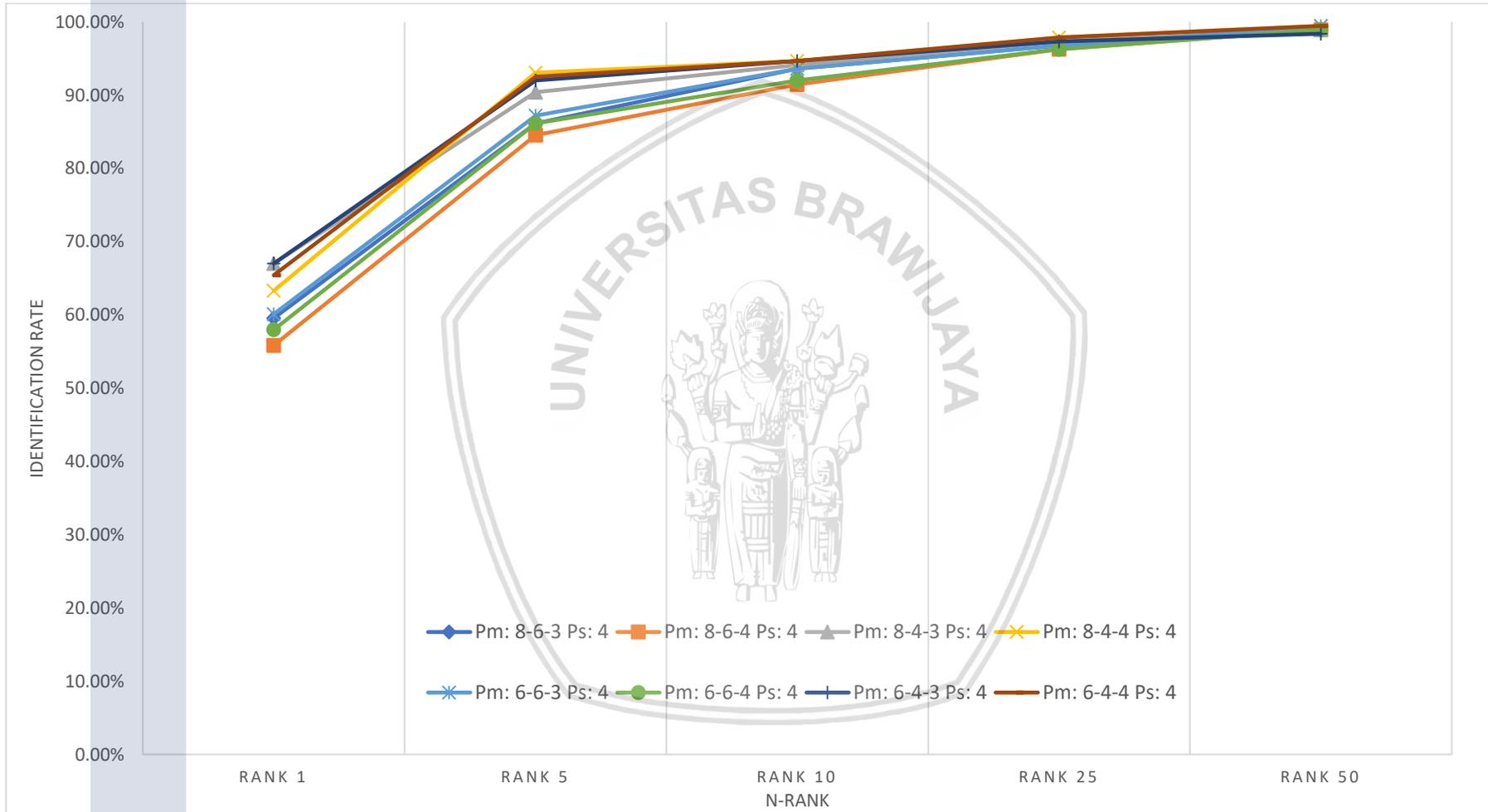
Gambar 6.9 Grafik CMC SIFT+MCWLD Sliding window 16 Non-overlapping SIFT 8



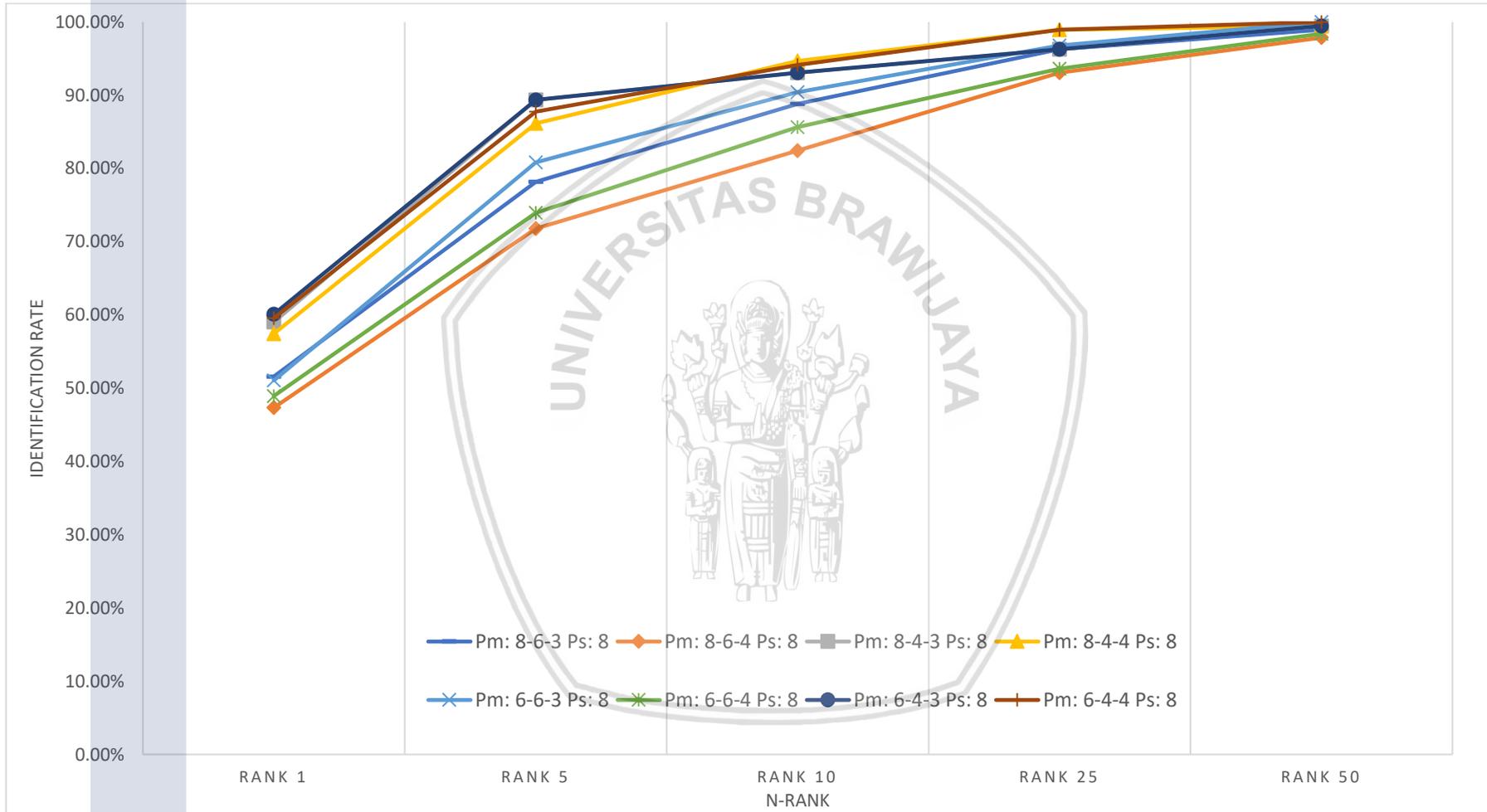
Gambar 6.10 Grafik CMC SIFT+MCWLD Sliding window 32 Overlapping SIFT 4



