

MAKALAH SEMINAR TUGAS AKHIR
**PEMISAHAN SINYAL AUDIO TERCAMPUR SECARA OFFLINE
DENGAN ALGORITMA INDEPENDENT COMPONENT ANALYSIS**

Syarif¹⁾, Achmad Hidayatno, ST.,MT.²⁾, Ajub Ajulian Zahra, ST.,MT.²⁾

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro

Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

e-mail : syarifby@yahoo.com

Abstract

Mixed audio signals can be found in a room where some people are speaking simultaneously, such as cocktail party or meeting. Someone's ears will process the mixed signals automatically to get the meant audio signal, so the person can understand the message from the other. The mixed signals have random variables which aren't determined before. The mixed signals separation method is called Blind Source Separation (BSS).

Independent Component Analysis is a method (algorithm) for performing BSS. The principle of ICA is to estimate un-noised audio signals (closely related to the original signals) from the available mixed signals. A simulation program is needed to execute ICA in offline way. The program in this paper uses Matlab 7.1. It produces estimated audio signals and MSE (Mean Square Error) value between the estimated and original signals.

The result of test indicates that FastICA algorithm is the efficient method to compute parameters quickly to get the right estimated signals. The most effective model of this algorithm is symmetric approach.

Kata kunci: audio, offline, FastICA, MSE.

I PENDAHULUAN

1.1 Latar Belakang

Isyarat suara tercampur bisa ditemui pada suatu pembicaraan pada sebuah forum pertemuan banyak orang, misalnya : pesta atau rapat. Pembicaraan ini melibatkan lebih dari dua orang pembicara pada forum itu dalam waktu yang sama. Indra pendengaran seseorang akan melakukan pengolahan isyarat yang tercampur untuk mendapatkan isyarat suara yang diinginkan. Pada tugas akhir ini, teknik pemisahan isyarat suara tercampur akan dikaji dengan menggunakan algoritma ICA (*Independent Component Analysis*). Seiring berkembangnya teknologi yang mencakup berbagai bidang, termasuk medis, pengolahan digital (citra dan suara) serta telekomunikasi, algoritma ICA juga bisa diterapkan dengan baik untuk mengetahui sifat komunikasi isyarat syaraf otak manusia, mencari faktor tersembunyi pada data finansial suatu perusahaan, memisahkan derau pada citra digital serta untuk meningkatkan kinerja jaringan sistem telekomunikasi. Semua itu bertujuan untuk memenuhi kebutuhan manusia.

Prinsip algoritma ICA adalah menaksir isyarat suara tanpa derau dari isyarat suara tercampur yang diketahui. Dalam tugas akhir ini dibuat sebuah tampilan perangkat peraga untuk menunjukkan kinerja dari pemisahan isyarat sumber suara dengan

menghitung nilai MSE (*Mean Square Error*), sehingga suara yang diinginkan dapat didengar dengan baik tanpa adanya campuran suara lain.

1.2 Tujuan Tugas Akhir

Tujuan penelitian tugas akhir ini untuk :

1. Menggambarkan pemisahan isyarat suara tercampur dan mendapatkan isyarat suara taksiran dengan algoritma ICA (*Independent Component Analysis*).
2. Menganalisis MSE (*Mean Square Error*) dari isyarat suara taksiran terhadap isyarat sumber.

1.3 Pembatasan Masalah

Dalam penulisan tugas akhir ini, penulis membatasi permasalahan sebagai berikut :

1. Data berupa dua isyarat suara dengan format (.wav) PCM 8KHz, 8 bit *mono*.
2. Algoritma yang digunakan adalah ICA (*Independent Component Analysis*).
3. Perangkat lunak peraga menggunakan program Matlab 7.1 berbasis GUI (*Graphical User Interface*).
4. MSE (*Mean Square Error*) sebagai parameter untuk mengetahui kemiripan isyarat suara.
5. Hasil pemisahan isyarat suara disajikan secara *offline* (rekaman).

II DASAR TEORI

2.1 Konsep ICA ^[1]

Analisis Komponen Bebas atau *Independent Component Analysis* (ICA) adalah teknik pemrosesan isyarat untuk menemukan faktor-faktor atau komponen tersembunyi yang membentuk sekumpulan

¹⁾Mahasiswa Teknik Elektro

²⁾Dosen Teknik Elektro

peubah acak sebuah data. Variabel tersembunyi dianggap *nongaussian* dan secara statistik saling bebas satu sama lain. ICA didefinisikan pada Persamaan 1.

$$x_k = a_{k1}s_1 + a_{k2}s_2 + \dots + a_{kn}s_n \quad (1)$$

dengan $k = 1, 2, \dots, m$ dan x_k adalah isyarat suara tercampur; $a_{k1}, a_{k2}, \dots, a_{kn}$ adalah variabel pencampur yang menyatakan bobot dari sumber dan s_1, s_2, \dots, s_n menyatakan isyarat suara asli dari sumber. Jika

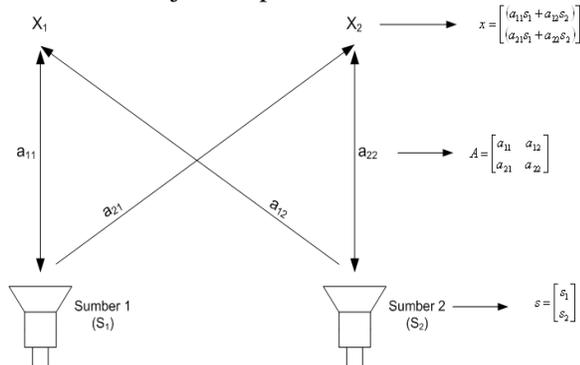
Dalam bentuk ICA yang paling sederhana, variabel acak skalar x_k dianggap sebagai gabungan linear dari n komponen bebas (secara statistik bebas dan mempunyai rerata nol) s_1, s_2, \dots, s_n yang tidak diketahui. Secara umum, huruf kecil tebal menandakan vektor dan huruf besar tebal menandakan matriks. Dengan menggunakan notasi matriks, ICA ditunjukkan pada Persamaan 2.

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2)$$

Komponen \mathbf{s} bersifat bebas dan memiliki distribusi *nongaussian*. Matriks pencampuran \mathbf{A} tidak diketahui. Hal yang diamati adalah vektor acak \mathbf{x} dan yang harus dilakukan adalah memperkirakan baik \mathbf{A} maupun \mathbf{s} dengan vektor tersebut. Setelah menghitung matriks \mathbf{A} , akan didapatkan *invers* dari matriks \mathbf{A} yang disebut matriks \mathbf{W} , sehingga komponen bebas \mathbf{s} diperoleh dengan Persamaan 3.

$$\mathbf{s} = \mathbf{W}\mathbf{x} \quad (3)$$

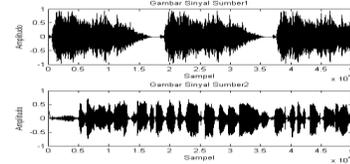
Pada tugas akhir ini, hanya dipakai dua buah sumber suara saja. Proses pencampuran isyarat dengan rumus ICA ditunjukkan pada Gambar 1.



