



TESIS KI142502
KLASIFIKASI AKTIVITAS MENTAL DARI DATA
EEG MENGGUNAKAN NEURAL NETWORK DAN
FUZZY PARTICLE SWARM OPTIMIZATION
DENGAN CROSS-MUTATED OPERATOR

Stendy B. Sakur
NRP. 5113201044

DOSEN PEMBIMBING
Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D

PROGRAM MAGISTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015



TESIS KI142502
CLASSIFICATION OF MENTAL ACTIVITIES FROM EEG
DATA USING NEURAL NETWORK AND FUZZY
PARTICLE SWARM OPTIMIZATION WITH CROSS-
MUTATED OPERATOR

Stendy B. Sakur
NRP. 5113201044

SUPERVISOR
Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D

MAGISTER PROGRAMME
INFORMATICS ENGINEERING DEPARTMENT
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2015

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)

Di
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:
Stendy B. Sakur
NRP. 5113201044

Dengan judul:
**KLASIFIKASI AKTIVITAS MENTAL DARI DATA EEG MENGGUNAKAN NEURAL
NETWORK DAN FUZZY PARTICLE SWARM OPTIMIZATION DENGAN
CROSS-MUTATED OPERATOR**

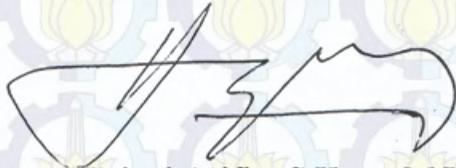
Tanggal Ujian: 07 Juli 2015
Periode Wisuda: September 2015

Disetujui oleh:



Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D
NIP. 194908231976032001

(Pembimbing)



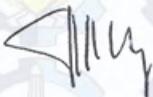
Dr. Agus Zainal Arifin, S.Kom, M.Kom
NIP. 197208091995121001

(Penguji 1)



Dr. Darlis Heru Murti, S.Kom, M.Kom
NIP. 197712172003121001

(Penguji 2)



Isye Arieshanti, S.Kom, M.Phil.
NIP. 197804122006042001

(Penguji 3)



Direktur Program Pasca Sarjana,

Prof. Dr. Ir. Anil Soeprijanto, M.T.
NIP. 196404051990021001

KLASIFIKASI AKTIVITAS MENTAL DARI DATA EEG MENGUNAKAN NEURAL NETWORK DAN FUZZY PARTICLE SWARM OPTIMIZATION DENGAN CROSS- MUTATED OPERATOR

Nama mahasiswa : Stendy B. Sakur
NRP : 5113201044
Pembimbing : Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D

ABSTRAK

Brain-computer interface (BCI) adalah sebuah sistem yang dapat mentransformasikan aktivitas listrik pada otak terhadap kegiatan mental yang berbeda ke dalam pengontrolan sinyal. *Electroencephalogram (EEG)* berdasarkan BCI merupakan hasil pengukuran aktivitas gelombang listrik dari otak yang dihasilkan oleh *celebral cortex*. Secara umum, *neural network* banyak digunakan sebagai metode klasifikasi bersifat *multi-class* namun sangat lambat dalam mencapai konvergensi atau solusinya dapat terjebak dalam lokal minimum. Algoritma *evolutionary* diusulkan untuk mengoptimasi pencarian bobot dari jaringan saraf tiruan diantaranya *Particle Swarm Optimization (PSO)*. Persoalan yang mendasar adalah kecepatan dari PSO kurang stabil sehingga perlu peningkatan pada semua komponen partikel. Komponen tersebut terdiri dari bobot inersia dari kecepatan sebelumnya, *individual knowledge* dan *group knowledge*. Ketiganya memiliki pengaruh besar dalam mencapai tingkat konvergensi sehingga sangat penting untuk di perhatikan pada saat melakukan optimasi pembobotan.

Tujuan dari penelitian ini adalah mengusulkan strategi baru pada metode *Improved Particle Swarm Optimization (IPSO)* yang menggunakan *Modified Evolutionary Direction Operator* dengan bobot Inertia Adaptif. Dimana *Fuzzy inference system* digunakan sebagai bobot inersia adaptif dalam mengoptimalkan pembobotan dari Neural Network. Strategi tersebut diharapkan dapat meningkatkan akurasi.

Hasil penelitian menunjukkan bahwa klasifikasi untuk Subjek Pertama adalah 54.20%, Subjek Kedua sebesar 58.40% dan Subjek Ketiga adalah 50.80% dengan rata – rata akurasi sebesar 54.48%. Dimana peningkatan untuk Subjek Pertama adalah 1.26%, Subjek Kedua 12.63% dan Subjek Ketiga adalah 0.80%. Dengan demikian Metode yang diusulkan masih lebih baik dari metode sebelumnya.

Kata kunci: *Brain-computer interface, Electroencephalogram, Artificial Neural Network, Particle Swarm Optimization, Fuzzy Inertia Weight, Evolutionary Direction Operator, Cross-mutated Operation.*

CLASSIFICATION OF MENTAL ACTIVITIES FROM EEG DATA USING NEURAL NETWORK AND FUZZY PARTICLE SWARM OPTIMIZATION WITH CROSS-MUTATED OPERATOR

By : Stendy B. Sakur
Student Identity Number : 5113201044
Supervisor : Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D

ABSTRACT

Brain-computer interface (BCI) is a system that can transform electrical activities in the brain to different mental activities into the control signals. *Electroencephalogram* (EEG) based BCI is the result of the measurement of the electrical activity of the brain waves generated by the cerebral cortex. In general, a neural network is widely used as a method of multi-class classification, but very slow in reaching convergence or the solution can be trapped in a local minimum. Evolutionary algorithm is proposed to optimize the search weights of the neural network, including the *Particle Swarm Optimization* (PSO). The fundamental issue is the speed of PSO is less stable that need improvement in all components of the particle. The component consists of the weight of the inertia of the previous speed, knowledge of individual and group knowledge. The component has a great influence in achieving the level of convergence, so it is important to note at the time of optimization weighting.

The purpose of this study is to propose a new strategy on methods *Improved Particle Swarm Optimization* (IPSO) which uses *Modified Evolutionary Direction Operator* with *Adaptive Inertia weights*. Where *Fuzzy Inference System* is used as adaptive inertia weight in optimizing the weighting of Neural Network. The strategy is expected to improve the accuracy.

The results show, for classification first subject is 54.20%, second subject of 58.40% and 50.80% for subject third with the average accuracy of 54.48%. Where the increase for first subject is 1.26%, second subject is 12.63% and the third subject 0.80%. Thus the proposed method is *Fuzzy Modified Evolutionary Direction operator* with *cross-mutated operation* is better than the previous method.

Keywords: *Brain-computer interfaces, Electroencephalogram, artificial Neural Network, Particle Swarm Optimization, Fuzzy Inertia Weight, Evolutionary Direction Operator, Cross-mutated operation*

UCAPAN TERIMA KASIH

Tanpa mereka penulis tidaklah berarti apa – apa oleh karena itu ucapan terima kasih yang tak terhingga penulis sampaikan kepada seluruh pihak yang telah membantu dan mendorong untuk penyelesaian Tesis ini, khususnya kepada:

1. Ibu Prof. Dr. Handayani Tjandrasa M.Sc, Ph.D, selaku dosen pembimbing yang selama ini telah membantu menyelesaikan Tesis penulis. Semoga kebaikan dan kemudahan yang di berikan kepada penulis akan dibalas dengan rahmat dan barokah dari Allah SWT.
2. Bapak Prof. Dr. Ir. Frans G. Ijong, M.Sc, selaku Direktur Politeknik Negeri Nusa Utara yang telah banyak memberikan bantuan, motivasi dan memfasilitasi proses perkuliahan penulis.
3. Bapak Dr. H. Agus Zainal Arifin S.Kom, M.Kom, Dr. Darlis Heru Murti, S.Kom, M.Kom, ibu Isye Arieshanti, S.Kom, M.Phil, selaku dosen penguji yang telah memberikan masukan dan saran perbaikan dalam Tesis ini.
4. Bapak Waskitho Wibisono, S.Kom, M.Eng, Ph.D, selaku Ketua Program Studi S2 Teknik Informatika yang telah memberikan kesempatan kepada penulis untuk menempuh pendidikan di Jurusan Teknik Informatika setelah penulis menyelesaikan pendidikan Program Pra-S2 Sainstek di jurusan Matematika Fakultas MIPA ITS selama dua semester.
5. Ibu Dr. Chastine Faticah, S.Kom, M.Kom, selaku Sekretaris Program Studi S2 Teknik Informatika yang telah banyak membantu penulis dalam proses perkuliahan.
6. Bapak Royyana Muslim I, S.Kom., M.Kom., Ph.D, selaku Dosen Wali yang telah memberikan arahan dalam proses pemilihan matakuliah dan telah banyak membantu menyelesaikan berbagai persoalan yang penulis hadapi.
7. Bapak Subchan, M.Sc., Ph.D (Dosen Jurusan Matematika), yang telah merekomendasikan penulis untuk melanjutkan studi di Program Pascasarjana ITS Fakultas MIPA, semoga beliau diberi kemudahan oleh Allah SWT

sebagaimana beliau telah memudahkan penulis dalam mengikuti perkuliahan di Jurusan Matematika.

8. Ibu Dr. Melania Suweni Muntini, MT (Dosen Jurusan Fisika), yang telah memfasilitasi penulis dalam mengikuti Program Pra-S2 Sainstek, semoga beliau diberikan kemudahan dalam segala tindakan. Dan penulis berharap Program Pra-S2 Sainstek tetap berlanjut karena dapat menjembatani berbagai perbedaan kultur pendidikan dan hal ini sangatlah bermanfaat bagi penulis sendiri yang telah merasakan keberhasilan program Pra-S2 Sainstek.
9. Bapak dan ibu Dosen Jurusan Teknik Informatika yang telah dengan ikhlas memberikan ilmunya kepada penulis selama masa perkuliahan.
10. Admin Lab. Pasca S2 dan Lab. Pemrograman 1 dan 2 yang telah memudahkan penulis untuk memanfaatkan segala fasilitasnya.
11. Teman – teman Jurusan Teknik Informatika terkhususnya untuk Gigih, Kartika, Laili, Yayan, Wanda, Lala, Muti, Khoirul, Eko, Pak Dimyati, Saiful (*malang*), Saiful (*ternate* – Pace Ipul), Denny, Fidi, Qulpi, Fahrur, Ichal, Yanuar, Dikul, Hani, Wilda, Hanifah, Rahmat, Isti, Yunita, Gregorius, Suastika, Kadijah, Febri, Doni, Indra, Pak Pasnur, Dhika Cikul dan Andika semoga diskusi yang mereka lakukan dengan penulis selama ini akan memberikan banyak manfaat.
12. Teman – teman Jurusan Matematika program S2-Sainstek Pak Alfian (dan istri beserta syifa), Pohet, Yudi, Winda, Rahma, Mba Restu, Asra, Pak Heru, Santoso, Abil, Akbar dan Etri semoga persahabatan yang kita bangun tidak pupus oleh jarak dan waktu.
13. Dan seluruh teman – teman yang tidak dapat penulis sebutkan satu persatu semoga semuanya diberikan kebaikan di dunia dan akhirat.

Kupersembahkan karya sederhana ini untuk

Istriku tersayang **Winnie Ayu Perwitasari, S.Kom**, bidadari penyejuk hati, penghibur dikala keputusasaan melanda dan penyemangat dalam menjalani hidupku. Terima kasih atas semua cinta, kasih sayang, perhatian dan pengertian serta doa yang selalu di berikan. Semoga Allah SWT senantiasa memberikan keberkahan dalam keluarga kita. Amin.

Untuk jagoan kecilku yang tersayang, **Muhammad Isa Al Kahfi Sakur**, buah hatiku yang dengan senyum tiga jarinya membuat ayah menghilangkan segala kegundahan dan keputusasaan yang sedang melanda. Semoga Allah SWT menjadikan sikecil manusia yang berilmu serta bermanfaat bagi Agama Islam dan menjadi sumber kebahagiaan dunia dan akhirat. Insya Allah, amin.

Ayahanda tercinta Abdul Sakur dan Ibunda Kusnariati dan mama Asni Abidin yang selalu mengiringi setiap langkah penulis dengan doa, cinta, kasih sayang yang tulus semoga Allah SWT memberikan keberkahan umur panjang dalam kesehatan dan keberkahan. Amin.

Saudara – saudaraku Seli, Shinta, Syeni dan Juwita. Serta keponakan yang selalu menggembirakan hati, Algifari, Anindita, Vira, Salsa bila, Faza, dino, Fizhan Ramadhansyah, dan semua keluarga besarku yang tidak dapat penulis sebutkan satu persatu, terima kasih atas Doa dan dukungan semuanya. Semoga Allah SWT memberikan kesehatan dan kebahagiaan kepada kita semua. Amin Ya Robbal'alam.

Akhirnya, semoga Berkah Ramadhan tahun ini dapat membuka pintu surga bagi kita semua. Insya Allah. Amin.

KATA PENGANTAR

Alhamdulillahirobbil'alamin, segala puji bagi Allah yang telah memberikan karunia ilmu yang luas serta kesehatan sehingga penulis dapat menyelesaikan tesis yang berjudul “*Klasifikasi Aktivitas Mental Dari Data EEG Menggunakan Neural Network Dan Fuzzy Particle Swarm Optimization Dengan Cross – Mutated Operator*”. Sholawat dan salam semoga tetap tercurahkan kepada junjungan kita Nabi besar *Muhammad Shollallohu ‘Alaihi Wassalam* – rosul mulia yang telah membimbing manusia dari jurang kenistaan ke tanah lapang yang penuh harapan dan cahaya kebenaran – para sahabat, tabi’in dan sampai kepada kita pengikutnya yang selalu setia sampai akhir zaman.

Tesis ini disusun sebagai salah satu persyaratan akademis bagi mahasiswa untuk meraih gelar Magister di Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Untuk itu penulis mengucapkan terima kasih yang sedalam – dalamnya kepada semua dosen pembimbing, penguji, pengajar, karyawan serta rekan – rekan kuliah yang telah banyak membantu menyelesaikan Tesis ini. Penelitian yang dilakukan dengan mengusulkan strategi baru yaitu Fuzzy improved particle swarm optimization dan cross-mutated operation yang mampu meningkatkan akurasi pada proses klasifikasi namun hasilnya masih perlu ditingkatkan lagi sehingga studi lanjut diperlukan dalam melakukan penyempurnaan. Penulis menyadari bahwa penelitian ini masih banyak kekurangannya sehingga saran dan kritik yang bersifat membangun sangatlah dibutuhkan.

Akhirnya, hanya kepada Allah Subahanahu Wa Ta’ala penulis berlindung dari segala kesalahan dan kehilafan semoga penelitian ini dapat bermanfaat bagi masyarakat luas.

Surabaya, Juli 2015

Penulis

DAFTAR ISI

ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vii
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Tujuan dan Manfaat	4
1.4 Kontribusi Penelitian	4
1.5 Batasan Masalah	5
BAB 2 DASAR TEORI	7
2.1 Brain-Computer Interface	7
2.2 Electroencephalogram (EEG)	9
2.3 Ekstraksi Fitur EEG	12
2.3.1 Properti Fitur	12
2.3.2 Pertimbangan Variasi Waktu EEG	13
2.4 Metode Ekstraksi Fitur	13
2.4.1 Metode Welch's Periodogram	14
2.4.2 Metode Principle Component Analysis	17
2.5 Klasifikasi Algoritma	18
2.5.1 Klasifikasi Taksonomi	18
2.5.2 Masalah utama klasifikasi dalam penelitian BCI	18
2.5.3 Metode Neural Network	19
2.6 Algoritam Evolusioner	22
2.6.1 Algoritma Genetik	23

2.6.2	Particle Swarm Optimization.....	25
2.6.2a	IPSO: Constriction and Inertia Weight Factos	27
2.6.2b	IPSO: Fuzzy Inertia Weight	28
	<i>Fuzzification</i>	32
	<i>Fuzzy Inference System</i>	33
	<i>Defuzzification</i>	36
2.7	Dataset Penelitian	37
2.7.1	Percobaan.....	37
2.7.2	Format Data	38
2.7.3	Informasi Teknis.....	38
BAB 3 METODE PENELITIAN		39
3.1	Studi Literatur	40
3.2	Perancangan Algoritma	41
3.2.1	Pre-Processing	41
3.2.2	Fitur Ekstraksi.....	42
3.2.3	Pelatihan Data.....	44
3.2.4	Peningkatan Algoritma IPSO	44
3.3	Perangkat Lunak	51
BAB 4 HASIL PENGUJIAN DAN EVALUASI		53
4.1	Lingkungan Dan Data Pengujian	53
4.2	Pre-Processing dan Fitur Ekstraksi.....	53
4.2.1	Prosess Filtering	54
4.2.2	Proses Ekstraksi Fitur	60
4.2.3	Reduksi Data EEG	65
4.2.4	Normalisasi Data	66
4.2.5	Data Akhir	67
4.3	Pelaksanaan Pengujian	67
4.3.1	Parameter Pengujian	68
4.3.2	Pengujian Metode IPSONN	70
4.3.3	Pengujian Metode FMEDO	75

4.4	Evaluasi	82
4.4.1	Perbandingan Fitness	82
4.4.2	Pengaruh Jumlah Pengujian	83
4.4.3	Pengaruh Dimensi PCA	85
4.4.4	Pengaruh Pembatasan Kecepatan dan Posisi	86
4.4.5	Pengaruh Parameter	89
4.4.6	Pengaruh Fuzzy Bobot Inersia	90
4.4.7	Pengaruh Nilai Constriction	92
4.4.8	Pengaruh Fitness dan Akurasi	92
BAB 5 KESIMPULAN DAN SARAN		93
5.1	Kesimpulan	93
5.2	Saran	93
DAFTAR PUSTAKA		95

DAFTAR GAMBAR

Gambar 2.1 A <i>Brain-Computer Interface</i>	7
Gambar 2.2 Desain dasar dan operasi Sistem BCI	8
Gambar 2.3 Peralatan untuk merekam EEG	9
Gambar 2.4 Sistem peletakan elektroda 10 – 20	10
Gambar 2.5 Empat gelombang EEG	11
Gambar 2.6 Ilustrasi proses segmen	17
Gambar 2.7 Struktur neuron pada jaringan saraf	19
Gambar 2.8 Fungsi pengaktifan sederhana pada jaringan saraf	20
Gambar 2.9 Arsitektur jaringan <i>backpropagation</i>	22
Gambar 2.10 Skema dari Operator <i>evolutionary direction</i>	23
Gambar 2.11 Prosedur optimasi	26
Gambar 2.12 Fungsi keanggotaan dari (a) x-axis: $\ \zeta(t)\ $, y-axis: $\mu_{N_1} \ \zeta(t)\ $, (b) x-axis: t/T , y-axis: $\mu_{N_2} t/T$	31
Gambar 2.13 Kurva Gaussian	31
Gambar 2.14 Output singleton fuzzy untuk <i>bobot inersia adaptif</i>	33
Gambar 2.15 Output singleton fuzzy untuk <i>control parameter</i>	36
Gambar 3.1 Tahapan metodologi penelitian	39
Gambar 3.2 Diagram alir untuk prosedur analisis	42
Gambar 3.3 Prosedur perhitungan PCA	43
Gambar 3.4 Tiga Layer <i>feed-forward Neural Network</i>	44
Gambar 3.5 Struktur SWARM	45
Gambar 3.6 Strategi baru yang di usulkan pada algoritma IPSO	46
Gambar 3.7 Bagan Alir dari IPSONN	49
Gambar 3.8 Flowchart untuk menentukan <i>Fuzzy Inertia Weight</i>	50
Gambar 4.1 Channel C3 – Sinyal EEG (atas) dan SSL Head Geometry (Bawah) untuk Subjek 1	54
Gambar 4.2 Channel C3 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	55

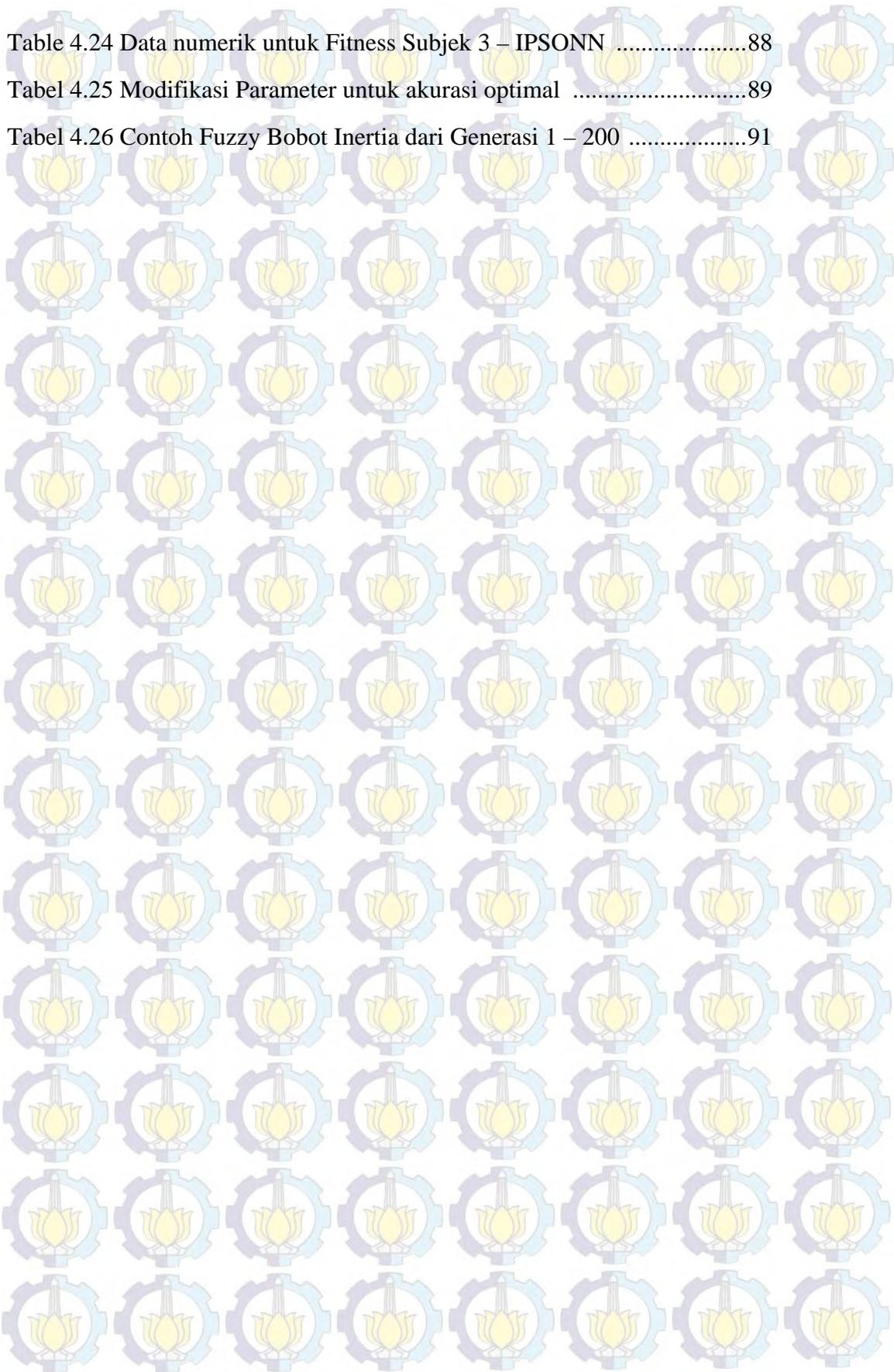
Gambar 4.3 Channel Cz – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	55
Gambar 4.4 Channel C4 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	55
Gambar 4.5 Channel CP1 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	56
Gambar 4.6 Channel CP2 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	56
Gambar 4.7 Channel P3 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	56
Gambar 4.8 Channel Pz – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	57
Gambar 4.9 Channel P4 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)	57
Gambar 4.10 Rata – rata Nilai PSD Subjek 1 (Left, Right dan Word)	61
Gambar 4.11 Fitness Subjek Pertama untuk Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	71
Gambar 4.12 Kemampuan Pelatihan Neural Network Subjek Pertama untuk Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	72
Gambar 4.13 Fitness Subjek Kedua Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	72
Gambar 4.14 Kemampuan Pelatihan Neural Network Subjek kedua Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	73
Gambar 4.15 Fitness Subjek Ketiga Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	73
Gambar 4.16 Kemampuan Neural Network Subjek Ketiga Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	74
Gambar 4.17 Fitness Subjek Pertama dari Metode FMEDO (<i>Fuzzy Modified Evolutionary Direction dan Cross - Mutated Operation</i>).....	76
Gambar 4.18 Kemampuan Neural Network dari Subjek Pertama untuk Metode FMEDO (<i>Fuzzy Modified Evolutionary Direction Operator dan Cross-Mutated Operation</i>)	77

Gambar 4.19 Fitness Subjek Kedua untuk Metode FMEDO (Fuzzy Modified Evolutionary Direction Operator dan Cross-Mutated Operation)	77
Gambar 4.20 Kemampuan Neural Network dari Subjek Kedua untuk Metode FMEDO (<i>Fuzzy Modified Evolutionary Direction Operator dan Cross-Mutated Operation</i>).....	78
Gambar 4.21 Fitness Subjek Ketiga untuk Metode FMEDO (Fuzzy Modified Evolutionary Direction Operator dan Cross-Mutated Operation)	78
Gambar 4.22 Kemampuan Neural Network dari Subjek Ketiga untuk Metode FMEDO (<i>Fuzzy Modified Evolutionary Operator dan Cross-Mutated Operation</i>)	79
Gambar 4.23 Distribusi Fuzzy Inertia Weight Subjek Pertama untuk Metode FMEDO (<i>Fuzzy Modified Evolutionary Direction Operator dan Cross-Mutated Operation</i>)	80
Gambar 4.24 Distribusi Fuzzy Inertia Weight Subjek Kedua untuk Metode FMEDO (<i>Fuzzy Modified Evolutionary Direction Operator dan Cross-Mutated Operation</i>)	80
Gambar 4.25 Distribusi Fuzzy Inertia Weight Subjek Ketiga untuk Metode FMEDO (<i>Fuzzy Modified Evolutionary Direction Operator dan Cross-Mutated Operation</i>)	81
Gambar 4.26 Fitness Subjek 1 (Metode IPSONN & FMEDO)	82
Gambar 4.27 Fitness Subjek 2 (Metode IPSONN & FMEDO)	83
Gambar 4.28 Fitness Subjek 3 (Metode IPSONN & FMEDO)	83
Gambar 4.29 Fitness Subjek Pertama (Tunggal) untuk Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	84
Gambar 4.30 Fitness Subjek Pertama (Rata – rata fitness dari seluruh percobaan) untuk Metode IPSONN (<i>Improved Particle Swarm Optimization Neural Network</i>)	84
Gambar 4.31 Grafik Pengaruh Dimensi reduksi Principle Component Analysis (PCA) terhadap Akurasi untuk Subjek pertama	85
Gambar 4.32 Bentuk Fitness dari Subjek 3 (Rata – rata fitness dari tiga subjek)	89
Gambar 4.33 Bentuk Fitness dari Subjek 3 untuk Metode IPSONN (<i>Improved Partilce Swarm Optimization Neural Network</i>)	90

DAFTAR TABEL

Tabel 4.1 Contoh data mentah EEG dengan SSL Head Geo	58
Tabel 4.2 Nilai Rata PSD Subjek 1, 8 Channel Kelas 2 (left)	61
Tabel 4.3 Nilai Rata PSD Subjek 1, 8 Channel Kelas 3 (Right)	62
Tabel 4.4 Nilai Rata PSD Subjek 1, 8 Channel Kelas 7 (Word)	62
Tabel 4.5 Nilai Rata PSD Subjek 2, 8 Channel Kelas 2 (Left)	62
Tabel 4.6 Nilai Rata PSD Subjek 2, 8 Channel Kelas 3 (Right)	63
Tabel 4.7 Nilai Rata PSD Subjek 2, 8 Channel Kelas 7 (Word)	63
Tabel 4.8 Nilai Rata PSD Subjek 2, 8 Channel Kelas 2 (Left)	64
Tabel 4.9 Nilai Rata PSD Subjek 2, 8 Channel Kelas 3 (Right)	64
Tabel 4.10 Nilai Rata PSD Subjek 3, 8 Channel Kelas 7 (Word)	64
Tabel 4.11 Contoh dan hasil reduksi dengan PCA (hanya ditampilkan 10 sampling) untuk Subjek 1 Training 1	65
Tabel 4.12 Jumlah data yang akan dipergunakan untuk proses klasifikasi aktivitas mental (left, right, word)	67
Tabel 4.13 Parameter yang digunakan pada penelitian	68
Tabel 4.14 Struktur dan komponen yang digunakan pada PSO dan NN	69
Tabel 4.15 Pembagian data Cross-Validation untuk pengujian sistem	70
Tabel 4.16 Hasil Akurasi dari Metode IPSONN	71
Tabel 4.17 Nilai Mean dan Standard Deviation Metode IPSONN	75
Tabel 4.18 Hasil Pengujian dari Metode FMEDO	76
Tabel 4.19 Penggunaan Fuzzy adaptif pada Metode FMEDO (Nilainya diperoleh secara otomatis dari sistem)	79
Tabel 4.20 Mean dan Standard Deviation pada Metode FMEDO	79
Tabel 4.21 Perbandingan Akurasi klasifikasi dari Metode IPSONN dan FMEDO	81
Tabel 4.22 Pengaruh Dimensi PCA terhadap jumlah data	85
Tabel 4.23 Data numerik untuk Fitness Subjek 3 (Rata – rata dari ketiga subjek)	87

Table 4.24 Data numerik untuk Fitness Subjek 3 – IPSONN	88
Tabel 4.25 Modifikasi Parameter untuk akurasi optimal	89
Tabel 4.26 Contoh Fuzzy Bobot Inertia dari Generasi 1 – 200	91



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Brain-Computer Interface (BCI) menggunakan *Electroencephalography (EEG)* untuk mengukur aktivitas gelombang listrik pada otak yang merefleksikan keinginan ataupun tujuan dari pengguna sehingga dapat menyediakan keluaran *nonmuscular* baru untuk sarana komunikasi sebagai sebuah teknologi antarmuka manusia dan komputer (Chai et al., 2014). *Invasive* merupakan salah cara untuk mendapatkan aktivitas gelombang listrik dari otak dengan menanamkan elektroda secara langsung ke bagian *celebral cortex* sehingga gelombang listrik yang diperoleh tidak mengalami distorsi. Namun cara ini beresiko besar sehingga diperlukan alternatif lain yaitu pendekatan *Non-Invasive* dimana aktivitas gelombang listrik di dapatkan melalui elektroda yang diletakkan pada permukaan kulit kepala sehingga aman. Akan tetapi gelombang yang diperoleh tidak sekuat tipe *Invasive* karena aktivitas listrik yang di hasilkan oleh neuron dari otak *celebral cortex* dengan tegangan yang kecil (mV) akan dikirim melalui jaringan saraf menuju ke permukaan kulit kepala yang konsekuensinya akan mengalami distorsi yang cukup besar. *Brain-Computer Interface* bermanfaat bagi pengguna yang mengalami kerusakan total untuk saraf pengendali otot gerak ataupun penyakit lain semisal *Amyotrophic lateral sclerosis (ALS)*, *brainstem stroke*, *brain* atau *spinal cord injury* (Cedera tulang belakang), *cerebral palsy*, *muscular dystrophies*, *multiple sclerosis* yang telah mempengaruhi hampir dua juta orang di Amerika Serikat (Wolpaw et al., 2002).