Gambar 6.11 Grafik CMC SIFT+MCWLD Sliding window 32 Overlapping SIFT 8

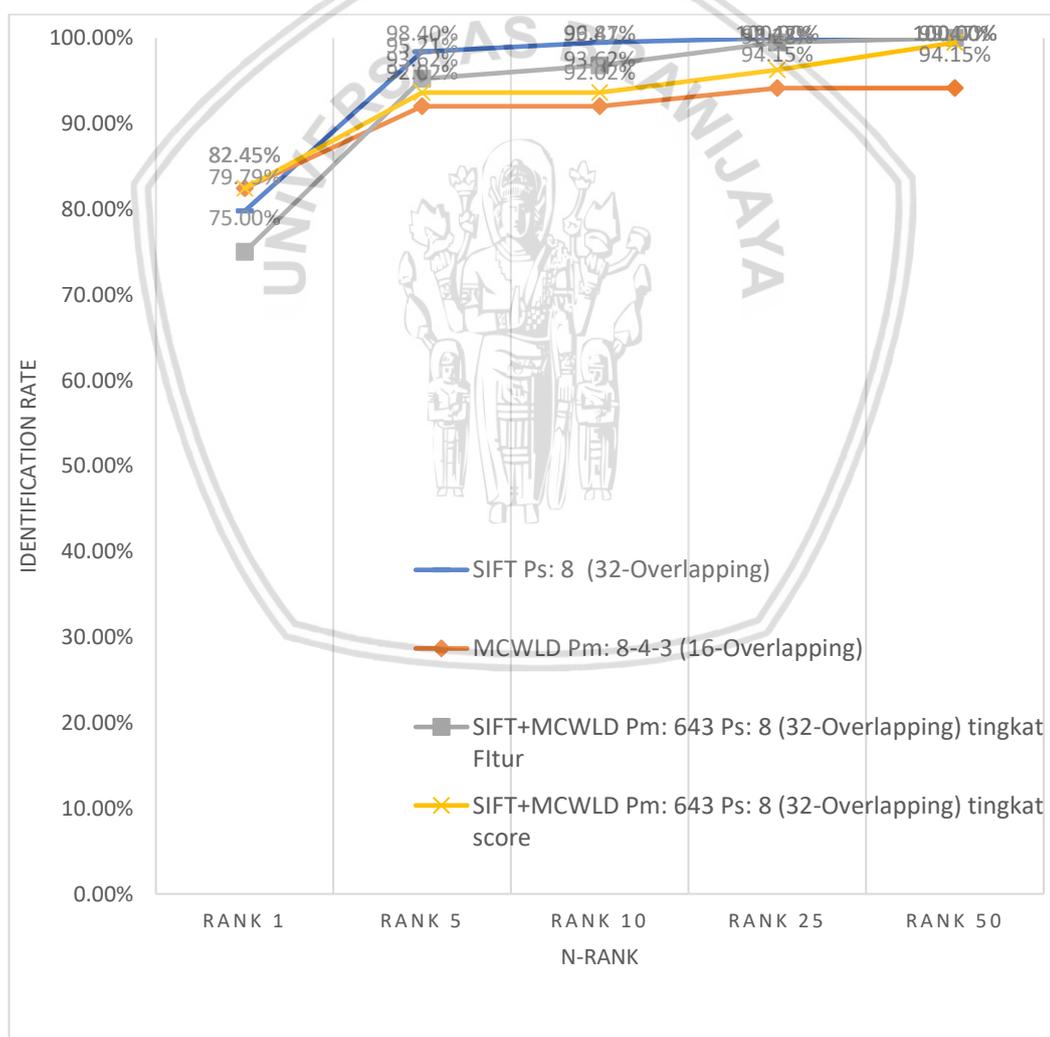


Gambar 6.12 Grafik CMC SIFT+MCWLD Sliding window 32 Non-overlapping SIFT 4



Gambar 6.13 Grafik CMC SIFT+MCWLD Sliding window 32 Non-overlapping SIFT 8

Dari Gambar 6.6, Gambar 6.7, Gambar 6.8, Gambar 6.9, Gambar 6.10, Gambar 6.11, Gambar 6.12, dan Gambar 6.13 terdapat kecenderungan bahwa nilai *identification rate* yang dihasilkan berdampak sepenuhnya oleh metode MCWLD. Pada metode SIFT yang difusikan, konfigurasi parameter terbaik juga yang memberikan dampak bagi metode MCWLD. Dari seluru grafik tersebut terbukti bahwa parameter milik SIFT dengan *window* kecil ukuran 8 dapat meningkatkan nilai *identification rate* yang dihasilkan. Meski dengan konfigurasi terbaik dari metode SIFT, nilai terbaik tetap merupakan konfigurasi terbaik dari metode MCWLD. Adapun pada Gambar 6.14 dapat terlihat dengan jelas bahwa nilai *identification rate* terbaik tetap dipegang oleh metode MCWLD dengan konfigurasi terbaiknya. Terdapat persamaan nilai *identification rate* pada metode SIFT dengan SIFT+MCWLD. Dari kedua metode tersebut, bentuk grafik yang dihasilkan sama dan berbeda jauh dengan metode SIFT. Selanjutnya hasil dari pengujian ini akan dianalisis pada subbab berikutnya.



Gambar 6.14 Grafik CMC SIFT+MCWLD dengan konfigurasi terbaik pada *top-match*



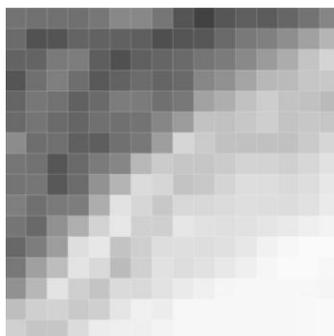
### 6.1.4 Analisis Hasil Pengujian

Analisis dari pengujian yang dilakukan sebelumnya dirangkum dan dibagi kepada masing-masing metode dengan mempertimbangkan faktor-faktor yang telah disebutkan pada subbab 4.3 Perancangan Skenario Pengujian. Adapun hasil analisis adalah sebagai berikut:

#### 6.1.4.1 SIFT

##### 1. Analisis parameter dan konfigurasi *sliding window*

Pengaruh konfigurasi *sliding window* terhadap metode SIFT menunjukkan bahwa nilai terbaik diperoleh pada konfigurasi *overlapping* pada semua ukuran *window* yang digunakan. Hal ini disebabkan karena pada konfigurasi *overlapping*, nilai fitur yang diekstraksi lebih banyak dan ada pada setiap besaran perpindahannya. Nilai yang diberikan sebagai beragam sehingga dapat membedakan sketsa dari orang yang berbeda. Untuk ukuran *window* yang digunakan, ukuran 32 x 32 *pixel* memiliki nilai *top match* yang lebih baik dibandingkan dengan ukuran *window* 16 x 16 *pixel*. Ukuran ini mempengaruhi luasan dalam mengekstraksi nilai pada metode SIFT. Dengan ukuran 32 x 32 *pixel* dan ukuran *window* kecil bernilai 4 x 4 *pixel*, tentu Ukuran dari fitur ini juga berbeda dengan ukuran fitur pada ukuran *window* 16 x 16 *pixel*. Namun secara teknis, *histogram binning* yang dilakukan oleh SIFT tidak dipengaruhi oleh ukuran *window* tersebut. Semakin besar ukuran *window* kecil yang digunakan, semakin besar nilai dalam satu bin. Untuk hasil dari pepadanan ada pada lampiran. Hal lain membuktikan bahwa konfigurasi tersebut dapat terbilang baik adalah pada kondisi *top match* dan pada rank ke-25, nilai *identification rate* pada ukuran *window* 32 *overlapping* sudah mendapati nilai yang sempurna. Kemudian pengujian parameter pada metode SIFT menunjukkan bahwa ukuran *window* kecil yang dijadikan sebagai luasan untuk mengekstraksi fitur gradien dan orientasi mempengaruhi terhadap hasil yang diberikan. Ukuran dari *window* kecil tersebut mempengaruhi pada proses *histogram binning* pada satu *window* kecil.



Gambar 6.15 Citra sketsa 16x16 *pixel*



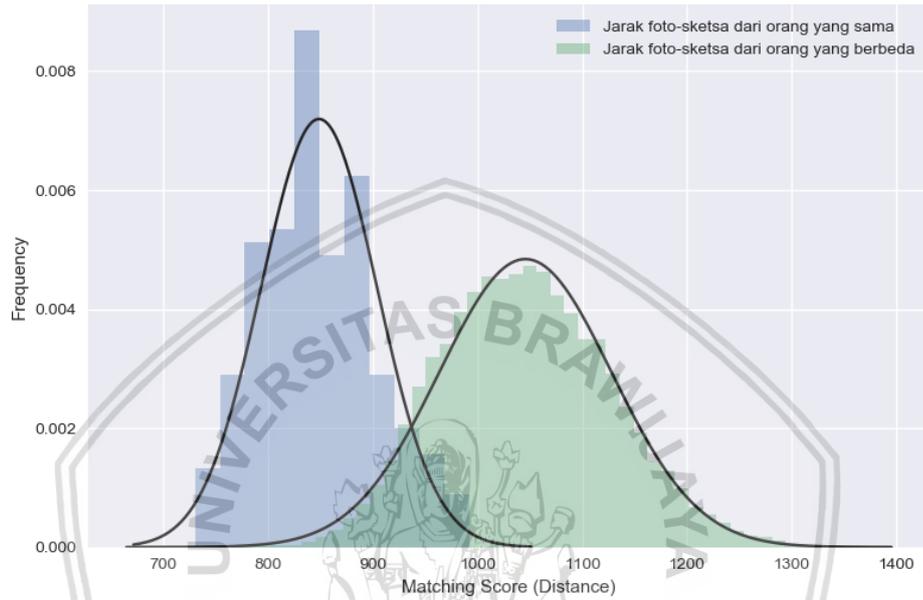
**Gambar 6.16** Citra sketsa 32 x 32 pixel

Jumlah nilai yang diekstraksi pada ukuran *window* kecil 4 dengan 8 berbeda sehingga menghasilkan perbedaan pada hasil yang diberikan. Jumlah nilai yang dipadankan antara foto dengan sketsa pada masing-masing patch atau *window* lebih banyak pada ukuran *window* kecil 8, sehingga nilai jarak antar *window* dapat semakin kecil atau semakin besar. Jika dilihat dari grafik *distance frequency distribution* (Gambar 6.17 dan Gambar 6.18), pada kasus *sliding window* dengan ukuran 16 dan 32 dengan konfigurasi *overlapping* menunjukkan bahwa dengan ukuran *window* 32 dapat memisahkan antara sketsa dari orang tersebut dengan orang lain. Hal ini terbukti dari rasio *within-between class variance ratio* pada persamaan (2.4) dari grafik tersebut.

