



UNIVERSITAS INDONESIA

**PENGEMBANGAN ALGORITMA *ADABOOST* BERBASIS *MULTI*
BOUNDARY CLASSIFIER UNTUK SEGMENTASI ORGAN JANIN PADA
CITRA ULTRASONOGRAFI**

TESIS

MUHAMMAD ANWAR MA'SUM

00906510382

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER**

DEPOK

2013



UNIVERSITAS INDONESIA

**PENGEMBANGAN ALGORITMA *ADABOOST* BERBASIS *MULTI*
BOUNDARY CLASSIFIER UNTUK SEGMENTASI ORGAN JANIN PADA
CITRA ULTRASONOGRAFI**

TESIS

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Ilmu
Komputer**

MUHAMMAD ANWAR MA'SUM

00906510382

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI MAGISTER ILMU KOMPUTER

DEPOK

2013

HALAMAN PERSETUJUAN

Judul : Pengembangan Algoritma *Adaboost* Berbasis *Multi Boundary Classifier* Untuk Segmentasi Organ Janin Pada Citra Ultrasonografi
Nama : Muhammad Anwar Ma'sum
NPM : 0906510382

Tesis ini telah di periksa dan disetujui.

23 Desember 2013



Dr. Eng. Wisnu Jatmiko S.T., M.Kom

Pembimbing Tesis

HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Muhammad Anwar Ma'sum

NPM : 0906510382

Tanda Tangan :  _____

Tanggal : 23 Desember 2013

HALAMAN PENGESAHAN

Tesis ini diajukan oleh :
Nama : Muhammad Anwar Ma'sum
NPM : 0906510382
Program Studi : Magister Ilmu Komputer
Judul Tesis : Pengembangan Algoritma *Adaboost* Berbasis *Multi Boundary Classifier* Untuk Segmentasi Organ Janin Pada Citra Ultrasonografi

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Ilmu Komputer pada Program Studi Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. Eng. Wisnu Jatmiko, S.T., M.Kom
Penguji : Ito Wasito, Ph.D
Penguji : Denny, S.Kom, MIT, Ph.D
Penguji : Prof. Dr. Ir. Aniati Murni Arymurthy, M.Sc.



Ditetapkan di : Depok
Tanggal : 8 Januari 2014

KATA PENGANTAR

Penulis mengucapkan puji dan syukur kehadiran Allah SWT, dengan izin-Nya penulis dapat menyelesaikan kegiatan penelitian dan penyusunan tesis dengan judul: Pengembangan Algoritma *Adaboost* Berbasis *Multi Boundary Classifier* Untuk Segmentasi Organ Janin Pada Citra Ultrasonografi.

Pada kesempatan ini, penulis ingin menyampaikan terima kasih yang kepada semua pihak yang telah ikut serta memberikan dukungan serta bantuan mencakup dorongan semangat dan moral, sehingga akhirnya kegiatan penelitian tugas akhir ini dapat berjalan lancar seperti sebagaimana mestinya. Penulis mengucapkan terima kasih kepada:

1. Kedua orang tua penulis Imam Ma'sum dan Sumarni, serta adik penulis yaitu Muhammad Zainal Abidin dan Hani'atul Khoiriah yang selalu memberikan dukungan moril maupun materil serta menyertai langkah penulis dalam setiap doa mereka.
2. Bapak Dr.Eng. Wisnu Jatmiko, S.T., M.Kom., selaku pembimbing tugas akhir yang telah mengarahkan penulis dalam melaksanakan kegiatan penelitian ini hingga proses penyusunan laporan.
3. Bapak Hisar Maruli Manurung, S.Kom, Ph.D, selaku Pembimbing Akademik yang telah memberikan arahan kepada penulis selama masa studi di Fakultas Ilmu Komputer, Universitas Indonesia.
4. Seluruh Staf Dosen Fasilkom yang telah membimbing dalam setiap mata kuliah yang penulis ambil.
5. Mas Ari, Mas Hilman, Mbak Mira, Mbak Ririn, Mas Wayan, Mas Nanda, Mas Adi, Mas Abie, Kholid, Mian, Andros, Grafika, Ricky, dan Rachmad selaku asisten riset lab 1231 yang telah menyalurkan ilmu dan membantu penulis dalam penelitian ini.
6. Rekan-rekan seperjuangan di Lab 1231 yang telah berbagi pengetahuan dan pengalaman selama kegiatan penelitian.

7. Mahasiswa Fasilkom Angkatan 2009 yang tidak bisa disebutkan satu persatu yang selama 4.5 tahun ini bersama-sama, terima kasih atas segala pengalaman baik suka dan duka yang telah diberikan
8. Semua pihak yang telah membantu penulis dalam pengerjaan tugas akhir ini.

Penulis menyadari masih terdapat kekurangan pada penyusunan laporan ini. Oleh karena itu, penulis dengan tangan terbuka bersedia menerima kritik dan saran yang berguna. Semoga karya ini bermanfaat bagi pembaca semua.

Depok, 27 Desember 2013

Penulis

Abstrak

Pengawasan pertumbuhan dan perkembangan janin merupakan hal yang perlu dilakukan guna menjaga kesehatan bayi, mencegah terjadinya pertumbuhan janin terhambat (PJT), dan juga kematian bayi. Dalam melakukan pengawasan tersebut dibutuhkan pengukuran data biometri janin. Pada penelitian sebelumnya telah dikembangkan metode aproksimasi elips berbasis *Hough Transform (HT)*. Akan tetapi pendekatan tersebut masih memiliki kekurangan karena *hit rate* yang mampu dicapai masih di bawah 70%. Proses aproksimasi dapat ditingkatkan dengan proses segmentasi terlebih dahulu. Oleh sebab itu penulis bermaksud melakukan penelitian untuk mengembangkan metode segmentasi objek organ janin pada citra USG. Dalam penelitian ini, proses segmentasi dilakukan dengan pendekatan deteksi objek. Untuk melakukan deteksi objek, penulis mengusulkan algoritma *Adaboost.MH* berbasis *Multi Boundary Classifier*. Metode tersebut digunakan untuk melakukan seleksi fitur *Haar* sekaligus membentuk *ensemble classifier* yang dipasangkan pada setiap fitur terpilih. Setelah dilakukan uji coba, metode ini mampu mencapai akurasi 93.9%, koefisien *kappa* 0.854 dan *Hamming loss error* 0.032 untuk dataset citra USG. Selain itu metode ini juga dibandingkan dengan algoritma *Adaboost.MH* dengan basis *classifier* lain. Hasil eksperimen menunjukkan bahwa metode yang diusulkan dapat unggul pada dataset (USG, USPS, dan MNIST). Setelah dilakukan proses deteksi organ janin dilakukan aproksimasi bentuk organ janin. Algoritma aproksimasi yang digunakan adalah *Randomized Hough Transform (RHT)*. Hasil aproksimasi dengan proses terlebih dahulu mencapai tingkat *hit* 81% untuk organ kepala, 57 % untuk organ perut, 72 % untuk organ tulang paha dan 66% untuk organ tulang lengan atas.

Kata Kunci : Segmentasi, organ janin, ultrasonografi, deteksi objek, *Adaboost.MH*, *Multi Boundary Classifier*, *Randomized Hough Transform*.

Abstract

Monitoring of fetal growth is a very important in order to keep the health of the baby, especially to avoid fetal growth disorder or even worse fetal mortality. Biometry data of the fetal is demanded during the monitoring. In the previous research, a method for ellipse approximation using Hough Transform (HT) have been developed. However, that approach still have some limitations because the hit rate is still below 70%. The result can be improved by applying segmentation before the approximation process is executed. Therefore, author intended to conduct a research to develop a segmentation method to for fetal detection from the images captured by USG. In this research, the segmentation process was done using object detection approach. In order to detect an object, a Multi Boundary Classifier based Adaboost.MH algorithm is proposed. This method was used to select Haar-like features as well as to construct an ensemble classifier which will be joined to each selected features. The experiment results showed that this proposed method can obtain detection accuracy of 93.9%, kappa coefficient of 0.854, and Hamming loss error of 0.032 for USG images dataset. Moreover, this proposed method was also compared to several Adaboost.MH based algorithms that uses different classifiers. From the experiment result it can be concluded that the proposed method showed improvement for two different datasets (USG, USPS, and MNIST). The detection process is followed by the approximation of fetal organs. The algorithm used for approximation is the Randomized Hough Transform (RHT). The approximation result showed that the proposed method can achieve hit ratio of 81% for fetal head approximation, 57% hit rate for fetal abdomen, 72% for femur, and 66% for fetal humerus approximation.

Keywords: *Segmentation, fetal organs, ultrasonography, object detection, Adaboost.MH, Multi Boundary Classifier, Randomized Hough Transform.*

Daftar Isi

HALAMAN PERSETUJUAN.....	Error! Bookmark not defined.
HALAMAN PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
HALAMAN PENGESAHAN	Error! Bookmark not defined.
KATA PENGANTAR	v
Abstrak	vii
Daftar Gambar.....	xiii
Daftar Tabel	xv
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan.....	5
1.4 Batasan Masalah.....	5
1.5 Manfaat.....	6
1.6 Posisi Penelitian	6
1.7 Metodologi Penelitian	9
Bab 2 Tinjauan Pustaka	11
2.1 <i>State of The Art</i>	11
2.2 Ultrasonografi (USG) dan Pengukuran Biometri Janin	12
2.3 Fitur <i>Haar</i>	14
2.4 <i>Ensemble Classifier</i>	15
2.5 <i>Adaboost</i>	18
2.5.1 <i>Binary Class Adaboost</i>	18
2.5.2 <i>Multi Class Adaboost</i>	21
2.6 <i>Base Classifier</i>	26
2.6.1 <i>Stump Algorithm</i>	26
2.6.2 <i>Product base Classifier</i>	28

2.6.3 <i>Learning Vector Quantization</i>	29
2.6.4 <i>Classification Tree</i>	32
2.9 Hough Transform Untuk Pengukuran Biometri	33
Bab 3 Rancangan dan Implementasi	38
3.1. Ide Dasar <i>Multi Boundary Classifier</i>	38
3.2. Perumusan Algoritma <i>Multi Boundary Classifier</i>	43
3.3. <i>Ensemble Multi Boundary Classifier</i> Menggunakan <i>Adaboost.MH</i>	49
Bab 4 Metodologi Penelitian	53
4.1 Dataset	53
4.1.1 Dataset Ultrasonografi (USG)	53
4.1.2 Dataset Benchmark	54
4.2 Kerangka Kerja Penelitian.	55
4.3 Ekstraksi Fitur <i>Haar</i>	57
4.3.1 Generate Fitur Haar	57
4.3.2 Penghitungan Nilai Fitur <i>Haar</i>	59
4.3 Seleksi Fitur Dan Klasifikasi	60
4.3.1 Seleksi Fitur dan <i>Ensemble Classifier</i>	60
4.3.1 Deteksi Objek	62
4.4 Aproksimasi Bentuk Organ Janin	62
4.4.1 <i>Thresholding</i>	62
4.4.2 <i>Randomize Hough Transform</i> (RHT)	63
4.5 Metode Evaluasi	64
4.5.1 Metode Evaluasi Klasifikasi	64
4.5.2 Metode Evaluasi Aproksimasi Organ Janin	67
4.6 Lingkungan Implementasi Program	68
Bab 5 Hasil Percobaan Dan Evaluasi Unjuk Kerja	69
5.1 Skenario Percobaan	69
5.1.1 Skenario Percobaan 1	69
5.1.2 Skenario Percobaan 2	70
5.1.3 Skenario Percobaan 3	70

5.2 Hasil Percobaan 1	70
5.3 Hasil Percobaan 2.....	75
5.3.1 Hasil Percobaan Dataset USPS.....	75
5.3.2 Hasil Percobaan Dataset MNIST	78
5.4 Hasil Percobaan 3.....	83
5.5 Evaluasi Hasil Percobaan 1	86
5.5.1 Perbandingan Performa Metode pada Dataset USG Dilihat dari Faktor <i>Hamming Error</i>	87
5.5.2 Perbandingan Performa Metode pada Dataset USG Dilihat dari Faktor Akurasi.....	88
5.5.3. Perbandingan Performa Metode pada Dataset USG Dilihat dari Faktor <i>Kappa</i>	90
5.5.4. Perbandingan Waktu <i>Training</i> Metode pada Dataset USG.....	92
5.6 Evaluasi Hasil Percobaan 2.1	94
5.6.1 Perbandingan Performa Metode pada Dataset USPS Dilihat dari Faktor <i>Hamming Error</i>	95
5.6.2. Perbandingan Performa Metode pada Dataset USPS Dilihat dari Faktor Akurasi.....	96
5.6.3 Perbandingan Performa Metode pada Dataset USPS Dilihat dari Faktor <i>Kappa</i>	98
5.7. Evaluasi Hasil Percobaan 2.2	100
5.7.1. Perbandingan Performa Metode pada Dataset MNIST Dilihat dari Faktor <i>Hamming Error</i>	101
5.7.2. Perbandingan Performa Metode pada Dataset MNIST Dilihat dari Faktor Akurasi.....	102
5.7.3 Perbandingan Performa Metode pada Dataset MNIST Dilihat dari Faktor <i>Kappa</i>	104
5.8. Evaluasi Hasil Percobaan 3	105
5.9. Fitur Terbaik Yang Diperoleh	107
Bab 6 Kesimpulan Dan Saran	109
6.1 Kesimpulan.....	109
6.2 Saran.....	110

Referensi	111
-----------------	-----

Daftar Gambar

Gambar 1-1 : Gambaran Umum Tahapan Penelitian.....	7
Gambar 1-2 : Posisi Penelitian.....	8
Gambar 2-1 Contoh Alat USG.....	12
Gambar 2-2 Pengukuran Data Biometri Janin (a) BPD dan HC (b) AC (c) FL (d) HL (e) CRL.	14
Gambar 2-3 Fitur dasar Haar	14
Gambar 2-4 Contoh pengaplikasian fitur Haar dasar pada citra USG	15
Gambar 2-5 Ilustrasi Pemodelan Klasifikasi [18].....	16
Gambar 2-6 Ilustrasi Cara Kerja Adaboost.....	20
Gambar 2-7 Pseudocode Algoritma Adaboost.....	21
Gambar 2-8 Pseudocode Algoritma Adaboost.MH	26
Gambar 2-9 Ilustrasi Algoritma Stump.....	27
Gambar 2-10 Arsitektur Algoritma Klasifikasi Learning Vector Quantization (LVQ)	30
Gambar 2-11 Ilustrasi representasi LVQ untuk di-ensemble menggunakan Adaboost. (a) Representasi LVQ dengan dikomposisi menjadi binary classification (b) Representasi LVQ tanpa didekomposisi menjadi binary classification.....	30
Gambar 2-12 Ilustrasi arsitektur Classification Tree	32
Gambar 2-13 Prinsip dari metode Hough Transform yang mentransformasi semua titik pada koordinat (x, y) ke dalam ruang Hough (m, n)	34
Gambar 2-14 Pseudocode Algoritma Hough Transform Untuk Garis	35
Gambar 2-15 Pseudocode Algoritma Hough Transform Untuk Elips	37
Gambar 3-1 Lima fitur terbaik dari dataset USG.....	39
Gambar 3-2 Konsep dasar algoritma Multi Boundary Classifier	40
Gambar 3-3 Simulasi Perbedaan Proses Klasifikasi (a) Menggunakan multi boundary classifier. (b) Menggunakan multi stump classifier	42
Gambar 3-4 Ilustrasi pembentukan Boundary dari algoritma Multi Boundary Classifier	44
Gambar 3-5 Pseudo code algoritma Multi Boundary Classifier	48
Gambar 3-6 Diagram algoritma Adaboost.MH berbasis Multi Boundary Classifier .	52
Gambar 4-1 Contoh tampilan organ janin dalam citra USG, kepala (kiri atas), perut (kanan atas), femur (kiri bawah), dan humerus (kanan bawah).....	54
Gambar 4-2 Dataset Benchmark USPS dan MNIST.	55
Gambar 4-3 Kerangka Kerja Penelitian	56
Gambar 4-4 Proses Generate Fitur Haar	58

Gambar 4-5 (a) Proses perhitungan nilai integral image (b) Perhitungan nilai fitur haar menggunakan integral image	59
Gambar 4-6 Ilustrasi ensemble Multi Boundary Clasifier dengan algoritma Adaboost.MH.....	61
Gambar 4-7 Hasil deteksi objek organ pada citra USG	62
Gambar 4-8 Hasil Konversi Citra dari Grayscale ke Binary Image (a) Kepala (b) Perut (c) Tulang paha (d) Tulang Lengan	63
Gambar 4-9 Aproksimasi elips dan garis pada organ janin (a) pada ROI (b) Hasil dari ROI dipalikasikan ke citra asli.	64
Gambar 5-1 Hasil Deteksi Organ Janin	71
Gambar 5-2 Perbandingan Nilai Hamming Error pada dataset USG	72
Gambar 5-3 Perbandingan Nilai Akurasi pada dataset USG	73
Gambar 5-4 Perbandingan Nilai Kappa pada dataset USG	74
Gambar 5-5 Perbandingan Hamming Error pada dataset USPS.....	76
Gambar 5-6 Perbandingan Akurasi pada dataset USPS.....	77
Gambar 5-7 Perbandingan Nilai Kappa pada dataset USPS.....	78
Gambar 5-8 Perbandingan Nilai Hamming Error pada dataset MNIST.	80
Gambar 5-9 Perbandingan Nilai Akurasi pada dataset MNIST.....	81
Gambar 5-10 Perbandingan nilai koefisien kappa pada dataset MNIST.	82
Gambar 5-11 Contoh Hasil Deteksi dan Aproksimasi Organ Janin	85
Gambar 5-12 Perbandingan nilai performa metode klasifikasi pada dataset USG.....	86
Gambar 5-13. Perbandingan waktu training metode klasifikasi	93
Error! Bookmark not defined.	
Gambar 5-14 Perbandingan performa metode klasifikasi pada dataset USPS	94
Gambar 5-15 Perbandingan nilai koefisien kappa pada dataset MNIST.	100
Gambar 5-16 Aproksimasi organ janin yang gagal hit (a)Kepala (b) Perut (c) Tulang Paha (d) Tulang Lengan.....	107

Daftar Tabel

Tabel 4-1 Jumlah Fitur Haar yang Digenerate Untuk Masing-Masing Dataset	59
Tabel 4-2 Intepretasi Nilai Kappa.....	66
Tabel 4-3 Contoh Matriks Perbandingan Pair-Wise Comparison	67
Tabel 4-4 Lingkungan Implementasi Program	68
Tabel 5-1 Algoritma yang akan dibandingkan pada percobaan 1.....	69
Tabel 5-2 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor HammingError.....	87
Tabel 5-3 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor HammingError.....	88
Tabel 5-4 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor Akurasi.....	89
Tabel 5-5 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor Akurasi.....	89
Tabel 5-6 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor Kappa.....	90
Tabel 5-7 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor Kappa.....	91
Tabel 5-8 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor HammingError.....	95
Tabel 5-9 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor HammingError.....	96
Tabel 5-10 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor Akurasi.....	97
Tabel 5-11 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor Akurasi	97
Tabel 5-12 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor Kappa.....	98
Tabel 5-13 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor Kappa	99
Tabel 5-14 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor HammingError.....	101
Tabel 5-15 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor HammingError	102

Tabel 5-16 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor Akurasi.....	103
Tabel 5-17 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor Akurasi	103
Tabel 5-18 Matriks Head to Head Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor Kappa.....	104
Tabel 5-19 Ringkasan Head to Head Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor Kappa	105
Tabel 5-20 Hit rate aproksimasi organ janin.....	106
Tabel 5-21 Sepuluh fitur terbaik untuk klasifikasi citra USG (window 20x20).....	108

Daftar Notasi

Berikut adalah notasi yang sering digunakan dalam laporan tesis ini

Notasi	Istilah	Keterangan
$x_{i,j}$	Nilai fitur sampel	Nilai fitur ke-j dari sampel ke-i
$y_{i,l}$	Label kelas sampel	Nilai label kelas ke-l dari sampel ke-i
$w_{i,l}$	Bobot sampel	Bobot sampel ke-i pada kelas ke-l
n	Jumlah sampel	
K	Jumlah kelas	
$h(x)$	<i>Weak classifier</i>	<i>Classifier-classifier</i> dengan performa rendah (digabungkan dalam proses <i>ensemble</i> untuk mendapatkan hasil <i>classifier</i> yang memiliki performa tinggi
$H(x)$	<i>Strong classifier</i>	Hasil gabungan <i>weak classifier</i>
α	Bobot / koefisien <i>weak classifier</i>	Merepresentasikan nilai bobot voting pada <i>weak classifier</i> yang bersangkutan
E_H	<i>Hamming error</i>	Jumlahan <i>error</i> yang dihitung pada masing-masing <i>binary classifier</i>
E_Z	<i>Margin error</i>	Jumlahan eksponensial <i>error</i> yang dihitung pada masing-masing <i>binary classifier</i>
Z	<i>Energy</i>	Jumlahan eksponensial dari <i>error</i> yang dihitung pada masing-masing <i>binary classifier</i> dikalikan dengan <i>bobot classifier</i> (α)
μ_{l-}	<i>Half edge</i> negatif	Jumlah bobot sampel yang diklasifikasikan ke kelas yang salah oleh <i>binary classifier</i> ke-l
μ_{l+}	<i>Half edge</i> positif	Jumlah bobot sampel yang diklasifikasikan ke kelas yang benar oleh <i>binary classifier</i> ke-l
v_l	<i>Vektor vote</i>	Nilainya ada dua kemungkinan, -1, atau +1, Merepresentasikan daerah boundary kelas ke-l
γ_l	<i>Classification edge</i>	Jumlahan bobot sampel dikalikan dengan benar tidaknya prediksi dari <i>classifier</i> ke-l (kalai benar dilaikan +1, kalau salah dikalikan -1)
ϕ_l	<i>Binary function / binary classifier</i>	<i>Binary function / binary classifier</i> kelas ke-l
T	Jumlah <i>classifier</i> yang di <i>ensemble</i>	Jumlah <i>classifier</i> yang di <i>ensemble</i> pada proses <i>Adaboost</i> . merepresentasikan jumlah iterasi pada <i>Adaboost</i>
(x_0, x_0)	Titik pusat elips	
(a, b)	Sumbu mayor dan minor elips	

Bab 1 Pendahuluan

1.1 Latar Belakang

Lembaga Survey Demografi Kesehatan Indonesia (SDKI) tahun 2007 mengatakan bahwa angka kematian bayi masih tinggi, yaitu mencapai 34 kematian per 1000 kelahiran hidup [1]. Tingginya angka kematian bayi tersebut disebabkan oleh kurangnya kesadaran masyarakat dalam melakukan pemeriksaan dan perawatan kandungan. Dengan demikian, apabila terjadi anomali pada pertumbuhan dan perkembangan janin dalam kandungan, maka hal itu tidak bisa dideteksi sejak awal dan diatasi. Fakta tersebut menginformasikan bahwa pengawasan kandungan merupakan hal yang sangat penting guna menjamin pertumbuhan dan perkembangan bayi yang baik. Melalui pengawasan yang rutin, adanya anomali atau pertumbuhan dan perkembangan bayi yang terhambat bisa diketahui sejak dini, untuk dilakukan upaya penanganan medis selanjutnya. Oleh sebab itu, pemeriksaan dan pengawasan kandungan yang rutin diharapkan dapat mengurangi angka kematian bayi.

Untuk melakukan pemeriksaan janin dalam kandungan dibutuhkan suatu alat yang dapat menghasilkan gambar kondisi janin dalam kandungan. Salah satu alat yang sering digunakan untuk keperluan pemeriksaan kandungan tersebut adalah ultrasonografi (USG). Alat ini bekerja menggunakan prinsip pantulan gelombang ultrasonik. Pertama kali, alat ini mentransmisikan gelombang ultrasonik ke dalam tubuh pasien. Selanjutnya alat ini akan menganalisis pantulan gelombang yang ditransmisikan tadi. Melalui informasi intensitas gelombang pantulan dan rentang waktu transmisi dengan pantulan gelombang diterima kembali, alat ini dapat menghasilkan gambar kondisi bagian tubuh yang diamati. Selanjutnya gambar tadi divisualisasikan pada monitor untuk dianalisis.

Alat ultrasonografi memiliki peranan yang cukup penting dalam dunia medis. Selain digunakan dalam pemeriksaan kandungan, ultrasonografi (USG) juga digunakan dalam pemeriksaan organ tubuh yang lain, seperti jantung, paru-paru, payudara, dan prostat [2]. Alat ini tergolong aman digunakan karena tidak

menggunakan pancaran radiasi gelombang dengan frekuensi sangat tinggi seperti sinar X dan sinar laser yang dapat membahayakan pasien. Selain itu alat ini juga dapat menggambarkan kondisi dalam tubuh pasien tanpa dilakukan pembedahan. Disamping itu, harga alat ini juga relatif lebih murah dibandingkan dengan alat pemeriksaan lainnya.

Pertumbuhan dan perkembangan janin dapat diamati melalui data biometri janin. Data biometri merupakan besaran fisik dari kondisi janin itu sendiri, yang meliputi *Crown Rump Length* (CRL), *Biparietal Diameter* (BPD), *Head Circumference* (HC), *Femur Length* (FL) dan *Abdominal Circumference* (AC) [3]. Besaran-besaran ini nantinya akan diukur pada setiap pemeriksaan janin. Perkembangan dan pertumbuhan janin nantinya akan diukur dari proses perubahan data-data tersebut dari satu pemeriksaan ke pemeriksaan selanjutnya.

Untuk mendapatkan data-data biometri diperlukan pengukuran terhadap janin yang sedang diamati. Saat ini, kebanyakan pengukuran dilakukan secara manual oleh tenaga medis. Pengukuran jenis ini cenderung susah dilakukan, misalnya saja untuk mengukur keliling kepala yang bentuknya menyerupai elips. Seiring dengan perkembangan ilmu dan teknologi, seharusnya pengukuran data biometri tersebut dapat dilakukan secara otomatis melalui program komputer. Akan tetapi, ada berbagai tantangan untuk upaya otomasi pengukuran data biometri tersebut. Diantaranya adalah data USG memiliki nilai kontras yang rendah, memiliki noise yang tinggi, dan kesamaan tekstur yang menyebabkan segmentasi gambar janin susah dilakukan [4].

Pada penelitian sebelumnya oleh Putu Satwika, dilakukan aproksimasi bentuk elips pada organ kepala janin [5]. Pada penelitian tersebut digunakan algoritma aproksimasi berbasis *Hough Transform* untuk kurva elips. Algoritma yang diujicobakan adalah *Randomized Hough Transform (RHT)*, *Iterative Randomize Hough Trnsform (IRHT)*, dan *Eliminating Particle Swarm Optimization Hough Transform (EPSO-HT)* [6][7][8]. Dalam penelitian tersebut juga diusulkan algoritma baru yaitu *Particle Swarm Optimization* berbasis *Randomized Hough Transform* (PSO-RHT). Dalam penelitian tersebut, disebutkan bahwa algoritma-algoritma aproksimasi

tersebut memiliki kelemahan. Pertama, algoritma-algoritma ini memiliki resiko gagal jika noise dalam gambar terlalu tinggi. Kedua, algoritma tersebut akan memiliki *running time* yang cukup lama, jika diaplikasikan pada gambar USG secara penuh. Oleh karena itu perlu dilakukan proses segmentasi terlebih dahulu sebelum dilakukan proses aproksimasi.

Salah satu metode segmentasi yang dapat digunakan dalam kasus ini adalah dengan pendekatan deteksi objek. Ide dasarnya, objek-objek organ janin yang ada pada citra ultrasonografi dideteksi terlebih dahulu baru selanjutnya dilakukan aproksimasi dan pengukuran biometri. Pendekatan deteksi terlebih dahulu diharapkan dapat memperkecil resiko gagal saat aproksimasi, dan pada akhirnya meminimalisir kesalahan pengukuran biometri. Selain itu, pendeteksian objek juga diharapkan dapat mempersingkat waktu pengukuran, karena hanya diterapkan pada area tempat objek terdeteksi saja.

Salah satu algoritma yang populer untuk digunakan dalam kasus deteksi objek pada citra adalah *Adaboost*. Algoritma tersebut merupakan jenis algoritma *multi classifier* yang merupakan gabungan (*ensembles*) dari *classifier-classifier* kecil, disebut *weak classifier*. *Adaboost* pertama kali diusulkan oleh Freund, Y., dan Schapire, R. E. pada tahun 1997 [9]. Pada saat itu algoritma *Adaboost* terbatas pada kasus *binary classification* atau klasifikasi dengan keluaran dua kelas saja. Selanjutnya, pada tahun 1999 *Adaboost* dikembangkan oleh Schapire, R. E., dan Singer, Y yang dapat menangani kasus *ensemble classifie* multi *class* [10]. Dalam aplikasinya, algoritma *Adaboost* dipopulerkan oleh Viola dan Jones dalam aplikasi pendeteksi wajah pada citra [11]. Dalam aplikasi tersebut *Adaboost* digunakan untuk seleksi fitur *Haar* sekaligus melakukan penggabungan (*ensemble*) *classifier-classifier* yang dipasangkan dalam setiap fitur *Haar*.

Berdasarkan permasalahan tersebut penulis berkeinginan untuk mengembangkan metode pendeteksian objek multi kelas pada citra USG. Pendeteksian multi objek ini dimaksudkan agar sistem yang dibuat selain bisa mendeteksi organ-organ janin pada citra, juga bisa membedakan gambar organ kepala, *femur*, *humerus*, dan perut pada citra USG. Diharapkan dengan metode ini

nantinya pendeteksian dapat dilakukan secara online saat *scanning* gambar. Metode yang diajukan untuk melakukan proses pendeteksian adalah *Adaboost* yang menggabungkan beberapa *weak classifier* berupa *Multi Boundary Classifier* yang diusulkan sendiri oleh penulis sebagai kontribusi dalam penelitian ini. Metode *Multi Boundary Classifier* merupakan metode klasifikasi sederhana yang dikembangkan dari *Stump Classifier* yang diusulkan oleh Balazs dan Robert [12]. Selanjutnya, proses pendeteksian dilanjutkan dengan aproksimasi bentuk organ janin. Untuk proses aproksimasi organ janin akan digunakan metode *Randomize Hough Transform* (RHT). Pengembangan metode pada penelitian ini diharapkan memberikan manfaat kepada tenaga medis untuk melakukan pemeriksaan dengan lebih cepat (otomatis) dan akurat.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, penulis merumuskan permasalahan penelitian sebagai berikut:

1. Bagaimanakah representasi metode *Adaboost* berbasis *Multi Boundary Classifier* untuk melakukan seleksi fitur dan klasifikasi objek organ janin pada citra USG.
2. Bagaimanakah performa metode *Adaboost* berbasis *Multi Boundary Classifier* digunakan untuk seleksi fitur dan klasifikasi objek organ janin pada citra USG.
3. Bagaimanakah performa metode *Adaboost* berbasis *Multi Boundary Classifier* dibandingkan dengan metode lain untuk melakukan seleksi fitur dan klasifikasi.
4. *Haar* fitur jenis mana sajakah yang paling merepresentasikan objek organ janin pada citra USG.
5. Seberapa besar tingkat keberhasilan aproksimasi bentuk organ janin setelah dilakukan segmentasi menggunakan *Adaboost* berbasis *Multi Boundary Classifier*.

1.3 Tujuan

Adapun tujuan utama dari penelitian ini adalah membuat sistem pendeteksian dan pengukuran organ janin pada citra ultrasonografi. Selain itu penelitian ini juga memiliki tujuan sebagai berikut:

1. Mengembangkan metode *Adaboost* berbasis *Multi Boundary Classifier* untuk melakukan seleksi fitur dan klasifikasi objek organ janin pada citra USG.
2. Mengetahui performa metode *Adaboost* berbasis *Multi Boundary Classifier* digunakan untuk seleksi fitur dan klasifikasi objek organ janin pada citra USG.
3. Mengetahui performa metode *Adaboost* berbasis *Multi Boundary Classifier* dibandingkan dengan metode lain untuk melakukan seleksi fitur dan klasifikasi.
4. Mengetahui jenis *Haar* fitur yang paling merepresentasikan objek organ janin pada citra USG.
5. Mengetahui tingkat keberhasilan aproksimasi bentuk organ janin setelah dilakukan segmentasi menggunakan *Adaboost* berbasis *Multi Boundary Classifier*.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

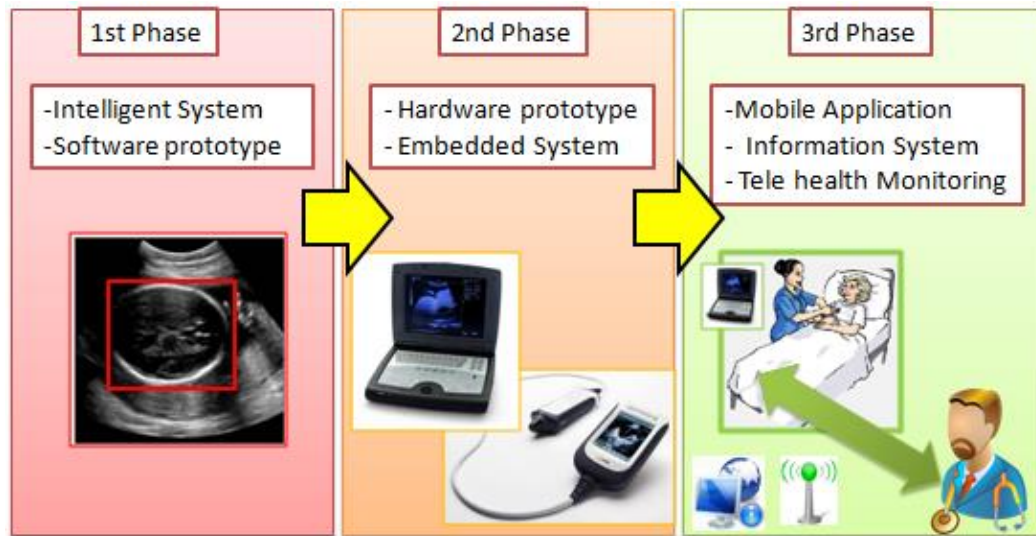
1. Citra USG yang digunakan adalah citra USG dua dimensi (2D) trimester ketiga. Hal ini dikarenakan keterbatasan dataset yang digunakan selama penelitian.
2. Organ janin yang akan dideteksi adalah kepala, femur, homerus, dan perut.
3. Penelitian berfokus pada pengembangan algoritma untuk deteksi objek organ janin pada citra ultrasonografi.
4. Setelah dilakukan pendeteksian objek organ janin akan dilakukan aproksimasi bentuk organ janin. Bentuk yang digunakan adalah elips untuk kepala dan perut, serta garis untuk femur dan humerus.

1.5 Manfaat

Penelitian ini bertujuan mengembangkan suatu metode pendeteksian dan pengukuran besaran biometri janin. Sistem yang dikembangkan ini diharapkan dapat digunakan untuk membantu proses pengukuran biometri janin secara berkala. Melalui pengukuran besaran biometri secara berkala dapat dilakukan prediksi apakah pertumbuhan dan perkembangan janin berlangsung secara normal atau ada suatu hambatan. Dengan demikian, apabila ada kelainan maupun gejala pertumbuhan bayi terhambat (PJT), maka dapat diketahui sedini mungkin untuk segera diatasi. Pendekatan pencegahan PJT tersebut diharapkan dapat mengurangi angka kematian bayi lahir di Indonesia. Selain itu, metode yang dikembangkan akan diimplementasikan dalam suatu *prototype* perangkat lunak yang berjalan di *personal computer* (PC). Dengan demikian, peluang hasil penelitian ini untuk diimplementasikan menjadi suatu produk aplikasi siap pakai sangatlah besar.