Electroencephalogram (EEG) berdasarkan pada BCI merupakan hasil pengukuran dari aktivitas gelombang listrik dari otak yang dihasilkan oleh bagian *celebral cortex* sekalipun beberapa bagian otak juga memberikan kontribusi gelombang listrik. Gelombang yang dihasilkan dari EEG – BCI berkisar antara 1 – 30 Hz band, dimana gelombang δ (1 – 3 Hz), θ (4 – 7 Hz), α (8 – 13 Hz) dan β (14 – 30 Hz) (Chai et al., 2014). Teknik untuk memonitoring aktivitas otak termasuk sinyal *Electroencephalogram (EEG)*, *Electrocorticogram (ECoG)*,

Positron Emission Tomography (PET), *functional Magnetic Resonance Imaging (fMRI)* dan *Magnetoencephalography (MEG)*, namun sinyal EEG lebih populer digunakan untuk mengimplementasi BCI dengan biaya yang rendah, *non-invasive* dan relatif mudah untuk merekam sinyal otak (Zhou et al., 2008). Sinyal EEG dapat menjadi salah satu alternatif bagi pengguna keterbatasan fisik untuk menggerakkan kursi roda (Chai et al., 2014) dengan mengirimkan perintah hanya ke alat elektronik melalui aktivitas otak (Lotte et al., 2007), berbagai macam penyakit seperti epilepsi dapat ditentukan melalui sinyal EEG (Subasi and Gursoy, 2010). Sinyal EEG yang di peroleh melalui pendekatan *Non-Invasive* memiliki gelombang yang cukup lemah sehingga sangatlah penting untuk melakukan proses digital agar dapat mengklasifikasikan perintah dengan benar.

Banyak penelitian dilakukan untuk mencari metode yang dapat menentukan tingkat keakuratan dan waktu komputasional dalam proses klasifikasi sinyal EEG, termasuk metode *hybrid* diusulkan untuk mencapai tujuan tersebut. Metode *Recurrent Neural Networks (RNNs)* digunakan untuk mengklasifikasikan sinyal EEG berdasarkan *Power Spectral Density (PSD)* menggunakan metode *Welch* yang bertujuan untuk menentukan penyakit *Epileptic Seizure*. Penelitian ini melakukan tiga tahapan proses yaitu ekstraksi fitur menggunakan metode *Welch*, mereduksi fitur menggunakan nilai statistik dan klasifikasi menggunakan *Recurrent Neural Networks*. Untuk menguji kinerja dari RNN dalam masalah klasifikasi yang sama maka digunakan *multilayer-perceptron neural network* dengan menggunakan *learning backpropagation* yang tingkat akurasinya mencapai 100% sekalipun tidak dijelaskan waktu komputasinya (Naderi and Mahdavi-Nasab, 2010). Metode *hybrid* juga diusulkan untuk menentukan penyakit *Epileptic Seizure* dengan melakukan ekstraksi fitur menggunakan *Discrete Wavelet Transform (DWT)* yang kemudian direduksi menggunakan *Principle Component Analysis (PCA)*, *Independent Component Analysis (ICA)* dan *Linear Discriminant Analysis (LDA)*, hasil reduksi menjadi *input* untuk *Support Vector Machine (SVM)* dalam menentukan klasifikasi yaitu mengidap penyakit *Epileptic Seizure* atau tidak hasilnya menunjukkan LDA mencapai 100% lebih tinggi dari keduanya yaitu ICA (99.5%) dan PCA (98.75%) (Subasi and Gursoy, 2010), SVM hanya dapat digunakan untuk mengklasifikasikan dua kelas

saja. Penelitian lainnya yang tidak kalah penting adalah melakukan pengontrolan terhadap *electronic device* untuk berinteraksi dengan lingkungan sekitar, misalnya pengguna dengan keterbatasan fisik dapat menggunakan kursi roda yang dapat dikontrol dengan mengirimkan perintah melalui aktivitas otak ke BCI system. Metode *Artificial Neural Networks (ANN)* dengan menggunakan *Cross-Mutated Operation (CM)* digunakan untuk mengklasifikasikan tiga perintah dasar tersebut, dimana sinyal EEG di ekstraksi fiturnya menggunakan *Hilber-Huang Transform (HHT)*, yang hasilnya lebih baik dibandingkan dengan *Fast Fourier Transform (FFT)*, kemudian pembobotan dari ANN ditentukan dengan menggunakan *Particle Swarm Optimization (PSO)* dimana kecepatan dari partikel ditingkatkan dengan menggunakan *Fuzzy Inertia Weight* sehingga memiliki karakteristik non-linear (Chai et al., 2014). Metode ini dikembangkan dari *improve particle swarm optimization* yang mengusulkan penambahan bobot inersia untuk menyeimbangkan pencarian global dan lokal (Eberhart and Shi, 2000; Ling et al., 2012). Klasifikasi pekerjaan mental merupakan hal penting untuk di perhatikan sehingga banyak penelitian dilakukan sehubungan dengan hal tersebut. Metode *Neural Networks* yang proses pembobotannya dilakukan dengan menggunakan *Particle Swarm Optimization (PSO)* dimana PSO ditingkatkan dengan menggunakan *Evolutionary Direction Operator (EDO)* dan *Migration Operator* yang bertujuan untuk mempercepat pencarian global. Metode ini memiliki keuntungan yaitu (1) dapat mempertahankan dari konvergensi yang premature; (2) dapat mempercepat pencarian global dengan menggunakan MEDO; dan (3) dapat menemukan solusi yang terbaik. Namun kelemahannya masih menggunakan bobot inersia yang memiliki sifat linear (Lin and Hsieh, 2009). Klasifikasi dengan menggunakan metode *Correlation* berdasarkan EEG map di usulkan untuk menentukan aktivitas mental (Ubeda et al., 2013).

Berdasarkan analisis tersebut strategi baru diusulkan untuk menghilangkan kelemahan yang dimiliki oleh metode IPSO (Lin and Hsieh, 2009) dimana kecepatan dari PSO masih menggunakan bobot inersia yang berusaha menyeimbangkan antara pencarian global dan lokal namun masih bersifat linear akan tetapi kebanyakan optimasi memiliki karakteristik non-linear. Sehingga diusulkan metode *Neural Networks* dengan mengoptimalkan proses pembobotan

melalui *Particle Swarm Optimization* yang menggunakan *Fuzzy Inertia Weight* (bobot inersia adaptif) dan *Cross-mutated Operation* berdasarkan metode *Improved Particle Swarm Optimization* (IPSO) untuk klasifikasi aktivitas mental dari sinyal EEG.

1.2 Perumusan Masalah

Berdasarkan penelitian yang telah di analisa pada bagian sebelumnya maka penting untuk membuat strategi baru dalam rangka peningkatan algoritma yang optimal. Oleh karena itu, permasalahan yang akan di selesaikan pada penelitian ini adalah:

1. Bagaimana membangun strategi baru kedalam Algoritma *Metode Improved Particle Swarm Optimization* dengan menambahkan nilai Fuzzy terhadap bobot inersia (*Inertia Weight*) untuk kecepatan partikel swarm sehingga dapat meningkatkan akurasi.
2. Bagaimana mempertahankan Metode *Particle Swarm Optimization* (PSO) agar tidak terjebak kedalam minimum lokal dengan menggunakan *Cross-Mutated Operation*.

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah mengusulkan strategi baru pada metode *Improved Particle Swarm Optimization Neural Network* (IPSONN) yang menggunakan *Modified Evolutionary Direction Operator* dengan menggunakan bobot inersia adaptif berdasarkan pada *Fuzzy inference system* dalam mengoptimalisasi pembobotan dari *Neural Network*. Manfaat penelitian ini diharapkan agar strategi yang diusulkan dapat meningkatkan akurasi yang lebih baik.

1.4 Kontribusi Penelitian

Kontribusi yang diusulkan pada penelitian ini adalah strategi penentuan bobot *Neural Network* melalui optimalisasi dari *particle swarm optimization* menggunakan *Fuzzy Inertia Weight* berdasarkan metode *improved particle swarm optimization neural network* (IPSONN).

1.5 Batasan Masalah

Batasan penelitian ini menggunakan data berdasarkan pada IDIAP Research Institute pada BCI Competition III untuk dataset V, mental imagery multi-class (Chiappa and Millan).

BAB 2

DASAR TEORI

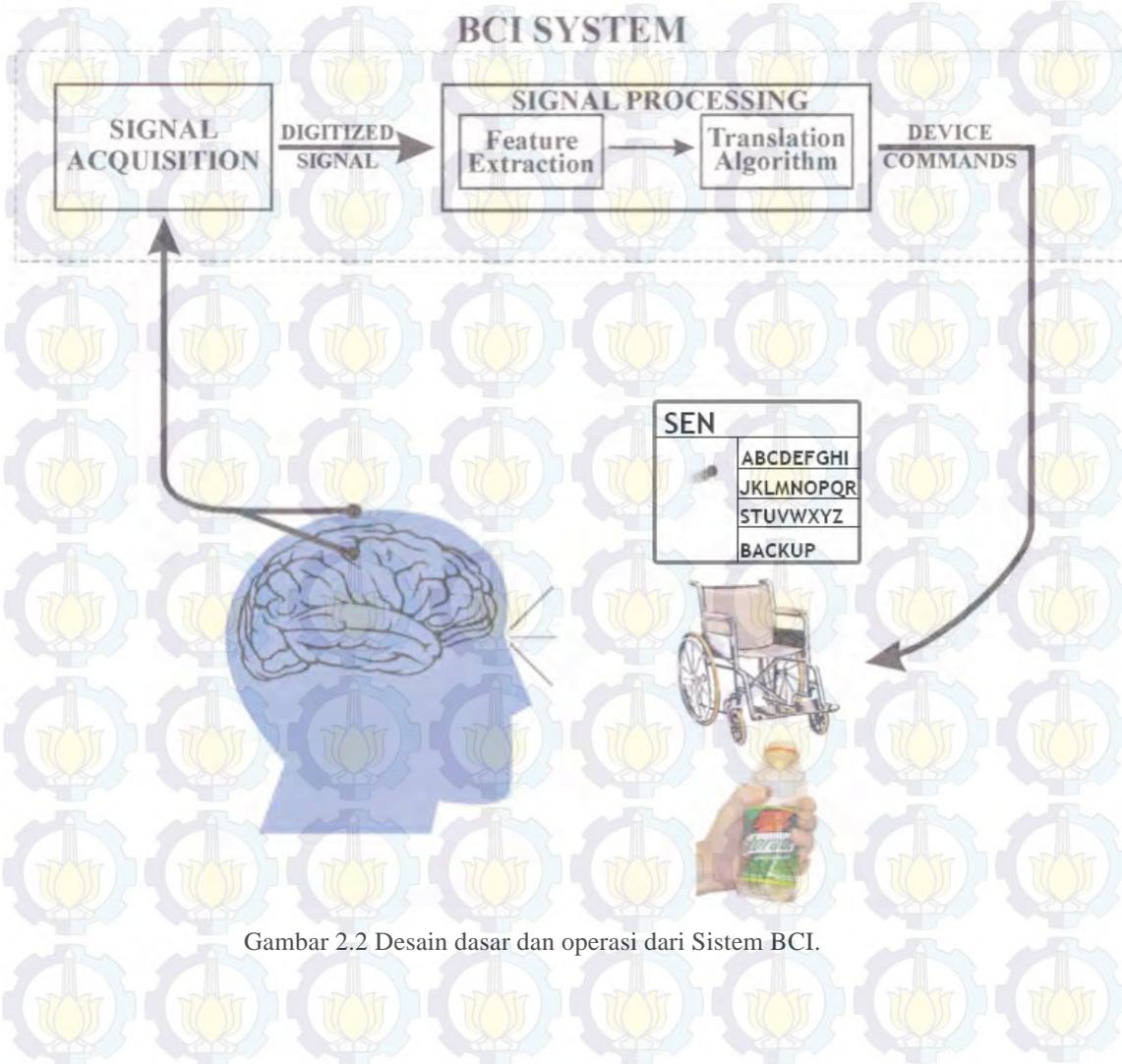
2.1 Brain-Computer Interface

Brain-Computer Interface (BCI) merupakan salah satu sistem yang dapat digunakan untuk berkomunikasi antara manusia dengan komputer melalui perintah yang dibangun oleh aktivitas gelombang listrik berdasarkan keinginan atau minat dari pengguna. Metode yang dapat digunakan untuk memonitoring aktivitas otak yang berfungsi sebagai BCI diantaranya, *Electroencephalography (EEG)* yang *non-invasive* sedangkan untuk *invasive* dapat terdiri dari *Magenetoencephalography (MEG)*, *Positron Emission Tomography (PET)*, *Single Photon Emission Computed Tomography (SPECT)*, *functional Magnetic Resonance Imageing (fMRI)*, dan *optical imaging*. MEG, PET, fMRI dan *optical imaging* cukup mahal penggunaannya dan masih berdasarkan pada aliran darah sehingga membutuhkan waktu yang lama. Dengan demikian kurang tepat digunakan untuk berkomunikasi (Wolpaw et al., 2002). Gambar 2.1 menunjukkan peralatan yang dapat mengirimkan sinyal atau gelombang listrik dari otak ke komputer yang lebih *portable* dibandingkan peralatan lainnya.



Gambar 2.1 A brain-computer interface (BCI) being demonstrated in 2009, Image credit: rubra. (Sumber: Read more at <http://endthelie.com/2012/08/17/hacking-the-human-brain-researchers-demonstrate-extraction-of-sensitive-data-via-brain-computer-interface/#WXYFRUrMmt2s0kcq.99>)

Sistem BCI seperti terlihat pada Gambar 2.2 dimana sinyal yang dihasilkan dari aktivitas otak akan diproses secara digitalis, kemudian dilakukan pemrosesan sinyal untuk mendapatkan fitur penting dan diterjemahkan ke dalam algoritma untuk membuat perintah pada *device*. Gambar 2.3 menunjukkan peralatan sederhana dan *portable* untuk merekam EEG terdiri dari *amplifier unit*, *electrode cup*, *conductive jelly*, *injection* dan *aid for disinfection* (Teplan, 2002). Posisi elektroda pada permukaan kepala mengikuti standar internasional yaitu sistem 10 – 10 ataupun 10 – 20 yang menggunakan bentuk anatomi dari kepala seperti terlihat pada Gambar 2.4. Lokasi penempatan elektroda dibagi berdasarkan interval 10% sampai 20% dan menunjukkan tempat dimana elektroda akan di tempatkan. Minimal 21 elektroda yang direkomendasikan untuk studi klinis, meskipun EEG digital sekarang memiliki kemampuan untuk jumlah yang besar.



Gambar 2.2 Desain dasar dan operasi dari Sistem BCI.

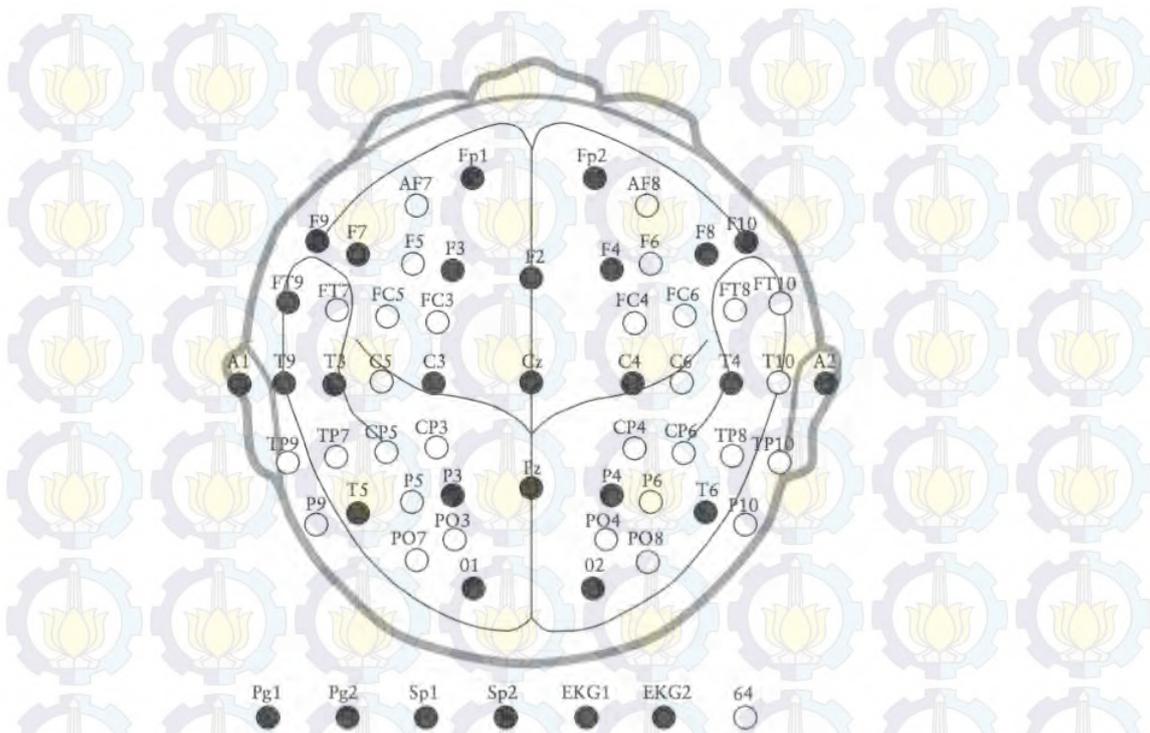


Gambar 2.3 Peralatan untuk merekam EEG: *amplifier unit, electrode cap, conductive jelly, injection, and aid for disinfection.*

Penggunaan jumlah elektroda tergantung pada usia dan ukuran dari kepala. Sebuah modifikasi sistem elektroda kombinasi baru menggunakan penempatan elektroda yang mendekati sistem 10 – 10 (Gambar 2.4). Penandaan elektroda, Fp (frontopolar), F (frontal), T (temporal), O (occipital), C (central), and P (parietal) digunakan pada sistem 10 – 20. Selanjutnya angka gabungan yang mengikuti huruf dari lokasi menunjukkan posisi sebelah kiri (odd numbers) dan kanan (even numbers) sisi penempatan elektroda. Huruf “z” menunjukkan posisi bagian tengah (misl., Cz = Central middle). Di dalam sistem 10 – 10 angka rendah menunjukkan posisinya yang dekat dengan bagian tengah, dan T3/T4 menjadi T7/T8 sedangkan T5/T6 menjadi P7/P8 (William O. Tatum et al., 2008).

2.2 Electroencephalogram (EEG)

Electroencephalogram (EEG) merupakan pengukuran yang unik dan berharga dari fungsi gelombang listrik pada otak dan ditampilkan dalam bentuk grafik dengan tegangan yang berbeda dari dua lokasi fungsi otak dan direkam setiap saat. Sedangkan *Electroencephalography* (EEG) melibatkan studi tentang bagaimana melakukan perekaman sinyal listrik yang dihasilkan oleh otak (William O. Tatum et al., 2008).



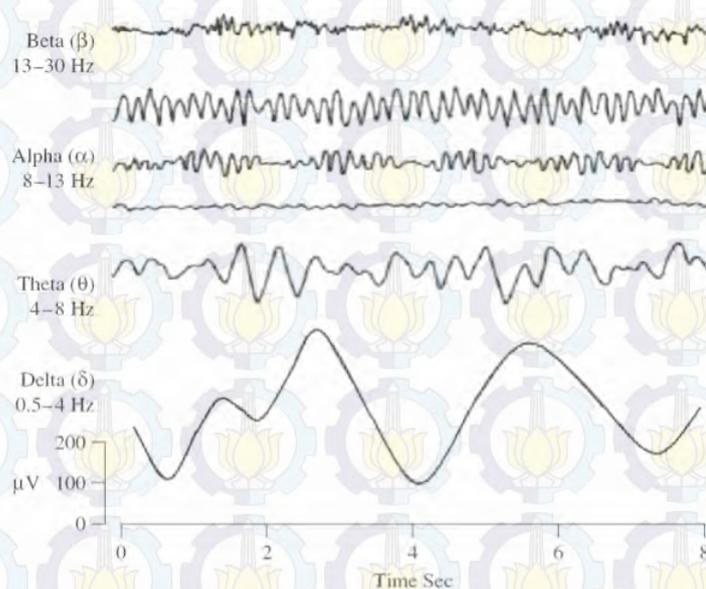
Gambar 2.4 Sistem peletakan Elektroda menggunakan 10-20 sistem (lingkaran hitam) atau modifikasi system kombinasi dengan 10-10 peletakan elektroda (lingkaran hitam + lingkaran putih) (Sumber: *Handbook of EEG Interpretation*).

Richard Caton¹ tahun 1875 menemukan adanya arus listrik di otak dengan mengamati EEG dari otak kelinci dan monyet. Penelitian ini menjadi dasar bagi Hans Berger² tahun 1942 yang memperkuat aktivitas listrik otak yang diukur pada kulit kepala manusia dengan menggunakan peralatan radio biasa. Berger meletakkan dasar bagi banyak aplikasi sekarang *electroencephalography* dia juga menggunakan kata *Electroencephalogram* sebagai yang pertama untuk menggambarkan potensi listrik otak pada manusia. Tahun 1934 Adrian Matthews mempublikasikan jurnal untuk melakukan verifikasi konsep dari “*human brains waves*” dan mengidentifikasi *oscillations* biasa sekitar 10 sampai 12 Hz yang mereka sebut “*alpha rhythm*” (irama alpha) (Teplan, 2002). Gelombang otak di kategorikan menjadi empat group yaitu gelombang δ (1 – 3 Hz) atau δ (0.5 – 4), θ

¹ Richard Caton, berasal dari Liverpool, Inggris dan merupakan seorang ilmuwan dan dokter. Lahir 1842 dan meninggal tahun 1926. Tahun 1987, Caton dilaporkan British Medical Association di Edinburgh bahwa ia telah menggunakan galvanometer untuk mengamati impuls listrik dari permukaan otak yang tinggal di subject hewan (http://en.wikipedia.org/wiki/Richard_Caton).

² Hans Berger, lahir 21 Mei 1887 dan meninggal 1 Juni 1941 dari Universitas of Jena merupakan seorang ahli Neurologis, kebangsaan Jerman (http://en.wikipedia.org/wiki/Hans_Berger).

(4 – 7 Hz) atau θ (4 – 8), α (8 – 13 Hz) dan β (14 – 30 Hz) atau β (13 – 30 Hz) seperti terlihat pada Gambar 2.5. Untuk EEG khusus mental task biasanya menggunakan 8 elektroda untuk sistem standar internasional 10 – 20 yang terdiri dari C3, Cz, C4, CP1, CP2, P3, Pz dan P4 dimana kisaran dari frekuensi EEG adalah 1 – 30 Hz, namun demikian terdapat juga frekuensi yang lebih besar dari 30 Hz yaitu gelombang gamma. Penelitian dan aplikasi klinis dari EEG pada manusia dan hewan dapat digunakan untuk monitoring kewaspadaan koma dan kematian otak, menemukan daerah kerusakan setelah cedera kepala, stroke dan tumor, jalur tes aferen, memantau keterlibatan kognitif, menghasilkan situasi biofeedback, kedalaman control anestesi, menyelidiki epilepsi dan menemukan asal kejang, efek obat uji epilepsy, memantau perkembangan otak manusia dan hewan, obat uji untuk kejang dan menyelidiki gangguan tidur dan fisiologi. Dalam proses perekaman EEG biasanya dapat terjadi *noise* ataupun *artifek* yang kebanyakan sumber *artifek* EEG dapat diklasifikasikan kedalam beberapa cara dimana yang berhubungan dengan pasien terdiri dari (1) segala macam pergerakan kecil dari badan; (2) EMG; (3) ECG (pulse, pace-maker); (4) pergerakan mata dan (5) berkeringat. Dan secara teknis terdiri dari (1) frekuensi 50/60 Hz; (2) fluktuasi impedensi; (3) gerakan kabel; (4) kontak kawat yang rusak dan (5) terlalu banyak pasta (jeli) elektroda atau kering dan (6) baterai lemah (Teplan, 2002).



Gambar 2.5 Empat gelombang EEG yang memiliki kisaran dari 0.5 – 30 Hz

2.3 Ekstraksi Fitur EEG

Dalam rangka untuk memilih *classifier* yang paling tepat untuk sistem BCI, maka penting untuk memahami dengan jelas apa fitur yang digunakan, sifat – sifat dari sinyal dan bagaimana mereka digunakan. Bagian berikut bertujuan untuk menggambarkan fitur BCI yang paling umum dan lebih khusus sifat mereka serta cara menggunakannya.

2.3.1 Properti Fitur

Beberapa fitur yang telah banyak digunakan untuk merancang BCI seperti Nilai Amplitudo dari sinyal EEG, *Band Powers* (BP), nilai *Power Spectral Density* (PSD) (Unde and Shriram, 2014; Lin and Hsieh, 2009), *AutoRegressive* (AR), *parameter Adaptive AutoRegressive* (AAR), fitur *Time-Frequency* dan fitur *Inverse Model-based*. Untuk merancang sistem BCI, beberapa sifat penting dari fitur harus di pertimbangkan (Lotte et al., 2007), yaitu

1. *Noise dan Outliers*: fitur BCI banyak mengandung *noise* dan *outliers* karena sinyal EEG memiliki rasio yang rendah terhadap gangguan sinyal.
2. *High dimensionality*: di dalam sistem BCI vector fitur seringkali berupa dimensi yang besar bahkan beberapa fitur secara umum merupakan ekstraksi dari beberapa *channel* dan segmen waktu sebelum dilakukan proses penggabungan kedalam satu vektor tunggal.
3. *Time information*: fitur BCI harus memiliki informasi waktu sebagai pola aktivitas otak yang secara umum berhubungan dengan variasi waktu tertentu dari EEG.
4. *Non-stationarity*: fitur BCI merupakan *non-stasioner* karena sinyal EEG dapat cepat bervariasi dari waktu ke waktu dan lebih khususnya selama sesi perekaman.
5. *Small training sets*: pengaturan pelatihan relatif kecil, karena proses pelatihan memakan waktu yang lama dan merupakan tuntutan dari subjek.

Properti tersebut telah di verifikasi sebagai fitur yang saat ini banyak digunakan dalam penelitian yang berhubungan dengan BCI.

2.3.2 Pertimbangan Variasi Waktu EEG

Kebanyakan pola aktivitas otak digunakan untuk menggerakkan BCI terkait dengan variasi waktu tertentu dari EEG yang mungkin dalam *band frequency* tertentu. Oleh karena itu, perjalanan waktu sinyal EEG harus diperhitungkan selama ekstraksi fitur. Untuk menggunakan informasi sementara ini, tiga pendekatan telah diusulkan (Lotte et al., 2007) yaitu:

1. *Concatination of features from different time segments*: proses ekstraksi fitur dari beberapa waktu segmen dan digabungkan kedalam satu vektor fitur.
2. *Combination of classifications at different time segments*: melakukan ekstraksi fitur dan proses klasifikasi pada beberapa segmen waktu dan kemudian menggabungkan hasil pengklasifikasi yang berbeda.
3. *Dynamic classification*: ekstraksi fitur dari beberapa waktu segmen untuk membangun urutan sementara dari fitur vektor. Urutan ini dapat diklasifikasikan menggunakan dinamis *classifier*.