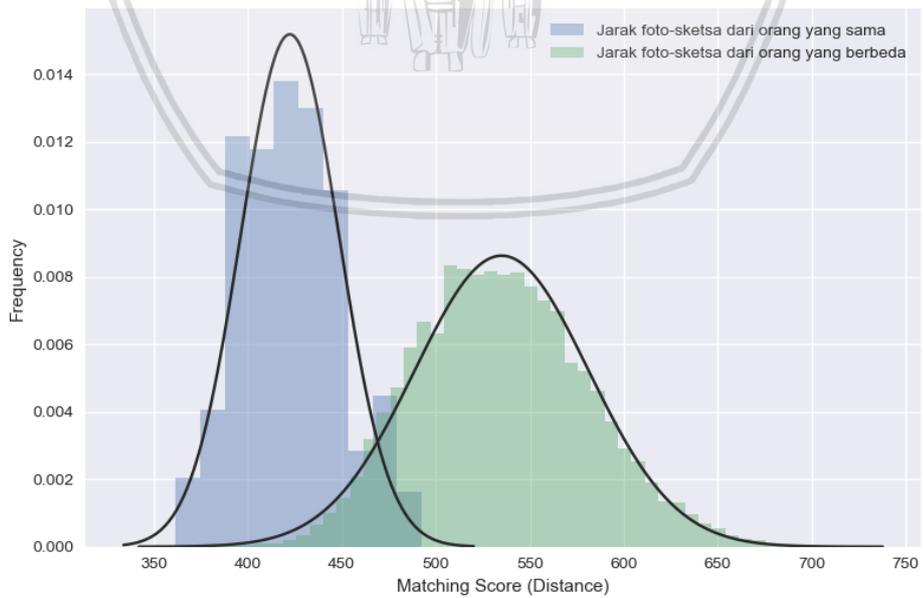
2. Analisis EER dan *within-between class variance ratio*

Pada Gambar 6.17, nilai rasio yang diberikan adalah 0.00144 sedangkan pada Gambar 6.18 nilai rasionya adalah 0.0013. Tentu nilai yang lebih kecil yang lebih baik dalam memisahkan antar dua kelas. Namun hal tersebut tidaklah cukup untuk memberikan sebuah kesimpulan bahwa konfigurasi tersebut adalah yang terbaik untuk metode SIFT. Selain melihat dari rasio *within-between class variance ratio*, analisis ini juga dapat dilihat dari nilai *equal error rate* (EER) antara *false match rate* dengan *false non-match rate* yang dimiliki metode tersebut. Nilai EER pada Gambar 6.17 adalah 8.19% sedangkan untuk Gambar 6.18 adalah 6.70%. Nilai EER tersebut menunjukkan titik perpotongan yang memiliki nilai *false match*

rate dan false non-match rate yang sama. Semakin rendah nilai EER, maka semakin tinggi nilai akurasi. Berdasarkan analisis EER dan *within-between class variance ratio* dapat diambil kesimpulan bahwa konfigurasi untuk metode SIFT terbaik adalah menggunakan *window* kecil berukuran 8 dengan konfigurasi *sliding window* yang *overlapping* dan menggunakan *window* berukuran 32 karena dapat memisahkan kelas *genuine* dan *impostor* atau meminimalisir nilai *between class*-nya.



**Gambar 6.17** Grafik Distance Frequency Distribution metode SIFT dengan *sliding window 16 overlapping* dan ukuran *window 8*



**Gambar 6.18** Grafik Distance Frequency Distribution metode SIFT dengan *sliding window 32 overlapping* dan ukuran *window 8*

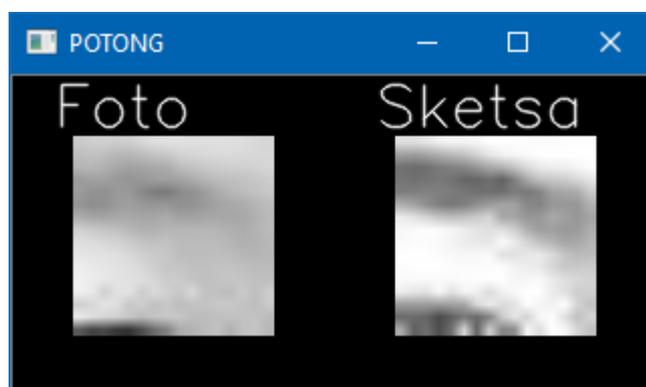


### 3. Analisis fitur

Untuk menganalisis metode SIFT dari sisi fitur tidak dapat dilakukan dengan melihat persebaran dari nilai *magnitude gradient* dan orientasinya, karena metode SIFT sendiri mengelompokkan nilai *magnitude gradient* dalam bin yang berkorespondensi dengan nilai orientasi pada *pixel* tersebut. terlebih nilai *magnitude gradient* pada SIFT telah diberi bobot sehingga analisis dengan melihat persebaran nilai tersebut tidak dianjurkan. Karena posisi dari setiap patch atau *window* tersebut sama maka yang dapat dianalisis adalah nilai yang dibentuk dari *histogram binningnya* bukan posisi dari *pixel* tersebut. Salah satu alasan mengapa nilai jarak yang diberikan antara sketsa dengan foto tersebut bisa dekat antara lain karena proses ekstraksinya. Meski nilai yang *pixel* pada kedua citra (foto dan sketsa) berbeda, nilai gradien yang dihasilkan tidak terlalu jauh.



Gambar 6.19 Visualisasi *sliding window*



Gambar 6.20 Potongan *Sliding window*

Seperti pada Gambar 6.19 (Individu M (99)) terlihat bahwa posisi *window* ada pada posisi yang sama dan pada Gambar 6.20 (diperbesar untuk kebutuhan visualisasi) terdapat gambar alis mata pada foto dan sedikit gambar mata pada sketsa. Meski dari kedua potongan citra tersebut berbeda, namun nilai *magnitude gradient* yang dihasilkan tentu tidak jauh. Hal ini terbukti dari nilai jarak yang diberikan pada posisi tersebut (*distance: 3.8308189251520517*). Komposisi warna pada citra tersebut adalah sebagai berikut:

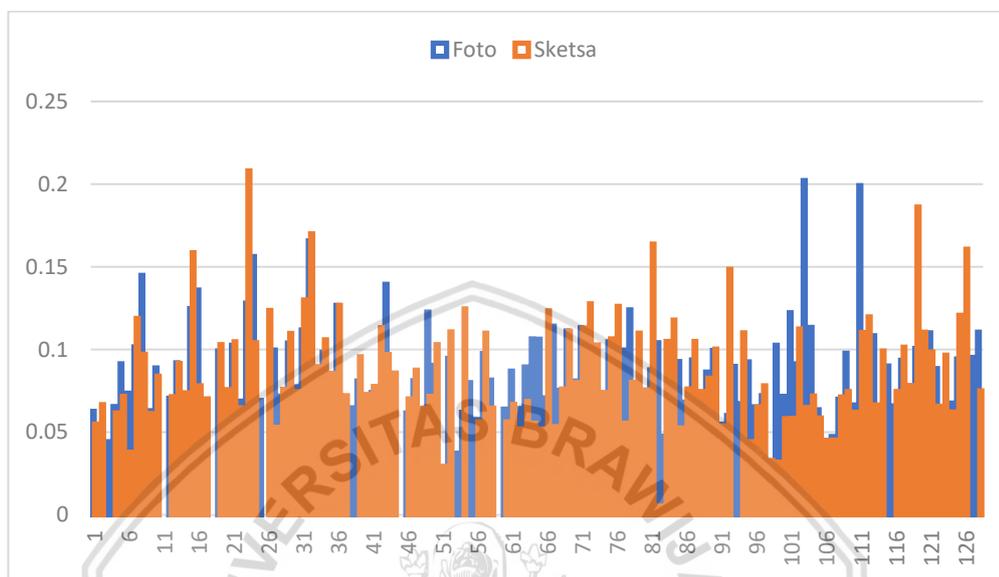
**Tabel 6.8 Nilai *pixel* pada potongan Foto**

