



TUGAS AKHIR - RF 1483

**PEMISAHAN BANYAK SUMBER SUARA MESIN
DARI *MICROPHONE ARRAY* DENGAN METODE
INDEPENDENT COMPONENT ANALYSIS (ICA)
UNTUK DETEKSI KERUSAKAN**

BAGUS TRIS ATMAJA
NRP. 2405 100 019

Dosen Pembimbing
Dr. Ir. Aulia Siti Aisyah, MT.
Dr. Dhany Arifianto, ST, M.Eng.

JURUSAN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2009



FINAL PROJECT - RF 1483

**MACHINES SOUND SEPARATION FROM
MICROPHONE ARRAY USING
INDEPENDENT COMPONENT ANALYSIS (ICA)
FOR FAULT DETECTION**

BAGUS TRIS ATMAJA
NRP. 2405 100 019

Supervisor
Dr. Ir. Aulia Siti Aisyah, MT.
Dr. Dhany Arifianto, ST, M.Eng.

DEPARTMENT of ENGINEERING PHYSICS
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2009

**PEMISAHAN BANYAK SUMBER SUARA MESIN
DARI MICROPHONE ARRAY DENGAN METODE
INDEPENDENT COMPONENT ANALYSIS (ICA)
UNTUK DETEKSI KERUSAKAN**

TUGAS AKHIR

Oleh :

**BAGUS TRIS ATMAJA
NRP. 2405 100 019**

Surabaya, 10 Agustus 2009

Mengetahui/Menyetujui

Pembimbing I

Pembimbing II

**Dr. Ir. Aulia Siti Aisyah, MT Dr. Dhany Arifianto, ST, M.Eng
NIP. 131 843 901 NIP. 132 206 864**

**Ketua Jurusan
Teknik Fisika FTI – ITS**

**Dr. Bambang L. Widjiantoro, ST, MT
NIP. 132 137 895**

**PEMISAHAN BANYAK SUMBER SUARA MESIN
DARI MICROPHONE ARRAY DENGAN METODE
INDEPENDENT COMPONENT ANALYSIS (ICA)
UNTUK DETEKSI KERUSAKAN**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Akustik dan Fisika Bangunan
Program Studi S-1 Jurusan Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :
BAGUS TRIS ATMAJA
Nrp. 2405 100 019

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr.Ir. Aulia Siti Aisyah, MT.(Pembimbing I)
2. Dr. Dhany Arifianto, ST, M.Eng.(Pembimbing II)
3. Katherin Indriawati, ST, MT.(Penguji I)
4. Andi Rahmadiansah, ST, MT.(Penguji II)

**SURABAYA
AGUSTUS, 2009**

PEMISAHAN BANYAK SUMBER SUARA MESIN DARI MICROPHONE ARRAY DENGAN METODE INDEPENDENT COMPONENT ANALYSIS (ICA) UNTUK DETEKSI KERUSAKAN

Nama Mahasiswa : Bagus Tris Atmaja
NRP : 2405 100 019
Dosen Pembimbing : Dr. Ir. Aulia Siti Aisyah, MT
Dr. Dhany Arifianto, ST, M.Eng

ABSTRAK

Pemeliharaan mesin-mesin di industri sangat penting untuk menjaga kelangsungan proses produksi. Analisa getaran adalah salah satu yang handal dalam mendiagnosa kondisi mesin berdasarkan pola getarannya. Perubahan getaran mesin tersebut akan mengakibatkan perubahan pola suara yang diemisikan mesin. Pada penelitian ini beberapa suara motor direkam secara bersamaan melalui *microphone array* sebagai sensornya. Tiap-tiap mikrofon menerima sinyal dari ketiga motor tersebut, sehingga sinyal output dari tiap mikrofon merupakan sinyal campuran. Blind Source Separation (BSS) merupakan teknik memisahkan sinyal campuran berdasarkan analisa kebebasan statistik (ICA) antar sumber. Dengan memperhatikan jarak dan sudut datang antara mikrofon dengan motor berbeda satu sama lain maka tiap mikrofon menerima sinyal berbeda pula. Sinyal campuran dari tiap mikrofon akan diestimasi untuk mendeteksi kerusakan motor berdasarkan analisa pola suara dengan frekuensi sesaat. Pada penelitian ini pemisahan sinyal dilakukan pada *time-domain*, *frequency-domain* dan gabungan keduanya (*multistage*) dengan algoritma *natural gradient*. Berdasarkan hasil penelitian diperoleh pemisahan sinyal terbaik pada *time-domain* serta pola frekuensi sesaat untuk deteksi kondisi mesin pada penelitian ini adalah: normal pada 51 Hz, 1000-1300 Hz, 1770 Hz dan 2650 Hz; *unbalance* pada 46 Hz, 1000 Hz, 1770 Hz dan 1990 Hz; *misalignment* pada frekuensi 46 Hz, 1772 Hz, 2600 Hz dan 3542 Hz; dan *bearing fault* terjadi pada frekuensi 73 Hz, 250 Hz dan 350 Hz.

Kata kunci: ICA, Sinyal Suara, Deteksi Kerusakan.

MACHINES SOUND SEPARATION FROM MICROPHONE ARRAY USING INDEPENDENT COMPONENT ANALYSIS FOR FAULT DETECTION

Name : Bagus Tris Atmaja
NRP : 2405 100 019
Supervisor : Dr. Ir. Aulia Siti Aisyah, MT
Dr. Dhany Arifianto, ST, M.Eng

ABSTRACT

Machine condition monitoring plays an important in industry to ensure the continuity of the process. This work presents a simple and yet, fast approach to detect simultaneous machinery faults using sound mixture emitted by machines. We developed a microphone array as the sensor. By exploiting the independency of each individual signal, we compared time-domain independent component analysis (TDICA), frequency-domain independent component analysis (FDICA) and Multi-stage ICA. In this research, four fault conditions commonly occurred in industry were evaluated, namely normal (as baseline), unbalance, misalignment and bearing fault. The results showed that the best separation process by SNR criterion was time-domain ICA. At the final stage, the separated signal was analyzed using Instantaneous Frequency technique to determine the exact location of the frequency at the specific time better than spectrogram. Result of this research indicate that normal machine condition have frequency at 51 Hz, 1000-1300 Hz, 1770 Hz and 2650 Hz; unbalance at 46 Hz, 1000 Hz, 1770 Hz and 1990 Hz; misalignment at normally frequency 46 Hz, 1772 Hz, 2600 Hz and 3542 Hz; and bearing fault at 73 Hz, 250 Hz and 350 Hz.

Keywords: ICA, Sound Signal, Fault Detection

Kata Pengantar

Alhamdulillah wa Syukurillah, Segala puji-syukur bagi Allah SWT yang menguasai langit dan bumi, dengan limpahan berkat, rahmat dan hidayahNyalah Tugas Akhir ini dapat penulis selesaikan. Tugas Akhir (TA) merupakan mata kuliah wajib di Jurusan Teknik Fisika sebagai salah satu syarat untuk memperoleh gelar sarjana.

Sinyal suara, seperti halnya sinyal-sinyal yang lain dapat dikarakterisasi berdasarkan parameternya: Amplitudo, frekuensi dan fasa. Sinyal suara ini biasanya tidak tunggal, namun bercampur dengan sinyal-sinyal suara yang lain. Untuk memisahkan antar sinyal tersebut digunakan metode pemisahan sinyal, salah satunya *blind source separation* (BSS) berdasarkan *independent component analysis* (ICA). Metode tersebut penulis aplikasikan untuk memisahkan sinyal suara banyak mesin yang selanjutnya digunakan untuk menentukan frekuensi kerusakan mesin berdasarkan perhitungan frekuensi sesaat.

Ucapan terima kasih penulis sampaikan kepada semua pihak yang telah membantu penulis dalam pelaksanaan Tugas Akhir ini. Tanpanya penulis akan mengalami kendala dalam menyelesaikan persoalan baik yang timbul di lapangan ataupun secara teoritis. Beberapa pihak tersebut, tanpa dapat menyebutkan satu persatu adalah:

1. Dr. Bambang L. Widjiantoro, ST. MT. selaku Ketua Jurusan Teknik Fisika Institut Teknologi Sepuluh Nopember.
2. Dr. Ir. Aulia Siti Aisyah MT dan Dr. Dhany Arifianto, ST. M.Eng selaku dosen pembimbing.
3. Ir. Yerri Susatio, MT selaku Kepala Lab Akustik TF ITS.
4. Bpk Augie Widyotriatmo ST, MT (Pusan, Korea) dan dr Alexander Ypma BSc, MSc (TU Delft, Netherlands) atas hardcopy & softcopy Tesis serta diskusinya.

5. Keluarga: Bapak Tris, Ibuk Ida, Mbak Itta, Dek Diah, Mas Nanang, Mbak Atul dll.
6. Teman – teman Team Akoestische, F-40, HMTF, Ninth, FUSI Ulul Albab dll.
7. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah banyak membantu dalam penyusunan Laporan Tugas Akhir ini sampai selesai.

Tak ada gading yang tak retak, begitu pula dalam penelitian dan penyusunan laporan tugas akhir ini. Masukan untuk memperbaiki laporan ini akan penulis terima dengan senang hati agar laporan selanjutnya menjadi lebih baik. Penulis juga meminta maaf kepada pihak-pihak yang merasa dirugikan baik secara sengaja maupun tidak dalam pelaksanaan penelitian ini. Seperti halnya upaya memahami perilaku sinyal fisis yang penulis lakukan dalam penelitian ini, maka upaya memahami sinyal Sang Pencipta jauh lebih penting. Terakhir, tidak ada sesuatu yang sempurna karena kesempurnaan itu hanya milikNya.

Surabaya, 10 Agustus 2009

Penulis

DAFTAR ISI

	Hal
HALAMAN JUDUL	... I
HALAMAN PENGESAHAN	... iii
ABSTRAK	... V
KATA PENGANTAR	... vii
DAFTAR ISI	... ix
DAFTAR GAMBAR	... xi
DAFTAR TABEL	... xiii
DAFTAR SIMBOL	... xv
 BAB I PENDAHULUAN	
1.1 Latar Belakang	... 1
1.2 Rumusan Permasalahan	... 3
1.3 Tujuan	... 3
1.4 Pendekatan Sistem	... 4
1.5 Sistematika Laporan	... 5
1.6 Manfaat	... 5
 BAB II DASAR TEORI	
2.1 Independent Component Analysis (ICA)	... 7
2.1.1 Teori ICA	... 7
2.1.2 Non-Gaussianity	... 10
2.1.3 Mutual Information	... 14
2.1.4 Maximum Likelihood	... 16
2.1.5 Pre-Processing	... 17
2.1.6 Optimasi ICA	... 18
2.1.7 Time-Frequency Domain ICA	... 22
2.2 Microphone Array	... 25
2.3 Frekuensi Sesaat	... 28

BAB	III	METODOLOGI	
3.1	Perekaman Multikanal	...	35
3.2	Pemisahan Sinyal Suara	...	38
3.3	Perhitungan Frekuensi Sesaat	...	43
BAB	IV	ANALISA DATA DAN PEMBAHASAN	
4.1	Analisa Data	...	47
4.2	Pembahasan	...	62
BAB	V	PENUTUP	
5.1	Kesimpulan	...	67
5.2	Saran	...	67
DAFTAR PUSTAKA			
...			
PUBLIKASI			
...			
LAMPIRAN			
A	Spesifikasi Microphone	...	76
B	Listing Code Matlab	...	77

DAFTAR GAMBAR

			Hal
Gambar	2.1	Blok Diagram Model ICA	... 8
Gambar	2.2	Distribusi Multivariate 2 Variabel GausSian	... 11
Gambar	2.3	Sinyal Gaussian dan Non-Gaussian	... 12
Gambar	2.4	Diagram Venn Mutual Information	... 15
Gambar	2.5	Sphering dan Rotating	... 17
Gambar	2.6	Proses FFT dan IFFT	... 23
Gambar	2.7	Multistage ICA	... 24
Gambar	2.8	Pola Keterarahan Microphone	... 25
Gambar	2.9	Konsep Microphone Array	... 26
Gambar	2.10	Perbandingan Lebar Window pada Spektrogram	... 29
Gambar	2.11	Karakteristik Frekuensi Window	... 32
Gambar	2.12	Definisi IFAS	... 33
Gambar	2.13	STFT dan IFAS Diskrit	... 33
Gambar	3.1	Proses Perekaman	... 35
Gambar	3.2	Diagram Blok Perekaman Multikanal	... 36
Gambar	3.3	Setting Perekaman di Ruang Kedap Lab Akustik Teknik Fisika	... 37
Gambar	3.4	Diagram Proses ICA	... 39
Gambar	3.5	Diagram Alir ICA	... 40
Gambar	3.6	ICA dalam domain waktu (TDICA)	... 41
Gambar	3.7	Proses TDICA dengan 3 input -3 output	... 42
Gambar	3.8	ICA dalam domain Frekuensi (FDICA)	... 43

Gambar	3.9	Proses FDICA dengan 3 input – 3 output	...	43
Gambar	3.10	Skema Sistem Keseluruhan	...	44
Gambar	3.11	Diagram Alir Perhitungan Frekuensi Sesaat	...	45
Gambar	4.1	Realisasi Sistem	...	49
Gambar	4.2	Histogram Sinyal Asli	...	50
Gambar	4.3	Histogram Gabungan 2 Sinyal	...	51
Gambar	4.4	Sinyal Asli Kondisi Bearing Fault dan Misalignment	...	52
Gambar	4.5	Sinyal Campuran tiap Mikrofon	...	52
Gambar	4.6	Sinyal Estimasi TDICA	...	53
Gambar	4.7	Sinyal Estimasi FDICA	...	55
Gambar	4.8	Sinyal Estimasi TFICA	...	56
Gambar	4.9	Sinyal Estimasi FTICA	...	57
Gambar	4.10	Spektrogram Sinyal Asli	...	58
Gambar	4.11	Pola Frekuensi Sesaat Sinyal kondisi Normal	...	59
Gambar	4.12	Pola Frekuensi Sesaat Sinyal Kondisi Unbalance	...	60
Gambar	4.13	Pola Frekuensi Sesaat Sinyal Kondisi Misalignment	...	61
Gambar	4.14	Pola Frekuensi Sesaat Sinyal Kondisi Bearing Fault	...	61

DAFTAR TABEL

		Hal
Tabel	2.1 Beberapa Fungsi Window	... 25
Tabel	4.1 Pengaturan Kondisi Motor 2 Mic – 2 Motor	... 48
Tabel	4.2 Pengaturan Kondisi Motor 3 Mic – 3 Motor	... 48
Tabel	4.3 Pengaturan Kondisi Motor 4 Mic – 4 Motor	... 48
Tabel	4.4 Kondisi dan Kualitas TDICA	... 53
Tabel	4.5 Perbandingan Sinyal Asli, Campuran dan Estimasi pada TDICA	... 54
Tabel	4.6 Kondisi dan Kualitas FDICA	... 55
Tabel	4.7 Kondisi dan Kualitas MSICA 1	... 56
Tabel	4.8 Kondisi dan Kualitas MSICA 2	... 57

DAFTAR SIMBOL

A	Matrik pencampur
a_{ij}	Komponen matrik pencampur
B	Blok proses (pemisahan sinyal)
E	Ekspektasi/Nilai Harapan
W	Filter pemisah (A^{-1})
w	Fungsi window
s	Sinyal sumber
x	Sinyal terukur sensor/sinyal campuran
y	Sinyal estimasi
J	Matrik Jacobian
J	Negentropy
kurt	Kurtosis
H	Entropy, Transformasi Hilbert
I	Mutual information
u	Sinyal estimasi sebelum non-linearitas
$p(\cdot)$	Peluang
$f(\cdot)$	Fungsi
μ	Besar langkah pembelajaran (learning rate)
τ	Waktu tunda

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Strategi pemeliharaan (*maintenance*) pada dunia industri mengarah pada *predictive maintenance* (PdM), yakni metode pemeliharaan yang didasarkan pada kondisi peralatan yang sedang dicek. Analisa getaran merupakan salah satu metode yang handal dalam PdM untuk monitoring kondisi mesin, khususnya untuk mesin berputar. Selain analisa getaran beberapa teknik yang lain yang digunakan untuk *predictive maintenance* yaitu *thermografi*, *tribology* dan *motor current*. Analisa getaran menganalisa pola getaran berdasarkan parameter-parameter getaran seperti frekuensi, amplitudo dan phasa. Perubahan terhadap parameter tersebut menunjukkan adanya kelainan pada mesin yang dapat diidentifikasi sebagai kerusakan mesin.

Adanya perubahan getaran menimbulkan perubahan terhadap suara yang diemisikan mesin. Dengan kata lain, perubahan suara merupakan manifestasi adanya perubahan pola getaran mesin. Seperti halnya sinyal getaran, sinyal suara juga sensitif terhadap parameternya: frekuensi, amplitudo dan phasa. Teknik analisa sinyal suara ini telah dimanfaatkan oleh G. Ramroop untuk mendeteksi kerusakan *gearbox*. Perubahan magnitudo dari spektrum menunjukkan terjadinya kerusakan. Hasil penelitian tersebut juga membuktikan bahwa bising lingkungan tidak berpengaruh secara signifikan dalam sistem monitoring gearbox tersebut [2]. Sedang P.Potočnik et.al (2005) telah mengaplikasikan penapisan psikoakustik untuk deteksi kerusakan mesin. Teknik ini berhasil mendeteksi kerusakan kompressor saat kegiatan produksi.

Pada penelitian sebelumnya, Fuad Hasan (2002) telah menggunakan metode *linear predictive coding* (LPC) untuk menganalisa sinyal suara dari motor listrik. Model LPC dibangun

atas dasar spektra sinyal lambat berubah pada suatu rentang *window* yang pendek (< 30 msec), padahal kerusakan motor listrik dapat terjadi mendadak pada saat motor beroperasi [2]. Untuk memperbaiki hal ini, Rifqi Anda (2006) memakai metode frekuensi sesaat (*Instantaneous Frequency*/IF). Dengan menggunakan IF, diperoleh frekuensi-frekuensi fundamental dimana terjadi kerusakan mesin, sedangkan model LPC hanya frekuensi-frekuensi dominannya saja. Metode frekuensi sesaat ini dipakai kembali oleh Yuniar Rahmadana (2007) untuk menganalisa kerusakan *real plant* dengan menggunakan data primer yang tidak didapatkan pada penelitian sebelumnya.

Pengukuran sinyal akustik pada penelitian-penelitian tersebut diatas dilakukan dengan medan dekat dimana *background noise* diabaikan dengan menggunakan microphone ataupun sound level meter (SLM). Selain itu, pengukuran hanya dilakukan pada satu sumber suara (motor), padahal di *real plant* mesin yang digunakan lebih dari satu mesin. Banyak mesin di industri akan mengemisikan suara yang bercampur satu sama lain. *Blind Source Separation* (BSS) adalah teknik memisahkan sinyal yang lazim dipakai pada telekomunikasi nirkabel, dimana pengguna (telpon genggam, internet nirkabel dll) selalu terhubung pada antena (sensor) terdekat. Tugas sensor adalah memisahkan antara pengguna satu dengan yang lain sehingga pesan yg dikirim tidak salah alamat. Secara garis besar, BSS merupakan teknik untuk memisahkan jumlahan dari sinyal output tanpa mengetahui karakteristik dan jumlah sumbernya. Berdasarkan statistik sinyal masukan (independensi) maka dapat ditentukan komponen-komponen sinyalnya. Metode analisa komponen independen (*Independent Component Analysis* – ICA) inilah yang sering digunakan untuk menyelesaikan permasalahan BSS.

Implementasi Algoritma BSS dapat diterapkan di beberapa bidang, misalnya akustik, spektroskopi, hiperspektral, MEG dan EEG [16]. Melalui penelitian ini penulis mencoba mengimplementasikan algoritma BSS berdasarkan analisa

komponen independen (ICA) untuk memisahkan sinyal akustik yang diemisikan beberapa mesin untuk identifikasi kerusakan, yang nantinya diharapkan dapat memberikan kontribusi untuk praktisi di bidang industri. *Microphone array* diajukan sebagai sensor yang dipakai dalam penelitian ini, dimana emisi suara akan direkam dan dianalisa lebih lanjut untuk menentukan kerusakan beberapa mesin berputar sekaligus (*multi-fault machinery*). Implikasinya asumsi medan dekat dapat diabaikan tanpa mengurangi keandalan akuisisi datanya. Seperti dalam telekomunikasi nirkabel, diharapkan hasil akhir dari penelitian ini adalah teknik yang mempermudah operator di pabrik untuk mendeteksi sekaligus menentukan secara tepat lokasi kerusakan mesin, dimana dalam skala besar, industri memakai lebih dari satu mesin berputar dan bekerja terus menerus selama proses produksi.

1.2 Rumusan Permasalahan

Dari paparan pada latar belakang diatas, maka permasalahan yang diangkat pada penelitian ini dapat dibagi menjadi dua bagian. Pertama, yang menjadi inti penelitian ini, adalah memisahkan sinyal jumlahan yang diemisikan oleh banyak mesin dari multi-mic dengan metode *independent component analysis*. Tahap terakhir adalah menganalisa kerusakan mesin berdasarkan sinyal suara yang telah terpisah dengan metode frekuensi sesaat. Untuk mengatasi permasalahan tersebut diperlukan sebuah simulator *multi-fault machinery* sebagai instrumen penelitian.

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk menjawab permasalahan yang telah dikemukakan, yakni:

1. Untuk memisahkan sinyal suara yang diemisikan oleh beberapa mesin sekaligus dan direkam dengan *microphone array*.

- Untuk menganalisa sinyal suara pisahan sehingga dapat digunakan untuk mengidentifikasi kerusakan mesin.