Dari ketiganya, pendekatan pertama paling banyak digunakan dalam proses klasifikasi pada sinyal EEG.

2.4 Metode Fitur Ekstraksi

Memahami model fitur ekstraksi akan memberikan kemudahan dalam menentukan metode yang digunakan pada saat melakukan fitur ekstraksi. Metode fitur ekstraksi sangatlah banyak dan bervariasi baik metode standar maupun metode *hybrid* diantaranya *Hilber-huang Transform (HHT)* yang di rancang oleh Huang tahun 1998 untuk menghitung nilai FFT dari input mengganti koefisien FFT yang sesuai dengan frekuensi negatif dengan nilai nol dan menghitung *inverse* FFT dari hasilnya. *Principle Component Analysis (PCA)* yang diperkenalkan pertama kali oleh Karl Pearson tahun 1901, merupakan teknik *non-parametric* didalam mengekstraksi informasi dari data dan mengungkapkan hal itu dalam kesamaan dan perbedaannya. *Independent Component Analysis (ICA)*, metode yang digunakan untuk masalah *blind source separation (BSS)*. ICA diperkenalkan oleh Comon di tahun 1994, *Local Discriminant Bases (LDB)* secara

teknis diperkenalkan oleh Saito dan Coifman tahun 1994 yang masih menggunakan sepenuhnya *Wavelet Packet Decomposition* (WPD) dan lainnya. Penelitian ini menggunakan metode Welch's Periodogram untuk mengestimasi *Power Spectral Density* (PSD) dan *Principle Component Analysis* (PCA) untuk mereduksi dimensi fitur.

2.4.1 Metode Welch's Periodogram

Peter D. Welch menjelaskan metode penggunaan algoritma *fast Fourier Transform* untuk mengestimasi kekuatan spectrum yang melibatkan perekaman bagian untuk mengambil perubahan periodograms dari bagian tersebut dan merata-ratakan perubahan periodograms (Welch, 1967). Periodogram digunakan untuk mengestimasi kekuatan spectrum dengan berusaha menemukan waktu *Diskrete Fourier Transform* dari contoh proses dan mengambil besarnya kuadrat dari hasil (MathWorks, 1998 – 2000) dengan kata lain dapat digunakan untuk mengestimasi atau mengukur *Power Spectral Density* (PSD) yang merupakan salah satu fitur dari sinyal. PSD mendeskripsikan seberapa besar kekuatan sinyal atau *time series* yang di distribusikan dengan frekuensi dan menunjukkan kekuatan variasi sebagai fungsi dari frekuensi (Unde and Shriram, 2014). Estimasi periodogram dari PSD dengan panjang L sinyal $x_L[n]$ adalah

$$\hat{P}_{xx}(f) = \frac{|X_L(f)|^2}{f_s L}, \quad (2.1)$$

dimana

$$X_L(f) = \sum_{n=0}^{L-1} X_L[n] e^{-\frac{2\pi jfn}{f}}. \quad (2.2)$$

Perhitungan sebenarnya $X_L(n)$ dapat dilakukan hanya pada jumlah frekuensi point N yang terbatas dan biasanya menggunakan FFT. Dalam prakteknya sebagian besar implementasi dari metode periodogram menghitung estimasi PSD untuk N -Point.

$$\hat{P}_{xx}[f_k] = \frac{|X_L[f_k]|^2}{f_s L}, \quad f_k = \frac{kf_s}{N}, \quad k = 0, 1, \dots, N \quad (2.3)$$

dimana

$$X_L[f_k] = \sum_{n=0}^{N-1} X_L[n] e^{-\frac{2\pi jkn}{N}}. \quad (2.4)$$

Adalah lebih baik memilih $N > L$ maka N dua kali lebih besar dari L . Untuk mengevaluasi $X_L[f_k]$ secara sederhana digunakan $x_L[n]$ dengan nilai 0 pada panjang N . Jika $L > N$ maka harus diringkaskan dahulu $x_L[n]$ modulo- N sebelum dihitung $X_L[f_k]$.

Beberapa metode diusulkan untuk peningkatan estimasi periodogram diantaranya *Bartlett method* (averaging periodogram), *Blackman – Turkey method* (periodogram smoothing) dan *Welch method* (averaging modified periodogram). Metode Welch menggunakan urutan yang lebih pendek dari rekaman sehingga memiliki keuntungan ketika proses perhitungan dilakukan pada mesin dengan penyimpanan yang terbatas. Estimasi PSD Welch membagi data *time series* kedalam segmen (mungkin tumpang tindih atau overlapping), menghitung perubahan periodogram (*modified periodogram*) pada setiap segmen, kemudian hitung rata – rata perkiraan PSD (*averaging periodogram*) (Welch, 1967). Metode *Estimation Welch* (Welch, 1967) dapat diuraikan sebagai berikut misalkan $X(j), j = 0, \dots, N - 1$, Contoh dari stationary second-order schocastic. Assumsikan $E(X) = 0$ dan $X(j)$ merupakan spectral density $P(f), |f| \leq \frac{1}{2}$. Gunakan segmen yang mungkin bertumpuk dengan panjang L dan titik awal dari segmen D dari bagian segmen. Misl, $X_1(j), j = 0, \dots, L - 1$ menjadi segmen pertama kemudian

$$X_1(j) = X(j), \quad j = 0, \dots, L - 1$$

demikian pula

$$X_2(j) = X(j + D), \quad j = 0, \dots, L - 1$$

dan akhirnya

$$X_k(j) = X(j + (K - 1)D), \quad j = 0, \dots, L - 1$$

Misalkan segmen yang ada $X_1(j), \dots, X_k(j)$ dan mewakili seluruh *record* yaitu $(K - 1) D + L = N$ seperti terlihat pada Gambar 2.6. Metode estimasi yang diusulkan sebagai berikut untuk setiap segmen dengan panjang L akan dihitung

modified periodogram. Oleh karena itu, dipilih sebuah data window $W(j)$, $j = 0, \dots, L - 1$, dan bentuk urutan (*sequence*) $X_1(j)W(j), \dots, X_k(j)W(j)$. Kemudian ditentukan *finite Fourier transforms* $A_1(n), \dots, A_k(n)$ dari urutan tersebut dengan

$$A_k(n) = \frac{1}{L} \sum_{j=0}^{L-1} X_k(j)W(j)e^{-\frac{2kijn}{L}}, \quad (2.5)$$

dan $i = (-1)^{\frac{1}{2}}$. Akhirnya diperoleh K *modified periodograms*

$$I_k(f_n) = \frac{L}{U} |A_k(n)|^2, \quad k = 1, 2, \dots, k \quad (2.6)$$

dimana

$$f_n = \frac{n}{L}, \quad n = 0, \dots, \frac{L}{2} \quad (2.7)$$

dan

$$U = \frac{1}{L} \sum_{j=0}^{L-1} W^2(j). \quad (2.8)$$

Estimasi *spectral* adalah rata – rata dari *periodogram* tersebut:

$$\hat{P}(f_n) = \frac{1}{K} \sum_{k=1}^K I_k(f_n), \quad (2.9)$$

Sekarang dapat ditunjukkan bahwa

$$E\{\hat{P}(f_n)\} = \int_{-\frac{1}{2}}^{\frac{1}{2}} h(f)P(f - f_n)df, \quad (2.10)$$

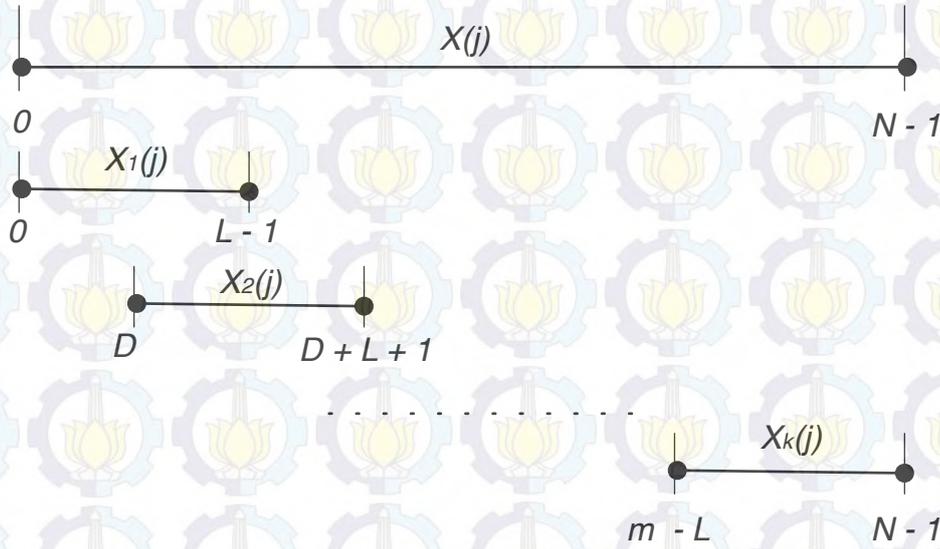
dimana

$$h(f) = \frac{1}{LU} \left| \sum_{j=0}^{L-1} W(j)e^{2\pi ifj} \right|^2, \quad (2.11)$$

dan

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} h(f)df = 1. \quad (2.12)$$

Oleh karena itu estimator spectral $P(f)$ dengan jendela spectral yang luasnya merupakan bidang kesatuan dan lebarnya dari urutan $\frac{1}{L}$.



Gambar 2.6 Ilustrasi proses segmen

2.4.2 Metode Principle Component Analysis

Principle Component Analysis (PCA) pada dasarnya merupakan metode yang banyak digunakan untuk melakukan reduksi atau pengurangan vektor fitur yang berdimensi tinggi menjadi k -dimensional yang lebih rendah hal ini bertujuan untuk melakukan efisiensi dari proses komputasional. Metode ini diperkenalkan oleh Prof. Karl Pearson (Pearson, 1901) dimana ide dibalik prinsip komponen adalah arah dari *principle component* merupakan varians maksimum data. Secara matematis dengan menghitung eigen vektor matriks kovarians dari input asli PCA mengubah data dimensi tinggi ke ruang dimensi yang lebih rendah dan tidak berkorelasi dan ortogonal. Pertama menghitung nilai eigen dari fitur matriks kemudian mengurutkan nilai eigen dan mengabaikan nilai yang benar – benar kecil. Selanjutnya mengubah data ke dalam ruang eigen yang dibentuk oleh eigenvektor yang dipilih.

2.5 Klasifikasi Algoritma

Klasifikasi merupakan proses untuk memetakan suatu objek kedalam kelompok atau group yang memiliki sifat sama berdasarkan pada fitur yang dimiliki oleh suatu objek. Untuk memilih *classifier* yang paling tepat perlu diketahui dengan benar sifat – sifat pengklasifikasian yang tersedia.

2.5.1 Klasifikasi Taksonomi

Beberapa istilah yang sering digunakan untuk menggambarkan berbagai jenis pengklasifikasian (1) *Generative* yang digunakan untuk mempelajari model kelas dan *discriminative* mempelajari cara membedakan kelas atau keanggotaan kelas; (2) *Static* dimana klasifikasi berdasarkan fitur vektor tunggal dan *Dynamics*, mengklasifikasikan berdasarkan urutan fitur vektor dan menangkap dinamika sementara; (3) *Stable* yang memiliki kompleksitas rendah dan *unstable* memiliki kompleksitas yang tinggi dimana perubahan kecil dari data training dapat menyebabkan perubahan penting dalam prosesnya; (4) *Regularized* dimana secara berhati – hati mengendalikan kompleksitas *classifier* dalam rangka mencegah overtraining (Lotte et al., 2007).

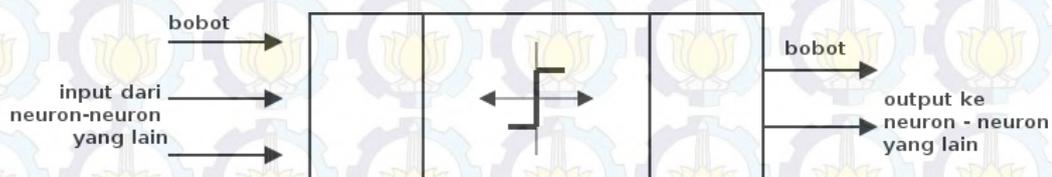
2.5.2 Masalah Utama Klasifikasi dalam Penelitian BCI

Dalam proses pengenalan tugas pengklasifikasian mungkin saja menghadapi beberapa masalah yang berkaitan dengan sifat fitur seperti *outlier*, *overtraining* dll. Dalam *brain-computer interface (BCI)* terdapat dua masalah yang perlu di garis bawahi yaitu masalah *curse-of-dimensionality* dimana jumlah data yang diperlukan untuk menentukan keakuratan kelas yang berbeda meningkat secara eksponensial dengan dimensi fitur vektor. Jika jumlah data pelatihan sedikit dibandingkan dengan ukuran fitur vektor *classifier* akan memberikan hasil yang buruk. Disarankan untuk menggunakan ukuran setidaknya 5 – 10 kali lebih banyak contoh data pelatihan perkelas. Sayangnya hal ini tidak dapat diterapkan dalam semua sistem BCI secara umum yaitu dimensi yang tinggi dan data pelatihan yang kecil. Hal tersebut menjadi perhatian utama dalam desain. Masalah kedua adalah *The Bias-variance tradeoff* dimana terdapat tiga sumber kesalahan klasifikasi yaitu *noise*, *bias* dan *variance* (Lotte et al., 2007).

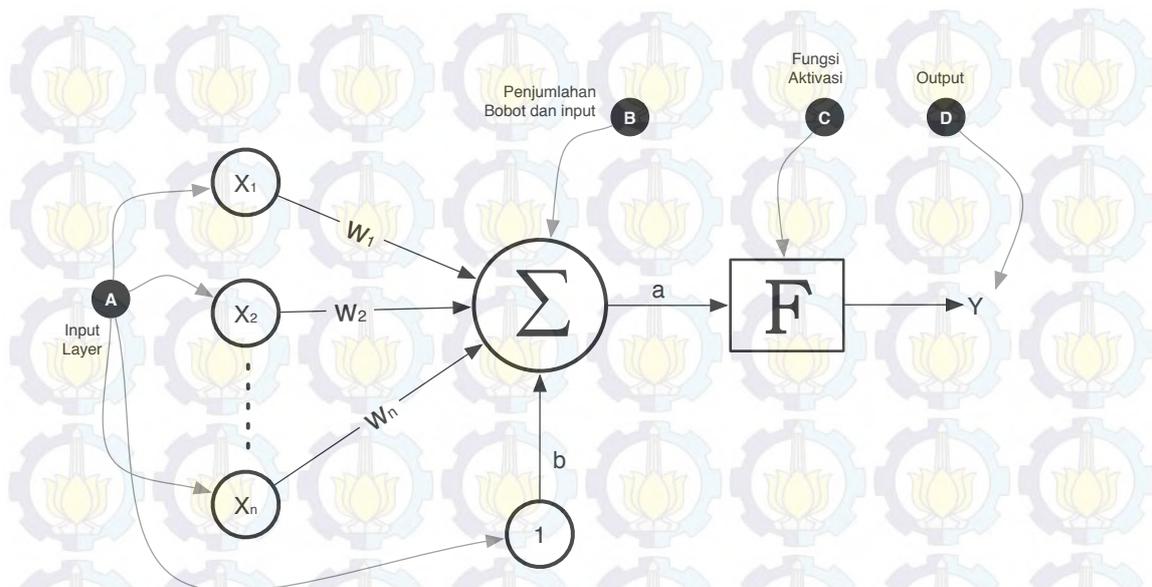
2.5.3 Metode Neural Network

Terdapat beberapa metode klasifikasi yang digunakan pada *brain-computer interface (BCI)*, diantaranya *linear classifiers*, merupakan algoritma diskriminan yang menggunakan fungsi liner untuk membedakan kelas (misl. *Linear Discriminant Analysis (LDA)* dan *Support Vector Machine (SVM)*), *Neural Network*, di susun dari beberapa neuron yang memungkinkan untuk menghasilkan batasan keputusan nonlinear. Metode yang paling umum digunakan adalah *Multilayer Perceptron*, selain itu terdapat metode pengembangannya misl. *Learning Vector Quantization (LVQ) Neural Network*, *Fuzzy ARTMAP Neural Network*, *Dynamic Neural Network*, *RBF Neural Network*, *Bayesian Logistic Regression Neural Network*, *Adaptive Logic Network (ALN)* dan *Probability estimating Guarded Neural Classifier* (Lotte et al., 2007). Penelitian ini menggunakan *Neural Network Feedforward*.

Konsep dasar dari jaringan saraf tiruan mengikuti konsep sistem saraf otak manusia yang memiliki neuron dan saling berhubungan dalam membentuk informasi. Terdapat beberapa tipe jaringan saraf tiruan namun hampir semuanya memiliki komponen yang sama. Seperti halnya otak manusia jaringan saraf terdiri dari beberapa neuron dan setiap neuron berhubungan dengan neuron lainnya yang akan mengirimkan informasi dan dikenal dengan bobot. Gambar 2.7 menunjukkan struktur dari jaringan saraf, masukan yang dikirim dari neuron akan dihitung dengan suatu fungsi perambatan untuk menjumlahkan semua nilai bobot yang dikirim dan hasilnya akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi pengaktifan setiap neuron. Dalam jaringan saraf, neuron akan di kumpulkan dalam lapisan yang disebut lapisan neuron, biasanya setiap lapisan akan di hubungkan dengan lapisan lain yang tersembunyi yang disebut *hidden layer* ke *output layer*.



Gambar 2.7 Struktur neuron pada jaringan saraf



Gambar 2.8 Fungsi pengaktifan sederhana pada jaringan saraf

Tergantung pada sistem pembelajarannya, informasi dapat dirambatkan secara mundur dalam jaringan. Gambar 2.8 menunjukkan jaringan saraf sederhana dengan fungsi pengaktifan F dimana sebuah neuron akan mengolah data sejumlah N (x_1, x_2, \dots, x_n) yang masing – masing memiliki bobot (w_1, w_2, \dots, w_n) dan ditambah bobot bias b yang dapat dirumuskan sebagai berikut:

$$a = \sum_{i=1}^N x_i w_i + b . \quad (2.13)$$

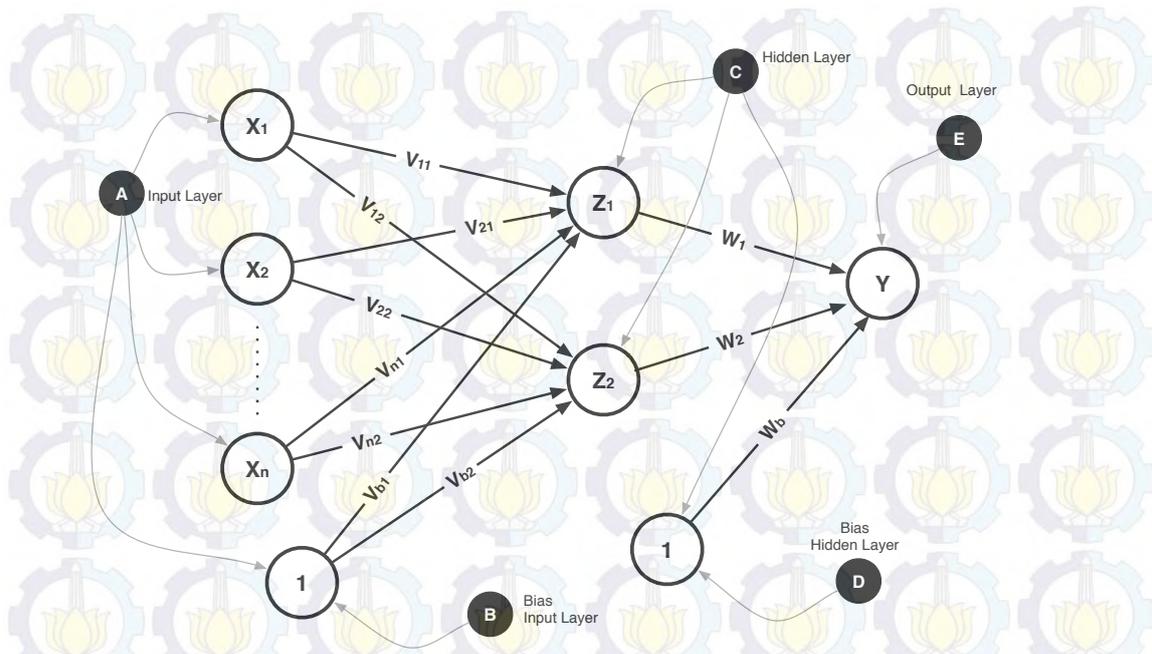
Selanjutnya fungsi pengaktifan F akan mengaktifasi a menjadi output jaringan. Terdapat beberapa fungsi pengaktifan yang sering digunakan dalam jaringan syaraf tiruan yaitu *Fungsi biner, bipolar, linear, saturating linear, symmetric saturating, sigmoid biner* dan *sigmoid bipolar*. *Fungsi biner* dan *sigmoid biner* yang paling sering digunakan. *Sigmoid biner*, merupakan fungsi pengaktifan yang digunakan sebagai pelatihan untuk *metode backpropagation* yang memiliki jangkauan nilai dari 0 sampai 1. Sehingga fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai output berada diantara interval 0 sampai 1. Fungsi *sigmoid biner* dirumuskan sebagai berikut:

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}} , \quad (2.14)$$

dengan turunan pertamanya adalah $f'(x) = \sigma f(x)[1 - f(x)]$ atau

$$f'(x) = \sigma \left(\frac{1}{1 + e^{-\sigma x}} \right) \left[1 - \left(\frac{1}{1 + e^{-\sigma x}} \right) \right]. \quad (2.15)$$

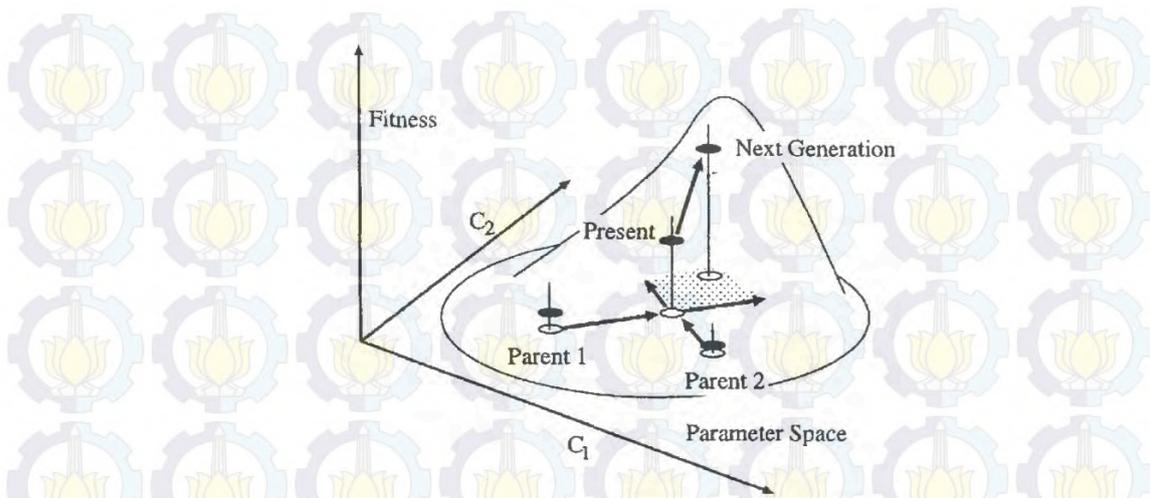
Terdapat beberapa model jaringan saraf tiruan salah satu model yang populer adalah jaringan saraf tiruan model *backpropagation*. *Backpropagation* merupakan pembelajaran *supervise* (terawasi) dengan banyak lapisan dan melakukan perubahan pada bobot. Algoritma *backpropagation* menggunakan kesalahan (*error*) hasil keluaran untuk mengubah nilai bobotnya dalam arah mundur (*backward*), untuk mendapatkan nilai kesalahan (*error*) maka tahap perambatan maju (*forward propagation*) harus di lakukan. Pada saat melakukan perhitungan maju maka neuron diaktifkan dengan menggunakan fungsi pengaktifan *sigmoid biner* ataupun *sigmoid bipolar*. Arsitektur jaringan saraf tiruan untuk *backpropagation* terlihat seperti Gambar 2.9 yang terdiri dari tiga lapisan yaitu *Input layer* (lapisan masukan), *hidden layer* (lapisan tersembunyi) dan *Output layer* (lapisan keluaran). *Input layer* terdiri dari misl (x_1, x_2, \dots, x_n) ditambah dengan satu input lain yaitu *bias* bernilai 1 *bias* dapat diatur dengan bilangan yang sangat kecil ataupun satu. Bobot dari *input layer* ke *hidden layer* adalah $(v_{11}, v_{12}, \dots, v_{n1}, v_{n2})$ ditambah dengan bobot dari bias yaitu (v_{b1}, v_{b2}) kemudian bobot dari *hidden layer* dan *bias* terdiri dari (w_1, w_2, w_b) . *Bacpropagation* dapat menggunakan beberapa *hidden layer* yang di sesuaikan dengan kebutuhan sistem dimana bobot dari jaringan saraf tiruan akan di acak pada proses perambatan maju. Terkadang hal ini dapat menyebabkan proses pembelajaran menjadi lebih lama untuk mencari parameter yang ideal di setiap proses klasifikasinya. Saat ini banyak penelitian yang ditujukan untuk mencari proses pembobotan awal tersebut tanpa harus melalui proses *learning* dari jaringan *bacpropagation*, misalnya penelitian dari Cheng-Jian Lin dan Ming-Hua Hsieh yang menggunakan *particle swarm optimization* untuk melakukan optimasi parameter dari *neural network* dengan meningkatkan proses pencarian parameter global dari kecepatan partikel (Lin and Hsieh, 2009), Sai Ho Ling dkk melakukan proses klasifikasi menggunakan *neural network* untuk mengoptimasi bobot melalui *particle swarm optimization* dengan bobot inersia (Chai et al., 2014).



Gambar 2.9 Arsitektur jaringan backpropagation

2.6 Algoritma Evolusioner

Metode evolusioner pada dasarnya merupakan algoritma yang dikembangkan berdasarkan pada sifat ataupun perilaku makhluk hidup dengan tujuan mengambil konsep evolusi dari berbagai makhluk hidup. Algoritma ini akan mengikuti perilaku makhluk hidup misalnya, semua individu akan hidup dalam suatu populasi tertentu, adanya tujuan yang akan di capai, berusaha untuk mempertahankan hidup dalam suatu populasi dan lain sebagainya. Biasanya perhitungan algoritma evolusioner sering dibandingkan dengan teknik seperti halnya *genetic algorithms*, *evolution strategies*, *evolutionary programming* dan *genetic programming*. Algoritma evolusioner sangat berhasil pada kasus – kasus optimasi dibandingkan dengan metode optimasi lainnya karena prinsip dasar dari algoritma ini melakukan perhitungan dengan membangun populasi dari bilangan acak dan proses penyelesaiannya pun menggunakan pendekatan bilangan acak. Berikut ini akan dijelaskan metode *genetic algorithm* dan metode *particle swarm optimization*.