[[202 207 208 206 208 216 223 225 225 224 223 223 223 221 217 214]
[191 194 191 184 183 189 197 200 212 214 217 219 219 218 216 215]
[181 184 179 169 163 165 170 172 189 195 202 206 208 209 211 214]
[156 161 161 154 150 153 159 161 163 172 183 189 192 195 200 205]
[157 147 146 154 153 138 125 121 139 145 158 173 183 189 195 201]
[150 143 144 155 159 153 150 153 144 147 155 164 169 173 181 189]
[174 167 165 169 169 166 167 173 163 163 166 168 167 168 176 186]
[210 203 197 193 185 177 177 182 182 182 183 181 176 173 179 187]
[223 220 216 212 204 196 193 195 187 187 188 187 182 176 178 184]
[230 228 226 223 217 209 203 199 190 188 189 191 189 183 181 183]
[238 234 231 227 223 217 207 198 198 193 192 195 196 192 188 188]
[241 237 232 228 225 223 215 204 204 196 192 195 198 196 193 191]
[240 235 207 226 227 204 219 210 200 217 194 192 188 204 191 205]
[206 215 229 223 200 208 193 200 189 195 195 202 201 195 190 202]
[237 223 214 210 199 187 197 199 200 189 189 191 193 194 196 188]
[ 89 65 73 54 49 78 109 131 172 178 188 194 176 193 197 184]]

**Tabel 6.9 Nilai *pixel* pada potongan Sketsa**

[[224 224 232 242 245 244 248 255 254 254 252 252 251 253 253 253] [160 158 165 177 187 195 212 231 244 248 252 255 255 255 255 255]
[112 108 113 124 131 136 151 170 179 197 220 235 239 241 247 253]
[117 113 116 120 114 102 102 112 96 129 169 194 201 205 217 230]
[125 127 130 129 124 118 113 111 111 125 120 131 162 168 171 202]
[175 162 143 126 120 128 144 157 164 162 138 132 152 152 151 178]
[236 227 212 198 191 195 205 214 167 166 151 152 170 162 151 165]
[248 252 254 252 246 237 229 224 225 208 177 158 160 154 156 175]
[250 252 252 249 243 237 233 231 252 237 214 194 181 166 168 182]
[255 255 254 252 249 247 248 251 225 225 232 234 219 197 187 189]
[252 254 255 255 253 252 252 253 254 243 238 234 215 201 201 204]
[249 250 251 250 250 253 254 255 255 246 247 251 236 223 217 211]

[239 230 227 250 225 245 241 219 242 250 255 254 236 225 206 209]  
 [156 156 157 156 162 166 216 237 220 227 254 245 241 255 212 194]  
 [106 108 102 74 96 119 105 129 203 230 210 248 254 223 193 212]  
 [171 95 112 78 113 217 97 208 132 139 247 222 255 248 215 193]]



**Gambar 6.21 Perbandingan histogram foto dan sketsa pada potongan citra**

Dari komposisi tersebut yang menjadi pembeda antara nilai *magnitude* gradien pada foto dan sketsa hanya pada pembobotannya saja yang dipengaruhi oleh posisi dari *pixel* tersebut. Nilai histogram yang dihasilkan dapat dilihat pada Gambar 6.21 yang menunjukkan bahwa kedua histogram tersebut memiliki bentuk identik sehingga jarak yang dihasilkan menjadi lebih kecil. Di bawah ini adalah perbandingan individu m 99 dengan f 14 yang diambil berdasarkan jarak terjauh pada kondisi baseline. Jarak yang diberikan adalah 6.122406642499616. Nilai jarak tersebut lebih besar dibanding dengan sebelumnya. Hal ini juga disebabkan oleh nilai *magnitude* gradien dan orientasi yang diekstraksi juga berbeda. Nilai yang diberikan cenderung lebih besar pada indeks awal atau pada *window* kecil di baris pertama. Posisi *pixel* yang terlampaui jauh ini yang menyebabkan nilai jarak tersebut menjadi lebih besar. Jika nilai histogram yang diberikan sama atau mendekati meski merupakan individu berbeda juga menyebabkan individu tersebut dianggap sebagai individu yang sama.

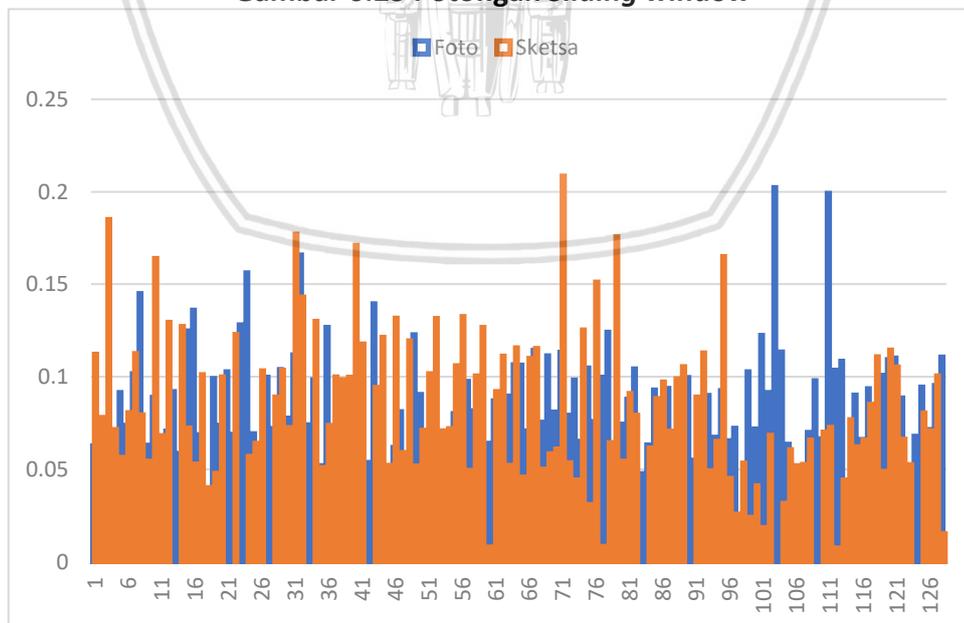




Gambar 6.22 Visualisasi *sliding window*



Gambar 6.23 Potongan *Sliding window*



Gambar 6.24 Perbandingan histogram foto dan sketsa pada potongan citra pada individu yang berbeda

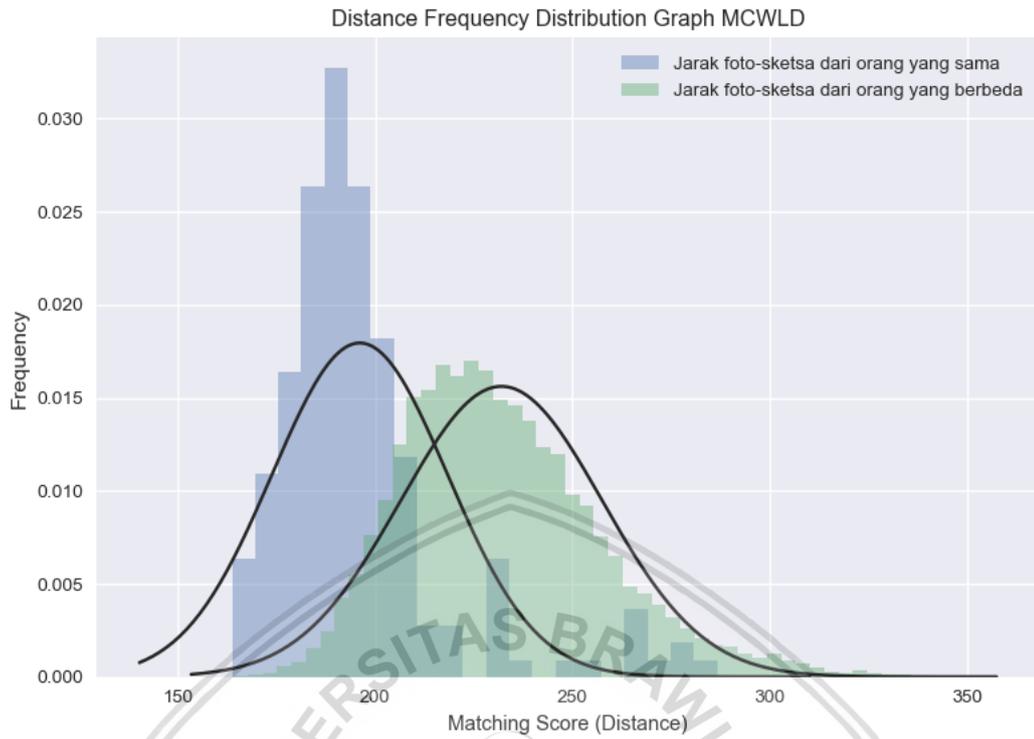


Dari Gambar 6.22 diambil potongan citra yang seperti yang terlihat pada Gambar 6.23. kedua potongan *window* ini kemudian diekstraksi nilai fitur SIFT seperti yang terlihat pada Gambar 6.24 sebagai perbandingan sketsa dan foto dari individu yang berbeda. Berikutnya, jika analisis ini dikorespondesikan dengan parameter yang diuji maka histogram maka yang dipengaruhi hanya pada proses ekstraksi nilai *magnitude* gradien dan orientasinya saja yang lebih besar nilainya pada setiap bin.