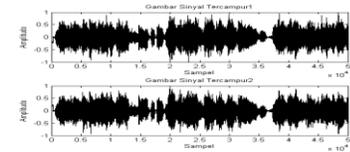
Gambar 1 Proses Pencampuran Isyarat dengan ICA

Gambar 2 menunjukkan konsep ICA, bagian (a) menunjukkan isyarat asli dari masing-masing sumber suara. Isyarat tercampur diperlihatkan pada bagian (b). Permasalahannya adalah memperoleh kembali isyarat seperti pada (a) hanya menggunakan data pada (b). Pada model ICA, varians dan urutan dari komponen bebas tidak dapat ditentukan secara pasti sehingga perlu ada pendekatan untuk memecahkan permasalahan itu.

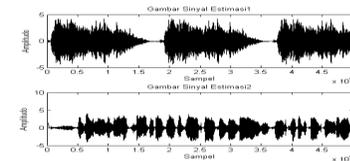
Pendekatan tersebut berupa beberapa informasi tentang sifat-sifat statistik dari isyarat sumber untuk memperoleh parameter-parameter campuran. Sebuah metode ICA digunakan untuk memperkirakan parameter-parameter tersebut berdasarkan informasi tentang isyarat yang bersangkutan, yang memungkinkan untuk memisahkan isyarat-isyarat asli dari campurannya. Gambar 2 bagian (c) memperlihatkan dua isyarat hasil perkiraan dengan metode ICA. Seperti yang tampak bahwa isyarat perkiraan ini sangat mirip dengan isyarat aslinya.



(a) Isyarat asli



(b) Isyarat campuran yang diamati



(c) Perkiraan dari isyarat sumber asli

Gambar 2 Gambaran konsep ICA

2.2 Prinsip Perkiraan ICA [1]

2.2.1 Nongaussian

Pada teorema batas pusat (*central limit theorem*) dinyatakan bahwa fungsi distribusi peluang dari penjumlahan variabel acak bebas dengan jumlah yang banyak mendekati sebuah distribusi *Gaussian*. Hal ini berarti penjumlahan dari dua buah variabel acak bebas biasanya mempunyai distribusi yang lebih mendekati distribusi *Gaussian* daripada kedua variabel acak tersebut.

Dalam tugas akhir ini, untuk memperkirakan satu dari komponen-komponen bebas yang merupakan sebuah kombinasi linear dari \mathbf{x} seperti pada persamaan $\mathbf{s} = \mathbf{W}\mathbf{x}$, maka dinyatakan persamaan $\mathbf{y} = \mathbf{w}\mathbf{x}$, dengan \mathbf{w} adalah vektor yang akan dicari. Jika \mathbf{w} adalah salah satu baris dari \mathbf{W} (*invers* dari \mathbf{A}), kombinasi linear tersebut sebenarnya sama dengan satu dari komponen-komponen bebas tersebut. Variabel \mathbf{s} memiliki distribusi seragam dan \mathbf{w} sebagai suatu vektor yang memaksimumkan nongaussianitas dari \mathbf{y} . Terdapat dua kemungkinan nilai \mathbf{s} , yaitu \mathbf{s} dan $-\mathbf{s}$ (komponen bebas

hanya dapat diperkirakan sampai besar perkalian bukan tandanya). Dibutuhkan sederet perhitungan untuk menentukan nilai-nilai maksimum dari komponen bebas tersebut. Hal ini tidak sulit mengingat komponen-komponen bebas yang berbeda tidak berkorelasi.

2.2.2 Negentropi

Tingkat nongaussianitas pada sebuah variabel acak perkiraan ICA dapat diukur secara kuantitatif. Untuk menyederhanakan permasalahan diasumsikan bahwa \mathbf{y} mempunyai rerata nol dan varians satu. Salah satu cara mengukur tingkat nongaussianitas adalah dengan menggunakan negentropi yang didasarkan pada kuantitas teoritis informasi dari entropi.

Menurut teori informasi, sebuah variabel *Gaussian* mempunyai entropi terbesar diantara semua variabel acak yang mempunyai varians yang setara. Hal ini berarti bahwa entropi dapat digunakan mengukur gaussianitas. Untuk memperoleh pengukuran nongaussianitas yang selalu tidak negatif dan nol untuk sebuah variabel *Gaussian*, digunakanlah negentropi, ditunjukkan pada Persamaan 4.

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y}) \quad (4)$$

dengan $\mathbf{y}_{\text{gauss}}$ adalah sebuah variabel acak *Gaussian* dengan matriks kovarians yang sama dengan \mathbf{y} . Sehingga, negentropi adalah selalu tidak negatif dan akan nol jika dan hanya jika \mathbf{y} mempunyai distribusi *Gaussian*.

Persamaan 4 dapat ditulis kembali dengan pendekatan lain menjadi Persamaan 5.

$$J(\mathbf{y}) \approx \left[E\{g(\mathbf{y})\} - E\{g(\mathbf{y}_{\text{gauss}})\} \right]^2 \quad (5)$$

Sangat dianjurkan menggunakan nonlinearitas (fungsi perkiraan) dalam menentukan komponen bebas. Dua buah fungsi nonlinearitas (g) berikut yang digunakan dalam tugas akhir ini (sebagaimana yang ditunjukkan pada Persamaan 6 dan 7) merupakan fungsi nonlinearitas yang paling baik (bersifat meningkat secara lambat dan tahan dari gangguan).

$$g_1(\mathbf{u}) = -\exp\left(-\frac{\mathbf{u}^2}{2}\right), \quad g_1'(\mathbf{u}) = \mathbf{u} \exp\left(-\frac{\mathbf{u}^2}{2}\right) \quad (6)$$

$$g_2(\mathbf{u}) = \frac{1}{a} \log \cosh a\mathbf{u}, \quad g_2'(\mathbf{u}) = \tanh(a\mathbf{u}) \quad (7)$$

dengan a merupakan konstanta. Dalam tugas akhir ini digunakan nilai $a = 1$.