1.6 Posisi Penelitian

Penelitian ini merupakan bagian dari penelitian pengembangan dan perancangan alat *Portable Smart* Ultrasonografi. Target akhir dari rangkaian penelitian tersebut adalah produk jadi (siap pakai) perangkat ultrasonografi cerdas. USG cerdas ini merupakan perangkat *embedded system* yang terdiri dari *hardware* dan *software*. *Hardware* dalam USG tersebut terdiri dari sensor (*probe*), *controller*, layar visualisasi dan komponen elektronik lainnya. Sedangkan *software* merupakan program-program untuk menjalankan fungsi USG, yang juga memuat kecerdasan buatan pada perangkat USG tersebut. Selanjutnya, alat juga didukung dengan sistem informasi pemeriksaan janin sehingga dapat digunakan untuk melakukan pemeriksaan jarak jauh (*tele health*). Hal ini dimaksudkan agar sistem pemeriksaan kandungan dapat digunakan oleh seluruh masyarakat di Indonesia, terkait dengan keterbatasan fasilitas alat pemeriksaan, jumlah dokter spesialis kandungan, dan distribusi dokter spesialis kandungan itu sendiri. Melalui sistem tersebut, diharapkan seluruh masyarakat, baik yang ada di kota maupun di daerah pelosok dapat mendapatkan pelayanan kesehatan yang memadai.

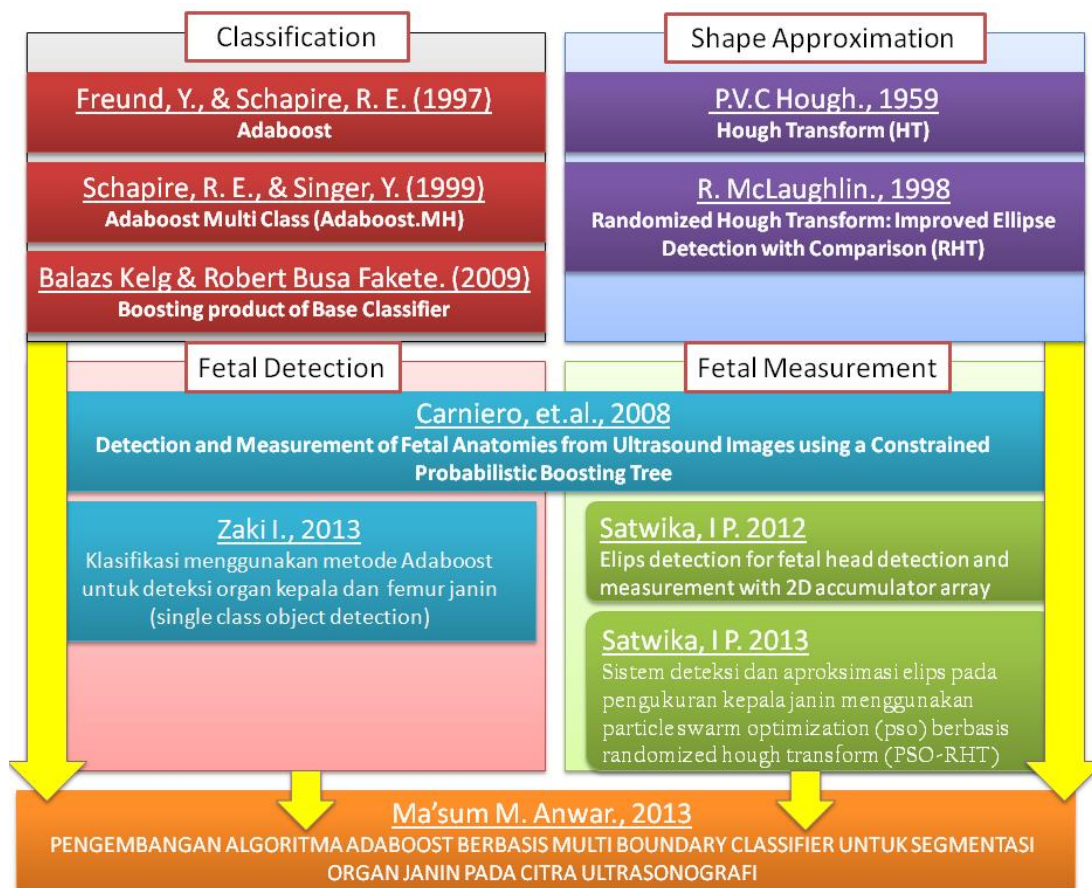


Gambar 1-1 : Gambaran Umum Tahapan Penelitian

Diagram pelaksanaan penelitian diilustrasikan oleh gambar 1.1. Dalam diagram tersebut dapat dilihat bahwa ada tiga fase utama dalam melaksanakan rangkaian penelitian. Saat ini, fokus penelitian adalah pada fase pertama yaitu pengembangan suatu sistem cerdas (*intelligent system*) untuk melakukan pendeteksian dan pengukuran besaran biometri janin. Pengembangan metode *intelligent system* ini nantinya akan diimplementasikan sebagai suatu *software* yang berjalan di PC atau platform lainnya.

Pada penelitian ini akan dilakukan pengembangan metode *machine learning* untuk mendeteksi organ-organ janin pada citra USG. Pendekatan yang digunakan adalah *multi object detection* dengan klasifikasi. Melalui metode yang akan dikembangkan ini, cukup dibuat satu *classifier* saja untuk mendeteksi beberapa objek berbeda dalam citra USG. Dengan pendekatan *multi object detection* ini diharapkan pendeteksian yang dilakukan lebih efisien dibandingkan dengan pendekatan *single object detection*. Selain itu dengan pendekatan *multi object detection* dapat mengurangi risiko salah deteksi pada citra.

Pada penelitian sebelumnya, telah dikembangkan beberapa metode untuk pendeteksian dan pengukuran organ janin. Beberapa penelitian tersebut digambarkan pada diagram posisi penelitian (gambar 1.2).



Gambar 1-2 : Posisi Penelitian

Beberapa topik penelitian yang terkait dengan penelitian ini dapat dikelompokkan menjadi empat topik utama. Pertama, adalah tentang pengembangan metode klasifikasi dengan menggunakan pendekatan *ensemble boosting* (*Adaboost*). Versi pertama *Adaboost* (tahun 1997) digunakan untuk menggabungkan (*ensemble*) kasus *binary classification*. Metode *Adaboost* dikembangkan lagi pada tahun 1999 untuk menangani kasus klasifikasi *multi class*. Topik kedua yang berkaitan dalam penelitian ini adalah deteksi objek organ janin pada citra ultrasonografi. Salah satu penelitian sebelumnya dilakukan oleh Carneiro dkk. Penelitian ini menggunakan metode *Probabilistic Boosting Tree* untuk melakukan klasifikasi. Selain itu Imaduddin juga melakukan penelitian serupa, menggunakan algoritma *Adaboost*. Kedua penelitian tersebut menggunakan fitur *Haar* [13]. Meskipun demikian kedua penelitian tersebut menggunakan pendekatan *binary classifier*, yang mana kurang

cocok untuk dataset citra ultrasonografi yang penulis gunakan karena ada dua objek organ janin sekaligus pada satu citra. Topik penelitian lainnya adalah metode aproksimasi kurva berbasis *Hough Transform* diantaranya *Hough Transform (HT)*, *Randomize Hough Transform (RHT)* atau *Eliminating Particle Swarm Optimization Hough Transform (EPSO-HT)*. Pada penelitian yang dilakukan sebelumnya, metode aproksimasi berbasis *Hough Transform* tersebut digunakan untuk mengaproksimasi bentuk organ kepala janin [5].

Fokus utama dari penelitian ini adalah mengembangkan metode *machine learning* untuk pendeteksian multi objek pada citra USG. Penelitian ini merupakan lanjutan dari penelitian sebelumnya, yaitu dengan mengembangkan metode klasifikasi pendeteksian multi objek pada citra. Dengan demikian *classifier* yang dibentuk dapat dilakukan untuk mendeteksi berbagai jenis organ janin sekaligus. Keuntungan dari pendekatan deteksi objek *multi class* ini yaitu pertama, tidak perlu dibuat *classifier* yang berbeda untuk objek organ janin yang berbeda. Kedua, dengan melakukan pembelajaran untuk beberapa objek sekaligus, *classifier* mengurangi risiko salah deteksi antar jenis objek organ janin atau antara objek organ janin dengan *background*. Selanjutnya dilakukan aproksimasi bentuk organ janin setelah dideteksi. Untuk organ kepala dan perut akan diaproksimasi dengan bentuk elips, sedangkan organ *femur* dan *humerus* diaproksimasi dengan bentuk garis. Metode aproksimasi yang digunakan adalah *Randomized Hough Transform (RHT)*. Metode RHT dipilih karena pada kasus data USG memiliki performa yang cukup bagus, dan membutuhkan waktu yang cepat dalam melakukan aproksimasi [5].

1.7 Metodologi Penelitian

1. Studi literatur

Tahapan ini penulis akan melakukan kajian terhadap berbagai penelitian yang telah dilakukan oleh orang lain dan telah dipublikasikan. Selain itu penulis juga melakukan studi literatur terhadap beberapa metode yang sering digunakan dan paling cocok untuk diimplementasikan pada penelitian ini. Literatur ini diperoleh dari berbagai sumber ilmiah seperti buku, tesis, disertasi dan artikel baik dalam

jurnal ilmiah dalam dan luar negeri. Selain itu, informasi-informasi yang bersifat tambahan juga dapat diperoleh melalui pemanfaatan jaringan internet untuk menambah pengetahuan penulis.

2. Pengumpulan data

Proses ini adalah proses untuk mengumpulkan data yang dibutuhkan untuk perancangan sistem yang akan dibangun. Data-data tersebut meliputi data USG 2D janin dalam kandungan pada trimester kedua yang disimpan dalam bentuk citra digital. Selain data USG juga digunakan data *benchmark* untuk menguji performa metode klasifikasi yang dikembangkan.

3. Perancangan sistem dan implementasi sistem

Pada tahap ini, penulis melakukan perancangan terhadap sistem yang akan dibangun dengan berdasarkan pada metode yang akan dikembangkan. Setelah proses perancangan selesai selanjutnya hasil perancangan sistem diimplementasikan pada suatu *software* komputer sehingga mempermudah proses eksperimen dan analisis sistem yang dikembangkan.

4. Eksperimen dan analisis sistem

Pada tahapan ini dilakukan percobaan terhadap data yang telah dikumpulkan pada tahap sebelumnya dengan beberapa skenario yang telah direncanakan. Kemudian selanjutnya hasil dari percobaan akan dianalisis dan dimuat dalam laporan.

5. Penulisan laporan

Penulisan laporan merupakan bagian terakhir dari metode penelitian ini. Penulis akan memuat semua hasil penelitian dan eksperimen yang telah dilakukan.

Bab 2 Tinjauan Pustaka

2.1 *State of The Art*

Pengawasan pertumbuhan dan perkembangan janin merupakan hal yang perlu dilakukan guna menjaga kesehatan bayi, mencegah terjadinya pertumbuhan janin terhambat (PJT) dan juga kematian bayi. Dalam melakukan pengawasan tersebut dibutuhkan pengukuran data biometri janin. Pada penelitian sebelumnya, dikembangkan algoritma deteksi dan aproksimasi elips berbasis *Hough Transform* pada organ kepala janin [5]. Akan tetapi pendekatan ini memiliki kekurangan, diantaranya adalah masih rendahnya *hit rate* aproksimasi yang dilakukan. Hal ini dikarenakan *noise* pada citra USG terlalu tinggi, dan tidak bisa dihilangkan hanya dengan menggunakan metode *thresholding*. Hal ini juga dikonfirmasi dengan tingkat *hit* yang masih dibawah 70%. Salah satu solusi yang dapat digunakan untuk menanggulangi masalah ini adalah dengan melakukan segmentasi pada citra janin. Salah satu pendekatan segmentasi yang sering digunakan dalam proses segmentasi adalah dengan menggunakan prinsip deteksi objek. Dengan proses deteksi terlebih dahulu, diharapkan proses aproksimasi dapat dilakukan pada area tempat objek berada. Dengan demikian kemungkinan aproksimasi berhasil akan semakin tinggi.

Pada penelitian ini, penulis bermaksud mengembangkan metode deteksi objek untuk melakukan segmentasi organ janin pada citra USG. Metode deteksi objek yang dikembangkan akan dibuat sedemikian hingga dapat mendeteksi beberapa objek yang berbeda-beda. Dengan demikian dapat digunakan untuk mendeteksi beberapa macam organ janin, baik kepala, perut, tulang paha dan tulang lengan atas. Metode yang digunakan untuk deteksi objek adalah dengan membuat suatu *ensemble classifier*. *Base classifier* yang akan dibentuk adalah *Multi Boundary Classifier*, yaitu *classifier* yang membentuk beberapa *boundary* berdasarkan data pelatihan yang diberikan. Selanjutnya *base classifier* tersebut akan digabungkan (*di-ensemble*) menggunakan algoritma *Adaboost.MH*. Setiap *base classifier* yang *di-ensemble* akan dipasangkan dengan satu fitur, sedemikian hingga, proses *ensemble* juga merupakan seleksi fitur.

Setelah dibentuk *classifier*, selanjutnya *classifier* tersebut digunakan untuk melakukan deteksi objek organ janin yang ada pada citra USG. Setelah dideteksi, selanjutnya dilakukan aproksimasi bentuk organ janin. Untuk organ kepala dan perut akan diaproksimasi dengan bentuk elips, sedangkan untuk organ tulang paha dan tulang lengan atas akan diaproksimasi dengan menggunakan bentuk kurva garis.

2.2 Ultrasonografi (USG) dan Pengukuran Biometri Janin

Ultrasonografi merupakan suatu kegiatan mengamati suatu objek biasanya adalah organ tubuh manusia menggunakan alat yang menggunakan gelombang ultrasonik sebagai sensor [2]. Alat ini tergolong aman digunakan karena tidak menggunakan pancaran radiasi gelombang dengan frekuensi sangat tinggi seperti sinar X dan sinar laser yang dapat membahayakan pasien. Cara kerja alat ini tidak jauh beda dengan alat pemeriksaan pasien lainnya. Pertama kali, alat ini mentransmisikan gelombang ultrasonik ke dalam tubuh pasien melalui *transducer*. Selanjutnya alat ini akan menganalisis pantulan gelombang yang ditransmisikan tadi. Melalui informasi intensitas gelombang pantulan dan rentang waktu transmisi dengan pantulan gelombang diterima kembali, alat ini dapat menghasilkan gambar kondisi bagian tubuh yang diamati. Selanjutnya gambar tadi divisualisasikan pada monitor untuk dianalisis. Dengan demikian alat ini dapat menggambarkan kondisi dalam tubuh pasien tanpa dilakukan pembedahan. Contoh gambar alat USG dapat dilihat pada gambar 2.1



Gambar 2-1 Contoh Alat USG

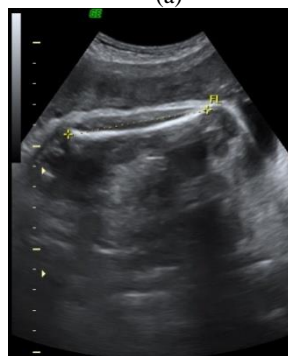
Dari pasien, nantinya akan dihasilkan gambar janin dalam kandungan pasien. Gambar yang diambil biasanya meliputi empat macam bagian. Lima bagian janin tersebut adalah bagian kepala (*head*), perut (*abdomen*), paha (*femur*), *humerus* (lengan tangan) dan tulang belakang (dari kepala sampai tulang ekor). Dari masing-masing bagian tersebut akan dicari data biometri. Dari kelima bagian tersebut akan dihitung besaran *Crown Rump Length* (CRL) yaitu panjang tubuh dari kepala sampai tulang ekor, *Biparietal Diameter* (BPD), diameter lingkaran kepala, *Head Circumference* (HC) yaitu keliling lingkaran kepala, *Femur Length* (FL) dan *Abdominal Circumference* keliling lingkaran perut (AC), dan *Humerus Length* (HL). Gambar Pengukuran data biometri janin dapat dilihat pada gambar 2.2.



(a)



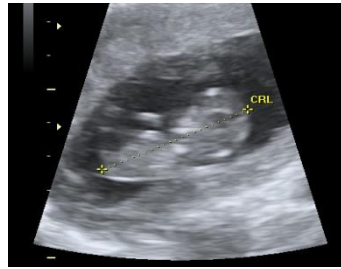
(b)



(c)



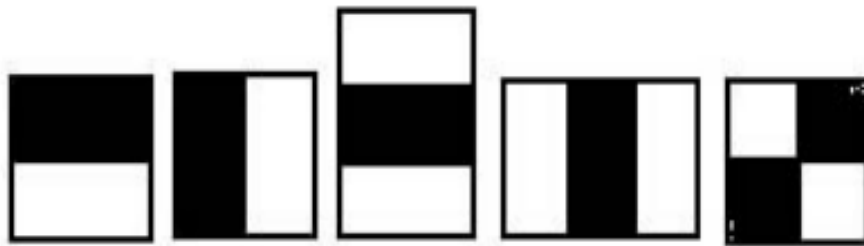
(d)



Gambar 2-2 Pengukuran Data Biometri Janin (a) BPD dan HC (b) AC (c) FL (d) HL (e) CRL.

2.3 Fitur *Haar*

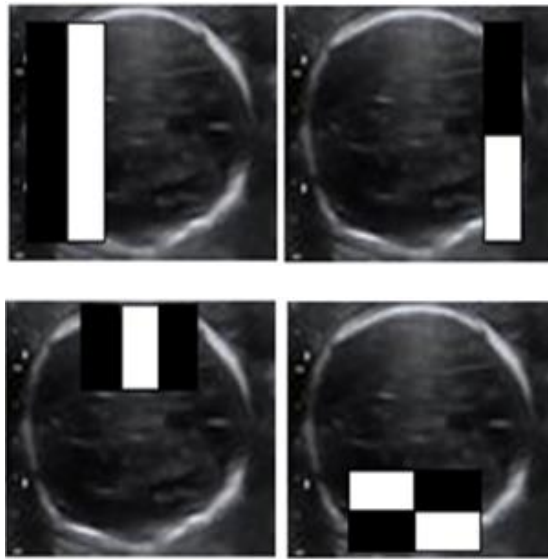
Fitur *Haar* merupakan kernel (fitur) yang digunakan untuk proses pendeteksian objek. Fitur ini diperkenalkan oleh Viola dan Jones pada pendeteksian objek wajah manusia pada suatu citra [11]. Variasi jenis Fitur *Haar* yang digunakan dapat bervariasi sampai motif apapun yang diinginkan, akan tetapi ada lima fitur dasar yang digunakan untuk dalam proses pendeteksian wajah manusia. Lima fitur dasar tersebut dapat dilihat pada gambar 2.3. Contoh pengaplikasian fitur *Haar* dalam citra USG dapat dilihat pada gambar 2.4



Gambar 2-3 Fitur dasar Haar

Selanjutnya, dari fitur dasar tersebut, diekspan menjadi fitur serupa dengan variasi posisi dan ukuran (lebar dan tinggi). Nilai dari setiap fitur tersebut adalah selisih jumlah piksel yang ada di kotak hitam dan jumlah piksel yang ada di kotak putih, atau lebih jelasnya ditulis dengan aturan berikut :

- 2-kotak : (selisih jumlah *pixel* antar kotak kiri-kanan, atas-bawah)
- 3-kotak : (selisih jumlah *pixel* antara di kotak pinggir dengan di kotak tengah)
- 4-kotak : (selisih jumlah *pixel* anatar kotak yang berdiagonal)

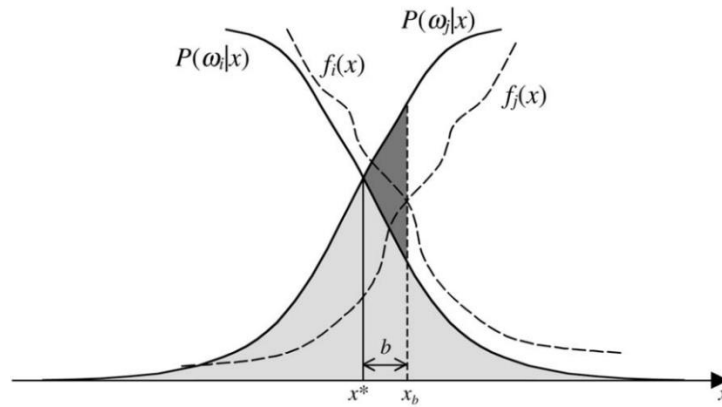


Gambar 2-4 Contoh pengaplikasian fitur Haar dasar pada citra USG

2.4 Ensemble Classifier

Seperti yang disebutkan di subbab sebelumnya bahwa *Adaboost* merupakan metode penggabungan (*ensembles*) *classifier*. Metode *ensembles* yang sering digunakan adalah dengan kombinasi linier dari *weak classifier* yang digunakan. Kombinasi linier sendiri ada dua macam, yaitu *simple averaging* dan *weighted averaging* [14]. Menurut Roli, F. [14] keduanya memiliki kearakteristik masing-masing. Oleh sebab itu Roli, F. melakukan analisis terhadap masing-masing metode *ensembles classifier*.

Suatu permasalahan klasifikasi dimodelkan sebagai model prediksi suatu atribut kelas berdasarkan fitur-fitur data tersebut. Ilustrasi pemodelan klasifikasi dapat dilihat pada gambar 2.8 [14]. Pada problem klasifikasi, ingin dilakukan perhitungan probabilitas suatu kelas ω_i dan ω_j diketahui fitur x $P(\omega_i/x)$. Akan tetapi yang dibentuk oleh klasifikasi adalah probabilitas pada kelas i untuk fitur x , $f_i(x)$.



Gambar 2-5 Ilustrasi Pemodelan Klasifikasi [18]

Pada gambar di atas, area putih merupakan area dimana prediksi dilakukan dengan tepat. Area abu-abu terang adalah area dimana ada *Bayes error*, yaitu *error* yang disebabkan secara alami karena adanya *overlap* pada dataset. Selain itu terjadi *error* yang diakibatkan perubahan dari $P(\omega_i | x)$ ke $f_i(x)$, yang disebut *added error*, dituliskan sebagai:

$$f_k(x) = P(\omega_k | x) + \varepsilon_k \quad (2.1)$$

Pada kasus *simple averaging*, *classifier* digabungkan dengan dicari rata-rata dari semua *weak classifier* yang dibuat. Dengan demikian *added error* untuk penggabungan secara *simple averaging* ini dapat ditulis sebagai persamaan berikut:

$$\varepsilon_k^{sa}(x) = \frac{1}{N} \sum_{m=1}^N \varepsilon_k^m(m) \quad (2.2)$$

Sedangkan *weighted averaging* merupakan penggabungan secara linier dari semua *weak classifier*, sedemikian hingga *added error* untuk *weighted averaging* dapat ditulis sebagai persamaan berikut:

$$\varepsilon_k^{wa}(x) = \frac{1}{N} \sum_{m=1}^N w_m \varepsilon_k^m(m) \quad (2.3)$$

Kondisi optimal, *simple averaging*, adalah ketika seluruh *weak classifier* memiliki nilai *added error* yang hampir sama. Pada kondisi ini nilai *added error* untuk *simple averaging* dituliskan dengan persamaan berikut:

$$E_{add}^{sa}(x) = \frac{1}{N^2} \sum_{m=1}^N E_{add}^m \quad (2.4)$$

Sedangkan untuk *weighted averaging*, kondisi optimal dicapai jika pemberian bobot untuk masing-masing *classifier* diberikan secara optimal, dilihat dari *performance classifier* itu sendiri. Pemberian bobot yang optimal untuk setiap *classifier* dapat dilakukan dengan menggunakan persamaan berikut:

$$w_m = \left(\sum_{m=1}^N \frac{1}{E_{add}^m} \right)^{-1} \frac{1}{E_{add}^m} \quad (2.5)$$

Pada kondisi optimal tersebut, nilai *added error* untuk *weighted averaging* dapat ditulis dengan persamaan berikut:

$$E_{add}^{wa} = \frac{1}{\frac{1}{E_{add}^1} + \dots + \frac{1}{E_{add}^N}} \quad (2.6)$$

Roli menyebutkan bahwa kedua pendekatan *ensembles classifier* tersebut memiliki karakteristik yang berbeda. *Simple averaging*, cocok digunakan untuk kondisi di mana semua *weak classifier* memiliki *performance* yang hampir sama (tidak beda jauh). Akan tetapi, jika kondisi sebaliknya, maka *weighted averaging* memiliki *performance* yang lebih bagus. Selain itu, dapat dilihat dari persamaan *added error* dari kedua pendekatan, bahwa pada kondisi optimal *simple averaging* akan menjamin performa *classifier* yang lebih baik dari *worst classifier* yang digabung, dan *weighred averaging*, akan menjamin performa *classifier* yang lebih baik dari *best classifier* yang digabung.

2.5 Adaboost

2.5.1 Binary Class Adaboost

Binary Class Adaboost merupakan suatu metode *ensembles multi classifier* untuk mendapatkan *classifier* baru yang memiliki performa lebih bagus dengan prinsip *boosting*. Metode ini pertama kali diusulkan oleh Freund dan Schapire pada tahun 1997 [9]. *Classifier-classifier* yang digabungkan tadi biasanya disebut *weak classifier*. Pada umumnya *weak classifier* yang digunakan adalah *classifier* sesederhana mungkin, seperti fungsi *thresholding* atau *Naive Bayes*. *Boosting* sendiri adalah prinsip men-*generate* banyak *classifier* yang nantinya akan dilakukan penggabungan baik melalui *voting* maupun *averaging*. Prinsip-prinsip *boosting* adalah sebagai berikut:

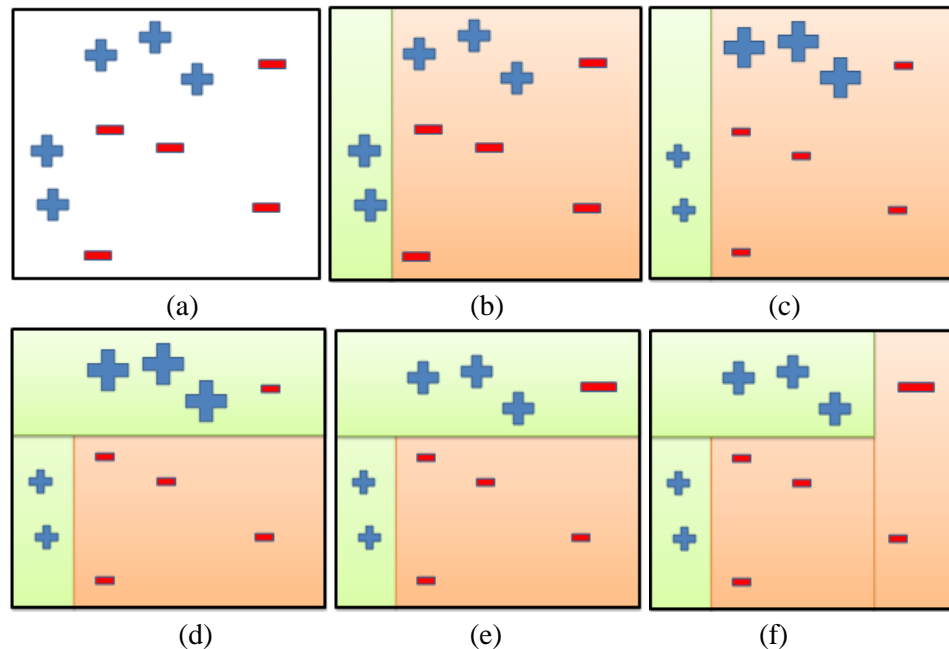
- (i) *Generate* sejumlah m *classifier*
- (ii) *Generate* training set untuk setiap *classifier* (secara acak diambil dari dataset)
- (iii) Lakukan *training* dan *testing* untuk setiap *classifier*
- (iv) Lakukan penggabungan (*ensembles*) dari semua *classifier*

Selanjutnya, metode *boosting* dikembangkan lagi menjadi algoritma yang lebih adaptif, dikenal sebagai *Adaboost*. *Adaboost* dipopulerkan oleh Viola, Jones, untuk seleksi fitur sekaligus klasifikasi objek/gambar wajah dalam suatu citra [11]. Selanjutnya, metode ini juga dioptimasi dengan algoritma ide dari algoritma *Adaboost* adalah memilih *weak classifier*/fitur dalam setiap iterasi yang memiliki *error* paling kecil.

Pertama kali semua data sampel diberi bobot yang sama, baik sampel positif maupun negatif. Bobot ini nantinya akan digunakan sebagai pertimbangan dalam memilih *weak classifier* terbaik dalam setiap iterasi. Selanjutnya, untuk setiap fitur, dibuat satu *weak classifier*. Setiap *weak classifier* tersebut di-*training* dan diuji coba (*testing*). Selanjutnya dipilih satu *classifier* dengan jumlah *error* paling kecil. *Error* yang dimaksud adalah jumlah bobot sampel yang diklasifikasikan ke kelas yang salah. Selanjutnya dilakukan *update* bobot. Sampel yang diklasifikasikan ke kelas

yang benar, bobotnya diperkecil, sedangkan sampel-sampel yang diklasifikasikan ke kelas yang salah satu bobotnya diperbesar. Hal ini adalah salah satu ide utama *Adaboost*. Dengan prinsip ini, pada iterasi selanjutnya, *classifier* yang akan dibentuk selanjutnya akan difokuskan untuk membedakan sampel-sampel yang salah klasifikasi saja. Selanjutnya, *classifier-classifier* yang telah terbentuk akan digabungkan (*ensemble*) sebagai satu *strong classifier*.

Pada gambar 2.6 diilustrasikan langkah-langkah algoritma *Adaboost*. Misalnya kita memiliki dataset yang sudah dianotasi dengan label kelas (+) dan (-) seperti pada gambar (a). Pada iterasi pertama *Adaboost* akan membuat suatu *threshold* dengan *error* paling kecil seperti gambar (b). Selanjutnya, akan dilakukan *update* bobot sampel. Sampel yang diklasifikasikan ke kelas yang benar, bobotnya akan semakin kecil. Sampel yang diklasifikasikan ke kelas yang salah bobotnya akan diperbesar, seperti yang diilustrasikan gambar (c). Hal ini dilakukan terus menerus pada iterasi selanjutnya. Misalnya pada iterasi kedua dilakukan pemilihan *threshold* lagi, yang memiliki *error* paling kecil, seperti pada gambar (d). Selanjutnya, akan dilakukan *update* bobot seperti iterasi sebelumnya, yang ditunjukkan oleh gambar (e). Setelah melalui seluruh iterasi, hasil akhir seperti ditunjukkan oleh gambar (f).



Gambar 2-6 Ilustrasi Cara Kerja Adaboost

Cara kerja algoritma *Adaboost* dapat dituliskan dengan tahapan-tahapan berikut.

1. Diberikan suatu dataset (*sample*) : $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N)$, dimana x_i adalah nilai fitur dari sampel $x_i = \{f_{i,1}, f_{i,2}, \dots, f_{i,m}\}$ dan y_i adalah label kelas dari sampel $y_i \in \{0, 1\}$
2. Inisialisasikan bobot $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ untuk $y_i = 0, 1$ berturut-turut, di mana m dan l adalah jumlah sampel negatif dan positif berturut-turut.
3. Untuk setiap iterasi $t = 1$ sampai T , lakukan proses berikut. T adalah jumlah *classifier* yang ingin di-*ensemble*.
 - a. Lakukan *training* pada setiap *weak classifier* $h_j(x)$, dimana j adalah indeks *classifier*, $j = 1, 2, \dots, M$. M adalah jumlah *classifier* yang di-*generate* untuk dipilih. Dalam beberapa kasus *ensemble* untuk seleksi fitur dan pembentukan *classifier*, M adalah jumlah fitur.
 - b. Lakukan evaluasi setiap kandidat *classifier* lemah tersebut dengan perhitungan *error* sebagai berikut:

$$\epsilon = \sum_{i=1} w_i |h_i(x_i) - y_i| \quad (2.7)$$

- c. Pilih kandidat *weak classifier* lemah dengan *error* yang paling sedikit.
- d. Lakukan *update* bobot

$$w_{t+1,i} = w_{t,i} \beta_t^{1-b_i} \quad (2.8)$$

Dimana $b_i = 0$ jika $h(x_i) = y_i$, jika tidak nilai $b_i = 1$ dan

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t} \quad (2.9)$$

4. Hasil *classifier* akhir adalah sebagai berikut:

$$H(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (2.10)$$

di mana

$$\alpha_t = \log \frac{1}{\beta_t} \quad (2.11)$$

Secara lebih detail metode *Adaboost* dapat ditunjukkan oleh Algoritma 2.1

Algorithm 2.1 : Binary Class Adaboost:

Samples = : $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N)$, $y \in \{0, 1\}$

Init weight : if $y_i = 0$ $w_i = 1/2m$ else, $w_i = 1/2l$, $m = \text{num_of(positive)}$, $l = \text{num_of(negative)}$

For ($t = 1$ to T)

for each feature j , train base classifier $h_j()$

compute error h_j : $\epsilon = \sum_{i=1} w_i |h_j(x_i) - y_i|$

$h_t = \text{smallest_error}(h_j)$

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$$

$$w_{t+1,i} = w_{t,i} \beta_t^{1-y_i}$$

$$\alpha_t = \log \frac{1}{\beta_t}$$

end for

$$H(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

Gambar 2-7 Pseudocode Algoritma Adaboost

2.5.2 Multi Class Adaboost

Multi Class Adaboost merupakan pengembangan dari *binary class Adaboost*. Metode ini pertama kali diusulkan oleh Schapire dan Singer pada tahun 1999 [10]. Pengembangan metode *Adaboost* jenis ini sering disebut dengan *Adaboost Multiclass based on Hamming Loss (Adaboost.MH)*. Selain digunakan untuk klasifikasi *multi class*, *Adaboost.MH* juga digunakan dalam kasus klasifikasi *multi label*, dimana suatu sampel bisa mempunyai dua atau lebih label kelas yang berbeda-beda.

Prinsip umum metode *Adaboost.MH* mirip dengan metode *Adaboost*. yaitu mengambil *classifier* dengan *error* terkecil pada setiap iterasi. Selanjutnya akan dilakukan *update* bobot sampel. Dalam melakukan klasifikasi *multi class*, *Adaboost.MH* membentuk K buah *binary classifier*, dimana K adalah jumlah kelas. Atau dengan kata lain *base classifier* yang digabung (*di-ensemble*) merupakan vektor

dengan K elemen *binary classifier*. Selain itu, label kelas juga direpresentasikan sebagai vektor dengan K elemen. Masing-masing elemen dari vektor tersebut dapat bernilai -1 atau 1. Nilai 1 pada elemen ke- j artinya sampel yang dimaksud diprediksi dari kelas j . Bobot sampel juga direpresentasikan sebagai vektor dengan K elemen, dimana setiap elemen dari vektor tersebut merepresentasikan bobot sampel untuk kelas yang bersesuaian.