1.4 Pendekatan Sistem

Sebagai model awal, penelitian ini didekati dengan sistem yang di desain oleh peneliti sebagai sampel untuk dikembangkan atau digunakan secara luas. Tanpa mengurangi tujuan penelitian, maka penelitian ini didekati dengan system sebagai berikut:

- Mesin yang digunakan adalah motor listrik (pompa air) INTERDAB DB-105 dengan spesifikasi: max cap. 37 lt/min, suct head 9 mtr, disc head 24 mtr, total head 33 mtr, size 1" x 1", output 100 watt, V/Hz/Ph : 220/50/1, RPM 2850.
- Sensor yang digunakan adalah *Microphone Array* dengan beberapa mikrofon Carol tipe e845 (*Cardioid*), spesifikasi terlampir.
- Pengambilan data melalui perekaman suara moto dilakukan di Ruang Kedap Suara, Laboratorium Akustik dan Fisika Bangunan, Teknik Fisika ITS.
- Kondisi kerusakan yang diteliti adalah *unbalance*, *misalignment* dan *bearing Fault*.
- Metode yang dipakai untuk memisahkan sinyal suara mesin adalah *Blind Source Separation* berdasarkan *Time Domain ICA*, *Frequency Domain ICA* dan multistage ICA .
- Analisa pola suara dilakukan dengan teknik frekuensi sesaat.

Diharapkan dari sistem yang digunakan penelitian ini nantinya dapat dikembangkan dalam skala laboratorium atau diterapkan pada *real plant*.

1.5 Sistematika Laporan

Penulisan laporan penelitian ini mengacu pada standar penulisan Tugas Akhir di lingkungan ITS sebagai berikut.

Bab I : Pendahuluan

Bab ini menjelaskan tentang latar belakang penelitian, permasalahan yang akan dipecahkan, tujuan, pendekatan sistem, sistematika dan manfaat penelitian.

Bab II : Tinjauan Pustaka

Berisi teori-teori dasar dan penunjang penelitian yang digunakan untuk menyelesaikan permasalahan.

Bab III : Metodologi

Merupakan uraian metode beserta tahapan-tahapannya yang digunakan untuk menyelesaikan masalah penelitian.

Bab IV : Hasil dan Pembahasan

Berisi data-data hasil penelitian beserta analisa dan pembahasannya.

Bab V : Kesimpulan

Penutup laporan yang berisi kesimpulan hasil penelitian dan saran untuk penelitian selanjutnya.

1.6 Manfaat Penelitian

Penelitian ini diharapkan memberikan kegunaan atau manfaat bagi masyarakat, khususnya kalangan akademisi maupun praktisi industri dalam mengembangkan teknik baru untuk monitoring kondisi mesin. Pada tahap awal, penelitian ini diharapkan mampu mengembangkan persoalan yang terjadi pada penelitian sebelumnya. Selain itu, diharapkan penelitian ini memberikan kontribusi terhadap penelitian sejenis tentang analisa pola suara untuk deteksi kerusakan mesin yang sampai saat ini masih dalam tahap riset dan pengembangan.

BAB II

DASAR TEORI

Pada bab ini akan dibahas tentang teori yang dijadikan dasar dalam melakukan penelitian yakni metode pemisahan sinyal dengan *independent component analysis* (ICA). Konsep ICA dipaparkan dari permalahannya hingga solusi-solusi yang dapat diterapkan. Beberapa metode optimasi ICA diberikan antara lain *information maximisation* (infomax), *natural gradient* dan FastICA. Selanjutnya akan dibahas juga pemrosesan sinyal dari *microphone array*, representasi BSS pada elemen sensor. Terakhir, dipaparkan konsep frekuensi sesaat (IF) untuk menganalisa pola suara yang telah terpisahkan untuk identifikasi jenis kerusakan motor.

2.1 *Independent Component Analysis* (ICA)

2.1.1 Teori ICA

Independensi merupakan suatu kejadian yang tidak bergantung pada kejadian yang lain. Untuk suatu sumber, maka kejadian pada sumber tersebut tidak bergantung pada sumber yang lain. Dengan memanfaatkan sifat independensi statistik ini, maka pada metode ICA dapat di estimasi sinyal sumber dari sinyal tercampur.

Bila ada sumber - sumber yang saling bebas satu sama lain (*independent*) tercampur maka komponen terukur sensor dalam domain waktu dapat dimodelkan sebagai,

$$x_j(t) = a_{j1}s_1(t) + a_{j2}s_2(t) + \dots + a_{jn}s_n(t) \quad (2.1)$$

Persamaan diatas apabila ditulis dalam bentuk vektor-matriks maka menjadi:

$$\mathbf{x} = \mathbf{As} \quad (2.2)$$

atau

$$\mathbf{x} = \sum_{i=1}^n \mathbf{a}_i s_i \quad (2.3)$$

Persamaan (2.2) diatas dikenal sebagai model dari *Independent Component Analysis* (ICA). Model tersebut bersifat *general*. Komponen independen, \mathbf{s} , merupakan komponen laten, dimana komponen tersebut tidak dapat langsung diamati. Matriks pencampur, \mathbf{A} , juga tidak diketahui, hanya sinyal yang terukur saja, \mathbf{x} , yang diketahui. Komponen \mathbf{A} dan \mathbf{s} diestimasi dari sinyal terukur \mathbf{x} tersebut [10].

Tujuan dari metode ICA adalah mencari sinyal estimasi \mathbf{y} , Bila persamaan (2.3) diatas ditulis dalam bentuk matriks dua dimensi maka,

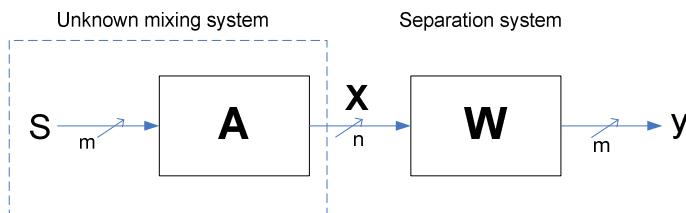
$$\begin{bmatrix} \overline{x_1} \\ \overline{x_2} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} * \begin{bmatrix} \overline{s_1} \\ \overline{s_2} \end{bmatrix} \text{ dan } \begin{bmatrix} \overline{y_1} \\ \overline{y_2} \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} * \begin{bmatrix} \overline{x_1} \\ \overline{x_2} \end{bmatrix} \quad (2.3b)$$

Sehingga sinyal estimasi dapat dinyatakan,

$$\mathbf{y} = \mathbf{Wx} \quad (2.4)$$

dimana $\mathbf{W} = \mathbf{A}^{-1}$.

Permasalahan utama metode ICA adalah mencari model filter linear \mathbf{W} , dimana \mathbf{W} merupakan invers dari matriks pencampur \mathbf{A} . Ketepatan pemilihan model filter ini akan mempengaruhi kualitas pemisahan sinyal. Beberapa metode optimasi untuk mencari nilai \mathbf{W} ini akan dijelaskan kemudian.



Gambar 2.1 Blok Diagram Model ICA

Syarat dapat diterapkannya metode ICA adalah asumsi bahwa sinyal independent secara statistik. Suatu variabel dikatakan bebas secara statistik jika memiliki *joint probability function* yang dapat difaktorkan sebagai berikut:

$$p(y_1, y_2) = p(y_1).p(y_2) \quad (2.5)$$

Persamaan (2.5) diatas dapat dipenuhi sebagai berikut:

$$p(y_1) = \int_{-\infty}^{\infty} p(y_1, y_2) dy_2 = \int_{-\infty}^{\infty} p(y_1 | y_2) p(y_2) dy_2 \quad (2.6)$$

Karena y_1 dan y_2 tidak saling bergantung maka peluang bersyarat diatas dapat dikeluarkan dari integral, sehingga menjadi,

$$p(y_1) = p(y_1 | y_2) \int_{-\infty}^{\infty} p(y_2) dy_2 = p(y_1 | y_2).1 = p(y_1 | y_2) \quad (2.7)$$

Persamaan diatas berlaku juga untuk y_2 . Sedangkan peluang bersyarat $p(y_1, y_2)$ didefinisikan sebagai,

$$p(y_1 | y_2) = \frac{p(y_1, y_2)}{p(y_2)} \quad (2.8)$$

Sehingga didapatkan persamaan (2.5) untuk dua variabel yang independen.

Secara umum ICA dapat dimodelkan secara generatif menjadi tiga model sebagai berikut:

1. Linear Noiseless ICA

Linear Noiseless ICA merupakan bentuk dasar ICA seperti pada persamaan (2.2). Model ini secara realita cukup sulit diwujudkan, namun pendekatan terhadap model ini lebih mudah dalam perhitungannya.

2. Linear Noisy ICA

Linear noisy ICA merupakan model dasar ICA ditambah dengan noise, secara matematik dinyatakan:

$$\mathbf{x} = \mathbf{As} + \mathbf{n} \quad (2.9)$$

- dimana \mathbf{n} adalah noise (m dimensi vektor acak).
3. Non-Linear ICA
Pada model ini, proses pencampuran ICA tidak linear. Fungsi penampuran non-linear yang digunakan adalah $\mathbf{f}(\cdot | \boldsymbol{\theta})$ dengan parameter $\boldsymbol{\theta}$, maka non-linear ICA dapat dimodelkan sebagai berikut,

$$\mathbf{x} = \mathbf{f}(\mathbf{s} | \boldsymbol{\theta}) + \mathbf{n} \quad (2.10)$$

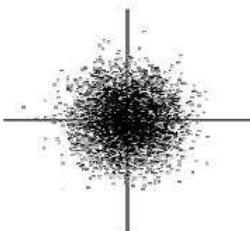
Ada beberapa cara untuk mengoptimalkan pemisahan sinyal dengan metode *ICA*: maksimalisasi non-gaussian, minimalisasi *mutual information* dan estimasi *maximum likelihood*.

2.1.2 Non-Gaussianity

Metode *Independent Component Analysis* dapat diterapkan bila asumsi bahwa sinyal independen/bebas secara statistik terpenuhi. Independensi sinyal ini ditandai dengan distribusi non-gaussian. Pembuktian bahwa sinyal dengan distribusi gaussian merupakan sinyal dependen/tak bebas dapat dijelaskan melalui formulasi berikut. Bila dua sinyal terukur, x_1 dan x_2 gaussian, tidak berkorelasi dan memiliki variansi unit maka fungsi kepadatan bersama-nya (pdf) adalah,

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{x_1^2 + x_2^2}{2}\right) \quad (2.11)$$

Jika digambarkan distribusinya (Gambar 2.2), maka kedua variabel tersebut tersebar secara simetris. Salah satu syarat untuk menentukan matriks pencampur \mathbf{A} adalah data yang diolah tidak tersebar secara simetris agar dapat dicari fungsi yang memetakan nilai matriks pencampur dalam suatu baris dan kolom tertentu.



Gambar 2.2 Distribusi Multivariante 2 Variabel Gaussian [12]

Pada teori statistik klasik, variabel acak dianggap memiliki ditribusi gaussian. Teorema limit pusat (*Central Limit Theorem*) menyatakan bahwa jumlahan dari variabel bebas (non-gaussian) mendekati ditribusi gaussian.

Model *ICA* pada persamaan (2.2) merupakan jumlahan komponen independen. Untuk memudahkan, diasumsikan semua komponen independen memiliki distribusi yang identik. Untuk mengestimasi satu komponen independen, maka digunakan kombinasi linear \mathbf{x}_i (2.3) sehingga,

$$y = \mathbf{w}^T \mathbf{x} = \sum_i w_i x_i \quad (2.12)$$

dimana \mathbf{w} adalah vektor yang akan dicari.

Jika \mathbf{w} adalah satu baris invers matrik \mathbf{A} , kombinasi linear ini sama dengan satu komponen independen. Permasalahannya adalah bagaimana menggunakan dalil pusat limit untuk mengetimasi \mathbf{w} sehingga sama dengan satu baris invers matriks \mathbf{A} . Dalam praktiknya, matrik \mathbf{w} sulit diestimasi secara tepat karena matrik \mathbf{A} pun tidak diketahui. Namun dapat dicari estimator yang memberikan pendekatan cukup baik.

Untuk melihat bagaimana prinsip estimasi ICA, digunakan variabel baru. \mathbf{z} , didefinisikan:

$$\mathbf{z} = \mathbf{A}^T \mathbf{w} \quad (2.13)$$

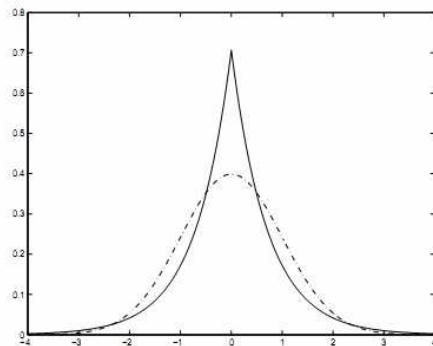
$$y = \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \mathbf{A} \mathbf{s} = \mathbf{z}^T \mathbf{s} \quad (2.14)$$

y adalah kombinasi linear dari s_i , dengan bobot diberikan oleh z_i . Karena jumlahan komponen independen lebih gaussian daripada variabel asal, $z^T s$ lebih gaussian daripada s_i , dan menjadi sedikit gaussian ketika sama dengan satu dari s_i . Pada kasus ini, hanya satu dari z_i yang tidak nol.

Oleh karenanya dapat dikatakan bahwa vektor w memaksimalkan non-gaussian dari $w^T x$. Beberapa vektor berhubungan dengan vektor z yang hanya memiliki satu komponen tidak-nol. Hal ini berarti $w^T x = z^T s$ sama dengan satu komponen independen. Memaksimalkan $w^T x$ memberikan satu komponen independen. Selanjutnya akan dibahas beberapa prinsip untuk memaksimalkan non-gaussian.

Kurtosis

Distribusi non-gaussian dapat dibedakan menjadi dua yakni *skewness* dan kurtosis. Kurtosis adalah derajat keruncingan suatu distribusi (biasanya diukur relatif terhadap distribusi normal). Ilustrasi lebih jelas dapat dilihat pada Gambar 2.3, dimana garis putus-pustus merupakan distibusi gaussian sedang garis tebal adalah non-gaussian (super-gaussian) yang memiliki nilai kurtosis.



Gambar 2.3 Sinyal Gaussian dan Non-Gaussian [12]

Kurtosis atau *fourth order cumulant* dirumuskan dengan,

$$Kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \quad (2.15)$$

Untuk y Gaussian, *fourth moment* ($E\{y^4\}$) sama dengan $3(E\{y^2\})^2$ sehingga nilai kurtosisnya nol. Untuk variabel acak non-gaussian maka nilai kurtosisnya tidak sama dengan nol.

Analisa teoritis kurtosis menjadi mudah karena mengikuti sifat linearitas: jika x_1 dan x_2 merupakan variabel acak independen, maka:

$$kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2) \quad (2.16)$$

Dan

$$kurt(\alpha x_1) = \alpha^4 kurt(x_1) \quad (2.17)$$

Dimana α adalah skalar. Untuk menggambarkan bagaimana minimalisasi atau maksimalisasi kurtosis dapat mengestimasi komponen independen, maka dapat dijabarkan:

$$y = \mathbf{z}^T \mathbf{s} = \mathbf{z}_1 \mathbf{s}_1 + \mathbf{z}_2 \mathbf{s}_2 \quad (2.18)$$

$$\begin{aligned} kurt(y) &= kurt(z_1 s_1) + kurt(z_2 s_2) \\ &= z_1^4 kurt(s_1) + z_2^4 kurt(s_2) \end{aligned} \quad (2.19)$$

Permasalahan optimasi kurtosis pada persamaan (2.19) adalah maksimalisasi fungsi tersebut pada unit lingkaran. Delfouse dan Loubaton (1995) menunjukkan bahwa fungsi tersebut akan maksimal pada titik dimana satu elemen vektor \mathbf{z} adalah nol sedang yang lain tidak nol. Pada praktiknya, optimasi dapat dimulai dengan sembarang nilai vektor \mathbf{w} , menghitungnya dimana kurtosis $y = \mathbf{w}^T \mathbf{x}$ bertambah (untuk kurtosis positif) atau berkurang (untuk kurtosis negatif) berdasarkan data sampel $\mathbf{x}(1), \dots, \mathbf{x}(T)$. Dapat pula menggunakan metode gradien atau salah satu dari perluasannya untuk mencari vektor \mathbf{w} baru [10].

Negentropy

Negentropi merupakan kuantitas teoritis informasi perbedaan entropi. Seperti halnya kurtosis, nilai negentropi ini dihitung terhadap nilai gaussian. Entropi dari variabel acak dapat diinterpretasikan sebagai tingkat informasi yang diberikan dari pengamatan variabel. Semakin acak, tidak terstruktur dan tidak terprediksi maka semakin besar nilai entropinya.

Entropi H didefinisikan untuk variabel random Y sebagai,

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i) \quad (2.20)$$

Dimana a_i adalah nilai yang mungkin untuk Y . Definisi diatas dapat diperluas untuk variabel acak dan vektor yang bernilai kontinyu, disebut dengan *differential entropy*. *Differential entropy* dari vektor acak \mathbf{y} dengan kepadatan $p(\mathbf{y})$ dirumuskan sebagai,

$$H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} \quad (2.21)$$

Dari kedua persamaan diatas maka didapatkan rumus negentropy sebagai berikut:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \quad (2.22)$$

\mathbf{y}_{gauss} merupakan variabel acak gaussian dari kovarian matrik yang sama dengan \mathbf{y} . Variabel gaussian memiliki nilai entropi terbesar diantara variabel acak untuk variansi yang sama. Nilai $J(\mathbf{y})$ ini selalu positif karena $H(\mathbf{y})$ tidak mungkin lebih dari $H(\mathbf{y}_{gauss})$, dan akan bernilai nol jika dan hanya jika \mathbf{y} memiliki ditribusi gaussian.

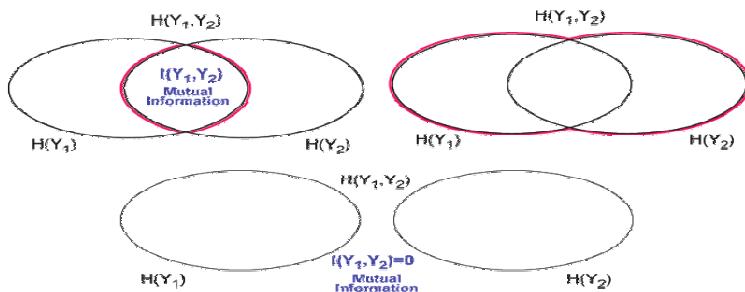
2.1.3 Mutual Information

Bila ada dua variabel acak, y_1 dan y_2 , maka untuk mengukur *mutual information* diantara keduanya didefinisikan:

$$I(y_1, y_2) = \sum_{i=1}^2 H(y_i) - H(y_1, y_2) \quad (2.23)$$

Dari persamaan (2.4) didapatkan bahwa ukuran *mutual information* bersifat simetrik. $I(y_1, y_2)$ dapat diartikan sebagai ukuran keterkaitan antara y_1 dan y_2 . Bila y_1 dan y_2 independen maka $H(y_1, y_2)=H(y_1)$ sehingga $I(y_1, y_2)=0$.

Diagram Venn ukuran *mutual information* dapat disajikan sebagai berikut:



Gambar 2. 4 Diagram Venn *Mutual Information* [4]

Agar asumsi bahwa variabel teramati adalah independen, maka nilai *mutual information* harus sama dengan nol atau mendekati nol. Untuk itu dilakukan minimalisasi ukuran *mutual information* sehingga metode *ICA* dapat diterapkan dengan tepat. Persamaan (2.23) dapat dijabarkan sebagai berikut,

$$\begin{aligned} I(y_1, y_2) &= \sum_{i=1}^2 H(y_i) - H(y_1, y_2) \\ &= \int p(y_1) \log \frac{1}{p(y_1)} dy_1 + \int p(y_2) \log \frac{1}{p(y_2)} dy_2 - \\ &\quad \int p(y_1, y_2) \log \frac{1}{p(y_1, y_2)} dy_1 dy_2 \end{aligned}$$

$$I(y_1, y_2) = \int p(y_1, y_2) \log \frac{p(y_1, y_2)}{p(y_1)p(y_2)} dy_1 dy_2 \quad (2.24)$$

Sisi kanan pada persamaan (2.24) diatas dikenal sebagai fungsi jarak Kullback-Leibler. Keuntungan dari persamaan diatas adalah pada kestabilan dan efisiensi pada pembelajarannya. Selanjutnya akan dicari \mathbf{W} yang dapat meminimalisasi fungsi tersebut. Gradien dari \mathbf{W} dapat diturunkan dengan menurunkan I terhadap \mathbf{W} , menggunakan $y=\mathbf{W}\mathbf{x}$.

$$\Delta \mathbf{W} \propto -\frac{\partial I(y_1, y_2)}{\partial \mathbf{W}} \mathbf{W}^{-T} \mathbf{W} = (I - E[\varphi(y)y^T])\mathbf{W} \quad (2.25)$$

dimana,

$$\varphi(y) = -\frac{d \log p(y)}{dy} \quad (2.26)$$

Dengan pemilihan model y yang tepat, maka karakteristik sinyal sumber dapat terkekstraksi dengan baik [20].

2.1.4 Maximum Likelihood

Pada disiplin ilmu statistika, *likelihood* dari suatu model adalah fungsi probabilitas sekumpulan data yang digunakan sebagai parameter model. Model sinyal tercampur, \mathbf{x} , diparameteri oleh matriks pencampuran, \mathbf{A} , dan densitas sinyal sumber p didefinisikan sebagai berikut,

$$p_x(x) = |\det \mathbf{W}| \prod_i p_i(s_i) \quad (2.27)$$

dimana $\prod p_i(s_i)$ merupakan fungsi likelihoood.