2.3 Komputasi *FastICA* ^{[1], [5], [8]}

2.3.1 Langkah-Langkah Sebelum Melakukan ICA

Sebelum melakukan proses ICA untuk menentukan komponen bebas dari beberapa data,

perlu dilakukan penerapan beberapa teknik untuk mendapatkan perhitungan ICA yang efisien, yang biasa disebut dengan *FastICA* (perhitungan ICA dengan hasil yang mendekati kebenaran).

1 Pemusatan Data

Hal yang mendasar yang perlu dilakukan adalah dengan memusatkan data \mathbf{x} . Prinsipnya mengurangi data tersebut dengan rerata data dari vektor yang ada ($\mathbf{m} = E\{\mathbf{x}\}$), sehingga \mathbf{x} menjadi variabel bererata nol (*zero-mean*). Akibatnya, \mathbf{s} merupakan variabel yang bererata nol. Pemusatan data ini semata-mata untuk membuat algoritma ICA menjadi mudah dan cepat. Setelah melakukan taksiran matriks pencampur \mathbf{A} dengan data memusat, kita melengkapi taksiran tersebut dengan menambahkan vektor rerata \mathbf{s} kepada hasil taksiran \mathbf{s} yang memusat. Vektor rerata \mathbf{s} adalah hasil kali *invers* matriks pencampur dan vektor \mathbf{x} yang memusat (\mathbf{m}), ditunjukkan pada Persamaan 8.

$$\mathbf{s} = \mathbf{A}^{-1} \mathbf{m} = \mathbf{W} \mathbf{m} \quad (8)$$

2 Pemutihan (Whitening)

Pemutihan (*whitening*) merupakan praproses yang berfungsi untuk me“mutih”kan variabel yang diamati. Sehingga didapatkan sebuah vektor baru yang variansnya sama dengan satu serta komponen nyata dan imajinernya tidak berkorelasi dengan nilai varians yang sama. Secara singkat, matriks kovarians dari $\tilde{\mathbf{x}}$ sama dengan matriks identitas (*orthogonal*) dan digambarkan dalam persamaan 9.

$$E\{\tilde{\mathbf{x}} \tilde{\mathbf{x}}^T\} = \mathbf{I} \quad (9)$$

Metode yang sering digunakan adalah dekomposisi nilai *eigen* (*eigenvalue decomposition* – EVD) dari matriks kovarians $E\{\mathbf{x}\mathbf{x}^T\}$. Pada tahap ini, terlebih dahulu mengamati nilai dan vektor *eigen* dari $E\{\mathbf{x}\mathbf{x}^T\}$, seperti yang dilakukan pada Analisis Komponen Utama (*Principal Component Analysis* – PCA). Persamaan untuk mengamati nilai dan vektor *eigen* dirumuskan dengan Persamaan 10.

$$\mathbf{x} \lambda = \mathbf{D} \lambda \quad (10)$$

dengan \mathbf{x} menunjukkan matriks yang akan dicari nilai dan vektor *eigen*nya. \mathbf{D} merupakan matriks nilai *eigen* berbentuk diagonal dan λ adalah vektor *eigen* dari matriks \mathbf{x} . Untuk mendapatkan Persamaan 9, dilakukan pemutihan isyarat yang dapat dilakukan dengan Persamaan 11.

$$\tilde{\mathbf{x}} = \mathbf{D}^{-1/2} \lambda^T \mathbf{x} \quad (11)$$

Nilai $\mathbf{D}^{-1/2} \lambda^T$ disebut sebagai matriks pemutih (*whitening*) dan $\lambda \mathbf{D}^{1/2}$ untuk mengembalikan proses disebut invers matriks pemutih (*dewithening*).

2.3.2 Algoritma *FastICA*

Setelah data mengalami praproses yaitu pemusatan dan pemutihan. Kemudian data diolah dengan suatu metode yang efisien yang disebut *FastICA*. Algoritma *FastICA* dilakukan setiap iterasi.

1. Memperkirakan satu unit komponen bebas (Pendekatan paralel)

Berikut adalah bentuk algoritma *FastICA* untuk satu unit komponen bebas.

- Memilih sebuah nilai awal vektor kompleks \mathbf{w} , dapat secara acak
- Menghitung nilai \mathbf{w} yang baru dengan Persamaan 12.

$$\mathbf{w}^+ = \mathbf{w} - \frac{E\{\mathbf{x}g(\mathbf{w}^T \mathbf{x})\} - \beta \mathbf{w}}{E\{g'(\mathbf{w}^T \mathbf{x})\} - \beta} \quad (12)$$

Fungsi g pada langkah ini adalah fungsi nonlinearitas. Pilihan g yang digunakan ditunjukkan pada Persamaan 6 dan 7.

- Menormalkan nilai \mathbf{w} yang baru (Persamaan 13).

$$\mathbf{w} = \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|} \quad (13)$$

- Memeriksa konvergensi, bila tidak konvergen maka kembali ke langkah b. Konvergen pada langkah d berarti bahwa nilai \mathbf{w} yang baru dan yang lama mempunyai arah yang sama, yaitu perkalian titik diantaranya sama dengan 1. Konvergensi dapat tercapai jika galat yang terbentuk bernilai di bawah ambang batas nilai yang dimasukkan. Untuk mempermudah maka didefinisikan *epsilon* yang merupakan nilai kesalahan terbesar (toleransi) dari nilai tersebut, pada tugas akhir ini sebesar 10^{-4} .

2. Memperkirakan beberapa komponen bebas (Pendekatan simetris)

Untuk memperkirakan beberapa komponen bebas, dilakukan dengan menjalankan algoritma *FastICA* untuk satu unit di muka untuk masing-masing komponen bebas dengan vektor $\mathbf{w}_1, \dots, \mathbf{w}_n$. Setelah itu dilakukan dekorelasi untuk masing-masing *output* \mathbf{w} setiap iterasi, untuk menjaga agar vektor yang berbeda tidak konvergen ke besaran yang sama.

Pendekatan ini berarti memperkirakan komponen bebas yang bersangkutan satu persatu. Ketika telah diperkirakan 2 komponen bebas, atau 2 vektor \mathbf{w} ($\mathbf{w}_1, \mathbf{w}_2$) lalu menjalankan algoritma *FastICA* maka

setiap akhir proses iterasi dilakukan proses penghitungan seperti Persamaan 14 berikut.