Misalnya diberikan himpunan sampel $X = \{x_1, x_2, \dots, x_n\}$, setiap sampel memiliki m fitur $\{f_1, f_2, \dots, f_m\}$, dan label kelas $Y = \{y_1, y_2, \dots, y_n\}$. Bobot setiap sampel juga direpresentasikan sebagai vektor dengan K elemen. Dengan kata lain, setiap sampel memiliki K buah bobot yang berkorespondensi dengan masing-masing kelas $w_i = \{w_{i,1}, w_{i,2}, \dots, w_{i,n}\}$. Pertama kali dilakukan inisiasi bobot sampel yang dilakukan dengan persamaan berikut:

$$w_{i,l} = \begin{cases} \frac{1}{2n} & , \quad \text{if } (y_{i,l} = 1) \\ \frac{1}{2n(K-1)} & , \quad \text{otherwise } (y = -1) \end{cases} \quad (2.12)$$

Dalam setiap iterasi, *Adaboost.MH* akan memilih *classifier* yang terbaik, yaitu yang memiliki *error* paling kecil. *Error* yang digunakan dalam hal ini adalah *Hamming loss error* yang dapat dinyatakan sebagai persamaan berikut:

$$E_H = \sum_{i=1}^n \sum_{l=1}^K w_{i,l} \mid \{ \text{sign}(f_l(x_i)) \neq y_{i,l} \} \quad (2.13)$$

Atau dapat dihitung dengan *margin error* yang dapat didefinisikan sebagai persamaan berikut:

$$E_Z = \sum_{i=1}^n \sum_{l=1}^K w_{i,l} \exp(-f_l(x_i) \cdot y_{i,l}) \quad (2.14)$$

Menurut Schapire dan Singer, dalam proses *ensemble* ini akan digabungkan beberapa *classifier*, di mana setiap *classifier* yang digabungkan tadi memiliki *koeffisien* bobot α . Sehingga nilai *error* untuk setiap *classifier* dapat ditulis dengan persamaan berikut:

$$Z = \sum_{i=1}^n \sum_{l=1}^K w_{i,l} \exp(-\alpha \cdot f_l(x_i) \cdot y_{i,l}) \quad (2.15)$$

Dalam persamaan 2.15 di atas, Z dapat juga disebut sebagai *energy*. Seperti yang disebutkan dalam penjelasan sebelumnya bahwa *base classifier* dinyatakan sebagai vektor yang memiliki K elemen. Setiap elemen tersebut adalah *binary classifier*. Secara formal, *base classifier* dinyatakan sebagai persamaan:

$$h(x) = \alpha \mathbf{v} \phi(x) \quad (2.16)$$

Dimana α adalah koefisien *base classifier*, \mathbf{v} adalah vektor *voting* yang memiliki K elemen ($\mathbf{v} = \{+1, -1\}^K$) dan ϕ adalah *binary classifier*. Dengan representasi *weak classifier* seperti pada persamaan 2.16 dan, nilai Z pada persamaan 2.16 dapat diubah menjadi persamaan berikut :

$$Z = \sum_{i=1}^n \sum_{l=1}^K w_{i,l} \exp(-\alpha \cdot v_l \cdot \phi_l(x_i) \cdot y_{i,l}) \quad (2.17)$$

Persamaan *energy* di atas dapat diuraikan sebagai persamaan berikut :

$$\begin{aligned} Z &= \sum_{i=1}^n \sum_{l=1}^K w_{i,l} \exp(-\alpha \cdot v_l \cdot \phi_l(x_i) \cdot y_{i,l}) \\ &= \sum_{i=1}^n \sum_{l=1}^K (w_{i,l} | \{v_l \cdot \phi_l(x_i) \cdot y_{i,l} = 1\}) \exp(-\alpha) \\ &\quad + w_{i,l} | \{v_l \cdot \phi_l(x_i) \cdot y_{i,l} = -1\}) \exp(\alpha) \end{aligned} \quad (2.18)$$

$$\begin{aligned} &= \sum_{l=1}^K (\mu_{l+} | \{v_l = 1\} + \mu_{l-} | \{v_l = -1\}) \exp(-\alpha) \\ &\quad + \sum_{l=1}^K (\mu_{l-} | \{v_l = 1\} + \mu_{l+} | \{v_l = -1\}) \exp(\alpha) \end{aligned} \quad (2.19)$$

$$\begin{aligned} &= \sum_{l=1}^K ((\exp(-\alpha) \cdot \mu_{l+} + \exp(\alpha) \cdot \mu_{l-}) | \{v_l = 1\} \\ &\quad + (\exp(-\alpha) \cdot \mu_{l-} + \exp(\alpha) \cdot \mu_{l+}) | \{v_l = -1\})) \end{aligned} \quad (2.20)$$

$$= \sum_{l=1}^K \left(\frac{1 + v_l}{2} \right) (\exp(-\alpha) \cdot \mu_{l+} + \exp(\alpha) \cdot \mu_{l-})$$

$$+ \left(\frac{1 - v_l}{2} \right) (\exp(-\alpha) \cdot \mu_{l-} + \exp(\alpha) \cdot \mu_{l+})) \quad (2.21)$$

$$= \left(\frac{1}{2} \right) \sum_{l=1}^K ((\exp(\alpha) + \exp(-\alpha)) (\mu_{l+} + \mu_{l-}) - v_l (\exp(\alpha) - \exp(-\alpha)) (\mu_{l+} - \mu_{l-})) \quad (2.22)$$

$$= \left(\frac{\exp(\alpha) + \exp(-\alpha)}{2} \right) - \left(\frac{\exp(\alpha) - \exp(-\alpha)}{2} \right) \sum_{l=1}^K v_l (\mu_{l+} - \mu_{l-}) \quad (2.23)$$

di mana μ_{l-} adalah *weighted error per-class* :

$$\mu_{l-} = \sum_{i=1}^n w_{i,l} | \{ \varphi_l(x_i) \neq y_{i,l} \} \quad (2.25)$$

Sedangkan μ_{l+}

$$\mu_{l+} = \sum_{i=1}^n w_{i,l} | \{ \varphi_l(x_i) = y_{i,l} \} \quad (2.26)$$

μ_{l+} dan μ_{l-} memenuhi

$$\sum_{l=1}^K v_l (\mu_{l+} + \mu_{l-}) = 1 \quad (2.27)$$

dan v_l

$$v_l = \begin{cases} +1, & \text{if } \mu_{l+} > \mu_{l-} \\ -1, & \text{otherwise} \end{cases} \quad l=1, \dots, K \quad (2.28)$$

Pada setiap *classifier* akan dicari nilai α yang meminimalkan nilai *energy* untuk nilai μ_{l+} dan nilai μ_{l-} yang diperoleh dari proses *training*. Oleh sebab itu untuk mendapatkan nilai *energy* (ambil persamaan 2.19) yang minimal, dapat diperoleh dengan nilai α sebagai berikut :

$$\frac{d(Z)}{d(\alpha)} = 0 \quad (2.29)$$

$$p = \sum_{l=1}^K (\mu_{l+} | \{v_l = 1\} + \mu_{l-} | \{v_l = -1\}) \quad (2.30)$$

$$q = \sum_{l=1}^K (\mu_{l-} | \{v_l = 1\} + \mu_{l+} | \{v_l = -1\}) \quad (2.31)$$

$$\frac{d(p \cdot \exp(-\alpha) + q \cdot \exp(\alpha))}{d(\alpha)} = 0 \quad (2.32)$$

$$(-p \cdot \exp(-\alpha) + q \cdot \exp(\alpha)) = 0 \quad (2.33)$$

$$p \cdot \exp(-\alpha) = q \cdot \exp(\alpha) \quad (2.34)$$

$$\ln(p) - \alpha = \ln(q) + \alpha \quad (2.35)$$

$$2\alpha = \ln(p) - \ln(q) \quad (2.36)$$

$$\alpha = \frac{1}{2} (\ln(p) - \ln(q)) \quad (2.37)$$

$$\alpha = \frac{1}{2} \ln\left(\frac{p}{q}\right) \quad (2.38)$$

$$\alpha = \frac{1}{2} \ln \frac{\sum_{l=1}^K (\mu_{l+} | \{v_l = +1\} + \mu_{l-} | \{v_l = -1\})}{\sum_{l=1}^K (\mu_{l-} | \{v_l = +1\} + \mu_{l+} | \{v_l = -1\})} \quad (2.39)$$

Dalam proses pembelajaran, $\phi(x)$ ditujukan untuk memaksimalkan *classification edge* (γ) yang dapat dinyatakan sebagai persamaan berikut:

$$\gamma(\phi) = \sum_{i=1}^n (\mu_{l+} - \mu_{l-}) \quad (2.40)$$

$$\gamma(\phi) = \sum_{i=1}^n \sum_{l=1}^K w_{i,l} \cdot v_l \cdot \phi_l(x_i) \cdot y_{i,l} \quad (2.43)$$

Secara iteratif, algoritma *Adaboost* dapat ditunjukkan pada Algoritma 2.2:

Algorithm 2.2 : Adaboost.MH

```

Given samples  $X = \{x_1, \dots, x_N\}$ 
Init weight, equation (2.12)

For ( $t=1$  to  $T$ )
  for each feature  $j$ , train base classifier  $h_j = \alpha_j \vee_j \varphi_j$ 
  compute edge of  $\varphi_j$  : equation (2.20)
  compute  $\mu_{l-}$  and  $\mu_{l+}$  for  $l = 1, \dots, K$  (equation (2.18) and (2.19))
  compute margin error : equation (2.14)
   $h_t = h_j$  with smallest_error
  compute  $\alpha$  : equation (2.17)

  update samples weight
  
$$w_{i,l}^{t+1} = w_{i,l}^t \frac{\exp(-\alpha \cdot h_l^t(x_i) \cdot y_{i,l})}{\sum_{l=1}^n \sum_{l=1}^K w_{i,l}^t \exp(-\alpha \cdot h_l^t(x_i) \cdot y_{i,l})}$$


end for


$$H(x) = \sum_{t=1}^T \alpha_t h_t(x)$$


```

Gambar 2-8 Pseudocode Algoritma Adaboost.MH

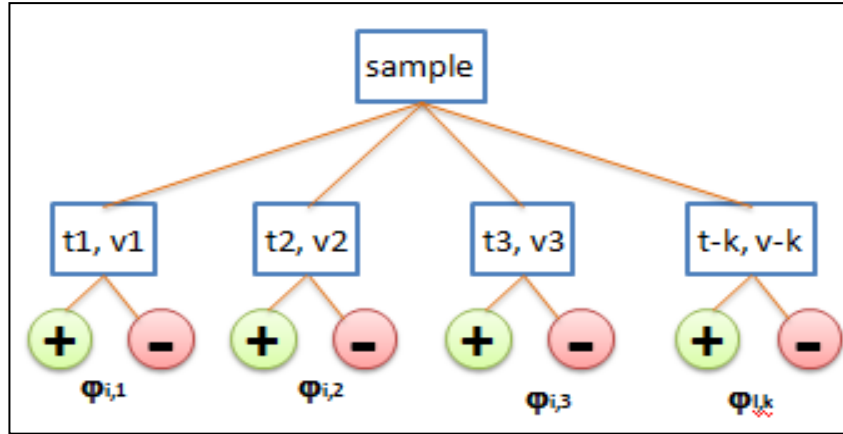
2.6 Base Classifier

Dalam *ensemble classifier*, *base classifier* merupakan *classifier-classifier* yang digabungkan untuk membentuk satu *classifier* baru. *Classifier* yang digabungkan tersebut, biasanya juga disebut baru ini sering disebut *weak classifier*, sedangkan *classifier-classifier* hasil penggabungan tadi disebut *strong classifier*. Pada subbab ini akan dijelaskan berbagai jenis *classifier-classifier* yang di-*ensemble* menggunakan metode *Adaboost* dalam penelitian-penelitian sebelumnya.

2.6.1 Stump Algorithm

Stump algorithm merupakan algoritma klasifikasi sederhana yang membuat batas dari suatu data *training* menjadi dua bagian, yaitu bagian positif dan bagian negatif. Bagian positif dianggap sebagai kelas yang bersangkutan. Pembagian data ini dilakukan dengan mencari suatu batas (*threshold*) yang paling optimal untuk membagi kedua bagian tadi. Konsep dasar dari algoritma ini memang berasal dari permasalahan *binary classification*, akan tetapi dalam permasalahan *multi class classification* algoritma ini tetap bisa digunakan, dengan cara membuat vektor *classifier* yang terdiri dari K kelas *binary classifier*. Algoritma ini dipopulerkan oleh

Balazs dan Robert pada tahun 2009 [12]. Ilustrasi arsitektur algoritma *stump* ini dapat dilihat pada gambar 2.9.



Gambar 2-9 Ilustrasi Algoritma *Stump*

Aturan klasifikasi dari algoritma *stump* ini dapat dinyatakan sebagai berikut:

$$prediction(x_i) = \begin{cases} +1, & \text{if } (v_i \cdot x_i) > t_i \\ -1, & \text{otherwise} \end{cases} \quad i=1, \dots, K \quad (2.44)$$

Berdasarkan persamaan (2.22), dapat dikatakan bahwa untuk suatu sampel, akan dicobakan kepada semua *binary classifier*. Dengan kata lain setiap sampel dapat diprediksi dalam dua atau lebih kelas. Dengan demikian algoritma ini juga dapat digunakan untuk *multi label classification*.

Untuk memberikan hasil klasifikasi yang terbaik, diperlukan nilai referensi batas (t) dan vote (v) yang paling optimal. Oleh sebab itu, selama pembelajaran akan dicari nilai-nilai tersebut yang memenuhi *error* klasifikasi yang terkecil. Langkah-langkah pencarian nilai batas dan nilai vote tersebut dapat dilakukan sebagai berikut:

1. Diberikan sampel $X = \{x_1, x_2, \dots, x_n\}$, label kelas $Y = \{\{y_{1,1}, y_{1,2}, \dots, y_{1,K}\}, \{y_{2,1}, y_{2,2}, \dots, y_{2,K}\}, \dots, \{y_{n,1}, y_{n,2}, \dots, y_{n,K}\}\}$ dan bobot sampel $W = \{\{w_{1,1}, w_{1,2}, \dots, w_{1,K}\}, \{w_{2,1}, w_{2,2}, \dots, w_{2,K}\}, \dots, \{w_{n,1}, w_{n,2}, \dots, w_{n,K}\}\}$.
2. Inisiasi *half edges* μ_{l-} dan μ_{l+} dengan nilai 0.
3. *Looping* untuk setiap sampel yang dilakukan melalui langkah-langkah berikut

- a. Tambahkan nilai *half edge* dengan bobot sampel dikalikan label kelasnya. Proses ini dapat dilakukan dengan persamaan berikut:

Jika $w_{i,l} \cdot y_{i,l} > 0$

$$\mu_{l+} \leftarrow \mu_{l+} + (w_{i,l} \cdot y_{i,l}) \quad (2.45)$$

Jika tidak

$$\mu_{l-} \leftarrow \mu_{l-} - (w_{i,l} \cdot y_{i,l}) \quad (2.46)$$

- b. Tentukan nilai v berdasarkan nilai *half edges*

$$v_l = \text{sign}(\mu_{l+} - \mu_{l-}) \quad (2.47)$$

- c. Cek apakah *half edge* saat ini lebih baik dari *half edge* sebelumnya dan lakukan update *best half edge* serta nilai batas (*threshold*) t .

Jika $(\mu_l \cdot v) > (\text{best } \mu_l \cdot \text{best } v_l)$ maka:

$$\text{best } \mu_l \leftarrow \mu_l \quad (2.48)$$

$$\text{best } v_l \leftarrow v_l \quad (2.49)$$

$$t_l \leftarrow \frac{1}{2} (x_i + x_{i-1}) \quad (2.50)$$

- d. Hitung nilai α untuk *best half edge* terbaru

2.6.2 Product base Classifier

Metode *product base classifier* merupakan metode yang menggabungkan beberapa *base classifier* dengan operasi *product*. Metode ini diperkenalkan oleh Balazs dan Robert pada tahun 2009 [12]. Secara formal aturan penggabungan metode ini dapat dinyatakan dengan persamaan berikut:

$$h(.) = \alpha \prod_{j=1}^m v_j \cdot \phi_l(.) \quad (2.51)$$

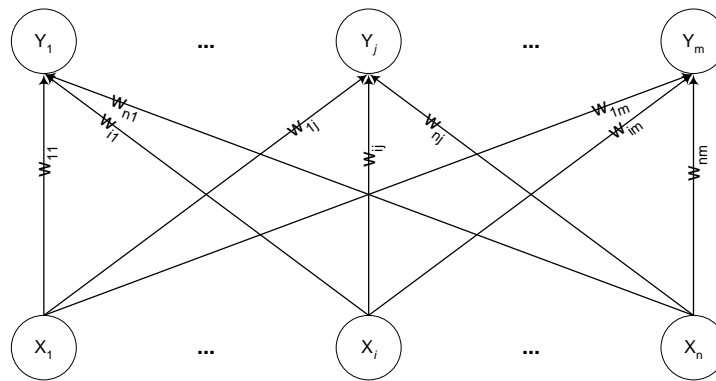
Dapat dikatakan bahwa metode ini merupakan *intermediate ensemble*, sebelum dilakukan *ensemble* menggunakan *Adaboost*. Kelebihan dari teknik ini

adalah meningkatkan konsistensi antara *classifier-classifier* yang digabungkan dengan operasi *product*. Hal ini dikarenakan jika prediksi dari *classifier-classifier* yang digabungkan berbeda, maka hasil akhir dari operasi *product* akan salah. Dengan demikian, selama proses *training*, selain harus mencari parameter terbaik, *classifier-classifier* yang digabungkan juga harus memiliki konsistensi satu dengan yang lain. Dalam operasi ini, *classifier* yang digabungkan bisa bermacam sesuai kebutuhan.

2.6.3 Learning Vector Quantization

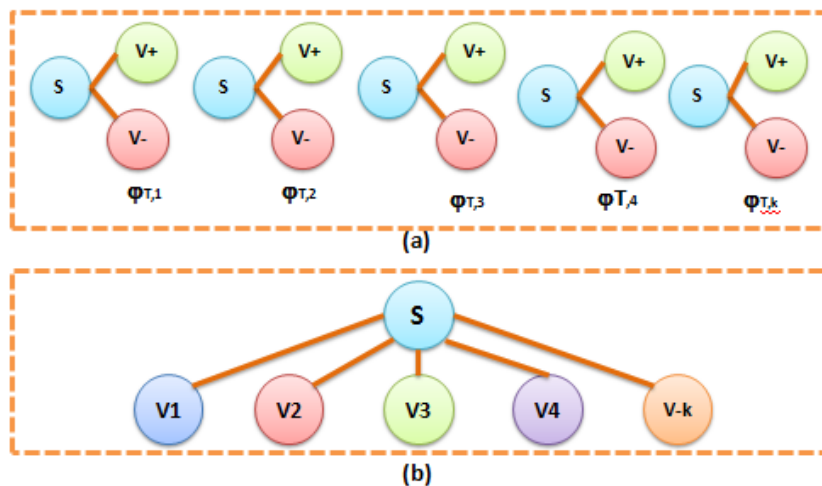
Pada penelitian sebelumnya oleh Xiang, Liu et al., dilakukan *ensemble* metode *Learning Vector Quantization* (LVQ) dengan metode *Adaboost* [15]. *Learning Vector Quantization* merupakan suatu metode untuk melakukan klasifikasi data dan digolongkan sebagai salah satu jenis algoritma pembelajaran yang memerlukan supervisi. Algoritma ini pertama kali dikembangkan oleh Kohonen [16] tahun 1986 dengan struktur jaringan yang menyerupai Self Organizing Map, namun tidak memiliki struktur topologis. Keluaran yang diharapkan dari LVQ adalah suatu vektor yang dapat mengelompokkan data ke dalam kelas atau kategori yang berbeda.

Ilustrasi arsitektur LVQ dapat dilihat pada gambar 2.10. Dalam melakukan klasifikasi, algoritma LVQ melakukan klasifikasi data berdasarkan jarak antar masukan dengan suatu pembanding yang disebut sebagai vektor perwakilan atau *codebook* atau *reference vector*. Masukan dari algoritma ini dapat dinyatakan sebagai suatu vektor $x = (x_1, x_2, x_3, \dots, x_n)$ di mana setiap elemen dari vektor tersebut merepresentasikan fitur yang berbeda-beda. Karena LVQ adalah algoritma yang memerlukan supervisi, maka sebelumnya kategori yang akan dihasilkan harus didefinisikan terlebih dahulu. Sebagai contoh, kita dapat menyatakan kelompok yang diharapkan sebagai C_j dimana j menyatakan kategori yang berbeda-beda. Selanjutnya, didefinisikan juga T sebagai kategori yang benar untuk setiap vektor masukan x .



Gambar 2-10 Arsitektur Algoritma Klasifikasi *Learning Vector Quantization* (LVQ)

Dari konsep dasar, LVQ sudah bisa melakukan klasifikasi *multi class*. Oleh sebab itu, proses *ensemble* LVQ dengan menggunakan *Adaboost* dapat dilakukan dengan dua pendekatan. Pendekatan pertama adalah dengan langsung meng-*ensemble* LVQ *classifier*, dimana *classifier* ini adalah *multi class classifier*. Pendekatan kedua adalah dengan mendekomposisi *classifier LVQ tersebut* ke dalam K *binary classifier*. Pendekatan pertama ditunjukkan pada gambar



Gambar 2-11 Ilustrasi representasi LVQ untuk di-*ensemble* menggunakan Adaboost. (a) Representasi LVQ dengan dikomposisi menjadi *binary classification* (b) Representasi LVQ tanpa didekomposisi menjadi *binary classification*

Pada algoritma LVQ, fungsi jarak dihitung menggunakan *Euclidian distance* dengan persamaan sebagai berikut:

$$d(j) = \sum_{i=1}^N (x_i - w_{ij})^2 \quad (2.52)$$

di mana x_i adalah vektor masukan ke- i dan w_{ij} adalah vektor perwakilan yang merepresentasikan kategori j . Bila saat proses pembelajaran nilai kategori T dari vektor x berbeda yang terdekat dari vektor perwakilan w_j yang merepresentasikan kategori C_j , maka jarak antara vektor x dengan dengan vektor perwakilan w_j akan dijauhkan. Sebaliknya, bila nilai kategori T yang direpresentasikan oleh vektor x sama dengan vektor w_j terdekat, maka nilai dari vektor w_j akan didekatkan ke vektor masukan. Vektor perwakilan terdekat ini juga disebut sebagai vektor pemenang. Proses pembaruan nilai vektor perwakilan ini dapat dinyatakan dengan persamaan berikut:

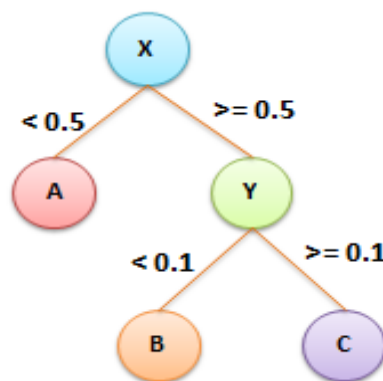
$$w_p \leftarrow w_p + \alpha(x - w_p) \text{ bila } C_{w_p} = C_j \quad (2.53)$$

$$w_p \leftarrow w_p - \alpha(x - w_p) \text{ bila } C_{w_p} \neq C_j \quad (2.54)$$

Algoritma LVQ juga telah mengalami beberapa kali perbaikan sehingga menghasilkan beberapa varian. Pada bagian berikut ini akan diuraikan beberapa varian LVQ, yaitu LVQ1, LVQ2.0, LVQ2.1, dan GLVQ [17].

2.6.4 Classification Tree

Algoritma *Classification Tree* atau pohon klasifikasi merupakan suatu metode yang banyak digunakan dalam bidang pembelajaran mesin, statistik, dan penambangan data. Algoritma ini berfungsi untuk membuat suatu model *tree* (pohon) untuk melakukan prediksi kelas dari suatu data masukan [18]. Diberikan suatu sampel dengan atributnya, lalu model ini menentukan kelas sampel tersebut dengan menggunakan model *tree* dan data atribut dari sampel tadi. Struktur *classification tree* dapat dilihat pada gambar 2.11. Pada struktur *tree* ini, terdapat dua terminologi utama, yaitu cabang dan *leave* (daun). Cabang merepresentasikan suatu aturan (*rule*) yang berkaitan dengan atribut data, sedangkan *leaves* menunjukkan suatu label dalam kelas. Pada gambar 2.11 A, B dan C merepresentasikan kemungkinan label kelas dari sampel, sedangkan X dan Y merepresentasikan atribut dari sampel.



Gambar 2-12 Ilustrasi arsitektur *Classification Tree*

Proses pembentukan *classification tree* pada umumnya dilakukan dengan membentuk satu node awal (*root*), lalu dari *root* dilakukan *split*, untuk mejadi node baru baik itu cabang maupun daun. Dari *node* yang baru terbentuk tadi bisa dilakukan proses *split* lagi, sampai akhirnya konvergen. Pemilihan atribut yang menjadi *root*, cabang pertama, cabang kedua, dan seterusnya didasarkan pada dua kriteria. Kriteria yang sering digunakan yaitu *information gain* dan *gain ratio*. Berikut adalah definisi dari kedua kriteria tersebut:

1. *Information Gain*

Information Gain dari suatu *output* data atau variabel dependen C (label kelas) yang berdasarkan atribut X , dinotasikan $\text{gain}(C, X)$, dirumuskan sebagai berikut :

$$\text{Gain}(C, X) = \text{entropy}(C) - \sum_{C \in \text{value}(A)} \frac{C_v}{C} \text{entropy}(C_v) \quad (2.55)$$

Dimana $\text{value}(X)$ merupakan semua kemungkinan nilai yang dapat dijadikan sebagai calon *test* pada sebuah atribut X . Sedangkan C_v adalah subset dari C dimana atribut X mempunyai nilai v .

2. *Gain Ratio*

Kriteria *Gain Ratio* berbeda dengan kriteria *information gain*, namun dalam perhitungannya masih menggunakan *information gain*. Hanya saja, menggunakan tambahan parameter lain yang disebut *split information*. Berikut ini adalah rumusan untuk menghitung parameter *split information*:

$$SI(C, X) = - \sum_{C \in \text{value}(A)} \frac{C_v}{C} \log_2 \left(\frac{C_v}{C} \right) \quad (2.56)$$

$$\text{Gain Ratio}(C, X) = \left(\frac{\text{Gain}(C, X)}{SI(C, X)} \right) \quad (2.57)$$

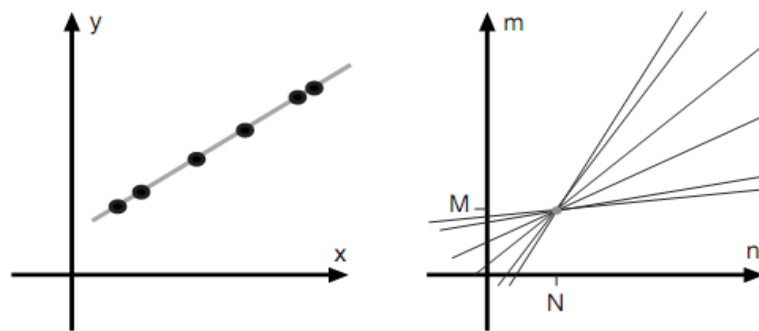
2.9 Hough Transform Untuk Pengukuran Biometri

Metode *Hough Transform* merupakan salah satu metode aproksimasi terhadap suatu bentuk kurva. Metode *Hough Transform* diperkenalkan oleh Paul Hough pada tahun 1959. Pada mulanya metode ini digunakan untuk mencari atau mendeteksi garis yang ada pada citra. Akan tetapi, seiring dengan perkembangan ilmu dan teknologi, saat ini telah diturunkan untuk berbagai persamaan kurva seperti lingkaran, elips dan *polygon* [19].

Metode ini merupakan salah satu metode yang *handal* untuk aproksimasi kurva pada citra meskipun terdapat *noise* yang tinggi pada citra tersebut. Meskipun

demikian metode ini memiliki kelemahan yaitu memiliki kompleksitas yang tinggi untuk *running time* dan membutuhkan penggunaan memori yang tinggi. Ide dasar dari metode ini adalah mentransformasikan suatu bentuk persamaan kurva ke dalam ruang parameter. Sebagai contoh sebuah garis dalam sistem koordinat *Cartesian* (x,y) dapat dideskripsikan dalam persamaan 2.51 berikut ini:

$$y = mx + n \quad (2.58)$$



Gambar 2-13 Prinsip dari metode *Hough Transform* yang mentransformasi semua titik pada koordinat (x, y) ke dalam ruang *Hough* (m, n)

dengan m sebagai nilai gradien (*slope line*) dan n adalah *intercept*. Setiap garis memiliki keunikan bila ditransformasikan ke dalam bentuk persamaan ini. Jika dimisalkan suatu set titik (y_k, x_k) yang terletak pada suatu garis $y = Mx + N$ ditransformasikan ke dalam ruang parameter *Hough* (m,n), maka semua titik pada garis akan bertemu pada titik (M,N) yang dapat diperhatikan pada gambar 2.13. Titik M,N merupakan titik hasil transformasi parameter m,n ke ruang Hough. Setiap titik yang direpresentasikan dalam ruang *Hough* dapat dilakukan dengan persamaan 2.59.

$$m = \frac{y_k}{x_k} - \frac{1}{x_k} n \quad (2.59)$$

Algorithm 2.3 : Hough Transform for Lines[19]

```
function HTLine(inputimage)
    variable : image size, accumulator, image
    [rows,columns]=size(inputimage);
```

```

variable : acc1=zeros(rows,91);
variable acc2=zeros(columns,91);
for x=1:columns
    for y=1:rows
        if(inputimage(y,x)==0
            for m=-45:45
                b=round(y-tan((m*pi)/180)*x);
                if(b<rows & b>0)
                    acc1(b,m+45+1)=acc1(b,m+45+1)+1;
                end
            end
            for m=45:135
                b=round(x-y/tan((m*pi)/180));
                if(b<columns & b>0)
                    acc2(b,m-45+1)=acc2(b,m-45+1)+1;
                end
            end
        end
    end
end
end
end

```

Gambar 2-14 Pseudocode Algoritma Hough Transform Untuk Garis

Contoh lainnya adalah aproksimasi kurva elips yang diderkrisikan dengan persamaan 2.60.

$$\frac{(x - x_c)^2}{a^2} + \frac{(y - y_c)^2}{b^2} = 1 \quad (2.60)$$

Dimana a dan b merupakan semi-major dan semi-minor axes, x_c dan y_c adalah titik pusat dari elips. Namun demikian biasanya terdapat variabel tambahan, yaitu rotasi terhadap elips yang terletak pada citra terhadap suatu sudut θ . Maka dari itu, rumusan yang lebih umum dapat diperhatikan pada persamaan 2.61.

$$\frac{(x \cos \theta + y \sin \theta)^2}{a^2} + \frac{(x \sin \theta - y \cos \theta)^2}{b^2} = 1 \quad (2.61)$$

Untuk menentukan parameter elips jika telah diketahui titik pada bidang maka harus dilakukan proses perubahan persamaan tersebut sesuai persamaan 2.62.

$$x^2 + y^2 - U(x^2 - y^2) - 2Vxy - Rx - Sy - T = 0 \quad (2.62)$$

Dimana masing-masing parameter tersebut adalah sebagai berikut:

$$e = \frac{b}{a} \quad (2.63)$$

$$U = \frac{1 - e^2}{1 + e^2} \cos 2\theta \quad (2.64)$$

$$V = \frac{1 - e^2}{1 + e^2} \sin 2\theta \quad (2.65)$$

$$R = 2x_c(1 - U) - 2y_cV \quad (2.66)$$

$$S = 2y_c(1 - U) - 2x_cV \quad (2.67)$$

$$T = \frac{2a^2b^2}{a^2 + b^2} - \frac{x_cR}{2} - \frac{y_cS}{2} \quad (2.68)$$

Untuk memperoleh nilai dari masing-masing parameter elips $[a, b, x_0, y_0, \theta]$ dapat dilakukan dengan persamaan berikut:

$$x_0 = \frac{SV + R + RU}{2(1 - U^2 - V^2)} \quad (2.69)$$

$$y_0 = \frac{RV + S - SU}{2(1 - U^2 - V^2)} \quad (2.70)$$

$$a = \sqrt{\frac{2T + x_0R + y_0S}{2(1 - \sqrt{U^2 + V^2})}} \quad (2.71)$$

$$b = \sqrt{\frac{2T + x_0R + y_0S}{2(1 + \sqrt{U^2 + V^2})}} \quad (2.72)$$

$$\phi = \frac{1}{2} \arctan \frac{V}{U} \quad (2.73)$$

Untuk menyelesaikan persamaan 2.55 dibutuhkan paling sedikit 5 buah titik yang melalui elips pada bidang. Dengan penggunaan rumus ini maka dibutuhkan

akumulator dengan dimensi 5 yang tentunya akan membutuhkan biaya memori dan komputasi yang tidak sedikit. Salah satu cara untuk mengurangi kompleksitas memori untuk melakukan pendeteksian suatu elips menggunakan metode *Hough Transform* dapat dilakukan jika parameter panjang semi-mayor (a) dan semi-minor axis (b) telah diketahui. Adapun algoritma *Hough Transform* untuk mendeteksi elips jika telah diketahui nilai panjang semi-mayor (a) dan semi-minor axis (b) dapat diperhatikan sesuai dengan algoritma 2.4.

Algorithm 2.4 *Hough Transform For Elips*[19]

```
function HTelips(input_image, a, b)
    [rows, columns] = size(input_citra);
    accumulator [] = zeros (rows,columns)
    for x ← 1 to columns
        for y ← 1 to rows
            if(input_image (y, x) == 255)
                for θ ← 0 to 360
                    t ← (θ*PI)/180
                    x0 ← round(x-a*cos(t))
                    y0 ← round(y-b*cos(t))
                    if(x0 < columns && x0 > 0 & y0 < rows & y0 > 0 )
                        accumulator (y0,x0) = accumulator(y0,x0)+1;
                    end
                end
            end
        end
    end
```

Gambar 2-15 Pseudocode Algoritma *Hough Transform* Untuk Elips

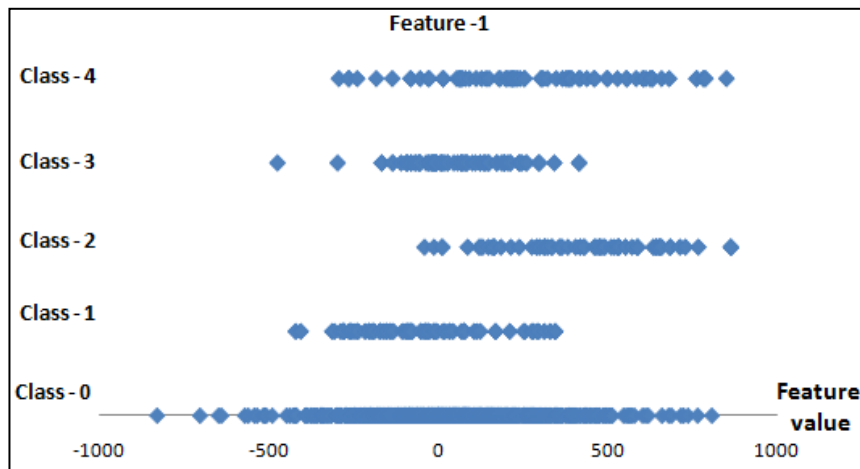
Algoritma *Randomize Hough Transform (RHT)* merupakan pengembangan dari HT yang menggunakan metode *randomization* untuk mempercepat metode HT standar. Perbedaan RHT dengan HT adalah pada sisi pengecekan titik sampelnya. HT akan mengecek seluruh titik sampel yang ada pada gambar, sedangkan RHT akan melakukan *sampling* secara acak dari titik-titik yang ada pada gambar.