Jika T sampel sinyal tercampur $\mathbf{x}=[x(1), \dots, x(T)]$ diasumsikan secara bebas secara statisitik, maka $p(x_1, x_2, \dots, x_T) = p(x(1)) \dots p(x(T))$. Logaritmik *likelihood* dari \mathbf{x} ternormalisasi untuk parameter \mathbf{A} dan p adalah

$$\log p(x) = \sum_{t=1}^T \sum_{i=1}^n \log p_i(\mathbf{w}_i^T \mathbf{x}(t)) + T \log |\det \mathbf{W}| \quad (2.28)$$

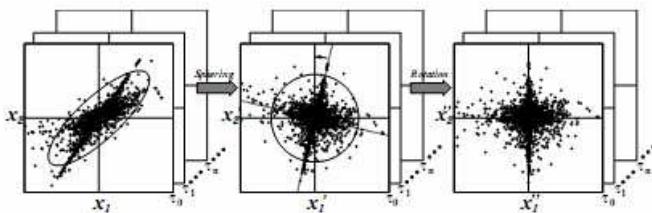
Maksimalisasi nilai *likelihood* akan memodelkan sinyal sumber $\mathbf{s}=\hat{\mathbf{W}}\mathbf{x}$ dengan syarat sumber bebas secara statistik. Estimasi maksimum nilai likelihood merupakan ide dasar dari penyesuaian kerapatan yang dapat disederhanakan dengan menyesuaikan filter pemisah \mathbf{W} sehingga fungsi distribusi kerapatan bersama (PDF) dari sinyal estimasi \mathbf{y} sedekat mungkin dengan model *a priori* distribusi model [20].

2.1.5 Pre-Processing

Sebelum data diolah dengan metode ICA terlebih dahulu data di *pre-processing*. Teknik yang digunakan untuk *pre-processing* ICA adalah *centering* dan *whitening/sphering*. *Centering* adalah tahapan yang dilakukan untuk membuang mean/removing mean, $\mathbf{m}=\mathbf{E}\mathbf{x}$. Pada tahapan ini data inputan diolah sehingga didapatkan matriks rata-rata/mean. Matriks pusat inilah yang kemudian akan diolah, sedangkan matriks rata-rata akan ditambahkan saat algoritma ICA telah selesai dilakukan [31].

Sphering merupakan suatu teknik agar data input tidak berkorelasi atau dipaksa untuk berkorelasi seminimal mungkin. Pada tahapan ini data ditransformasikan sebelum diputar (rotating) sehingga didapatkan vektor data baru dengan karakteristik :

- antara baris yang satu dan baris yang lainnya tidak saling berkorelasi
- memiliki variasi yang sama
- kovariansi matriks = matriks identitas [I]



Gambar 2.5 Spherling dan Rotating [13]

2.1.6 Optimasi ICA

2.1.6.1 Infomax

Algoritma *information maximisation* (Infomax) memaksimalkan nilai *mutual information* pada persamaan (2.22). Bobot filter pemisah \mathbf{W} akan berubah mengikuti perubahan entropy $H(\mathbf{y})$. Bila sumber memiliki distribusi super Gaussian maka dapat diasumsikan fungsi distribusi kumulatif sumber berupa fungsi sigmoid yang dinyatakan dengan,

$$f(u) = \frac{1}{1 + e^{-u}} \quad (2.29)$$

Dimana $u = \mathbf{Wx}$ merupakan output sebelum linearitas. Didefinisikan $g(u_i)$ transformasi variabel output secara sigmoid, maka aturan pembobotan dinyatakan dengan,

$$\Delta \mathbf{W} \propto \frac{\partial H(f(u))}{\partial \mathbf{W}} = E \left[\frac{\partial \ln |J|}{\partial \mathbf{W}} \right] \quad (2.30)$$

Dalam hal ini E merupakan nilai ekspektasi dari $f(u) = g[(u_1) \dots g(u_n)^T]$ dan $|J|$ nilai absolut dari matriks Jacobian:

$$J = \det \left[\frac{\partial f(u_i)}{\partial x_j} \right]_{ij} \quad (2.31)$$

Pada *stochastic gradient ascent*, nilai ekspektasi pada persamaan (2.30) dapat dihilangkan dan menghasilkan aturan pembobotan berikut,

$$\Delta \mathbf{W} \propto [\mathbf{W}^T]^{-1} + f(\mathbf{u})\mathbf{x}^T \quad (2.32)$$

Dimana nilai $f(\mathbf{u}) = [f(u_1), f(u_2), \dots, f(u_n)]^T$; elemen yang bergantung pada non-linearitas sebagai berikut:

$$f(u_i) = \frac{\partial}{\partial f(u_i)} \frac{\partial f(u_i)}{\partial u_i} = \frac{\partial}{\partial u_i} \ln \frac{\partial f(u_i)}{\partial u_i} \quad (2.33)$$

Persamaan (2.32) dapat diformulasikan kembali untuk mendapatkan aturan pembobotan filter pemisah infomax sebagai berikut [20],

$$\begin{aligned} \mathbf{W}_{n+1} &= \mathbf{W}_n - \mu \frac{\partial \hat{J}\mathbf{W}_n}{\partial \mathbf{W}} \\ &= \mathbf{W}_n + \mu \left[\mathbf{W}_n^{-T} - f(\mathbf{u})\mathbf{x}^T \right] \end{aligned} \quad (2.34)$$

Dengan μ menyatakan besar langkah positip pembelajaran.

2.1.6.2 Natural Gradient

Amari, Chicocki & Yang (1996) mengusulkan modifikasi dari metode pembelajaran Infomax dengan menggunakan *natural gradient* daripada gradien absolut $H(y)$. Aturan pembobotan filter pemisah \mathbf{W} pada metode Natural gradient diberikan dengan mengalikan aturan pembobotan infomax pada persamaan (2.32) dengan $\mathbf{W}^T\mathbf{W}$ sehingga didapatkan aturan pembelajaran baru sebagai berikut:

$$\Delta \mathbf{W} \propto \frac{\partial H(f(\mathbf{u}))}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} = (\mathbf{I} + f(\mathbf{u})\mathbf{u}^T) \mathbf{W} \quad (2.35)$$

Seperti halnya persamaan (2.35) maka persamaan diatas dapat diformulasikan kembali menjadi,

$$\begin{aligned}
\mathbf{W}_{n+1} &= \mathbf{W}_n - \mu \frac{\partial \hat{J} \mathbf{W}_n}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} \\
&= \mathbf{W} + \mu [I - f(\mathbf{u}) \mathbf{u}^T] \mathbf{W}
\end{aligned} \tag{2.36}$$

Gradien natural didapatkan dari matriks dengan mentransformasikan matriks nonsingular. Metode natural gradient ini secara umum diselesaikan dengan struktur metrik *Riemannian*.

2.1.6.3 FastICA

Metode-metode yang telah dibahas untuk memisahkan sinyal dengan metode *ICA* diatas (maksimalisasi non-gaussian, minimalisasi *mutual information*, dan maksimum likelihodd) identik satu sama lain. Pada disiplin ilmu matematika, teorema fixed-point menyatakan bahwa fungsi F minimal akan memiliki satu *nilai tetap* (pada nilai \mathbf{x} dimana $F(\mathbf{x})=\mathbf{x}$), pada kondisi beberapa kondisi sama F dapat dinyatakan secara umum.

Teorema titik tetap (*fixed-point*) ini dikembangkan untuk iterasi titik tetap. Metode iterasi titik-tetap dapat diformulasikan sebagai berikut

$$x_{n+1} = g(x) \tag{2.37}$$

Dengan g adalah fungsi yang diperoleh dari transformasi $\mathbf{x}=g(\mathbf{x})$. Metode FastICA didapatkan dari iterasi *fixed-point* ini untuk menemukan non-gaussianity dari $\mathbf{w}^T \mathbf{x}$. Vektor $\mathbf{w}^T \mathbf{x}$ merupakan proyeksi dari vektor \mathbf{w} untuk memaksimalisasi non-gaussian. Non-gaussianity tersebut diukur dengan pendekatan negentropy $J(\mathbf{w}^T \mathbf{x})$. Metode pendekatan negentropy dapat diperoleh dari *high-order moments*, yakni

$$J(y) \approx \frac{1}{12} E \{y^3\}^2 + \frac{1}{48} \text{kurt}(y)^2 \tag{2.38}$$

Variabel y pada persamaan diatas diasumsikan memiliki *mean* nol dan variansi unit. Hyvärinen (1998) mengajukan persamaan negentropy dengan pendekatan prinsip maksimum entropy secara umum sebagai berikut,

$$J(y) \approx \sum_{i=1}^p k_i [E\{G_i(y)\} - E\{G_i(v)\}]^2 \quad (2.39)$$

Dimana k_i merupakan konstanta positif dan v adalah variable Gaussian dari *zero mean* dan variansi unit. Variabel y diasumsikan memiliki *zero mean* dan variansi unit, dan G_i adalah beberapa fungsi non-kuadratik. Jika hanya ada satu fungsi non-kuadratik maka

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2 \quad (2.40)$$

Pemilihan fungsi G berikut terbukti sangat berguna untuk pendekatan negentropy diatas [10].

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u \quad (2.41)$$

$$G_2(u) = -\exp\left(-\frac{u^2}{2}\right) \quad (2.42)$$

Dimana $1 < a_1 < 2$ adalah variable konstan. Persamaan diatas dapat diturunkankan sebagai berikut

$$g_1(u) = \tanh(a_1 u) \quad (2.43)$$

$$g_2(u) = u \exp\left(-\frac{u^2}{2}\right) \quad (2.44)$$

Jika $a_1=1$ maka algoritma FastICA untuk satu unit adalah sebagai berikut:

1. Pilih inisial bobot vector \mathbf{w} (secara acak)
2. Misalkan $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T \mathbf{x})\} - E\{g'(\mathbf{w}^T \mathbf{x})\}\mathbf{w}$

$$3. \text{ Misalkan } \mathbf{w} = \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|}$$

4. Jika tidak konvergen, kembali ke 2

Konvergen artinya bahwa nilai \mathbf{w} yang lama dan baru berada pada arah yang sama.

Algoritma FastICA dapat diturunkan sebagai berikut. Maksimum pendekatan negentropy dari $\mathbf{w}^T \mathbf{x}$ dicapai pada optima tertentu $E(G(\mathbf{w}^T \mathbf{x}))$. Menurut kondisi Kuhn-Tucker, optima dari $E(G(\mathbf{w}^T \mathbf{x}))$ dibawah batasan $E\{(\mathbf{w}^T \mathbf{x})^2\} = \|\mathbf{w}\|^2 = 1$ diperoleh pada titik dimana,

$$E\{\mathbf{x}g(\mathbf{w}^T \mathbf{x})\} - \beta \mathbf{w} = 0 \quad (2.45)$$

Persamaan diatas dapat diselesaikan dengan metode Newton. Jika sisi kiri persamaan tersebut ditunjukkan dengan F , maka matriks Jacobian $JF(\mathbf{w})$ adalah,

$$JF(\mathbf{w}) = E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T \mathbf{x})\} - \beta \mathbf{I} \quad (2.46)$$

Untuk menyederhanakan inverse matrik tersebut maka dilakukan pendekatan pada bentuk persamaan (2.43). Karena data *sphered* maka pendekatan yang sesuai adalah,

$$E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T \mathbf{x})\} \approx E\{\mathbf{x}\mathbf{x}^T\} E\{g'(\mathbf{w}^T \mathbf{x})\} = E\{g'(\mathbf{w}^T \mathbf{x})\} \mathbf{I} \quad (2.47)$$

Karenanya, matrik Jacobian menjadi diagonal dan mudah untuk diinverskan. Kemudian, diperoleh pendekatan iterasi Newton sebagai berikut

$$\mathbf{w}^+ = \mathbf{w} - \frac{E\{\mathbf{x}g(\mathbf{w}^T \mathbf{x})\} - \beta \mathbf{w}}{E\{g'(\mathbf{w}^T \mathbf{x})\} - \beta} \quad (2.48)$$

Algoritma ini dapat disederhanakan dengan mengalikan kedua sisi dengan pada persamaan (2.48) dengan menyebut

$\beta - E\{g'(\mathbf{w}^T \mathbf{x})\}$. Hasilnya, setelah penyederhanaan aljabar, didapatkan iterasi FastICA (poin kedua algoritma FastICA).

2.1.7 Time-Frequency Domain ICA

Analisa sinyal dapat dilakukan dalam domain waktu dan frekuensi. Dalam domain waktu, sinyal digambarkan dengan *waveform*, dimana sumbu \mathbf{x} menyatakan waktu dan sumbu \mathbf{y} menyatakan besarnya amplitudo pada tiap waktu t . Dalam domain frekuensi suatu sinyal dapat digambarkan dengan spektrum yang menyatakan besarnya maginitudo tiap komponen spektra. Proses perubahan dari *waveform* menjadi spektrum ini dilakukan dengan mentransformasikan sinyal dengan transformasi fourier. Jika $f(t)$ merupakan sinyal dalam domain waktu maka transformasi fouriernya adalah,

$$G(j\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \quad (2.49)$$

Secara komputasi transformasi fourier dapat dilakukan dengan cepat melalui teknik FFT (fast fourier transform) yang merupakan *discrete fourier transform* (DFT) dari vektor \mathbf{x} yang dihitung sebagai berikut:

$$X(k) = \sum_{j=1}^N x(j) \omega_N^{(j-1)(k-1)} \quad (2.50)$$

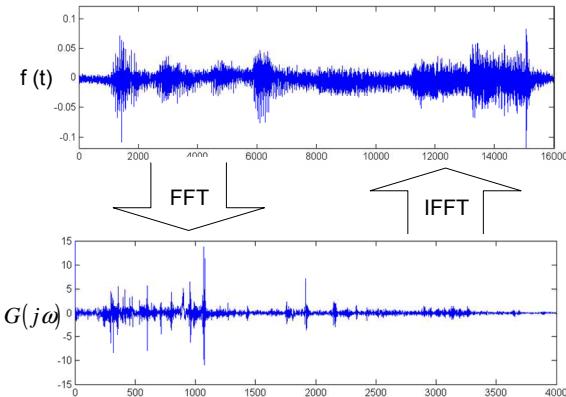
$$x(j) = (1/N) \sum_{k=1}^N X(k) \omega_N^{-(j-1)(k-1)} \quad (2.51)$$

Dimana N merupakan panjang vektor FFT dan $\omega_N = e^{(-2\pi i)/N}$.

Untuk mengembalikan dalam domain waktu maka dapat digunakan transformasi fourier balik sebagai berikut,

$$y(k,l) = \frac{1}{M} \sum_{m=1}^M u(m,l) e^{j 2\pi(m-1)(k-1)/M} \quad k=1, \dots, M \quad (2.52)$$

k -th merupakan masukan dari l -th output channel, $y(k,l)$ ekual dengan k -th point dari sejumlah M -point *Inverse Discrete Fourier Transform* (IDFT) dari l -th *channel* input [25]. Proses FFT diatas dapat digambarkan sebagai berikut,



Gambar 2.6 Proses FFT dan IFFT

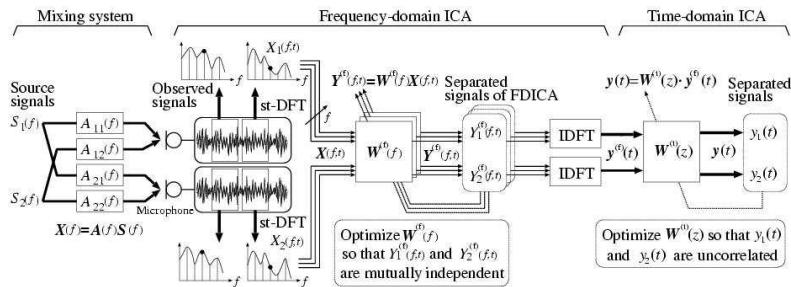
Saruwatari dkk (2002) mengusulkan untuk memakai teknik analisa domain frekuensi dan waktu secara bertahap (*multistage*) untuk menyelesaikan permasalahan BSS-ICA. Dalam dalam domain waktu, model ICA dapat dinyatakan dengan,

$$x(t) = A * s(t) \quad (2.53)$$

Sedang dalam domain frekuensi, dengan mentransfromasikan persamaan diatas maka diperoleh model ICA dalam domain frekuensi [18],

$$\hat{x}(\omega) = \hat{A}(\omega) \cdot \hat{s}(\omega) \quad (2.54)$$

TDICA dan FDICA memiliki kelebihan dan kelemahan. FDICA menyederhanakan jumlahan konvolusi menjadi jumlahan sesaat (*instantaneous*) serta mudah mencapai konvergensi dalam iterasinya, Sedangkan TDICA memiliki konvegensi yang tinggi di dekat titik optimalnya dan dapat menghilangkan sinyal *fullband* dimana asumsi independensi sinyal biasanya tertahan [17]. Skema algoritma MSICA yang diusulkan Saruwatari et.al adalah sebagai berikut.



Gambar 2.7 Multistage ICA [18]

FDICA dapat dilakukan lebih dahulu sebelum TDICA (Gambar 2.7), output dari FDICA di-inputkan pada TDICA untuk menghasilkan pemisahan sinyal yang lebih optimal, begitu pula sebaliknya. Metode pembelajaran filter pemisah, \mathbf{W} , dapat menggunakan salah satu dari metode optimasi yang telah dijelaskan sebelumnya.

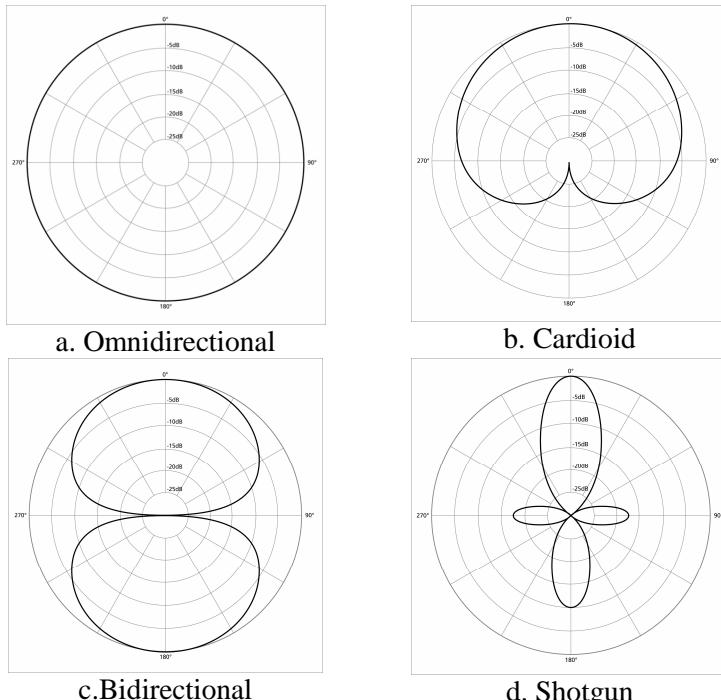
2.2 Microphone Array

Mikrofon merupakan transduser yang mengubah sinyal suara menjadi sinyal listrik. Pembagian jenis mikrofon dapat digolongkan menjadi beberapa macam berdasarkan bahanannya, keterarahannya, dan fungsinya. Berdasarkan keterarahannya mikrofon secara umum dapat dibagi menjadi empat jenis:

1. Omnidirectional
2. Unidirectional/Cardioid

3. Bidirectional
4. Shotgun

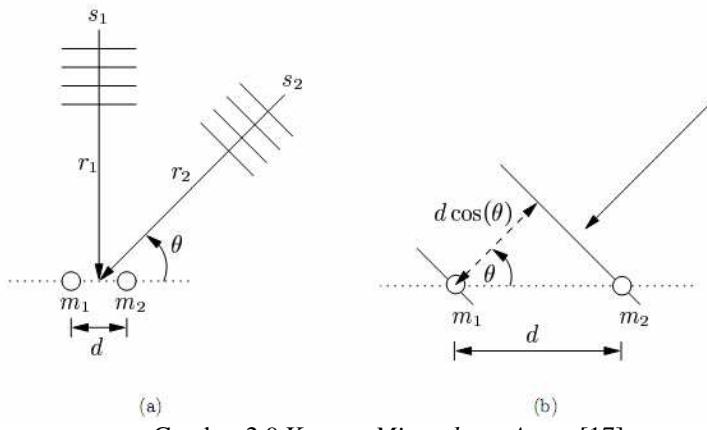
Pola keterarahan keempat microphone tersebut dapat dilihat pada gambar 2.8 dibawah.



Gambar 2.8 Pola Keterarahan Mikrofon

Mikrofon Array merupakan sekumpulan mikrofon yang terhubung menjadi satu kesatuan untuk menerima serta mentransmisikan sinyal suara. Dengan menggunakan susunan mikrofon daripada satu mikrofon, maka dapat dicapai seleksi spasial, memperkuat propagasi sumber dari arah tertentu dan melemahkan propagasi sumber dari arah lainnya [17].

Konsep pemrosesan *Microphone array* dapat di ilustrasikan sebagai berikut (Gambar 2.9). Misalkan dua sumber bunyi s_1 dan s_2 diletakkan pada jarak r_1 dan r_2 dari sepasang mikrofon (m_1 dan m_2). Jarak antara s_1 dan s_2 sama panjang terhadap kedua mikrofon (“on-axis”), sedang s_2 lebih dekat ke mikrofon m_2 daripada mikrofon m_1 (“off-axis”). Karena jarak s_1 terhadap dua mikrofon adalah sama maka akan diterima oleh kedua mikrofon pada fasa yang sama dan kombinasi antar keduanya akan memperkuat sinyal dua kali. Sedangkan jarak antar sumber s_2 dengan kedua mikrofon berbeda sehingga terjadi perbedaan fasa antar keduanya dan kombinasi antar kedua mikrofon akan saling melemahkan. Perbedaan ini bergantung pada sudut datang (θ) dan jarak antar mikrofon (d).