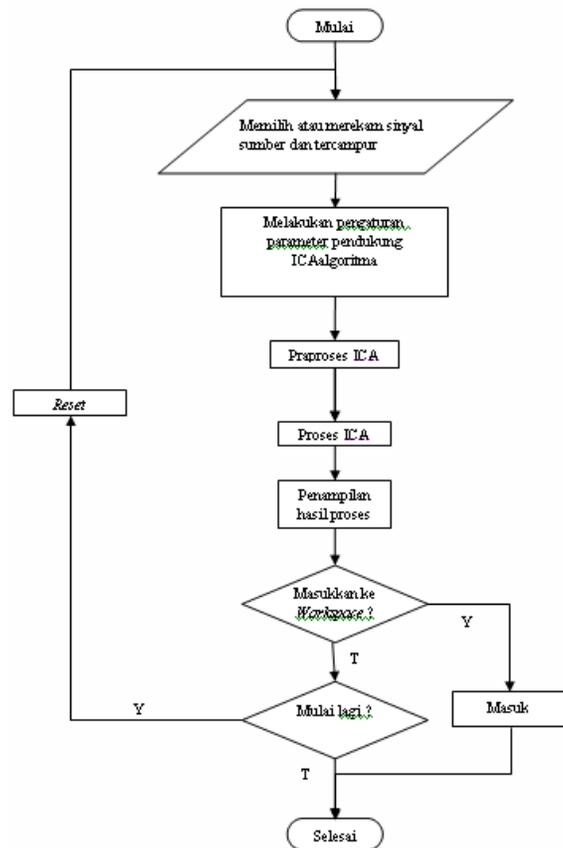
$$\mathbf{w}^+ = \mathbf{w} + \text{diag}(\alpha_i) [\text{diag}(\beta_i) + E\{g(\mathbf{y})\mathbf{y}^T\}] \mathbf{w} \quad (14)$$

dimana $\mathbf{y} = \mathbf{W}\mathbf{x}$, $\beta_i = -E\{\mathbf{y}_i g(\mathbf{y}_i)\}$, dan $\alpha_i = -1/(\beta_i - E\{g'(\mathbf{y}_i)\})$. Matriks \mathbf{W} selalu ortogonal pada akhir setiap langkah.

III PERANCANGAN SISTEM

Perancangan analisis komponen bebas ini hanya terdiri atas perangkat lunak untuk melakukan pencampuran secara linear pada isyarat masukan yang berupa isyarat suara serta melakukan ICA pada isyarat campuran secara *offline* dan menghitung kinerjanya. Perangkat lunak yang digunakan adalah MATLAB 7.1 dalam tampilan grafis.

Pada tugas akhir ini, program mengeksekusi isyarat dengan format *wav* PCM 8 KHz, 8 bit *mono*. Sumber proses perekaman ada dua isyarat. Perekaman dilakukan dengan cara membunyikan secara serempak kedua sumber dan merekamnya secara bersamaan dengan menggunakan dua buah unit komputer. Adapun diagram alir kerja program ini ditunjukkan pada Gambar 3.



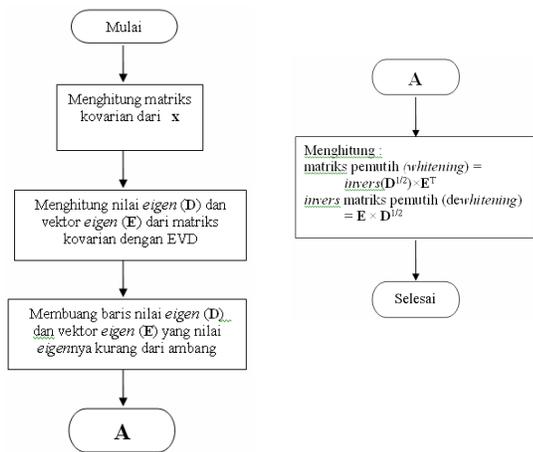
Gambar 3 Diagram Alir Kerja Program

3.1 Perangkat Lunak untuk Melakukan ICA [3], [4], [6], [7], [9]

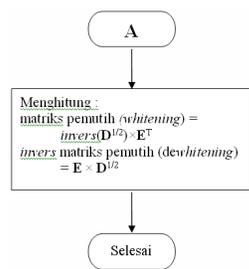
Perangkat lunak dirancang agar dapat melakukan ICA pada isyarat suara yang menjadi masukan yang kemudian diamati proses yang telah terjadi. Secara umum, dari perangkat lunak peraga ini dapat dibuat suatu diagram alir yang menggambarkan cara kerja program. Langkah-langkahnya adalah menentukan isyarat sumber dan tercampur, lalu menentukan parameter pendukung ICA, melakukan praproses dan proses ICA pada isyarat tercampur, selanjutnya adalah menampilkan hasil. Sedangkan Gambar 4 menunjukkan diagram alir fungsi untuk membuat varians data isyarat menjadi satu. Gambar 4(a) merupakan fungsi untuk melakukan PCA dan Gambar 4(b) merupakan fungsi untuk melakukan pemutihan data.

Perangkat lunak peraga ini memiliki sebelas fungsi sebagai berikut :

1. Fungsi untuk Memilih Isyarat Sumber
2. Fungsi untuk Merekam Isyarat Suara
3. Fungsi *Transpose*
4. Fungsi untuk Memusatkan Data dan Membuat Isyarat Mempunyai Varians Satu (Pemutihan)
5. Fungsi untuk Melakukan ICA pada Isyarat
6. Fungsi untuk Menghitung Kinerja
7. Fungsi untuk Melihat Grafik Isyarat
8. Fungsi untuk Mendengarkan Suara
9. Fungsi untuk Menampilkan Variabel ke dalam *Workspace* Matlab
10. Fungsi Reset
11. Fungsi Keluar



(a) Diagram Alir Fungsi Melakukan PCA



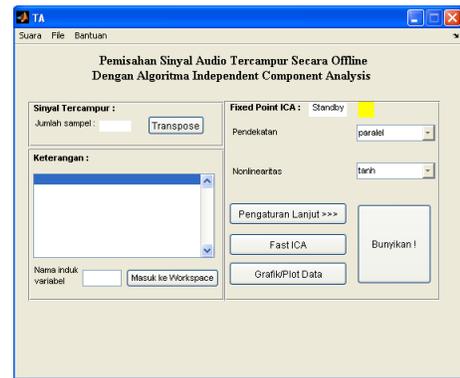
(b) Diagram Alir Fungsi Melakukan Pemutihan

Gambar 4 Diagram Alir Fungsi Untuk Membuat Varians Data Isyarat Menjadi Satu

3.2 Perangkat Lunak Peraga Berbentuk GUI [2], [3], [4], [9]

Perangkat lunak peraga berbentuk GUI dibuat dengan bantuan bahasa komputasi teknis MATLAB 7.1 yang mempunyai fasilitas antarmuka grafis bagi pengguna. Program ini dirancang untuk melakukan ICA dengan algoritma *FastICA*, dengan isyarat sumber suara dapat berasal dari *file* suara ataupun direkam terlebih dahulu. Sedangkan isyarat tercampur diperoleh dengan melakukan perekaman ataupun sudah tercampur terlebih dahulu. Hasil proses ICA kemudian disajikan dalam bentuk tampilan dan bunyi isyarat suara.