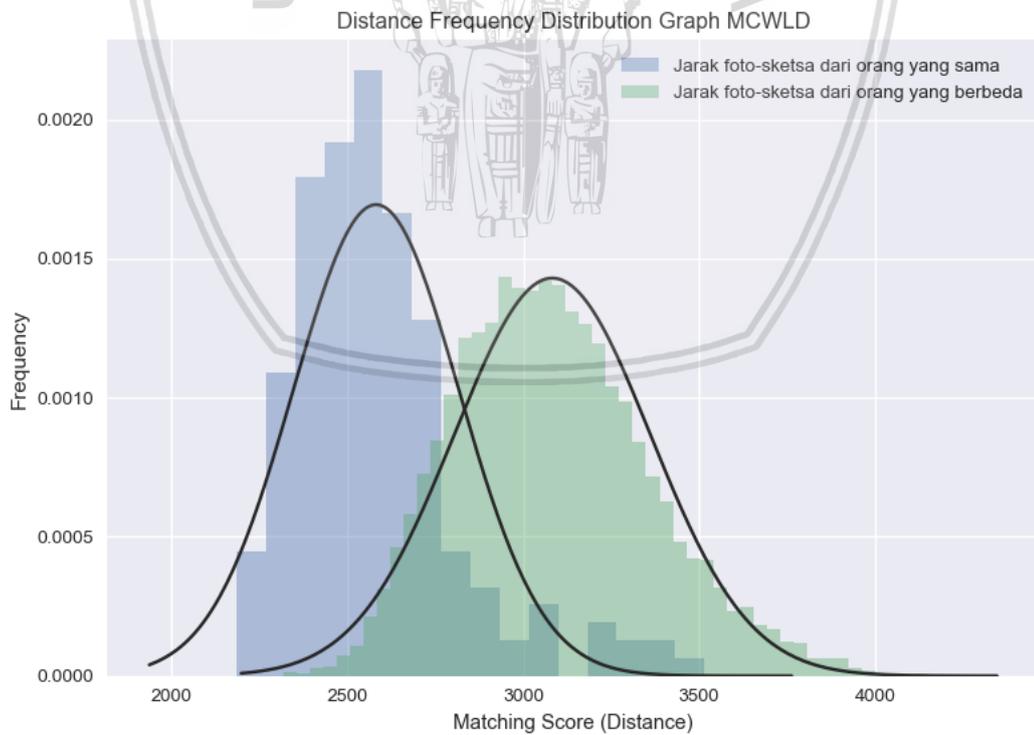
#### 6.1.4.2 MCWLD

##### 1. Analisis parameter dan konfigurasi *sliding window*

Pada metode MCWLD untuk pengujian pengaruh *overlapping* dan *non-overlapping* menunjukkan bahwa menggunakan *sliding window* dengan konfigurasi *overlapping* memiliki *identification rate* yang lebih baik. Hal ini dipengaruhi oleh semakin padatnya nilai yang diekstraksi oleh MCWLD. Berbeda dengan metode SIFT, semakin kecil *window* yang digunakan, hasilnya semakin baik. Dari hasil pengamatan ini disebabkan karena jika ukuran *window* semakin banyak, maka akan menyebabkan variasi dari fitur untuk masing-masing patch atau *window* semakin sedikit. Sebaliknya jika ukuran *window* tersebut semakin kecil, maka metode MCWLD dapat memberikan nilai terbaik untuk masing-masing patch atau *window*nya. Hal ini dapat terlihat pada Tabel 6.6 Pengujian Parameter MCWLD pada semua konfigurasi *sliding window* yang menunjukkan bahwa pengaruh dari *overlapping* pada metode MCWLD dapat meningkatkan nilai *identification rate* dikarenakan fitur yang diekstraksi oleh metode MCWLD ada pada setiap perpindahan *pixel*nya. Dengan begitu nilai yang didapat menjadi lebih banyak variasinya. Berbeda dengan SIFT yang dapat mengeliminasi konfigurasi yang lain, pada metode MCWLD, nilai pada masing-masing *rank* memiliki konfigurasi yang berbeda. Namun dari semua parameter yang diuji, terdapat sebuah kecenderungan terhadap salah satu parameter, yakni *M-segment*. Pada *top-match*, konfigurasi terbaik (*sliding window overlapping* dan *non-overlapping*) adalah MCWLD dengan parameter  $T=8$ ,  $M=4$  dan  $S=3$ . Sedangkan pada *rank-5* dan *rank-10* memiliki parameter  $T=6$ ,  $M=4$  dan  $S=3$ . Pada *rank-25* memiliki parameter  $T=6$ ,  $M=4$  dan  $S=4$ . Dari keseluruhan *rank-1* hingga *rank-25* ada pada konfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping*. Namun pada *rank-50* konfigurasi terbaik ada pada *non-overlapping* dan *window* berukuran 16 dengan parameter  $T=8$ ,  $M=4$ , dan  $S=4$ . Tentu jika mengambil kesimpulan dari analisis ini, maka yang terbaik adalah konfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$  karena berada pada nilai *rank* rendah.



**Gambar 6.25** Grafik Distance Frequency Distribution metode MCWLD *sliding window 16 overlapping* dan parameter  $T=8, M=4, S=3$



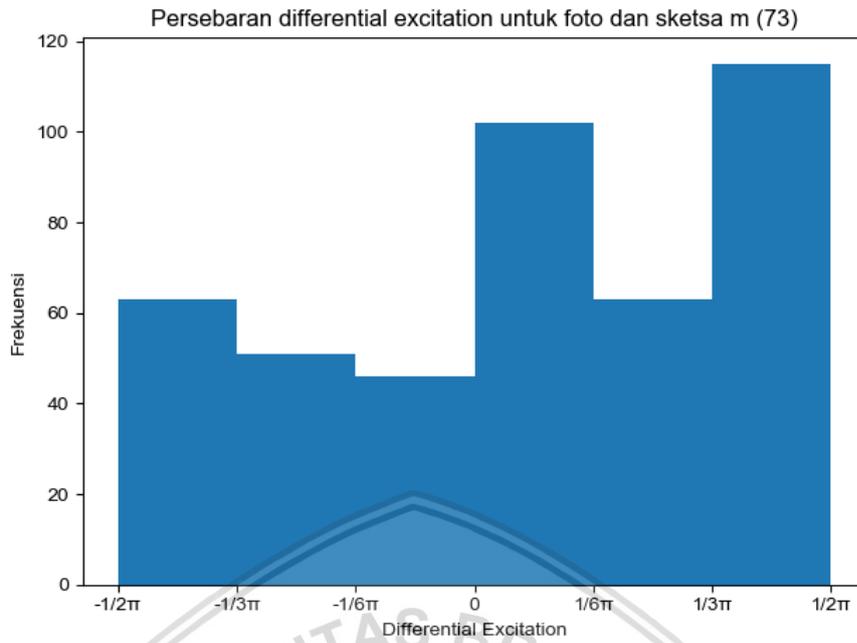
**Gambar 6.26** Grafik Distance Frequency Distribution metode MCWLD *sliding window 16 overlapping* dan parameter  $T=6, M=4, S=3$