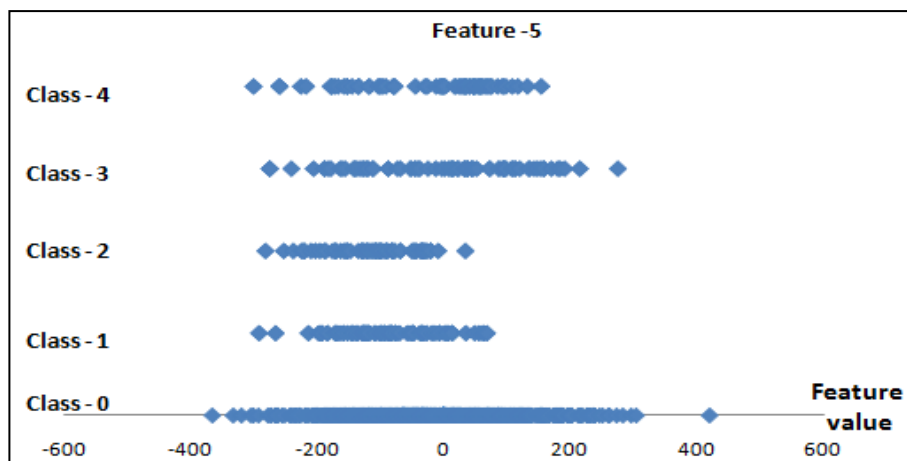
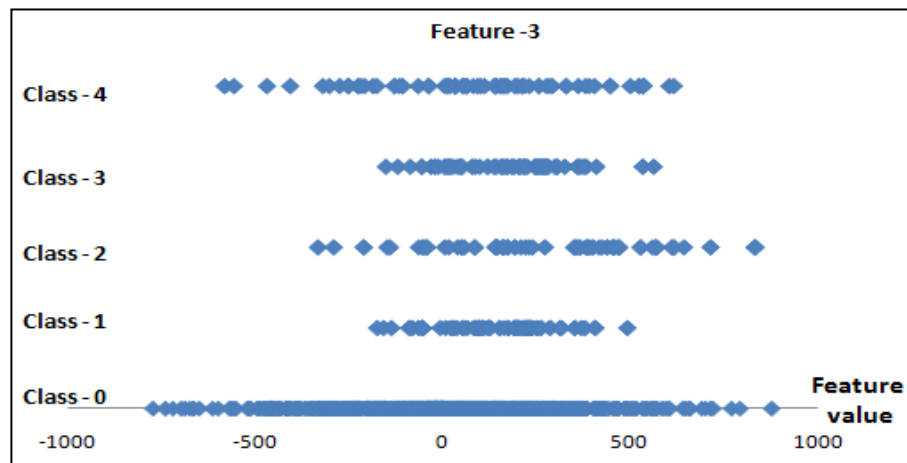
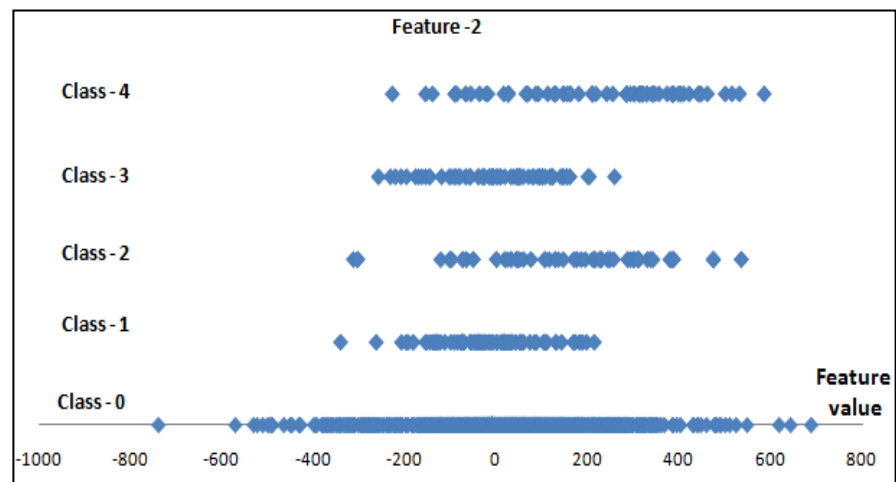
Bab 3 Rancangan dan Implementasi

Pada bab bagian ini dijelaskan mengenai algoritma yang dirancang serta implementasinya. Pada bab ini juga akan dijelaskan komponen yang penting dalam algoritma serta langkah-langkah dari metode yang dikembangkan.

3.1.Ide Dasar *Multi Boundary Classifier*

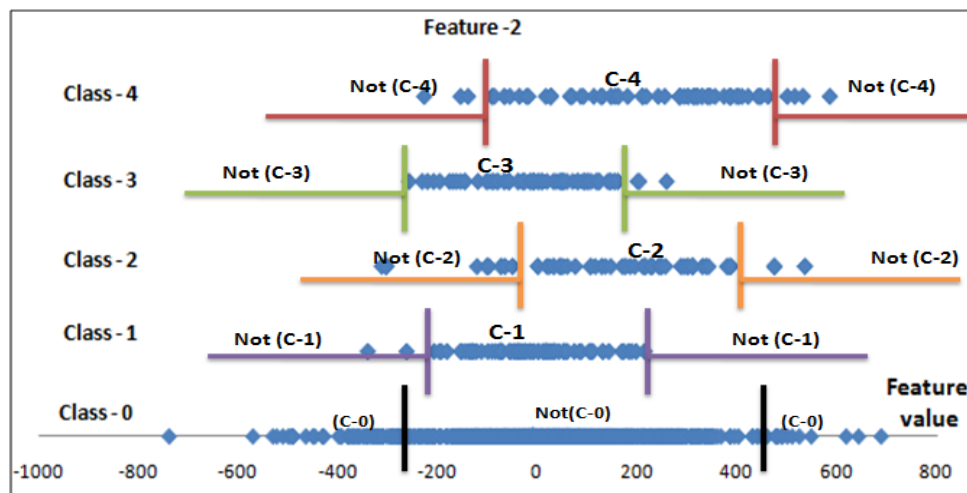
Permasalahan yang ingin dipecahkan dalam penelitian ini adalah segmentasi organ janin pada citra USG. Pendekatan yang digunakan adalah dengan menggunakan metode deteksi objek. Proses yang dilakukan dalam deteksi objek adalah klasifikasi sub *window* dari citra USG. Dari sekian banyak sub *window* yang dievaluasi, dapat dideteksi mana bagian yang merupakan organ janin. Sebelum dilakukan perancangan *classifier*, terlebih dahulu dilakukan observasi sifat data USG yang akan diolah. Pada gambar 3.1 dapat dilihat plot lima fitur terbaik dari citra USG yang akan digunakan. Fitur-fitur tersebut diperoleh dari fitur *Haar*, dan diambil lima fitur terbaik menggunakan metode *Adaboost.MH* berbasis *stump classifier* yang dikembangkan oleh peneliti sebelumnya.





Gambar 3-1 Lima fitur terbaik dari dataset USG

Pada gambar 3.1 dapat dilihat bahwa dataset USG, nilai fitur untuk sampel-sampel dalam kelas yang sama, berada pada suatu rentang tertentu. Berdasarkan sifat data tersebut, penulis bermaksud untuk menyusun suatu *classifier* berbasis *boundary*, Pendekatan untuk mengelompokkan data tersebut berdasarkan kelasnya masing-masing dengan sebaik mungkin. Ilustrasi pembuatan *boundary* ini dapat dilihat pada gambar 3.2. Pendekatan arsitektur *classifier* yang digunakan adalah dengan membuat K-binary *classifier* seperti halnya *stump classifier*. Meskipun sudah dicari *boundary* yang paling optimal untuk masing-masing kelas, tetap aja ada sampel-sampel yang *overlap* yang memungkinkan terjadinya error (*misclassification*). Akan tetapi hal ini bisa ditangani dengan *ensemble classifier* menggunakan *Adaboost*. Sesuai dengan prinsip *Adaboost* bahwa sampel-sampel yang diprediksi ke dalam kelas yang salah, akan diperbesar bobot sampelnya, sedangkan sampel-sampel yang diklasifikasikan ke dalam kelas yang benar akan diperkecil bobotnya. Oleh sebab itu, pada suatu iterasi diharapkan *classifier* ini difokuskan pada sampel-sampel yang bobotnya besar, atau dengan kata lain yang pada iterasi sebelumnya mengalami *misclassification*.

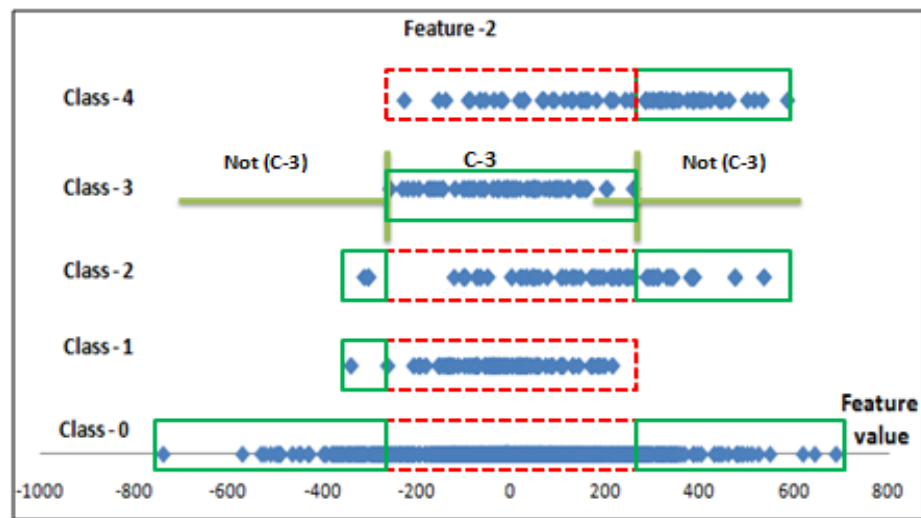


Gambar 3-2 Konsep dasar algoritma Multi Boundary Classifier

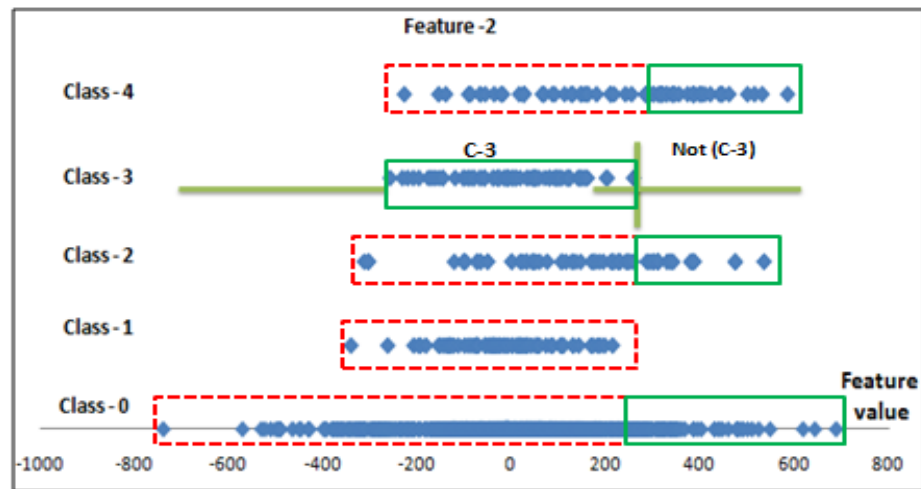
Pada gambar 3.2 dapat dilihat bahwa untuk kelas-1, kelas-2, kelas-3, dan kelas-4 dibuat *boundary* dimana sampel kelas yang bersangkutan berada dalam *rentang* batas minimal dan maksimal *boundary* yang diberikan. Lain halnya dengan

kelas-0, dimana sampel-sampel yang terletak di dalam *boundary* adalah justru sampel yang bukan merupakan sampel dari kelas-0. Hal ini disebabkan, bahwa jika yang terletak di dalam *boundary* adalah sampel kelas-0, maka nilai *error* nya lebih tinggi. Jika yang di dalam *boundary* didefinisikan sebagai kelas-0, maka seluruh sampel dari kelas 1, 2, 3 dan 4 akan dianggap sebagai kelas-0 (*misclassification*). Akan tetapi jika yang dilakukan adalah hal yang sebaliknya, maka sampel kelas-1,2,3, dan 4 akan dianggap bukan kelas-0 saat proses klasifikasi. Meskipun demikian memiliki tingkat *overlapping* yang cukup tinggi.

Peneliti sebelumnya memperkenalkan suatu model *base classifier* berbasis suatu batas, yaitu *stump classifier*. Akan tetapi, algoritma ini menggunakan satu nilai batas saja. Dengan kata lain, untuk fitur yang sama, algoritma *multi boundary classifier* diharapkan bisa mendapatkan hasil dengan *error* lebih kecil. Hal ini dapat diilustrasikan pada gambar 3.3. Pada gambar 3.3 disimulasikan proses klasifikasi dengan menggunakan *multi boundary classifier* (a) dan *stump classifier* (b) untuk *binary classifier* kelas-3. Pada proses ini akan dicek untuk memutuskan apakah setiap sampel diklasifikasikan ke dalam kelas 3 atau tidak. Pada gambar 3.3, garis putus-putus (warna merah) menunjukkan sampel-sampel yang diklasifikasikan ke dalam kelas yang salah. Sedangkan garis penuh (warna hijau) merupakan sampel-sampel yang diklasifikasikan ke kelas yang benar. Pada gambar 3.3 dapat dilihat bahwa untuk fitur yang sama, algoritma *boundary classifier* memiliki kesalahan (*error*) yang lebih kecil.



(a)



(b)

Gambar 3-3 Simulasi Perbedaan Proses Klasifikasi (a) Menggunakan *multi boundary classifier*. (b) Menggunakan *multi stump classifier*

Berdasarkan observasi yang dilakukan, metode *stump* juga memiliki kelemahan sebagai berikut:

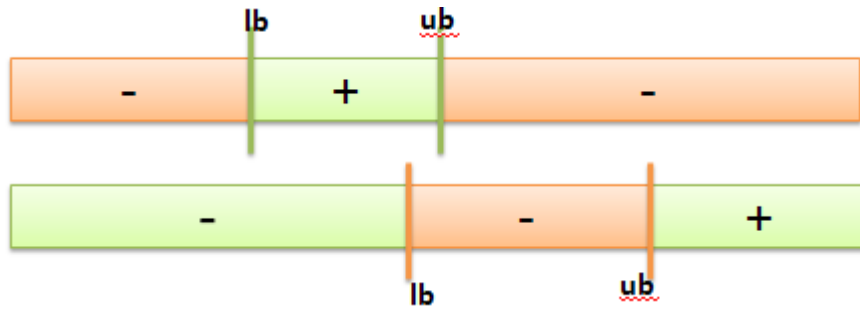
1. Risiko tidak menemukan batas yang dekat dengan nilai fitur. Misalnya saja, kita gunakan fitur 2 seperti pada gambar 3.3. Pada gambar 3.3 (b) algoritma *stump classifier* mempunyai suatu nilai batas untuk kelas-3, katakanlah X. Nilai batas ini tidak dapat ditemukan dalam satu kali iterasi. Hal ini dikarenakan saat algoritma

stump beriterasi dari sampel pertama sampai sampel terakhir, menemukan sampel-sampel negatif (bukan kelas 3) terlebih dahulu. Saat mencapai sampel-sampel positif (kelas 3), sering bertemu kembali dengan sampel-sampel negatif (bukan kelas 3) dari kelas lain, sehingga *update* nilai batas gagal dilaksanakan. Baru saat iterasi berikutnya, setelah ada update bobot sampel, pencarian nilai batas tersebut dapat dilakukan, karena sampel-sampel negatif tadi sudah diperkecil bobotnya.

2. Akibat dari resiko nomor 1, dalam dua kali iterasi yang berurutan, tidak selalu bisa dibentuk *boundary* yang optimal, seperti yang diharapkan pada *multi boundary classifier*. Hal ini juga yang melatarbelakangi bahwa *multi boundary* tidak dibentuk dengan dua kali iterasi *stump*. Ditambah lagi, dalam dua iterasi yang bersambungan, tidak selalu bisa terpilih jenis fitur yang sama.

3.2. Perumusan Algoritma *Multi Boundary Classifier*

Sesuai dengan ide dasar *multi boundary classifier* yang dijelaskan pada sub bab 3.1. bahwa tujuan utama dari algoritma ini adalah untuk membentuk *boundary* untuk masing-masing *binary classification* yang mewakili kelas-kelas pada data. Untuk setiap *binary classifier*, *boundary* yang dibentuk dapat diilustrasikan pada gambar 3.4. Ada dua macam *boundary* yang dibentuk, pertama, yang terletak di antara *upper bound* (ub) dan *lower bound* (lb) adalah sampel-sampel positif dari kelas yang bersangkutan. *Boundary* jenis ini berguna ketika sampel-sampel kelas yang bersangkutan berkumpul di antara *boundary* tersebut, sementara sampel-sampel negatif (sampel-sampel) kelas lain, terletak di *boundary* lain atau menyebar. *Boundary* jenis kedua adalah yang terletak di antara *lower bound* dan *upper bound* adalah sampel negatif (bukan dari kelas yang bersangkutan). Hal ini untuk mengatasi bahwa data sampel untuk kelas yang bersangkutan sangat menyebar, sementara sampel dari kelas-kelas lain lebih fokus pada interval tadi.



Gambar 3-4 Ilustrasi pembentukan *Boundary* dari algoritma *Multi Boundary Classifier*

Representasi *multi boundary classifier* mengikuti dengan representasi base classifier yang diensemble menggunakan *Adaboos.MH*. Yaitu setiap classifier terdiri dari K , binary classifier. Aturan klasifikasi algoritma *multi boundary clasiffier* dapat dirumuskan dengan persamaan berikut:

$$p_l = \begin{cases} +1, & \text{if } lb_l < (x_{i,l}) < ub_l \\ -1, & \text{otherwise} \end{cases} \quad (3.1)$$

$$prediction_l(x_i) = p_l \cdot v_l \quad (3.2)$$

Dimana lb dan ub adalah *lower bound* dan *upper bound* untuk kelas yang bersangkutan, v adalah nilai vote $v=\{-1,+1\}$ yang menentukan jenis *boundary* dari kelas tersebut.

Selanjutnya, dalam proses *learning* perlu dicari *boundary* yang paling optimal untuk setiap *binary classifier*. Kriteria yang digunakan untuk menentukan *boundary* yang paling optimal adalah untuk suatu kelas l adalah dengan menggunakan nilai *classification edges*:

$$\gamma_l(\phi) = \sum_{i=1}^n w_{i,l} \cdot v_l \cdot \phi_l(x_i) \cdot y_{i,l} \quad (3.3)$$

Proses pemebentukan *boundary* dilakukan dengan dimulai dari titik sampel pertama. Sebelumnya, sampel perlu diurutkan (*sort*) dari kecil ke besar berdasarkan nilai fiturnya. Secara iteratif *lower bound* dinaikkan nilainya. Untuk setiap nilai *lower bound* dicari nilai *upper bound* yang memberikan nilai *edge* paling besar. Proses ini

diiterasi terus menerus sampai nilai *lower bound* dicobakan pada sampel terakhir. Proses tersebut dapat dilakukan dengan langkah-langkah berikut:

1. Tentukan titik start *lower bound* (s). Biasanya merupakan *sampel* pertama
2. Inisiasi *lower bound* dan *upper bound*

$$lb_l = -numeric(max). \quad (3.4)$$

$$ub_l = +numeric(max). \quad (3.5)$$

numeric (max), merupakan nilai numeric maksimal yang dapat dicapai oleh tipe data yang digunakan.

3. Inisiasi nilai *1-side half edge* ($\mu_{1_{l+}}$, dan $\mu_{1_{l-}}$), dan *2-side half edge* ($\mu_{2_{l+}}$, dan $\mu_{2_{l-}}$), dan *2-side half edge* terbaik ($best_mu_{2_{l+}}$, dan $best_mu_{2_{l-}}$). *1-side edge* merupakan nilai *edge* untuk suatu variasi nilai *lower bound*, sedangkan *2-side edge* merupakan nilai *edge* untuk suatu variasi nilai *lower bound* dan *upperbound*. Dalam proses *learning*, yang dicari adalah nilai *2-side half edge* terbaik, yaitu yang nilainya paling besar

$$\mu_{1_{l+}} \leftarrow \mu_{1_{l-}} \leftarrow 0 \quad (3.6)$$

$$best_mu_{2_{l+}} \leftarrow best_mu_{2_{l-}} \leftarrow \mu_{2_{l-}} \leftarrow \mu_{2_{l+}} \leftarrow 0 \quad (3.7)$$

4. Inisiasi *lower index* (i) dan *upper index* (j) untuk proses iterasi :

$$i \leftarrow index(s). \quad (3.8)$$

$$j \leftarrow index(N). \quad (3.9)$$

5. Iterasikan nilai *lower bound* ke sampel berikutnya.

$$i \leftarrow i + 1 \quad (3.10)$$

- a. Tambahkan bobot sampel ke *1-sides half edge*

Jika $w_{i,l} \cdot y_{i,l} < 0$

$$\mu 1_{l+} \leftarrow \mu 1_{l+} - (w_{i,l} \cdot y_{i,l}) \quad (3.11)$$

Jika tidak

$$\mu 1_{l-} \leftarrow \mu 1_{l-} + (w_{i,l} \cdot y_{i,l}) \quad (3.12)$$

b. Salin nilai *1-side half edge* ke *2-side half edge*.

$$\mu 1_{l+} \leftarrow \mu 1_{l+} \quad (3.13)$$

$$\mu 2_{l-} \leftarrow \mu 1_{l-} \quad (3.14)$$

c. Tentukan nilai v berdasarkan nilai *2-side half edge*

$$v_l \leftarrow \text{sign}(\mu 2_{l+} - \mu 2_{l-}) \quad (3.15)$$

d. Cek apakah nilai *2-side half* saat ini lebih baik dari *half edge* sebelumnya, dan lakukan update *best half edge* serta nilai batas *lower bound (lb)*.

Jika $(\mu 2_{l+} - \mu 2_{l-}) \cdot v_l > (best_{\mu_{l+}} - best_{\mu_{l-}}) \cdot best_{v_l}$ maka:

$$best_{\mu_{l+}} \leftarrow \mu 2_{l+} \quad (3.16)$$

$$best_{\mu_{l-}} \leftarrow \mu 2_{l+} \quad (3.17)$$

$$best_{v_l} \leftarrow v_l \quad (3.18)$$

$$lb_l \leftarrow \frac{1}{2} (feature(x_i) + feature(x_{i-1})) \quad (3.19)$$

6. Cek kemungkinan pelebaran nilai *upper bound*, dengan nilai *lb* yang diperoleh dari proses pada langkah 5. Pertama kali nilai j diberi nilai N . Lalu dilakukan iterasi selama nilai j tidak sama dengan i (sampel terakhir), lakukan proses berikut ini.

$$j = j - 1 \quad (3.20)$$

a. Tambahkan bobot sampel ke *2-side half edges*.

Jika $w_{j,l} \cdot y_{j,l} < 0$

$$\mu_{2_{l+}} \leftarrow \mu_{2_{l+}} - (w_{j,l} \cdot y_{j,l}) \quad (3.21)$$

Jika tidak

$$\mu_{2_{l-}} \leftarrow \mu_{2_{l-}} + (w_{j,l} \cdot y_{j,l}) \quad (3.22)$$

b. Tentukan nilai v berdasarkan nilai *2-side half edges*.

$$v_l = \text{sign}(\mu_{2_{l+}} - \mu_{2_{l-}}) \quad (3.23)$$

c. Cek apakah *half edge* saat ini lebih baik dari *half edge* sebelumnya, dan lakukan update *best half edge* serta nilai batas *upper bound (ub)*.

Jika $(\mu_{2_{l+}} - \mu_{2_{l-}}) \cdot v_l > (best_{\mu_{l+}} - best_{\mu_{l-}}) \cdot best_{v_l}$ maka :

$$best_{\mu_{l+}} \leftarrow \mu_{2_{l+}} \quad (3.24)$$

$$best_{\mu_{l-}} \leftarrow \mu_{2_{l-}} \quad (3.25)$$

$$best_{v_l} \leftarrow v_l \quad (3.26)$$

$$ub_l \leftarrow \frac{1}{2} (feature(x_i) + feature(x_{i+1})) \quad (3.27)$$

7. Langkah 5 dan 6 dilakukan untuk semua label kelas l ($1, 2, \dots, N$).
8. Nilai *edge* untuk satu kesatuan *classifier* yang terdiri dari K *binary classifier* dapat dihitung dengan persamaan berikut:

$$\chi(\phi) = \sum_{l=1}^K best_{v_l} \cdot best_{\mu_l} \quad (3.28)$$

Secara iterative, algoritma *Multi Boundary Classifier* untuk suatu kelas l dapat dilakukan dengan *psedudo code* pada gambar 3-5.

Algorithm 3.1 Multi Boundary Classifier

```

Input : { (x1, y1), (x2, y2), ..., (xN, yN) } , x : feature value, y: class label : input is sorted based on feature value.
//Variable
lidx = 0 //lower index
uidx = 0 // upper index
 $\mu_{1+} = \mu_{1-} = \mu_{2+} = \mu_{2-} = 0$ 
 $best\_ \mu_{2+} = best\_ \mu_{2-} = 0$ 
 $v_l = 0$ 
 $best\_v_l = 0$ 
lbl = numeric(min) // lower bound (for class l)
ubl = numeric(max) //upper bound (for class l)
for lidx = 0 to N
  if ((wlidx . ylidx) > 0)
     $\mu_{1+} = \mu_{1+} - (w_{lidx} . y_{lidx})$ 
  else
     $\mu_{1-} = \mu_{1-} + (w_{lidx} . y_{lidx})$ 
  end if
   $\mu_{2+} = \mu_{1+}$ 
   $\mu_{2-} = \mu_{1-}$ 
   $v_l = sign(\mu_{2+} - \mu_{2-})$ 
  if ( ( $\mu_{2+} - \mu_{2-}$ ) . vl) > ( (best $\mu_{1+}$  - best $\mu_{1-}$ ) . best_vl)
    best_ $\mu_{1+}$  =  $\mu_{2+}$ 
    best_ $\mu_{1-}$  =  $\mu_{2-}$ 
    best_vl = vl
    lbl =  $\frac{1}{2} ( (x_{lidx}) + (x_{lidx-1}) )$ 
  end if
  for uidx N to lidx
    if ((wuidx . yuidx) > 0)
       $\mu_{2+} = \mu_{2+} - (w_{uidx} . y_{uidx})$ 
    else
       $\mu_{2-} = \mu_{2-} + (w_{uidx} . y_{uidx})$ 
    end if
     $v_l = sign(\mu_{2+} - \mu_{2-})$ 
    if ( ( $\mu_{2+} - \mu_{2-}$ ) . vl) > ( (best $\mu_{1+}$  - best $\mu_{1-}$ ) . best_vl)
      best_ $\mu_{1+}$  =  $\mu_{2+}$ 
      best_ $\mu_{1-}$  =  $\mu_{2-}$ 
      best_vl = vl
      ubl =  $\frac{1}{2} ( (x_{uidx}) + (x_{uidx+1}) )$ 
    end if
  end for
end for

```

Gambar 3-5 Pseudo code algoritma Multi Boundary Classifier

Pada paragraf sebelumnya telah dijelaskan langkah-langkah pembentukan *boundary* dari suatu titik *start*. Permasalahan sekarang adalah bagaimana cara untuk mendapatkan *boundary* yang paling optimal. Penentuannya tidak bisa dilakukan dari sembarang titik *start* karena akan mempengaruhi hasil *boundary* yang terbentuk.

Oleh sebab itu, diperlukan titik *start* yang tepat dalam membentuk *boundary*. Cara yang paling sederhana untuk menentukan himpunan titik *start* tersebut adalah dengan menggunakan seluruh titik sampel. Dengan cara tersebut akan diperoleh N buah kandidat *boundary* untuk setiap kelas. Nantinya, dari N calon *boundary* tersebut akan dipilih satu *boundary* terbaik. Kriteria yang digunakan untuk memilih *boundary* terbaik adalah dengan mencari nilai *edge* yang paling besar. Teknik lain untuk memilih titik-titik *start* ini adalah dengan menggunakan *seed point*. *Seed point* bisa diperoleh dengan pemilihan secara acak dari sampel, atau dengan ditentukan pada setiap interval tertentu.

3.3. *Ensemble Multi Boundary Classifier Menggunakan Adaboost.MH*

Pada penelitian ini, *multi boundary classifier* akan di-*ensemble* menggunakan metode *Adaboost.MH*. Penjelasan lengkap tentang metode *Adaboost.MH* dapat dilihat pada bab 2. *Ensemble* ini dilakukan untuk membentuk suatu *Strong classifier* yang terdiri dari banyak *base classifier*.

Metode *ensemble* ini berguna untuk meningkatkan performa *base classifier* yang dibentuk, karena saling melengkapi satu sama lain. Kesalahan klasifikasi oleh suatu *classifier* (dari suatu fitur) dapat diatasi dengan klasifikasi oleh *classifier* lain dengan fitur yang berbeda. Setiap *classifier* yang digabungkan/*diensemble* memiliki bobot masing-masing. Bobot ini nantinya akan digunakan saat *voting*, sehingga hasil akhir dari prediksi yang diberikan oleh *classifier* gabungan adalah total *weighted voting* dari *classifier-classifier* yang digabungkan. Berikut ini adalah langkah-langkah *ensemble Multi Boundary Classifier* menggunakan algoritma *Adaboost.MH*.

1. Diberikan himpunan sampel dengan $X = \{x_1, x_2, \dots, x_n\}$, label kelas dari sampel $Y = \{y_1, y_2, \dots, y_n\}$. Setiap sampel x_i merupakan vektor dengan m elemen, dimana m adalah jumlah total fitur pada sampel. Setiap label (y_i) adalah vektor dengan k elemen, dimana k adalah jumlah kelas.
2. Lakukan inisiasi bobot sampel yang dilakukan dengan persamaan berikut:

$$w_{i,l} = \begin{cases} \frac{1}{2n} , & \text{if } (y_{i,l} = 1) \\ \frac{1}{2n(K-1)} , & \text{otherwise } (y_{i,l} = -1) \end{cases} \quad (3.29)$$

Bobot dari setiap sampel adalah vektor dengan k elemen, seperti halnya label kelas.

3. *Generate Multi Boundary Classifier* sebanyak m *classifier*.
4. Lakukan *training* pada setiap *classifier*. Dari setiap *classifier* yang di-*training* akan diperoleh nilai *classification edge*:

$$\gamma(\varphi) = \sum_{l=1}^K best_v_l \cdot best_μ_l \quad (3.30)$$

Dimana $best_μ_l$ terdiri dari $best_μ_{l+}$ dan $best_μ_{l-}$

5. Pilih *classifier* dengan nilai *classification edge* tertinggi

$$h(x)_t = h(x)_i, \{i = \arg \max_{i=1 \text{ to } m} (\gamma(h(x)_i))\} \quad (3.31)$$

6. Hitung nilai α_t

$$\alpha_t = \frac{1}{2} \ln \frac{\sum_{l=1}^K (\mu_{l+} | \{v_l = +1\} + \mu_{l-} | \{v_l = -1\})}{\sum_{l=1}^K (\mu_{l-} | \{v_l = +1\} + \mu_{l+} | \{v_l = -1\})} \quad (3.32)$$

Nilai μ_{l+} dan μ_{l-} serta v_l dalam hal ini adalah nilai terbaik yang diperoleh dari proses *training* yaitu $best_μ_{l+}$, $best_μ_{l-}$, dan $best_v$.

7. Lakukan update nilai bobot sampel untuk proses selanjutnya. *Update* bobot sampel dilakukan pada setiap elemen vektor bobot

$$w_{i,l}^{t+1} = w_{i,l}^t \frac{\exp(-\alpha \cdot h_l^t(x_i) \cdot y_{i,l})}{\sum_{i=1}^n \sum_{l=1}^K w_{i,l}^t \exp(-\alpha \cdot h_l^t(x_i) \cdot y_{i,l})} \quad (3.33)$$

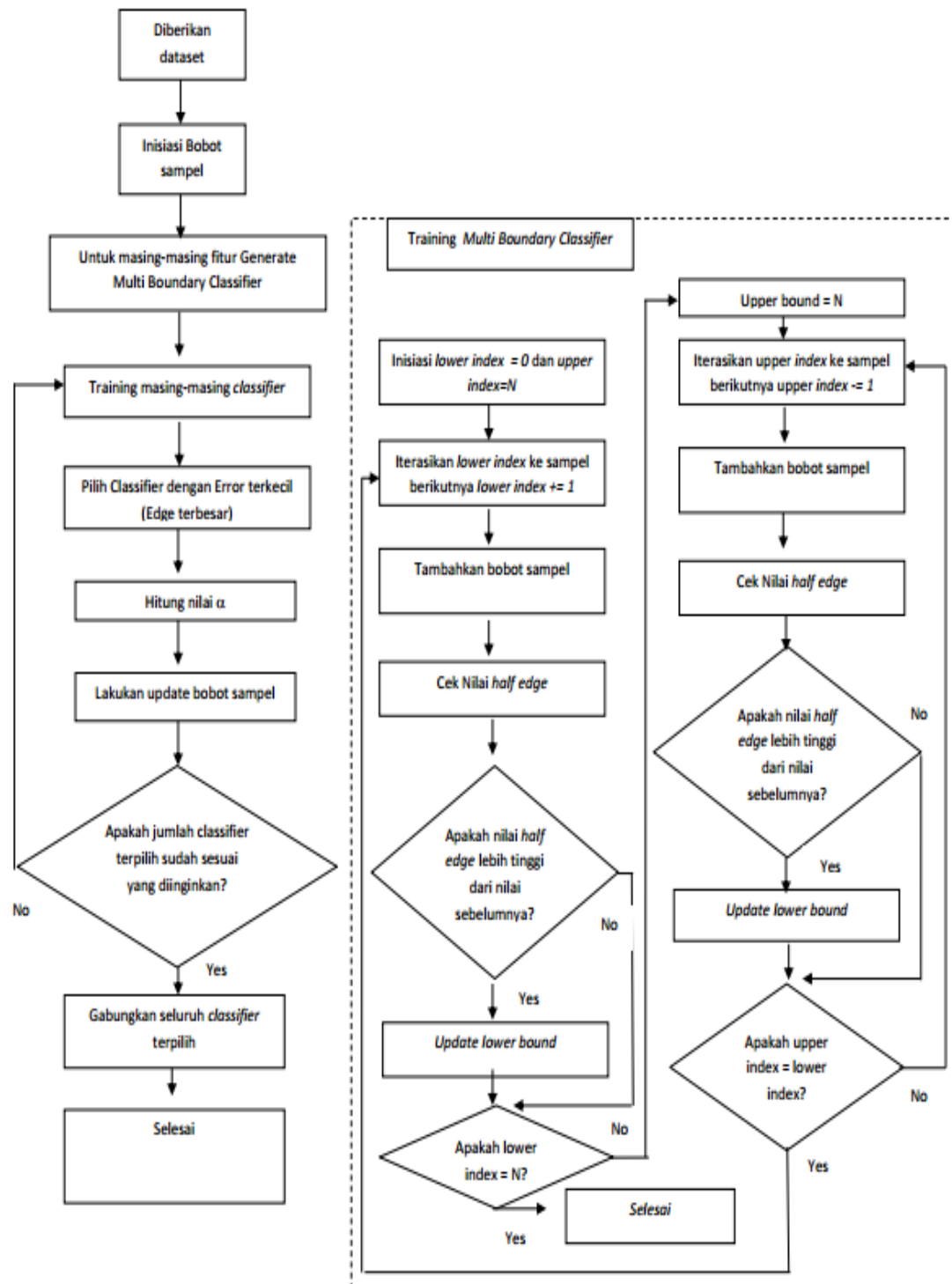
8. Langkah nomor 4 sampai 7 dilakukan sebanyak T iterasi. Dimana T adalah jumlah *classifier* yang ingin di-*ensemble* digabungkan.
9. *Classifier* hasil gabungan dari proses ini adalah:

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (3.34)$$

10. *Classifier* $h_t(x)$ adalah *classifier* dengan k elemen. Artinya $h_t(x)$ memiliki k *binary classifier*, 1 untuk masing – masing kelas. Untuk mencari kelas hasil prediksi dari suatu sampel yang diberikan, dicari nilai maksimal dari hasil nilai $H(x)$. Atau dapat ditulis sebagai berikut

$$prediksi(x) = \arg \max_l \left(\sum_{t=1}^T \alpha_t h_{t,l}(x) \right) \quad (3.35)$$

Framework metode *Adaboost.MH* berbasis *Multi Boundary Classifier* yang diajukan dalam penelitian ini dapat dilihat pada gambar 3-6.