Gambar 2.10 Skema dari Operator Evolutionary direction

2.6.1 Algoritma Genetik

Algoritma Genetik di perkenalkan pertama kali oleh Holland tahun 1975 dan merupakan metode yang umum digunakan untuk algoritma pencarian dan terinspirasi oleh populasi genetik dalam mencari solusi dari masalah. Algoritma ini juga mengikuti konsep Carles Darwin dengan teori evolusinya dimana individu yang kuat akan bertahan dari populasinya. Algoritma ini diawali dengan penentuan kromosom dari setiap individu dimana terdiri dari *gen* atau *allele* yang akan di proses. *Crossover* atau kawin silang digunakan untuk menentukan generasi berikutnya (*offspring*) yang dapat dilakukan dengan salah satu metode berikut *Single Point Crossover*, *Two Point Crossover*, *Multi-Point Crossover (N-Point crossover)*, *Uniform Crossover*, *Three Parent Crossover*, *Crossover with Reduced Surrogate*, *Suffle Crossover*, *Precendence Preservative Crossover (PPX)*, *Ordered Crossover* dan *Partially Matched Crossover (PMX)* (S.N.Sivanandam and S.D.Deepa). Untuk mengontrol kecepatan *crossover* maka digunakan parameter *probabilitas crossover* (P_c) dimana nilainya $0.6 \leq P_c \leq 0.9$ penggunaan P_c bertujuan untuk memberikan nilai konvergensi yang tepat. Proses *mutation* (mutasi) digunakan untuk mendapatkan generasi baru atau *offspring* dengan cara menggantikan *gen* dari sebuah kromosom untuk mendapatkan keturunan yang baru, penggunaan *probabilitas mutation* (P_m) yang terlalu besar mengakibatkan sifat dari individu awal akan hilang oleh karena itu nilainya diberikan berkisar $0.01 \leq P_m \leq 0.1$. Fitness digunakan untuk menentukan generasi

yang terbaik sedangkan operator *crossover* dan *mutation* bertujuan untuk melakukan proses regenerasi terhadap suatu individu akan tetapi tidak memiliki kemampuan dalam menentukan arah evolusinya, metode *Gradient-based Optimizer* digunakan untuk menentukan hal tersebut akan tetapi membutuhkan waktu komputasi yang tinggi karena harus mengevaluasi gradient oleh karena itu diusulkan suatu operator baru pada algoritma genetik untuk menentukan arah evolusi yaitu *evolutionary direction operator* dengan cara yang lebih sederhana (Yamamoto and Inoue, 1995). Gambar 2.10 menunjukkan skematik dari operator arah evolusi dimana algoritma ini mendefinisikan populasi untuk memilih individu secara acak (*random*). Kemudian mengevaluasi kromosom dan menentukan nilai fitness dari setiap individu yang bergantung pada fungsi fitness. Jika hasil evaluasi tidak terpenuhi maka proses evolusi dilakukan. Evolusi dari kromosom sekarang dan kromosom berikutnya diasumsikan berdasarkan kromosom dan fitness dari kedua orang tua sekarang. Gambar 2.11 merupakan prosedur optimasi. Misalkan, kromosom n integer non-negatif C_p ($p = 1, \dots, n$):

$$\text{Chromosome} : \{ C_1, C_2, \dots, C_n \}$$

Setiap integer akan menggambar parameter yang di rancang. Nilai dari C_p di assumsikan memiliki jarak dari 0 sampai C_{MAX} :

$$C_p \in [0, C_{MAX}], \quad p = 1, 2, \dots, n$$

Gambar 2.10, menunjukkan skematik dari evolusi individu yang ditentukan berdasarkan operator *evolutionary direction* dalam permasalahan optimasi dua dimensi. Koordinat C_1 dan C_2 adalah parameter yang dirancang untuk dioptimasi. Koordinat ketiga adalah fitness. Dengan menggunakan operator *evolutionary direction*, *offspring* dapat ditentukan berdasarkan kromosom dan *fitness* sekarang dan parents seperti berikut ini,

Inputs:

$$\text{Parent 1: Chromosome: } \{ CP1_1, CP1_2, \dots, CP1_n \}$$

$$\text{Fitness: } F_{p1}$$

$$\text{Parent 2: Chromosome: } \{ CP2_1, CP2_2, \dots, CP2_n \}$$

$$\text{Fitness: } F_{p2}$$

$$\text{Present: Chromosome: } \{ C_1, C_2, \dots, C_n \}$$

Fitness: F

Output:

Child: Chromosome: {CC₁, CC₂, ..., CC_n}

dalam ekspresi sebelumnya:

$$CC_p = \max[\min(CC_p^*, CMAX), 0], \quad p = 1, 2, \dots, n \quad (2.16)$$

dimana

$$CC_p^* = CC_p + S \cdot \text{sign}(F - F_{p1}) \cdot (CC_p - CP1_p) + T \cdot \text{sign}(F - F_{p2}) \cdot (CC_p - CP2_p) \quad (2.17)$$

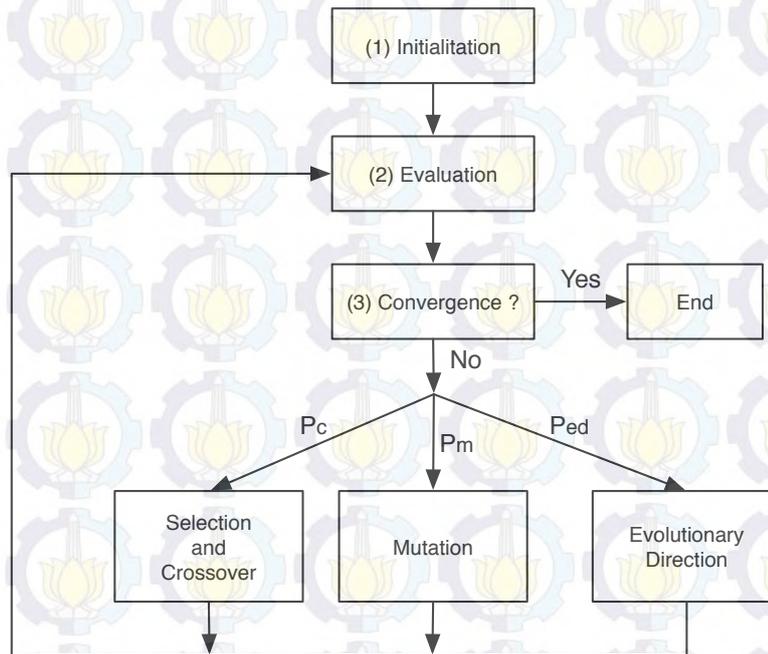
dan

$$S, T \in [0, 1] \quad (2.18)$$

Dari Gambar 2.10, terdapat dua arah evolusi, pertama berasal dari parent 1 menuju ke kondisi sekarang dan lainnya berasal dari parent 2. Evolusi atau arah dari generasi selanjutnya (*offspring*) ditentukan dari dua arah menggunakan persamaan 2.16 dan 2.18 kontribusi arah dari parent 1 terdapat pada persamaan 2.17 bagian tengah sedangkan kontribusi dari parent 2 pada bagian ketiga dari persamaan 2.17. Parameter S dan T merupakan bilangan acak dari 0 sampai 1 untuk memberikan pembesaran (*magnitude*) dari kontribusi dua arah evolusi. Karena seluruh variabel sudah di ketahui maka kromosom dari turunan CC_p dengan mudah dapat ditentukan. Fitness dari Child ditentukan dari kromosom child setelah CC_p ($p=1, \dots, n$) diperoleh. Evaluasi dan evolusi diproses kembali sampai evaluasi mencapai konvergensi.

2.6.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) merupakan salah satu metode yang digunakan untuk mencari optimasi dan lebih baik dari metode algoritma genetik, dimana konsepnya mengadopsi kumpulan perilaku sosial organisme. Penelitian yang berhubungan dengan perilaku sosial organisme menjadi dukungan dalam melakukan perancangan algoritma optimasi yang sangat efisien.



Gambar 2.11 Prosedur Optimasi

Particle swarm optimization (PSO) menggunakan model perilaku sosial dari sekawanan burung yang akan mencari posisi individu dan dikenal dengan partikel dalam suatu kawanan (Kenedy, 1995). Seperti pada algoritma evolutionary lainnya PSO membutuhkan populasi yang merupakan sekawanan organisme yang didalamnya terdapat partikel dengan sejumlah elemen. Misalkan $X(t)$ disimbolkan sebagai sebuah kawanan pada iterasi ke t . Setiap partikel $\mathbf{x}^i(t) \in X(t)$ berisi n elemen $x_j^i(t) \in \mathbf{x}^i(t)$ pada iterasi ke t , dimana $i = 1, 2, \dots, \gamma$ dan $j = 1, 2, \dots, n$, dimana γ merupakan jumlah dari partikel di dalam kawanan dan n adalah dimensi dari partikel. Awalnya partikel dari kawanan didefinisikan dan dievaluasi melalui fungsi fitness (*objective*). Dalam PSO salah satu komponen yang penting adalah kecepatan (*velocity*) dimana berhubungan dengan pergerakan dalam ruang pencarian. Kecepatan (*velocity*) $v_j^i(t)$ dan posisi (*position*) $x_j^i(t)$ dari elemen ke j dan partikel ke i pada iterasi ke t yang dapat dihitung sebagai berikut. Untuk update kecepatan (*velocity*):

$$v_j^i(t) = v_j^i(t-1) + \varphi_1 \cdot r_1 \cdot (p_j^i - x_j^i(t-1)) + \varphi_2 \cdot r_2 \cdot (g_j - x_j^i(t-1)) \quad (2.19)$$

Sedangkan untuk posisi (*Position*):

$$x_j^i = x_j^i(t-1) + v_j^i(t) \quad (2.20)$$

dimana $p_i = [p_1^i, p_2^i, \dots, p_n^i]$ dan $g = [g_1, g_2, \dots, g_n]$ posisi partikel i yang terbaik ditunjukkan dengan p^i dan posisi terbaik dari partikel diantara seluruh partikel ditunjukkan dengan g , r_1 dan r_2 nilai acak diantara $[0,1]$.

2.6.2a IPSO: Constriction and Inertia Weight Factors

Particle Swarm Optimization (PSO) memiliki kecepatan dan posisi yang merupakan bagian penting dari proses optimasi seperti pada persamaan (2.19). Persamaan tersebut memiliki tiga komponen yaitu kecepatan sebelumnya pada bagian pertama, *cognition* atau *individual knowledge* yang merupakan perilaku pribadi dari partikel pada bagian kedua (bagian tengah) dan pada bagian akhir merupakan *social* atau *group knowledge* yang menunjukkan hubungan diantara individu dari kawanan. Persamaan tersebut dapat menyebabkan proses terlalu cepat sehingga nilai konvergensinya belum tercapai untuk mengatasi masalah tersebut diusulkan penambahan nilai "Inertia" ω pada ruas pertama dari persamaan 2.19 yang bertujuan untuk menyeimbangkan antara pencarian global dan lokal. Selain itu penambahan *faktor constriction* yang mungkin sangat penting untuk memastikan konvergensi dari algoritma *particle swarm* (Eberhart, 2011). Persamaan 2.19 dimodifikasi dengan menambahkan nilai bobot inersia (*inertia weight*) dan nilai penyempitan (*constriction*) seperti berikut:

$$v_j^i(t) = k \cdot \left\{ \omega_k(t) \cdot v_j^i(t-1) + \varphi_1 \cdot r_1 \cdot (p_j^i - x_j^i(t-1)) + \varphi_2 \cdot r_2 \cdot (g_j - x_j^i(t-1)) \right\} \quad (2.21)$$

Secara matematis nilai k (*constriction*) adalah fungsi dari φ_1 dan φ_2 yang terlihat seperti pada persamaan berikut:

$$k = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \quad (2.22)$$

dimana $\varphi = \varphi_1 + \varphi_2$ dan $\varphi > 4$ dan ω_k adalah faktor bobot inersia untuk menyediakan keseimbangan antara global dan lokal eksplorasi, φ_1 dan φ_2 adalah konstanta kecepatan, k merupakan faktor *constriction* yang berasal dari persamaan

2.21 untuk memastikan sistem konvergen tapi tidak dalam kondisi prematur (Ling, 2011) sedangkan secara umum ω_k diatur oleh persamaan berikut (Ling, 2012)

$$\omega(t) = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T} \times t, \quad (2.23)$$

dimana t jumlah iterasi yang aktif, T total jumlah iterasi, ω_{\max} dan ω_{\min} batas atas dan bawah dari bobot inersia yang secara normal di atur 1.1 dan 0.1 secara berturut – turut.

2.6.2b IPSO: Fuzzy Inertia Weight

Persamaan 2.21 update kecepatan (*velocity*) dan *inertia weight* $\omega(t)$ digunakan untuk menyediakan keseimbangan antara global eksplorasi dan lokal exploitasi. Dari hubungan linear antara ω dan t pada persamaan 2.23 terlihat bahwa ketika nilai $\frac{t}{T}$ adalah kecil maka global eksplorasi terjadi sebaliknya ketika nilai $\frac{t}{T}$ adalah besar maka lokal exploitasi akan terjadi. Bagaimanapun banyak masalah optimasi yang memiliki sifat nonlinear maka fuzzy bobot inersia $\tilde{\omega}(t)$ diusulkan untuk meningkatkan kemampuan pencarian (Ling, 2012). Nilai $\tilde{\omega}(t)$ akan di evaluasi melalui dua masukan *fuzzy inference system* yang juga akan menjadi keluaran untuk $\beta(t)$. Input dari sistem *fuzzy* adalah $\|\zeta(t)\|$ dan $\frac{t}{T}$. Parameter $\|\zeta(t)\|$ merupakan *normalisasi standar deviasi* diantara seluruh partikel jika nilainya besar mengindikasikan lokasi dari partikel jauh dari lainnya begitupun sebaliknya. Persamaan $\|\zeta(t)\|$ didefinisikan sebagai berikut:

$$\|\zeta(t)\| = \sqrt{\frac{1}{Y} \sum_{i=1}^Y (\|f(\mathbf{x}^i(t))\| - \|\bar{f}(\mathbf{x}^i(t))\|)^2}, \quad (2.24)$$

dimana

$$\|\bar{f}(\mathbf{x}^i(t))\| = \frac{1}{Y} \sum_{i=1}^Y \|f(\mathbf{x}^i(t))\| \quad (2.25)$$

Dengan $\|\cdot\|$ di simbolkan sebagai *vector norm* (panjang vector). Sedangkan bobot inersia *fuzzy* mengikuti aturan berikut:

$$\text{Rule } j: \text{IF } \|\zeta(t)\| \text{ is } N_1^j, \text{ AND } \frac{t}{T} \text{ is } N_2^j, \text{ THEN } \tilde{\omega}(t) = \sigma_j, \quad (2.26)$$

dimana $j = 1, 2, \dots, \varepsilon$, N_1^j dan N_2^j aturan dari rule j , ε disimbolkan dengan jumlah dari aturan, $\sigma_j \in [w_{\min}, w_{\max}]$ untuk kecepatan. Nilai akhir di berikan secara berurutan dari 0.1 sampai 1.1. Sehingga nilai akhir dari parameter $\tilde{\omega}(t)$ diberikan seperti persamaan berikut:

$$\tilde{\omega}(t) = \sum_{j=1}^{\varepsilon} m_j(t) \sigma_j, \quad (2.27)$$

dimana

$$m_j(t) = \frac{\mu_{N_1^j}(\|\zeta(t)\|) \times \mu_{N_2^j}\left(\frac{t}{T}\right)}{\sum_{j=1}^{\varepsilon} \left(\mu_{N_1^j}(\|\zeta(t)\|) \times \mu_{N_2^j}\left(\frac{t}{T}\right) \right)} \quad (2.28)$$

nilai $\mu_{N_1^j}(\|\zeta(t)\|)$ dan $\mu_{N_2^j}\left(\frac{t}{T}\right)$ merupakan nilai fungsi keanggotaan dari nilai N_1^j dan N_2^j . Nilai bobot inersia $\omega(t)$ akan diganti dengan $\tilde{\omega}(t)$ untuk membuat kecepatan (*velocity*) $v_j^i(t)$ menggunakan *fuzzy inertia weight*. Sehingga persamaan 2.21 akan menjadi sebagai berikut:

$$v_j^i(t) = k \cdot \left\{ \tilde{\omega}(t) \cdot v_j^i(t-1) + \varphi_1 \cdot r_1 \cdot (p_j^i - x_j^i(t-1)) + \varphi_2 \cdot r_2 \cdot (g_j - x_j^i(t-1)) \right\}. \quad (2.29)$$

Selain itu Ling dkk (Ling, 2012) mengusulkan penggunaan *Cross-mutated Operation* yang ide dasarnya adalah menggabungkan operator *crossover* dan *mutation* dari algoritma genetik. Tujuan dari *cross-mutated operation* untuk membantu partikel keluar dari optimum lokal dengan menggunakan parameter $\beta(t)$ yang diperoleh dari beberapa fungsi *fuzzy* berdasarkan pengetahuan manusia. Setiap elemen dalam partikel dari kawanan akan memiliki kesempatan untuk mengalami operasi *Cross-mutated* (CM) berdasarkan pada probabilitas CM, $p_{cm} \in [0, 1]$ yang didefinisikan oleh pengguna. Setiap elemen dalam partikel R_{cm}

akan membuat bilangan acak antara 0 dan 1 jika nilainya lebih kecil dari p_{cm} maka operasi CM akan menggantikan posisi elemen tersebut. Pemilihan nilai p_{cm} akan memberi efek terhadap kualitas solusinya. Hasil kecepatan (*velocity*) dari elemen partikel dibawah operasi CM diberikan sebagai berikut:

$$\tilde{v}_j^i(t) = \begin{cases} (1 - \beta(t))v_j^i(t) + \beta(t)\tilde{v}_j^i(t), & r_3 > 0.5 \\ (1 - \beta(t))v_j^i(t) - \beta(t)\tilde{v}_j^i(t), & r_3 \leq 0.5 \end{cases} \quad (2.30)$$

dimana

$$\tilde{v}_j^i(t) = 0.25 \left\{ r_4 \cdot (\rho_{\max_j} - \rho_{\min_j}) + \rho_{\min_j} \right\} \quad (2.31)$$

dimana $v_j^i(t)$ diperoleh dari persamaan 2.29 $\tilde{v}_j^i(t)$ merupakan bilangan acak untuk kecepatan (*velocity*) dari elemen partikel dan nilainya dibuat dari bilangan acak yang dibatasi pada jangkauan 0.25 dari nilai elemen partikel. Ini berarti kecepatan maksimum dan minimum dibatasi pada 0.25 dari jangkauan nilai elemen partikel. Diusulkan nilai $\beta(t)$ yang dapat dihitung berdasarkan aturan fuzzy berikut:

$$\text{Rule } j: \text{IF } \|\zeta(t)\| \text{ is } N_2^j, \text{ THEN } \beta(t) = \chi_j, \quad j = 1, 2, \dots, \varepsilon \quad (2.32)$$

Nilai akhir dari $\beta(t)$ diberikan pada persamaan berikut:

$$\beta(t) = \sum_{j=1}^{\varepsilon} m_j(t)\chi_j, \quad (2.33)$$

dimana nilai m_j diberikan pada persamaan 2.28. Gambar 2.12 merupakan input fuzzy yang berbentuk kurva *Gaussian* dimana (a) fungsi keanggotaan dari *normalized standar deviation*, (b) perbandingan antara t/T dimana t waktu iterasi yang aktif dan T merupakan jumlah iterasi. Derajat keanggotaan dari kurva *Gaussian* dapat dilihat pada persamaan 2.34.

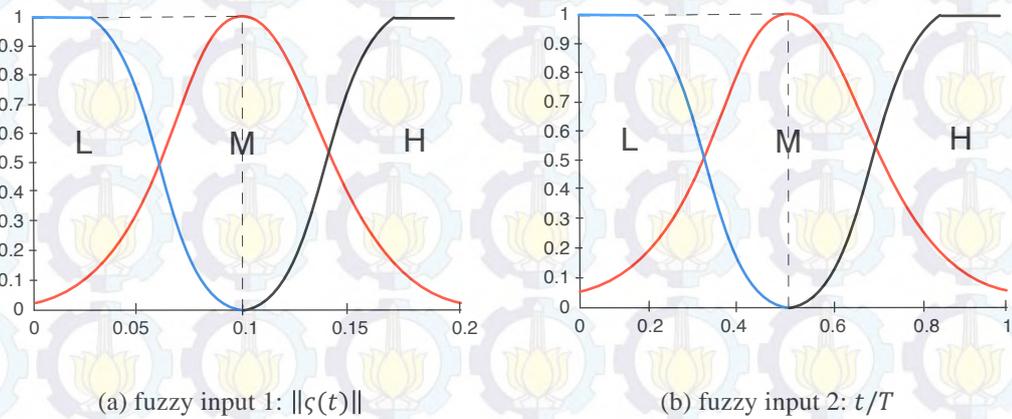
$$G(x; k, \gamma) = e^{-k(\gamma-x)^2}, \quad (2.34)$$

dimana k merupakan lebar dari fungsi atau kurva, γ merupakan titik atau jarak dimana fungsi mencapai nilai maksimum dan x merupakan posisi yang akan dicari (Cox, 1994) seperti Gambar 2.13. Dimana Gambar 2.13 (a) merupakan kurva

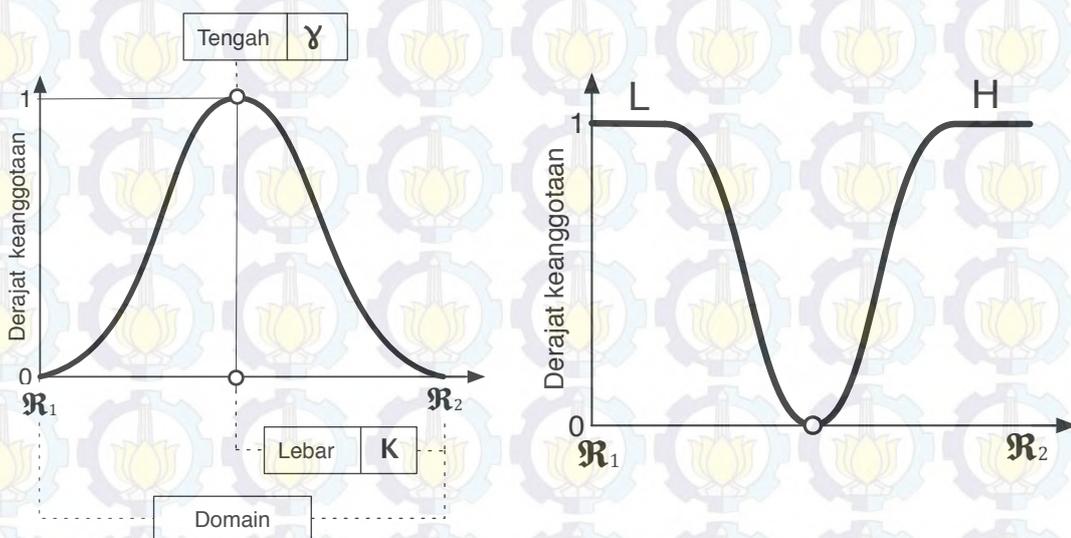
Gaussian dan (b) kurva yang menunjukkan kebalikan dari fungsi keanggotaan dalam hal ini NOT. Derajat keanggotaannya seperti pada persamaan 2.35

$$\overline{G}(x; k, \gamma) = 1 - e^{-k(\gamma-x)^2}. \quad (2.35)$$

Selain itu terdapat lima *linguistic value* yang digunakan sebagai *Output Singleton Variable* seperti terlihat pada Gambar 2.14. *Output singleton variable* digunakan sebagai output fuzzy karena aturan (*rule*) yang dibuat memiliki *consequent* dengan nilai konstanta.



Gambar 2.12. Fungsi keanggotaan dari (a) x-axis: $\|\zeta(t)\|$, y-axis: $\mu_{N_1} \|\zeta(t)\|$, (b) x-axis: t/T , y-axis: $\mu_{N_2} t/T$



Gambar 2.13. Kurva Gaussian (left)(COX, 1994), NOT (right) (LILLY,2010)

Fuzzification

Fuzzification digunakan untuk mengambil input *non-fuzzy (crisp)* dan menentukan derajat keanggotannya untuk nilai posisi dari himpunan *fuzzy* (Ling, 2011). Berdasarkan pada *Standar Operator Fuzzy* terdapat tiga operator dasar yaitu *Complement*, *Intersection* dan *Union* yang dapat dilihat pada persamaan berikut: (Kir, 1995; Lilly, 2010)

$$\overline{A(x)} = 1 - A(x), \quad (2.36)$$

$$(A \cap B)(x) = \min[A(x), B(x)], \quad (2.37)$$

$$(A \cup B)(x) = \max [A(x), B(x)] \quad (2.38)$$

Penelitian ini menggunakan dua input fuzzy yaitu *Normalisasi Standar Deviasi* $\|\zeta(t)\|$ dan *posisi iterasi* $\frac{t}{T}$ dari partikel. Berdasarkan Gambar 2.12a maka derajat keanggotannya terlihat pada persamaan 2.39 – 2.41 dengan parameter $k = 0.1$ dan $\gamma = 0.1$

$$G_{Low} (\|\zeta(t)\|) = 1 - e^{-0.1 (0.1-x)^2} \quad \text{jika, } 0 < x < 0.1 \quad (2.39)$$

$$G_{Medium} (\|\zeta(t)\|) = e^{-0.1 (0.1-x)^2} \quad (2.40)$$

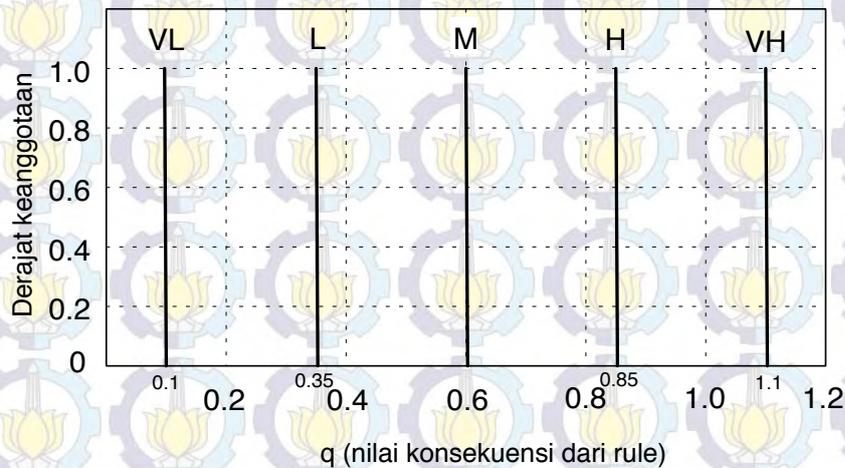
$$G_{High} (\|\zeta(t)\|) = 1 - e^{-0.1 (0.1-x)^2} \quad \text{jika, } 0.1 < x < 0.2 \quad (2.41)$$

Dengan cara yang sama derajat keanggotaan dari *fuzzy* input t/T berdasarkan Gambar 2.12b dapat dilihat pada persamaan 2.42 – 2.44 dengan $k = \gamma = 0.5$

$$G_{Low} (t/T) = 1 - e^{-\frac{1}{2} (0.5-x)^2}, \quad \text{jika, } 0 < x < 0.5 \quad (2.42)$$

$$G_{Medium} (t/T) = e^{-\frac{1}{2} (0.5-x)^2}, \quad (2.43)$$

$$G_{High} (t/T) = 1 - e^{-\frac{1}{2} (0.5-x)^2}. \quad \text{jika, } 0.5 < x < 1.0 \quad (2.44)$$



Gambar 2.14. Output Singleton Fuzzy untuk *Bobot Inersia Adaptif* ($\tilde{\omega}(t)$)

Fuzzy Inference System (Pembentukan Aturan Fuzzy)

Berdasarkan Ling dkk (Ling, 2012) penelitian ini menggunakan 3 fungsi keanggotaan untuk setiap input dengan nama L (Low), M (Medium) dan H (High). Sedangkan untuk terminologi output diberi nama VL (Very Low), L (Low), M (Medium), H (high), dan VH (Very High) seperti Gambar 2.14. Dari karakteristik $\|\zeta(t)\|$ dan $\frac{t}{T}$ maka terdapat 9 *Linguistic value* IF-THEN aturan *fuzzy* untuk menentukan $\tilde{\omega}(t)$ seperti berikut ini:

RULE 1: IF $\|\zeta(t)\|$ is “L” AND $\frac{t}{T}$ is “L” THEN $\tilde{\omega}(t)$ is “VH” (=1.1)

RULE 2: IF $\|\zeta(t)\|$ is “M” AND $\frac{t}{T}$ is “L” THEN $\tilde{\omega}(t)$ is “H” (=0.85)

RULE 3: IF $\|\zeta(t)\|$ is “H” AND $\frac{t}{T}$ is “L” THEN $\tilde{\omega}(t)$ is “VH” (=1.1)

RULE 4: IF $\|\zeta(t)\|$ is “L” AND $\frac{t}{T}$ is “M” THEN $\tilde{\omega}(t)$ is “VH” (=0.6)

RULE 5: IF $\|\zeta(t)\|$ is “M” AND $\frac{t}{T}$ is “M” THEN $\tilde{\omega}(t)$ is “M” (=0.6)

RULE 6: IF $\|\zeta(t)\|$ is “H” AND $\frac{t}{T}$ is “M” THEN $\tilde{\omega}(t)$ is “H” (=0.85)

RULE 7: IF $\|\zeta(t)\|$ is “L” AND $\frac{t}{T}$ is “H” THEN $\tilde{\omega}(t)$ is “VL” (=0.1)

RULE 8: IF $\|\zeta(t)\|$ is “M” AND $\frac{t}{T}$ is “H” THEN $\tilde{\omega}(t)$ is “VL” (=0.1)

RULE 9: IF $\|\zeta(t)\|$ is “H” AND $\frac{t}{T}$ is “H” THEN $\tilde{\omega}(t)$ is “L” (=0.35)

Aturan tersebut dibentuk berdasarkan penurunan nilai dari 1.1 ke 0.1 atau menurut Shi dan Eberhert, *inertia weight* menurun dari 0.9 ke 0.4 (Eberhert, 2000). Aturan *fuzzy* dapat diturunkan berdasarkan pada tiga fakta berikut (Yadmellat, 2009); (1) Ketika algoritma dimulai untuk meningkatkan pencarian dalam ruang populasi maka bobot inersia awal haruslah memiliki nilai yang besar untuk dapat melakukan pencarian perilaku diantara kawanan. Selain itu penurunan dari nilai bobot inersia pada langkah akhir dapat secara langsung mengarahkan kawanan untuk berkumpul sehingga pencarian global optima lebih cepat pada wilayah yang lebih kecil; (2) sebagai sebuah pendekatan pada target dimana hubungan antara *local best* dan posisi serta *global best* dengan posisi yang menuju ke nilai 0 hal ini berarti kecepatan akhirnya menurun dan nilai bobot inersia yang kecil dapat mengeliminasi efek kecepatan partikel pada lokasi berikutnya sehingga partikel dapat mengalami konvergen untuk solusi yang ditemukan. Resikonya kawanan mencapai konvergensi yang cepat pada suatu lokasi tertentu yang kemungkinan bukan merupakan optimum global sehingga partikel akan mencoba untuk mengeksplorasi. Sehingga nilai yang besar dari bobot inersia harus dipilih untuk menghilangkan partikel terjebak dalam optimum lokal; (3) untuk mengontrol eksplorasi global dan eksploitasi lokal maka disarankan untuk menggunakan nilai – nilai bobot inersia yang kecil.