## 1. Analisis EER dan *within-between class ratio*

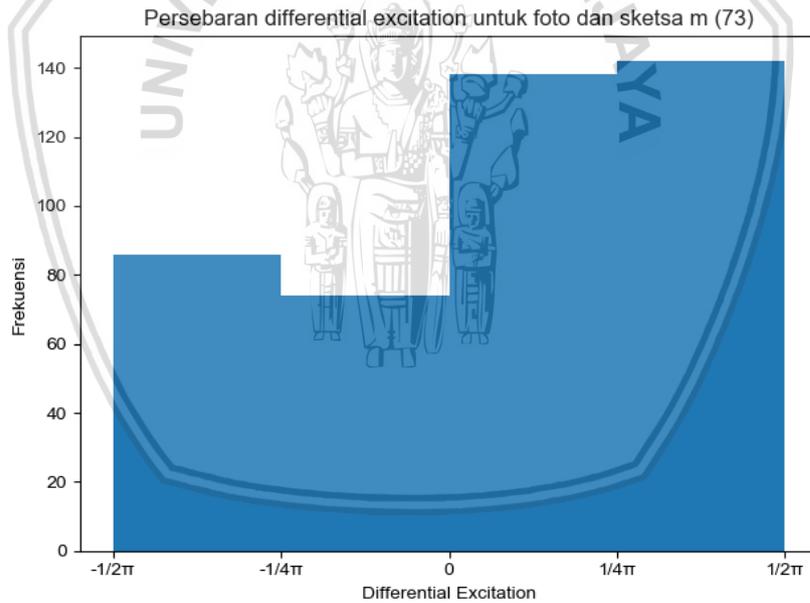
Berdasarkan dari analisis sebelumnya, konfigurasi terbaik adalah *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$ . Jika dilihat dari Gambar 6.25 (terbaik pada *top-match*) dan Gambar 6.26 (kesimpulan dari analisis parameter dan konfigurasi *sliding window*), luasan irisan pertama lebih kecil dibandingkan yang kedua. Kemudian jika dilihat dari nilai *within-between class variance ratio*, Gambar 6.25 memiliki nilai rasio 0.003211 dan Gambar 6.26 memiliki nilai rasio 0.002863. Nilai rasio pada Gambar 6.26 lebih kecil dikarenakan nilai jarak pada kelas yang sama lebih dekat. Hal ini terbukti dari nilai pada *top-match*. Jika dilihat dari nilai EER, Gambar 6.25 memiliki nilai EER sebesar 16.29% dan Gambar 6.26 memiliki nilai EER sebesar 13.76%. Nilai EER yang lebih kecil memiliki indikasi bahwa metode tersebut dapat meminimalisir dua tipe *error* sehingga dapat memiliki nilai akurasi yang tinggi. Namun dari keseluruhan data (dapat dilihat pada lampiran) menunjukkan bahwa nilai EER terkecil (dengan nilai EER 13.18%) ada pada konfigurasi *sliding window* dengan ukuran *window* 32 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$ . Nilai *within-between class variance ratio* pada konfigurasi tersebut adalah 0.003786. Nilai rasio tersebut tentu lebih kecil dari kedua konfigurasi yang telah dibahas sebelumnya. Berdasarkan grafik pada Gambar 6.4 terdapat kecenderungan bahwa konfigurasi tersebut lebih unggul dibanding yang lainnya. Jika mempertimbangkan dari analisis ini, konfigurasi terbaik adalah konfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$  karena memiliki nilai EER dan rasio yang lebih rendah dibanding yang lainnya.

## 2. Analisis fitur

Analisis fitur ini dilihat dari pengujian parameter terhadap MCWLD. Parameter tersebut antara lain adalah *T-dominant orientation*, *M-Segment* dan *S-bins*. *T-dominant orientation* merupakan nilai banyaknya orientasi yang paling dominan. Nilai ini menjadi parameter untuk melakukan *quantization* nilai orientasi. *M-Segment* merupakan pembagi dari nilai *differential excitation* menjadi  $m$ -bagian dengan rentang atas dan bawahnya sendiri. Karena MCWLD dibentuk oleh histogram 2-dimensi, maka *S-bins* digunakan sebagai bin untuk masing-masing segmen pada orientasi tertentu. Dari pengujian parameter tersebut membuktikan bahwa dengan mengecilkan nilai *M-Segment* dapat mempengaruhi nilai *identification rate* yang diberikan dibandingkan dengan merubah nilai *T-dominant orientation*. Hal ini disebabkan karena persebaran nilai *differential excitation* pada dapat dikelompokkan dengan baik dengan jumlah segmen yang lebih sedikit. Berikut adalah persebaran dari ukuran *window* 16 dan perbandingan baseline dengan parameter 8-4-3.



Gambar 6.27 Histogram persebaran nilai *differential excitation* (M=6)

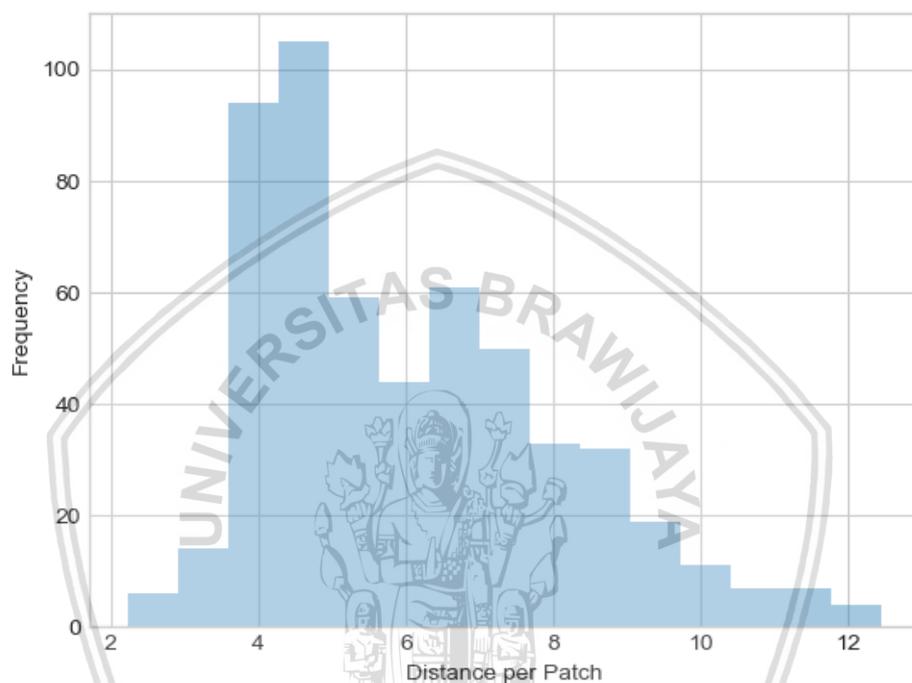


Gambar 6.28 Histogram persebaran nilai *differential excitation* (M=4)

Dari Gambar 6.27 dan Gambar 6.28 menunjukkan bahwa persebaran nilai *differential excitation* yang teruji baik ada pada jumlah *M-Segment* sama dengan 4. Gambar 6.27 dan Gambar 6.28 merupakan persebaran nilai *differential excitation* dengan jumlah bin yang sama dengan nilai *M-Segment*. Jumlah nilai *M-Segment* tersebut membagi nilai *differential excitation* menjadi m-segmen. Dengan jumlah yang semakin kecil, maka nilai *differential excitation* untuk masing-masing orientasi dari



dikelompokkan dengan baik. Mengingat bahwa *differential excitation* pada metode MCWLD dibagi terlebih dahulu oleh *T-dominant orientation* dari orientasinya maka akan semakin sedikit nilai *differential excitation* yang harus dikelompokkan. Dengan memperkecil nilai *M-Segment*, tidak akan menimbulkan nilai kosong yang terlalu banyak. Analisis ini menjawab bahwa ukuran citra 32 akan memperbanyak nilai *differential excitation* pada suatu segmen yang membuatnya menjadi sebuah fitur yang kurang baik untuk digunakan.



**Gambar 6.29** Grafik *distance frequency distribution* pada individu m (78)

Jika dibandingkan antara metode SIFT dengan MCWLD, yang kurang dari metode MCWLD adalah tidak adanya pembobotan. Fitur pembobotan pada metode SIFT dapat membantu untuk menghilangkan nilai *redundant* pada suatu fitur. Pada MCWLD, nilai jarak yang dihasilkan pada individu yang sama tentu tidak sebaik pada metode SIFT. Jika dilihat dari Gambar 6.22 Visualisasi *sliding window* nilai jarak yang diberikan oleh MCWLD adalah 13.36 dan jika dibandingkan dengan jarak pada seluruh *window* yang dihasilkan dari *sliding window*, nilai jarak tersebut termasuk jauh. Pembobotan pada kasus ini bisa menambah kedekatan antara *window* dari foto dengan sketsa pada individu yang sama. Baik pada metode SIFT dan MCWLD, keduanya baik pada salah satu individu. Pada metode SIFT, individu dengan identitas m-99 memiliki nilai jarak yang paling dekat, sedangkan pada MCWLD ada pada individu m-78. Dari sisi fitur ini juga, didapatkan kesimpulan bahwa ukuran dari *M-Segment* mempengaruhi dalam pembentukan fitur untuk masing-masing *window* karena *M-Segment* memetakan secara luas persebaran dari nilai *differential excitation* pada *window*. Nilai parameter lain juga

berdampak baik namun tidak terlalu signifikan sehingga perubahan parameter tersebut dapat diabaikan.