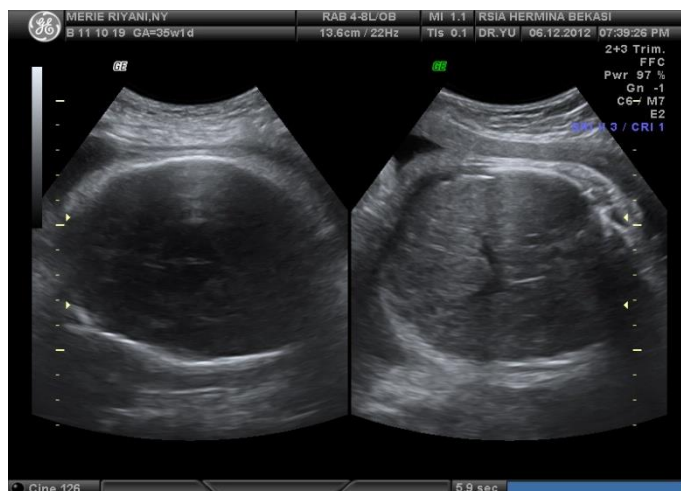
Gambar 3-6 Diagram algoritma *Adaboost.MH* berbasis *Multi Boundary Classifier*

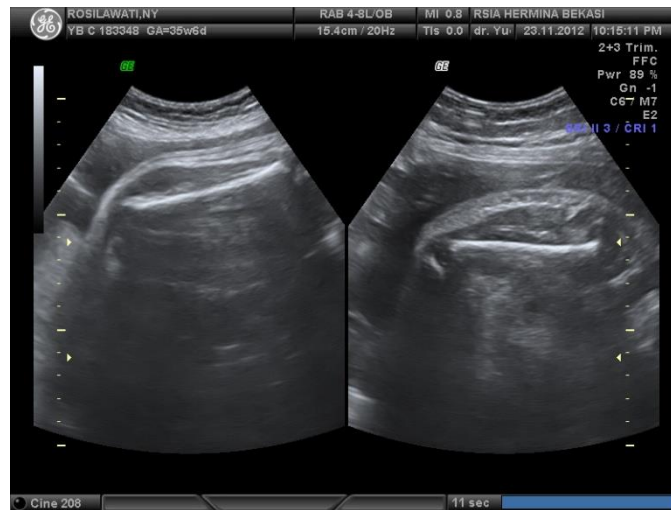
Bab 4 Metodologi Penelitian

4.1 Dataset

4.1.1 Dataset Ultrasonografi (USG)

Data yang digunakan dalam penelitian ini adalah data citra ultrasonografi (USG) hasil pemeriksaan janin dalam kandungan. Dataset yang digunakan terdiri dari gambar-gambar organ bayi yang meliputi kepala, perut, *femur* (tulang paha), dan *humerus* (tulang lengan). Dataset yang akan digunakan disertai dengan anotasi ahli sehingga dapat digunakan untuk validasi pengukuran otomatis. Dari organ kepala, nantinya dapat diukur besaran *biparietal diameter* (BPD) yang merupakan diameter lingkaran kepala, dan *head circumference* (HC) yang merupakan keliling lingkaran kepala. Dari organ perut akan diukur besaran *Abdominal Circumference* (AC) yang merupakan lingkaran perut. Sedangkan dari *femur* dan *humerus* akan diukur besaran *femur length* (FL) dan *humerus length* (HL). Akan tetapi dalam penelitian ini dibatasi pada aproksimasi bentuknya saja, tidak sampai dilakukan pengukuran besaran-besaran tersebut. Contoh gambar USG yang akan digunakan pada penelitian ini dapat dilihat pada gambar 4.1





Gambar 4-1 Contoh tampilan organ janin dalam citra USG, kepala (kiri atas), perut (kanan atas), *femur* (kiri bawah), dan *humerus* (kanan bawah)

4.1.2 Dataset Benchmark

Selain menggunakan data USG, dalam penelitian ini juga akan digunakan dataset *benchmark*. Dataset *benchmark* digunakan untuk melakukan verifikasi algoritma yang digunakan dalam proses klasifikasi. Dataset *benchmark* yang digunakan adalah sebagai berikut:

1. Dataset *digital handwritten* dari *United States Postal Services* (USPS)
2. Dataset *digital handwritten* dari Mixed National Institute of Standards and Technology database (MNIST).

Kedua dataset tersebut merupakan dataset gambar angka dari 0 sampai 9 yang diambil dari tulisan tangan. Pada dataset USPS, setiap gambar berukuran 16x16, sedangkan pada dataset MNIST setiap gambar berukuran 28x28. Kedua dataset ini tersedia dalam *repository multiboost* dan UCI [20][21]. Contoh gambar tulisan tangan dari data USPS dan MNIST dapat dilihat pada gambar 4.2.

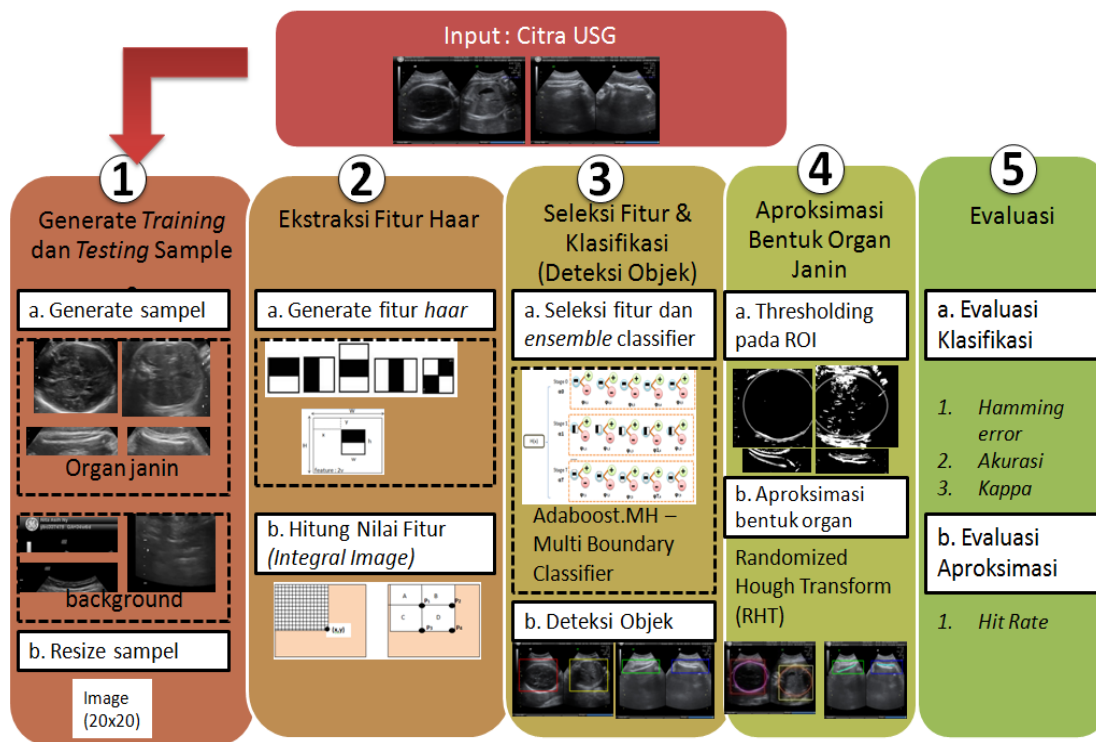


Gambar 4-2 Dataset Benchmark USPS dan MNIST.

4.2 Kerangka Kerja Penelitian.

Pada gambar 4.3 ditampilkan kerangka kerja penelitian. Data masukan pada penelitian ini adalah citra USG. Selanjutnya citra masukan tersebut akan diolah melalui lima langkah seperti pada gambar 4.3. Pertama kali, dilakukan pembuatan *training* dan *testing* sampel. Pembuatan sampel dilakukan dengan cara *cropping* bagian dari gambar yang merupakan organ janin, baik kepala, perut, tulang paha, maupun tulang lengan. Selain dilakukan pembuatan *training* dan *testing* sampel pada organ janin, juga dilakukan pembuatan *training* dan *testing* sampel pada citra USG yang menjadi *background* (bukan objek organ). Selanjutnya, seluruh sampel dilakukan normalisasi, yaitu ukurannya disamakan menjadi 20x20 *pixel*.

Tahap kedua, dilakukan ekstraksi fitur. Dalam penelitian ini, penulis menggunakan fitur *Haar*. Fitur *Haar* merupakan fitur yang diperoleh dengan membuat suatu area berbentuk persegi panjang, lalu menghitung nilai *pixel* di area tersebut. Penjelasan mengenai fitur ini, akan dijelaskan pada subbab selanjutnya. Pertama kali di-*generate* fitur *Haar* untuk luas jendela yang diberikan (20x20 untuk sampel dari citra USG). Setelah di-*generate* dilakukan perhitungan nilai fitur. Perhitungan nilai fitur ini dilakukan menggunakan metode *integral image*.



Gambar 4-3 Kerangka Kerja Penelitian

Setelah diperoleh keseluruhan fitur *Haar*, selanjutnya dilakukan seleksi fitur sekaligus pembentukan *classifier*. Untuk setiap fitur *Haar*, akan dibuat satu *classifier*. Selanjutnya seluruh *classifier* akan di-*training*. Untuk setiap iterasi *training*, akan diambil *classifier* dengan *error* paling kecil. Setelah T iterasi akan diperoleh T *classifier* dan fitur terbaik yang akan digunakan untuk proses deteksi objek. Seluruh *classifier* yang diperoleh tadi digabungkan menggunakan metode *Adaboost.MH*. Penjelasan mengenai proses ini akan dijelaskan pada subbab selanjutnya. *Classifier* yang diperoleh tadi nantinya akan digunakan dalam proses deteksi objek. Dalam proses deteksi objek, sistem akan melakukan *scanning* ke seluruh citra USG. Dalam proses tersebut, setiap *window* akan dievaluasi menggunakan *classifier*, apakah termasuk dalam kelas *background*, atau organ janin.

Setelah objek berhasil dideteksi, proses selanjutnya adalah dilakukan aproksimasi bentuk organ janin. Untuk proses pengukuran data biometri akan digunakan metode *Randomize Hough Transform*, (RHT). Metode ini hanya bekerja

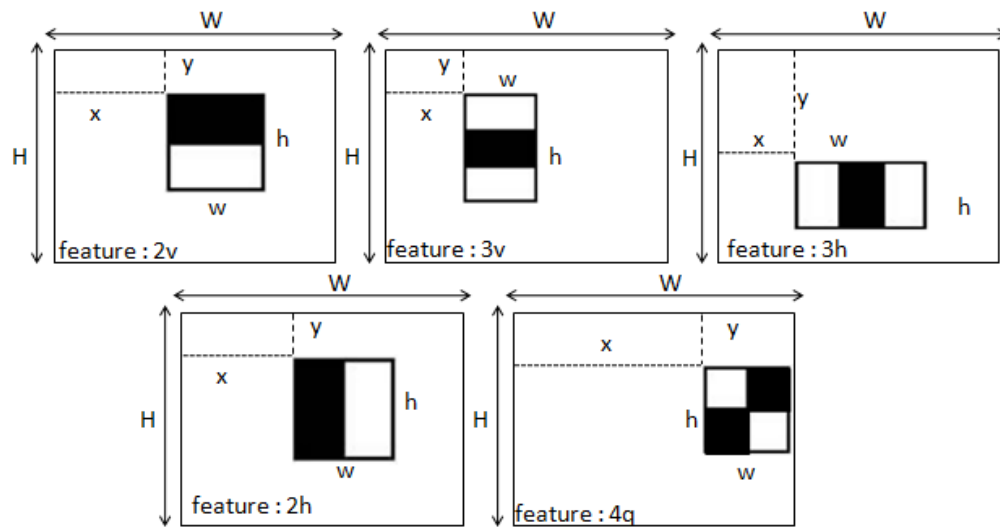
untuk segmentasi objek pada domain citra biner. Oleh sebab itu, akan dilakukan *thresholding* pada citra sebelum diterapkan metode RHT. Akan tetapi, proses *thresholding* cukup dilakukan pada *Region of Interest* (ROI), yaitu area yang terdeteksi sebagai organ. Dalam proses aproksimasi organ kepala dan perut akan diaproksimasi dengan bentuk kurva elips, sedangkan organ tulang pada dan tulang lengan akan diaproksimasi menggunakan bentuk garis.

Rangkaian terakhir dari kerangka kerja penelitian ini adalah evaluasi. Dalam penelitian ini akan dilakukan dua macam evaluasi. Pertama, dilakukan evaluasi terhadap proses deteksi organ janin pada citra. Evaluasi ini dilakukan untuk menguji performa klasifikasi yang digunakan. Pada proses evaluasi ini akan digunakan tiga macam kriteria, yaitu *Hamming error*, akurasi, dan koefisien *kappa*. Proses kedua adalah evaluasi pada tingkat aproksimasi organ janin yang dilakukan. Kriteria yang digunakan untuk proses evaluasi ini adalah tingkat *hit* (*hit rate*). Pada bab selanjutnya akan dijelaskan lebih detail mengenai tahapan penelitian yang dilakukan.

4.3 Ekstraksi Fitur *Haar*

4.3.1 Generate Fitur *Haar*

Seperti yang dijelaskan pada subbab sebelumnya bahwa setelah data citra USG diambil, selanjutnya dilakukan *generate* fitur *Haar*. Pada penelitian ini akan disediakan tujuh fitur *Haar like* dasar, seperti yang ditunjukkan pada gambar 4.4. Dalam proses *generate* ini, dari fitur *Haar like* dasar akan di-*generate Haar like* dengan berbagai ukuran, posisi, maupun sudut rotasi terhadap *Haar like* dasar yang digunakan. Ilustrasi *generate Haar like features* dapat dilihat pada bab 2.



Gambar 4-4 Proses Generate Fitur Haar

Pada gambar 4.4 dapat dilihat ada beberapa variabel (x, y, w, h) dalam proses *generate* fitur *Haar*. Variabel x dan y adalah posisi translasi fitur *Haar* dari posisi piksel $(0,0)$ pada gambar. W dan H adalah nilai lebar dan tinggi fitur *Haar like*. Dari suatu fitur *Haar* dasar akan dibuat banyak fitur dimana memiliki kombinasi nilai x, y, W, H dan yang unik. Tujuannya adalah melalui seluruh kernel/fitur tersebut, maka diperoleh informasi yang paling representatif dari suatu citra masukan.

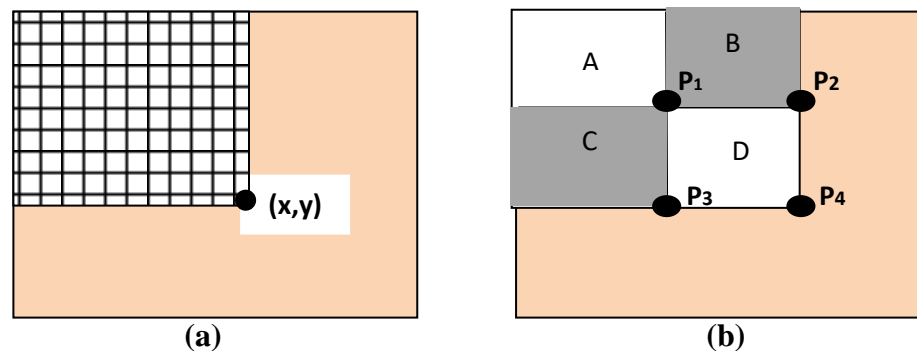
Sesuai dengan formula *generate* fitur *Haar*, bahwa jumlah fitur *Haar* yang *degenerate* tergantung dari ukuran *window* citra yang digunakan. Semakin besar ukuran *window*, maka semakin banyak pula jumlah fitur *Haar* yang akan di-*generate* untuk diseleksi. Pada penelitian ini digunakan tiga macam *dataset*. Ketiga *dataset* tersebut memiliki ukuran *window* yang berbeda-beda (setelah dilakukan normalisasi), sehingga menghasilkan jumlah fitur *Haar* yang berbeda pula. Jumlah fitur yang di-*generate* untuk masing-masing *dataset* yang digunakan, dapat dilihat pada Tabel 4.1.

Tabel 4-1 Jumlah Fitur Haar yang Digenerate Untuk Masing-Masing Dataset

Dataset	Ukuran Window	Jumlah fitur haar yang di-generate					
		2v	2h	3v	3h	4q	Total
USG	20x20	17100	17100	10830	10830	8100	63960
USPS	16x16	6720	6720	4200	4200	3136	24976
MNIST	28x28	68796	68796	44226	44226	33124	259168

4.3.2 Penghitungan Nilai Fitur *Haar*

Setelah dilakukan *generate* kernel fitur *Haar*, dilakukan penghitungan nilai fitur *Haar*. Untuk melakukan penghitungan fitur *Haar* digunakan metode *integral image*. *Integral image* adalah suatu metode yang digunakan untuk menghitung jumlah (integral) nilai *pixel* pada suatu area tertentu. Area yang dimaksud adalah area yang terbentuk dari posisi (0,0) sampai suatu posisi (x,y) tertentu. Prinsip kerja *integral image* dan penghitungan fitur *haar* dapat dilihat pada gambar 4.6 (a)



Gambar 4-5 (a) Proses perhitungan nilai integral image (b) Perhitungan nilai fitur haar menggunakan integral image

Gambar 4.6(a) diilustrasikan proses perhitungan nilai *integral image*. Secara formal nilai *integral image* dapat dihitung dengan menggunakan persamaan berikut:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (4.1)$$

Perhitungan nilai *integral image* dapat juga dilakukan dengan metode rekursif seperti persamaan berikut:

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (4.2)$$

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (4.3)$$

Dimana $ii(x, y)$ adalah nilai *integral image* pada titik (x, y) , dan $i(x, y)$ adalah nilai *pixel* pada titik (x, y) . Pada gambar 4.6 (b) diilustrasikan perhitungan nilai fitur *Haar* untuk jenis empat persegi panjang. Nilai fitur *Haar* empat persegi panjang tersebut dapat dihitung dengan penjabaran pada paragraf berikut.

Nilai fitur *Haar* dapat diperoleh dengan jumlah nilai *pixel* pada area A dan D, dikurangi dengan jumlah nilai *pixel* pada area B dan C. Untuk menghitung jumlah nilai *pixel* pada seluruh area tersebut dapat dilakukan dengan menggunakan formula *integral image*. Secara keseluruhan, proses perhitungan nilai fitur tersebut, dapat dilakukan dengan proses berikut:

$$\text{Nilai haar} = A + D - B - C \quad (4.4)$$

$$A = ii(p1) \quad (4.5)$$

$$B = ii(p2) - ii(p1) \quad (4.6)$$

$$C = ii(p3) - ii(p1) \quad (4.7)$$

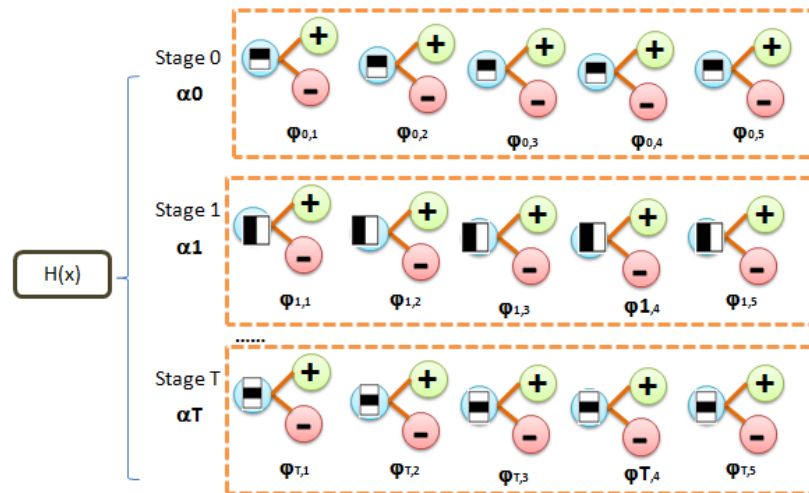
$$D = ii(p1) + ii(p4) - ii(p2) - ii(p3) \quad (4.8)$$

4.3 Seleksi Fitur Dan Klasifikasi

4.3.1 Seleksi Fitur dan *Ensemble Classifier*

Setelah seluruh fitur *Haar* diekstrak dari sampel, proses selanjutnya adalah seleksi fitur dan *ensemble classifier*. Seleksi fitur diperlukan karena tidak mungkin seluruh *Haar* fitur yang telah di-generate pada proses sebelumnya digunakan

seluruhnya. Selain akan menimbulkan kompleksitas komputasi yang berat (lama), pastinya juga akan menambah kapasitas *memory* yang cukup besar. Pembentukan *classifier* dilakukan dengan pendekatan *ensemble*, yaitu menggabungkan beberapa *classifier* secara bersamaan untuk mendapatkan hasil *classifier* gabungan yang akurat.

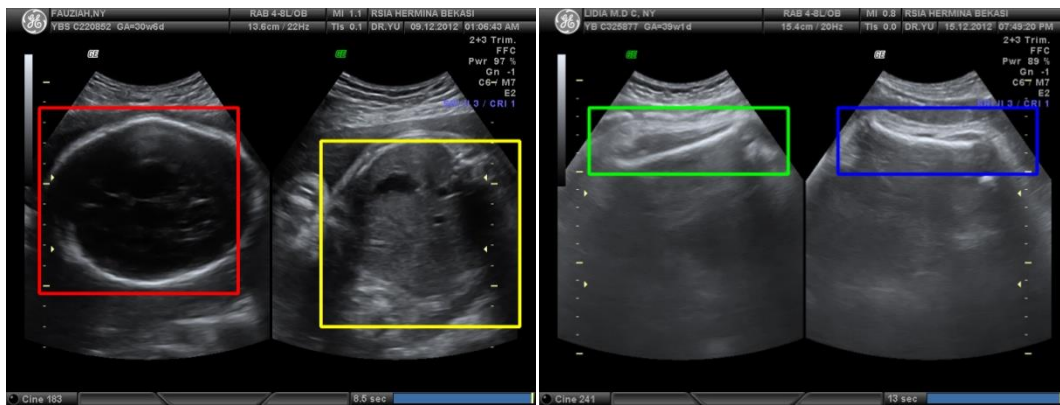


Gambar 4-6 Ilustrasi ensemble Multi Boundary Classifier dengan algoritma Adaboost.MH. Proses seleksi fitur dan *ensemble classifier* merupakan proses yang berjalan dalam satu kesatuan. Proses seleksi fitur ini diilustrasikan pada gambar 4.7. Untuk setiap fitur *Haar* akan dibentuk *base classifier*. *Base classifier* dalam hal ini adalah *multi boundary classifier*. Dalam proses *ensemble* digunakan algoritma *Adaboost Multi Class (Multi Label) based on Hamming Lost Error (Adaboost.MH)*.

Prinsip utama proses ini adalah melakukan *training* pada seluruh *base classifier* yang dibentuk. Dalam setiap iterasi akan diambil *classifier* terbaik, yaitu *classifier* dengan *error* paling kecil. Selanjutnya dilakukan *update* bobot sampel (sampel diberikan bobot awal terlebih dahulu). Sampel yang salah klasifikasi akan diperbesar bobotnya, sedangkan sampel yang benar, akan diperkecil bobotnya. Hasil dari proses ini adalah suatu *classifier* gabungan dari beberapa *base classifier* yang digabungkan. Langkah-langkah proses seleksi fitur dan *ensemble classifier* ini dijelaskan lebih detail pada sub bab 3.3.

4.3.1 Deteksi Objek

Setelah diperoleh *classifier* dari proses sebelumnya, selanjutnya *classifier* tersebut digunakan untuk proses deteksi objek. Pada proses ini, sistem akan melakukan *scanning* dari titik (0,0) pada citra sampai titik kanan bawah. *Scanning* dilakukan untuk berbagai variasi ukuran *window*. Ukuran *window* terkecil adalah (300x100) sedangkan ukuran *window* terbesar adalah (300x300). Setiap ukuran *window* akan dicobakan pada berbagai titik koordinat *window* tersebut. Untuk proses perpindahan dari satu titik ke titik berikutnya menggunakan jeda 25 *pixel* untuk setiap sumbu. Selanjutnya setiap *window* tersebut akan diklasifikasikan, apakah dikenali sebagai objek organ janin (kepala, perut, tulang paha, tulang lengan), atau diklasifikasikan sebagai *background*.



Gambar 4-7 Hasil deteksi objek organ pada citra USG

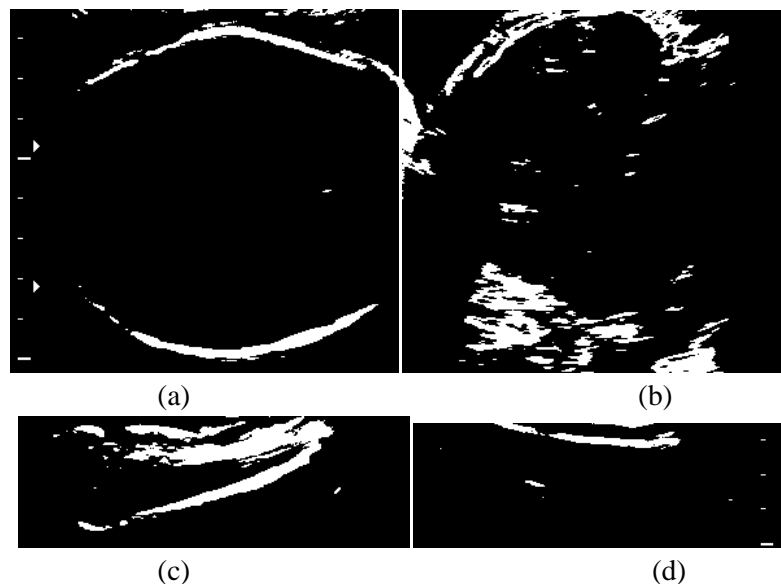
Pada gambar 4.8 ditampilkan hasil deteksi dari sistem yang dibuat. Pada gambar 4.8 kiri ditampilkan hasil deteksi objek organ kepala (warna merah), dan organ perut (warna kuning). Pada gambar 4.8 kanan ditampilkan hasil deteksi objek organ tulang paha (warna hijau), dan organ tulang lengan (warna biru).

4.4 Aproksimasi Bentuk Organ Janin

4.4.1 Thresholding

Thresholding merupakan metode filter suatu nilai berdasarkan suatu nilai batas. *Thresholding* dilakukan untuk menghilangkan *noise* di daerah tempat objek terdeteksi. Setelah proses deteksi masih ada *noise* pada area organ, yaitu titik-titik

pixel yang berwarna abu-abu. Oleh sebab itu perlu dihilangkan agar proses pengukuran dapat dilakukan dengan tepat. Selain itu, metode pengukuran yang akan digunakan yaitu *Randomize Hough Transform* (RHT) tidak dapat diaplikasikan pada selain citra biner. Cara kerja algoritma *thresholding* cukup sederhana, yaitu hanya menggunakan batas nilai tertentu. Contoh perubahan citra dari *grayscale* ke dalam bentuk *binary* menggunakan *thresholding* pada ROI yang terdeteksi sebagai objek organ janin dapat dilihat pada gambar 4.9

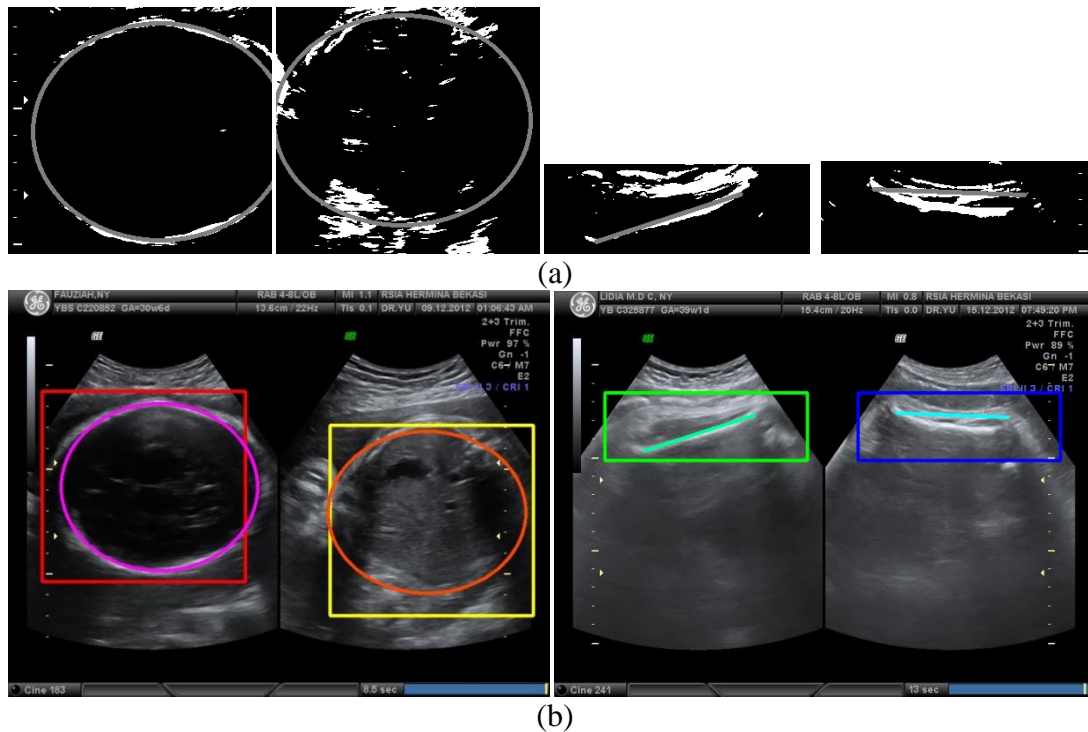


Gambar 4-8 Hasil Konversi Citra dari Grayscale ke Binary Image (a) Kepala (b) Perut (c) Tulang paha (d) Tulang Lengan

4.4.2 *Randomize Hough Transform* (RHT)

Sesuai dengan penjelasan yang dituliskan pada bab 2, *Randomize Hough Transform* (RHT) merupakan salah satu metode aproksimasi kurva. Dalam penelitian ini RHT akan digunakan untuk mengaproksimasi organ kepala, perut organ femur, femur, dan humerus dengan bentuk kurva. Sebelum melakukan aproksimasi tentunya sudah dilakukan proses pendeteksian dan *thresholding* yang dilakukan pada tahapan sebelumnya. Ada dua kurva yang akan diaproksimasi dalam penelitian ini, yaitu kurva garis lurus dan kurva elips. Aproksimasi garis lurus digunakan untuk organ femur dan humerus. Sedangkan aproksimasi elips digunakan untuk organ kepala dan

perut. Proses aproksimasi dilakukan pada citra binary ROI, selanjutnya hasil aproksimasi kurva tersebut diaplikasikan ke citra asli. Contoh hasil aproksimasi untuk keliling linkar kepala dapat dilihat pada gambar 4.10.



Gambar 4-9 Aproksimasi elips dan garis pada organ janin (a) pada ROI (b) Hasil dari ROI dipalikasi ke citra asli.

4.5 Metode Evaluasi

4.5.1 Metode Evaluasi Klasifikasi

Untuk mengukur performa dari metode klasifikasi yang sudah diusulkan pada penelitian ini, hasil yang diperoleh dievaluasi dengan menggunakan beberapa metode pengujian, yaitu *Hamming error*, akurasi dan koefisien *kappa*. Selain itu, performa klasifikasi juga akan diuji menggunakan *Friedman test* untuk menguji signifikansi dengan metode-metode yang dibandingkan.

4.5.1.1 Hamming Error

Hamming loss error atau yang sering disebut *Hamming error* merupakan kesalahan klasifikasi yang dapat dinyatakan sebagai berikut:

$$E_H = \sum_{i=1}^n \sum_{l=1}^K w_{i,l} \mid \{ \text{sign}(f_l(x_i)) \neq y_{i,l} \} \quad (4.9)$$

Dalam kasus *ensemble classifier* menggunakan *Adaboost.MH* setiap *classifier* merupakan vektor yang beranggotakan k *binary classifier*. Untuk setiap sampel yang akan diprediksi, akan dicobakan ke semua *binary classifier* tersebut. Jika *binary classifier* mengklasifikasikan sampel ke kelas yang salah maka akan dihitung sebagai kesalahan (*error*). Misalnya, bila *binary classifier* l memprediksi sampel sebagai anggota kelasnya ($\text{label}_l = +1$), padahal sampel bukan anggota kelas itu ($\text{label}_l = -1$), prediksi tersebut akan dihitung sebagai *error*.

4.5.1.1 Akurasi

Pengukuran akurasi dilakukan untuk menghitung ketepatan prediksi dari metode yang digunakan. Persamaan untuk menghitung akurasi adalah sebagai berikut:

$$\text{akurasi} = \frac{TP + TN}{N} \quad (4.10)$$

dimana TP (*true positive*) menyatakan jumlah data benar yang dikenali oleh sistem, TF (*true negative*) menyatakan jumlah data salah yang dikenali sistem, dan N menyatakan total data yang diuji. Dengan kata lain, akurasi dapat didefinisikan sebagai perbandingan antara jumlah data yang dapat dikenali secara benar oleh sistem dibandingkan dengan total data secara keseluruhan.

4.5.1.2 Koefisien Kappa

Pada penelitian ini, perhitungan *Kappa Statistics* digunakan untuk menghitung persetujuan antara setiap pengamat terhadap suatu kejadian. Koefisien *Kappa* dapat dihitung melalui persamaan berikut:

$$Kappa = \frac{P(a) - P(e)}{1 - P(e)} \quad (4.11)$$

Nilai Kappa yang diperoleh berdasarkan persamaan di atas menyatakan proporsi kebersetujuan sempurna antar pengamat bila memiliki nilai 1 dan menyatakan kebersetujuan yang hanya disebabkan oleh suatu kejadian acak saja bila bernilai 0, dan nilai negatif bila nilai kebersetujuan yang diperoleh lebih kecil dari pada kebersetujuan yang diperoleh melalui kejadian acak. Berikut adalah intepretasi dari nilai kappa [22].

Tabel 4-2 Intepretasi Nilai Kappa

Nilai Kappa	Agreement
< 0	Less than chance agreement
0.01-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-0.99	Almost perfect agreement

4.5.1.3 Perbandingan Metode Dengan *Pair-wise Comparison*

Untuk melakukan perbandingan metode yang diajukan dengan metode-metode lain, pada penelitian ini digunakan uji *Pair-wise Comparison*. Metode ini digunakan dalam penelitian sebelumnya [24] Prosedur untuk melakukan uji statistik tersebut dijelaskan pada uraian berikut.

1. Dilakukan perbandingan metode secara *head to head* pada sepasang metode yang dibandingkan. Suatu metode akan dibandingkan dengan

metode lainnya satu persatu. Selanjutnya akan dibentuk matriks seperti tabel berikut :

Tabel 4-3 Contoh Matriks Perbandingan *Pair-Wise Comparison*

	A	B	C	D	E
A					
B					
C					
D					
E					

Tabel di atas menunjukkan contoh matriks perbandingan 5 metode (A sampai E). *Cell* (i,j) menunjukkan nilai berapa kali metode ke-i menang terhadap metode ke-j.

2. Dari matriks yang diperoleh dari langkah 1, selanjutnya akan dievaluasi masing-masing pasangan metode yang dibandingkan. Misalnya saja pada proses evaluasi metode ke-1 dengan metode ke-3. Jika *cell* (1,3) lebih besar dari pada *cell* (3,1) maka dapat dikatakan bahwa metode ke-1 menang (unggul) terhadap metode ke-3, begitu juga sebaliknya. Proses ini akan dilakukan pada seluruh pasangan metode yang dibandingkan. Selanjutnya, akan disimpulkan bahwa untuk masing-masing metode menang terhadap metode yang mana saja, dan kalah terhadap metode yang mana saja.

4.5.2 Metode Evaluasi Aproksimasi Organ Janin

Untuk mengetahui performa dari aproksimasi bentuk organ janin, digunakan perhitungan *hit rate* dari aproksimasi yang diaplikasikan kepada seluruh organ janin. Nilai *hit rate* dihitung dengan persamaan berikut :

$$Hit\ rate = \frac{jumlah\ hit}{N} \quad (4.15)$$

di mana N adalah jumlah keseluruhan data. Untuk menilai apakah suatu aproksimasi mencapai *hit* atau tidak dilakukan pengamatan langsung oleh penulis. Secara garis besar, *hit* dipenuhi c berimpit dengan kurva organ janin.

4.6 Lingkungan Implementasi Program

Dalam penelitian ini akan dibuat program untuk mengimplementasikan keseluruhan metodologi yang digunakan. Dalam pelaksanaan implementasi digunakan media *hardware* berupa computer dan juga *software* pendukung. Berikut ini adalah tabel media komputasi serta aplikasi pendukung yang digunakan.