Hubungan dari penentuan aturan fuzzy untuk menentukan $\tilde{\omega}(t)$ dapat diuraikan sebagai berikut, nilai $\tilde{\omega}(t)$ ditentukan oleh input fuzzy $\|\zeta(t)\|$ dan $\frac{t}{T}$ kondisi iterasi, nilai yang kecil menunjukkan proses pencarian baru dimulai. Dengan demikian nilai bobot inersia harus memiliki nilai yang besar karena nilai $\frac{t}{T}$ kecil, sehingga kecepatan akan memiliki nilai yang besar maka proses pencarian global terjadi. Demikian sebaliknya jika nilai $\frac{t}{T}$ besar maka bobot inersia haruslah kecil sehingga kecepatan akan melakukan pencarian lokal. Dengan demikian aturan 1,2,3 digunakan “L” dan 4,5,6 lebih besar yaitu “M”. Standar deviasi menunjukkan *cost value* atau jarak diantara partikel dimana aturan 1 sampai 3 pencarian baru saja dimulai. Ketika Nilai $\|\zeta(t)\|$ adalah “H”, menunjukkan lokasi partikel tersebar luas sehingga nilai $\tilde{\omega}(t)$ yang besar dapat meningkatkan global eksplorasi begitupun ketika $\|\zeta(t)\|$ adalah “L”, maka nilai $\tilde{\omega}(t)$ diatur dengan

nilai yang besar hal ini disebabkan peluang untuk solusi terjebak dalam optimum lokal sangat besar. Untuk aturan 2 nilai $\|\zeta(t)\|$ diatur dengan “M” dengan nilai lebih kecil dibandingkan dengan aturan 1 dan 3. Aturan 4 sampai 6 dimana pencarian berada pada bagian pertengahan ($\frac{t}{T} = \text{“M”}$) sehingga nilainya $\tilde{\omega}(t)$ yang diusulkan hampir sama dengan nilai dari aturan 1 sampai 3. Namun terdapat perbedaan dimana ketika $\|\zeta(t)\|$ adalah “L” nilai dari $\tilde{\omega}(t)$ lebih kecil dibandingkan dengan $\|\zeta(t)\|$ yang bernilai “H” hal ini disebabkan karena kemungkinan solusi optimal ditemukan pada bagian pertengahan dari proses dengan nilai $\tilde{\omega}(t)$ yang kecil. Pada aturan 7 sampai 9 proses pencarian berada pada bagian akhir ($\frac{t}{T} = \text{“H”}$) dimana proses lokal eksploitasi dilakukan untuk mendapatkan solusi yang optimal. Sebagai hasilnya ketika nilai dari $\|\zeta(t)\|$ adalah “L” lokasi dari partikel dekat antara satu dengan yang lain akan tetapi jauh dari solusi optimal. Oleh karena itu, $\tilde{\omega}(t)$ akan menggunakan nilai yang kecil. Dengan aturan tersebut akan diperoleh nilai $\tilde{\omega}(t)$ yang digunakan pada persamaan 2.29.

Selain itu penelitian ini menggunakan 9 *linguistic Value* IF-THEN aturan fuzzy untuk menentukan $\beta(t)$ yang diusulkan oleh Ling dkk (Ling, 2012) sebagai berikut:

Rule 1: IF $\|\zeta(t)\|$ is “L” AND $\frac{t}{T}$ is “L”, THEN $\beta(t)$ is “VH” (=0.5)

Rule 2: IF $\|\zeta(t)\|$ is “M” AND $\frac{t}{T}$ is “L”, THEN $\beta(t)$ is “H” (=0.4)

Rule 3: IF $\|\zeta(t)\|$ is “H” AND $\frac{t}{T}$ is “L”, THEN $\beta(t)$ is “VH” (=0.5)

Rule 4: IF $\|\zeta(t)\|$ is “H” AND $\frac{t}{T}$ is “M”, THEN $\beta(t)$ is “H” (=0.4)

Rule 5: IF $\|\zeta(t)\|$ is “M” AND $\frac{t}{T}$ is “M”, THEN $\beta(t)$ is “M” (=0.3)

Rule 6: IF $\|\zeta(t)\|$ is “H” AND $\frac{t}{T}$ is “M”, THEN $\beta(t)$ is “H” (=0.4)

Rule 7: IF $\|\zeta(t)\|$ is “L” AND $\frac{t}{T}$ is “H”, THEN $\beta(t)$ is “VL” (=0.1)

Rule 8: IF $\|\zeta(t)\|$ is “M” AND $\frac{t}{T}$ is “H”, THEN $\beta(t)$ is “L” (=0.2)

Rule 9: IF $\|\zeta(t)\|$ is “H” AND $\frac{t}{T}$ is “H”, THEN $\beta(t)$ is “L” (=0.2)

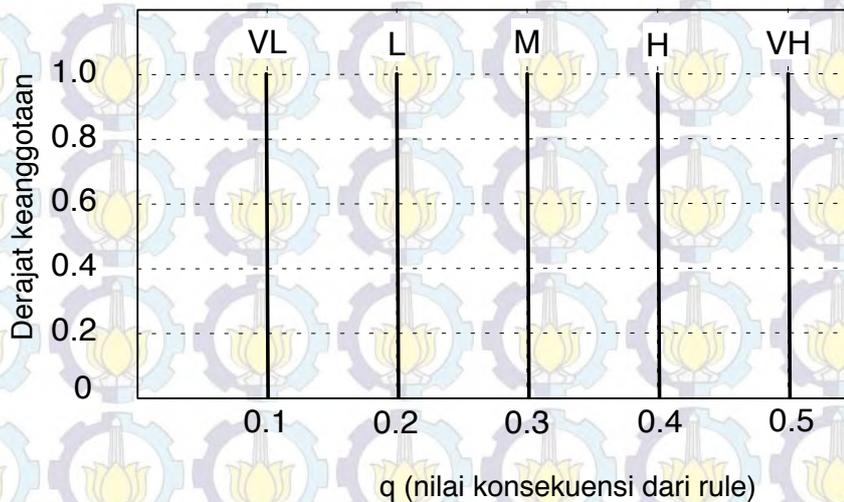
Hubungan untuk aturan fuzzy sama dengan perumusan pada penentuan aturan fuzzy pada fuzzy inertia weight.

Defuzzification

Defuzzification merupakan proses untuk menerjemahkan keluaran dari aturan fuzzy (*fuzzy rule*) kedalam sebuah nilai *crisp* (Ling, 2011) untuk digunakan pada persamaan *velocity* dari PSO. Dengan menggunakan *output singleton fuzzy* akan lebih sederhana untuk melakukan *defuzzification* hal ini disebabkan karena tidak ada area yang berada pada bagian bawah dari fungsi keanggotaan himpunan fuzzy. Nilai keluaran dari bilangan *crisp* dapat dihitung dengan menggunakan *Center Average (CA) Defuzzification* sebagai berikut:

$$\tilde{\omega}(t) = \frac{\sum_{i=1}^n q_i G_i \left(\|\zeta(t)\|, \frac{t}{T} \right)}{\sum_{i=1}^n G_i \left(\|\zeta(t)\|, \frac{t}{T} \right)}, \quad (2.45)$$

dimana q_i adalah lokasi dari karakteristik fungsi keanggotaan himpunan fuzzy tunggal (*singleton*) yang ditentukan didalam *consequent* dari *rule* ke i , n merupakan jumlah dari *rule* (aturan) yang digunakan, $G_i(\|\zeta(t)\|, \frac{t}{T})$ merupakan nilai input fuzzy dari nilai *crisp*. Hal yang sama dilakukan pada *operasi cross-mutated* menggunakan *Output Singleton Fuzzy* seperti Gambar 2.15. Untuk melakukan proses *defuzzification* dari bobot *Cross-Mutated Operation* dapat menggunakan Persamaan 2.46.



Gambar 2.15. Output Singleton Fuzzy untuk *Control Parameter* ($\beta(t)$)

$$\beta(t) = \frac{\sum_{i=1}^n q_i G_i \left(\|\zeta(t)\|, \frac{t}{T} \right)}{\sum_{i=1}^n G_i \left(\|\zeta(t)\|, \frac{t}{T} \right)}, \quad (2.46)$$

dimana q_i merupakan lokasi karakteristik fungsi keanggotaan himpunan fuzzy tunggal berdasarkan *consequent*. Untuk parameter lainnya sama dengan penjelasan dari persamaan 2.45.

2.7 Dataset Penelitian

Penelitian ini menggunakan dataset yang disediakan oleh IDIAP research institute (Silvia Chiappa, Jose del R. Millan) (Chiappa, 2003) Data Competition III, Dataset V (Mental Imagery, multi-class).

2.7.1 Percobaan

Dataset ini terdiri dari 3 subjek normal dengan 4 sesi *non-feedback*. subjek duduk dengan santai dan meletakkan tangan di atas kakinya pada sebuah kursi normal. Terdapat tiga kegiatan mental yang harus dilakukan yaitu,

1. Membayangkan secara berulang – ulang untuk menggerakkan tangan kiri, (*kiri*, class 2).
2. Membayangkan secara berulang – ulang untuk menggerakkan tangan kanan, (*kanan*, class 3).
3. Membuat kata yang diawali dengan huruf depan yang sama secara acak, (*kata*, class 7).

Seluruh subjek melakukan 4 sesi dalam satu hari yang sama, kemudian 4 – 5 menit waktu istirahat diantara 4 menit terakhir. Setiap subjek melakukan kegiatan mental kira – kira 15 detik dan secara acak bergantian untuk melakukan kegiatan mental lain sesuai permintaan dari operator. EEG data tidak dipisah karena subjek melaksanakan setiap kegiatan mental secara menerus. Algoritma harus menyediakan keluaran setiap 0.5 detik menggunakan waktu terakhir data. Data tersedia dalam dua cara:

1. *Raw EEG sinyals*. Dengan sampling rate 512 Hz.
2. *Precomputed features*. Data mentah potensi EEG pertama kali secara spasial di filter menggunakan rata-rata dari *Surface Laplacian*. Kemudian, setiap

62.5 ms – berkisar 16 kali per detik – power spectral density (PSD) di dalam band 8 – 30 Hz telah di perkirakan pada bagian akhir dari data dengan resolusi frekuensinya dari 2 Hz untuk 8 channel centro-parental C3, Cz, C4, CP1, CP2, P3, Pz, dan P4. Hasilnya, sampel EEG memiliki ukuran 96 vector (8 channel dengan 12 kali komponen frekuensi).

2.7.2 Format data

Setiap subjek terdapat 3 file untuk pelatihan dan 1 file pengujian (sesi yang terakhir di rekam). File pelatihan diberikan label sedangkan file pengujian tidak diberi label. Data tersedia dalam format ASCII.

1. *Precomputed features*: file yang berisi contoh PSD per baris (yaitu, 12 komponen pertama adalah PSD di dalam band 8 – 30 Hz pada channel C3, dan seterusnya, sehingga total adalah 96 komponen). Pada file pelatihan, komponen ke 97 menunjukkan label kelas.
2. Sinyal EEG mentah setiap baris dari file berisi 32 potensi EEG diperoleh pada waktu instan yang diberikan dalam urutan: Fp1, AF3, F7, F3, FC1, FC5, T7, C3, CP1, CP5, P7, P3, Pz, PO3, O1, Oz, O2, PO4, P4, P8, CP6, CP2, C4, T8, FC6, FC2, F4, F8, AF4, Fp2, Fz, Cz. Dalam file pelatihan setiap baris memiliki komponen ke 33 yang menunjukkan label kelas.

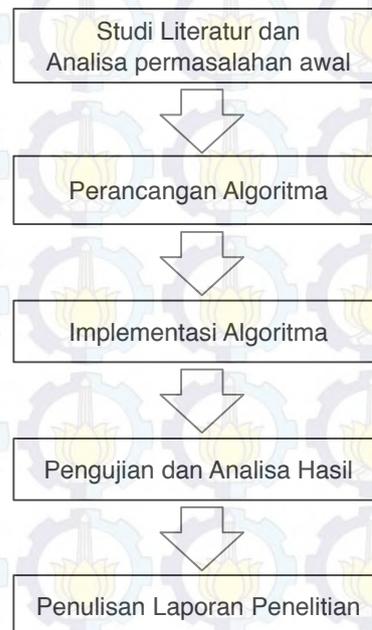
2.7.3 Informasi Teknis

Sinyal EEG telah direkam dengan menggunakan sistem Biosemi menggunakan penutup kepala dengan 32 elektroda yang terletak di posisi mengikuti standar internasional 10 – 20 sistem. Menggunakan sampling rate 512 Hz. Sinyal diperoleh pada full DC. Tidak ada artefak penolakan atau koreksi dipekerjakan.

BAB 3

METODE PENELITIAN

Metodologi yang digunakan untuk menyelesaikan penelitian optimalisasi pembobotan menggunakan *particle swarm optimization* terlihat pada Gambar 3.1. Penelitian bertujuan untuk menemukan kondisi optimal terhadap tiga komponen penting pada kecepatan partikel agar mencapai nilai konvergensi yaitu, pertama *inersia* dari kecepatan sebelumnya, kedua "*cognition*" atau "*individual knowledge*" yang mewakili pemikiran diri sendiri dan ketiga "*social*" atau "*group knowledge*" yang menggambarkan hubungan kerjasama diantara partikel. Metode IPSO (Lin, 2009) bertujuan mengoptimalkan bagian *social* dari partikel dengan mencari parameter *global best* namun tetap menggunakan kondisi *bobot inersia* sehingga masih memiliki sifat linear. Pada umumnya optimasi memiliki sifat *non-linear* sehingga metode IPSO perlu ditingkatkan dengan menambahkan *fuzzy inertia weight* (Ling, 2012) untuk mengoptimalkan pencarian yang dilakukan pada komponen *bobot inersia* dan *kecepatan sebelumnya* (Yadmetlat, 2009).



Gambar 3.1 Tahapan metodologi penelitian

3.1 Studi Literatur

Memahami masalah optimalisasi secara keseluruhan sangatlah membantu dalam proses penelitian ini sehingga perlu dilakukan penelaah lebih jauh tentang masalah yang sedang diteliti. Studi literatur bertujuan untuk mengikuti perkembangan penelitian yang berkaitan dengan optimalisasi bobot baik penelitian terdahulu maupun saat ini. Memahami metode yang digunakan akan membantu untuk dapat menemukan titik kelemahan dari framework, prosedur ataupun algoritma. Dengan dasar pengetahuan tersebut dapat diperoleh berbagai informasi sebagai landasan dalam membangun dan menyempurnakan suatu metode ataupun algoritma sebelumnya sehingga dapat dijadikan sebagai kontribusi dalam penelitian. Studi literatur yang dilakukan meliputi (1) penyediaan dan analisis data yang relevan untuk sinyal EEG; (2) dasar teori yang memberikan penjelasan tentang fitur ekstraksi, reduksi data dan klasifikasi; (3) penelitian yang berhubungan dengan berbagai metode untuk melakukan klasifikasi pada sinyal EEG; (4) metode klasifikasi EEG menggunakan *Artificial Neural Network* dengan proses pembobotan dilakukan melalui *backpropagation*; (5) penelitian yang berhubungan dengan optimalisasi pembobotan melalui berbagai metode optimasi yang digunakan pada jaringan saraf tiruan menggunakan *feedforward*; (6) penelitian yang mengusulkan berbagai metode dan framework dalam melakukan optimasi terhadap parameter kecepatan dari *particle swarm optimization* yang terdiri dari bobot inersia kecepatan sebelumnya *cognition* sebagai gambar perilaku diri sendiri dan sifat *social* dari partikel *swarm* optimasi; (7) penelitian yang berhubungan dengan penggunaan operator baru dari algoritma genetik yaitu *evolutionary direction operator*.

Analisa awal sangat penting dilakukan untuk menemukan sumber informasi yang dijadikan sebagai landasan untuk membangun kontribusi penelitian. Berdasarkan analisis yang dilakukan pada penelitian terdahulu maupun saat ini dapat disimpulkan bahwa metode IPSONN melakukan proses klasifikasi menggunakan metode *neural network* dimana proses pembobotan dioptimasi melalui *particle swarm optimization* (PSO). Penelitian tersebut bertujuan untuk mendapatkan kualitas *global best* dengan meningkatkan kinerja dari algoritma

menggunakan metode *evolutionary direction operator* (Lin, 2009) namun metode tersebut masih menggunakan *bobot inersia* yang memiliki sifat linear. Penelitian Ling dan Chai (Ling, 2012; Chai, 2014) bertujuan untuk menghilangkan kelemahan ini dengan mengubah sifat linear menjadi non-linear melalui perubahan pada bagian bobot inersia menggunakan *fuzzy inertia weight* untuk mengoptimalkan pencarian. Penelitian lain (Eberhart, 2000) telah memberikan perbandingan antara nilai *constriction* (k) dengan nilai *bobot inersia* (ω) yang memberikan hasil bahwa penggunaan kedua faktor merupakan suatu kondisi khusus. Selain itu peningkatan kinerja dapat diperoleh dengan cara berhati – hati memilih bobot inersia ω , C_1 dan C_2 . Metode dinamis yang dapat beradaptasi dengan bobot inersia sedang dalam penelitian lebih lanjut dimana telah dilakukan oleh Ling, dkk dan Chai, dkk.

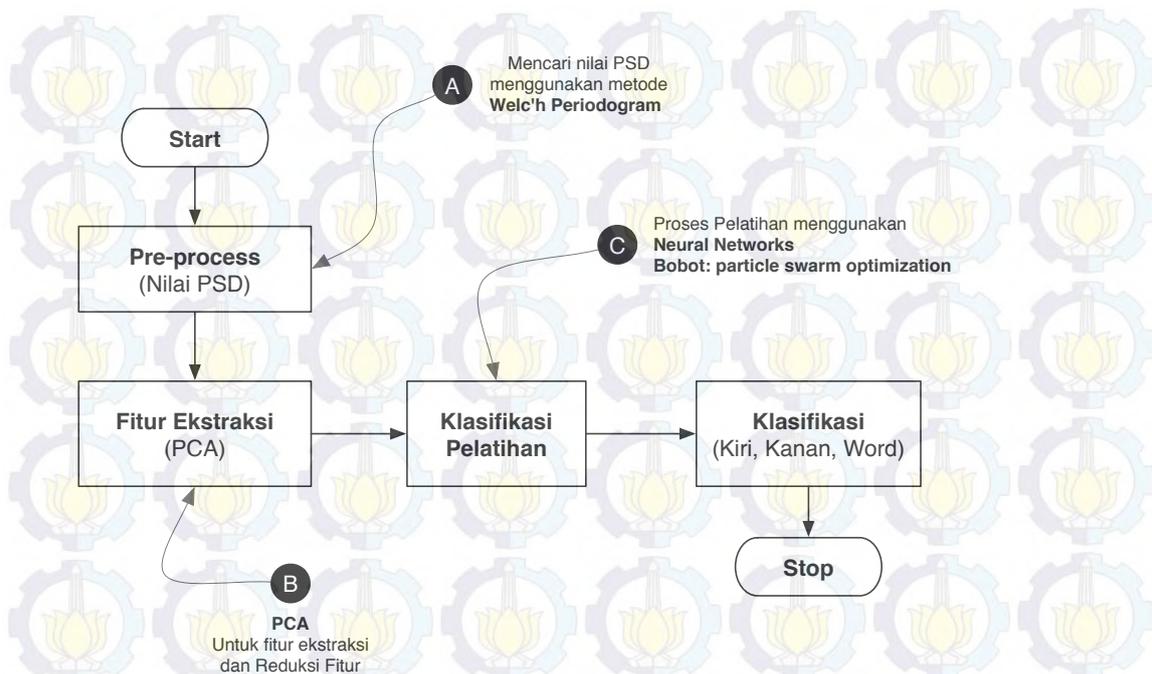
Berdasarkan informasi tersebut diusulkan penelitian yang menggabungkan algoritma untuk menentukan nilai *global best* yang di usulkan oleh Lin, dkk (Lin, 2009) dengan menambahkan bobot inersia dinamis melalui penerapan *fuzzy inertia weight* yang di usulkan oleh Ling (Ling, 2012) dan Chai (Chai, 2014) sehingga diharapkan dapat mengoptimalkan proses penentuan pembobotan yang akan digunakan pada metode *neural network* untuk proses klasifikasi EEG.

3.2 Perancangan Algoritma

Perancangan algoritma sangat penting untuk dilakukan agar mendapatkan hasil yang diharapkan. Secara keseluruhan algoritma utama terlihat seperti Gambar 3.2 dimana terdapat beberapa proses yang perlu dilakukan ketika melakukan proses klasifikasi pada sinyal EEG.

3.2.1 Pre-Processing

Pengolahan data awal digunakan untuk mendapatkan nilai *Power Spectral Density* (PSD) yang digunakan sebagai fitur ekstraksi dan menjadi input pada jaringan saraf tiruan. Sumber data yang digunakan adalah data mentah dari EEG yang diambil dari *IDIAP research* yang harus di filter menggunakan *surface laplacian*. Metode *Welc'h Periodogram* digunakan untuk menentukan nilai PSD.



Gambar 3.2 Diagram alir untuk prosedur analisis

Nilai PSD dari setiap channel akan dihitung dan di rata – ratakan untuk setiap pekerjaan mental yaitu *left*, *right* dan *word*. Ukuran vektor akan terbentuk sebanyak 96 komponen karena 8 channel di kali 12 frekuensi. Setelah seluruh PSD dapat ditentukan maka akan dilakukan prosedur untuk mereduksi dengan menggunakan PCA.

3.2.2 Fitur Ekstraksi

Tujuan fitur ekstraksi pada sub bab ini digunakan untuk menjelaskan proses reduksi data sinyal EEG yang diperoleh dari *pre-processing* untuk mengurangi dimensi dari vector fitur asli. Dengan *Principle Component Analysis* (PCA) data direduksi dari *high dimensional* menjadi *k-dimensi* yang lebih kecil Gambar 3.3 merupakan prosedur perhitungan metode PCA. Berdasarkan Lin (Lin, 2009) proses PCA dapat dilakukan dengan prosedur berikut:

Step 1: hitung *mean vector* (vektor rata – rata),

$$m = \frac{1}{nTr} \sum_{i=1}^{nTr} P_i, \quad (3.1)$$

dimana $P_i = [p_1 \dots p_t]^t$ adalah nilai ke- i dimensi- d untuk contoh pelatihan, nTr adalah jumlah total dari sampel pelatihan. Step 2, hitung matriks koefisien, seperti berikut:

$$\Sigma = \sum_{i=1}^{nTr} (p_i - m)(p_i - m)^t, \quad (3.2)$$

Selanjutnya Step 3, mencari nilai *eigenvalue* dan *eigenvector* dari *covariance matrix* yang diselesaikan dengan:

$$\Sigma \mathbf{x} = \lambda \mathbf{x} \quad (3.3)$$

dan dapat ditulis kembali sebagai berikut:

$$(\Sigma - \lambda \mathbf{I}) \mathbf{x} = 0, \quad (3.4)$$

dimana \mathbf{I} merupakan matriks identitas dan 0 adalah vektor nol.

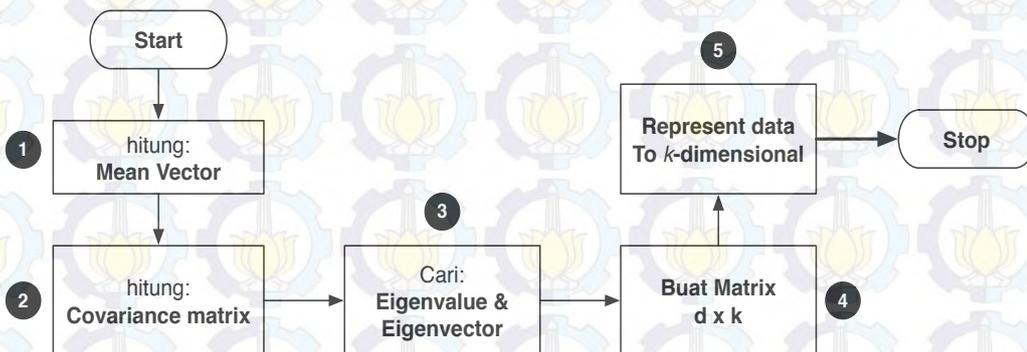
Step 4, membentuk matriks $d \times k$,

$$\mathbf{A} = [e_1, e_2, \dots, e_k] \quad (3.5)$$

Step 5, tampilkan data original melalui proyeksi data ke dalam dimensi $-k$ subruang tergantung pada:

$$\mathbf{p}' = \mathbf{A}^t (\mathbf{p} - \mathbf{m}), \quad (3.6)$$

dimana $\mathbf{p}' = [p'_1 \dots p'_k]^T$ adalah fitur vektor PCA.



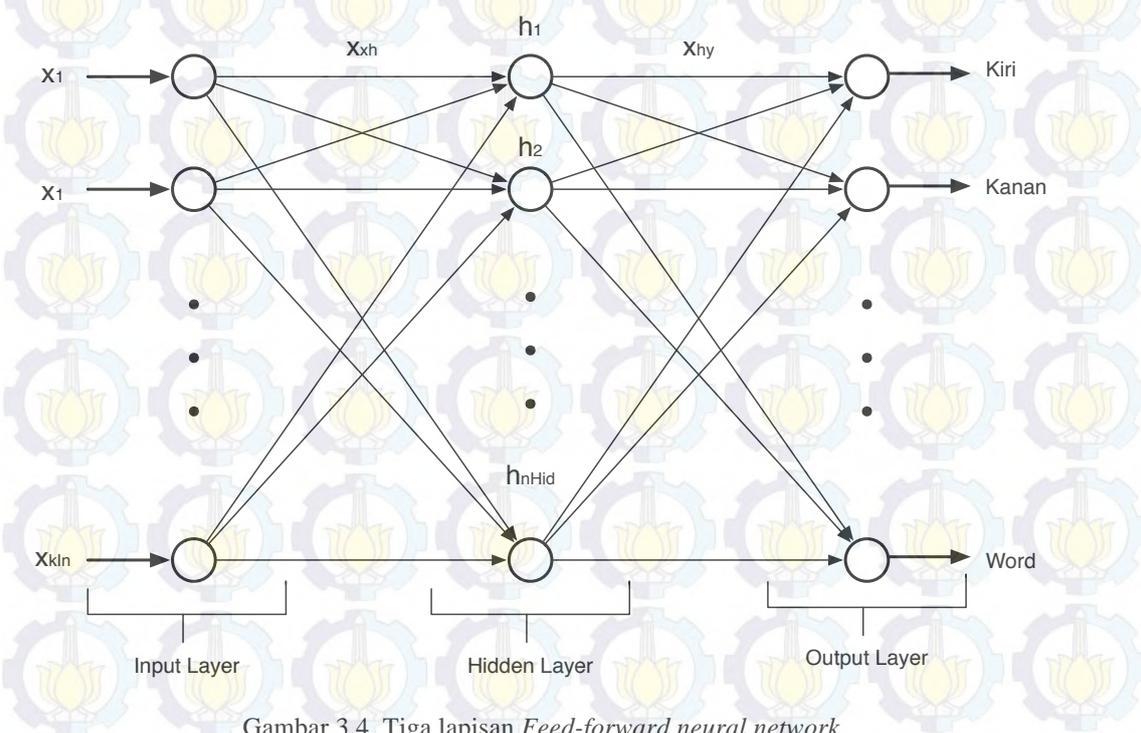
Gambar 3.3 Prosedur perhitungan PCA

3.2.3 Pelatihan Data

Neural network digunakan sebagai *classifier* untuk membedakan fitur PCA dari nilai PSD didalam sinyal EEG. Dalam rangka untuk menyesuaikan parameter NN secara efisien maka digunakan algoritma pembelajaran menggunakan IPSO. Sebelum melakukan proses klasifikasi data EEG perlu di normalisasi diantara nilai 0 dan 1. Penelitian ini menggunakan tiga layer *feed-forward* untuk jaringan saraf tiruan dengan satu *input layer*, *hidden layer* dan *output layer* seperti terlihat pada Gambar 3.4. Jumlah simpul dari *input layer* ditentukan berdasarkan ukuran fitur PCA. Karena sinyal EEG diklasifikasikan ke dalam tiga kegiatan mental maka *output node* juga diatur berjumlah tiga simpul yang berhubungan dengan kegiatan mental kiri, kanan dan kata. Tiga output ditampilkan melalui vektor unit, kiri = [1 0 0], kanan = [0 1 0] dan kata = [0 0 1]. Nilai bobot dari NN ini dioptimasi dengan menggunakan *particle swarm optimization*.