#### 6.1.4.3 FUSI (SIFT+MCLWD)

Berdasarkan apa yang telah dijabarkan pada subbab pengujian, metode SIFT+MCWLD menunjukkan bahwa metode SIFT berdampak atas menurunnya nilai *identification rate* pada metode MCWLD. Hal ini disebabkan karena nilai fitur pada MCWLD terlampu lebih besar dibandingkan dengan metode SIFT. Nilai fitur pada SIFT tidak dapat mengangkat nilai *identification rate*. Fusi pada tingkat fitur hanya pada sebatas menggabungkan fitur dari kedua metode yang tentunya tidak menghilangkan pengaruh dari nilai yang kurang baik pada proses pepadanan pada salah satu metode.

Jika diteliti pada masing-masing parameter, metode MCWLD lebih menguasai representasi fitur fusi itu sendiri. Kekurangan dari fusi tingkat fitur pada penelitian ini adalah proses fusi yang terlalu sederhana sehingga menyia-nyiakan kemampuan dari metode SIFT yang sudah baik dalam penerapannya. Pada fusi tingkat fitur ini, nilai *top-match* diperoleh dari fusi SIFT dengan konfigurasi ukuran *window* kecil 8 dan *sliding window 32 overlapping* dengan metode MCWLD berkonfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$  dengan Nilai *identification rate* pada *top-match* yang diberikan adalah 75% dengan nilai EER 8.28%.

Adapun fusi pada tingkat *score*, nilai *identification rate* pada *top-match* tertinggi dari fusi yang dilakukan menggunakan *min-max normalization* diperoleh dari fusi SIFT dengan konfigurasi ukuran *window* kecil 8 dan *sliding window 32 overlapping* dengan metode MCWLD berkonfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$ . Nilai *identification rate* pada *top-match* yang diberikan adalah 82.45% dengan nilai EER 9.26%. Dari keseluruhan pengujian fusi, nilai parameter yang digunakan oleh SIFT tetap pada kondisi terbaiknya baik pada konfigurasi *sliding window* maupun parameternya. Sedangkan MCWLD berbeda-beda namun tetap pada nilai *M-Segment* yang sama. Dari hasil analisis ini, didapat kesimpulan bahwa melakukan fusi tidak selalu mendapatkan hasil terbaik karena pada salah satu metode dapat mengakibatkan menurunnya performa pada metode yang lain. Baik ada tingkat fitur maupun *score*.

Dari hasil analisis diatas, fusi terhadap kedua metode ini perlu diperhatikan bahwa fusi pada tingkat fitur tidak dapat dilakukan dengan bentuk yang terlalu sederhana seperti menggunakan *min-max normalization*. Pada tingkat *score*, fusi dengan dengan *min-max normalization* juga tidak menghasilkan akurasi terbaik. Salah satu cara untuk menghasilkan nilai yang baik adalah dengan melakukan fusi menggunakan *Neyman-Pearson Decision Rule*. Metode ini dapat menentukan suatu *probe* dapat dikatakan *impostor* kepada suatu individu berdasarkan

*likelihood ratio test* menggunakan *probability density*. Adapun rumus dari *Neyman-Pearson Decision Rule* adalah sebagai berikut:

$$\frac{P(S_1, S_2 | W_{genuine})}{P(S_1, S_2 | W_{impostor})} \geq \eta \quad (6.1)$$

Suatu *probe* dinyatakan 'cocok' jika nilai perbandingan tersebut memenuhi syarat, yakni melebihi nilai ambang batas yang telah ditentukan. Nilai *likelihood* tersebut didapat dengan menggunakan persamaan berikut:

$$p(S_1, S_2 | W_{genuine}) = p(S_1 | W_{genuine}) \times p(S_2 | W_{genuine}) \quad (6.2)$$

$$p(S_1, S_2 | W_{impostor}) = p(S_1 | W_{impostor}) \times p(S_2 | W_{impostor}) \quad (6.3)$$

Pada persamaan diatas, nilai *score*  $S_1$  dan  $S_2$  bersifat independen sehingga dapat menggunakan pendekatan *Naïve Bayes*. *Probability Density Function* adalah sebagai berikut:

$$p(S_i | W_{genuine}) = \frac{1}{\sqrt{2\pi\sigma_{i,genuine}^2}} e^{-\frac{(S_i - \mu_{i,genuine})}{2\sigma_{i,genuine}^2}} \quad (6.4)$$

$$p(S_i | W_{impostor}) = \frac{1}{\sqrt{2\pi\sigma_{i,impostor}^2}} e^{-\frac{(S_i - \mu_{i,impostor})}{2\sigma_{i,impostor}^2}} \quad (6.5)$$

Dengan menggunakan persamaan diatas, dapat dilakukan pengujian biometrik lain seperti menggunakan *open-set identification*. Dari fusi menggunakan *Neyman-Pearson Decision Rule*, nilai akurasi yang didapatkan untuk SIFT dengan konfigurasi ukuran *window* kecil 8 dan *sliding window* 32 *overlapping* dengan metode MCWLD berkonfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah T=6, M=4 dan S=3 adalah 91% (tabel ada pada bagian lampiran).

## BAB 7 KESIMPULAN DAN SARAN

### 7.1 kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan pada bab 6, penulis menjabarkan poin-poin yang didapatkan sebagai kesimpulan sebagai berikut ini:

1. Sebagaimana dijelaskan pada Bab 5 Implementasi, tahapan yang perlu dilakukan adalah melakukan *preprocessing* terhadap citra barulah kemudian dapat mengekstraksi fiturnya yang akan disimpan pada sebuah basis untuk dicocokkan dengan *probe*.
2. Metode MCWLD dan SIFT keduanya mampu mengidentifikasi sketsa wajah dari seseorang dengan pemadanan dengan perhitungan jarak dengan menggunakan *euclidean distance*. Untuk metode SIFT, konfigurasi terbaik adalah ukuran *window* kecil 8 dan *sliding window* 32 *overlapping* dengan nilai *identification rate* 79.79% sedangkan untuk MCWLD dengan konfigurasi *sliding window* dengan ukuran *window* 16 dan *overlapping* dan parameter yang digunakan adalah  $T=6$ ,  $M=4$  dan  $S=3$  dengan nilai *identification rate* 82.45%. Parameter yang perlu diperhatikan dari SIFT adalah semakin besar luasan area untuk diekstraksi fitur, maka fiturnya semakin kuat. Sedangkan untuk MCWLD, semakin kecil luasan area untuk diekstraksi fiturnya dan semakin kecil bentuk dari histogram yang dihasilkan, maka fiturnya akan semakin baik. Sedangkan pada fusi, hasil terbaik pada *top-match* adalah fusi pada tingkat *score* dengan nilai *identification rate* 82.45%. Namun jika dilihat dari *top rank*, metode terbaik adalah SIFT.
3. Fusi dari kedua metode merupakan hal yang tidak mudah dilakukan karena representasi fitur dari beda metode tidak bisa disatukan secara sembarangan yang berakibat pada menurunnya performa dari kedua metode tersebut. Cara terbaik yang dilakukan pada penelitian adalah pada tingkat *score*. Nilai dari hasil pemadanan antara kedua metode tersebut dapat dilakukan fusi salah satunya menggunakan *min-max normalization*. Hasil yang diberikan lebih baik daripada fusi pada tingkat fitur. Adapun dari hasil analisis, fusi pada tingkat *score* jauh lebih baik jika menggunakan persamaan *Neyman-Pearson Decision Rule*.
4. Penelitian masih dapat digali kembali baik dari sisi metode maupun proses pemadannya. Dengan memodifikasi atau merubah parameter dalam penelitian ini, nilai akurasi yang dihasilkan bisa lebih baik. Termasuk juga pada data yang digunakan karena pada penelitian ini hanya sebatas menggunakan *viewed sketch*. Pada realitanya, sketsa forensik jauh berbeda dengan data yang digunakan pada penelitian ini.
5. Representasi fitur merupakan kunci utama untuk mencocokkan kedua buah citra yang berbeda modalitas. Dengan menyamakan kedua modal tersebut

dalam suatu bentuk atau representasi, maka kedua citra tersebut dapat dicocokkan.