Proses pengujian bermuara pada tampilan utama program ini. Tampilan utama program yang digunakan ditunjukkan pada gambar 5.



Gambar 5 Tampilan Utama Program Peraga

Tampilan utama program ini, terdiri dari tiga bagian yaitu bagian **Isyarat Tercampur**, bagian **Fixed Point ICA** dan bagian **Keterangan**. Juga terdapat tiga buah menu yaitu menu **Suara**, menu **File**, dan menu **Bantuan**.

Ketiga bagian pada tampilan utama program dijelaskan berikut ini.

➤ Bagian Isyarat Tercampur

Bagian ini berhubungan dengan isyarat **Tercampur**. Pada bagian ini ditampilkan jumlah sampel dari isyarat. Jika isyarat tercampur dan atau isyarat sumber memiliki perbedaan jumlah sampel maka secara otomatis, program ini akan mengambil jumlah yang lebih sedikit untuk dilakukan proses ICA dan membuang sisanya. Terdapat pula tombol **Transpose** untuk melakukan *transpose* matriks isyarat sumber dan tercampur.

➤ Bagian Fixed Point ICA

Bagian ini berhubungan dengan algoritma *FastICA*. Pada bagian ini terdapat pilihan pendekatan yakni

paralel (proses ICA tiap unit) dan simetris (proses ICA untuk beberapa unit), pilihan nonlinearitas yakni *tanh* (tangen hiperbolik) dan *gauss* (eksponensial) serta penanda kondisi program standby (warna hijau) atau sibuk (warna merah). Juga terdapat tombol **Pengaturan Lanjut** untuk mendefinisikan parameter yang dibutuhkan dalam algoritma *FastICA* sesuai keinginan, tombol **FastICA** melakukan algoritma *FastICA*, tombol **Grafik/Plot Data** untuk mengamati grafik isyarat suara dan tombol **Bunyikan** untuk mendengarkan suara masukan dan keluaran.

➤ **Bagian Keterangan**

Bagian ini berhubungan dengan tampilan informasi hasil algoritma PCA dan pemutihan serta MSE yang dihasilkan. Terdapat juga tombol **Masuk ke Workspace** untuk memasukkan peubah-peubah baru hasil algoritma *FastICA* ke dalam *Workspace* Matlab.

Selain panel-panel jendela utama juga terdapat menu-menu untuk berbagai fungsi yang diberikan oleh program. Menu-menu itu dijelaskan sebagai berikut.

➤ **Menu Suara**

Menu ini digunakan untuk menentukan isyarat-isyarat suara masukan program. Isyarat suara yang akan digunakan sebagai isyarat sumber bisa didapatkan dari *file* suara atau melalui perekaman.

➤ **Menu File**

Menu ini berhubungan dengan operasi pasca proses ICA yakni penyimpanan hasil taksiran suara, pengulangan parameter masukan dan menutup program.

➤ **Menu Bantuan**

Menu ini berisi informasi dari program.

Setelah dipilih *file-file* yang dikehendaki, langkah pertama yang perlu dilakukan adalah memberi definisi awal seluruh parameter yang dibutuhkan pada proses ICA. Seluruh parameter ini terangkum dalam jendela **Pengaturan Lanjut**. Parameter-parameter yang disediakan antara lain:

1. Epsilon (atau batas ambang error)
2. Konstanta belajar
3. Jumlah maksimum iterasi
4. Ukuran sampling
5. Interval tampilan iterasi
6. Keterangan
7. Pengkhususan penghitungan

Pada jendela **Pengaturan Lanjut** ini, disediakan pula panel untuk memasukkan matriks vektor dan nilai eigen sebagai identifikasi awal penghitungan PCA serta panel untuk memasukkan

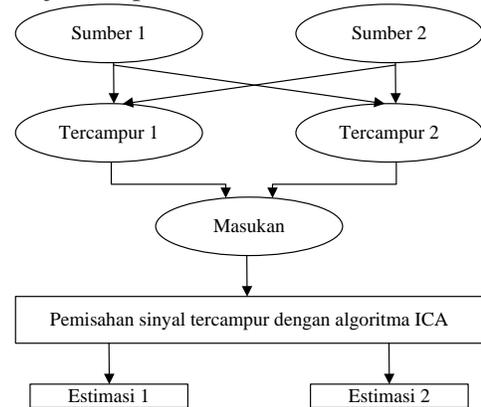
isyarat putih, matriks pemutih dan inversnya sebagai identifikasi awal penghitungan pemutihan sinyal.

Setelah seluruh isyarat masukan dan parameter peubah proses ICA selesai ditentukan, pengguna dapat melakukan proses ICA. Setelah itu, pengguna bisa melihat besar vektor dan nilai *eigen*, matriks pemutih dan *inversnya* beserta MSE pada kotak yang tersedia di panel **Keterangan**. Pengguna juga dapat memasukkan seluruh variabel hasil proses ICA ke dalam *Workspace* Matlab dengan menuliskan nama induk variabel terlebih dahulu. Lalu, pengguna juga bisa melihat tampilan grafik data seluruh isyarat suara dan mendengarkan suara yang ada dalam program ini.

3.3 Pengaturan Ruang Perekaman

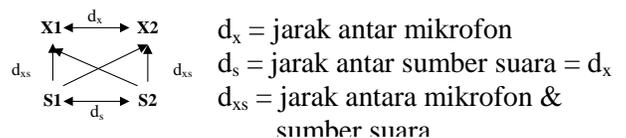
Isyarat suara diambil dengan cara melakukan perekaman suara antara sumber satu yaitu suara orang yang sedang berbicara dan sumber dua yang merupakan suara lagu atau orang lain. PC akan menyimpan hasil rekaman campuran1 (x1) yang berisi campuran suara antara sumber suara satu dan sumber suara dua. Demikian juga pada *notebook* akan menyimpan hasil rekaman campuran2 (x2).

Secara keseluruhan, blok diagram algoritma ICA ditunjukkan pada Gambar 6 berikut.



Gambar 6 Blok Diagram Algoritma ICA

Ruang yang digunakan pada saat perekaman berukuran 3m x 3m dalam kondisi tertutup rapat. Peralatan rekaman diatur pada tengah ruangan. Pada saat perekaman berlangsung, akan diatur agar sumber suara saling tercampur satu dengan yang lain. Mekanisme perekaman suara ditunjukkan pada Gambar 7.