3.2.4 Peningkatan Algoritma IPSO

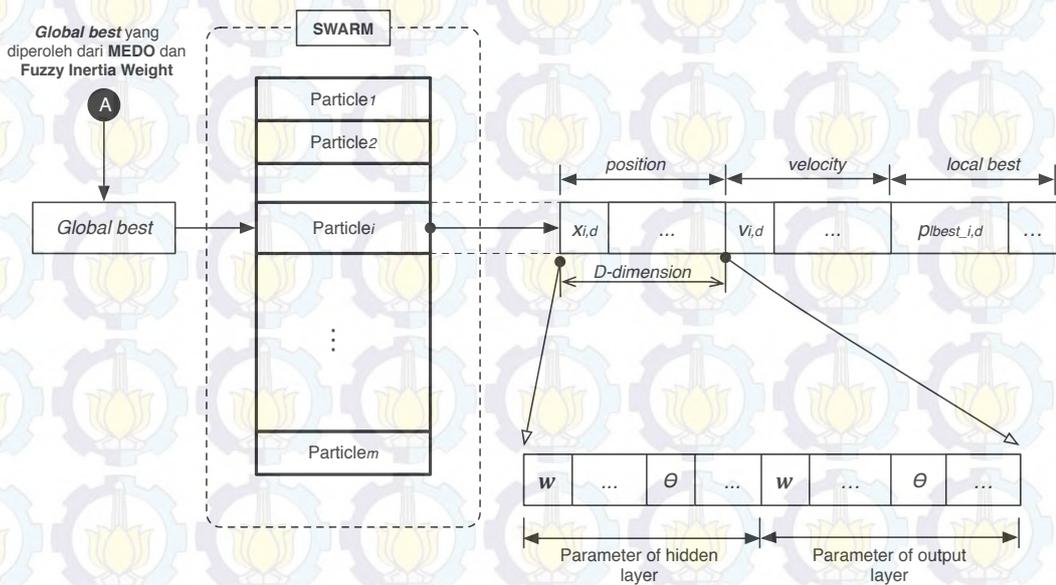
Jaringan saraf tiruan digunakan sebagai proses klasifikasi dimana pada saat proses pelatihan jaringan akan membuat bobot secara acak dengan nilai yang berada pada jangkauan 0 sampai 1.



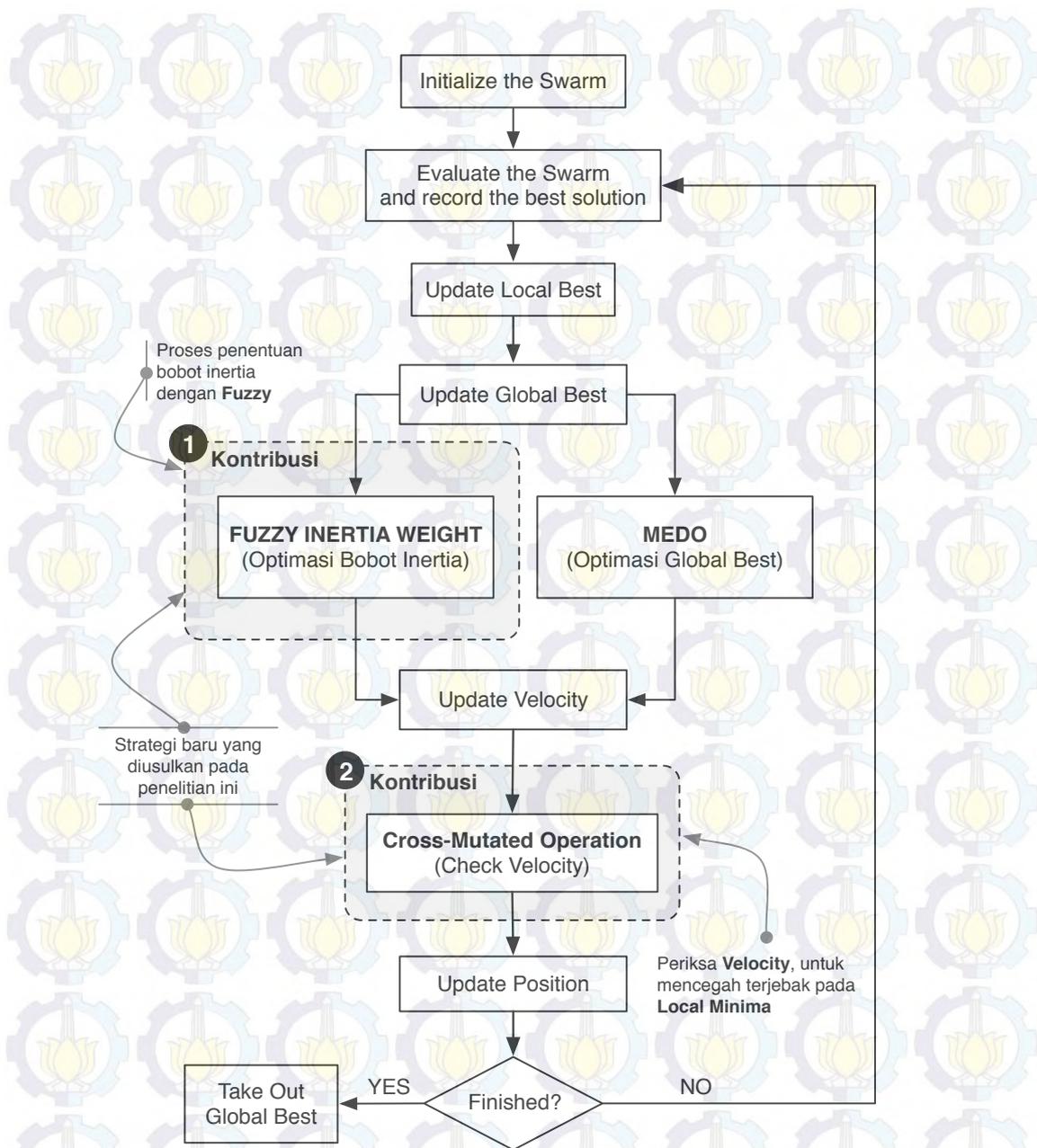
Gambar 3.4 Tiga lapisan *Feed-forward neural network*

Jika proses pelatihan dilakukan maka jaringan akan mencari bobot yang optimal sehingga proses mencapai nilai konvergen hal ini lakukan secara berulang dimulai dari proses *feed-forward* jika nilai masih belum konvergen maka bobot akan di update dimana jaringan akan bergerak mundur. Hal ini membuat proses pembobotan memakan waktu cukup lama selain itu kemungkinan terjebak dalam minimum lokal dapat terjadi.

Untuk mengatasinya maka proses pembobotan dioptimasi dengan metode *Particle Swarm Optimization* (PSO) yang dapat menentukan bobot pelatihan dengan lebih baik dan cepat. Gambar 3.5 merupakan struktur dari swarm dimana setiap partikel memiliki elemen yang merupakan anggota dari partikel ke t setelah *global best* ditemukan maka partikel yang terbaik diantara seluruh partikel dalam swarm atau kawanan dapat ditentukan. Partikel dengan *global best* terdiri dari elemen yang memiliki posisi, kecepatan dan *local best*. Selanjutnya koordinat posisi dari elemen dalam partikel akan memiliki hubungan dengan bobot yang ada pada jaringan saraf tiruan yaitu bobot *hidden layer* dan *output layer*. Gambar 3.6 menunjukkan diagram prosedur penelitian yang dikembangkan berdasarkan penelitian Lin (Lin, 2009).



Gambar 3.5 Struktur SWARM



Gambar 3.6 Strategi baru yang diusulkan pada algoritma FMEDO

Gambar 3.6 terdapat dua strategi baru yang di tambahkan pada algoritma IPSO (Lin, 2009) yaitu point (1) menambahkan *Adaptive inertia weight* (dengan menggunakan *Fuzzy Inference System*) untuk membuat bobot inersia dinamis sehingga proses pencarian lebih optimal beberapa alasan perlunya menggunakan fuzzy yaitu (a) bobot inersia yang bersifat linier tidak dapat digunakan karena kebanyakan kasus optimasi bersifat non-linier, sehingga fuzzy dapat dengan mudah untuk memodelkan sistem dengan sifat non-linear, (b) nilai bobot inersia

berada pada interval antara 0 sampai 1 sehingga perlu dikategorikan nilainya kedalam himpunan fuzzy dengan variabel “Low”, “Medium” dan “High” menggunakan *Fuzzy inference system*, (2) melakukan pemeriksaan terhadap kecepatan (*velocity*) partikel swarm untuk memastikan bahwa parameter tidak terjebak dalam minimum lokal dengan menggunakan operasi *cross-mutated*. Selanjutnya deskripsi dari prosedur tersebut dijelaskan berikut ini,

1. Definisi awal kawanan (swarm)

Mendefinisikan nilai awal jumlah kawanan m partikel dengan posisi dan kecepatan secara acak dari dimensi D dalam ruang pencarian. Dalam penelitian ini, koordinat posisi dari partikel di dalam dimensi d berhubungan dengan setiap variabel (bobot dan bias) yang akan di latih pada jaringan saraf tiruan.

2. Mengevaluasi kawanan

Setiap partikel akan dievaluasi *fitness* tergantung pada optimasi yang diinginkan. Untuk evaluasi dari NN menggunakan *mean absolute error* (MAE) yang di hitung sebagai berikut:

$$MAE = \sqrt{\frac{1}{nTr} \sum_{i=1}^{nTr} \sum_{j=1}^{nOut} |t_{i,j} - y_{i,j}|}, \quad (3.7)$$

dimana nTr merupakan jumlah data training, $nOut$ jumlah dari output jaringan, dan $t_{i,j}$ dan $y_{i,j}$ merupakan target output ke- j dan jaringan ouput sesungguhnya dari data pelatihan ke- i . fungsi fitness didefinisikan sebagai:

$$fitness = \frac{1}{1 + MAE} \quad (3.8)$$

3. Mengubah *local best* dan *global best*

Kemudian dilakukan perubahan pada *local best* dan *global best*. Jika nilai *fitness* dari pertikel lebih tinggi daripada *local best* maka *local best* akan menggantikan partikel dan lebih lanjut jika *local best* lebih baik dari *global best* saat ini maka *global best* harus di ubah di dalam kawanan.

4. Modified Evolutionary Direction Operator

Dengan menggunakan MEDO akan dipilih tiga solusi terbaik di dalam setiap generasi untuk melakukan operasi arah evolusi. Gambar 3.7 memperlihatkan bagan alir untuk IPSONN (Lin, 2009). Setelah proses pembelajaran akan di tentukan tiga nilai partikel yang terbaik dan akan di urutkan berdasarkan fungsi fitness yaitu “low”, “medium” dan “high” partikel. Tiga input (preferred) dan partikel keluaran (created) yang di simbolkan sebagai berikut:

Partikel Input:

- Partikel “low”, $C_l = (C_{l1}, C_{l2}, C_{l3}, \dots, C_{ld})$, dengan fitness F_l
- Partikel “medium”, $C_m = (C_{m1}, C_{m2}, C_{m3}, \dots, C_{md})$, dengan fitness F_m
- Partikel “high”, $C_h = (C_{h1}, C_{h2}, C_{h3}, \dots, C_{hd})$, dengan fitness F_h

Partikel Output, $C_o = (C_{o1}, C_{o2}, C_{o3}, \dots, C_{od})$, dengan fitness F_o .

Secara ringkas dapat di jelaskan prosedur dari MEDO langkah pertama mengatur *magnitudes* dari dua arah evolusi dengan 1 (*misl*, $D_1 = 1$ dan $D_2 = 1$), langkah kedua lakukan operasi migrasi untuk membangun populasi baru yang beragam dan mencegah individu dari pengelompokan bertahap. Langkah ketiga hitung nilai C_o dengan menggunakan persamaan,

$$C_{oj} = C_{lj} + D_1 \times (C_{lj} - C_{mj}) + D_2 \times (C_{lj} - C_{hj}), \quad (3.9)$$

dimana C_{oj} merupakan output partikel ke $-j$, C_{lj} merupakan nilai partikel “low” yang ke $-j$, C_{mj} merupakan partikel “medium” yang ke $-j$ dan C_{hj} merupakan partikel “high” ke $-j$. Langkah keempat evaluasi fitness baru F_o dari partikel output yang baru dibuat (C_o). Langkah kelima update partikel “low” (C_l), partikel “medium” (C_m) dan partikel “high” (C_h). Dengan cara berikut:

- (1) IF $F_o > F_h$ then $C_h = C_o$, $C_m = C_h$, and $C_l = C_m$
- (2) ELSE if $F_o > F_m$ and $F_o < F_h$, then $C_m = C_o$ and $C_l = C_m$
- (3) ELSE if $F_o > F_l$ and $F_o < F_m$, then $C_l = C_o$
- (4) ELSE if $F_o = F_l = F_m = F_h$, then $C_o = C_o + N_r$ ($N_r \in [0,1]$)
- (5) ELSE if $F_o < F_h$, then $D_1 = D_1 \times -0.5$ and $D_2 = D_2 \times -0.5$

Atau dapat menggunakan cara berikut: (Chiang, 2005)

IF $F_o > F_h$ then $D_1 = D_1 \times -0.5$ and $D_2 = D_2 \times -0.5$
 IF $F_o < F_h$ and $F_o = F_l = F_m$ then $C_{oj} = C_{oj} + N_r$, dimana $N_r \in [0,1]$

IF $F_o < F_h$ and $F_o \neq F_l \neq F_m$ then

IF $F_o < F_l$ then $C_l = C_o$;

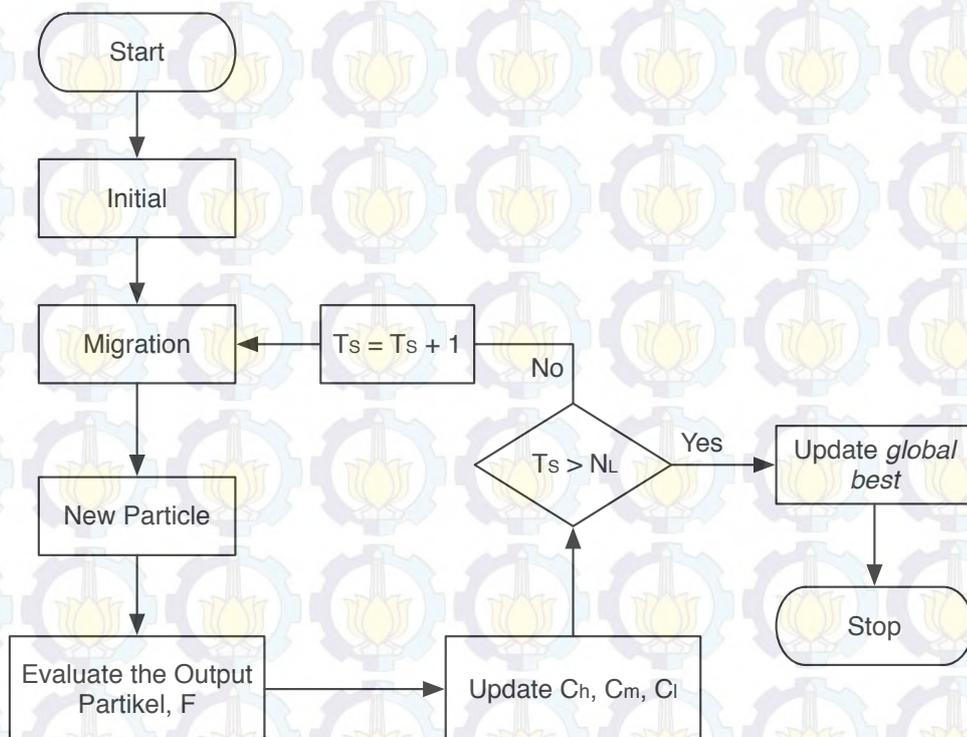
IF $F_o < F_m$ and $F_l < F_o$ then $C_m = C_o$

IF $F_o < F_h$ and $F_m < F_o$ then $C_h = C_o$

dimana nilai $C_{op} = \max[\min(C_{oj}, C_o^{max}), C_o^{min}]$, penelitian ini menggunakan usulan dari Chao-Lung Chiang (Chiang, 2005). Langkah keenam jika MEDO berakhir lanjut ke langkah terakhir jika tidak langkah kedua diulangi dengan menaikkan $T_s = T_s + 1$. Langkah terakhir update *global best* setelah itu hitung velocity beserta dengan komponen *fuzzy inertia weight*.

5. Hitung Adaptive Bobot Inertia

Jika *global best* telah menemukan parameter yang tepat maka dapat dihitung bobot inersia dengan menggunakan *fuzzy inertia weight* seperti pada Gambar 3.8 prosedur perhitungannya telah di uraikan pada bagian dasar teori (Kontribusi yang diusulkan).



Gambar 3.7 Bagan alir dari IPSONN (Lin, 2009)

6. Update Kecepatan

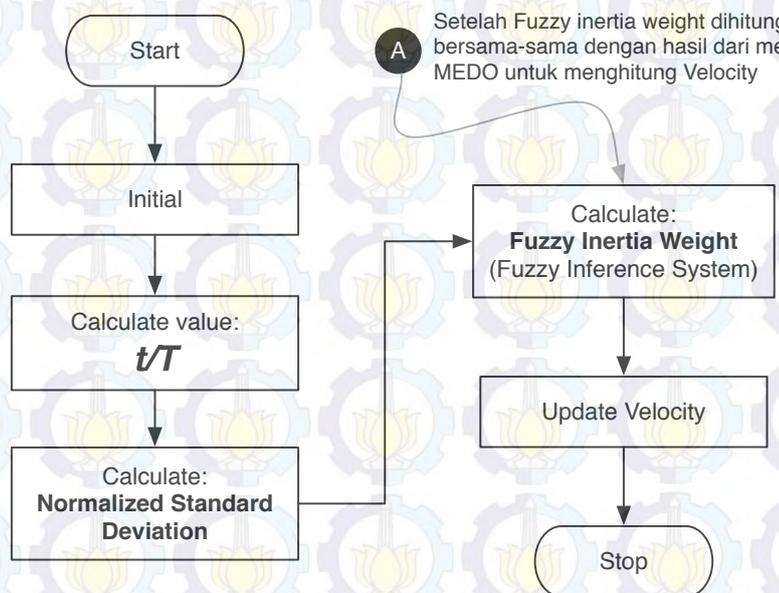
Setelah *global best* diperoleh melalui MEDO dan bobot inersia diperoleh dari metode *Fuzzy Inertia Weight* maka persamaan kecepatan (*velocity*) dapat dihitung.

7. Periksa Minimum Lokal

Langkah selanjutnya kecepatan akan diperiksa untuk mencegah agar tidak terjebak dalam minimum lokal dengan menggunakan Operasi *Cross-mutated* (CM), prosedur perhitungannya telah diuraikan pada dasar teori (Kontribusi yang diusulkan).

8. Hitung Posisi Partikel

Selanjutnya hitung posisi dari setiap elemen yang ada di dalam partikel dengan *global best* yang sudah ditemukan pada langkah MEDO. Jika kondisinya seperti yang di harapkan maka ambil nilai dari *global best* untuk menggunakan seluruh posisi dari elemen yang ada di dalam partikel dan dijadikan sebagai *bobot* dan *bias* pada jaringan saraf tiruan. Selanjutnya proses pelatihan dari jaringan dilanjutkan jika bobot sudah konvergensi maka gunakan sebagai parameter pengujian dari sinyal EEG.



Gambar 3.8 Bagan alir untuk menentukan Fuzzy Inertia Weight

3.3 Perangkat Lunak

Penelitian yang dilakukan diimplementasikan dengan menggunakan perangkat lunak MATLAB 2013a. Lingkungan penelitian untuk pengembangan sistem dalam rangka melakukan pengujian metode yang diusulkan adalah:

1. Spesifikasi perangkat keras (hardware):

Pengembangan Aplikasi menggunakan:

- a. CPU: 2.3 GHz Intel Core i5
- b. RAM: 8GB 1333 MHz DDR3

Proses pengujian dilakukan pada perangkat:

- a. CPU: Inter® Core™ i3-2130 CPU @ 3.40GHz
- b. RAM: 4GB

2. Spesifikasi perangkat lunak (software):

- a. Sistem Operasi MAC OS X Lion 10.7.5 (11G63) (Untuk pengembangan dan pembuatan aplikasi).
- b. Sistem Operasi WINDOWS 7 Enterprise, Service Pack1. 32-bit Operating System (Untuk proses pengujian aplikasi).

BAB 4

HASIL PENGUJIAN DAN EVALUASI

Bab ini digunakan untuk menguraikan cara peneliti dalam melakukan proses klasifikasi terhadap sinyal EEG (*electroencephalogram*) terhadap tiga pekerjaan mental. Proses pelatihan (*training*) dilakukan dengan menggunakan *Neural network feed-forward* dimana proses pembobotan dari jaringan di optimasi menggunakan *particle swarm optimization* yang menggunakan bobot inersia dinamis dengan aturan *fuzzy inference system* yang telah di tentukan. Berdasarkan tujuan penelitian analisa dilakukan untuk mendapatkan akurasi klasifikasi yang lebih baik dari metode sebelumnya yang merupakan pengembangan dari IPSONN dan *Fuzzy Inference System*.

3.1 Lingkungan dan Data Pengujian

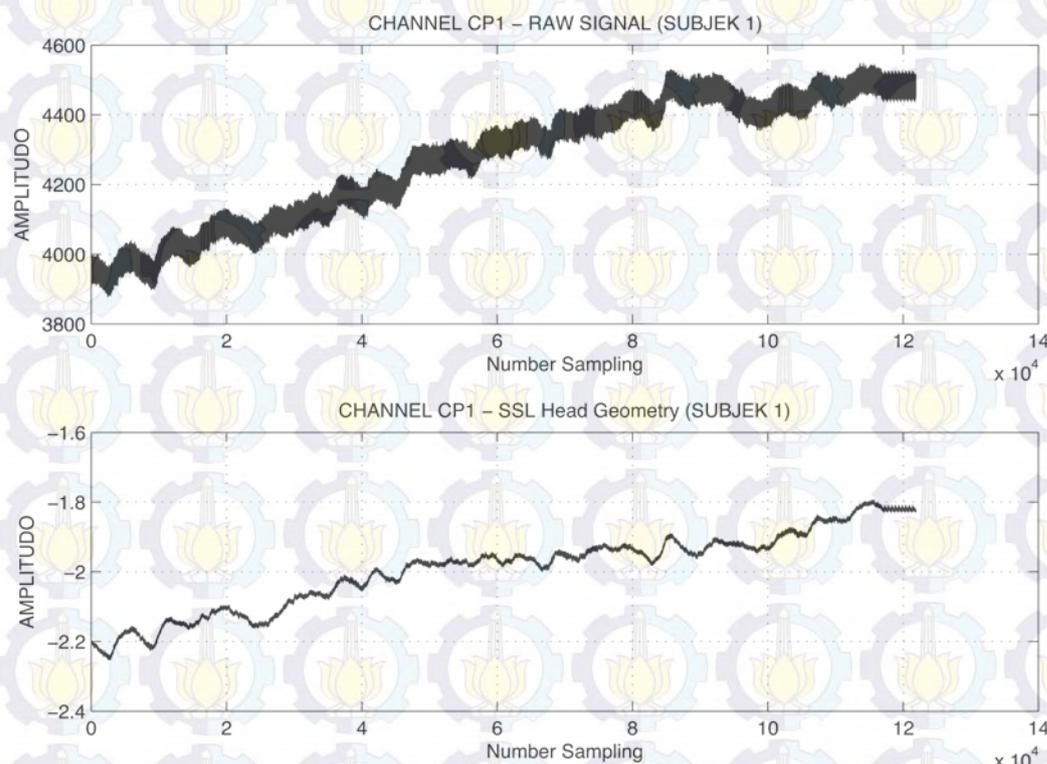
Lingkungan pengujian klasifikasi EEG akan dilakukan pada beberapa lingkungan *operating system* dengan spesifikasi perangkat keras sebagai berikut proses pengembangan system dilakukan pada CPU, 2.3 GHz Intel Core i5, RAM, 8GB 1333 MHz DDR3 dengan sistem operasi MAC OS X Lion 10.7.5 (11G63) sedangkan untuk lingkungan pengujian yaitu proses pelatihan dan pengujian dilakukan pada spesifikasi perangkat keras CPU, Inter® Core™ i3-2130 CPU @ 3.40GHz; RAM. 4GB, dengan *operating system* 32-bit WINDOWS 7 Enterprise, Service Pack1. Data yang digunakan dalam penelitian ini adalah data yang di distribusikan melalui BCI Competition III yang di sediakan oleh *IDIAP Research Institute* yang direkam oleh sistem Biosemi menggunakan penutup yang memiliki 32 elektroda. Dimana setiap data terdiri dari 33 kolom dimana kolom yang ke 33 merupakan tipe kelas sedangkan 32 kolom lainnya merupakan channel *electrode*, akan tetapi hanya 8 channel yang akan digunakan.

3.2 Pre-Processing dan Fitur Ekstraksi

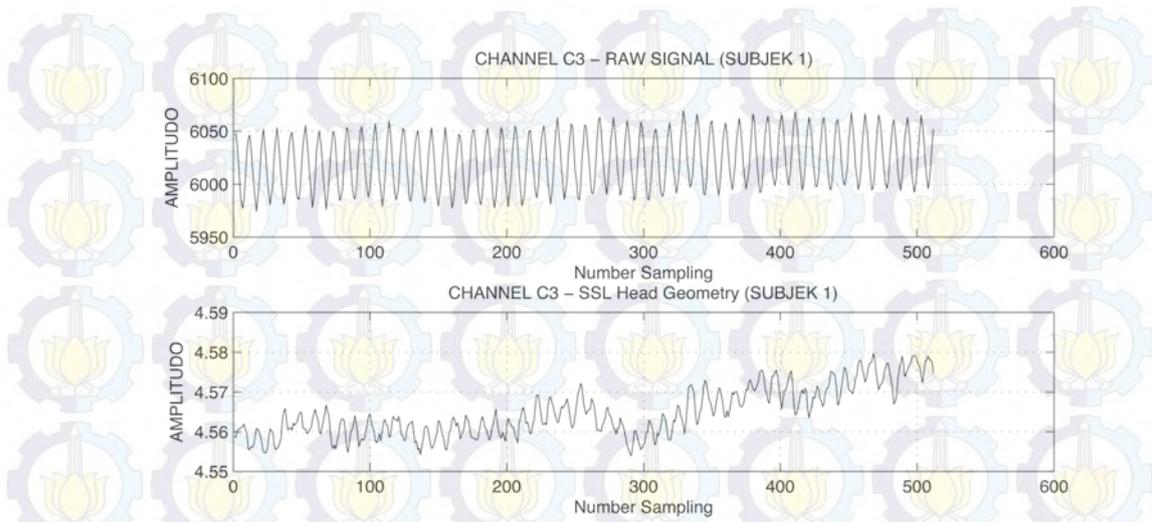
Berikut ini akan dilakukan serangkaian pengolahan data awal yang diawali dengan proses filter menggunakan *surface laplacian*, fitur ekstraksi dan proses reduksi vector fitur.

4.2.1 Proses Filtering

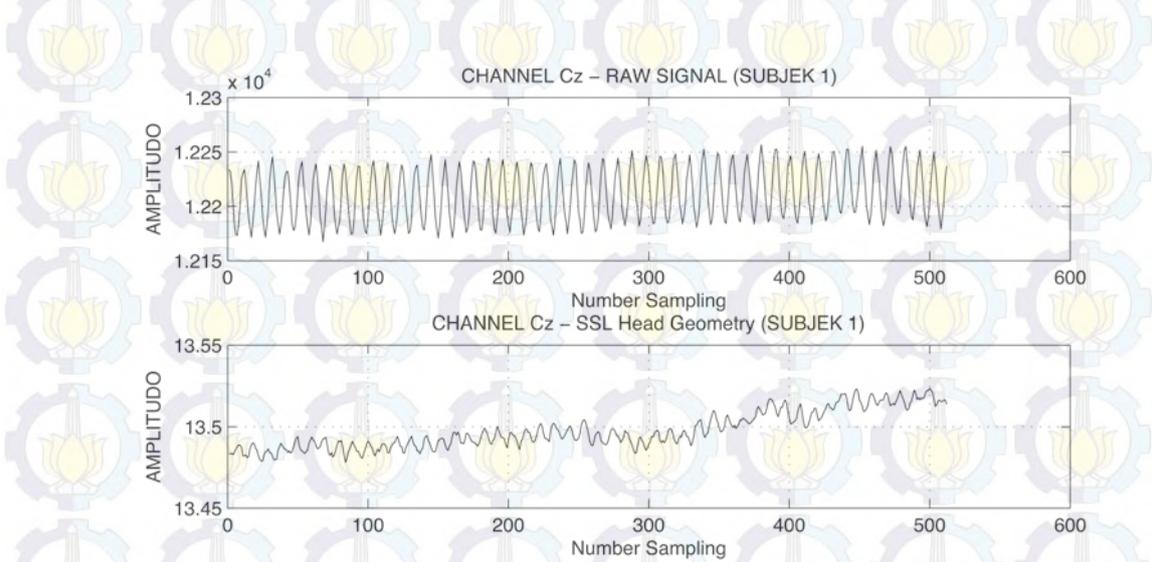
Tujuan dari proses filtering adalah untuk memperkuat sinyal (konsentrasi) dari setiap elektroda, hal ini disebabkan karena setiap *channel* elektroda di pengaruhi oleh elektroda tetangga yang menyebabkan tegangan menjadi lebih besar, dengan cara menghilangkan pengaruh tegangan dari elektroda tetangga sehingga *channel* yang ditentukan mendapatkan tegangan yang seharusnya dimiliki. Metode *Realistic head geometry* berdasarkan pada *spherical surface laplacian* akan digunakan pada penelitian ini seperti yang diusulkan Siyi Deng, dkk (Deng, 2011). Dengan bantuan Tools SSLTool (Deng, 2011) maka konsentrasi setiap elektroda yang dibutuhkan dapat dilihat pada Gambar 4.1 untuk Subject Pertama Channel C3. Gambar 4.2 s.d 4.9 merupakan gambar raw sinyal dengan SSL – head geometry.