## 7.2 Saran

Saran yang diberikan dari penulis antara lain adalah sebagai berikut:

1. Metode SIFT dapat dieksplorasi kembali dengan tetap menggunakan teknik *sliding window* untuk mendapatkan *window*nya serta merubah parameter seperti ukuran *window*, rentang nilai pada histogram, melakukan pembobotan yang berbeda atau dengan kondisi citra pada *channel* warna yang berbeda.
2. Metode MCWLD sama seperti SIFT, parameter yang diuji pada penelitian ini hanya berfokus pada perubahan bentuk histogram yang dihasilkan. Hal lain yang dapat dilakukan adalah merubah nilai radius atau memodifikasi ekstraksi nilai pada bagian *differential excitation* karena sejatinya fitur MCWLD adalah histogram dari *differential excitation* pada orientasi dominan. Selain itu pembobotan juga dapat dilakukan pada metode ini selayaknya pada SIFT
3. Fusi antar metode dapat menjadi topik baru sebagai salah satu bidang pada biometrik multimodal. Metode *Neyman-Pearson Decision Rule* seperti yang dijelaskan pada bagian analisis dapat dikembangkan seperti mengganti *normal-distribution* dengan *log-normal-distribution* atau memberikan nilai bobot kecil pada *score*. Fusi pada tingkat fitur pun juga masih dapat dimodifikasi seperti dengan menyeleksi dahulu fitur dari kedua metode dengan menggunakan reduksi dimensi dengan PCA atau LDA atau mengambil komponen independen dengan menggunakan ICA.
4. Menggunakan pendekatan lain seperti *Local Binary Pattern* untuk mengganti MCWLD atau melakukan fusi terhadap keduanya.
5. Selain representasi fitur, proses *preprocessing* yang dilakukan juga dapat dikembangkan sehingga baik pada saat mengekstraksi fitur maupun pepadannya, sudah terdapat sebuah kecocokan diantara keduanya.
6. Untuk penelitian selanjutnya, dapat dilakukan analisis lebih mendalam dari kedua metode yang dipakai baik pada performa maupun komputasinya.

## DAFTAR PUSTAKA

- Abdi, W., 2010. *Principal Component Analysis*. s.l.:John Wiley & Sons, inc.
- Anon., 2005. *Sketch Recognition*. [Online]  
Available at: <https://engineering.tamu.edu/research/sketch-recognition>
- Arca, S., Campadelli, P. & Lanzarotti, R., 2006. A face recognition system based on automatically determined facial fiducial points. *Pattern Recognition*, 39(3), pp. 432-443.
- Bhatt, H., Bharadwaj, S., Singh, R. & Vatsa, M., 2010. On Matching Sketches With Digital Face Images. *Proceeding International Conference of Biometrics: Theory, Applications and Systems*.
- Bhatt, H., Bharadwaj, S. & Vatsa, M., 2012. Memetically Optimized MCWLD for Matching Sketches With Digital Face Images. *IEEE Transactions on Information Forensics and Security*, 7(5), pp. 1522-1535.
- Chen, J. et al., 2010. WLD: A robust local image descriptor. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), pp. 1705-1720.
- Deshpande, A., 2016. *A Beginner's Guide To Understanding Convolutional Neural Network*. [Online]  
Available at: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>  
[Accessed 23 May 2017].
- Gao, X., Zhong, J., Li, J. & Tian, C., 2008. Face Sketch Synthesis Algorithm Based on E-HMM and Selective Ensemble. *IEEE Transaction Circuits System Video Technology*, 18(4), pp. 487-496.
- Gonzalez, R. C., 2010. *Digital Image Processing*. 3rd ed. Upper Saddle River, New Jersey: Pearson Education.
- Goodfellow, I., 2016. *Deep Learning*. s.l.:MIT Press.
- Ikeuchi, K., 2016. *Computer Vision: A Reference Guide*. New York: Springer Science.
- Jain, A. K., Flynn, P. & Ross, A. A., 2007. *Handbook of biometrics*. s.l.:Springer Science & Business Media.
- Jain, A. K. & Li, S. Z., 2011. *handbook of face recognition*. 2nd ed. New York: Springer.
- Jain, A. K., Ross, A. A. & Nandakumar, K., 2011. *Introduction to Biometrics*. s.l.:Springer Science & Business Media.

- Jain, K. A., 2008. *Handbook of Biometrics*. New York: Springer Science.
- Klare, B. & Jain, A., 2010. Sketch-to-photo matching: A Feature-based approach. *Proceeding Society of Photo-Optical Instrumentation Engineers Conference Series*, Volume 7667.
- Klare, B. & Jain, A., 2011. Heterogeneous Face Recognition Using Kernel Prototype Similarities.
- Klare, B. & Jain, A. K., 2010. Sketch to Photo Matching: A Feature-based Approach. *Biometric Technology for Human Identification VII*, 7667(International Society for Optics and Photonics), p. 766702.
- Klare, B., Zhifeng, L. & Jain, A., 2011. Matching Forensic Sketches to Mugshot Photos. *IEEE Transactions Pattern Analysis And Machine Intelligence*, 33(3), pp. 639-646.
- Ling, Y., Yin, X. & Bhandarkar, S., 2003. Sirface vs. Fisherface: Recognition using class specific linear projection. *ICIP 2003. Proceedings*, Volume 3, pp. III-885.
- Li, S. Z. & Jain, A., 2015. *Encyclopedia of Biometrics*. s.l.:Springer Publishing Company, Incorporated.
- Liu, Q. et al., 2005. A Nonlinear Approach For Face Sketch Synthesis and Recognition. *Proc. Int. Conf. Computer Vision and Pattern Recognition*, Volume 1, pp. 1005-1010.
- Lowe, D. G., 2004. Distinctive Image Features From Scale-invariant Keypoints. *International Journal of Computer Vision*, 60(2), pp. 91-110.
- Malik, J. et al., 2016. The three R's of computer vision: Recognition, reconstruction and reorganizatio. *Pattern Recognition Letters*, Volume 72, pp. 4-14.
- McQuiston-Surrett, D., Malpass, R. S. & Topp, L. D., 2006. Use of Facial Composite Systems in U.S Law Enforcement Agencies. *Crime and Law*, 12(5), pp. 505-517.
- Nagpal, S., Vatsa, M. & Singh, R., 2016. sketch recognition: what lies ahead?. *Image and Vision Computing*, Volume 55, pp. 9-13.
- Nizami, H. et al., 2009. A Biometric Database with Rotating head Videos and Hand-Drawn Face Sketches. *Proceeding International Conference Biometrics: Theory, Applications and Systems*, pp. 38-43.

- Peng, C., Gao, X., Wang, N. & Li, J., 2017. Graphical Representation for Heterogeneous Face Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(2), pp. 301-312.
- Sharma, A. & Jacobs, D. W., 2011. Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch. *In Computer Vision and Pattern Recognition (CVPR)*, , pp. 593-600.
- Srinivasan, G. N. & G, S., 2008. Statistical Texture Analysis. *Proceedings of world academy of science, engineering and technology*, Volume 36, pp. 1264-1269.
- Tang, X. & Wang, X., 2002. Face Photo Recognition using Sketch. *Proc. Int. Conf. Image Processing*, Volume 1, pp. 257-260.
- Tang, X. & Wang, X., 2003. Face Sketch Synthesis and Recognition. *Proc. Int. Conf. Computer Vision*, Volume 1, pp. 687-694.
- Tang, X. & Wang, X., 2004. Face Sketch Recognition. *IEEE Transactions On Circuits and Systems For Video Technology*, 14(1), pp. 50-57.
- Tuceryan, M. & Jain, A. K., 1993. Texture Analysis. In: *Handbook of pattern recognition and computer vision*. s.l.:World Scientific Publishing Co., pp. 235-276.
- Uhl, R. & Lobo, N., 1996. A Framework for Recognizing a Facial Image from police sketch. *Proceeding International Conference Computer Vision and Pattern Recognition*, pp. 586-593.
- Wang, X. & Tang, X., 2009. Face Photo-Sketch Synthesis and Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 31(11), pp. 1955-1967.
- Wu, X., Song, L., He, R. & Tan, T., 2017. Coupled Deep Learning for Heterogeneous Face Recognition. *arXiv preprint arXiv:1704.02450*.
- Yuen, P. & Man, C., 2007. Human Face Image Searching System Using Sketches. *IEEE Transaction Systems, Man, And Cybernetics: Systems*, 37(4), pp. 493-504.
- Yung-hui, L., Sawides, M. & Bhagavatula, V., 2006. Illumination Tolerant Face Recognition Using A Novel Face From Sketch Synthesis Approach and Advanced Correlation Filters. *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Volume 2.
- Zhang, P. et al., 2011. Recognizing Face Sketches by a Large Number of Human Subjects: A Perception-based study for facial distinctiveness. *Proceeding International Conference Automatic Face and Gesture Recognition*.

- Zhang, W., Wang, X. & Tang, X., 2010. Lightning and Pose Robust Face Sketch Synthesis. *Proceeding International Conference Computer Vision*, pp. 420-433.
- Zhang, W., Wang, X. & Tang, X., 2011. Coupled information-theoretic encoding for Face Photo-sketch Recognition. *Proceeding International Conference Computer Vision and Pattern Recognition*.
- Zhang, Y., McCullough, C., Sullins, J. & Ross, C., 2010. Hand-drawn Face Sketch Recognition by Human and a PCA-based Algorithm for Forensic Applications. *IEEE Transactions System, Man, and Cybernetics: Systems*, 40(3), pp. 475-485.
- Zhu, J., Zheng, W. S., Lai, J. H. & Li, S. Z., 2014. Matching NIR face to VIS face using transduction. *IEEE Transactions on Information Forensics and Security*, 9(3), pp. 501-514.