Gambar 2.9 Konsep *Microphone Array* [17]

Respon yang diterima mikrofon, $y[n]$ kombinasi dari m -mikrofon dengan asumsi medan jauh adalah,

$$y[n] = \sum_{m=0}^{M-1} x[n - m\tau] \quad (2.55)$$

Jika $x[n]=\delta[n]$ dan $y[n]=h[n]$, respon impuls dari system, maka respon frekuensi $h[n]$ dapat dicari dengan transformasi fourier diskrit dari persamaan (2.55) diatas,

$$\begin{aligned} H(\omega, \theta) &= \sum_{m=0}^{M-1} e^{-j 2\pi \omega(m\tau)} \\ &= \sum_{m=0}^{M-1} e^{-j 2\pi \omega \zeta(m) \left(\frac{d \cos \theta}{v}\right)} \end{aligned} \quad (2.56)$$

Dimana terlebih dahulu dihitung waktu tunda, τ , dari kecepatan suara,

$$\tau = \frac{d \cos(\theta)}{v} \quad (2.57)$$

Dari persamaan (2.56) diatas terlihat bahwa respon frekuensi bergantung pada jumlah mikrofon M, jarak d, frekuensi spectral ω , dan sudut datang θ .

Susunan sejumlah mikrofon pada dasarnya menyuplik medan suara pada titik yang berbeda pada suatu ruang. Sebagai akibatnya, pada pemrosesan *array* akan terjadi masalah *spatial aliasing* jika sinyal yang dicuplik sangat lambat. Ketika *spatial aliasing* terjadi maka *array* tidak bisa membedakan diantara sudut datang untuk frekuensi yang ingin ditangkap dan arah yang tidak diinginkan. Untuk menghindari *spatial aliasing* ini maka digunakan aturan sebagai berikut,

$$d < \frac{\lambda_{\min}}{2} \quad (2.58)$$

Dimana λ_{\min} adalah panjang gelombang minimum dari obyek sumber suara (sebanding dengan frekuensi maksimum). *Spatial Aliasing* juga dapat dihindari dengan menggunakan sekumpulan *array* yang harmonik atau terdistribusi secara random [17].

2.3 Frekuensi Sesaat

Suatu sinyal dapat ditampilkan dalam domain waktu dan domain frekuensi. Pada kedua domain inilah (waktu-frekuensi) biasanya analisa sinyal dilakukan. Pada domain waktu suatu sinyal dapat dituliskan,

$$x(t) = a \cos \omega_0 t \quad (2.59)$$

Dimana,

$$\omega_0 = 2\pi f_i \quad (2.60)$$

Maka persamaan (2.42) dapat dituliskan kembali:

$$x(t) = a \cos \left(\int_0^t 2\pi f_i(t) dt + \theta \right) \quad (2.61)$$

Perbedaan waktu tempuh sinyal menyebabkan adanya fase, yakni sudut ketika bergerak harmonis (cosinus), secara matematik dapat dituliskan,

$$\phi(t) = \int_0^t 2\pi f_i(t) dt + \theta \quad (2.62)$$

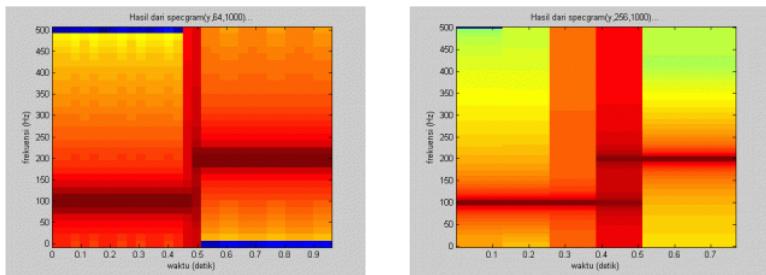
Frekuensi sesaat didefinisikan sebagai turunan dari fase, sehingga frekuensi sesaat dapat diturunkan dari persamaan (2.62) diatas,

$$f_i(t) = \frac{1}{2\pi} \frac{d\phi(t)}{dt} \quad (2.63)$$

Konsep frekuensi sesaat diatas dapat dicari/diturunkan dari Transformasi Hilbert. Bila diberikan sinyal real $s(t)$ dan sinyal analitik kompleks $z(t)$ maka Transformasi Hilbert untuk bagian imajinernya adalah,

$$z(t) = s(t) + jH[s(t)] = a(t) e^{j\phi(t)} \quad (2.64)$$

Dengan $H[s(t)]$ merupakan transformasi Hilbert dari $s(t)$. Sifat sinyal analitik adalah tidak mempunyai komponen negatif dalam domain frekuensi, konsekuensinya energi sinyal analitik adalah dua kali energi sinyal real. IF dikembangkan sebagai cara baru untuk meminimalisasi keterbatasan representasi/visualisasi parameter sinyal dari spektrogram yang berawal dari upaya dekomposisi sinyal dari transformasi Fourier yang bersifat linear. Ada semacam ketidak pastian dimana resolusi frekuensi bisa diperoleh secara tepat tapi informasi letak frekuensi terhadap waktu hilang pada *narrowband spektrogram*, begitu pula sebaliknya pada *wideband spektrogram* [6]. Visualisasi dari ketidakpastian tersebut dapat dilihat pada gambar 2.10 berikut.



a. Wideband

b. Narrowband

Gambar 2.10 Perbandingan Lebar Window pada Spektrogram [31]

Pada sinyal analitik (2.64), amplitudo $a(t)$ dan fase $\phi(t)$ diberikan oleh,

$$a(t) = \sqrt{s(t)^2 + (H[s(t)])^2} \quad (2.65)$$

Dan,

$$\phi(t) = \tan^{-1} \left(\frac{H[s(t)]}{s(t)} \right) \quad (2.66)$$

Dimana $a(t)$ merupakan *instantaneous amplitude* atau amplitudo sesaat sinyal, disebut juga sebagai *envelope* dan $\phi(t)$ merupakan

fase sinyal yang akan diturunkan. Transformasi Hilbert didapatkan dengan menyelesaikan nilai pokok pada integral berikut,

$$H[s(t)] = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{s(t)}{t - \tau} dt \quad (2.67)$$

Maka frekuensi sesaat dapat dihitung sebagai turunan dari fase (2.66) seperti halnya pada persamaan (2.63).

Frekuensi sesaat juga dapat diturunkan dari *Short Time Fourier Transform (STFT)*. STFT merupakan modifikasi dari trasnsformasi fourier,yakni teknik memecah suatu sinyal menjadi komponen-komponen sinusoid-nya dalam domain frekuensi. STFT digunakan untuk menganalisa sinyal yang bersifat non-stasioner, membagi menjadi bagian-bagian kecil yang diasumsikan stasioner. Pembagian ini dilakukan dengan fungsi *window* dengan lebar tertentu yang kemudian digeser dan dikalikan dengan sinyal asli untuk memperoleh sinyal stasioner yang sempit. Setelah sinyal-sinyal tersebut dibagi menjadi sinyal-sinyal stasioner yang sempit maka sinyal tersebut ditransformasikan dengan tranformasi fourier untuk memperoleh STFT. Pada beberapa aplikasi STFT diperoleh dari sekumpulan bandpass filter.

STFT dari suatu sinyal $x(t)$ dapat dituliskan,

$$STFT_s(\omega, t) = \int_{-\infty}^{\infty} w(\tau - t) x(\tau) e^{-j\omega\tau} d\tau \quad (2.68)$$

Dengan mengeluarkan $e^{j\omega t}$, persamaan diatas dapat dituliskembali menjadi,

$$\begin{aligned} X(\omega, t) &= e^{-j\omega t} \int_{-\infty}^{\infty} w(\tau - t) x(\tau) e^{-j\omega(\tau-t)} d\tau \\ &= e^{-j\omega t} G(\omega, t) \end{aligned} \quad (2.69)$$

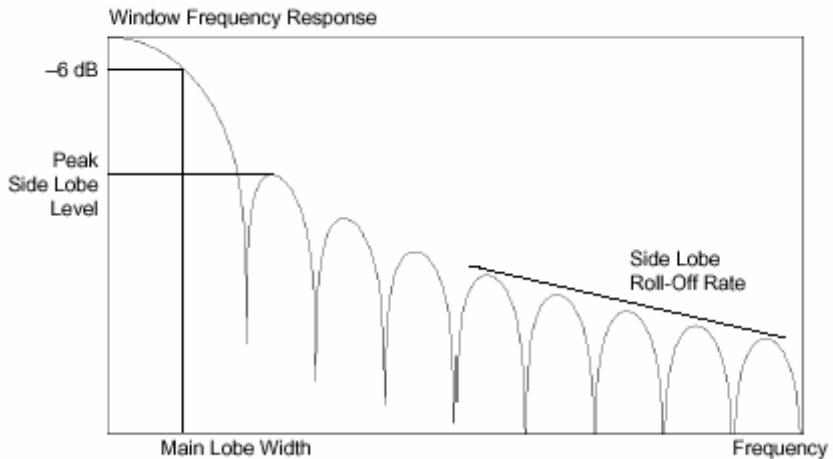
Dimana $w(t)$ merupakan fungsi window. Proses *windowing* pada dasarnya ada menyatukan spektrum sinyal asli dengan sinyal input. Ada beberapa macam window yang digunakan, Rectangular, Bartlet, Hanning, Hamming, Blackman dll. Persamaan fungsi window tersebut adalah sebagai berikut,

Tabel 2.1 Fungsi Window

No	Tipe	Fungsi Window
1	Rectangular	$W_{rec}(n) = 1$
2	Bartelt	$W_{tri}(n) = 1 - \frac{ n }{M}$,
3.	Hanning (hann)	$W_{hann}(n) = 0.5 + 0.5 \cos\left(\frac{n\pi}{M}\right)$
4.	Hamming	$W_{ham}(n) = 0.54 + 0.46 \cos\left(\frac{n\pi}{M}\right)$
5.	Blackman	$W_{black}(n) = 0.42 + 0.5 \cos\left(\frac{n\pi}{M}\right) + 0.08 \cos\left(\frac{2n\pi}{M}\right)$

Untuk semua $-M < n < M$.

Pemilihan window bergantung pada karakteristik fungsi window terhadap sinyal input. Karakteristik *side lobe* dari *window* secara langsung akan berpengaruh pada area dimana komponen frekuensi yang berdekatan akan bias (bocor) ke frekuensi *bin* yang terdekat. Respon *side lobe* dari sinyal sinusoidal yang kuat dapat mengakibatkan terjadinya *overpower* terhadap respon *main lobe* dari sinusoidal terdekat yang lemah. Karakteristik penting lainnya dari *spectral window* adalah lebar dari *main lobe*. Resolusi frekuensi dari sinyal yang diwindow dibatasi oleh lebar *main lobe* dari *spectral window*. Selain respon frekuensi, *spectral leakage* dan akurasi amplitudo juga dipertimbangkan dalam pemilihan fungsi window. Karakteristik respon frekuensi suatu window dapat ditampilkan pada gambar berikut.



Gambar 2. 11 Karakteristik Frekuensi Window [32]

Jika transformasi fourier dari window $w(t)$ merupakan low-pass filter maka $G(\omega,t)$ akan menjadi keluaran dari bandpass filter yang memiliki impuls respon $w(-t)e^{j\omega t}$. Band pass filter ini akan memiliki frekuensi yang digeser terhadap transformasi fourier $w(t)$ dan memiliki frekuensi tengah ω .

Frekuensi sesaat pada (ω,t) didefinisikan sebagai turunan fasa tehadap waktu dan dihitung dengan menggunakan bandpass filter dari STFT.

$$\lambda(\omega,t) = \frac{\partial}{\partial t} \arg[G(\omega,t)] \quad (2.70)$$

Persamaan tersebut menunjukkan informasi frekuensi pada waktu sesaat. Dari persamaan tersebut didapatkan,

$$\begin{aligned} \lambda(\omega,t) &= \frac{\partial}{\partial t} \arg[e^{j\omega t} (\omega,t)] \\ &= \omega + \frac{\partial}{\partial t} \arg[X(\omega,t)] \end{aligned} \quad (2.71)$$

Maka bentuk frekuensi sesaat dapat dituliskan,

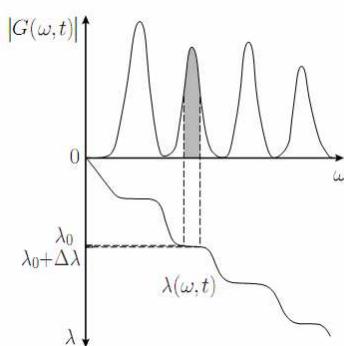
$$\frac{\partial}{\partial t} X(\omega, t) = \int_{-\infty}^{\infty} -w'(\tau - t) e^{-j\omega\tau} x(\tau) d\tau \quad (2.72)$$

Persamaan (2.72) diatas merupakan STFT jika fungsi window $w(t)$ diganti dengan $-w'(t)$. Dari persamaan (2.69) didapatkan $|G(\omega, t)| = |X(\omega, t)|$, sehingga $|G(\omega, t)|$ ekivalen dengan STFT amplitudo spektrum. Frekuensi sesaat amplitudo spektrum (Instantaneous amplitude spectrum/IFAS) pada frekuensi sesaat λ_0 dapat didefinisikan sebagai berikut,

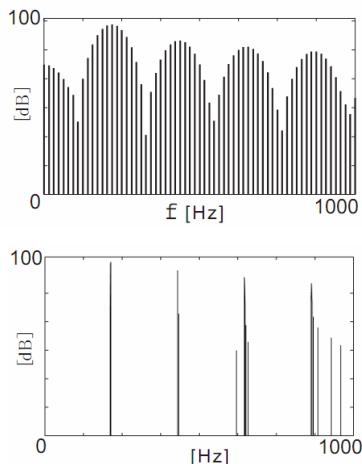
$$S(\lambda_0, t) = \lim_{\Delta\lambda \rightarrow 0} \frac{1}{\Delta\lambda} \int_{\Omega_0} |G(\omega, t)| d\omega \quad (2.73)$$

Dimana $\Omega_0 = \{\omega \mid \lambda_0 \leq \lambda(\omega, t) \leq \lambda_0 + \Delta\lambda\}$.

Definisi IFAS diatas dapat digambarkan 2.12, sedangkan pada gambar 2.13 diilustrasikan STFT diskrit pada sebelah atas dan IFAS diskrit pada gambar sebelah bawah. Sebagaimana terlihat, IFAS menunjukkan frekuensi yang lebih tajam daripada STFT.



Gambar 2.12 Definisi IFAS



Gambar 2.13 STFT & IFAS Diskrit

Frekuensi sesaat merupakan frekuensi kurva sinus yang sesuai dengan sinyal pengukuran. Nilai frekuensi ini bukan berarti frekuensi dalam *cycle* per satuan waktu, karena dalam frekuensi sesaat ada nilai negatifnya $(-\infty, \infty)$. Karena sifat ketidak-elastisan dan tipe kerusakan yang lain dapat dianggap berpengaruh terhadap komposisi frekuensi dari respon, maka frekuensi sesaat ini diharapkan sangat berguna dalam karakterisasi kerusakan.

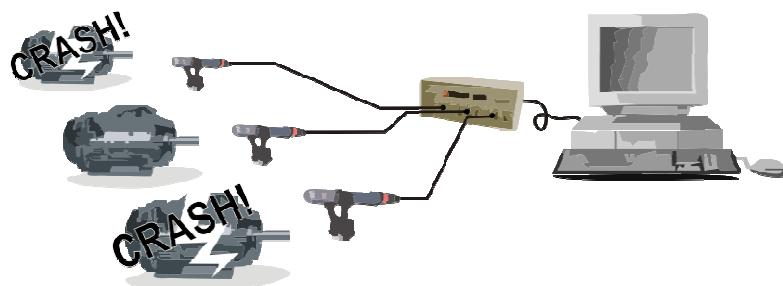
BAB III

METODOLOGI

Pada bab ini akan dipaparkan metode yang digunakan dalam penelitian. Bab ini dibagi menjadi tiga sub-bab yakni: Proses perekaman multikanal, Pemisahan sinyal suara dan Perhitungan frekuensi sesaat. Melalui studi literatur diperoleh metode yang sesuai untuk penelitian ini. Perekaman dilakukan secara multikanal dengan soundcard MIMO. Pemisahan sinyal suara dilakukan dengan pendekatan analisa komponen independen (ICA) yakni dalam domain frekuensi (FDICA) dan dalam domain waktu (TDICA), dan gabungan keduanya. Setelah terpisahkan, sinyal estimasi digunakan untuk mendeteksi kerusakan mesin dengan metode frekuensi sesaat (IF) yang dapat ditelusuri dari *spectrogram* sinyal.

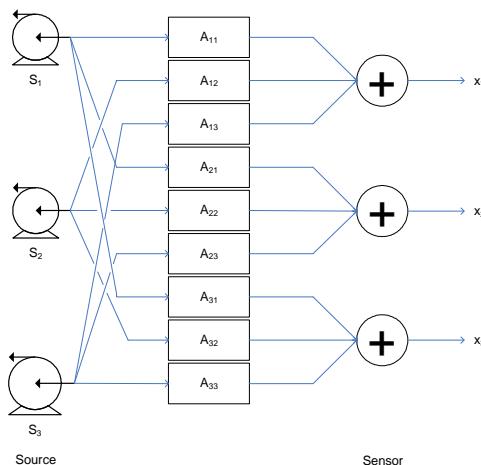
3.1 Proses Perekaman Multikanal

Proses perekaman merupakan tahapan penting penelitian ini, karena data penelitian didapatkan dari proses perekaman sinyal suara mesin. Skema proses perekaman suara mesin pada penelitian ini seperti ditunjukkan oleh Gambar 3.1



Gambar 3.1 Proses Perekaman

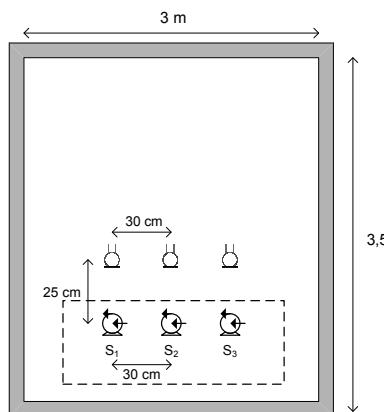
Proses perekaman dilakukan pada tiap kanal. Jumlah sensor yang digunakan harus memenuhi kriteria *miophone array* dimana $m \geq n$, pada penelitian ini digunakan dua sensor-dua sumber, tiga sensor-tiga sumber dan empat sensor-tiga sumber. Pada tiap-tiap kanal dilakukan proses perekaman secara simultan/bersamaan dan tiap file disimpan dengan ekstensi **.wav** (PCM). Frekuensi sampling pada proses perekaman adalah 11250 Hz, *mono* dan 16 bits. Frekuensi sampling ini diambil dengan memperhatikan bahwa frekuensi maksimum motor adalah 5000 Hz [2,14], untuk memenuhi kriteria Nyquist maka frekuensi sampling harus lebih besar sama dengan dua kali frekuensi maksimum. Mikrofon yang digunakan pada penelitian ini bertipe cardioid agar mampu menangkap suara dari ketiga sumber. Mikrofon tersebut terhubung dengan *breakout box* dari PCI soundcard (Delta M-Audio 66) yang memiliki empat input dan empat output. Proses perekaman pada Gambar 3.1 di atas dapat dijabarkan dalam diagram blok berikut.



Gambar 3.2 Diagram Blok Perekaman Multikanal
(3 sumber 3 sensor)

Seperti terlihat pada Gambar 3.1 sebelumnya, tiap mikrofon akan menerima sinyal dari ketiga sumber. Karena jarak antar sumber satu dan sumber lainnya berbeda terhadap mikrofon maka akan terjadi efek multisumber yang menyebabkan terjadinya kchselasi fasa yang diakibatkan oleh perbedaan jarak tempuh dan sudut datang. Perbedaan sifat sinyal pada sensor pertama dan sensor ke- n (dimana n adalah integer dan jumlah sensor) dipilah berdasarkan sifat statistik sinyal (independensi).

Salah satu fenomena yang harus dihindari pada penggunaan microphone array ini adalah adanya *spatial aliasing*. Untuk menghindari hal ini maka digunakan aturan seperti pada persamaan (2.57) dimana jarak antar mic harus lebih kecil dari setengah panjang gelombang suara. Panjang gelombang suara minimum sama artinya dengan frekuensi suara maksimum, dari sini dapat ditentukan jarak maksimum antar mikofon. Dengan mengambil frekuensi maksimum motor adalah 5000 Hz diperoleh jarak antar mikrofon maksimal 34,3 cm. Pada penelitian ini jarak antar mic diambil 30 cm dan jarak mic dengan sumber 25 cm. Skema perekaman yang dilakukan di ruang kedap Lab Akustik TF (waktu dengung, $RT=0,74$ detik) pada penelitian ini ditujukkan pada gambar di bawah.