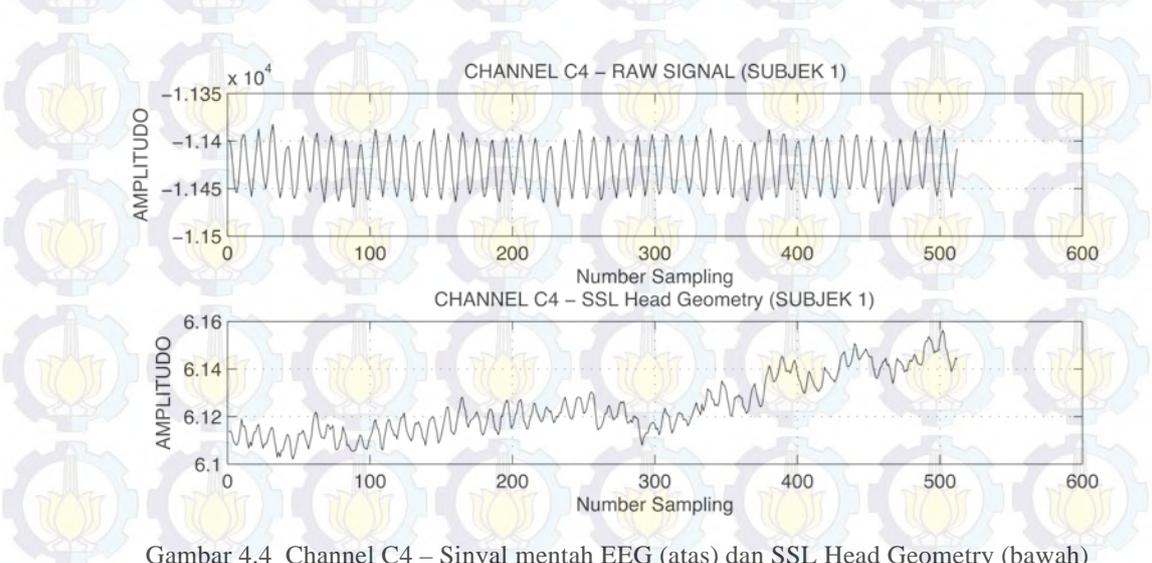
Gambar 4.1 Channel CP1 – Sinyal EEG (atas) dan SSL Head Geometry (bawah) untuk Subject 1.



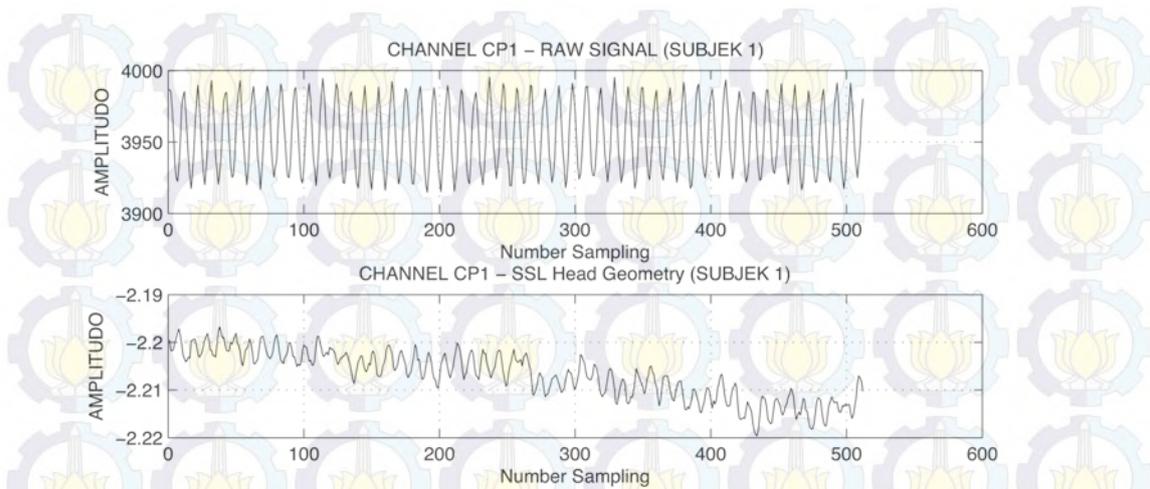
Gambar 4.2 Channel C3 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)



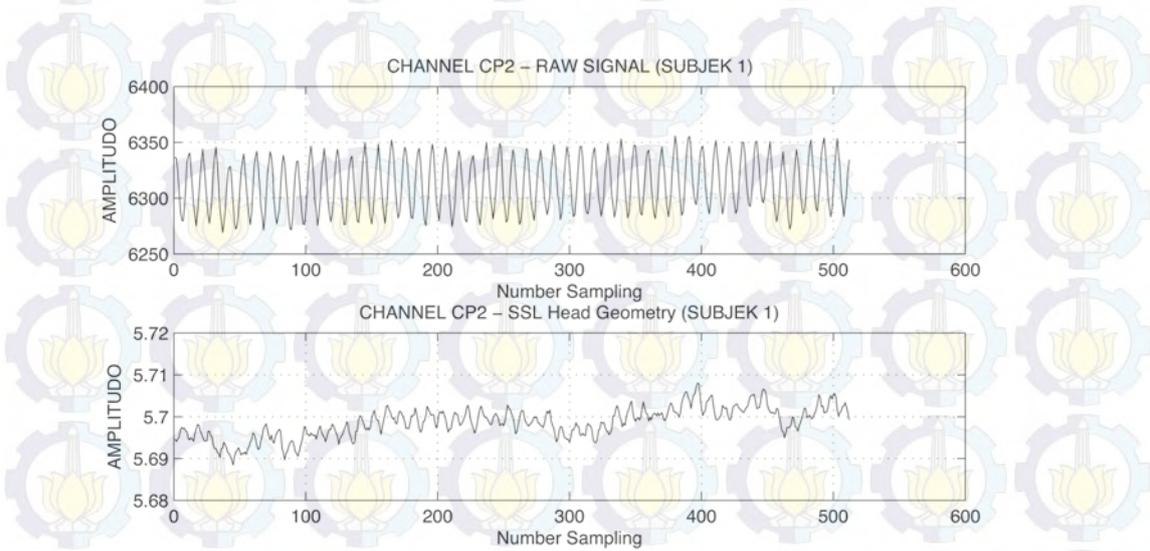
Gambar 4.3 Channel Cz – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)



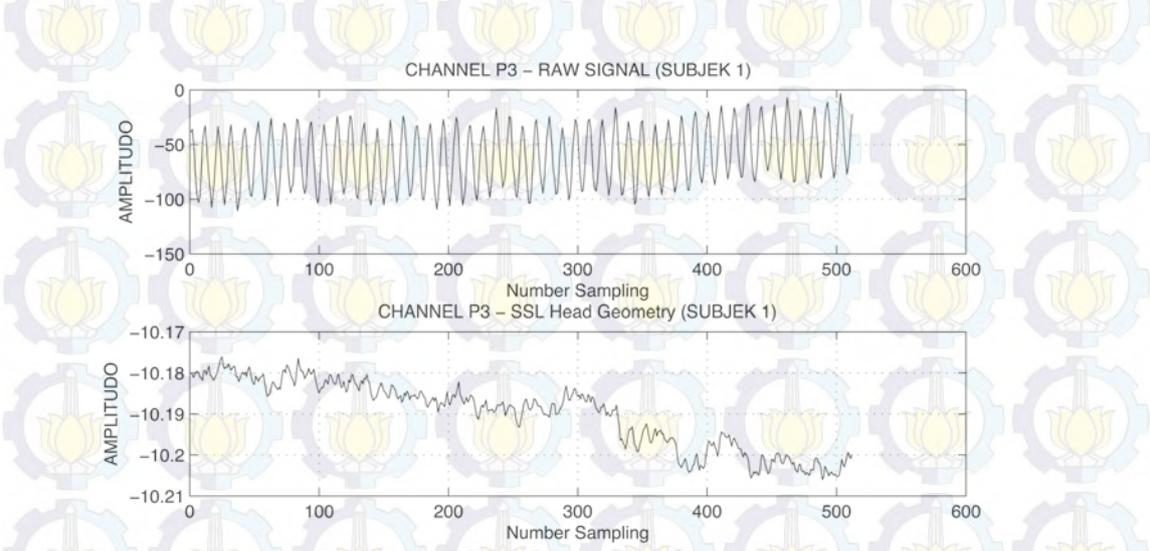
Gambar 4.4 Channel C4 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)



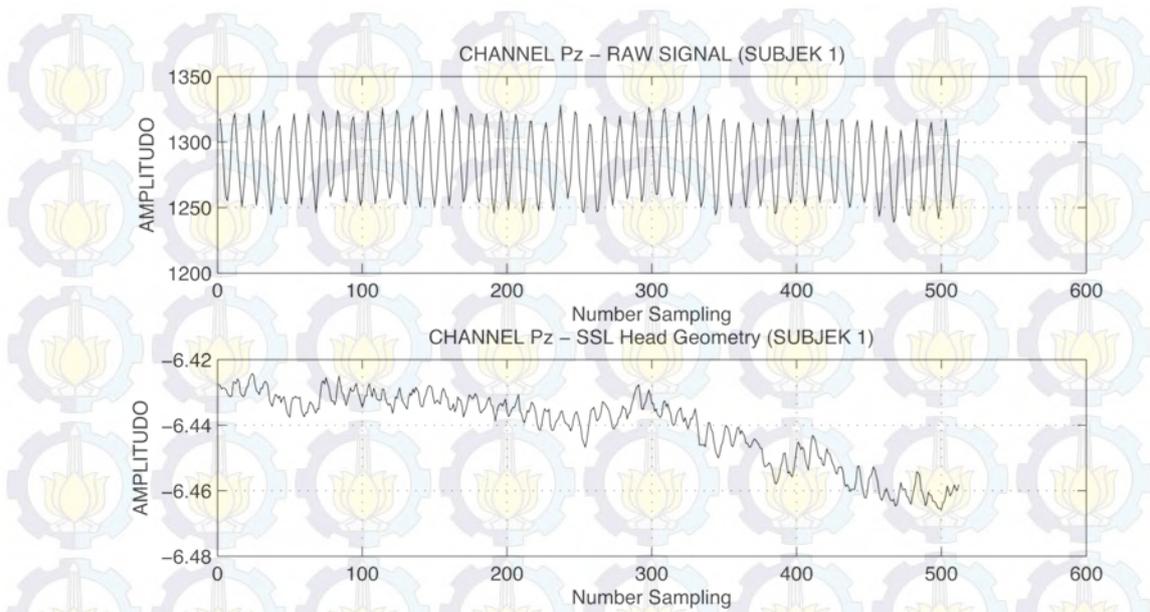
Gambar 4.5 Channel CP1 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)



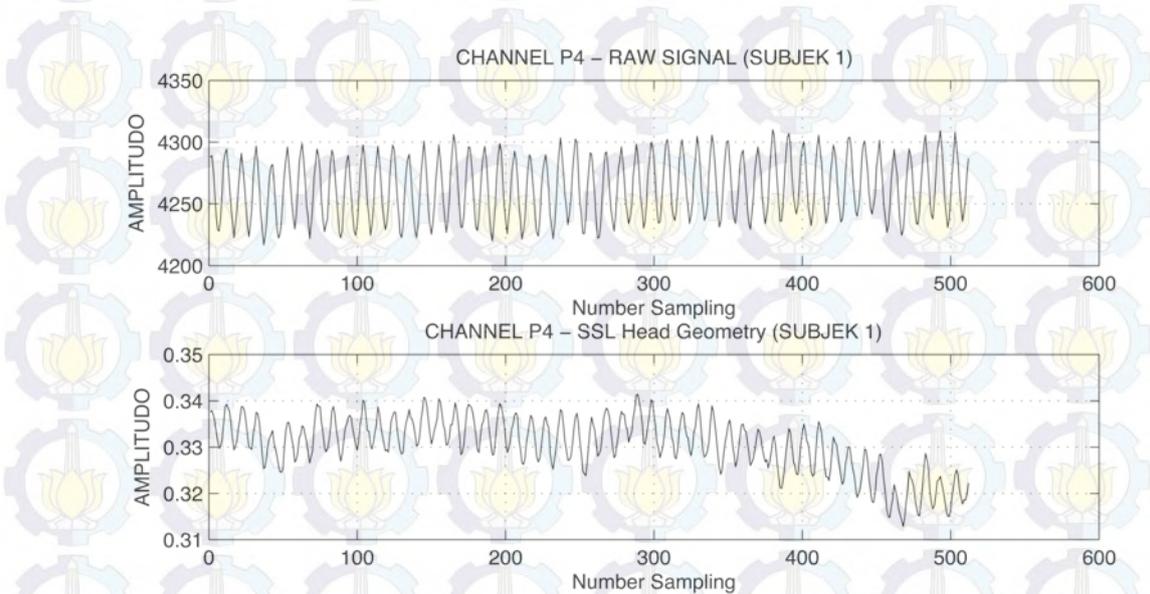
Gambar 4.6 Channel CP2 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)



Gambar 4.7 Channel P3 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)



Gambar 4.8 Channel Pz – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)



Gambar 4.9 Channel P4 – Sinyal mentah EEG (atas) dan SSL Head Geometry (bawah)

Gambar tersebut memperlihatkan bahwa nilai dari setiap *channel* lebih kecil yang menunjukkan nilai dari elektroda itu sendiri tanpa dipengaruhi oleh tegangan sekitarnya. Sebagai contoh sampling data asli dan hasil filtering dari *surface laplacian head geometry* untuk Gambar 4.1 terlihat seperti Tabel 4.1.

Tabel 4.1. Contoh data mentah EEG dengan SSL-Head Geometry.

SSL-Head Geo	Raw Data	SSL-Head Geo	Raw Data	SSL-Head Geo	Raw Data
0.614	6104.536	0.606	6182.629	0.614	6107.286
0.614	6113.754	0.605	6174.973	0.613	6114.942
0.612	6134.661	0.606	6155.348	0.612	6132.223
0.608	6160.754	0.608	6134.161	0.611	6155.348
0.608	6173.848	0.610	6119.973	0.609	6171.192
0.606	6185.567	0.613	6104.567	0.607	6183.223
0.608	6174.942	0.616	6111.473	0.608	6176.223
0.607	6156.723	0.613	6125.254	0.609	6161.379
0.608	6132.442	0.610	6149.942	0.609	6136.598
0.608	6116.254	0.608	6165.911	0.611	6122.754
0.612	6098.036	0.605	6183.504	0.611	6105.817
0.612	6103.286	0.605	6176.379	0.610	6110.661
0.610	6120.848	0.608	6159.067	0.612	6125.504
0.609	6147.379	0.610	6133.067	0.609	6150.004
0.609	6167.754	0.609	6116.504	0.606	6166.442
0.609	6181.786	0.610	6102.723	0.607	6186.285
0.607	6173.973	0.611	6109.786	0.609	6177.629
0.606	6156.629	0.610	6124.129	0.610	6164.786
0.606	6139.161	0.606	6143.629	0.610	6142.817
0.608	6122.567	0.604	6161.411	0.611	6123.317
0.610	6100.723	0.602	6179.473	0.613	6110.098
0.610	6108.473	0.601	6175.786	0.612	6110.786
0.606	6150.973	0.605	6142.692	0.611	6146.629
0.603	6166.879	0.607	6123.098	0.610	6163.317
0.602	6180.442	0.609	6104.411	0.609	6178.504
0.601	6180.973	0.609	6107.348	0.608	6181.254
0.600	6168.411	0.609	6120.629	0.609	6163.598
0.604	6144.629	0.608	6144.004	0.611	6141.567
0.606	6125.629	0.605	6171.161	0.612	6124.348
0.607	6108.848	0.600	6184.660	0.614	6111.504
0.610	6104.911	0.600	6175.786	0.617	6106.473
0.610	6117.317	0.601	6161.911	0.616	6119.504
0.610	6141.723	0.602	6140.129	0.613	6142.348
0.609	6163.192	0.604	6118.004	0.613	6162.942
0.608	6176.848	0.606	6101.567	0.610	6177.598
0.607	6180.879	0.606	6096.254	0.608	6183.285
0.608	6168.536	0.605	6108.661	0.608	6169.598
0.610	6150.754	0.605	6125.973	0.608	6151.817
0.612	6129.723	0.602	6151.411	0.610	6130.004
0.614	6113.598	0.600	6171.379	0.611	6116.004
0.616	6107.442	0.598	6180.067	0.613	6104.567
0.615	6117.348	0.598	6168.911	0.611	6110.692
0.611	6136.504	0.599	6149.348	0.609	6131.067
0.609	6159.098	0.601	6127.879	0.608	6160.317
0.608	6176.598	0.603	6110.254	0.607	6173.973
0.607	6186.410	0.606	6102.754	0.606	6186.410
0.608	6174.879	0.606	6114.942	0.609	6174.973
0.611	6155.192	0.605	6131.286	0.610	6164.067
0.612	6129.411	0.602	6155.723	0.611	6137.942
0.612	6120.192	0.600	6169.036	0.612	6121.786

Tabel 4.1 Contoh data mentah EEG dengan SSL-Head Geo (Lanjutan).

SSL-Head Geo	Raw Data	SSL-Head Geo	Raw Data	SSL-Head Geo	Raw Data
0.616	6104.754	0.598	6181.161	0.612	6109.817
0.617	6111.504	0.598	6171.411	0.612	6113.786
0.617	6129.504	0.596	6156.879	0.611	6126.754
0.615	6152.036	0.598	6134.473	0.609	6148.161
0.613	6169.098	0.602	6119.567	0.607	6167.598
0.612	6182.661	0.603	6105.942	0.605	6184.192
0.613	6171.692	0.601	6105.536	0.606	6180.348
0.612	6156.317	0.601	6118.254	0.606	6163.161
0.613	6135.161	0.599	6144.848	0.608	6138.223
0.615	6119.786	0.597	6161.692	0.610	6124.129
0.617	6107.317	0.594	6177.629	0.612	6110.504
0.618	6109.661	0.594	6172.504	0.611	6114.004
0.617	6122.817	0.597	6157.442	0.611	6121.379
0.615	6145.786	0.600	6137.067	0.610	6145.817
0.615	6163.348	0.601	6115.973	0.609	6165.973
0.614	6180.004	0.603	6104.973	0.606	6180.317
0.612	6177.598	0.603	6107.661	0.606	6178.629
0.613	6162.036	0.603	6114.223	0.606	6168.411
0.614	6139.692	0.600	6138.536	0.607	6144.348
0.614	6121.004	0.598	6156.754	0.610	6127.192
0.617	6106.817	0.595	6174.911	0.614	6111.379
0.617	6104.973	0.595	6182.129	0.616	6107.973
0.616	6113.379	0.596	6166.411	0.614	6114.286
0.616	6139.661	0.596	6145.879	0.613	6134.723
0.616	6159.942	0.599	6126.661	0.609	6163.598
0.612	6174.567	0.601	6109.911	0.606	6180.504
0.611	6178.723	0.603	6098.786	0.608	6185.598
0.612	6164.098	0.603	6109.786	0.607	6173.848
0.614	6145.098	0.602	6137.567	0.608	6151.879
0.617	6128.379	0.598	6159.286	0.610	6132.536
0.618	6109.817	0.596	6174.629	0.612	6118.129
0.620	6102.317	0.596	6184.785	0.616	6106.411
0.619	6112.192	0.596	6169.911	0.618	6114.411
0.617	6131.911	0.598	6146.723	0.618	6133.348
0.616	6158.848	0.600	6128.254	0.617	6156.973
0.614	6171.973	0.601	6116.161	0.615	6168.348
0.611	6179.504	0.602	6106.067	0.613	6183.192
0.612	6169.473	0.603	6112.786	0.613	6174.973
0.612	6153.004	0.602	6130.473	0.614	6154.223
0.613	6131.504	0.600	6153.286	0.615	6131.129
0.615	6114.723	0.600	6167.317	0.616	6117.911
0.615	6100.598	0.599	6182.129	0.617	6103.692
0.614	6108.848	0.598	6171.692	0.617	6110.129
0.614	6123.911	0.600	6153.786	0.614	6124.661
0.611	6150.848	0.601	6128.098	0.612	6151.036
0.608	6164.598	0.602	6119.598	0.609	6166.786
0.606	6179.629	0.603	6102.317	0.605	6183.317
0.606	6170.911	0.602	6106.786	0.603	6177.254
0.605	6155.192	0.602	6120.348	0.604	6158.004
0.610	6116.692	0.603	6162.161	0.613	6122.411
0.610	6102.567	0.602	6181.067	0.615	6105.848
0.611	6108.004	0.602	6172.411	0.615	6106.067

Tabel 4.1 Contoh data mentah EEG dengan SSL-Head Geo (Lanjutan).

SSL-Head Geo	Raw Data	SSL-Head Geo	Raw Data	SSL-Head Geo	Raw Data
0.612	6119.504	0.604	6158.036	0.614	6115.973
0.610	6143.379	0.605	6137.254	0.612	6140.004
0.611	6161.161	0.606	6119.692	0.609	6162.442
0.608	6182.098	0.607	6105.692	0.607	6179.098
0.606	6179.504	0.609	6107.692	0.605	6177.223
0.606	6167.848	0.608	6114.942	0.605	6163.629
0.608	6141.817	0.606	6143.442	0.608	6142.379
0.610	6122.098	0.606	6166.629	0.609	6124.723
0.615	6105.067	0.605	6176.754	0.609	6106.504
0.615	6106.473	0.604	6182.223	0.612	6103.192
0.615	6114.817	0.606	6169.973	0.611	6114.004
0.615	6139.629	0.608	6149.911	0.611	6131.223
0.613	6165.036	0.606	6128.879	0.608	6152.754
0.609	6179.192	0.607	6109.879	0.605	6170.442
0.608	6180.754	0.608	6104.629	0.604	6176.629
0.607	6167.879	0.608	6117.911	0.605	6165.192
0.607	6150.317	0.606	6136.879	0.609	6147.098
0.610	6129.098	0.606	6158.348	0.610	6126.536
0.612	6115.973	0.605	6174.973	0.610	6111.536

Untuk data mentah EEG Subject 2 dan 3 di tentukan dengan cara yang sama menggunakan tools SSLTool (Deng, 2011).

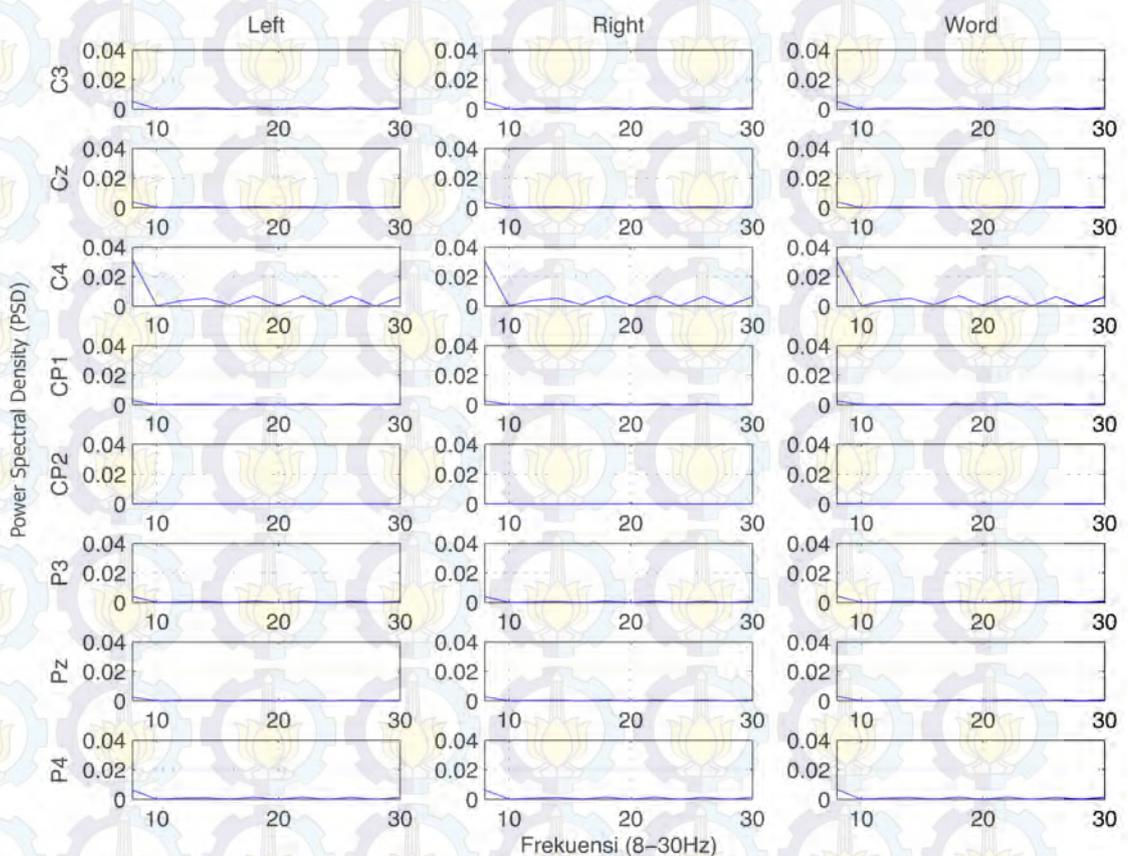
4.2.2 Proses Fitur Ekstraksi

Power Spectral Density (PSD) merupakan fitur ekstraksi yang diperoleh dari data mentah dengan menggunakan metode *welch periodogram*. Nilai PSD diperoleh dari 32 sampling yang akan di ambil pada data bagian akhir yang berukuran 2 Hz dari gelombang beta 8 Hz – 30 Hz. Sehingga dalam setiap channel akan terdapat 12 nilai psd dan total untuk seluruh channel adalah 12 x 8 channel yaitu 96 nilai psd. Gambar 4.9 memperlihatkan contoh untuk Subject 1 dari nilai rata – rata power spectral density untuk seluruh *channel* terhadap tiga kelas aktivitas mental. Nilai dari psd untuk Subject 1 dapat dilihat pada Tabel 4.2 untuk kelas *Left* (2), Tabel 4.3 untuk kelas *Right* (3) dan Tabel 4.4 untuk kelas *Word* (7). Dari Tabel tersebut terlihat bahwa setiap channel C3, Cz, C4, CP1, CP2, P3, Pz dan P4 memiliki 12 nilai PSD pada setiap 2 Hz dari 8 – 30 Hz sehingga total nilai psd atau fitur ekstraksi dari 8 *channel* adalah 96 fitur ekstraksi yang mewakili setiap kelas yaitu *Left*, *Right* dan *Word*. Setiap sampling sinyal EEG akan terdiri dari 96 nilai psd yang merupakan fitur ekstraksi yang akan digunakan pada proses

pelatihan dan pengujian data. Rata – rata nilai psd untuk subjek 2 dan 3 terlihat pada Tabel 4.5 s.d Tabel 4.10.