Gambar 7 Mekanisme Perekaman Suara

Hasil rekaman x_1 dan x_2 yang tersimpan dijadikan sebagai masukan untuk proses pemisahan isyarat suara dengan algoritma *FastICA*. Rekaman suara disimpan dengan format *.wav*. Program akan mengolah masukan tersebut sehingga menghasilkan dua macam taksiran isyarat suara yang diharapkan memiliki kemiripan antara isyarat sumber dengan isyarat taksiran. Pengujian ini dilakukan dengan empat model pendekatan-nonlinearitas yakni antara lain :

- Model Paralel-*Tanh*
- Model Paralel-*Gauss*
- Model Simetris-*Tanh*
- Model Simetris-*Gauss*

Setelah didapatkan isyarat taksiran, maka kemiripannya dengan isyarat sumber dapat dicari dengan menghitung nilai MSE (*Mean Square Error*) antara isyarat taksiran dan isyarat sumber yang ditunjukkan pada dua nilai berikut:

- Nilai L-2 *distance* merupakan nilai kesalahan rata-rata kuadrat selisih nilai antara isyarat sumber dengan hasil pemisahan (taksiran).
- Nilai L-4 *distance* merupakan nilai kesalahan rata-rata pangkat empat selisih nilai antara isyarat sumber dengan hasil pemisahan (taksiran).

IV PENGUJIAN DAN ANALISIS

Hasil proses perekaman yang telah ada dijadikan sebagai masukan. Satu suara manusia (S1) dan dua suara musik (S2) menjadi komponen dasar masukan dalam pengujian ini sehingga ada dua variasi pada satu percobaan. Selain itu, pengujian ini dilakukan dengan tiga variasi jarak antar mikrofon dan dua variasi jarak antara mikrofon dan sumber suara, sehingga ada enam variabel berdasarkan jarak.

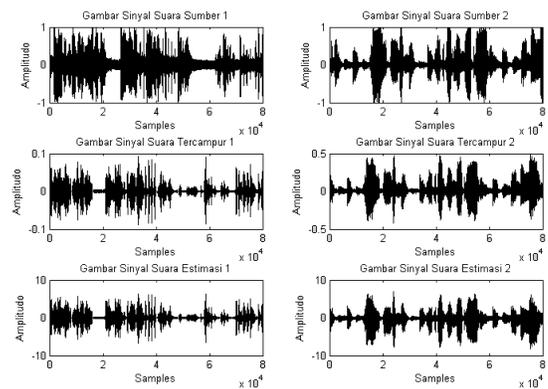
4.1 Hasil Pengujian

Untuk mempermudah penjelasan, maka pada kondisi jarak perekaman diberikan beberapa variabel:

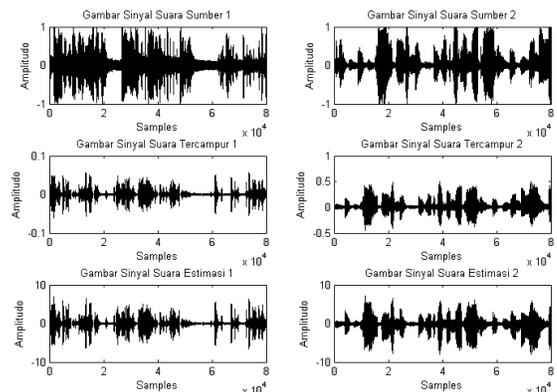
- S1 adalah sumber suara manusia, dengan 1 variabel :
 - s_{1a} adalah sumber suara manusia satu.
- S2 adalah sumber suara musik, dengan 2 variabel :
 - s_{2a} adalah sumber suara musik satu.
 - s_{2b} adalah sumber suara musik dua.
- X1 adalah mikrofon unit komputer satu yang menyimpan hasil rekaman campuran suara antara sumber satu dengan sumber dua dengan sudut pandang suara sumber satu.
- X2 adalah mikrofon unit komputer dua yang menyimpan hasil rekaman campuran suara antara sumber satu dengan sumber dua dengan sudut pandang suara sumber dua.
- Nilai MSE yang ditampilkan program antara lain:

- MSE1 adalah hasil penghitungan L-2 *distance* antara sumber satu dengan taksiran satu.
- MSE2 adalah hasil penghitungan L-4 *distance* antara sumber satu dengan taksiran satu.
- MSE3 adalah hasil penghitungan L-2 *distance* antara sumber dua dengan taksiran dua.
- MSE4 adalah hasil penghitungan L-4 *distance* antara sumber dua dengan taksiran dua.

Gambar 8 dan Gambar 9 di bawah ini menunjukkan grafik isyarat suara hasil pengujian dengan objek pengamatan s_{1a} dan s_{2b} model paralel-*tanh* dengan jarak antar mikrofon / antar sumber suara sebesar 20 cm sebagai percontohan. Gambar 8 menunjukkan jarak antara mik dan sumber suara sebesar 20 cm, sedangkan Gambar 9 menunjukkan jarak antara mik dan sumber suara sebesar 30 cm. Nilai MSE hasil pemisahan dengan jarak antar mikrofon / antar sumber suara sebesar 20 cm secara keseluruhan disajikan pada Tabel 1.



Gambar 8 Grafik Isyarat Suara Objek Pengamatan S_{1a} dan S_{2b} Model Paralel-*Tanh* dengan Jarak Antar Mikrofon / Antar Sumber suara 20 cm serta Jarak Antara mik & Sumber Suara 20 cm



Gambar 9 Grafik Isyarat Suara Objek Pengamatan S_{1a} dan S_{2b} Model Paralel-*Tanh* dengan Jarak Antar Mikrofon / Antar Sumber suara 20 cm serta Jarak Antara mik & Sumber Suara 30 cm

Tabel 1 Nilai MSE Hasil Pemisahan dengan jarak antar mikrofon / antar sumber suara 20 cm