Tabel 4-4 Lingkungan Implementasi Program

<i>Hardware</i>	Laptop Compaq Presario 510 dengan spesifikasi <ul style="list-style-type: none"> - CPU : Intel Core 2 Duo T 5870 2 GHz - RAM : 2 GB
<i>Software</i>	<ul style="list-style-type: none"> a. Sistem Operasi Windows 7 32 Bit b. Bahasa Pemrograman C++ c. Codeblocks IDE 10.05 d. Library Pengolahan Citra OpenCV 2.0 e. Library Multiboost (dibuat oleh peneliti sebelumnya [20])

Bab 5 Hasil Percobaan Dan Evaluasi Unjuk Kerja

Pada bagian ini akan dibahas hasil percobaan yang telah dilakukan dan penyajian data hasil evaluasi, baik berupa tabel dan grafik. Selain itu dalam bab ini juga akan disajikan evaluasi unjuk kerja algoritma yang diajukan.

5.1 Skenario Percobaan

5.1.1 Skenario Percobaan 1

Percobaan pertama menggunakan data citra ultrasonografi (USG). Deskripsi mengenai data USG telah dijelaskan pada bab sebelumnya. Pada percobaan ini, akan dicobakan variasi jumlah fitur atau jumlah *classifier* yang di-*ensemble*. Variasi jumlah fitur yang diujicobakan adalah pada rentang 1 sampai 200 fitur. Pada percobaan ini akan dihitung nilai *Hamming error*, akurasi dan nilai koefisien *kappa* dari metode klasifikasi yang diuji. Selain itu, pada percobaan ini akan dilakukan perbandingan performa antara metode yang diajukan dengan metode yang sudah ada. Adapun metode yang akan dijadikan pembanding dalam penelitian ini dapat dilihat pada tabel 5.1.

Tabel 5-1 Algoritma yang akan dibandingkan pada percobaan 1

No	Metode
1	Adaboost.MH-Multi Boundary Classifier (yang diusulkan)
2	Adaboost.MH-Multi Stump
3	Adaboost.MH-Single Stump
4	Adaboost.MH-Product of Multi Stump
5	Adaboost.MH-Tree
6	Adaboost.MH-LVQ
7	Adaboost.MH-GLVQ

5.1.2 Skenario Percobaan 2

Percobaan kedua dilakukan dengan tujuan melakukan verifikasi hasil percobaan pertama. Pada percobaan kedua akan digunakan data *benchmark*. Pada skenario II juga akan digunakan tiga kriteria evaluasi pada algoritma yang diujicobakan, yaitu *Hamming error*, akurasi, dan juga koefisien *kappa*. Seperti halnya dengan skenario sebelumnya bahwa pada skenario ini akan dilakukan uji coba dengan berbagai variasi jumlah fitur, atau jumlah *classifier* yang di-*ensemble*. Seperti yang disebutkan pada bab 4 bahwa dalam penelitian ini digunakan dua macam data *benchmark*, yaitu USPS dan MNIST, maka skenario ini akan dipisah menjadi dua sub skenario sebagai berikut:

1. Skenario 2.1 : Uji coba dengan menggunakan dataset USPS
2. Skenario 2.2 : Uji coba dengan menggunakan dataset MNIST

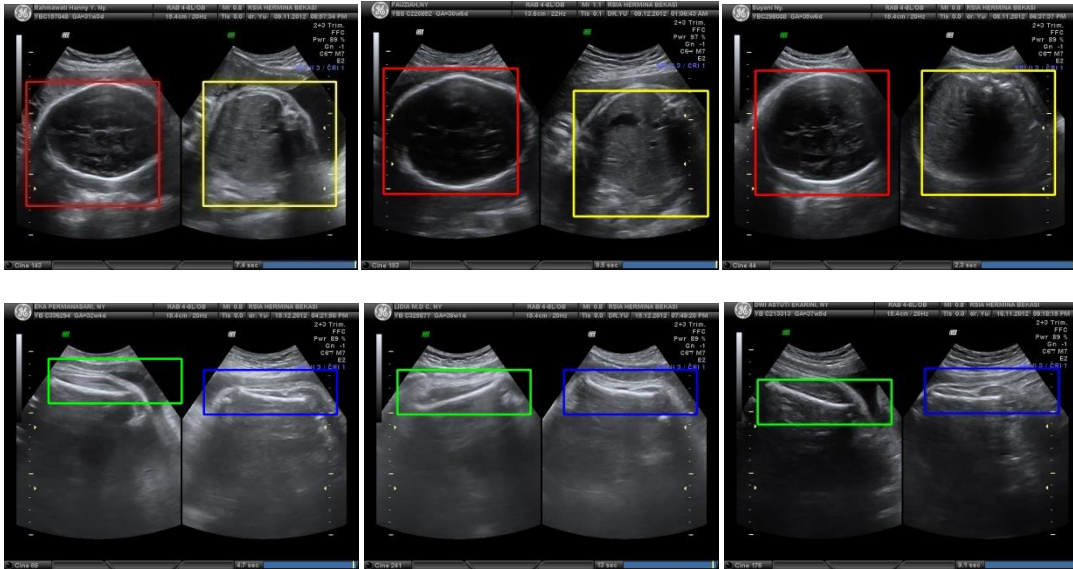
5.1.3 Skenario Percobaan 3

Percobaan ketiga dilakukan untuk mengetahui tingkat *hit* pada proses aproksimasi bentuk organ janin, setelah dilakukan proses deteksi terlebih dahulu. Skenario 3 ini dilakukan dari hasil deteksi objek yang dihasilkan dengan menggunakan algoritma *Adaboost.MH* berbasis algoritma *Multi Boundary Classifier*. Seperti yang disebutkan pada bab metodologi penelitian, bahwa proses aproksimasi dilakukan dengan menggunakan metode *Randomized Hough Transform* (RHT).

5.2 Hasil Percobaan 1

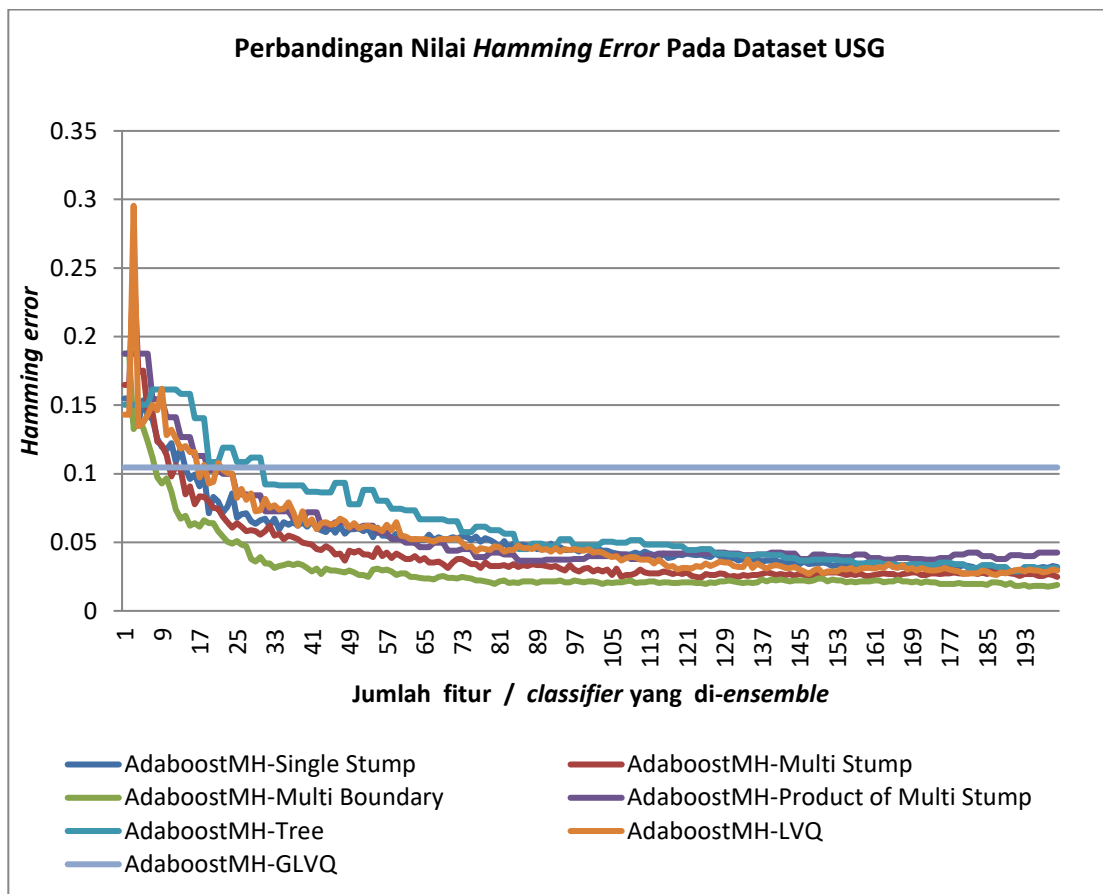
Sesuai dengan deskripsi skenario yang dijelaskan pada subbab sebelumnya, bahwa percobaan pertama dilakukan pada dataset USG. Karena jumlah dataset USG yang digunakan terbatas, maka dilakukan *k-fold cross validation*, dengan $k=5$. Percobaan ini bertujuan untuk mengukur tingkat akurasi klasifikasi, dalam hal ini digunakan untuk melakukan deteksi objek pada citra USG. Dalam melakukan proses deteksi objek sistem akan melakukan *scanning* data USG secara keseluruhan. Dalam proses tersebut, setiap *window* akan diklasifikasikan ke dalam kelas objek organ janin atau *background*. Organ janin yang dapat dideteksi oleh sistem ada empat macam,

yaitu kepala, perut, tulang paha, dan tulang lengan. Dengan demikian, sistem melakukan klasifikasi *multi classs* sekaligus melakukan deteksi multi objek pada citra. Contoh hasil deteksi organ janin dapat dilihat pada gambar 5.1.



Gambar 5-1 Hasil Deteksi Organ Janin

Setelah dilakukan percobaan, diperoleh data *Hamming error*, akurasi, dan koefisien *kappa* dari masing-masing algoritma yang diujicobakan. Tabel lengkap nilai Pada bab ini akan ditampilkan kurva hasil perhitungan kriteria-kriteria tersebut. Kurva *Hamming error* dari algoritma-algoritma klasifikasi yang diujicobakan dapat dilihat pada gambar 5.2.

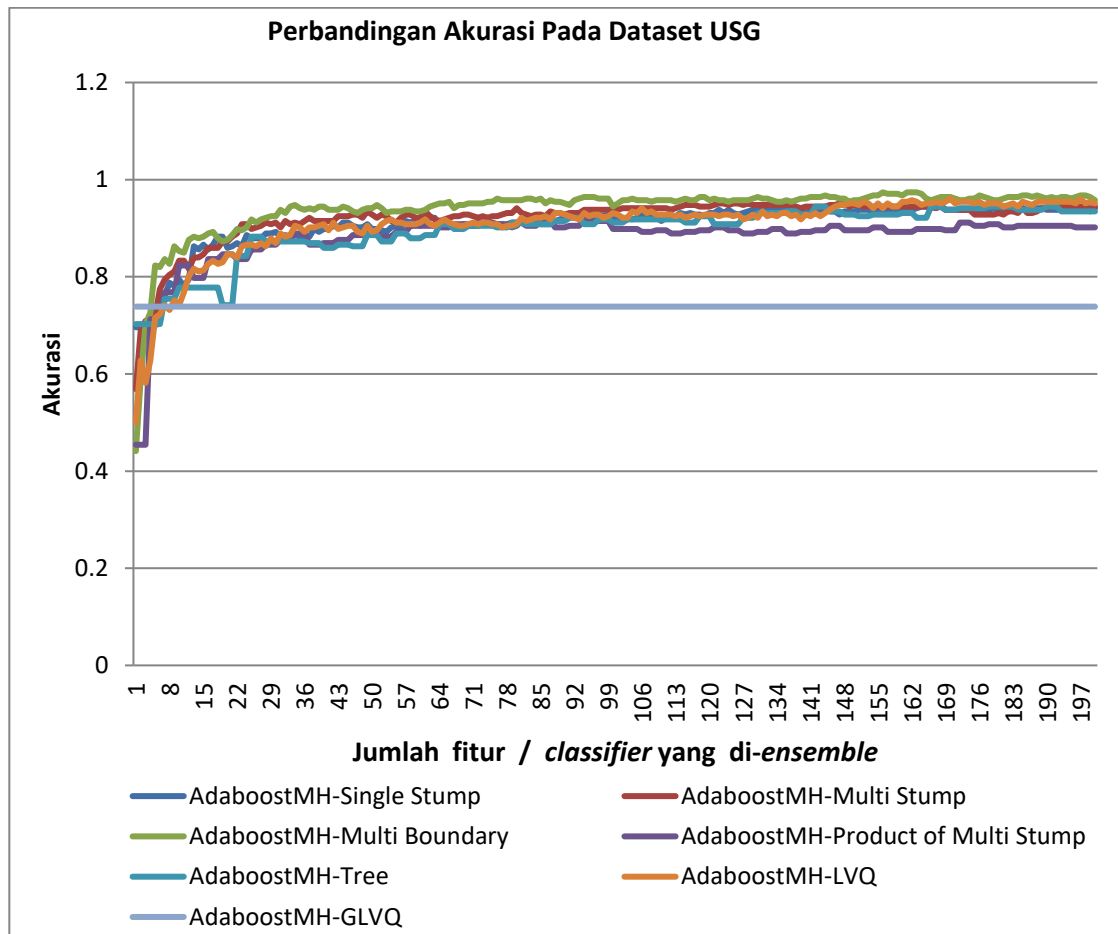


Gambar 5-2 Perbandingan Nilai Hamming Error pada dataset USG

Pada gambar 5.2 dapat kita lihat bahwa nilai *Hamming error* dari algoritma yang diajukan (*Adaboost.MH-Multi Boundary Classifier*) memiliki nilai yang paling kecil dari metode lainnya pada hampir seluruh nilai jumlah fitur (jumlah *classifier*) yang di-*ensemble*. Metode yang diajukan memiliki nilai *Hamming error* yang lebih tinggi dari metode lainnya dalam hal ini *Adaboost.MH-GLVQ* dengan jumlah fitur pada rentang 1-7 fitur. Di luar jumlah fitur pada rentang tersebut, metode yang diusulkan memiliki nilai *Hamming error* yang paling kecil di antara 6 metode lainnya. Dengan demikian, pada kasus data USG, metode yang diusulkan lebih baik dari metode lainnya dari sisi (faktor) *Hamming error*.

Selain mendapatkan data nilai akurasi dari metode-metode klasifikasi yang diuji, dalam percobaan ini dihitung juga nilai akurasi dari metode-metode klasifikasi yang diuji. Tabel lengkap akurasi dari seluruh metode klasifikasi yang digunakan

dalam penelitian ini dapat dilihat pada bagai lampiran. Pada gambar 5.3 ditampilkan plot tingkat akurasi dari metode-metode tersebut untuk variasi jumlah fitur (*classifier*) yang di-*ensemble*.

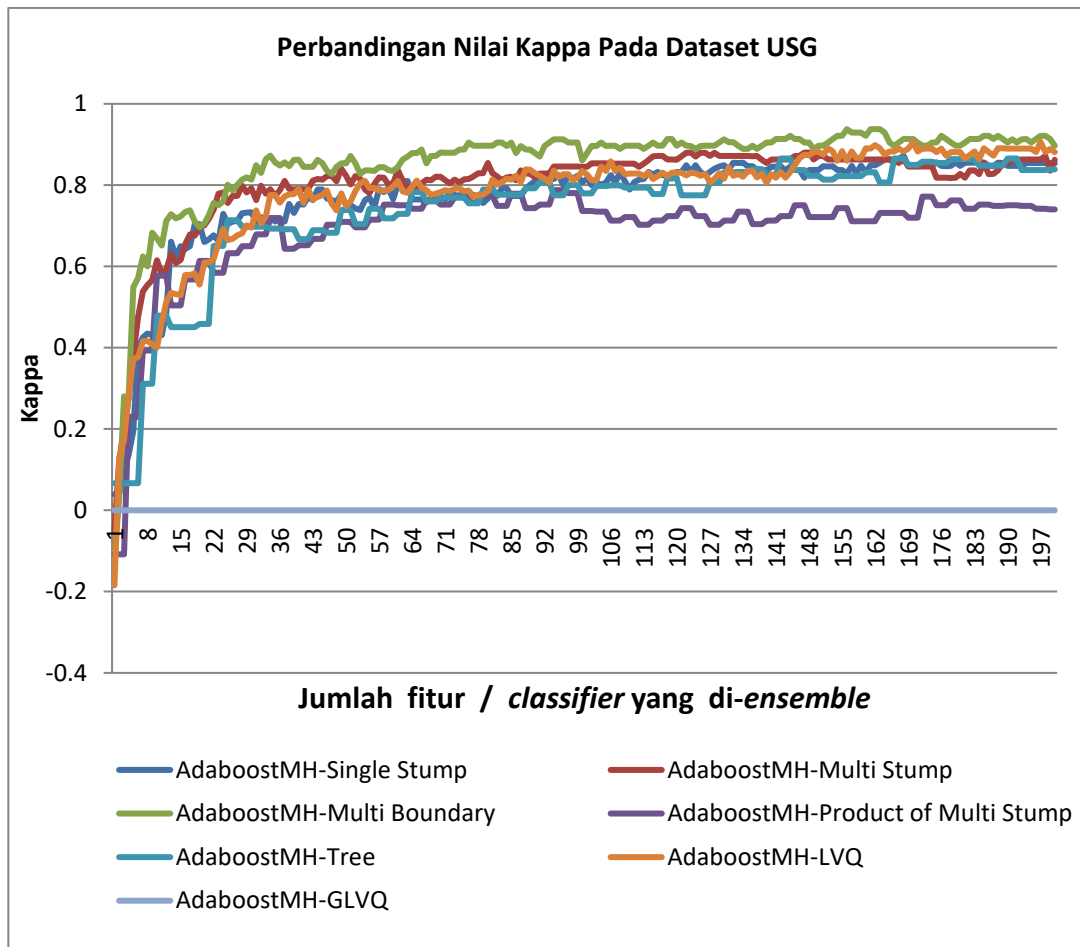


Gambar 5-3 Perbandingan Nilai Akurasi pada dataset USG

Dari gambar 5.3 dapat dilihat bahwa nilai akurasi dari metode yang diusulkan hampir selalu lebih baik dari algoritma-algoritma klasifikasi lain yang diujicobakan jika dilihat dari faktor akurasi. Meski pada saat jumlah fitur kurang dari 9 algoritma yang diusulkan terkadang memiliki akurasi yang lebih rendah dari algoritma lain, akan tetapi pada saat jumlah fitur/jumlah *classifier* yang di-*ensemble* lebih dari 9, algoritma yang diusulkan selalu memiliki nilai akurasi yang lebih tinggi daripada algoritma lainnya. Dengan demikian, dapat dikatakan bahwa pada dataset USG,

algoritma yang diusulkan lebih baik dari enam algoritma lainnya, dilihat dari faktor akurasi.

Selanjutnya, dihitung nilai *kappa* dari masing-masing algoritma yang diujicobakan, dengan variasi jumlah fitur dari 1 sampai 200. Kurva nilai *kappa* untuk ketujuh algoritma yang dibandingkan dapat dilihat pada gambar 5.4. Sedangkan tabel lengkap, perhitungan koefisien *kappa* untuk dataset USG dapat dilihat pada bagian lampiran.



Gambar 5-4 Perbandingan Nilai Kappa pada dataset USG

Dari gambar 5.4 dapat dilihat bahwa nilai *kappa* dari metode yang diusulkan hampir selalu lebih baik dari algoritma-algoritma klasifikasi lain yang diujicobakan. Pada saat nilai fitur cukup kecil, yaitu kurang dari 6, algoritma yang diusulkan memiliki nilai *kappa* yang lebih rendah daripada algoritma lain, dalam hal ini

Adaboost.MH-Multi Stump dan *Adaboost.MH-LVQ*. Akan tetapi, setelah nilai fitur lebih dari 6, maka algoritma yang diusulkan memiliki nilai *kappa* yang lebih tinggi daripada nilai *kappa* algoritma lain yang diujicobakan. Dengan demikian dapat dikatakan bahwa dalam kasus dataset USG, algoritma yang diusulkan lebih baik dari keenam algoritma lain yang diujicobakan jika dilihat dari sisi koefisien *kappa*.

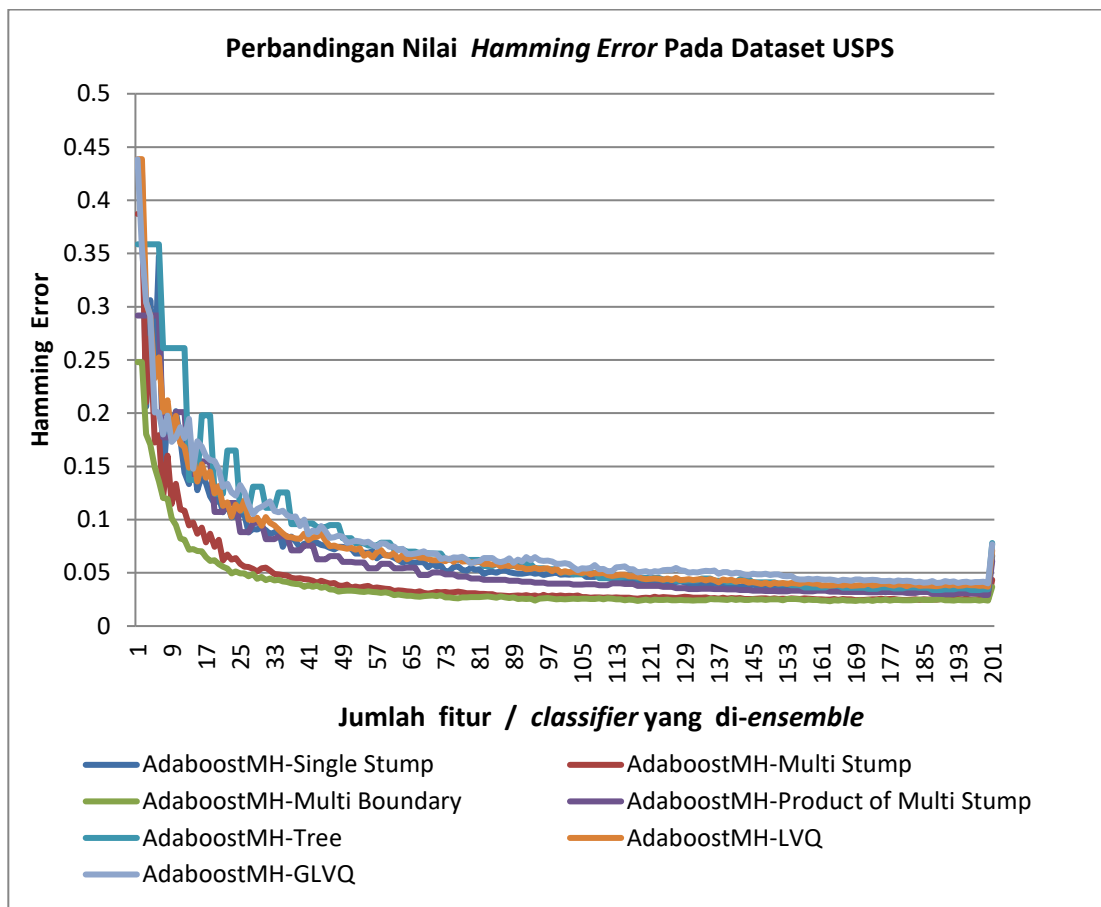
5.3 Hasil Percobaan 2

Sesuai dengan deskripsi skenario yang dijelaskan pada subbab sebelumnya, bahwa percobaan kedua dilakukan untuk melakukan verifikasi algoritma yang diusulkan. Percobaan kedua dilakukan dengan menggunakan dataset *benchmark* USPS dan MNIST. Pada subbab selanjutnya akan dijelaskan hasil percobaan pada kedua dataset *benchmark* tersebut.

5.3.1 Hasil Percobaan Dataset USPS

Deskripsi mengenai data USPS telah dijelaskan pada bab sebelumnya. Seperti halnya percobaan pada dataset USG, pada percobaan ini, akan dicobakan variasi jumlah fitur atau jumlah *classifier* yang di-*ensemble*. Variasi jumlah fitur yang diuji cobakan adalah pada rentang 1 sampai 200 fitur. Pada percobaan ini akan dihitung nilai *Hamming error*, akurasi, dan nilai koefisien *kappa* dari metode klasifikasi yang diuji.

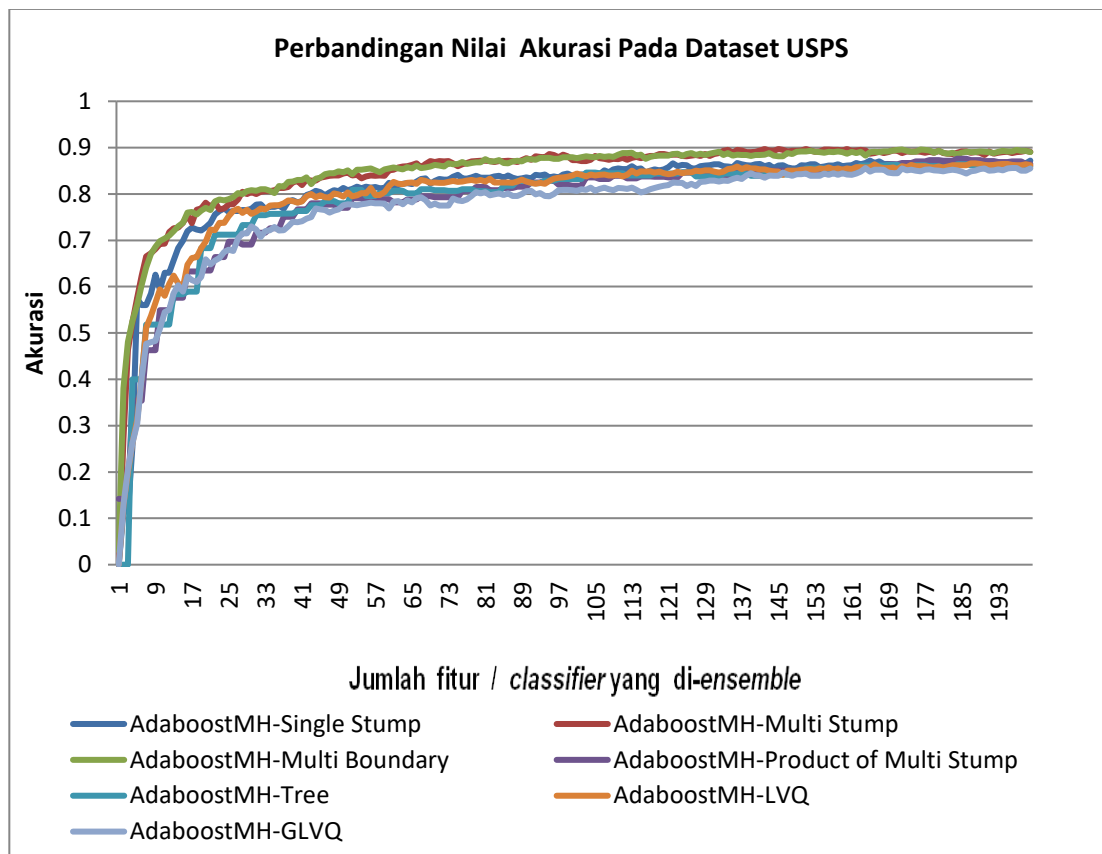
Setelah dilakukan eksperimen, diperoleh data *Hamming error* dari masing-masing algoritma klasifikasi untuk berbagai variasi jumlah fitur (jumlah *classifier* yang di-*ensemble*). Pada gambar 5.5 dipaparkan kurva nilai *Hamming error* dari masing-masing metode klasifikasi yang diujicobakan.



Gambar 5-5 Perbandingan Hamming Error pada dataset USPS

Berdasarkan gambar 5.5 dapat dilihat bahwa nilai *Hamming error* dari algoritma yang diusulkan hampir selalu paling kecil di antara nilai *Hamming error* dari keseluruhan algoritma klasifikasi yang diujicobakan. Mulai dari jumlah fitur 1 sampai dengan 184, algoritma yang diusulkan memiliki nilai *Hamming error* paling kecil di antara keenam algoritma yang dibandingkan. Baru setelah nilai fitur lebih dari 184, nilai *Hamming error* dari algoritma yang diajukan beberapa kali lebih tinggi dari algoritma lain, yang dalam hal ini adalah *Adaboost.MH-Multi Stump*. Meskipun demikian, dapat dikatakan bahwa dalam kasus dataset USPS, algoritma yang diusulkan lebih baik dari algoritma lain yang dibandingkan dilihat dari faktor *Hamming error*.

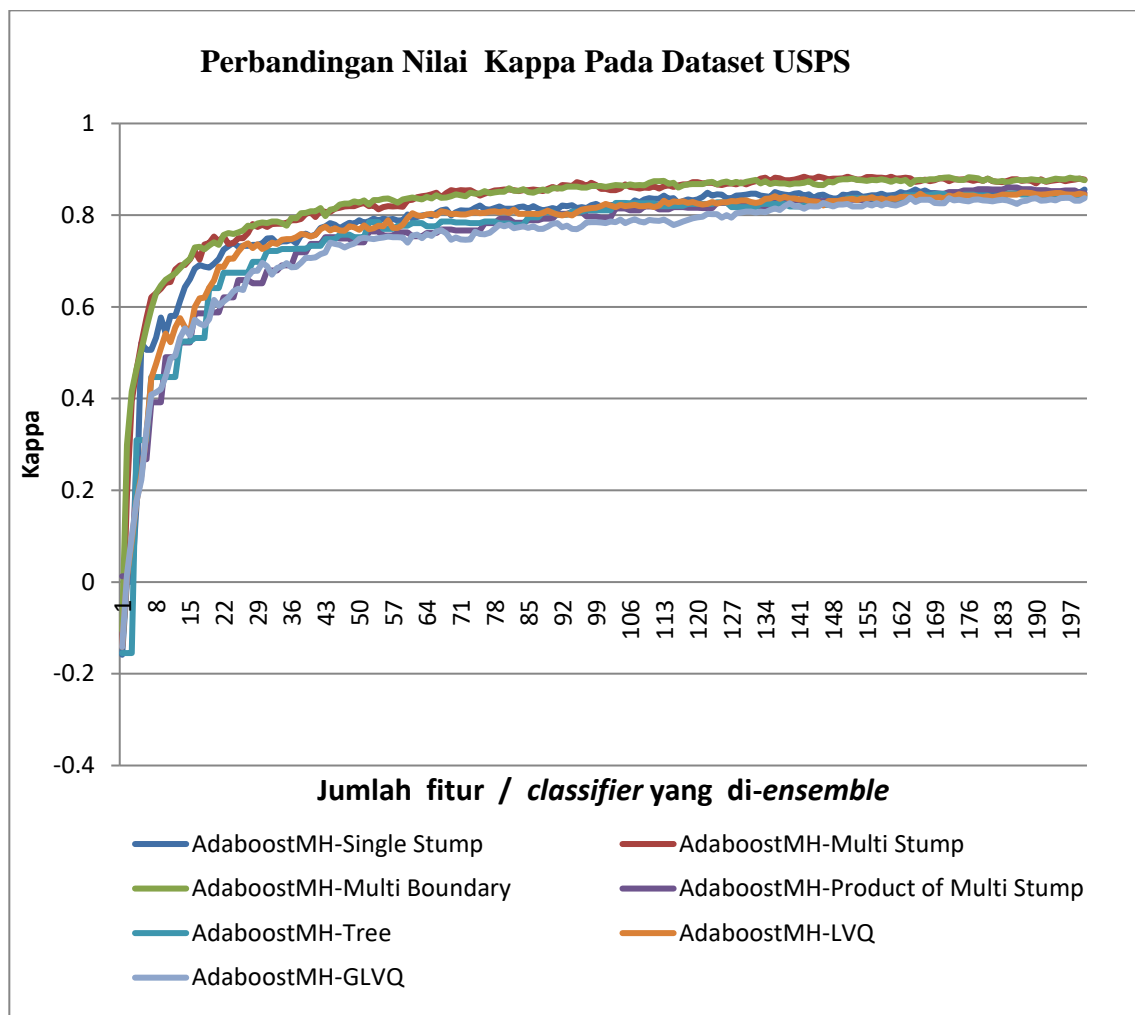
Berdasarkan gambar 5.6 dapat dilihat perbandingan nilai akurasi algoritma klasifikasi yang diujicobakan pada dataset USPS. Tabel lengkap perhitungan akurasi keseluruhan algoritma yang diujicobakan untuk dataset USPS tersedia pada lampiran. Dari gambar 5.6 dapat dilihat bahwa kurva tingkat akurasi algoritma yang diusulkan (*Adaboost-Multi Boundary Classifier*) sering berimpit dengan metode lain, yang dalam hal ini *Adaboost-Multi Stump*. Oleh sebab itu, dari grafik tersebut, belum bisa dikonfirmasi bahwa algoritma yang diusulkan memiliki performa yang lebih bagus dari algoritma lainnya, dilihat dari sisi tingkat akurasi. Oleh sebab itu diperlukan evaluasi lebih lanjut yang akan dijelaskan pada subbab selanjutnya.



Gambar 5-6 Perbandingan Akurasi pada dataset USPS

Seperti halnya pada percobaan 1, pada percobaan ini juga dihitung nilai *kappa* dari hasil klasifikasi. Tabel lengkap perhitungan koefisien *kappa* untuk data USPS dapat dilihat pada bagian lampiran. Untuk kurva nilai *kappa* dari setiap algoritma

klasifikasi yang diujicobakan dapat dilihat pada gambar 5.7. Jika dilihat dari kurva pada gambar 5.7, nilai $kappa$ dari algoritma yang diusulkan sering mengalami *overlapping* dengan algoritma lain, dalam hal ini adalah *Adaboost.MH-Multi Stump*. Nilai koefisien $kappa$ pada data USPS ini mirip halnya dengan nilai akurasi yang sering *overlapping* dengan algoritma lain. Oleh sebab itu akan dilakukan evaluasi lebih lanjut pada subbab selanjutnya.