Tabel 4.2 Nilai rata-rata PSD Subject 1, 8 Channel untuk kelas 2 (Left)

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.004433	0.016200	0.014853	0.000964	0.002701	0.008710	0.004929	0.001734
10	0.000026	0.000095	0.000087	0.000006	0.000016	0.000051	0.000029	0.000010
12	0.000543	0.001984	0.001820	0.000118	0.000331	0.001067	0.000604	0.000212
14	0.000765	0.002798	0.002566	0.000166	0.000466	0.001504	0.000851	0.000299
16	0.000133	0.000485	0.000444	0.000029	0.000081	0.000261	0.000147	0.000052
18	0.001007	0.003679	0.003373	0.000219	0.000613	0.001978	0.001119	0.000394
20	0.000041	0.000149	0.000136	0.000009	0.000025	0.000080	0.000045	0.000016
22	0.000985	0.003599	0.003300	0.000214	0.000600	0.001935	0.001095	0.000385
24	0.000012	0.000045	0.000041	0.000003	0.000008	0.000024	0.000014	0.000005
26	0.000932	0.003405	0.003122	0.000202	0.000568	0.001830	0.001036	0.000364
28	0.000002	0.000009	0.000008	0.000001	0.000002	0.000005	0.000003	0.000001
30	0.000900	0.003288	0.003015	0.000196	0.000548	0.001768	0.001000	0.000352



Gambar 4.10 Rata – rata Nilai PSD untuk Subject 1 (Left, Right dan Word)

Tabel 4.3 Nilai rata-rata PSD Subject 1, 8 Channel untuk kelas 3 (Right).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.004124	0.015030	0.014552	0.001017	0.002513	0.008282	0.004440	0.001780
10	0.000024	0.000088	0.000085	0.000006	0.000015	0.000048	0.000026	0.000011
12	0.000505	0.001841	0.001782	0.000125	0.000308	0.001015	0.000544	0.000218
14	0.000712	0.002596	0.002513	0.000176	0.000434	0.001431	0.000767	0.000308
16	0.000123	0.000450	0.000435	0.000030	0.000075	0.000248	0.000133	0.000053
18	0.000937	0.003413	0.003304	0.000231	0.000571	0.001881	0.001009	0.000404
20	0.000038	0.000138	0.000134	0.000009	0.000023	0.000076	0.000041	0.000016
22	0.000916	0.003339	0.003233	0.000226	0.000558	0.001840	0.000986	0.000396
24	0.000012	0.000042	0.000041	0.000003	0.000007	0.000023	0.000012	0.000005
26	0.000867	0.003159	0.003058	0.000214	0.000528	0.001741	0.000933	0.000374
28	0.000002	0.000008	0.000008	0.000001	0.000001	0.000005	0.000003	0.000001
30	0.000837	0.003051	0.002954	0.000206	0.000510	0.001681	0.000901	0.000361

Tabel 4.4 Nilai rata-rata PSD Subject 1, 8 Channel untuk kelas 7 (Word).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.004423	0.015287	0.015697	0.001077	0.002512	0.008307	0.004652	0.001890
10	0.000026	0.000089	0.000092	0.000006	0.000015	0.000049	0.000027	0.000011
12	0.000542	0.001872	0.001923	0.000132	0.000308	0.001017	0.000570	0.000232
14	0.000764	0.002640	0.002711	0.000186	0.000434	0.001435	0.000803	0.000326
16	0.000132	0.000457	0.000470	0.000032	0.000075	0.000249	0.000139	0.000057
18	0.001004	0.003471	0.003565	0.000245	0.000571	0.001887	0.001056	0.000429
20	0.000041	0.000140	0.000144	0.000010	0.000023	0.000076	0.000043	0.000017
22	0.000983	0.003396	0.003487	0.000239	0.000558	0.001845	0.001033	0.000420
24	0.000012	0.000043	0.000044	0.000003	0.000007	0.000023	0.000013	0.000005
26	0.000929	0.003213	0.003299	0.000226	0.000528	0.001746	0.000978	0.000397
28	0.000003	0.000009	0.000009	0.000001	0.000001	0.000005	0.000003	0.000001
30	0.000898	0.003103	0.003186	0.000219	0.000510	0.001686	0.000944	0.000384

Tabel 4.5 Nilai rata-rata PSD Subject 2, 8 Channel untuk kelas 2 (Left).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.064696	0.072780	0.012086	0.007028	0.008539	0.015700	0.106800	0.080741
10	0.000377	0.000425	0.000071	0.000041	0.000050	0.000092	0.000623	0.000471
12	0.007924	0.008914	0.001480	0.000861	0.001046	0.001923	0.013081	0.009890
14	0.011174	0.012570	0.002087	0.001214	0.001475	0.002711	0.018445	0.013945
16	0.001935	0.002177	0.000362	0.000210	0.000255	0.000470	0.003195	0.002415
18	0.014692	0.016528	0.002745	0.001596	0.001939	0.003565	0.024254	0.018336
20	0.000593	0.000668	0.000111	0.000065	0.000078	0.000144	0.000980	0.000741
22	0.014374	0.016169	0.002685	0.001561	0.001897	0.003488	0.023728	0.017938

Tabel 4.5 Nilai rata-rata PSD Subject 2, 8 Channel untuk kelas 2 (Left) (Lanjutan).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
24	0.000180	0.000203	0.000034	0.000020	0.000024	0.000044	0.000297	0.000225
26	0.013597	0.015296	0.002540	0.001477	0.001795	0.003300	0.022446	0.016969
28	0.000036	0.000041	0.000007	0.000004	0.000005	0.000009	0.000060	0.000045
30	0.013132	0.014772	0.002453	0.001426	0.001733	0.003187	0.021678	0.016388

Tabel 4.6 Nilai rata-rata PSD Subject 2, 8 Channel untuk kelas 3 (Right).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.064696	0.072780	0.012086	0.007028	0.008539	0.015700	0.106800	0.080741
10	0.000377	0.000425	0.000071	0.000041	0.000050	0.000092	0.000623	0.000471
12	0.007924	0.008914	0.001480	0.000861	0.001046	0.001923	0.013081	0.009890
14	0.011174	0.012570	0.002087	0.001214	0.001475	0.002711	0.018445	0.013945
16	0.001935	0.002177	0.000362	0.000210	0.000255	0.000470	0.003195	0.002415
18	0.014692	0.016528	0.002745	0.001596	0.001939	0.003565	0.024254	0.018336
20	0.000593	0.000668	0.000111	0.000065	0.000078	0.000144	0.000980	0.000741
22	0.014374	0.016169	0.002685	0.001561	0.001897	0.003488	0.023728	0.017938
24	0.000180	0.000203	0.000034	0.000020	0.000024	0.000044	0.000297	0.000225
26	0.013597	0.015296	0.002540	0.001477	0.001795	0.003300	0.022446	0.016969
28	0.000036	0.000041	0.000007	0.000004	0.000005	0.000009	0.000060	0.000045
30	0.013132	0.014772	0.002453	0.001426	0.001733	0.003187	0.021678	0.016388

Tabel 4.7 Nilai rata-rata PSD Subject 2, 8 Channel untuk kelas 7 (Word).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.064604	0.072721	0.012131	0.006999	0.008509	0.015701	0.106704	0.080629
10	0.000377	0.000424	0.000071	0.000041	0.000050	0.000092	0.000622	0.000470
12	0.007913	0.008907	0.001486	0.000857	0.001042	0.001923	0.013070	0.009876
14	0.011158	0.012559	0.002095	0.001209	0.001470	0.002712	0.018429	0.013925
16	0.001933	0.002175	0.000363	0.000209	0.000255	0.000470	0.003192	0.002412
18	0.014671	0.016515	0.002755	0.001590	0.001932	0.003566	0.024232	0.018311
20	0.000593	0.000667	0.000111	0.000064	0.000078	0.000144	0.000979	0.000739
22	0.014353	0.016156	0.002695	0.001555	0.001890	0.003488	0.023706	0.017913
24	0.000180	0.000202	0.000034	0.000020	0.000024	0.000044	0.000297	0.000224
26	0.013577	0.015283	0.002549	0.001471	0.001788	0.003300	0.022425	0.016945
28	0.000036	0.000041	0.000007	0.000004	0.000005	0.000009	0.000060	0.000045
30	0.013113	0.014760	0.002462	0.001421	0.001727	0.003187	0.021658	0.016365

Tabel 4.8 Nilai rata-rata PSD Subject 3, 8 Channel untuk kelas 2 (Left).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.006689	0.021911	0.019919	0.000685	0.003719	0.004126	0.010718	0.003306
10	0.000039	0.000128	0.000116	0.000004	0.000022	0.000024	0.000063	0.000019
12	0.000819	0.002683	0.002440	0.000084	0.000456	0.000505	0.001312	0.000405
14	0.001155	0.003784	0.003440	0.000118	0.000642	0.000713	0.001851	0.000571
16	0.000200	0.000656	0.000596	0.000021	0.000111	0.000123	0.000321	0.000099
18	0.001519	0.004976	0.004524	0.000156	0.000845	0.000937	0.002434	0.000751
20	0.000061	0.000201	0.000183	0.000006	0.000034	0.000038	0.000098	0.000030
22	0.001486	0.004867	0.004425	0.000152	0.000826	0.000917	0.002381	0.000734
24	0.000019	0.000061	0.000056	0.000002	0.000010	0.000012	0.000030	0.000009
26	0.001406	0.004605	0.004186	0.000144	0.000782	0.000867	0.002252	0.000695
28	0.000004	0.000012	0.000011	0.000000	0.000002	0.000002	0.000006	0.000002
30	0.001358	0.004447	0.004043	0.000139	0.000755	0.000838	0.002175	0.000671

Tabel 4.9 Nilai rata-rata PSD Subject 3, 8 Channel untuk kelas 3 (Right).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.006755	0.022139	0.020110	0.000674	0.003740	0.004208	0.010864	0.003329
10	0.000039	0.000129	0.000117	0.000004	0.000022	0.000025	0.000063	0.000020
12	0.000827	0.002712	0.002463	0.000083	0.000458	0.000515	0.001331	0.000408
14	0.001167	0.003824	0.003473	0.000116	0.000646	0.000727	0.001876	0.000575
16	0.000202	0.000662	0.000602	0.000020	0.000112	0.000126	0.000325	0.000100
18	0.001534	0.005027	0.004567	0.000153	0.000849	0.000955	0.002467	0.000756
20	0.000062	0.000203	0.000184	0.000006	0.000034	0.000039	0.000100	0.000031
22	0.001501	0.004919	0.004468	0.000150	0.000831	0.000935	0.002414	0.000740
24	0.000019	0.000062	0.000056	0.000002	0.000010	0.000012	0.000030	0.000009
26	0.001420	0.004653	0.004227	0.000142	0.000786	0.000884	0.002283	0.000700
28	0.000004	0.000013	0.000011	0.000000	0.000002	0.000002	0.000006	0.000002
30	0.001371	0.004494	0.004082	0.000137	0.000759	0.000854	0.002205	0.000676

Tabel 4.10 Nilai rata-rata PSD Subject 3, 8 Channel untuk kelas 7 (Word).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
8	0.006769	0.022123	0.020072	0.000689	0.003741	0.004192	0.010844	0.003337
10	0.000040	0.000129	0.000117	0.000004	0.000022	0.000024	0.000063	0.000020
12	0.000829	0.002709	0.002458	0.000084	0.000458	0.000513	0.001328	0.000409
14	0.001169	0.003821	0.003466	0.000119	0.000646	0.000724	0.001873	0.000576
16	0.000202	0.000662	0.000600	0.000021	0.000112	0.000125	0.000324	0.000100
18	0.001537	0.005023	0.004558	0.000156	0.000850	0.000952	0.002462	0.000758
20	0.000062	0.000203	0.000184	0.000006	0.000034	0.000039	0.000100	0.000031

Tabel 4.10 Nilai rata-rata PSD Subject 3, 8 Channel untuk kelas 7 (Word) (Lanjutan).

Frekuensi (Hz)	C3	Cz	C4	CP1	CP2	P3	Pz	P4
22	0.001504	0.004915	0.004459	0.000153	0.000831	0.000931	0.002409	0.000741
24	0.000019	0.000062	0.000056	0.000002	0.000010	0.000012	0.000030	0.000009
26	0.001423	0.004649	0.004218	0.000145	0.000786	0.000881	0.002279	0.000701
28	0.000004	0.000013	0.000011	0.000000	0.000002	0.000002	0.000006	0.000002
30	0.001374	0.004490	0.004074	0.000140	0.000759	0.000851	0.002201	0.000677

Untuk menghitung *power spectral density* digunakan metode *pwelch periodogram* yang telah disediakan oleh Matlab dan dibangun menjadi fungsi `ITSGetPSDValueFromMat()` selain itu hanya 8 *channel* yang akan digunakan untuk menentukan kegiatan mental.

4.2.3 Reduksi Data EEG

Berdasarkan pencarian fitur ekstraksi melalui *power spectral density* diperoleh 96 vector fitur yang akan menjadi input atau masukkan pada *neural network* dimana ukuran yang cukup besar tersebut akan mempengaruhi performan dan waktu pelatihan oleh karena itu dibutuhkan proses reduksi data untuk memperkecil ukuran vector fitur. Dengan menggunakan *Principal Component Analysis (PCA)* jumlah data dapat direduksi menjadi lebih kecil dalam penelitian ini data direduksi menjadi 20 fitur ekstraksi. Dengan menggunakan fungsi *pca* yang tersedia di Matlab maka akan diambil data 1 sampai 20 yang nantinya akan menjadi masukkan pada input layer dari *neural network*. Sebagai contoh, Tabel 4.11 merupakan hasil reduksi dari *pca* yang ditampilkan untuk 20 fitur dan hanya di tampilkan 10 sampling data untuk Subject 1 training 1.

Tabel 4.11 Contoh data hasil reduksi dengan *PCA* (hanya ditampilkan 10 sampling) untuk Subject pertama Training satu.

1	2	3	4	5	6	7
0.01562	0.02797	0.03786	-0.00451	0.00531	0.01328	-0.04642
0.01566	0.02781	0.03493	0.00097	0.02318	0.00531	-0.03673
0.01575	0.02825	0.03630	0.01495	0.01686	-0.00249	-0.03955
0.01567	0.02904	0.05188	-0.00884	-0.00051	0.00098	-0.00654
0.01566	0.02847	0.05533	-0.01843	0.01508	-0.00026	0.00212
0.01570	0.02774	0.04711	0.00066	0.00771	-0.00689	-0.01994

Tabel 4.11 Contoh data hasil reduksi dengan *PCA* (hanya ditampilkan 10 sampling) untuk Subject pertama Training satu (Lanjutan).

1	2	3	4	5	6	7
0.01554	0.02932	0.05040	-0.03872	0.00491	-0.00589	-0.00785
0.01553	0.02864	0.04485	-0.03965	0.02495	0.01247	-0.00113
0.01562	0.02809	0.03754	-0.00489	0.01448	0.01570	-0.02483
0.01564	0.02834	0.04022	-0.01136	0.01318	0.02107	-0.02009
8	9	10	11	12	13	14
0.03530	0.00665	-0.00611	0.00944	0.00022	-0.02934	0.05052
-0.00965	-0.02056	0.00824	-0.03144	0.01392	-0.00729	-0.00394
-0.00949	-0.02073	-0.02610	0.00732	-0.00277	-0.00127	0.00770
0.00383	0.01026	-0.02273	0.03431	0.00016	0.01129	0.03222
-0.02842	-0.02388	0.02041	0.01351	-0.01453	0.00158	0.00842
-0.00837	-0.00729	0.01046	0.00779	0.01706	-0.00019	0.01813
-0.00019	0.00340	-0.01674	0.01536	0.03306	0.00380	0.00266
-0.02043	-0.00466	-0.00303	-0.00669	0.00961	0.01425	0.03567
-0.00156	-0.00259	-0.00796	-0.01737	0.01031	-0.02390	0.02168
-0.02704	-0.00630	-0.02305	0.00936	-0.01126	-0.00696	0.04044
15	16	17	18	19	20	
0.01795	0.00090	-0.00466	-0.01225	-0.00141	0.00775	
0.03008	-0.01379	0.01207	-0.00884	0.01319	0.00657	
0.01570	0.00530	0.00833	0.02509	0.00274	0.00526	
0.01626	-0.03420	0.00089	0.03198	-0.02404	-0.02417	
0.02611	-0.03616	-0.00173	-0.00824	-0.01217	0.02615	
0.01985	-0.00001	0.01059	-0.00386	0.02229	0.01643	
0.03411	0.00141	-0.04206	-0.02409	0.02665	0.01109	
0.02187	0.00632	-0.01088	0.01319	0.01481	0.01040	
0.00846	-0.00476	-0.02557	0.00545	0.02033	-0.02166	
0.00726	-0.00746	-0.00921	0.01924	-0.00112	0.01250	

4.2.4 Normalisasi Data

Hasil reduksi data akan memiliki batasan nilai yang beragam termasuk nilai yang berada di bawah 0 seperti terlihat pada Tabel 4.11 banyak bernilai negatif yang seharusnya di normalisasikan antara nilai 0 dan 1. Semua data baik untuk pelatihan dan pengujian perlu di normalisasikan sebagai masukan pada layer input dari *neural network*. Proses normalisasi dapat dilakukan dengan menggunakan fungsi `ITSGetPCANormFirst()` dimana inputnya adalah matriks yang akan dinormalisasikan.

BAB 5

KESIMPULAN DAN SARAN

Dari hasil penelitian dapat dijelaskan beberapa kesimpulan yang dapat diambil dan menyarankan beberapa hal penting guna pengembangan penelitian berikutnya.

5.1 Kesimpulan

Kesimpulan dari pengujian yang dilakukan pada proses klasifikasi sinyal EEG, yaitu

1. Metode FMEDO yang menggunakan bobot inersia adaptif dengan *Fuzzy Inference System* memberikan hasil yang lebih baik dibandingkan dengan metode IPSONN dimana akurasi Subjek 1 sebesar 54.20%, Subjek 2 sebesar 58.40% dan Subjek 3 sebesar 50.80% dengan rata – rata akurasinya sebesar 54.48% dimana peningkatan yang terjadi pada Subjek 1 adalah 1.26%, Subjek 2 adalah 12.63% dan Subjek 3 adalah 0.80%.
2. Data yang digunakan perlu di analisa kembali karena dalam penelitian ini data mentah dari EEG tidak mempertimbangkan masalah *noise* dan *artefek* yang dapat mengganggu sinyal asli EEG.
3. Penggunaan bobot inersia adaptif dapat menyebabkan partikel lebih mudah untuk beradaptasi dengan pergerakan partikel.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan maka peneliti menyarankan beberapa hal yang dapat digunakan untuk pengembangan sistem pada masa mendatang, yaitu

1. Hasil fitness dan kemampuan jaringan masih terlihat bervariasi sehingga nilai konvergensinya masih belum tercapai, hal ini perlu diperbaiki pada penelitian selanjutnya.
2. Waktu komputasional yang tinggi perlu dikurangi dengan mengoptimasi sistem atau menggunakan teknik parallel programming.

DAFTAR PUSTAKA

CHAI, R., LING, S. H., HUNTER, G. P., TRAN, Y. & NGUYEN, H. T. 2014. Brain-Computer Interface Classifier for Wheelchair Commands Using Network with Fuzzy Particle Swarm Optimization. *IEEE Journal of Biomedical and Health Informatics*, Vol.18, No.5, September 2014.

CHIAPPA, S. & MILLAN, J. D. R. *BCI Competitions 2003 Web Source: Data Set V description*, viewed 28 Feb. 2015 [Online]. Available: http://bbci.de/competition/iii/desc_V.html.

EBERHART, R. C. & SHI, Y. 2000. Comparing Inertia Weight and Constriction factors in Particle Swarm Optimization. *IEEE - Evolutionary Computation*, 2000. *Proceedings of the 2000 Congress on*, Vol. 1, 16 - 19 Jul 2000, p.84-88, 84-88.

LIN, C.-J. & HSIEH, M.-H. 2009. Classification of Mental Task from EEG Data Using Neural Network based on Particle Swarm Optimization *Neurocomputing*, Vol. 72, 1121 - 1130.

LING, S. H., NGUYEN, H. T., LEUNG, F. H. F., CHAN, K. Y. & JIANG, F. 2012. Intelligent Fuzzy Particle Swarm Optimization with Cross-Mutated Operation. *WCCI 2012 IEEE World Congress on Computational Intelligence*, June, 10-15, 2012 - Brisbane, Australia.

LOTTE, F., CONGEDO, M., LECUYER, A. & LAMARCHE, F. 2007. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Natural Engineering, Institute of Physics: Hybrid Open Access*, 4. <inria-00134950>.

MATHWORKS. 1998-2000. *Signal Processing Toolbox; Matlab Toolbox Release 12; Access: 04 Maret 2015* [Online]. Available: <http://radio.feld.cvut.cz/matlab/toolbox/signal/spectra7.html>.

NADERI, M. A. & MAHDAVI-NASAB, H. Analysis and classification of EEG signals using spectral analysis and recurrent neural networks. *Proceedings of the 17th Iranian Conference of Biomedical Engineering (ICBME2010)*, 3-4 November 2010, 2010.

PEARSON, K. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 2:559-572, <http://pbil.univ-lyon1.fr/R/pearson1901.pdf>.

S.N.SIVANANDAM & S.D.DEEPA *Introduction to Genetic Algorithms*, Springer Berlin Heidelberg New York.

SUBASI, A. & GURSOY, M. I. 2010. EEG Signal Classification using PCA, ICA, LDA and Support vector machines. *Expert System With Applications*, Vol. 37, p. 8659-8666.

TEPLAN, M. 2002. Fundamentals Of EEG Measurement. *Measurement in Biomedicine, Volume 2, Section 2, 2002.*

UBEDA, A., LANEZ, E., AZORIN, J. M., SABATER, J. M. & FERNANDEZ, E. 2013. Classification method for BCIs based on the Correlation of EEG maps. *Neurocomputing*, 114, 98-106.

UNDE, S. A. & SHRIRAM, R. Coherence Analysis of EEG Signal Using Power Spectral Density. Fourth International Conference on Communication Systems and Network Technologies, 2014. IEEE.

WELCH, P. D. 1967. The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms. *IEEE Transactions on Audio And Electroacoustics*, Vol. AU-15, No. 2, June 1967.

WILLIAM O. TATUM, I., DO, AATIF M. HUSAIN, M., SELIM R. BENBADIS, M. & PETER W. KAPLAN. MB, F. 2008. *Handbook of EEG Interpretation.*

WOLPAW, J. R., BIRBAUMER, N., MCFARLAND, D. J., PFURTSCHELLER, G. & VAUGHAN, T. M. 2002. Brain-Computer Interfaces for Communication and Control. *Journal Clinical Neurophysiology*, Volume 113, 767-791.

YAMAMOTO, K. & INOUE, O. 1995. New Evolutionary Direction Operator for Genetic Algorithms. *American Institute of Aeronautics and Astronautics Journal*, Vol. 33, No. 10 : Technical Notes, 1990-1993.

ZHOU, S.-M., GAN, J. Q. & SEPULVEDA, F. 2008. Classifying mental task based on features of higher-order statistics from EEG signals in brain-computer interface. *Information Sciences*, 178, 1629-1640.

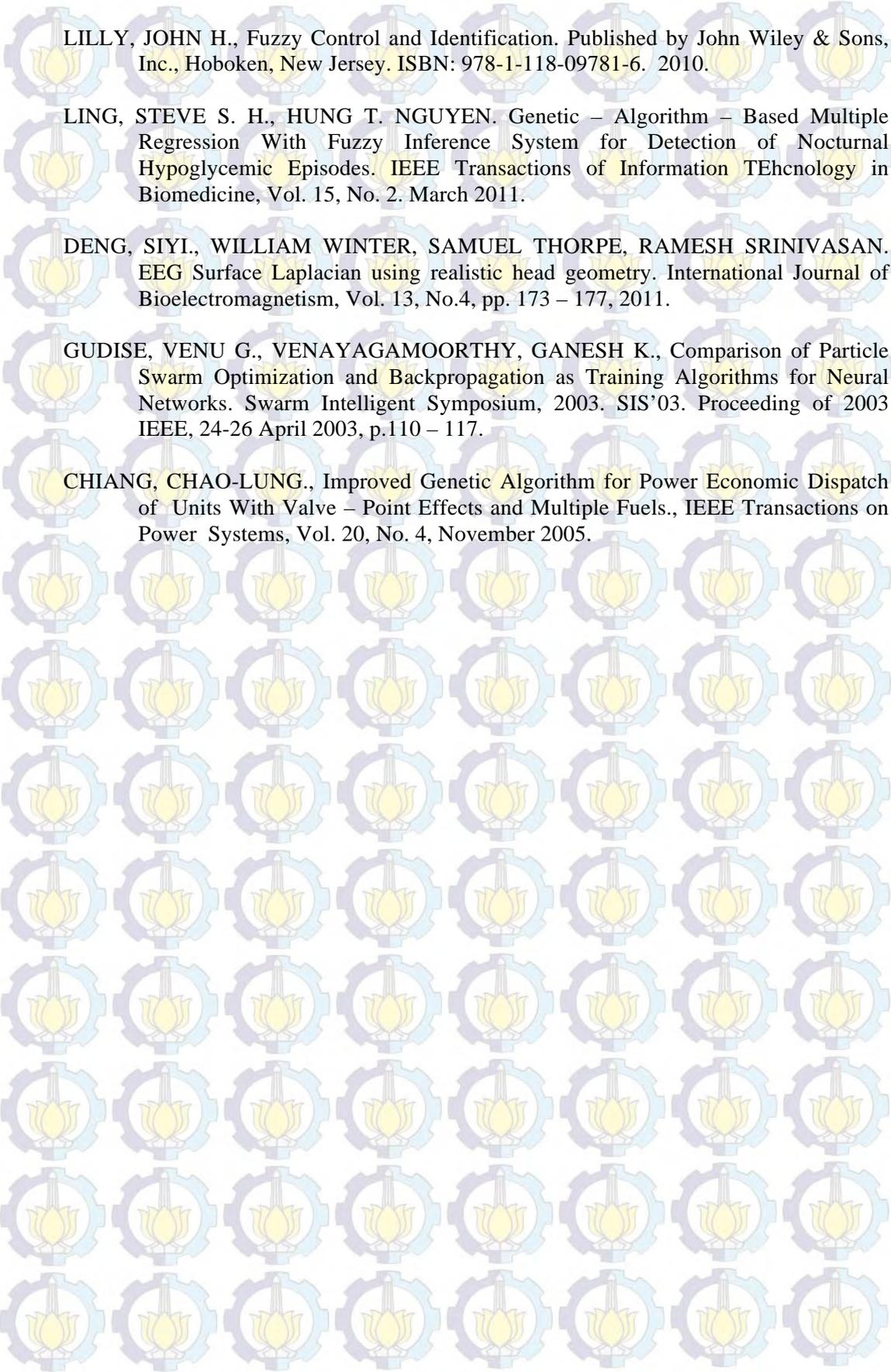
KENEDY, J. & EBERHART, R. 1995. Particle Swarm Optimization. *Neural Network, 1995. Proceedings., IEEE International Conference.* Vol. 4. Nov/Dec 1995, p.1942-1948.

LING, H. S, JIANG, F., NGUYEN, T.H., & CHAN, K.Y., 2011. Permutation Flow Shop Scheduling: Fuzzy Particle Swarm Optimization Approach. IEEE International Conferene on Fuzzy Systems, June 27-30, 2011, Taipei, Taiwan.

YADMELLAT, P., SALEHIZADEH, S.M.A., MENHAJ, M.B., A New Fuzzy Inertia Weight Particle Swarm Optimization. *IEEE Computational Intelligence and Natural Computing*, 6 -7 June 2009, p. 507 – 510, Wuhan.

COX, EARL. The Fuzzy Systems Handbook – A Practitioer’s Guide to Building, Using, And Maintaining Fuzzy System. *United Kingdom Edition published by ACADEMIC PRESS LIMITED 24-28 Oval Road, London NW1 7DX. ISBN: 0-12-194270-8. 1994.*

KIR, GEORGE J., BO YUAN. Fuzzy Sets and Fuzzy Logic – Theory and Applications. Published by Prentice Hall PTR, Prentice-Hall Inc. A Simon & Schuster Company Upper Saddle River, New Jersey 07458. ISBN: 0-13-101171-5. 1995.



LILLY, JOHN H., Fuzzy Control and Identification. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN: 978-1-118-09781-6. 2010.

LING, STEVE S. H., HUNG T. NGUYEN. Genetic – Algorithm – Based Multiple Regression With Fuzzy Inference System for Detection of Nocturnal Hypoglycemic Episodes. IEEE Transactions of Information Technology in Biomedicine, Vol. 15, No. 2. March 2011.

DENG, SIYI., WILLIAM WINTER, SAMUEL THORPE, RAMESH SRINIVASAN. EEG Surface Laplacian using realistic head geometry. International Journal of Bioelectromagnetism, Vol. 13, No.4, pp. 173 – 177, 2011.

GUDISE, VENU G., VENAYAGAMOORTHY, GANESH K., Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks. Swarm Intelligent Symposium, 2003. SIS'03. Proceeding of 2003 IEEE, 24-26 April 2003, p.110 – 117.

CHIANG, CHAO-LUNG., Improved Genetic Algorithm for Power Economic Dispatch of Units With Valve – Point Effects and Multiple Fuels., IEEE Transactions on Power Systems, Vol. 20, No. 4, November 2005.

BIODATA PENULIS



Stendy B. Sakur, dilahirkan di Manado, 28 Juli 1974. Menyelesaikan pendidikan dasar di kampung halaman, kemudian menyelesaikan Program Diploma IV di Politeknik Negeri Bandung (PolBan), Jurusan Teknik Sipil. Ketertarikan dalam bidang Teknologi Informasi membuat penulis mengikuti Program Pra-S2 Sainstek untuk melakukan alih bidang selama satu tahun di Fakultas MIPA Jurusan Matematika. Setelah menyelesaikan Program Pra-S2, melanjutkan Studi di Pasca Sarjana, Fakultas Teknologi Informasi, Jurusan Teknik Informatika dengan bidang minat *Komputasi Cerdas dan Visualisasi (KCV)* di Institut Teknologi

Sepuluh Nopember, Surabaya. Penulis merupakan Dosen Tetap di salah satu Perguruan Tinggi Negeri di Sulawesi Utara. Selain itu, menjadi penulis buku sejak tahun 2001 sampai saat ini dan salah satu buku yang banyak diminati adalah "*Pemrograman Berorientasi Objek dengan PHP 5 (Konsep dan Implementasi)*" yang di terbitkan oleh Andi Publisher. Penulis dapat dihubungi melalui email: sakur.stendy@gmail.com.