Gambar 3.3 Setting Perekaman di Ruang Kedap Lab Akustik TF

3.2 Pemisahan Sinyal Suara

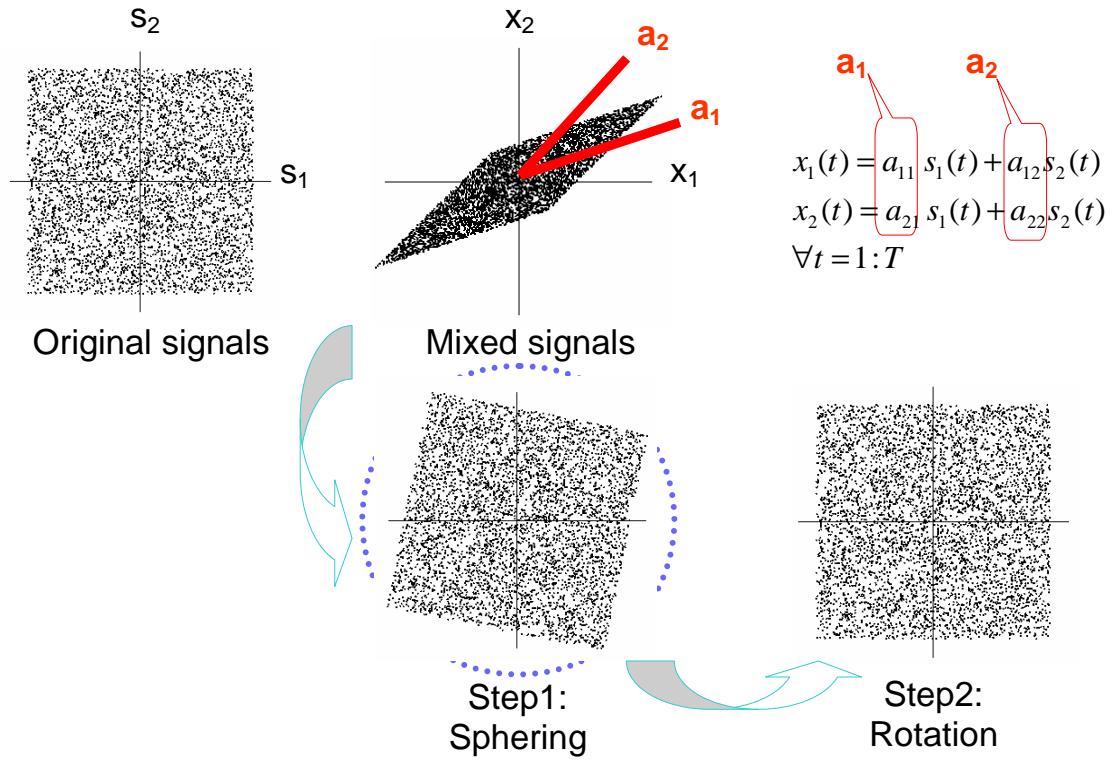
Pemisahan sinyal suara mesin pada penelitian ini dilakukan dengan empat metode. Keempat metode tersebut merupakan variasi domain serta gabungan antar domain secara bertingkat (multistage). Keempat metode yang digunakan pada penelitian ini adalah sebagai berikut:

1. Time Domain ICA (TDICA)
2. Frequency Domain ICA (FDICA)
3. Time-Frequency ICA (Multistage ICA 1/MSICA 1)
4. Frequency-Time ICA (Multistage ICA 2/MSICA 2)

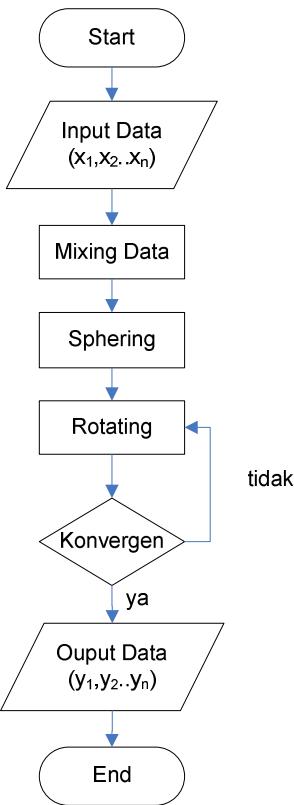
Untuk membandingkan maka digunakan nilai parameter yang sama untuk setiap proses iterasi pada keempat metode tersebut, yakni besar langkah (*learning rate*), blok proses serta algoritma yang digunakan, dimana pada penelitian ini digunakan algoritma natural gradient dengan jumlah iterasi 100, besar langkah 0,00001 dan blok proses 30.

Proses ICA pada penelitian ini diawali dengan menginputkan sinyal campuan dari tiap-tiap mikrofon. Sinyal yang sudah dalam bentuk matriks ini kemudian dicampur (mixing) satu sama lain untuk menghasilkan matriks campuran dengan orde jumlahan dari sinyal input. Matriks campuran akan *di-sphere* untuk mencari arah putar yang tepat. Setelah melalui proses *sphering* matriks akan diputar (*rotating*) untuk mencapai konvergensi melalui beberapa kali iterasi. Diagram proses ICA ini ditunjukkan oleh Gambar 3. sedangkan diagram alirnya ditunjukkan Gambar 3.

Pada metode pertama dilakukan pemisahan sinyal dalam domain waktu, sedangkan pada metode kedua sinyal campuran dipisahkan dalam domain frekuensi. Metode ketiga dan keempat merupakan gabungan antar dua metode sebelumnya yang dikombinasikan urutannya.



Gambar 3.4 Diagram Proses ICA



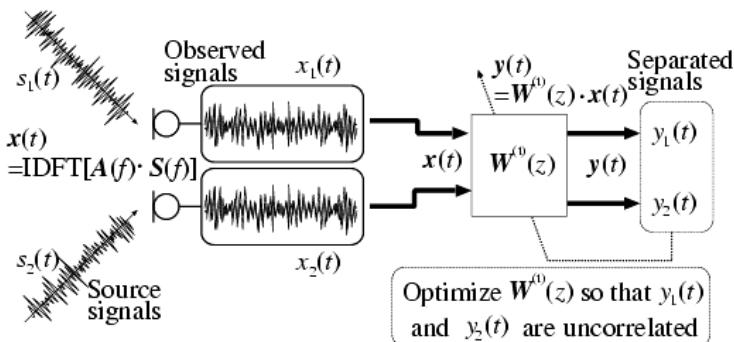
Gambar 3.5 Diagram Alir Proses ICA

3.2.1 Time Domain ICA (TDICA)

Proses pemisahan sinyal suara diawali dengan mengubah file suara kedalam bentuk matriks kemudian menyamakan ukurannya. Semakin kecil ukuran file, semakin cepat proses iterasi untuk mencari nilai W yang konvergen. Kemudian menggabungkan file tiap kanal, dan menginputkan file campuran pada algoritma ICA. Proses pencampuran sinyal suara ini dapat dimodelkan secara

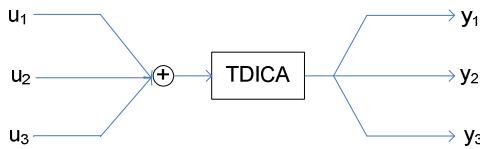
konolutif (*convolutive mixture*), dimana respon dari mikrofon dikalikan dengan respon ruang. Dengan mengasumsikan bahwa proses pencampuran terjadi secara linear maka secara matematis pencampuran sinyal suara ini dapat direpresentasikan dalam bentuk konvolusi, persamaan (2.48).

Pada TDICA, sinyal tidak perlu dirubah dalam domain frekuensi, jadi proses ini lebih cepat dan sederhana dibandingkan dengan FDICA. Iterasi filter pemisah, \mathbf{W} , pada proses ini menggunakan metode iterasi *Natural Gradient* dengan besar langkah nilai pembelajaran (*learning rate*) sebesar 0,0001. Proses pemisahan sinyal dengan TDICA dapat diilustrasikan seperti Gambar di bawah.



Gambar 3.6 ICA dalam domain waktu (TDICA)[18]

Output dari proses TDICA ini merupakan sinyal estimasi akhir yang diharapkan mampu mewakili sinyal sumber terukur. Sinyal ini di simpan dalam bentuk file **.wav** untuk diproses sebagai inputan program perhitungan frekuensi sesaat. Proses pemisahan sinyal dalam domain waktu dengan tiga input dan tiga output seperti pada penelitian ini diGambarkan pada Gambar 3.6.

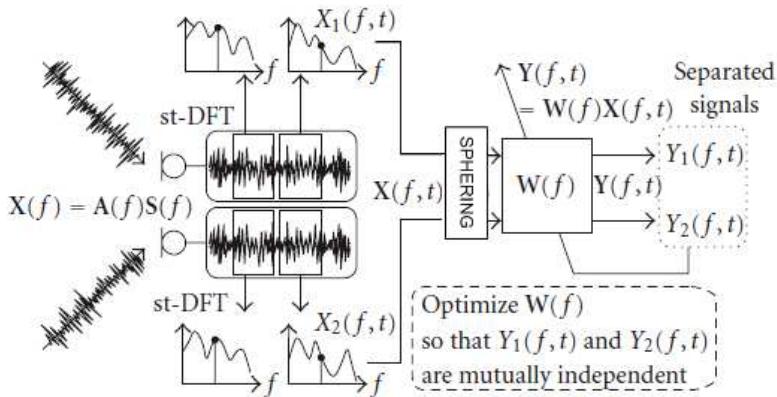


Gambar 3.7 Proses TDICA dengan 3 input – 3 output

3.2.2 Frequency Domain ICA (FDICA)

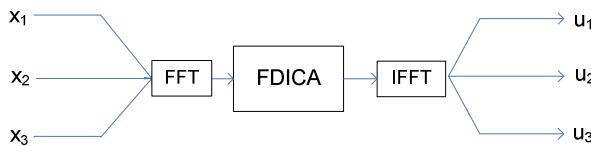
Seerti halnya pada TDICA, proses pemisahan sinyal suara FDICA diawali dengan mengubah file suara kedalam bentuk matriks kemudian menyamakan ukurannya. Pada proses FDICA, sinyal input dalam domain waktu diubah menjadi domain frekuensi dengan transformasi fourier (FFT). Sebelum dipisahkan maka data di spherling, pada proses ini juga dilakukan permutasi data input dengan vektor permutasi yang dibangkitkan agar stasioner.

Setelah dilakukan *pre-processing*, maka selanjutnya sinyal di-diputar (*rotating*), yakni dengan mencari filter pemisah, \mathbf{W} , agar pemisahan optimal. Algoritma yang digunakan untuk mencari filter pemisah \mathbf{W} (f) adalah natural gradien [1], dengan nilai langkah pembelajaran μ pada persamaan (2.36) sebesar 0,00001 dan jumlah iterasi 100 kali. Setelah nilai \mathbf{W} konvergen maka didapatkan sinyal estimasi dalam domain frekuensi (ω). Untuk mengubah sinyal sinyal estimasi menjadi domain waktu, dilakukan transformasi fourier balik (IFFT), maka didapatkan sinyal estimasi output dalam domain waktu (t). Proses pemisahan sinyal suara dalam domain frekuensi ini dapat di ilustrasikan melalui Gambar di bawah ini.



Gambar 3.8 ICA dalam domain Frekuensi (FDICA)[19]

Untuk tiga sumber dan tiga sensor, proses pemisahan sinyal dengan metode FDICA dapat diilustrasikan dalam blok diagram berikut.

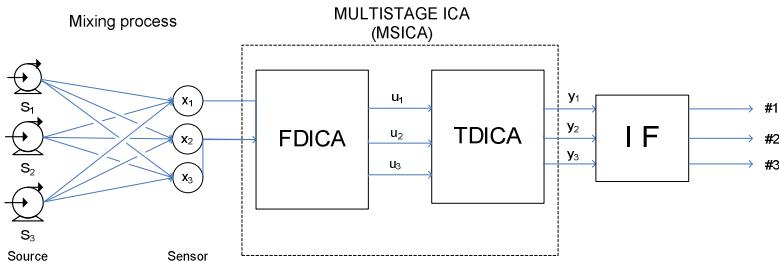


Gambar. 3.9 Proses FDICA dengan 3 input – 3 output

Pada metode MSICA 1, TDICA dilakukan terlebih dahulu daripada FDICA. Output dari TDICA akan menjadi input untuk FDICA, kemudian ditransformasi fourier dan dipisahkan dalam domain frekuensi dengan metode iterasi dan parameter yang sama. Output dari FDICA ini akan ditransformasi balik (IFFT) kedalam domain waktu, sinyal ini merupakan sinyal estimasi MSICA 1.

MSICA 2 merupakan kebalikan dari MSICA 1, dimana FDICA dilakukan lebih dahulu daripada TDICA. Setelah dipisahkan dalam domain frekuensi, output dari TDICA akan langsung dipisahkan kembali dalam domain waktu. Sinyal output

dari TDICA merupakan sinyal estimasi MSICA 2. Diagram blok MSICA 2 ditunjukkan oleh Gambar 3.9 di bawah ini.



Gambar 3.10 Skema Sistem Keseluruhan (MSICA 2)

3.3 Perhitungan Frekuensi Sesaat

Sesuai dengan konsep yang telah dijelaskan pada bab sebelumnya, ada 2 cara dalam menentukan pola suara dengan metode Frekuensi Sesaat dari suatu sinyal. Pada tugas akhir ini penulis menggunakan cara kedua, yaitu melalui *Short Time Fourier Transform* (STFT) dengan tahapan sebagai berikut :

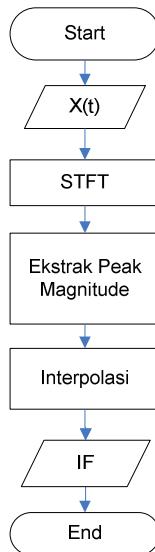
1. Perhitungan *Short Time Fourier Transform* (STFT)
STFT ini dihitung dengan cara membagi sinyal x(t) menjadi beberapa segmen dengan menggunakan *window* yang digeser sepanjang sampel n, dengan lebar pergeseran sebesar setengah dari lebar *window*. Perhitungan STFT ini menghasilkan *spectrogram* dengan pita lebar atau sempit sesuai dengan lebar *window*.
2. Perhitungan Frekuensi Sesaat (persamaan 2.67 – 2.71)
Frekuensi sesaat ini diperoleh dengan menurunkan fasa dari sinyal terhadap waktu. Untuk mengetahui fasa dari suatu sinyal, maka perlu diketahui bentuk sinyal analisanya. Turunan dari fasa ini dapat ditentukan dengan persamaan 2.67 dan selanjutnya dapat dihitung frekuensi sesaatnya dengan persamaan 2.70 dan 2.71.
3. Penelusuran Frekuensi Sesaat

Frekuensi sesaat yang ditelusuri adalah frekuensi sesaat yang memiliki amplitudo terbesar. Penelusuran ini dilakukan disetiap *bin* dari *window*, dengan menggunakan *tresehold* dari amplitudo tertentu. Sehingga diperoleh frekuensi sesaat yang mewakili frekuensi dominan (frekuensi fundamental) dari sinyal. Hasilnya berupa *spectrogram* dengan garis-garis tunggal yang menunjukkan frekuensi dengan amplitudo yang lebih besar dari *threshold* yang telah ditentukan.

4. Interpolasi Hasil Penelusuran

Interpolasi dilakukan agar diperoleh hasil frekuensi sesaat yang lebih baik. Sehingga frekuensi fundamentalnya akan terlihat lebih jelas.

Tahapan-tahapan perhitungan frekuensi sesaat di atas dapat digambarkan dengan diagram alir berikut



Gambar 3.11 Diagram Alir Perhitungan Frekuensi Sesaat

BAB IV

ANALISA DATA DAN PEMBAHASAN

Pada bab berikut akan dipaparkan hasil penelitian yang telah dilakukan dengan menyajikan data yang didapat kemudian menganalisisnya. Pembahasan dilakukan terhadap metode - metode yang digunakan untuk memisahkan suara mesin dengan analisa komponen independen (ICA) serta informasi frekuensi sesaat dari sinyal estimasi untuk deteksi kerusakan.

4.1 Analisa Data

Data yang didapatkan pada penelitian ini dibagi berdasarkan jumlah sensor dan jumlah sumber yang digunakan saat pengambilan data. Sensor mikrofon diletakkan secara spesifik terhadap kondisi salah satu motor, sehingga sinyal yang diterima mikrofon akan lebih kuat pada motor yang berada di dekatnya namun tercampur dengan sinyal emisi dari motor lainnya. Pada penelitian ini direkam suara motor sebagai sumber dengan mikrofon sebagai sensor dengan rincian sebagai berikut:

1. Satu mikrofon – satu motor.
2. Dua mikrofon – dua motor.
3. Tiga mikrofon – tiga motor.
4. Empat mikrofon – tiga motor.
5. Empat mikrofon – empat motor.

Perekaman *single mic* dilakukan sebagai *baseline* penelitian, dimana sinyal estimasi akan dibandingkan dengan sinyal ini untuk mendeteksi kerusakan motor. Tiap kondisi motor yakni: normal, *unbalance*, *misalignment* dan *bearing fault* direkam sendiri-sendiri secara terpisah dengan satu mikrofon dengan keadaan lingkungan yang sama. Pada perekaman dengan dua mikrofon, tiga mikrofon dan empat mikrofon dikombinasikan kondisi kerusakan antar motor seperti ditunjukkan pada tabel di bawah.

Tabel 4.1 Pengaturan Kondisi Motor
2 Mikrofon – 2 Motor

Percobaan ke-	Mic 1 – Motor 1	Mic 2 – Motor 2
1.	Bearing Fault	Misalignment
2.	Bearing Fault	Normal
3.	Misalignment	Normal
4.	Misalignment	Unbalance
5.	Normal	Unbalance
6.	Bearing	Unbalance

Tabel 4.2 Pengaturan Kondisi Motor
3 Mikrofon – 3 Motor

Percobaan ke-	Mic 1 – Motor 1	Mic 2 – Motor 2	Mic 3 – Motor 3
1.	Bearing Fault	Normal	Misalignment
2.	Bearing Fault	Unbalance	Misalignment
3.	Bearing Fault	Unbalance	Normal
4.	Misalignment	Unbalance	Normal

Tabel 4.3 Pengaturan Kondisi Motor
4 Mikrofon – 4 Motor

Mic – Motor	Kondisi Motor
Mic 1 – Motor 1	Normal
Mic 2 – Motor 2	Unbalance
Mic 3 – Motor 3	Misalignmnet
Mic 4 – Motor 4	Bearing Fault

Sinyal yang diterima masing-masing mikrofon pada tiap percobaan tersebut kemudian akan dipisahkan dengan metode ICA. Karena sinyal pada tiap mikrofon lebih kuat pada salah satu kondisi motor dikarenakan jarak dan sudut datangnya, maka sinyal ini dapat digunakan untuk mengestimasi sinyal sumber dari motor dengan menggabungkan informasi (sinyal) yang diterima oleh mikrofon lainnya. Sinyal estimasi yang diperoleh dicari pola frekuensi sesaatnya untuk menentukan kondisi kerusakan motor. Realisasi dari sistem yang dirancang untuk penelitian ini seperti terlihat pada Gambar 4.1.



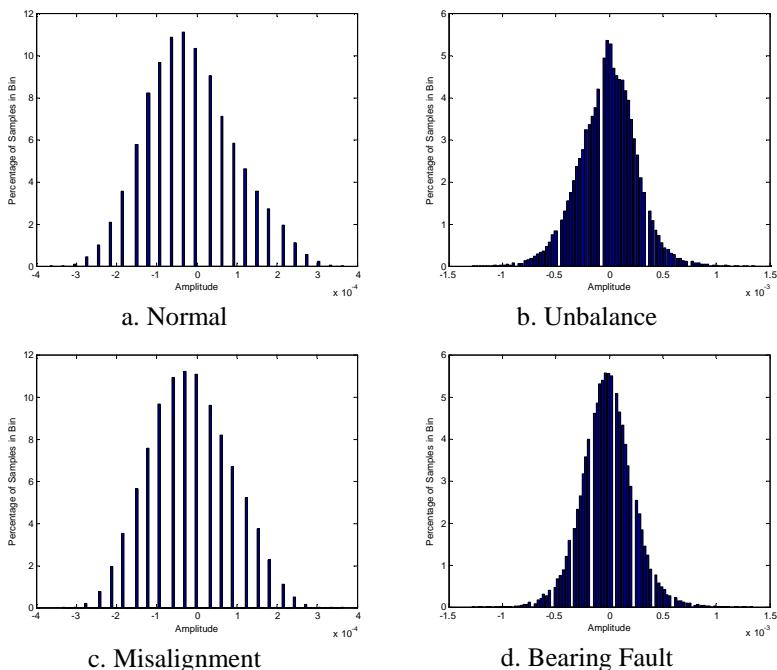
Gb 4.1 Realisasi Sistem

4.1.1 Pemisahan Sinyal Suara

Sinyal suara mesin yang didapatkan dari tiap mikrofon diolah berdasarkan proses perekamannya. Misalnya pada perekaman dengan dua mikrofon dan dua motor, pada percobaan pertama, mikrofon 1 akan menerima sinyal suara lebih kuat dari motor dengan kondisi *bearing fault* sedangkan mikrofon 2 lebih kuat menerima sinyal dari motor kondisi dengan kondisi *misalignement*, dikarenakan jarak dan sudut datang motor tersebut terhadap mikrofon. Sinyal dari kedua mikrofon ini akan dipisahkan untuk mencari sinyal estimasi, yakni sinyal sumber dari motor, berdasarkan asumsi bahwa motor dengan kondisi *bearing fault* independen terhadap motor dengan kondisi *misalignment*. Mikrofon menerima sinyal campuran, meski tiap mikrofon menerima sinyal lebih kuat dari kondisi tertentu. Untuk memisahkan sinyal campuran ini, digunakan metode ICA dengan pendekatan dari domain waktu (*Time Domain ICA-TDICA*), pendekatan domain frekuensi (*FDICA*), serta gabungan keduanya secara bertahap (*Multistage ICA-MSICA*). Pendekatan multi-stage ICA dilakukan baik dengan mendahuluikan *TDICA* maupun *FDICA*.

Syarat berlakunya metode ICA adalah bahwa asumsi sinyal sumber merupakan sinyal non-gaussian, atau minimal satu sinyal

gausian. Untuk membuktikan asumsi ini dapat dilihat karakteristik statistik sinyal sumber berdasarkan fungsi peluang kerapatannya (*pdf*). Pendekatan *pdf* dapat dilakukan dari estimasi histogram sinyal. Histogram dari keempat sinyal asli dari masing-masing kondisi motor yang menyatakan *pdf*-nya dapat dilihat pada gambar di bawah.

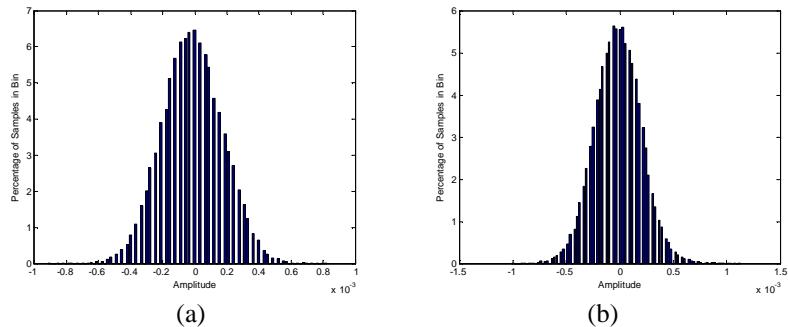


Gambar 4.2 Histogram Sinyal Asli

Dari keempat histogram sinyal asli dari masing-masing kondisi motor di atas terlihat bahwa hampir semuanya memiliki distribusi non-gaussian, hanya sinyal dari bearing fault yang menyerupai distribusi Gaussian. Dengan demikian asumsi bahwa sinyal non-gaussian dapat dipakai untuk analisa komponen independen.