Gambar 5-7 Perbandingan Nilai Kappa pada dataset USPS

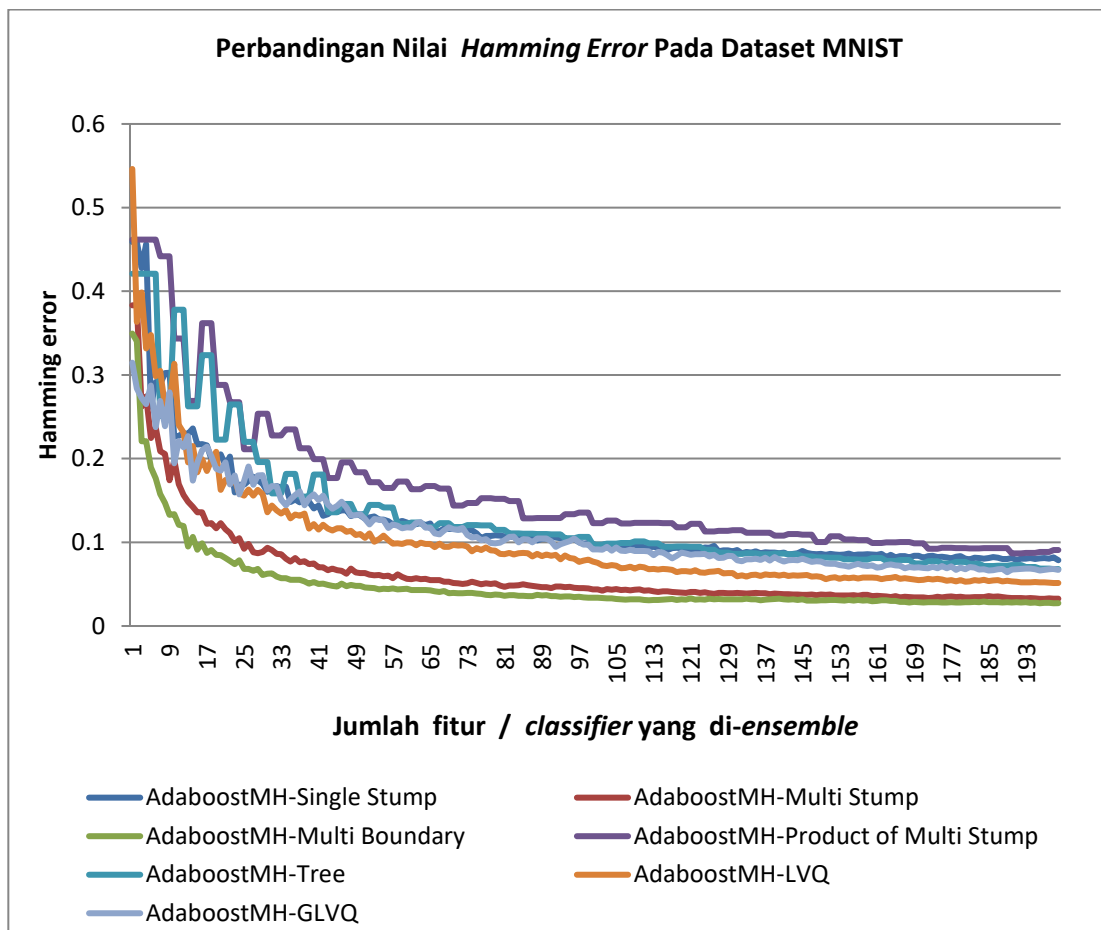
5.3.2 Hasil Percobaan Dataset MNIST

Deskripsi mengenai data MNIST telah dijelaskan pada bab sebelumnya. Seperti halnya percobaan pada dataset USG dan USPS, pada percobaan ini, akan

dicobakan variasi jumlah fitur atau jumlah *classifier* yang di-*ensemble*. Variasi jumlah fitur yang diujicobakan adalah pada rentang 1 sampai 200 fitur. Pada percobaan ini akan dihitung nilai *Hamming error*, akurasi dan nilai koefisien *kappa* dari metode klasifikasi yang diuji.

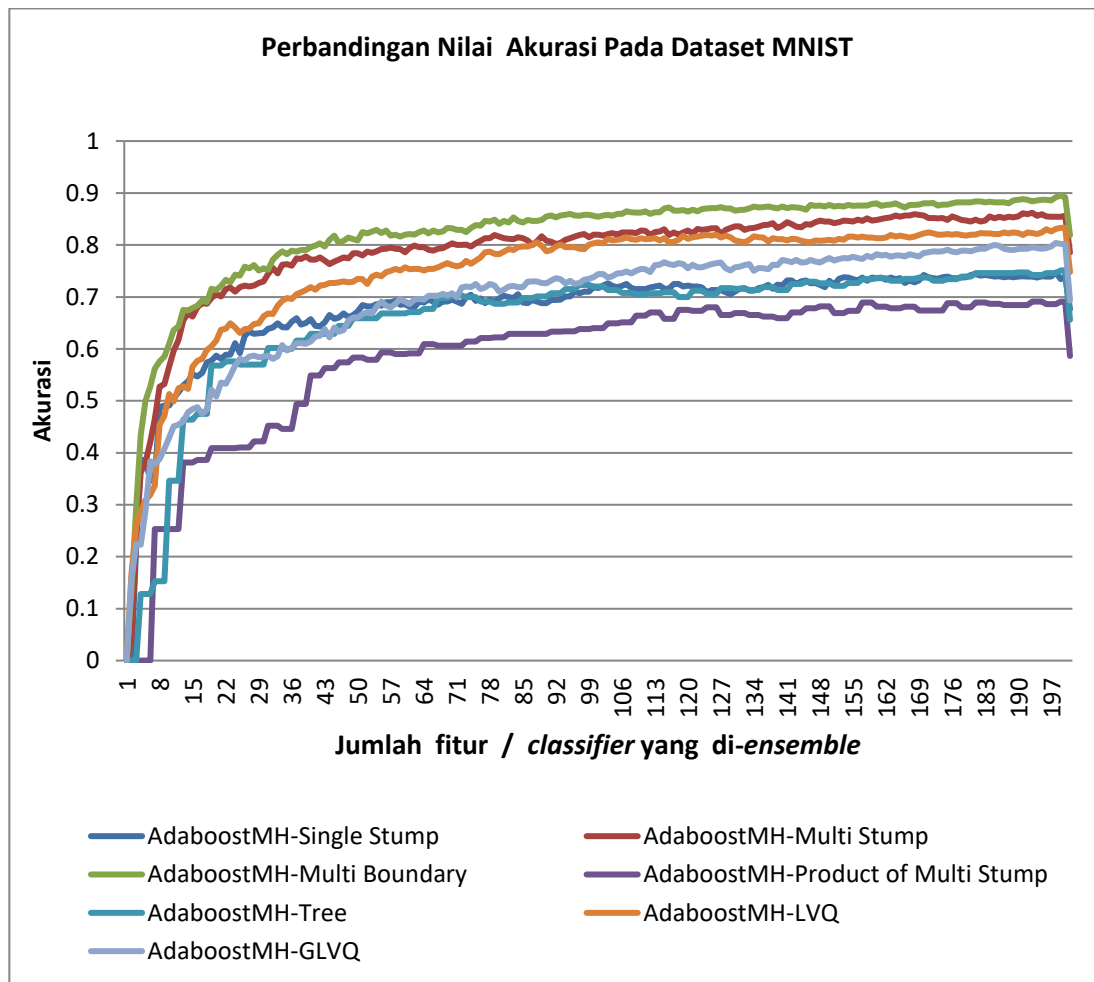
Setelah dilakukan eksperimen, diperoleh data *Hamming error* dari masing-masing algoritma klasifikasi untuk berbagai variasi jumlah fitur (jumlah *classifier* yang di-*ensemble*). Pada gambar 5.8 dipaparkan kurva nilai *Hamming error* dari masing-masing metode klasifikasi yang diujicobakan pada data MNIST.

Pada gambar 5.8 dapat dilihat bahwa nilai *Hamming error* dari metode klasifikasi yang diajukan hampir selalu menjadi yang terbaik (terkecil) dalam semua variasi jumlah fitur yang digunakan. Algoritma yang diusulkan memiliki *Hamming error* lebih besar dari pada nilai *Hamming error* metode lainnya hanya saat jumlah fitur sama dengan 1 atau 2 saja. Selebihnya, algoritma yang diusulkan unggul dari metode klasifikasi lain yang diujicobakan. Dengan demikian, dapat dikatakan bahwa pada kasus data MNIST, algoritma yang diusulkan memiliki performa yang lebih baik dari metode-metode lain yang dibandingkan bila dilihat dari faktor *Hamming error*.



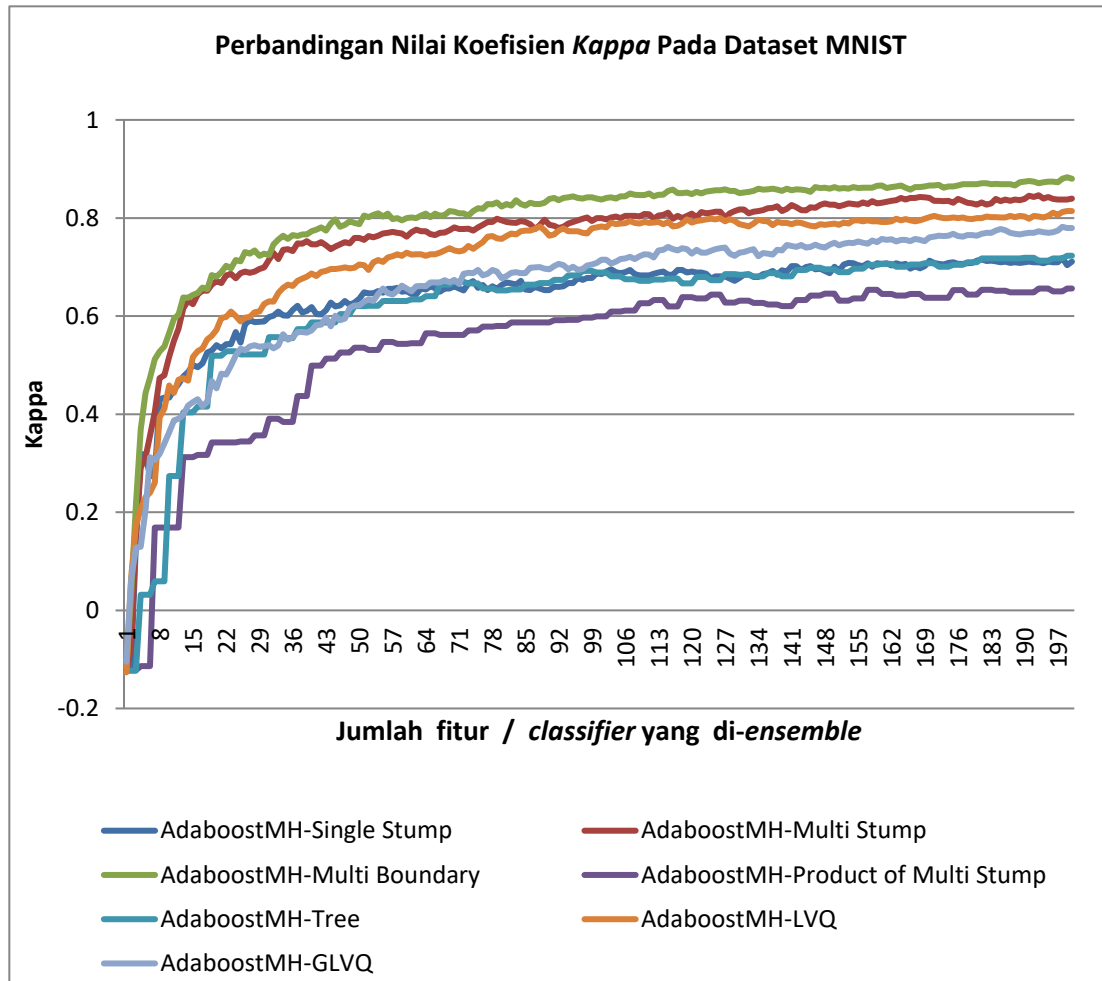
Gambar 5-8 Perbandingan Nilai *Hamming Error* pada dataset MNIST.

Selanjutnya, pada data MNIST juga dilakukan pengukuran akurasi dari metode-metode klasifikasi yang diujicobakan Pada gambar 5.9. Sedangkan tabel lengkap perhitungan akurasi dari dataset MNIST dapat dilihat pada bagian lampiran. Pada gambar 5.9 dapat dilihat bahwa nilai akurasi dari metode klasifikasi yang diajukan hampir selalu menjadi yang terbaik (terbesar) pada semua variasi jumlah fitur yang digunakan. Algoritma yang diusulkan memiliki akurasi lebih rendah daripada metode lainnya hanya saat jumlah fitur sama dengan 1 atau 2 saja. Selebihnya, algoritma yang diusulkan unggul dari metode klasifikasi lain yang diujicobakan. Dengan demikian, dapat dikatakan bahwa pada kasus data MNIST, algoritma yang diusulkan memiliki performa yang lebih baik dari metode-metode lain yang dibandingkan bila dilihat dari faktor akurasi.



Gambar 5-9 Perbandingan Nilai Akurasi pada dataset MNIST.

Selanjutnya, pada data MNIST juga dilakukan pengukuran nilai koefisien *kappa*. Koefisien *kappa* dari metode-metode klasifikasi yang diujicobakan dapat dilihat pada gambar 5.10. Untuk tabel lengkap pengukuran koefisien *kappa* dari seluruh metode klasifikasi pada data MNIST dapat dilihat pada bagian lampiran.

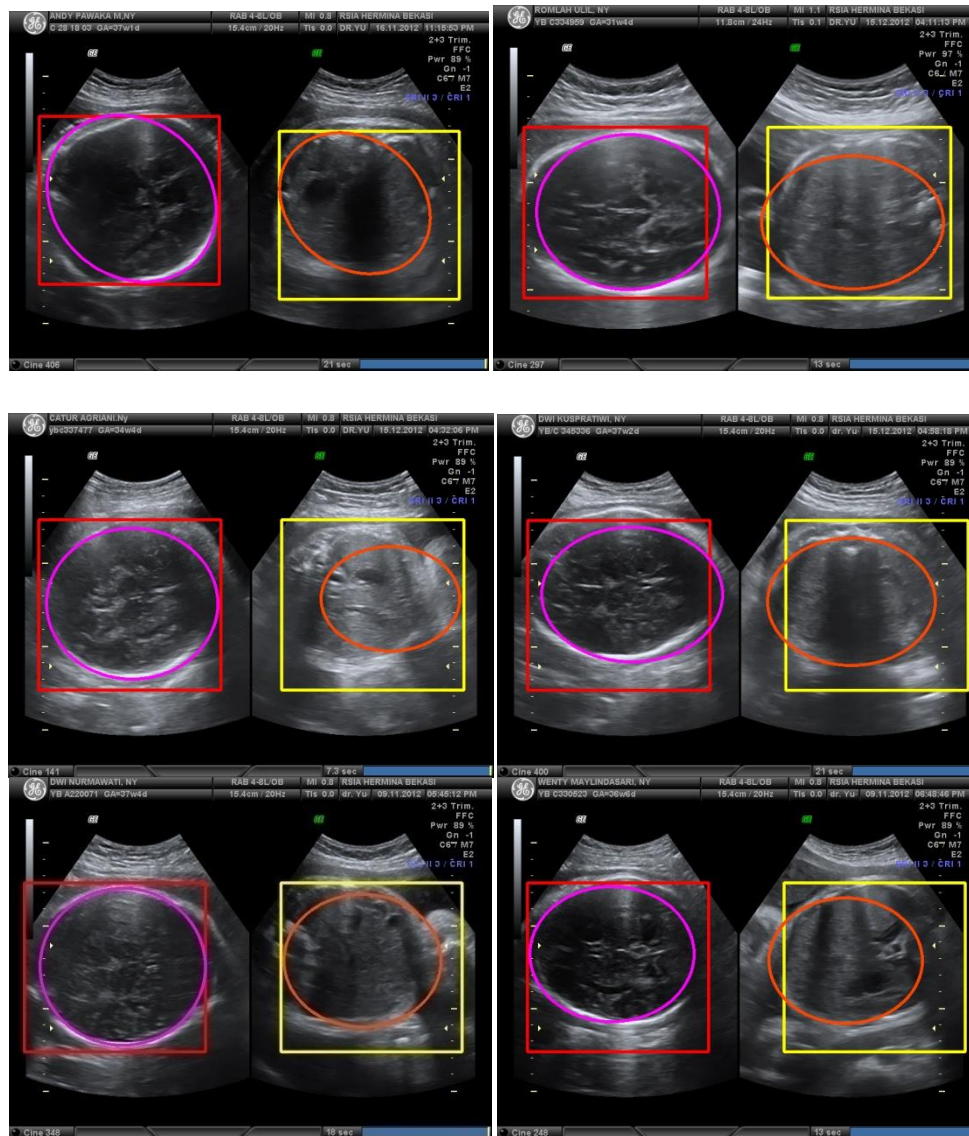


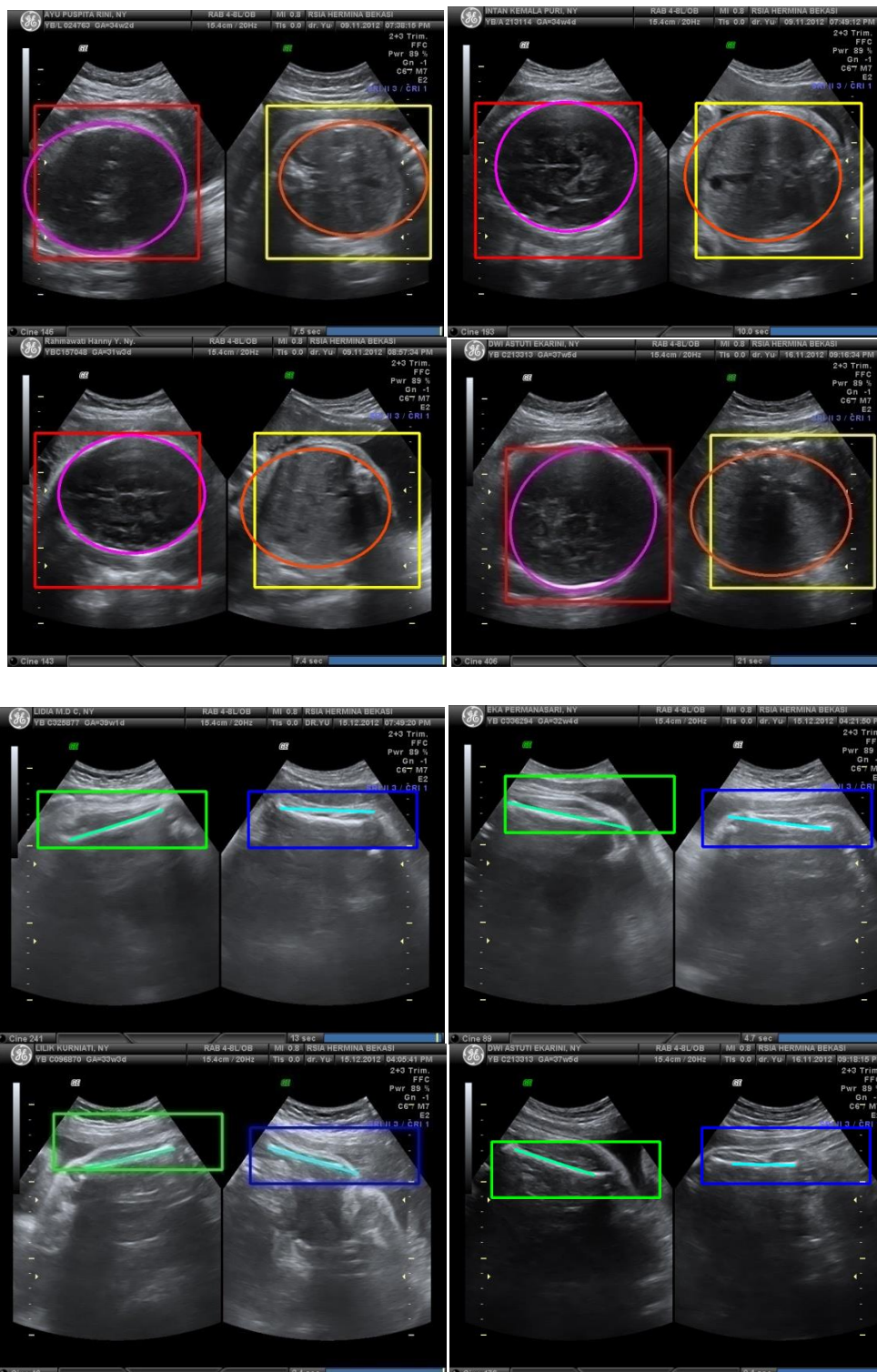
Gambar 5-10 Perbandingan nilai koefisien kappa pada dataset MNIST.

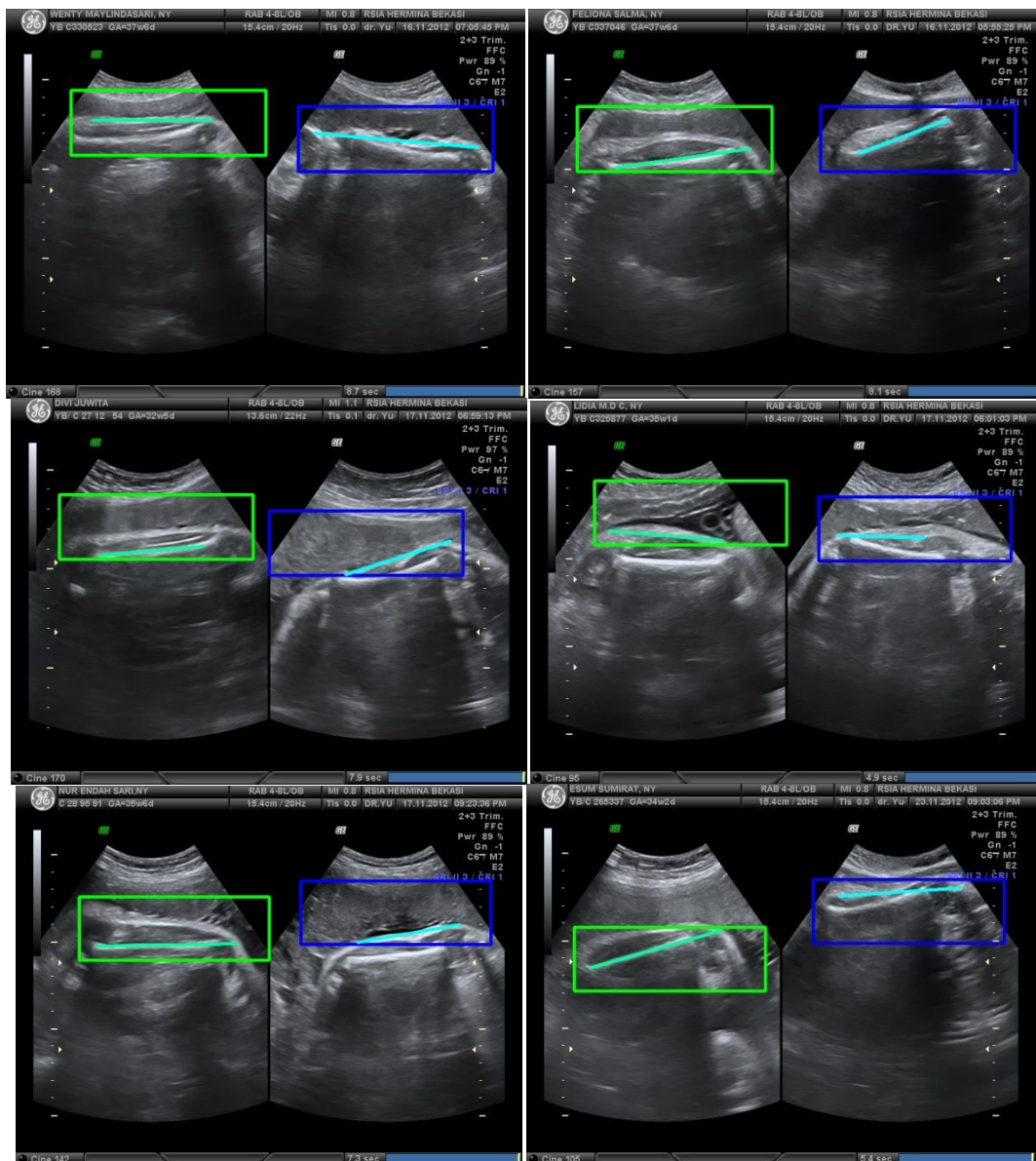
Dari gambar 5.10 dapat dilihat bahwa pada data MNIST nilai *kappa* dari metode yang diusulkan hampir selalu lebih tinggi dari nilai *kappa* metode-metode lain yang diujicobakan. Nilai *kappa* dari metode yang dihasilkan kurang dari nilai *kappa* metode lain pada saat nilai fitur yang dipilih sebanyak 1 dan 2. Pada saat fitur yang dipilih sebanyak 1 fitur, nilai *kappa* dari metode yang diusulkan kalah dari metode *Adaboost.MH-GLVQ*. Sedangkan pada saat fitur yang dipilih sebanyak 2 fitur, nilai *kappa* metode yang diusulkan kalah dengan algoritma *Adaboost.MH-LVQ*. Meskipun demikian, secara umum, metode yang diusulkan dapat dikatakan lebih baik daripada metode-metode lain yang diujicobakan dalam kasus dataset MNIST, dilihat dari faktor nilai koefisien *kappa*.

5.4 Hasil Percobaan 3

Pada penelitian ini dilakukan ujicoba aproksimasi bentuk organ janin setelah dilakukan deteksi. Dalam percobaan ini, hasil dari deteksi objek organ janin hasil dari algoritma *Adaboost.MH-Multi Boundary Classifier*. Seperti yang sudah dijelaskan pada sub bab sebelumnya bahwa proses *aproksimasi* dilakukan dengan menggunakan metode *Randomized Hough Transform* (RHT). Hasil deteksi dan aproksimasi organ janin. Dapat dilihat pada gambar 5.11.



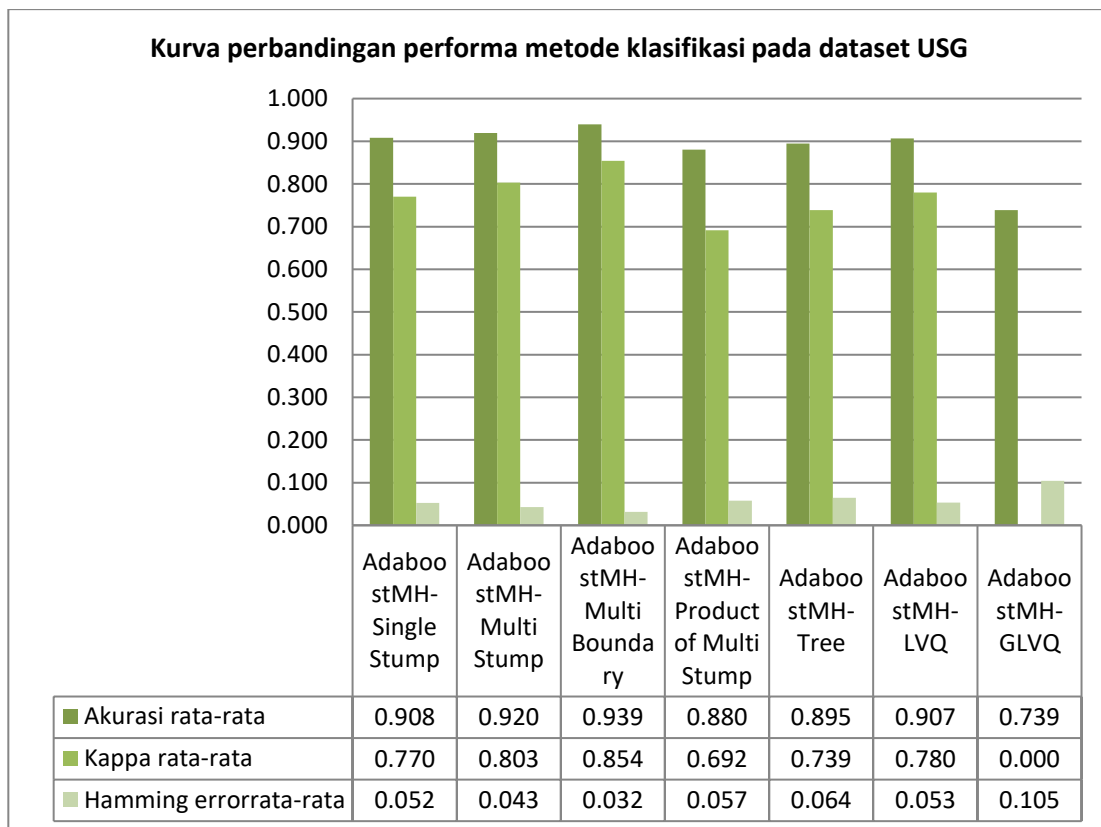




Gambar 5-11 Contoh Hasil Deteksi dan Aproksimasi Organ Janin

5.5 Evaluasi Hasil Percobaan 1

Berdasarkan data hasil percobaan 1, disebutkan bahwa pada dataset USG, algoritma yang diusulkan memiliki performa yang lebih baik daripada metode lainnya. Pada percobaan 1 dilakukan untuk berbagai nilai fitur, yaitu dari 1 sampai 200. Selanjutnya, dari 200 data uji tersebut diringkas dengan dicari rata-ratanya, baik untuk parameter *Hamming error*, akurasi maupun koefisien *kappa*. Pada gambar 5.12 ditampilkan rata-rata nilai *Hamming error*, akurasi maupun koefisien *kappa*.



Gambar 5-12 Perbandingan nilai performa metode klasifikasi pada dataset USG.

Berdasarkan grafik pada gambar 5.12 dapat dilihat bahwa menurut Faktor *Hamming error* metode yang diajukan dapat dikatakan sebagai metode yang paling baik, karena nilai rata-rata *Hamming error* nya paling kecil. Selanjutnya, apabila dilihat dari faktor akurasi, metode *Adaboost.MH-Multi Boundary Classifier* memiliki tingkat akurasi yang paling tinggi, diantara metode-metode yang dibandingkan.

Selanjutnya, jika dilihat dari sisi rata-rata koefisien *kappa*, metode yang diajukan memiliki nilai rata-rata yang paling tinggi di antara metode-metode yang dibandingkan tersebut. Dengan demikian dapat dikatakan bahwa untuk dataset USG metode yang diusulkan lebih unggul dari enam metode lain yang digunakan sebagai pembandingan. Selanjutnya akan dilakukan perbandingan metode yang diujicobakan dengan menggunakan *Pair-Wise Comparison* atau juga sering disebut dengan perbandingan secara *Head to Head*.

5.5.1 Perbandingan Performa Metode pada Dataset USG Dilihat dari Faktor *Hamming Error*

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset USG, dilihat dari faktor *Hamming error*. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-2 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor *Hamming Error*

Metode	A	B	C	D	E	F	G
A		10	3	143	178	71	187
B	190		2	198	195	187	189
C	197	198		198	198	197	193
D	57	1	2		120	74	182
E	22	5	2	80		7	170
F	129	13	3	126	193		182
G	13	11	7	18	30	18	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-2 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-3.

Tabel 5-3 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor *HammingError*

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D,E,G	B,C,F
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	E,G	A,B,C,F
E	G	A,B,C,D,F
F	A,D,E,G	B,C
G	-	A,B,C,D,E,F

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-3 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset USG dilihat dari faktor *hamming error*.

5.5.2 Perbandingan Performa Metode pada Dataset USG Dilihat dari Faktor Akurasi

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset USG, dilihat dari factor akurasi. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-4 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor Akurasi

Metode	A	B	C	D	E	F	G
A		31	4	177	155	85	195
B	169		5	198	168	142	195
C	196	195		199	197	198	196
D	23	2	1		63	19	194
E	45	32	3	137		31	194
F	115	58	2	181	169		193
G	5	5	4	6	6	7	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-4 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-5.

Tabel 5-5 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor Akurasi

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D,E,G	B,C,F
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	G	A,B,C,F,E
E	D, G	A,B,C,F
F	A,D,E,G	B,C
G	-	A,B,C,D,E,F

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-5 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset USG dilihat dari faktor akurasi.

5.5.3. Perbandingan Performa Metode pada Dataset USG Dilihat dari Faktor *Kappa*

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset USG, dilihat dari faktor *kappa*. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-6 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor *Kappa*

Metode	A	B	C	D	E	F	G
A		32	3	186	150	68	200
B	168		5	200	170	138	199
C	197	195		199	198	198	199
D	14	0	1		45	7	197
E	50	30	2	155		23	200
F	132	62	2	193	177		199
G	0	1	1	3	0	1	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-6 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-7.

Tabel 5-7 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset USG Berdasarkan Faktor *Kappa*

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D,E,G	B,C,F
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	G	A,B,C,F,E
E	D, G	A,B,C,F
F	A,D,E,G	B,C
G	-	A,B,C,D,E,F

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

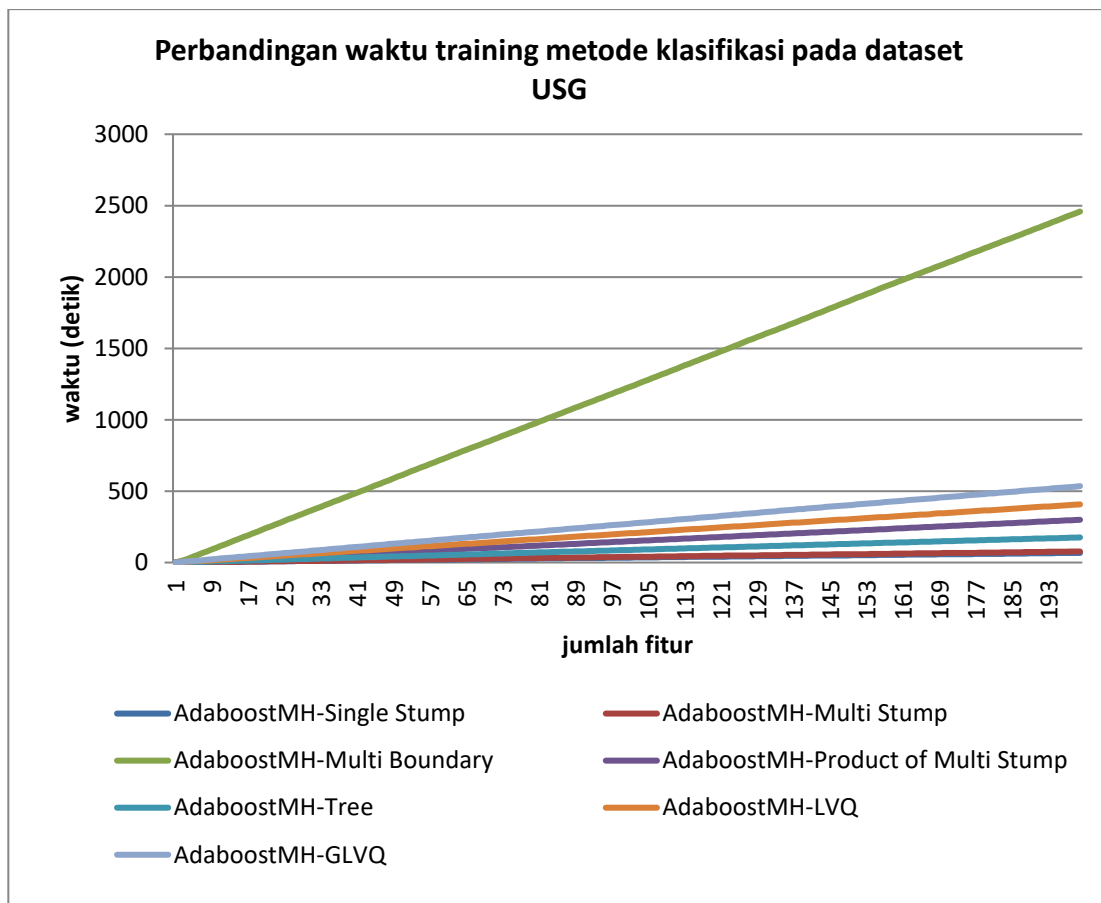
F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

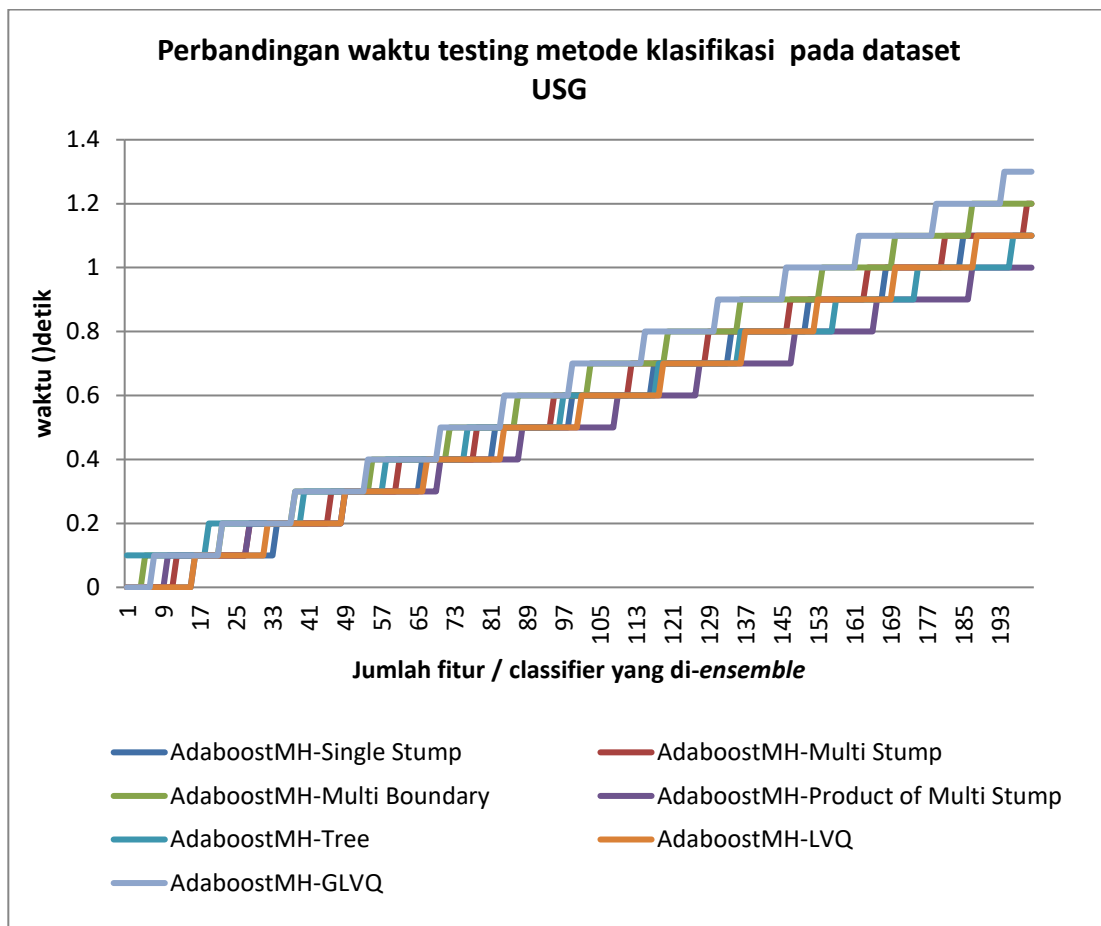
Berdasarkan tabel 5-7 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset USG dilihat dari faktor *kappa*.

