UNIVERSITAS INDONESIA

PENGENDALI MINIATUR ROBOT MOBIL BERBASIS SUARA MANUSIA

SKRIPSI

DANY ARSIDAD
0606039745

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM SARJANA EKSTENSI FISIKA INSTRUMENTASI
DEPOK
JUNI 2009

UNIVERSITAS INDONESIA

PENGENDALI MINIATUR ROBOT MOBIL BERBASIS SUARA MANUSIA

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

DANY ARSIDAD
0606039745

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI SARJANA EKSTENSI FISIKA INSTRUMENTASI
DEPOK
JUNI 2009

HALAMAN PERNYATAAN ORISINALITAS

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

Nama	**: Dany Arsidad**
NPM	**: 0606039745**
Tanda Tangan	**:**
Tanggal	**:**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh	:
Nama	: Dany Arsidad
NPM	: 0606039745
Program Studi	: Fisika Instrumentasi Elektronika
Judul Skripsi	: Pengendali Miniatur Robot Mobil Berbasis Suara Manusia

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Fisika Instrumentasi Elektronika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr.Prawito ( )

Penguji I : Dr.Santoso ( )

Penguji II : Dr.rer.nat. Martarizal ( )

Ditetapkan di :

Tanggal :

KATA PENGANTAR

Puji sukur kehadirat Allah SWT, yang telah memberikan segala kenikmatan dan anugrah kesehatan, Nabi Muhammad SAW junjungan sekaligus inspirasi kita dalam setiap pijakan kaki sehingga penulis dapat menyelesaikan tugas akhir ini sesuai rencana.Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Scient Jurusan Fisika pada Fakultas Matematika dan Ilmu pengetahuan Alam Universitas

Telah banyak tenaga dan pikiran yang telah penulis curahkan dalam menyelesaikan tugas akhir ini.Banyak kesulitan teknis maupun non teknis yang ditemui selama pengerjaan, tanpa bantuan dari orang-orang yang perduli, maka penulis tidak akan mampu menyelesaikan tugas akhir ini. Pada kesempatan ini penulis ucapkan terimakasih setulus-tulusnya kepada Yang Ter-Hormat :

Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya dan Nabi Muhammad SAW sebagai junjungan kita dalam menentukan semua tindakan.

1. Kedua Orang tuaku yang tercinta, Bapak Kasnadi (Alm) dan Ibu Maslimah, dan semua kakak-kakakku beserta keluarga besar tercinta yang telah memberi dukungan baik dalam bentuk spiritual dan materi yang telah diberikan selama ini.
2. Dr.Prawito selaku dosen pembimbing yang telah memberikan petunjuk, kemudahan dalam berpikir dan bimbingan dalam penyelesaian tugas akhir ini.
3. Dr. Syamsu Rosid, selaku Ketua Program Ekstensi Fisika.
4. Temen-temen teknik elektro UI dan ITS surabaya yang dengan sabar memberi petunjuk dalam memecahkan problem-problem yang sedang dikerjakan penulis.
5. Seluruh rekan-rekan Ekstensi Fisika angkatan 2007.
6. Seluruh rekan-rekan Instrumentasi, khususnya angkatan 2002.
7. Teman-teman Pecinta Alam NNKPG Club yang selalu memberikan ide-ide segar dalam berpijak dibumi ini.
8. Seluruh tim secretariat Fisika UI yang telah membantu dalam masalah administrasi.

9. Seluruh keluarga besar FMIPA UI.
10. Semua pihak yang secara tidak langsung terlibat dalam pembuatan skripsi ini dan tidak mungkin dapat disebutkan satu persatu, semoga amal baik yang telah dilakukan senantiasa dibalas oleh Allah SWT.

Semoga Allah SWT melimpahkan segala rahmat dan karunia-Nya atas kebaikan Bapak / Ibu dan Saudara/i sekalian.

Semoga penulisan ilmiah ini benar-benar dapat memberikan kontribusi positif dan menimbulkan sikap kritis kepada para pembaca khususnya dan masyarakat pada umumnya untuk senantiasa terus memperoleh wawasan dan ilmu pengetahuan di bidang teknologi.

Menyadari keterbatasan pengalaman dan kemampuan yang dimiliki saya, sudah tentu terdapat kekurangan serta kemungkinan jauh dari sempurna, untuk itu saya tidak menutup diri dan mengharapkan adanya saran serta kritik dari berbagai pihak yang sifatnya membangun guna menyempurnakan penulisan ilmiah ini.

Akhir kata semoga penulisan ilmiah ini dapat memberikan manfaat bagi semua pihak yang bersangkutan, khususnya bagi saya dan umumnya bagi para pembaca.

Depok, Juni 2009

Dany Arsidad

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini:

Nama : Dany Arsidad
NPM : 0606039745
Program Studi : Fisika Instrumentasi Elektronika
Departemen : Fisika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Free Right*)** atas karya ilmiah saya yang berjudul ”Pengendali Miniatur Robot Mobil Berbasis Suara Manusia” beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : Juni 2009
Yang menyatakan

(…….)