Untuk mengetahui bagaimana distribusi sinyal campuran maka beberapa komponen sinyal tersebut dapat digabungkan untuk

menghasilkan sinyal campuran (*mixing signal*). Misalnya gabungan antara sinyal normal dan *unbalance* atau *misalignment* dan *bearing fault*. Pada Gambar 4.3 histogram dari kedua gabungan sinyal tersebut ditampilkan, dimana keduanya merupakan sinyal yang diindera oleh mikrofon (2 mic – 2 sumber).

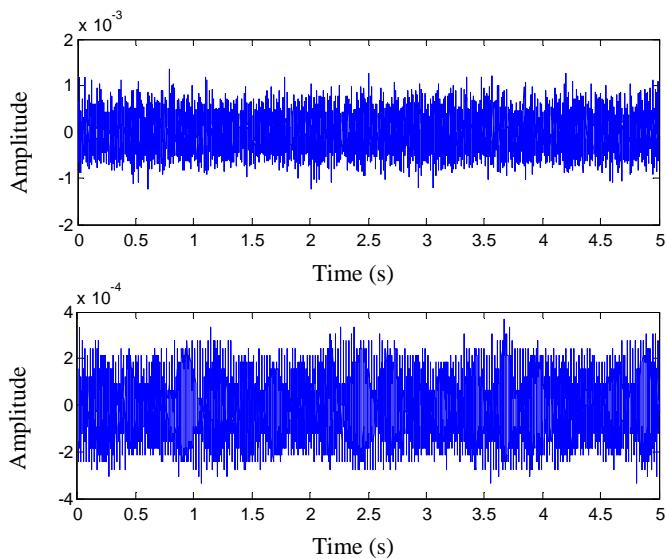


4.3 Histogram Gabungan 2 Sinyal
 a. Sinyal Normal dan Unbalance
 b. Sinyal Misalignment dan Bearing Fault

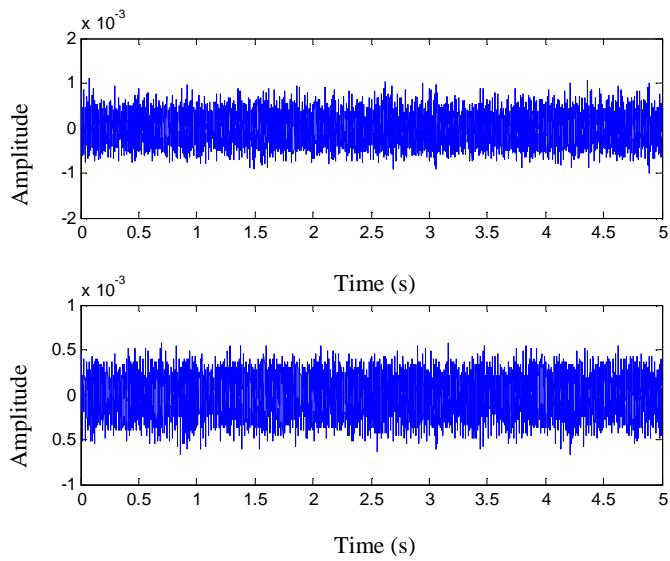
Dengan terbuktiunya asumsi sinyal sumber merupakan sinyal non-gaussian maka metode ICA dapat diterapkan untuk memisahkan sinyal suara mesin. Hasil pemisahan sinyal suara mesin dengan beberapa metode ICA dapat dijabarkan pada sub-bab berikut.

- **Time Domain ICA (TDICA)**

Analisa dasar suatu sinyal dapat dilakukan dalam domain waktu (*time-series*). Pada domain ini ditampilkan informasi besarnya amplitudo sinyal tiap waktu sampel. *Time domain ICA* (TDICA) memisahkan sinyal dalam domain waktu dengan model pencampuran konvolusi seperti pada persamaan (2.52). Pada percobaan pertama dengan dua mikrofon-dua motor dikondisikan motor dalam keadaan *bearing fault* dan *misalignmnet*. Berdasarkan perekaman *baseline* penelitian dengan *single mic* diperoleh sinyal asli dan campurran dari kondisi *bearing fault* dan *unbalance* dalam domain waktu sebagai berikut:

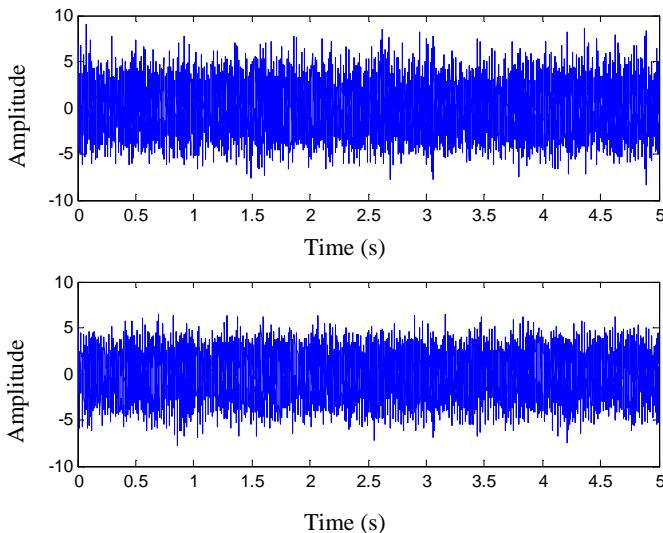


Gambar 4.4 Sinyal Asli Kondisi Bearing Fault dan Misalignment



Gambar 4.5 Sinyal Campuran tiap mikrofon

Hasil pemisahan sinyal dengan TDICA adalah sebagai berikut,



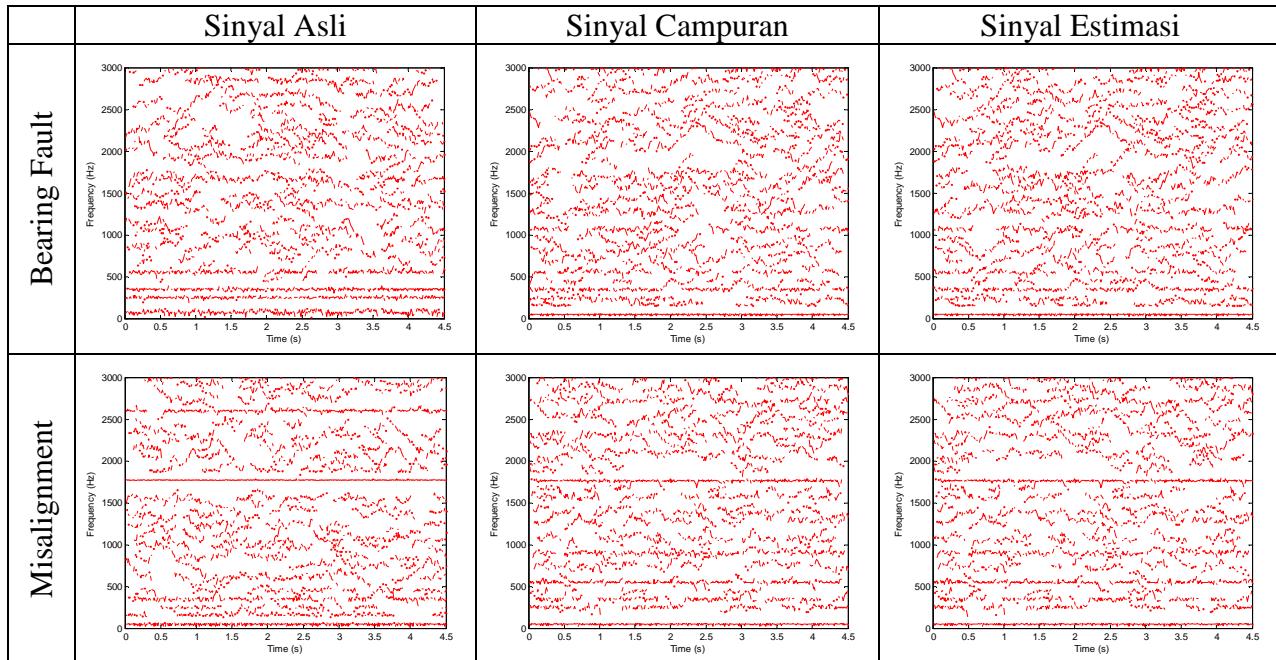
Gambar 4.6 Sinyal Estimasi TDICA

Untuk mengidentifikasi apakah sinyal tersebut adalah sinyal dari *bearing fault* atau *misalignment* sulit dianalisa dari domain waktu, seperti pada Gambar 4.6. Pola frekuensi sesaat digunakan untuk identifikasi sinyal dan menghasilkan diagram seperti pada Tabel 4.5. Dari pola frekuensi sesaat terlihat bahwa sinyal bagian atas adalah sinyal dari *bearing fault* dan sinyal bagian bawah merupakan sinyal kondisi *misalignment*. Kuantitas hasil pemisahan sinyal suara dalam domain waktu (TDICA) ini dapat ditunjukkan pada tabel di bawah.

Tabel 4.4 Kondisi dan Kualitas TDICA

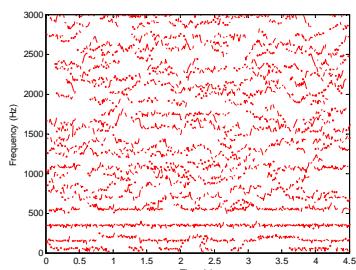
Parameter	Nilai
Besar Langkah	0.000001
Blok Proses	30
Jumlah Iterasi	100
Angle	0.2
SNR	-4.6321

Tabel 4.5 Perbandingan Sinyal Asli, Campuran dan Estimasi pada TDICA

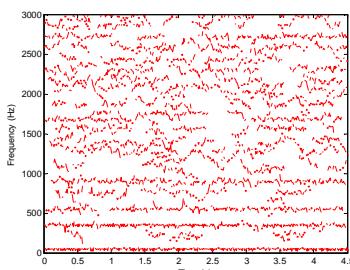


- **Frequency-Domain ICA (FDICA)**

Analisa dalam domain waktu terkadang kurang bisa memberikan informasi yang terkandung dalam suatu sinyal. Begitu juga dalam pemisahan sinyal dengan metode ICA. Maka, pemisahan dalam domain frekuensi menjadi alternatif, dan secara matematik lebih mudah karena hanya melibatkan perkalian *dot product* (persamaan 2.53). Untuk merubah sinyal dalam domain waktu menjadi domain frekuensi, sinyal ditransformasikan dengan *fourier transform* dan secara komputasi dilakukan dengan FFT (*fast fourier transform*). Hasil dari FDICA berupa sinyal estimasi dalam domain frekuensi, sehingga perlu ditransformasi balik dalam domain waktu dengan IFFT (*Inverse FFT*). Sinyal estimasi ini dicari pola frekuensi sesaatnya untuk identifikasi sinyal.



a. Estimasi 1



b. Estimasi 2

Gambar 4.7 Sinyal Estimasi FDICA (IF)

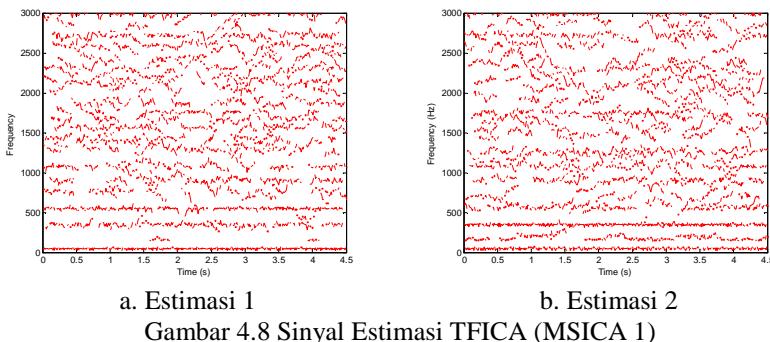
Pada pola frekuensi sesaat di atas terlihat bahwa FDICA hanya mampu memperoleh satu jenis sinyal estimasi, karena kedua sinyal estimasi tersebut mirip pola frekuensi sesaatnya dan keduanya dapat diidentifikasi sebagai sinyal dari kondisi *bearing fault*.

Tabel 4.6 Kondisi dan Kualitas FDICA

Parameter	Nilai
Besar Langkah	0.000001
Blok Proses	30
Jumlah Iterasi	100
Angle	1,4
SNR (dB)	11.9272

- **Time-Frequency ICA (MSICA 1)**

Pemisahan sinyal suara mesin dengan metode ICA pada kedua domain waktu dan frekuensi telah berhasil dilakukan dan diketahui hasil pemisahannya. Pengembangan pendekatan tersebut adalah dengan menggabungkan pemisahan dalam domain waktu dan frekuensi sekaligus secara bertahap (*Multistage ICA – MSICA*). Pada tahap pertama pengembangan ini (MSICA 1), TDICA dilakukan terlebih dahulu daripada FDICA. Ouput dari TDICA menjadi input bagi TDICA dan sinyal estimasi dari TDICA akan dicari pola frekuensi sesaatnya. Dengan menggunakan nilai parameter yang sama dengan sebelumnya diperoleh sinyal estimasi dengan pola dan kualitas seperti ditunjukkan oleh gambar dan tabel di bawah ini.



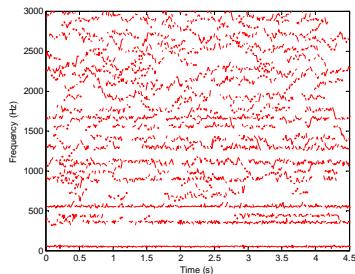
Untuk kondisi besar langkah, blok proses dan jumlah iterasi yang sama, MSICA 1 menghasilkan konvergensi yang lebih baik dari metode-metode sebelumnya.

Tabel 4.7 Kondisi dan Kualitas MSICA 1

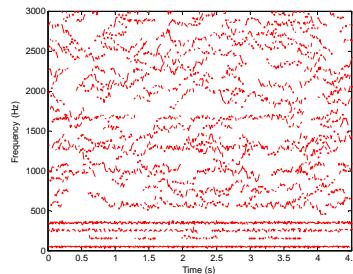
Parameter	Nilai
Besar Langkah	0.000001
Blok Proses	30
Jumlah Iterasi	100
Angle TDICA	0.0
Angle FDICA	0.0
SNR	12.6752

- **Frequency-Time ICA (MSICA 2)**

Pada pendekatan multi tahap selanjutnya, FDICA dilakukan terlebih dahulu terhadap TDICA. Sinyal input dalam domain waktu ditransformasi fourier dalam domain frekuensi. Sinyal estimasi output FDICA juga dalam domain frekuensi dan ditransformasi balik (IFFT) ke dalam domain waktu untuk di-inputkan pada TDICA. Output dari TDICA berupa sinyal estimasi akhir dalam domain waktu. Data pemisahan sinyal dengan MSICA 2 adalah sebagai berikut:



b. Sinyal Estimasi 2



Gambar 4.9 Sinyal Estimasi FTICA (MSICA 2)

Hasil pemisahan sinyal suara mesin dengan MSICA 2 untuk nilai parameter yg sama dengan metode sebelumnya dapat ditunjukkan pada tabel di bawah ini.

Tabel 4.8 Kondisi dan Kualitas MSICA 2

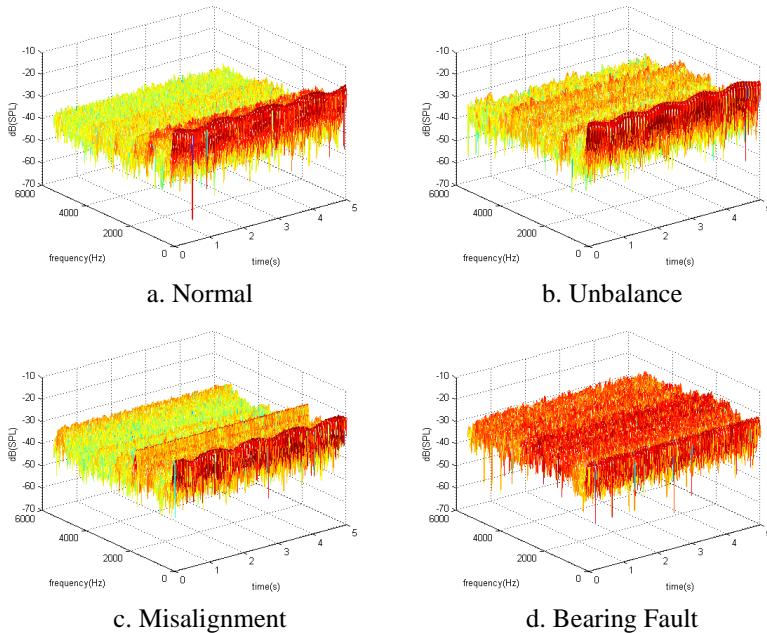
Parameter	Nilai
Besar Langkah	0.000001
Blok Proses	30
Jumlah Iterasi	100
Angle FDICA	1,4
Angle TDICA	0,3
SNR	22.9999

4.1.2 Perhitungan Frekuensi Sesaat

Pada sub-bab ini data yang ditampilkan dapat digolongkan menjadi dua jenis diagram atau grafik, yakni spektrogram sinyal dan frekuensi sesaatnya. Spektrogram menggambarkan sinyal dalam domain waktu-frekuensi. Diagram ini memperlihatkan perbedaan warna yang menunjukkan besarnya amplitudo (dB SPL) tiap frekuensi

waktu. Pola warna biru mewakili amplitudo yang rendah sedang pola garis merah mewakili amplitudo tinggi.

Pola frekuensi sesaat menggambarkan perubahan frekuensi tiap sampel waktu. Grafik atau diagram ini berupa titik-titik serta garis, dimana titik-titik yang terhubung terus sepanjang waktu membentuk garis lurus menunjukkan frekuensi dominan dari kondisi tersebut. Pola frekuensi sesaat ini diperoleh dengan menurunkan fasa sinyal terhadap waktu, sesuai perhitungan pada persamaan (2.63) dan (2.67).

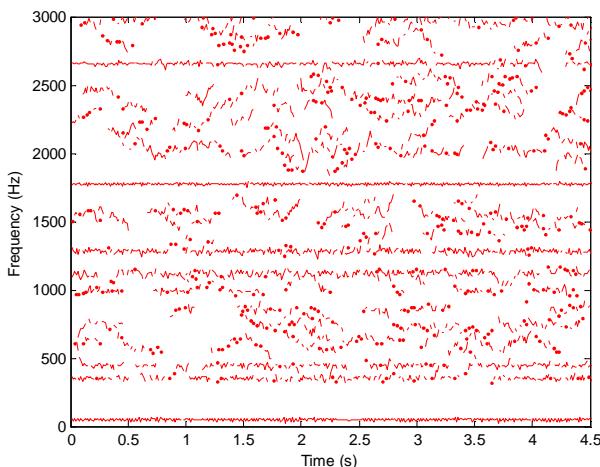


Gambar 4.10 Spektrogram 3-D Sinyal Asli

Identifikasi sinyal estimasi terhadap sinyal asli yang telah dilakukan sebelumnya mengacu pada pola frekuensi sesaat kedua sinyal tersebut. Untuk menentukan bahwa pola frekuensi sesaat sinyal tersebut adalah *bearing fault* atau *unbalance* maka perlu ditelusuri pola frekuensi sesat sinyal asli (*baseline* penelitian).

Penelusuran frekuensi sesaat dari empat kondisi motor tersebut adalah sebagai berikut,

- Kondisi Normal

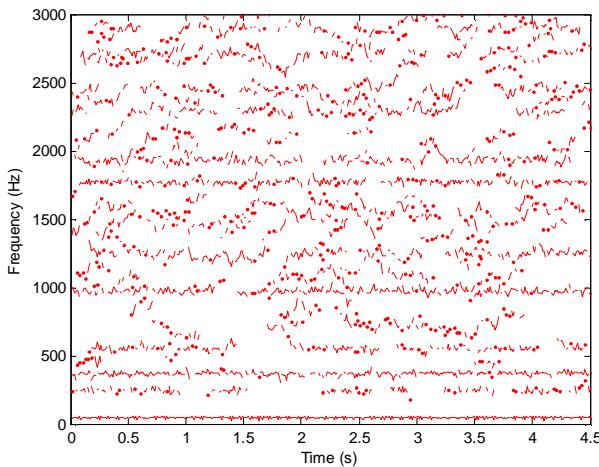


Gambar 4.11 Pola Frekuensi Sesaat Sinyal kondisi Normal

Pada spektrogram kondisi normal terlihat pola garis gelap pada frekuensi rendah. Pola ini terlihat kembali di sekitar frekuensi 1000 Hz ,1700 Hz serta 2600 Hz. Untuk mencari nilai numerik frekuensi tersebut maka spektrogram diubah menjadi diagram pola frekuensi sesaat yang menunjukkan lebih detil nilai frekuensi fundamentalnya. Dari penelusuran frekuensi sesaat akan didapat satu atau beberapa frekuensi fundamental, frekuensi tersebut dikarakterisasi sebagai frekuensi mesin pada kondisi normal.

- *Kondisi Unbalance*

Pada kondisi *unbalance* terlihat spektrogram mulai berubah warna menjadi lebih merah. Hal ini mengindikasikan peningkatan amplitudo dari kondisi normal. Sedangkan penelusuran frekuensi sesaat menghasilkan perubahan yang tidak mencolok dibandingkan kondisi normal.