Model Paralel-Tanh							
Jarak antara mik& sumber suara (cm)	Objek Pengamatan	MSE1	MSE2	Objek Pengamatan	MSE3	MSE4	Bentuk grafik
20	s1a_s2a	1,045	7,32	s2a_s1a	1,172	8,183	Tidak Sesuai
20	s1a_s2b	1,043	9,242	s2b_s1a	1,047	11,128	Sesuai
30	s1a_s2a	1,043	8,406	s2a_s1a	1,167	9,249	Tidak Sesuai
30	s1a_s2b	1,044	8,902	s2b_s1a	1,045	10,896	Sesuai
Model Paralel-Gauss							
Jarak antara mik& sumber suara (cm)	Objek Pengamatan	MSE1	MSE2	Objek Pengamatan	MSE3	MSE4	Bentuk grafik
20	s1a_s2a	1,046	7,31	s2a_s1a	1,162	8,077	Tidak Sesuai
20	s1a_s2b	1,048	9,319	s2b_s1a	1,042	11,144	Sesuai
30	s1a_s2a	1,05	8,485	s2a_s1a	1,167	9,333	Tidak Sesuai
30	s1a_s2b	1,047	8,94	s2b_s1a	1,042	10,891	Sesuai
Model Simetris-Tanh							
Jarak antara mik& sumber suara (cm)	Objek Pengamatan	MSE1	MSE2	Objek Pengamatan	MSE3	MSE4	Bentuk grafik
20	s1a_s2a	1,045	7,32	s2a_s1a	1,6283	6,9809	Sesuai
20	s1a_s2b	1,048	9,319	s2b_s1a	1,042	11,145	Sesuai
30	s1a_s2a	1,041	8,406	s2a_s1a	1,232	4,3614	Sesuai
30	s1a_s2b	1,047	8,94	s2b_s1a	1,045	10,896	Sesuai
Model Simetris-Gauss							
Jarak antara mik& sumber suara (cm)	Objek Pengamatan	MSE1	MSE2	Objek Pengamatan	MSE3	MSE4	Bentuk grafik
20	s1a_s2a	1,046	7,31	s2a_s1a	1,256	4,4816	Sesuai
20	s1a_s2b	1,048	9,319	s2b_s1a	1,042	11,144	Sesuai
30	s1a_s2a	1,05	8,484	s2a_s1a	1,644	7,117	Sesuai
30	s1a_s2b	1,047	8,94	s2b_s1a	1,045	10,896	Sesuai

Dari Gambar 8 dan Gambar 9, dapat dilihat bahwa kedua isyarat suara hasil taksiran sesuai (sama) dengan isyarat suara nyatanya. Tabel 1 di atas menyediakan data hanya ada dua belas grafik dari enam belas grafik isyarat suara hasil taksiran yang sesuai (sama) dengan isyarat suara nyatanya. Dua dari dua belas hasil yang sesuai ditunjukkan pada Gambar 8 dan Gambar 9.

4.2 Analisis Hasil Pengujian dan Pembahasan

Keandalan program ini dapat dilihat dengan melakukan percobaan pemisahan isyarat suara tercampur dari isyarat suara tercampur yang cukup sempurna. Isyarat suara ini diambil dari hasil koleksi milik Jaakko Särelä, Patrik Hoyer dan Ella Bingham. Nilai MSE yang dihasilkan cenderung stabil. Semua grafik isyarat suara hasil taksiran sesuai (sama) dengan isyarat suara nyata yang terekam. Tabel 2 berikut menunjukkan nilai MSE dari percobaan ini.

Tabel 2 Nilai MSE Isyarat Suara Koleksi Milik Jaakko, Patrik Hoyer dan Ella Bingham

Objek pengamatan	MSE 1	MSE 2	MSE 3	MSE 4
Campuran suara sirine dan suara berita CNN	0,830	3,560	0,672	1,896
Campuran suara sirine dan suara nyanyian	0,831	3,562	1,028	8,486
Campuran suara sirine dan suara musik	1,051	5,536	1,286	7,158
Rata-Rata MSE	0,904	4,219	0,996	5,847

Kualitas campuran suara sangat mempengaruhi proses pemisahan suara untuk menghasilkan suara yang terpisah. Apabila kualitas isyarat suara tercampur buruk maka proses pemisahan isyarat suara akan menghasilkan kualitas yang kurang baik. Nilai hasil pengujian berupa MSE dengan bersumber dari data hasil pengujian di atas dapat dilihat pada Tabel 3.

Tabel 3 Nilai MSE Berdasarkan Objek Pengamatan

Objek pengamatan	MSE 1	MSE 2	MSE 3	MSE 4
Campuran s1a dan s2a	1.046	7.2524	1.3426	6.8314
Campuran s1a dan s2b	1.0459	8.3528	1.0438	10.916
Rata-Rata MSE	1.0459	7.8026	1.1932	8.8736

Lebih kecil nilai MSE maka isyarat tersebut lebih stabil (mendekati kebenaran). Dari tabel 3, dapat dilihat bahwa rata-rata MSE 3 lebih besar

dibandingkan dengan MSE 1, begitu juga dengan MSE 4 lebih besar daripada MSE 2. Hal ini berarti hasil taksiran isyarat pertama (suara manusia) lebih akurat daripada isyarat kedua (suara lagu). Selain itu, pembesaran amplitudo isyarat lagu juga lebih tinggi daripada isyarat manusia. Di bawah ini diterangkan analisis hasil pengujian jika dilihat dari bentuk grafiknya berdasarkan beberapa variabel. Analisis bentuk grafik berdasarkan jarak antar mikrofon / antar sumber suara ditunjukkan pada Tabel 4, analisis bentuk grafik berdasarkan jarak antara mikrofon dan sumber suara pada Tabel 5 dan analisis bentuk grafik berdasarkan model pendekatan-nonlinearitas ditunjukkan pada Tabel 6.

Tabel 4 Analisis Bentuk Grafik Berdasarkan Jarak Antar Mikrofon / antar Sumber suara

Jarak antar mikrofon/ antar sumber suara	20 cm	30 cm	40cm
Jumlah grafik berbentuk baik	12	12	14

Tabel 5 Analisis Bentuk Grafik Berdasarkan Jarak Antara Mikrofon dan Sumber suara

Jarak antar mikrofon dan sumber suara	20 cm	30 cm
Jumlah grafik berbentuk baik	18	20

Tabel 6 Analisis Bentuk Grafik Berdasarkan Model Pendekatan-Nonlinearitas

Jarak antar mikrofon/antar sumber suara	Paralel- <i>Tanh</i>	Paralel- <i>Gauss</i>	Simetris- <i>Tanh</i>	Simetris- <i>Gauss</i>
Jumlah grafik berbentuk baik	7	7	12	12

Percobaan paling efektif dilakukan dengan jarak antar mikrofon atau antar sumber suara sebesar 40 cm dan jarak antara mikrofon dan sumber suara sejauh 20 cm. Kualitas campuran isyarat suara sangat ditentukan oleh pengaruh jarak penempatan mikrofon dari sumber suara. Semakin jauh jarak antar mikrofon dan atau antar sumber suara, maka kemungkinan perpaduan (*interferensi*) isyarat suara dari sumber suara yang berlainan akan semakin berkurang, sehingga kualitas isyarat suara yang dihasilkan akan semakin baik. Model pendekatan simetris pada penghitungan ICA lebih baik jika dibandingkan dengan pendekatan paralel.