5.5.4. Perbandingan Waktu Komputasi Metode Klasifikasi pada Dataset USG

Berdasarkan gambar 5-13 dapat dilihat perbandingan waktu *training* (5-13 (a)) dan *testing* (5-13 (b)) dari ketujuh metode klasifikasi yang diuji cobakan pada dataset USG.. Kedua kurva tersebut diperoleh dengan uji coba pada 1500 sampel yang di-generate dari dataset USG. Pada gambar 5-13 (a) dapat dilihat bahwa metode yang diajukan membutuhkan waktu *training* yang sangat jauh lebih lama dibandingkan dengan metode yang lain. Hal ini dikarenakan metode yang diusulkan memiliki kompleksitas $O(N^2)$, sedangkan metode lain yang dibandingkan memiliki kompleksitas $O(N)$. Urutan waktu training metode klasifikasi dimulai dari yang tercepat adalah *Adaboost.MH-Single Stump*, disusul oleh *Adaboost.MH-Multi Stump*, kemudian *Adaboost.MH-Tree*, *Adaboost.MH-Product of Multi Stump*, *Adaboost.MH-LVQ*, *Adaboost.MH-GLVQ*, dan terakhir *Adaboost.MH-Multi Boundary*.



(a)



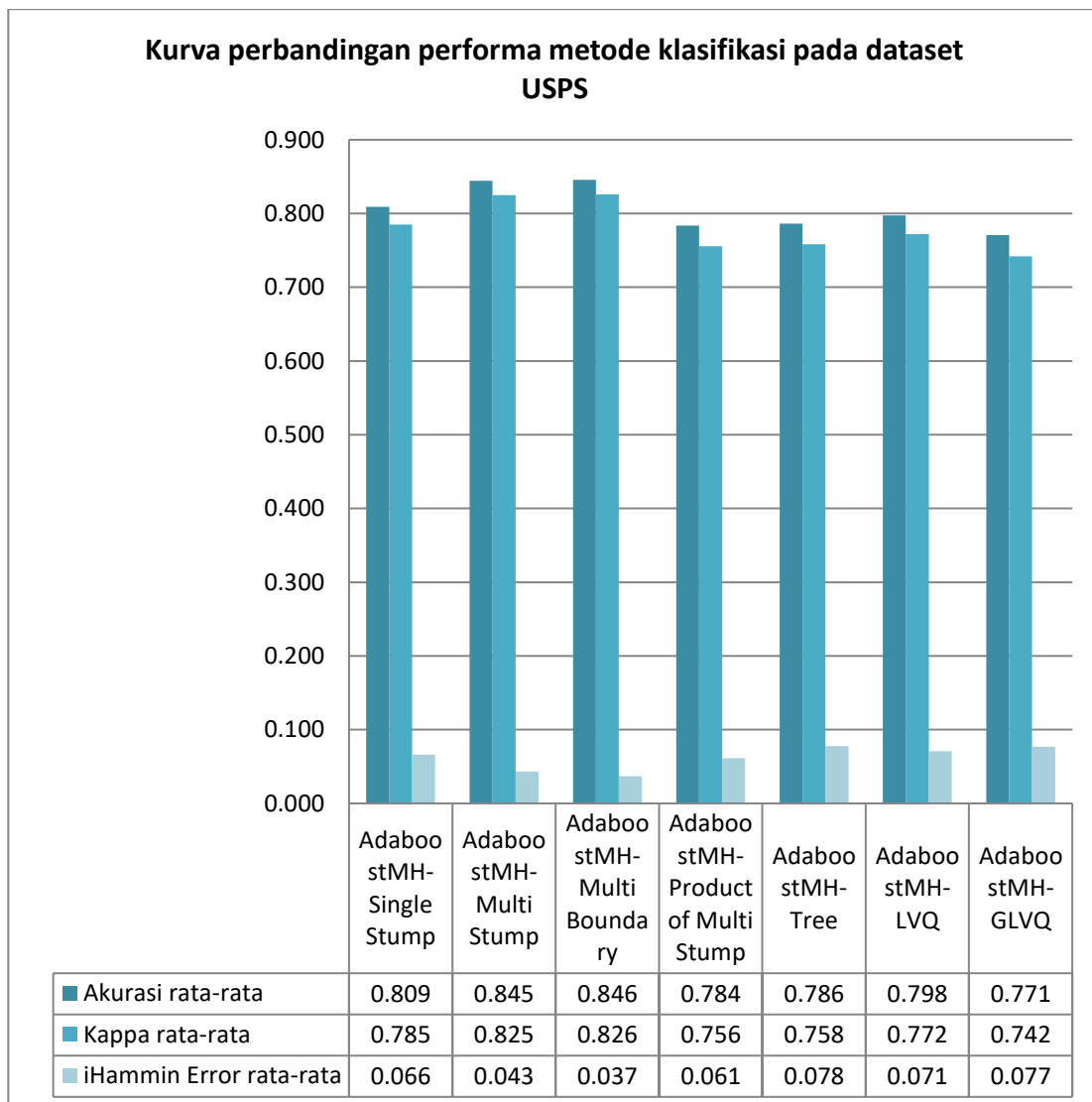
(b)

Gambar 5-13. Perbandingan waktu komputasi metode klasifikasi pada dataset USG (a) Waktu *training* (b) Waktu *testing*

Pada gambar 5-13 (b) ditampilkan waktu *testing* dari ketujuh metode klasifikasi yang diuji cobakan. Berdasarkan gambar 5-13 (b) dapat dikatakan bahwa waktu *testing* ketujuh metode klasifikasi yang diuji cobakan memiliki selisih yang sangat kecil satu sama lain. Urutan waktu testing dari metode klasifikasi dari yang tercepat adalah *Adaboost.MH-Single Stump*, disusul oleh *Adaboost.MH-LVQ*, selanjutnya *Adaboost.MH-Multi Stump*, *Adaboost.MH-Product of Multi Stump*, kemudian *Adaboost.MH-Tree*, *Adaboost.MH-Multi Boundary*, dan terakhir *Adaboost.MH-GLVQ*. Meskipun metode yang diusulkan memiliki waktu *testing* yang relatif lebih lama dari metode lainnya, akan tetapi selisih waktunya sangat kecil.

5.6 Evaluasi Hasil Percobaan 2.1

Berdasarkan data hasil percobaan 2, disebutkan bahwa pada dataset USPS, algoritma yang diusulkan memiliki performansi yang lebih baik daripada metode lainnya. Pada percobaan 1 dilakukan untuk berbagai nilai fitur, yaitu dari 1 sampai 200. Selanjutnya, dari 200 data uji tersebut diringkas dengan dicari rata-ratanya, baik untuk parameter *hamming error*, akurasi maupun koefisien *kappa*. Pada gambar 5.14 ditampilkan rata-rata nilai *hamming error*, akurasi maupun koefisien *kappa*.



Gambar 5-14 Perbandingan performa metode klasifikasi pada dataset USPS

Berdasarkan grafik pada gambar 5.14 dapat dilihat bahwa menurut Faktor *hamming error* metode yang diajukan dapat dikatakan sebagai metode yang paling baik, karena nilai rata-rata *hamming error* nya paling kecil. Akan tetapi apabila dilihat dari faktor akurasi, metode *Adaboost.MH-Multi Boundary Classifier* memiliki tingkat akurasi yang tidak jauh berbeda dari salah satu metode lain yang dibandingkan, dalam hal ini adalah *Adaboost.MH-Multi Stump*. Jika dilihat dari sisi nilai *kappa* pun demikian. Performa metode yang dihasilkan tidak jauh berbeda dari *Adaboost.MH-Multi Stump*, meskipun masih lebih unggul. Jika dibandingkan dengan metode lainnya, metode yang diusulkan memiliki selisih performa yang lebih besar. Selanjutnya performa ketujuh metode klasifikasi tersebut akan dievaluasi dengan metode *Pair-Wise Comparison* atau juga disebut dengan perbandingan *Head to Head*.

5.6.1 Perbandingan Performa Metode pada Dataset USPS Dilihat dari Faktor *Hamming Error*

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset USPS, dilihat dari faktor *Hamming error*. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-8 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor *HammingError*

Metode	A	B	C	D	E	F	G
A		1	0	21	192	184	193
B	199		1	198	198	200	199
C	200	199		200	200	200	200
D	179	2	0		197	185	193
E	8	2	0	3		92	152
F	16	0	0	15	108		187
G	7	1	0	7	48	13	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree
 F : Adaboost.MH-LVQ
 G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-8 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-9.

Tabel 5-9 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor *HammingError*

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	E,F,G	B,C,D
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	A,E,F,G	B,C
E	G	A,B,C,D,F
F	E,G	A,B,C,D
G	-	A,B,C,D,E,F

Keterangan:

A : Adaboost.MH-Single Stump
 B : Adaboost.MH-Multi Stump
 C : Adaboost.MH-Multi Boundary Classifier
 D : Adaboost.MH-Product of Multi Stump
 E : Adaboost.MH-Tree
 F : Adaboost.MH-LVQ
 G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-9 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset USPS dilihat dari faktor *hamming error*.

5.6.2. Perbandingan Performa Metode pada Dataset USPS Dilihat dari Faktor Akurasi

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset USPS, dilihat dari faktor akurasi. Dari 200 macam kasus percobaan yang

dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diuji cobakan sebagai berikut :

Tabel 5-10 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor Akurasi

Metode	A	B	C	D	E	F	G
A		1	1	167	192	178	196
B	199		89	199	200	200	200
C	199	111		199	200	200	200
D	33	1	1		80	51	176
E	8	0	0	120		45	183
F	22	0	0	149	155		195
G	4	0	0	24	17	5	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-10 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-11.

Tabel 5-11 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor Akurasi

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D,E,F,G	B,C
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	G	A,B,C,E,F
E	D,G	A,B,C,F
F	D,E,G	A,B,C

G	-	A,B,C,D,E,F
---	---	-------------

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-11 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset USPS dilihat dari faktor akurasi.

5.6.3 Perbandingan Performa Metode pada Dataset USPS Dilihat dari Faktor *Kappa*

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset USPS, dilihat dari faktor *kappa*. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-12 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor *Kappa*

Metode	A	B	C	D	E	F	G
A		1	1	167	192	178	196
B	199		89	199	200	200	200
C	199	111		199	200	200	200
D	33	1	1		80	51	176
E	8	0	0	120		45	183
F	22	0	0	149	155		195
G	4	0	0	24	17	5	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-12 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-13.

Tabel 5-13 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset USPS Berdasarkan Faktor *Kappa*

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D,E,F,G	B,C
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	G	A,B,C,E,F
E	D,G	A,B,C,F
F	D,E,G	A,B,C
G	-	A,B,C,D,E,F

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

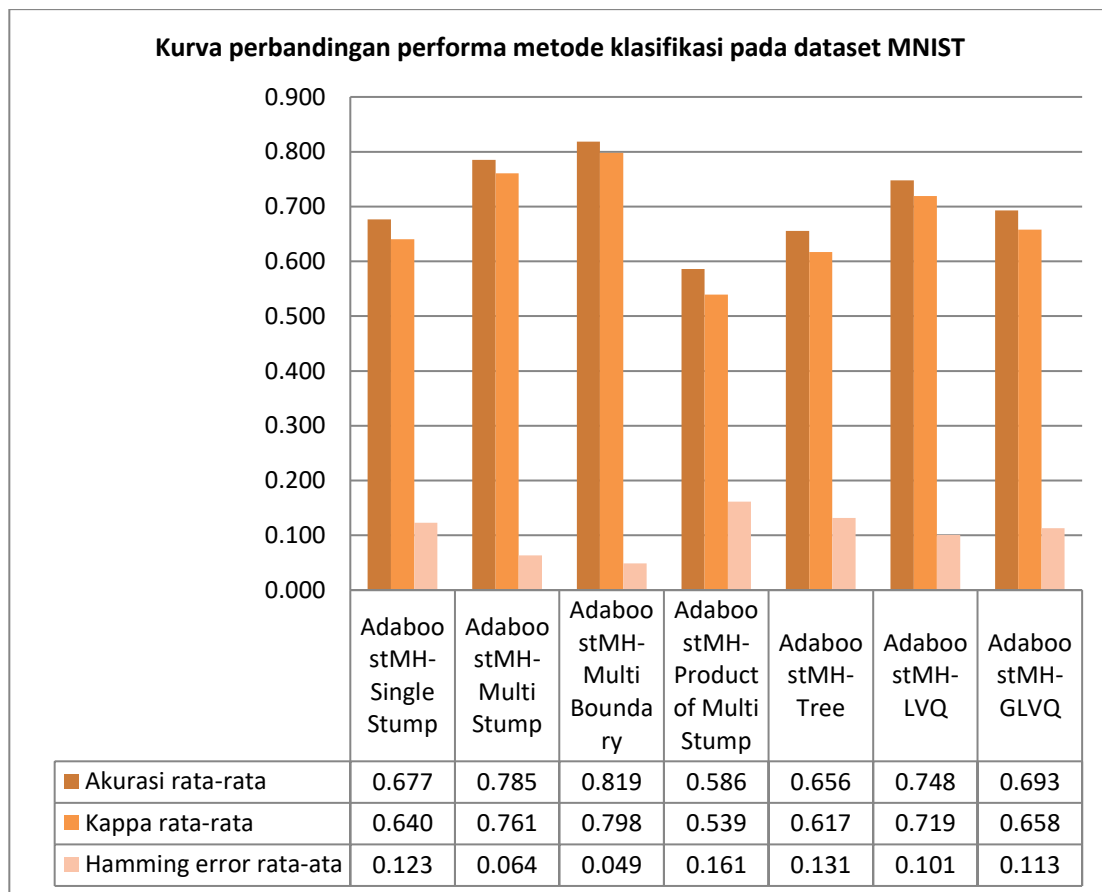
F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-13 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset USPS dilihat dari faktor *kappa*.

5.7. Evaluasi Hasil Percobaan 2.2

Berdasarkan data hasil percobaan 2, disebutkan bahwa pada dataset MNIST, algoritma yang diusulkan memiliki performa yang lebih baik daripada metode lainnya. Pada percobaan 2 dilakukan untuk berbagai nilai fitur, yaitu dari 1 sampai 200. Selanjutnya, dari 200 data uji tersebut diringkas dengan dicari rata-ratanya, baik untuk parameter *hamming error*, akurasi maupun koefisien *kappa*. Pada gambar 5.15 ditampilkan rata-rata nilai *hamming error*, akurasi maupun koefisien *kappa*.



Gambar 5-15 Perbandingan nilai koefisien kappa pada dataset MNIST.

Berdasarkan grafik pada gambar 5.15 dapat dilihat bahwa menurut Faktor *Hamming error* metode yang diajukan dapat dikatakan sebagai metode yang paling baik, karena nilai rata-rata *Hamming error* nya paling kecil. Selanjutnya, apabila dilihat dari faktor akurasi, metode *Adaboost.MH-Multi Boundary Classifier* memiliki tingkat akurasi yang paling tinggi, di antara metode-metode yang dibandingkan.

Selanjutnya, jika dilihat dari sisi rata-rata koefisien *kappa*, metode yang diajukan memiliki nilai rata-rata yang paling tinggi di antara metode-metode yang dibandingkan tersebut. Dengan demikian dapat dikatakan bahwa untuk dataset MNIST metode yang diusulkan unggul dari enam metode lain yang digunakan sebagai pembanding. Selanjutnya performa ketujuh metode klasifikasi tersebut akan dievaluasi dengan metode *Pair-Wise Comparison* atau juga disebut dengan perbandingan *Head to Head*.

5.7.1. Perbandingan Performa Metode pada Dataset MNIST Dilihat dari Faktor *Hamming Error*

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset MNIST, dilihat dari faktor *hamming error*. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-14 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor *HammingError*

Metode	A	B	C	D	E	F	G
A		0	0	200	104	9	33
B	200		0	200	200	199	196
C	200	200		200	200	200	198
D	0	0	0		6	1	0
E	96	0	0	194		4	11
F	191	1	0	199	196		185
G	167	4	2	200	189	15	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-14 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja.

Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-15.

Tabel 5-15 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor *HammingError*

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D, E	B,C,F,G
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	-	A,B,C,E,F,G
E	D	A,B,C,F,G
F	A,D,E,G	B,C
G	A,D,E	B,C,F

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-15 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset MNSIT dilihat dari faktor *hamming error*.

5.7.2. Perbandingan Performa Metode pada Dataset MNIST Dilihat dari Faktor Akurasi

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset MNIST, dilihat dari faktor akurasi. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-16 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor Akurasi

Metode	A	B	C	D	E	F	G
A		3	1	199	133	10	52
B	197		0	199	199	196	198
C	199	200		199	199	198	198
D	1	1	1		3	0	0
E	67	1	1	197		0	21
F	190	4	2	200	200		197
G	148	2	2	200	179	3	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-16 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-17.

Tabel 5-17 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor Akurasi

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D, E	B,C,F,G
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	-	A,B,C,E,F,G
E	D	A,B,C,F,G
F	A,D,E,G	B,C
G	A,D,E	B,C,F

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-17 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset MNSIT dilihat dari faktor akurasi.

5.7.3 Perbandingan Performa Metode pada Dataset MNIST Dilihat dari Faktor *Kappa*.

Berikut ini adalah proses perbandingan performa metode klasifikasi pada dataset MNIST, dilihat dari factor *kappa*. Dari 200 macam kasus percobaan yang dilakukan diperoleh matriks perbandingan *head to head* dari ketujuh metode yang diujicobakan sebagai berikut :

Tabel 5-18 Matriks *Head to Head* Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor *Kappa*

Metode	A	B	C	D	E	F	G
A		3	0	200	135	10	53
B	197		0	199	199	197	198
C	200	200		200	200	199	198
D	0	1	0		6	1	0
E	65	1	0	194		1	21
F	190	3	1	199	199		197
G	147	2	2	200	179	3	

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Selanjutnya, berdasarkan Table 5-18 akan dievaluasi untuk menentukan suatu metode menang terhadap metode mana saja, dan kalah terhadap metode mana saja. Setelah dilakukan perhitungan pada masing-masing pasangan, diperoleh data pada tabel 5-19.

Tabel 5-19 Ringkasan *Head to Head* Performa Metode Klasifikasi Pada Dataset MNIST Berdasarkan Faktor *Kappa*

Metode	Menang Terhadap Metode	Kalah Terhadap Metode
A	D, E	B,C,F,G
B	A,D,E,F,G	C
C	A,B,D,E,F,G	-
D	-	A,B,C,E,F,G
E	D	A,B,C,F,G
F	A,D,E,G	B,C
G	A,D,E	B,C,F

Keterangan:

A : Adaboost.MH-Single Stump

B : Adaboost.MH-Multi Stump

C : Adaboost.MH-Multi Boundary Classifier

D : Adaboost.MH-Product of Multi Stump

E : Adaboost.MH-Tree

F : Adaboost.MH-LVQ

G : Adaboost.MH-GLVQ

Berdasarkan tabel 5-19 dapat dikatakan bahwa metode yang diusulkan mampu menang (unggul) terhadap enam metode lain yang dibandingkan pada dataset MNSIT dilihat dari faktor *kappa*.

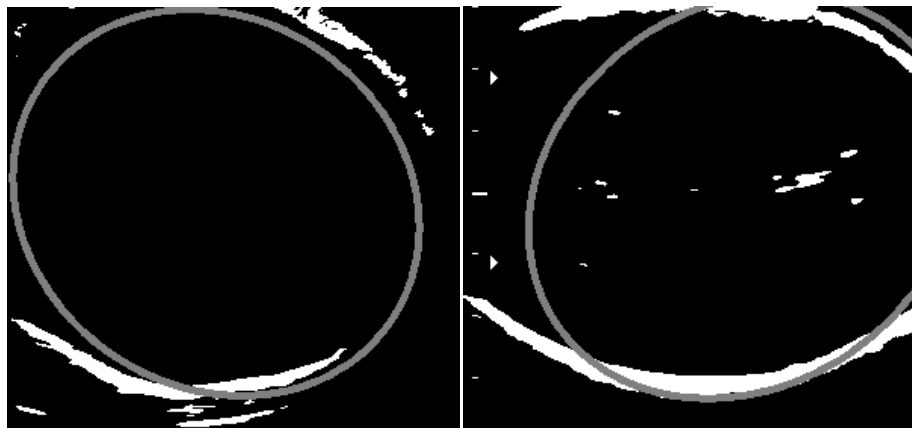
5.8. Evaluasi Hasil Percobaan 3

Berdasarkan data hasil percobaan 3, akan dihitung jumlah sampel yang berhasil *hit* pada proses aproksimasi. Jumlah sampel ini akan dihitung berdasarkan tipe organ janin. Setelah dilakukan penghitungan jumlah hit, diperoleh data pada tabel 5.20.

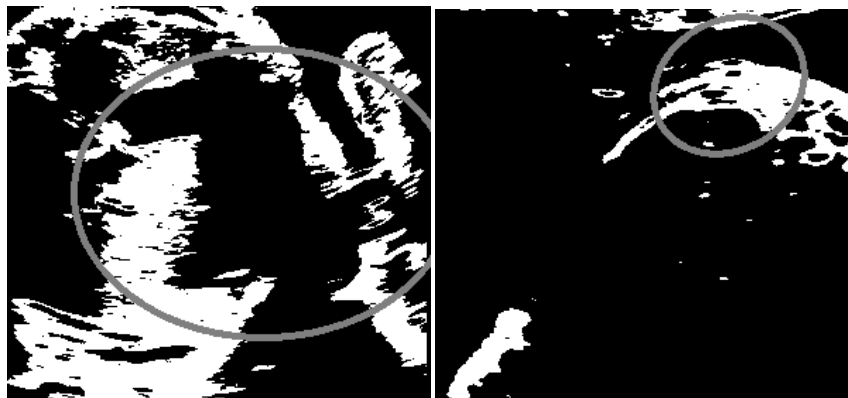
Tabel 5-20 Hit rate aproksimasi organ janin

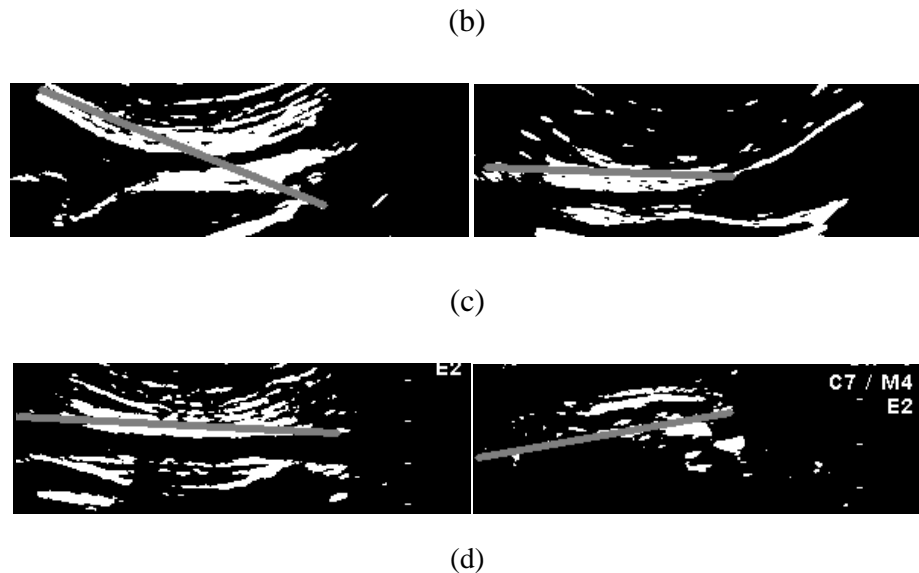
Organ Janin	Jumlah Hit	Jumlah Sampel	<i>Hit Rate</i>
Kepala	81	100	0.81
Perut	57	100	0.57
Tulang paha	72	100	0.72
Tuleng lengan atas	66	100	0.66

Dari tabel di atas dapat dilihat bahwa tingkat *hit* untuk organ kepala mencapai 81% dan untuk organ perut mencapai 57%. Untuk organ tulang paha (*femur*) mencapai 72 % dan untuk tulang lengan atas mencapai 66 %. Berbagai kesalahan aproksimasi pada organ janin dapat dilihat pada gambar 5-16.



(a)





Gambar 5-16 Aproksimasi organ janin yang gagal hit (a)Kepala (b) Perut (c) Tulang Paha (d) Tulang Lengan

Pada gambar 5-16 dapat dilihat bahwa kesalahan aproksimasi pada organ kepala bukan disebabkan karena tingginya *noise*, akan tetapi karena kesalahan dari proses aproksimasinya sendiri. Oleh sebab itu kesalahan aproksimasi pada organ kepala pada umumnya merupakan kesalan ukuran elips, atau orientasi elips itu sendiri. Sedangkan kesalahan aproksimasi pada organ perut, tulang paha dan tulang lengan atas sebagian besar disebabkan karena tingginya *noise* yang terletak pada area objek. Pada gambar 5-16 di bawah ini ditunjukkan contoh aproksimasi kurva yang salah pada organ perut, tulang paha dan tulang lengan atas disebabkan karena tingginya *noise* yang ada pada area objek.

5.9. Fitur Terbaik Yang Diperoleh

Berdasarkan pada saat proses seleksi fitur dan *ensemble classifier*, akan diperoleh fitur-fitur terbaik yang dapat dijadikan sebagai cirri untuk membedakan objek organ janin dengan gambar *background* citra USG. Pada peneilitian ini penulis menggunakan fitur yang sama untuk mengenali objek organ kepala, perut, tulang paha dan tulang lengan atas. Hal ini dikarenakan, pada setiap fitur sudah dipasangkan dengan *binary classifier* untuk masing-masing organ. Sehingga gabungan dari *binary classifier* dapat digunakan untuk mengenali bermacam-macam

objek organ janin pada citra, meskipun menggunakan fitur yang sama. Pada tabel berikut ini akan ditampilkan sepuluh fitur *Haar* terbaik yang dapat digunakan untuk proses deteksi objek organ janin pada citra USG dengan ukuran *window* 20x20.

Tabel 5-21 Sepuluh fitur terbaik untuk klasifikasi citra USG (window 20x20)

No	Janis Fitur <i>Haar</i>	Posisi (x)	Posisi (y)	Lebar (w)	Tinggi (h)
1	4q	3	1	13	15
2	4q	1	1	15	17
3	2h	7	1	11	7
4	4q	1	0	13	1
5	2h	2	1	15	15
6	2v	0	10	11	3
7	3h	3	5	2	10
8	2v	1	1	0	3
9	4q	4	2	9	9
10	4q	10	5	1	7

Dari tabel tersebut dapat dilihat bahwa fitur yang paling dominan adalah 4 *square*, disusul oleh fitur 2 *horizontal square*, dan dilanjutkan oleh fitur 2 *vertical square* dan 3 *horizontal square*.

Bab 6 Kesimpulan Dan Saran

Pada bagian ini akan dipaparkan kesimpulan yang dapat diambil mengenai hasil penelitian yang dilakukan. Selain itu, pada bab ini juga dipaparkan saran dari peneliti sebagai bentuk evaluasi dan kemungkinan perbaikan di masa yang akan datang.

6.1 Kesimpulan

Berbagai rangkaian penelitian yang telah dilakukan menghasilkan kesimpulan sebagai berikut :

1. Pada penelitian ini telah berhasil dikembangkan metode klasifikasi *Adaboost.MH* berbasis *Multi Boundary Classifier*. *Multi Boundary Classifier* merupakan *classifier* yang menggunakan dua nilai batas nilai untuk melakukan klasifikasi. Pencarian kedua batas nilai tersebut diperoleh dengan mencari rentang yang memaksimalkan *classification edge* dari data *training* yang diberikan.
2. Setelah dilakukan percobaan pada dataset USG, diperoleh hasil bahwa metode *Adaboost* berbasis *Multi Boundary Classifier* mampu mencapai rata-rata tingkat akurasi 93.9%, dengan rata-rata koefisien kappa mencapai 0.854. dan *hamming error* mencapai 0.032.
3. Setelah dilakukan perbandingan performa dengan metode lain yaitu *Adaboost-Multi Stump*, *Adaboost-Single Stump*, *Adaboost-Multi Stump*, *Adaboost-Tree*, *Adaboost-Product of Multi Stump*, *Adaboost-LVQ*, *Adaboost-GLVQ*, pada tiga jenis dataset (USG, USPS, dan MNIST), algoritma *Adaboost-Multi Boundary Classifier* mampu unggul enam metode algoritma yang dibandingkan pada ketiga dataset tersebut.
4. Dari sepuluh fitur *haar* terbaik yang terseleksi, lima diantaranya adalah jenis fitur *haar* dengan empat *square*, dua di antaranya adalah fitur *haar* dengan dua

square horizontal, dua di antaranya adalah fitur *haar* dengan dua *square vertical*, dan satu di antaranya adalah fitur *haar* dengan tiga *square horizontal*

5. Proses aproksimasi pada organ janin setelah dilakukan proses deteksi terlebih dahulu rata-rata mencapai tingkat *hit* 81% untuk organ kepala, 57% untuk organ perut, 72% untuk organ tulang paha, dan 66% untuk organ tulang lengan atas.

6.2 Saran

Penelitian ini tentunya masih memiliki banyak hal yang masih menjadi *open problem*, sehingga perlu dilakukan penelitian lanjutan. Oleh sebab itu penulis dapat memberikan saran untuk penelitian selanjutnya sebagai berikut ;

1. *Multi Boundary Classifier* dapat dikembangkan lagi sedemikian hingga, ada banyak *boundary* untuk setiap label kelas. Sementara ini baru dikembangkan satu *boundary* untuk setiap kelas.
2. Untuk mengetahui tingkat kehandalan metode *Adaboost.MH* berbasis *Multi Boundary Classifier*, penulis menyarankan untuk dilakukan uji coba dengan menggunakan data-data yang lebih bervariasi
3. Proses aproksimasi bentuk organ janin dengan menggunakan *thresholding* dan RHT masih memiliki tingkat *hit* yang rendah pada kasus organ perut dan tulang lengan atas. Oleh sebab itu diperlukan metode segmentasi tambahan agar sistem dapat melakukan aproksimasi dengan benar.

Referensi

- [1] Depkes RI. "Profil Kesehatan Indonesia 2008". Jakarta. 2008.
- [2] Saini K, M L Dewal, and Rohit M. "Ultrasound Imaging and Segmentation in the area of Ultrasound: A Review". *International Journal of Advance Science and Technology*, vol. 24, 2010.
- [3] Pribadi A. and dkk. "Ultrasonografi Obstetri & Ginekologi". Bandung: Sagung Seto, 2011.
- [4] L., Sisodia, R. S., Pallavi, V., Firtion, C., Ramachandran, G. Gupta. "Segmentation of 2D Fetal Ultrasound Images by Exploiting Context Information using Conditional Random Fields" in *IEEE Engineering in Medicine and Biology Society*, pp.7219-7222. 2011.
- [5] Satwika, I. Putu. "Tesis : Deteksi Dan Aproksimasi Elips Pada Pengukuran Kepala Janin Menggunakan Particle Swarm Optimizationberbasis Randomized Hough Transform Dua Dimensi". Universitas indonesia. 2013.
- [6] R. McLaughlin, "Randomized Hough Transform: Improved Ellipse Detection with Comparition," *Pattern Recognition Letters*, vol. 19, no. 3-4, pp. 299-305, 1998.
- [7] W. Lu, J. Tan and F. Floyd, "Automated Fetal Head Detection and Measurement in Ultrasound Images by Iterative Randomized Hough Transform," *Ultrasound in Med & Bio*, vol. 31, no. 7, pp. 929-936, 2005.
- [8] H.D, Guo, Y., Zhang, Y. Cheng, "A novel Hough transform based on eliminating particle swarm optimization and its applications," *Pattern Recognition*, vol. 42, pp. 1959-1969. 2009.
- [9] Freund, Y., & Schapire, R. E. "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences* 55. pp 119–139. 1997

- [10] Schapire, R. E., & Singer, Y. "Improved boosting algorithms using confidence-rated predictions". *Journal of Machine Learning* 37 pp 297–336. 1999.
- [11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 511–518. 2001.
- [12] Balazs Kegl and Robert Busa-Fekete. "Boosting products of base classifiers". *Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009*.
- [13] Imaddudin, Zaki. "Deteksi dan Pengukuran Biometri Janin Menggunakan Adaboost Classifier dan Randomize Hough Transform". Universitas Indonesia. 2013.
- [14] Fumera, Giorgio and Roli, Fabio. "A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems". *IEEE Transactions On Pattern Analysis And Machine Intelligence*, VOL. 27, NO. 6, JUNE 2005.
- [15] Xiang, Liu et all . "Research of Improved LVQ Neural Network by Adaboost Algorithm" *Journal of Applied Science* 13 (14) : 2658-2663. 2013.
- [16] T. Kohonen. "Learning Vector Quantization for Pattern Recognition" Finland: Report TKK-F-A601, Helsinki University of Technology, 1986.
- [17] A. Sato and K. Yamada, "A formulation of learning vector quantization using a new misclassification measure," in *Proceedings of the 14th International Conference on Pattern Recognition-Volume 1 - Volume 1*, ICPR '98, Washington, pp. 322-325. 1998
- [18] Wittlen, Ian H. "Data Mining : Practical Machine Learning Tools and Techniues 3rd Edition". 2011. Burlington. Elsevier.
- [19] M. Nixon and A. Aguado, *Feature Extraction & Image Pocessing Second Edition*, London: Elsevier Ltd, 2008.
- [20] Multiboot Library.<http://www.multiboost.org/benchmarks>

- [21] Handwritten Dataset
<http://archive.ics.uci.edu/ml/support/PenBased+Recognition+of+Handwritten+Digits>
- [22] Anthony J., dan Joane M. Garrett Viera. "Understanding Interobserver Agreement : The Kappa Statistic,". *Society of Teachers of Family Medicine*, pp.360-363, 2005.
- [23] J. Demmar. "Statistical Comparison of Classifiers over Multiple Data Sets". *Journal of Machine Learning Research* 7. pp.1-30.2006
- [24]. I. Wasito and B. Mirkin "Nearest Neighbour Approach In The Least-Squares Data Imputation Algorithms". *Information Science*. pp 1-25. 2005. Elsevier