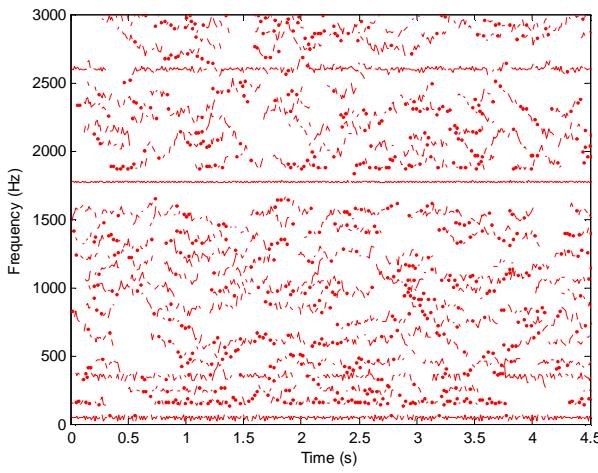


Gambar 4.12 Pola Frekuensi Sesaat Sinyal Kondisi Unbalance

Pada kondisi *unbalance* ini didapatkan frekuensi fundamental yang rendah yakni di sekitar 50 Hz, 360 Hz, 1700 Hz dan 1900 Hz. Frekuensi 2600 Hz yang sebelumnya tampak pada kondisi normal mulai tidak tampak pada kondisi *unbalance* ini. Kondisi *unbalance* pada penelitian ini dilakukan dengan penambahan masa 25 gr pada impeler motor/pompa.

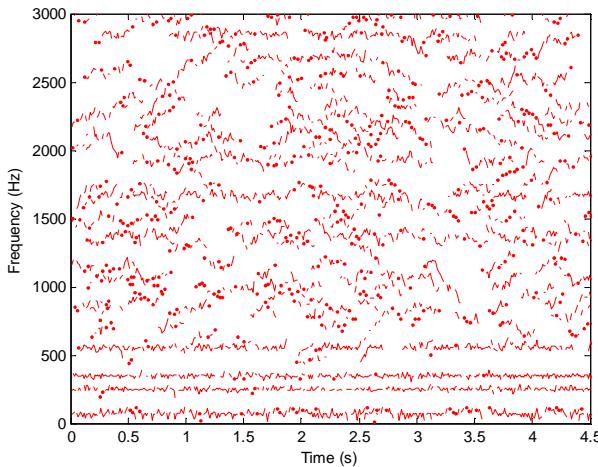
- Kondisi *Misalignment*

Pada kondisi misalignment, motor diletakkan dengan posisi miring 30° dari kondisi normal. Spektrogram yang didapatkan masih mirip dengan kondisi normal namun terjadi peningkatan amplitudo dan juga frekuensi fundamental baru di sekitar 5200 Hz dan 5300 Hz.



Gambar 4.13 Frekuensi Sesaat Sinyal Kondisi Misalignment

- Kondisi *Bearing Fault*



Gambar 4.14 Pola Frekuensi Sesaat Sinyal Kondisi Bearing Fault

Kondisi *Bearing Fault* atau kerusakan bearing diberikan dengan merusak rumah penahan *ball bearing* atau disebut *ball*

cage. Kerusakan ini mengakibatkan pergerakan bearing yang dapat merusak mesin/motor. Hasil citra spektrogram menunjukkan warna yang paling gelap daripada ketiga spektrogram pada kondisi sebelumnya. Hal tersebut menunjukkan bahwa pada kondisi ini suara mesin cenderung bertambah keras. Frekuensi fundamental pada kondisi ini cenderung dominan pada frekuensi-frekuensi rendah dengan amplitudo tinggi pada 70 Hz, 260 Hz, 350 Hz dan 586 Hz.

4.2 Pembahasan

Pemisahan sinyal suara dengan metode ICA dapat dilakukan dalam domain waktu dan atau frekuensi, serta gabungan keduanya. Pada penelitian ini pemisahan sinyal suara mesin dilakukan pada kedua domain tersebut, serta menggabungkan antara *Frequency Domain ICA* (FDICA) dengan *Time Domain ICA* (TDICA), selanjutnya disebut *Multistage ICA* (MSICA). Permasalahan utama pada metode ICA adalah mencari filter pemisah, \mathbf{W} , yang ideal. Pada penelitian ini digunakan metode *Natural Gradient* untuk mencari nilai \mathbf{W} .

Pembelajaran nilai \mathbf{W} menurut [21] dipengaruhi oleh dua parameter utama, yakni besar langkah (μ) dan nilai blok proses (B). Pada penelitian ini, untuk membandingkan beberapa pendekatan dalam domain ICA maka ditetapkan nilai parameter yang sama yakni besar langkah yang digunakan pada sebesar $\mu = 0.000001$ dan blok proses $B=30$. Nilai ini didapatkan berdasarkan uji-coba untuk memperoleh konvergensi nilai \mathbf{W} . Berdasarkan hasil pemisahan sinyal dengan keempat pendekatan yang dilakukan diperoleh bahwa secara kualitas pemisahan sinyal dalam domain waktu (TDICA) lebih baik dari ketiga metode lainnya, sedang secara kuantitas dari konvergensi iterasi filter pemisah, diperoleh data bahwa *Time-Frequency ICA* (MSICA 1), lebih baik dari ketiga metode lainnya.

Hasil pemisahan sinyal menunjukkan adanya pengukuran amplitudo sinyal yang cukup signifikan, contohnya pada percobaan pertama. Sinyal input memiliki *range* amplitudo (sumbu-y) $-2 \cdot 10^{-3}$ sampai dengan $2 \cdot 10^3$, namun setelah terpisah amplitudo sinyal

estimasi sebesar -10 sampai 10 dB. Begitu juga pada percobaan-percobaan selanjutnya, kenaikan amplitudo terlihat cukup besar. Atas dasar ini, analisa amplitudo (dB SPL) untuk deteksi kerusakan kurang begitu akurat karena ada perubahan amplitudo yang cukup besar antara sinyal asli dan sinyal estimasi. Uji perfromansi dengan SNR (*signal to noise ratio*) memberikan hasil yang tidak sesuai dengan hasil penelitian, dimana metode TDICA yang menghasilkan kualitas pemisahan sinyal terbaik memiliki nilai SNR terrendah. Kedepan, uji performansi sinyal estimasi suara mesin dengan metode SNR perlu dikaji kembali.

Pola suara didapat dengan metode frekuensi sesaat dari *spectrogram* yang menunjukkan besarnya nilai amplitudo untuk setiap frekuensi sinyal pada tiap waktu pengukuran. Frekuensi fundamental dari setiap kondisi mesin diketahui dengan melihat pola garis lurus yang terjadi pada frekuensi tertentu sepanjang waktu pengukuran. *Spectrogram* disebut juga sebagai *time-frequency representation* dari suatu sinyal, yang diperoleh dengan menggunakan *window* yang lebar. *Window* yang digunakan adalah Hanning *window* dengan lebar 256 dan memiliki lebar *main lobe* pada -3 dB sebesar 1,44 *bins*, dan pada *main lobe* -6 dB sebesar 2,00 *bins*. Dan untuk maksimum *slide lobe*-nya sebesar -32 dB dan *side lobe roll-off rate* sebesar 60 dB. Sedangkan *shifting* (pergeseran) *window* yang digunakan sebesar setengah dari lebar *window* dengan 256 titik *window*. Semakin besar nilai *window*, maka *spectrogram* yang diperoleh semakin baik [15].

Berdasarkan *baseline* penelitian dengan perekaman *single mic*, dapat diketahui frekuensi kerusakan mesin dari pola frekuensi sesaatnya. Pada kondisi normal, motor tidak dikenai perlakuan kerusakan apapun. Namun pada kondisi ini motor sudah memiliki frekuensi fundamental, yakni pada 51 Hz, antara 1000-1300 Hz, 1770 Hz dan 2650 Hz. Frekuensi 51 Hz dan 1770 Hz terus muncul pada kondisi kerusakan lain, frekuensi ini bisa diidentifikasi sebagai frekuensi pribadi motor. Untuk menyederhanakan analisa frekuensi, maka frekuensi motor dapat dibatasi sampai pada 3000 Hz saja, karena RPM motor yang digunakan hanya 2850 rpm [2].

Ketidakseimbangan (*unbalance*) merupakan cacat yang sering terjadi yang disebabkan oleh getaran. *Unbalance* adalah kondisi dimana pusat massa tidak sesumbu dengan sumbu rotasi sehingga rotor mengalami gaya getar terhadap bearing yang menghasilkan gaya sentrifugal. Ada beberapa faktor yang menyebabkan terjadinya *unbalance* yakni: kesalahan saat proses pemesinan dan assembly, eksentrisitas komponen, adanya kotoran saat pengecoran, korosi dan keausan, distorsi geometri karena beban termal dan beban mekanik serta penumpukan material [7]. Pada penelitian ini kondisi *unbalance* dilakukan dengan menambahkan massa sebesar 25 gram pada impeler motor. Penambahan massa ini mengakibatkan pergeseran frekuensi fundamental motor dari kondisi normal. Pada kondisi *unbalance* ini, motor memiliki frekuensi suara fundamental sebesar 46 Hz, 1000 Hz, 1770 Hz dan 1990 Hz.

Kondisi *misalignment* merupakan kondisi poros yang tidak sesumbu, Pada penelitian ini kondisi *misalignment* dilakukan dengan meletakkan motor normal dengan kondisi miring 30° (*misalignment base*). Pada kondisi ini motor memiliki frekuensi fundamental pada 46 Hz, 1772 Hz, 2600 Hz dan 3400 Hz.

Kondisi kerusakan motor terakhir pada penelitian ini adalah *bearing fault* atau kerusakan bearing. Kondisi ini dilakukan dengan merusak rumah penahan *ball bearing* (*ball cage*) sehingga gerakan bearing lebih leluasa daripada saat ditahan oleh *ball cage*. Kerusakan bearing ini ditandai pada frekuensi rendah saja, namun tidak tunggal yakni terdapat beberapa frekuensi fundamental suara motor pada 73 Hz, 250 Hz dan 350 Hz.

Secara umum, pemisahan sinyal suara mesin pada penelitian ini telah dapat dilakukan. Beberapa sinyal estimasi dapat dikenali sebagai sinyal aslinya, sedangkan beberapa yang lain kurang mirip dengan sinyal aslinya. Pendekatan TDICA dengan filter pemisah natural gradient memberikan hasil yang lebih baik dari ketiga metode pendekatan lainnya. Pada penelitian selanjutnya, dapat digunakan algoritma filter pemisah lain untuk menguji konsistensi hasil penelitian ini. Studi mengenai karakteristik suara mesin juga diperlukan untuk memodelkan filter pemisah yang sesuai sehingga

suara mesin bisa benar-benar terpisah untuk dapat dianalisa. Dari sisi teori dan algoritma, pemisahan sinyal sendiri saat ini masih berkembang. Metode dan algoritma baru muncul tiap saat untuk menghasilkan kualitas pemisahan sinyal yang lebih optimal. Hasil penelitian ini bersifat fleksibel mengikuti perkembangan algoritma ICA. Adanya penguatan amplitudo sinyal estimasi terhadap sinyal asli juga menjadi tugas yang bisa diselesaikan pada penelitian selanjutnya, karena dengan adanya penguatan ini pencarian frekuensi dominan akan menjadi bermasalah ketika penguatan sinyal yang satu lebih besar dari sinyal yang lain sehingga terjadi pergeseran perbandingan nilai amplitudo antar sinyal. Penguatan amplitudo ini tidak hanya terjadi metode yang digunakan pada penelitian ini namun juga terjadi pada penelitian lainnya yang sejenis [11].

Penggunaan frekuensi sesaat untuk karakterisasi kerusakan cukup efektif karena perubahan frekuensi-frekuensi tiap sample waktu dapat diketahui secara eksak. Untuk mencari amplitudo yang dominan diantara frekuensi tersebut dapat dilihat pada spektrogram 3D pada bidang Y-Z yang menunjukkan besarnya amplitudo tiap frekuensi. Kelemahan spektrogram ini adalah resolusi frekuensi dominan yang didapat dari pola frekuensi sesaat tidak sama dengan frekuensi dominan pada spektrogram. Hal ini dikarenakan terjadi semacam trade-off antara frekuensi dan waktu, dimana semakin besar ukuran window semakin tinggi resolusi frekuensi, namun resolusi waktunya semakin berkurang (mirip dengan ketidakpastian Heseinberg). Akibat berkurangnya resolusi waktu ini terlihat jelas dengan adanya pengurangan waktu pada sumbu-x spektrogram, yakni range waktu menjadi 4,5 detik dari sinyal estimasi 5 detik. Pada penelitian selanjutnya perlu dicari cara untuk memperoleh frekuensi yang lebih akurat terutama melalui penelusuran pola frekuensi sesaat sehingga diperoleh frekuensi kerusakan yang benar untuk mengindikasikan kerusakan tersebut.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. Telah berhasil dipisahkan suara dari beberapa mesin dengan beberapa metode ICA untuk deteksi kerusakan mesin, dengan menggunakan algoritma filter pemisah *natural gradient* didapatkan hasil pemisahan terbaik menggunakan Time-Domain ICA (TDICA).
2. Pola suara kerusakan mesin dapat dikarakterisasi melalui frekuensi sesaatnya, pada penelitian ini kondisi mesin normal memiliki frekuensi pada 51 Hz, antara 1000-1300 Hz, 1770 Hz dan 2650 Hz, unbalance pada frekunsi 46 Hz, 1000 Hz, 1770 Hz dan 1990 Hz, misalignment pada frekuensi 46 Hz, 1772 Hz, 2600 Hz dan 3542 Hz serta bearing fault pada frekuensi 73 Hz, 250 Hz dan 350 Hz dengan amplitudo tinggi.

5.2 Saran

Dari kesimpulan penelitian maka saran yang dapat diberikan berkaitan dengan hasil penelitian ini adalah:

1. Dilakukan studi pada karakteristik akustik suara mesin sehingga dapat dimodelkan metode pemisahan sumber yang lebih optimal.
2. Digunakan metode validasi serta perhitungan performansi sinyal estimasi yang lebih akurat untuk penelitian selanjutnya.
3. Dikembangkan metode analisa suara untuk deteksi kerusakan mesin sampai menuju tahap aplikasi.

DAFTAR PUSTAKA

- [1] Amari, S. 1997. **Natural Gradient Works Efficiently in Learning.** Saitama: RIKEN, Japan.
- [2] Anda, A R. 2006. **Penggunaan Frekuensi Sesaat Untuk Deteksi Pola Suara Kerusakan Motor Listrik.** Tugas Akhir, ITS Surabaya.
- [3] Antoni J. 2005. Blind separation of vibration component: Principle and demostration. **Elsevier Mechanical System and Signal Processing** 19 1166-1180.
- [4] Arifianto, D._____. **EEG Noise Removal With ICA.** Lecture Slide: Lancaster University.
- [5] Arifianto, D. 2005. **Fundamental Frequency Estimation and Voiced/Unvoiced Determination Using Instantaneous Frequency,** Doctoral Dissertation, Department of Information Processing. Tokyo Institute of Technology.
- [6] Arifianto, D. 2000. **Frekuensi Sesaat dan Aplikasinya.** Seminar on Air PPI Titech No.1:8-10.
- [7] Atmaja, B.T. 2008. **Analisa Vibrasi pada Turbin Uap dengan FFT untuk Predictive Maintenance di PT PJB UP Gresik.** Laporan Kerja Praktek, Teknik Fisika ITS.
- [8] Bell, A J and Sejnowksi T J. 1996. **The ‘Independent Components’ of Natural Scene are Edge Filters.** CNL: The Salk Institute (submitted to vission research).
- [9] Bylund, Johan. 2001. **A Humble Attempt to Solve the Cocktail Party Problem using Blind Source Separation.** Sidney: University of Sidney
- [10] Douglas, SC. 2001. Microphone Array Tehcnique and Application: **BSS of Acoustic Signal.** (Bredstein-D.Ware, Eds.). Berlin: Springer Inc, 2001.

- [11] Gunawan, C S. 2005. **Pemisahan Sinyal Audio Tercampur Secara Off-line Dengan Metode Independent Component Analysis**, Tugas Akhir, UK Petra Surabaya.
- [12] Hyvärinen, A., Erkki, H. 2000. **Independent Component Analysis: Algorithm and Applications**. Helsinki Univ. Of Tech..
- [13] Ikeda, N., Murata N._____. **An Approach to Blind Source Separation of Speech Signal** Wako: RIKEN Brain Science Institute, Japan.
- [14] Ikeda, N., Murata N._____. **A method of ICA in Time and Frequency Domain**. Wako: RIKEN Brain Science Institute, Japan
- [15] Rahmadana, Y. 2007. **Penentuan Pola Suara Mesin Berputar pada Beberapa Kondisi Kerusakan dengan Metode Frekuensi Sesaat**. Tugas Akhir, ITS.
- [16] Parra, Lucas. 2002. **Tutorial on Blind Source Separation and Independent Component Analysis**. Adaptif image and signal prcessing group, Sarnoff Corporation (slide presentasi).
- [17] Seltzer, ML., Mitchel, L. 2003. **Microphone Array Processing for Robust Speech Recognition**. PhD Thesis,Carnegie Mellon University, 2003.
- [18] Saruwatari, H et.al. 2002. **Blind Source Separation of Acoustic Signal Based on Multistage Independent Component Analysis**. Power Point File.
- [19] Saruwatari, H et.al. 2003. **A Fixed-Point ICA Algorithm for Convoluted Speech Signal Separation**. ICA 2003, Nara, Japan.
- [20] Widyotriatmo, Augie. 2006. **Pemisahan Sinyal Suara Dari Rekaman Multi Sumber Dan Multi Kanal Menggunakan Metode Blind Source Separation**

Dengan Fungsi Sigmoid Tergeneralisasi Sebagai Model Sinyal Suara Studi Kasus Pemisahan Sinyal Suara Dari Rekaman Komponen-Komponen Drum (Bass Drum, Snare, Dan Hi-Hat). Tesis S2, ITB Bandung.

- [21] Widjyotriatmo, A., Prasetyo, I. 2005. **Pemisahan Sinyal Suara Ucap dari Derau dengan Metode Blind Source Separation.** Prosiding Seminar Akustik dan Getaran (VibrAStic) 2005. Surabaya, Teknik Fisika ITS.
- [22] Ypma, A. 2003. **Blind Source Separation of rotating machine source: bilinear forms and convolutive mixtures,** Elsevier Neurocomputing 29.
- [23] Ypma, A. 2001. **Learning Methods For Machine Vibration Analysis and Health Monitoring.** PhD Dissertation. TU Delft. The Netherland.
- [24] Zhou, W., Chelidze, D. 2007. Blind source separation based vibration mode identification. **Elsevier Mechanical System and Signal Processing** 21, 2007.
- [25] Matlab Refference.

Sumber Internet

- [26] <ftp://ftp.cnl.salk.edu/pub/tony/sep96/public>
- [27] <http://labrosa.ee.columbia.edu/matlab/sinemodel/>
- [28] <http://statutorial.blogspot.com/2008/01/skewness-dan-kurtosis.html>
- [29] <http://library.adisanggoro.or.id/Information%20Theory/T-Informasi-L1.htm>

- [30] http://www.ittelkom.ac.id/library/index.php?option=com_content&view=article&id=81:independent-component-analysis-ica&catid=15:pemrosesan-sinyal&Itemid=15
- [31] <http://agfi.staff.ugm.ac.id/blog/index.php/tag/fourier-transform/>
- [32] <http://blinkdagger.com/matlab/matlab-windowing-part-3>

PUBLIKASI

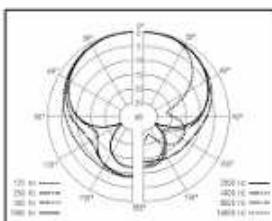
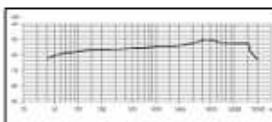
- Arifianto, D., Atmaja, B.T., Aisyah. A.S. 2009. **Blind Sound Separation Using Frequency-Domain And Time-Domain Independent Component Analysis For Machines Fault Detection.** Submitted to The International Conference on Instrumentation, Control and Automation (ICA) 2009, ITB, Bandung-Indonesia.
- Atmaja B.T., Arifianto, D. 2009. **Machinery Fault Diagnosis Using Independent Component Analyis and Instantaneous Frequency.** Submitted to International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME) 2009, ITB, Bandung-Indonesia.

LAMPIRAN

- A-1 Spesifikasi Mikrofon e845
- B-1 fp_tdica.m
Program pemisahan sinyal suara dalam domain waktu
- B-2 fp_fdica.m
Program pemisahan sinyal suara dalam domain frekuensi
- B-3 sep.m
Fungsi untuk pembelajaran data
- B-4 sepout.m
Fungsi untuk menampilkan teks output
- B-5. wchange.m
Fungsi untuk *tracking* ukuran dan arah perubahan pembobotan.
- B-6 fp_if.m
Program untuk menampilkan spektrogram dan frekuensi sesaat.
- B-7 ifgram.m
Fungsi untuk menghitung frekuensi sesaat
- B-8 extractrax.m
Fungsi untuk mengekstrak Peak amplitude (frekuensi sesaat)
- B-9 colinterpvals.m
Fungsi untuk menginterpolasi kolom (frekuensi sesaat)

A-1 Mikrofon Carol e845

evolution



e 845 High Performance Lead Vocal Microphone

The e845 is a high-performance super-cardioid lead vocal microphone with an extended high-frequency response specially designed for stage use. Sound Intel basket refined steel.