Pada saat proses perekaman dilakukan dengan jarak tertentu maka akan dihasilkan campuran suara yang berbeda-beda. Jarak yang diberikan pada saat proses perekaman akan mempengaruhi :

- Kualitas campuran isyarat suara.
- Volume (amplitudo) suara.
- Proses keberhasilan pemisahan suara dari isyarat tercampur.

Sehingga, jika campuran suara yang diberikan kualitas suaranya baik, maka pada saat dipisahkan akan menghasilkan kualitas suara yang baik pula. Begitu juga sebaliknya, jika kualitas campuran suara buruk, maka pada saat dipisahkan akan menghasilkan kualitas suara yang buruk pula.

V. PENUTUP

5.1 Kesimpulan

Dari tugas akhir ini, diperoleh kesimpulan sebagai berikut :

- 1 Pengujian dengan masukan berupa suara rekaman mandiri menghasilkan rata-rata MSE suara 1 senilai 1,0459 dan MSE suara 2 senilai 1,1932 serta 38 grafik yang baik dari 48 grafik seluruhnya, sedangkan dengan masukan berupa suara koleksi Jaakko Särelä, Patrik Hoyer dan Ella Bingham (kondisi ideal) menghasilkan rata-rata MSE suara 1 senilai 0,904 dan MSE suara 2 senilai 0,996 serta 12 grafik yang baik dari 12 grafik seluruhnya. Nilai MSE yang mendekati nilai varians data yakni sebesar satu menunjukkan kualitas suara yang dihasilkan semakin baik. Hal ini menunjukkan bahwa metode *FastICA* memerlukan masukan campuran isyarat yang baik sehingga dapat menghasilkan kualitas pemisahan isyarat suara yang baik pula.
- 2 Algoritma *FastICA* merupakan metode komputasi cepat yang cukup efisien untuk menghasilkan taksiran isyarat yang mendekati kebenaran. Pengujian dengan masukan berupa suara rekaman mandiri pada penelitian ini menunjukkan hasil ada 38 grafik isyarat suara taksiran yang baik dari 48 grafik hasil pengujian seluruhnya, sehingga efisiensi algoritma *FastICA* sebesar 80,2 %.
- 3 Grafik berbentuk baik dari model pendekatan simetris berjumlah 24, sedangkan dari model pendekatan paralel berjumlah 14, sehingga model pendekatan penghitungan secara simetris merupakan model paling efektif pada algoritma *FastICA*. Model nonlinearitas *tanh* memiliki efektivitas yang sama dengan *gauss* karena grafik berbentuk baik dari kedua model nonlinearitas tersebut (*tanh* dan *gauss*) sama-sama berjumlah 19.
- 4 Hasil analisis grafik dengan jarak antar mikrofon atau antar sumber suara sejauh 20 dan 30 cm menunjukkan grafik berbentuk baik sebanyak 12 grafik, sedangkan dengan jarak antar mikrofon atau

antar sumber suara sejauh 40 cm menunjukkan grafik berbentuk baik sebanyak 14 grafik. Semakin jauh jarak antar mikrofon atau antar sumber suara, semakin baik kualitas isyarat suara yang dihasilkan karena kemungkinan perpaduan (*interferensi*) isyarat-isyarat suara dari sumber yang berbeda-beda akan semakin kecil.

5.2 Saran

Dari tugas akhir ini, penulis memberi beberapa saran untuk pengembangan lebih lanjut ke depan seiring semakin meningkatnya IPTEK.

- 1 Algoritma ini perlu dicoba untuk digunakan pada aplikasi selain pengolahan suara, seperti: pengolahan isyarat biomedis, pengolahan citra, telekomunikasi, ilmu ekonomi, stoikiometri, dan lain-lain.
- 2 Perlu dilakukan analisis pengolahan suara tercampur dengan metode lainnya untuk dibandingkan dengan metode *FastICA*.
- 3 Algoritma ini bisa dikembangkan pula dalam hal pemisahan isyarat suara yang tercampur dengan menghilangkan derau secara langsung pada saat menghasilkan isyarat taksiran, selain itu dapat pula ditambah kemampuan memisahkan isyarat suara tercampur secara *online*.

DAFTAR PUSTAKA

- [1] Hyvarinen, Aapo., dan Oja, Erkki., *Independent Component Analysis: Algorithms and Applications*, Helsinki University of Technology, Finlandia, 2000.
- [2] Paulus, Erick., dan Nataliani, Yessica., "Cepat Mahir GUI Matlab", Andi Yogyakarta, 2007.
- [3] Arhami, Muhammad., dan Desiani, Anita., "Pemrograman Matlab", Andi Yogyakarta, 2005.
- [4] Sahid, Drs. MSc., "Panduan Praktis Matlab Disertai Latihan Langsung", Andi Yogyakarta, 2006.
- [5] Stroud, K. A., dan Sucipto, Erwin., "Matematika Untuk Teknik", Erlangga Jakarta, 1996.
- [6] Jek, Jong. S., "Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab", Andi Yogyakarta, 2005.
- [7] Hermawan, Arief, "Jaringan Syaraf Tiruan Teori dan Aplikasi", Andi Yogyakarta, 2006.

- [8] Soesianto, F. Ir. BScE. PhD., "Panduan Mata Kuliah Asas Sistem Teknik & Asas Pemecahan Masalah Program Pra Pasca Sarjana Teknik Elektro UGM", Universitas Gadjah Mada Yogyakarta, 2002.
- [9] Syarif, "Forecasting Trafik PDN Menggunakan Metode Jaringan Saraf Tiruan", Laporan Kerja Praktek Teknik Elektro Universitas Diponegoro, 2007.
- [10] -----, Materi berisi isyarat suara koleksi Jaakko Särelä, Patrik Hoyer dan Ella Bingham, http://www.cis.hut.fi/projects/ica/cocktail/cocktail_en.cgi, April 2005.

BIOGRAFI



Syarif (L2F 004 516)

Lahir di Kendal, menyelesaikan pendidikan dasar sampai SMA di kota kelahirannya. Saat ini masih tercatat sebagai mahasiswa Teknik Elektro, Universitas Diponegoro, Semarang angkatan 2004, dengan konsentrasi Elektronika Telekomunikasi.

Mengetahui dan Mengesahkan,

Dosen Pembimbing I

Dosen Pembimbing II

Achmad H, ST., MT.

NIP. 132.137.933

Ajub Ajulian Z, ST., MT.

NIP. 132 205 684

Tanggal : _____

Tanggal : _____