Features—Benefits

- Metal construction—rugged and reliable
- Full dynamic range and smooth frequency response—excellent signal quality
- Shock-mounted capsule—low sensitivity to impact and handling noise
- Hum compensating coil—reduces electrical interference
- Silent on/off switch (e 845-S only)
- Neodymium ferrous magnet with baron—keeps mic stable regardless of climate
- Fall control—consistent sound over time
- On/off switch (lead switch) can be locked in the on position if required

Technical Data

Pick-up pattern	super-cardioid
Frequency response	40-16,000 Hz
Sensitivity (free field, no load)(1 kHz)	1.8 mV/Pa at 1 kHz, nominal
Nominal impedance	350 Ω
Min. terminating impedance	1 k
Dimensions in inches	1.81 W x 7.28 L
Weight	11.61 oz

Product Variants

e845, e 845, EPACK845 (e 845, Stand, Cable), THREEPACK 845(3x 845), e 845-S (with on/off switch)

Supplied Accessories

T protective pouch
TM20 800 microphone clip

Optional Accessories

21-30LR microphone cable	MC21N
Gray foam windscreen	MCW 4032
Floor stand with boom arm	SEW/S 3000

Architect's Specifications

The unit shall be a supercardioid vocal microphone with a hum compensating coil with a frequency response of 40 Hz-16 kHz. The unit shall have a sensitivity (free field, no load) of 1.8 mV/Pa at 1 kHz and have nominal impedance of 350 Ω. The min. terminating impedance shall be 1 k. The unit shall be dimensions 1.81 x 7.28 inches, and weight 11.61 oz. The unit shall be a serial number e845 microphone.

B-1 fp_tdica.m

```
% Time Domain ICA with Natural Gradient Algorithm
close all, clear all, clc;
mic_1=wavread('nama_file1');
mic_2=wavread('nama_file2');

mic_1=mic_1'; %Reading file from microphone #1.
mic_2=mic_2'; %Reading file from microphone #2.

mic_1=mic_1(1:50000); mic_2=mic_2(1:50000);

figure(1)
subplot(211); plot(mic_1);
subplot(212); plot(mic_2);

mix=[mic_1;mic_2];
[N,P]=size(mix);
permute=randperm(N);
s=mix(permute,:);

x=s;
mx=mean(mix');
c=cov(mix');
x=x-mx'*ones(1,P);
wz=2*inv(sqrtm(c));
x=wz*x;

% w=eye(N); %inisiasi matriks pencampur
w=[1 1;1 2];
M=size(w,2);
sweep=0; oldw=w; olddelta=ones(1,N*N);
Id=eye(M);

%ITERASI TDICA
L=0.000001; B=30; for I=1:100, sep;
uu=w*wz*mix; % Pemisahan sinyal suara
uu11=uu(1,:);
uu12=uu(2,:);

% Plot sinyal estimasi TDICA
```

```

figure(2);
subplot(211); plot(uu11);
subplot(212); plot(uu12);

B-2 fp_fdica.m
% Frequency Domain ICA (FDICA)
clear all; close all; format compact; clf;

mic_1=wavread('nama_file1');
mic_2=wavread('nama_file2');

mic_1=mic_1';
mic_2=mic_2';

figure(1)
subplot(211); plot(mic_1);
subplot(212); plot(mic_2);

mic_1=mic_1(1:50000) mic_2=mic_2(1:50000);

Fs=11025;
f=[1:Fs/2];

mixfrek=[real(fft(mic_1));real(fft(mic_2))];

%plot frekuensi spektra sinyal input FDICA
figure(2);
subplot(2,1,1)
plot(f,mixfrek(1,f));
title('Frequency spectra of the microphone #1')
xlabel('Frequency (Hz)');

subplot(2,1,2)
plot(f,mixfrek(2,f));
title('Frequency spectra of the microphone #2')
xlabel('Frequency (Hz)');

[N,P]=size(mixfrek)
permute=randperm(P);
x=mixfrek(:,permute);

```

```

% Spheres the data (normalisation).
mx=mean(mixfrek');
c=cov(mixfrek');
x=x-mx'*ones(1,P);                                wz=2*inv(sqrtm(c));
x=wz*x;

w=[1 1;1 2];
% w=pi^2*rand(N);
M=size(w,2);                                     % M=N usually
sweep=0; oldw=w; olddelta=ones(1,N*N);
Id=eye(M);

%ITERASI FDICA
L=0.000001; B=30; for I=1:100; sep;
end;

yy=w*wz*mixfrek;
yy11=yy(1,f);
yy12=yy(2,f);

% plot frekuensi spektra sinyal estimasi
figure(3)
subplot(211); plot(yy11);
subplot(212); plot(yy12);

% Transform signals back to time domain.
yy11=real(ifft(yy(1,:)));
yy12=real(ifft(yy(2,:)));

% Plot time domain sinyal estimasi akhir
figure(4)
subplot(211); plot(yy11);
subplot(212); plot(yy12);

```

B-3 Sep.m

```

% SEP goes once through the scrambled mixed speech
signals, x
% (which is of length P), in batch blocks of size B,
adjusting weights,

```

```

% w, at the end of each block.
%
% I suggest a learning rate L, of 0.01 at least for
2->2 separation.
% But this will be unstable for higher dimensional
data. Test it.
% Use smaller values. After convergence at a value
for L, lower
% L and it will fine tune the solution.
%
% NOTE: this rule is the rule in our NC paper, but
multiplied by w^T*w,
% as proposed by Amari, Cichocki & Yang at NIPS '95.
This `natural'
% gradient' method speeds convergence and avoids
the matrix inverse in the
% learning rule.

sweep=sweep+1; t=1;
noblocks=fix(P/B);
BI=B*Id;
for t=t:B:t-1+noblocks*B,
    u=w*x(:,t:t+B-1);
    w=w+L*(BI+(1-2*(1./(1+exp(-u))))*u')*w;
end;
sepout

```

B-4 sepout.m

```
% SEPOUT - put whatever textual output report you
want here.
% Called after each pass through the data.
% If your data is real, not artificially mixed,
you will need
% to comment out line 4, since you have no idea
what the matrix 'a' is
```

```
[change,olddelta,angle]=wchange(oldw,w,olddelta);
oldw=w;
fprintf('****sweep=%d, change=% .4f angle=% .1f deg.,
[N%d,M%d,P%d,B%d,L%.5f] \n',...
sweep,change,180*angle/pi,N,M,P,B,L);
% w*wz*a      %should be a permutation matrix for
artif. mixed data.
```

B-5 wchange.m

```
function
[change,delta,angle]=wchange(w,oldw,olddelta)
[M,N]=size(w); delta=reshape(oldw-w,1,M*N);
change=delta*delta';
angle=acos((delta*olddelta')/sqrt((delta*delta')*(ol
ddelta*olddelta')));
```

B-6 fp_if.m

```
close all
clear all
clc
%r=11025;
%[d]=yy12;
[d,r]=wavread('nama_file.wav');
[I,S]=ifgram(d,256,256,128,r);
[R,M]=extractrax(abs(S));
a=size(R);
F=colinterpvals(R,I);
P=-colinterpvals(R,unwrap(angle(S)));
size(F);

tt=[1:a(2)]*128/r;           % default spectrogram
step is NFFT/2 i.e 128
```

```

size(tt);

specgram(d,256,r);
hold on;
figure; plot (tt,F','r');

```

B-7 ifgram.m

```

function [F,D] = ifgram(X, N, W, H, SR)
% [F,D] = ifgram(X, N, W, H, SR)
Instantaneous frequency by phase deriv.
% X is a 1-D signal. Process with N-point FFTs
applying a W-point
% window, stepping by H points; return (N/2)+1
channels with the
% instantaneous frequency (as a proportion of
the sampling rate)
% obtained as the time-derivative of the phase
of the complex
% spectrum
% as described by Toshihiro Abe et al in
ICASSP'95, Eurospeech'97
% Same arguments and some common code as
dpwebox/stft.m.
% Calculates regular STFT as side effect -
returned in D.
% after 1998may02 dpwe@icsi.berkeley.edu
% 2001-03-05 dpwe@ee.columbia.edu revised version
% 2001-12-13 dpwe@ee.columbia.edu Fixed to work
when N != W
% $Header: $

if nargin < 2
    N = 256;
end
if nargin < 3
    W = N;
end
if nargin < 4
    H = W/2;
end

```

```

if nargin < 5
    SR = 1;
end

s = length(X);
% Make sure it's a single row
if size(X,1) > 1
    X = X';
end

%win = [0,hanning(W-1)'];
win = 0.5*(1-cos([0:(W-1)]/W*2*pi));

% Window for discrete differentiation
T = W/SR;
dwin = -pi / T * sin([0:(W-1)]/W*2*pi);

% sum(win) takes out integration due to window, 2
% compensates for neg freq
norm = 2/sum(win);

% How many complete windows?
nhops = 1 + floor((s - W)/H);

F = zeros(1 + N/2, nhops);
D = zeros(1 + N/2, nhops);

nmwl = floor( (N-W)/2 );
nmw2 = N-W - nmwl;

ww = 2*pi*[0:(N-1)]*SR/N;

for h = 1:nhops
    u = X((h-1)*H + [1:W]);
    % if(h==0)
    % plot(u)
    % end
    % Apply windows now, while the length is right
    wu = win.*u;
    du = dwin.*u;

```

```

% Pad or truncate samples if N != W
if N > W
    wu = [zeros(1,nmw1),wu,zeros(1,nmw2)];
    du = [zeros(1,nmw1),du,zeros(1,nmw2)];
end
if N < W
    wu = wu(-nmwl+[1:N]);
    du = du(-nmwl+[1:N]);
end
% FFTs of straight samples plus differential-
weighted ones
t1 = fft(fftshift(du));
t2 = fft(fftshift(wu));
% Scale down to factor out length & window
effects
D(:,h) = t2(1:(1 + N/2))'*norm;

% Calculate instantaneous frequency from phase of
differential spectrum
t = t1 + j*(ww.*t2);
a = real(t2);
b = imag(t2);
da = real(t);
db = imag(t);
instf = (1/(2*pi))*(a.*db - b.*da)./(a.*a +
b.*b);
% 1/2pi converts rad/s into cycles/s
% sampling rate already factored in as constant
in dwin & ww
% so result is in Hz

F(:,h) = instf(1:(1 + N/2))';

end;

```

B-8 extractrax.m

```

function [trax,mags,dbgs] = extractrax(s,h,verbose)
% [T,M] = extractrax(S,H,VERB) Extract tracks from
a 2-d t-f style array
%      S is a matrix of values, which is searched
from left to right

```

```

%      looking for local maxima in each column,
then tracking their
%      evolution.  T is returned with one row for
each track that
%      is found, with nonzero values indicating
the continuously-
%      valued row (i.e. frequency, but starting
from 1.0) of the
%      track in the original matrix.
%      H is a threshold; only peaks > H*maxx(S) at
startup
%      are tracked. H defaults to 0.01.
%      Rows of T are sorted by column (i.e. start
time), then by freq,
%      of their first value. VERB=1 for progress
printouts.
% M returns the interpolated peak magnitude for
each track point.
% 1998may02 dpwe@icsi.berkeley.edu $Header: $ how
many times?

% Performance constants:

% Default proportion-of-max startup threshold
cutoff
defproptthresh = 0.01;

% Ratio of peak-height-to-start vs peak-height-to-
continue (hysteresis)
startupmargin = 5.0;

% Maximum col-to-col tolerable peak movement (in
bins)
maxdfdt = 1.0;

% How many steps before say a peak is finished
deadsteps = 3;

% However, energy in the channel below this factor
times the last
% seen peak means it really has gone

```

```

lowEthresh = 0.2;

% Prune away tracks that end up with fewer points
% than this
minNpts = 10;

% Maximum mag-increase before new track (dB/step)
maxdBincr = 20.0;

% Asymptote of adaptive magnitude-increase
% threshold level (dB/step)
finaldBincr = 1.5; % was 6.0

% Adptv mag thresh stays this far above allowed
% positive slopes (must be > 1.0)
dbIncrSafeRatio = 1.5;

% Decay time for adptv mag incr threshold (steps)
dbIncrTsteps = 20;

%%%

if (nargin < 2)
    h = defproptthresh;
end

if (nargin < 3)
    verbose = 0;
end

% Figure the absolute threshold
startupThresh = h*max(max(s));

[nr,nc] = size(s);

%% % Track all the local maxes we looked at
%% d = zeros(size(s));

% 'frequencies' (i.e. bin indexes) of current
% track's most recent peaks
Fs = [];

```

```

% 'energies' (i.e. values) of current track's most
recent peaks
Es = [];
% Start time of each track (to calc decaying mag
thresh)
Ss = [];
% 'times' (column indices) of current track's most
recent peaks
% this can be less than the previous column because
of permitted bridging
Ts = [];
% row numbers within the output "trax" array for
the data records
% of each current track
Rs = [];
% Current magnitude threshold limit. This adapts
Ms = [];

blockrows = 50;
tblock = NaN*ones(blockrows, nc);
ntrax = 0;
% initialize the data array
trax = tblock;
% Also follow our extracted magnitudes, for yuks
mags = tblock;
% Debug parameters the same size
dbgs = tblock;

txsize = size(trax, 1);

% Do the columnwise search
for c = 1:nc
    col = s(:,c)';

    % find locations and heights of best local maxes
    % (return peaks even startupmargin smaller than
startupThresh,
    % since that's the acceptable continuation
height; filter later)
    % original: fit peaks in linear domain

```

```

% [maxFs, maxEs] = quadmaxloc(col,
startupThresh/startupmargin);
% average |sgram| dB difference, weighted by
resynth |sgram| = 0.3585 dB
% new: fit peaks in dB domain:
[maxFs, maxEs] = quadmaxloc(log(col),
log(startupThresh/startupmargin));
maxEs = exp(maxEs);
% average |sgram| dB difference, weighted by
resynth |sgram| = 0.0868 dB
% - big improvement!

% Match up to existing tracks
numcurrent = size(Fs, 2);
% Setup mask to indicate which tracks should be
terminated this time
allowCont = ones(1,numcurrent);
for i = 1:numcurrent
    % fprintf(1, 'col %d: extending current#\n'
(r=%d) f=%f\n', c, i, Rs(i), Fs(i));
    f = Fs(i);
    foundcont = 0;
    % Unfortunately, when maxFs is empty, this
test crashes, so have to
    % do it as 2 nested tests.
    %if size(maxFs,2) > 0 & abs(f -
maxFs(bestix)) < maxdfdt
        if size(maxFs,2) > 0
            dfs = abs(maxFs - f);
            bestix = find(dfs == min(dfs));
            if size(maxFs,2) > 0 & abs(f - maxFs(bestix))
< maxdfdt
                % Found a continuation for this current
peak
                % Check that it doesn't fail the magnitude
increase check
%         magthr = finaldBincr + (maxdBincr-
finaldBincr)*exp(-(c - Ss(i))/dbIncrTsteps);
            magthr = Ms(i);
            magdBStep = 20*log10(maxEs(bestix)/Es(i));
            if magdBStep > Ms(i)

```

```

    % Step was too big: Kill this one & leave the
    continuation unused
    % so that a new track is created with it
    fprintf(1, 'col %d (l=%d) tk %d: large step
of %f at %f, breaking track\n', c, c-Ss(i), i,
magdBStep, f)
    allowCont(i) = 0;
else
    Fs(i) = maxFs(bestix);
    Es(i) = maxEs(bestix);
    Ts(i) = c;
    % delete the taken peak fro maxFs & maxEs
    mask = [1:size(maxFs,2)] ~= bestix;
    maxFs = maxFs(find(mask));
    maxEs = maxEs(find(mask));
    % Store 'freq' in output array
    trax(Rs(i), c) = Fs(i);
    mags(Rs(i), c) = Es(i);
    dbgs(Rs(i), c) = Ms(i);
    foundcont = 1;
    % Update the mag threshold
    Ms(i) = max(dbIncrSafeRatio*magdBStep, ...
                Ms(i) - (Ms(i)-finaldBincr)*(1-exp(-
1/dbIncrTsteps)));
end
end
end
if foundcont == 0
    % No acceptable continuation found - maybe
kill this track
    if (((c - Ts(i)) > deadsteps) |
(col(round(Fs(i))) < lowEthresh*Es(i)))
        % Mark this track to be removed from
current structures
        allowCont(i) = 0;
    end
    % otherwise, let it hang on for a bit
    % Set a value for the frequency anyway? Else
could leave gaps
    % in frq to indicate untracked parts...
%trax(Rs(i), c) = Fs(i);

```

```

        %mags(Rs(i), c) = Es(i);
    end
end % of loop over currently active tracks

% Remove the tracks that have ended from the
current records
if min(allowCont) == 0
    Fs = Fs(find(allowCont));
    Es = Es(find(allowCont));
    Ts = Ts(find(allowCont));
    Rs = Rs(find(allowCont));
    Ss = Ss(find(allowCont));
    Ms = Ms(find(allowCont));
end

% Create new tracks for all remaining peaks
for i = 1:(size(maxFs,2))
    if maxEs(i) > startupThresh
        if verbose ~= 0
            fprintf(1, 'Creating track at col %d
bin %f\n', c, maxFs(i));
        end
        ntrax = ntrax + 1;
        tkrow = ntrax;
        if ntrax > txsize
            % Need to expand the pre-allocated trax
buffer
            trax = [trax; tblock];
            mags = [mags; tblock];
            dbgs = [dbgs; tblock];
            txsize = size(trax,1);
            fprintf(1, 'trax extended to %d rows\n',
txsize);
        end
        % trax = [trax; NaN*ones(1,nc)];
        % mags = [mags; NaN*ones(1,nc)];
        % dbgs = [dbgs; NaN*ones(1,nc)];
        % tkrow = size(trax,1);
        Rs = [Rs, tkrow];
        Fs = [Fs, maxFs(i)];
        Es = [Es, maxEs(i)];
    end
end

```

```

Ts = [Ts, c];
Ss = [Ss, c];
Ms = [Ms, maxdBincr];
j = size(Rs,2);
trax(Rs(j), c) = Fs(j);
mags(Rs(j), c) = Es(j);
dbgs(Rs(j), c) = Ms(j);
end
end

end

% Prune away tracks with too few pts (including
% pre-allocated empty ones!)
keeprows = sum(~isnan(trax)') >= minNpts;
trax = trax(find(keeprows),:);
mags = mags(find(keeprows),:);
dbgs = dbgs(find(keeprows),:);

% Offset freqs to start from zero, not 1 (2002-03-
04)
trax = trax - 1;

% foreach col
%   find the local maxes
%   foreach local max
%     find the 'exact' location
%   foreach current track
%     if (nearest new peak)
%       extend track
%       remove that peak
%     elseif bridging less than max and actual val
not too small
%       guess-extend-track
%     else
%       remove track from active
%     foreach remaining new peak
%       create new active track
% done

```

```

%%%%%%%%%%%%%
Helper function
%%%%%%%%%%%%%
function [xmax,ymax] = quadmaxloc(v,th)
% [X,Y] = quadmaxloc(V,T) Quadratic-interpolated
index and values for loc maxs
% V is a uniformly-sampled vector. Find all
local maxes above T
% (absolute), then do a quadratic fit to
interpolate the location and
% height of the maxima. Return these as
corresponding elements of X
% and Y.
% 1998may02 dpwe@icsi.berkeley.edu $Header:
$ again?

if (nargin < 2)
    th = 0;
end

if (size(v,1) > 1)
    error('v must be a row vector');
end

nr = size(v,2);

% filter for local maxima; ensure edges don't win
gtl = (v > [v(1), v(1:(nr-1))]);
% allow greater-than-or-equal to catch plateaux
gtu = (v >= [v(2:nr), 1+v(nr)]);
vmax = v .* (v > th) .* gtl .* gtu;
maxixs = find(vmax);
% Interpolate the max pos's
xmax = zeros(size(maxixs));
ymax = zeros(size(maxixs));
for i = 1:size(maxixs,2)
    % Solve quadratic fit to 3 pts (as y = ax(x-b)
    % with 0,0 as col(rmax-1))
    rmax = maxixs(i);
    y1 = v(rmax)-v(rmax-1);
    y2 = v(rmax+1)-v(rmax-1);

```

```

a = (y2 - 2*y1)/2;
b = 1-y1/a;
xmax(i) = rmax-1+b/2;
ymax(i) = v(rmax-1)-a*b*b/4;
end

```

B-9 colinterpvals.m

```

function V = colinterpvals(T,M)
% V = colinterpvals(T,M) Interpolate columns from
real-valued bin tracks
% Each row of T defines a 'track' as a
continuously-valued 'bin index'
% (where zero indicates the 1st bin i.e. NOT
matlab-style indices)
% for the corresponding column of matrix M; the
conformal value of
% V is the linear interpolation of M at that bin.
% 1998may03 dpwe@icsi.berkeley.edu for AHI tracking
% $Header: $

```

```

[nr, nc] = size(M);

% Preallocate results array
V = zeros(size(T));

[nt,ntc] = size(T);

for c = 1:nc
    mc = M([1:nr] nr],c);
    tc = T(:,c)+1; % add 1 to make it valid matlab
indexing
    bin = floor(tc);
    binOK = (bin >= 1) & (bin <= nr);
    bin(find(~binOK)) = ones(1,sum(~binOK));
    base = mc(bin);
    delta = mc(bin+1) - base;
    % adding the tc restores the NaNs
    vals = base+(tc-bin).*delta;
    V(:,c) = vals;
end

```

Keterangan:

- Program yang di-run hanya main program (bukan fungsi/function).
- Semua file diletakkan dalam satu directory matlab.
- Untuk MSICA (multistage ICA) dapat dilakukan dengan menggabungkan fp_tdica.m dan fp_fdica.m.

BIODATA PENULIS



Penulis dilahirkan di Wonorejo, Pasuruan, Jawa Timur pada tanggal 10 Agustus 1986. Pendidikan SD dan SMP ditempuh di SDN Gulun III (1994-1999) dan SMP 1 Maospati (1999-2002), Magetan, tempat dimana penulis dibesarkan. Pendidikan menengah atas penulis selesaikan di SMA Darul Ulum 2 – BPPT, Peterongan – Jombang (2002-2005). Pada tahun itu juga penulis diterima di Jurusan Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. Di Teknik Fisika, penulis mengambil spesialisasi bidang minat akustik dan getaran, terutama pada analisa serta pemrosesan sinyalnya. Penulis bisa dihubungi lewat email di bagustris@yahoo.com.