ABSTRAK

Nama : Dany Arsidad
Program Studi : Ekstensi Fisika Instrumentasi Elektronika
Judul : Pengendali Miniatur Robot Mobil Berbasis suara Manusia

Pengenalan suara mempresentasikan teknik pengolahan signal digital yang digunakan untuk berbagai aplikasi. Seiring berkembangnya teknologi dari waktu ke waktu, maka timbulah ide untuk mengembangkan proses pengolahan sinyal wicara menjadi suatu aplikasi untuk mengendalikan pergerakan miniatur robot mobil dengan perintah suara manusia. Proses pengenalan wicara ini menggunakan metode *HMM* yang diintegrasikan dengan perintah-perintah program yang terdapat pada Matlab. Dalam mekanisme kerja yang dilakukan, semua inputan suara dari *spekear verivication* akan diproses dengan berbagi metode untuk mengurangi nilai error yang disebabkan oleh faktor luar. Pada proyek akhir ini terdapat dua fase yaitu fase pembelajaran dan fase pengujian. Fase pembelajaran mempelajari tentang ekstrasi suara manusia yang akan diproses lebih lanjut agar mudah dikenali dalam fase pengujian. Sedangkan metode untuk pengenalan suara manusia menggunakan dependent speaker dimana metode yang digunakan hanya untuk mengenali karakter suara satu orang saja.

Kata kunci :
Speech Recognition, HMM, Suara Manusia

ABSTRACT

Name : Dany Arsidad
Study Program : Ekstensi Fisika Instrumentasi Elektronika
Title : Voice-Based Miniatur Car Controller

Voice recognition represented the techniques of digital signal conditioning which can be used in many everyday appliances. As the growth of technologies emphasized an idea to develop voice signal conditioning process to become a controller of miniature mobile robot using human voices. This processes are involving HMM methods that integrated with Matlab programs.

In addition, all of the voices input from speaker verification will be processed with various methods to decrease the error value from the environment.

This final assignment include 2 phases that support the process, the first phase, studying phase was used to learn about human voice extraction in the other phase. Furthermore, the voice recognition methods were only using speaker dependent, which can be obtained for personal voice character.

Key words:
Speech Recognition, Hidden Markov Model, Voice Recognition

DAFTAR ISI

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Komunikasi antara manusia dan mesin dengan menggunakan media suara telah menjadi kecenderungan bagi dunia modern. Kemudahan dalam menggunakan ucapan suara menjadi alasan utama diciptakan dan dikembangkannya suatu kendali suara manusia untuk mengendalikan dan mengontrol berbagai peralatan yang telah ada ataupun sedang dalam pengembangan. Suatu sistem pengenalan ucapan dalam bahasa Inggris telah dikembangkan oleh Claudio Becchetti, seorang profesor ahli dalam bidang teori dan implementasi C dengan bekerjasama dengan Lucio Prina Ricotti, seorang ahli *digital signal processing* dari Roma yang memperkenalkan konsep *Hidden Markov Model (HMM).* Sistem pengenalan ucapan yang mereka kembangkan merupakan sistem yang sangat rumit seperti halnya sistem pengenalan ucapan pada umumnya. Tetapi sistem itu sangat moduler, sehingga berpotensi untuk dikembangkan.

Pengembangan sistem pengenalan ucapan kedua ahli itu memerlukan pemahaman yang cukup tentang cara kerja sistem dan data-data yang diperlukan dalam menjalankannya. Tanpa suatu alat bantu, pemahaman tersebut memakan waktu yang cukup lama dan pada akhirnya memperlambat pengembangan sistem itu. Suatu pendokumentasian atas sistem pengenalan ucapan yang telah ada, diperlukan sebagai sarana untuk memangkas waktu yang diperlukan oleh para developper untuk mempelajari sistem pengenalan ucapan.

Dengan dikembangkannya speech recognition oleh para ahli untuk berbagai aplikasi sehari-hari diharapkan akan mempermudah dalam pengoperasian suatu sistem alat sehingga efisiensi waktu, tenaga ataupun masalah keselamatan akan tetap terlindungi tanpa mengurangi kualitas pada hasil nilai yang diinginkan. Dengan dasar pemikiran diatas , maka penulis akan mencoba membuat suatu aplikasi speech recognition untuk mengontrol pergerakan robot mobil dengan metode-metode yang telah dikembangkan sebelumnya.

1.2 Tujuan Penelitian

Tujuan dari skripsi ini adalah untuk merancang suatu sistem akses kontrol pada miniatur robot mobil dengan menggunakan suara manusia melalui mikropon yang dioalah oleh PC dengan sistem koneksinya menggunakan bluetooth, sehingga kita dapat mengendalikan robot mobil hanya dengan perintah suara manusia.

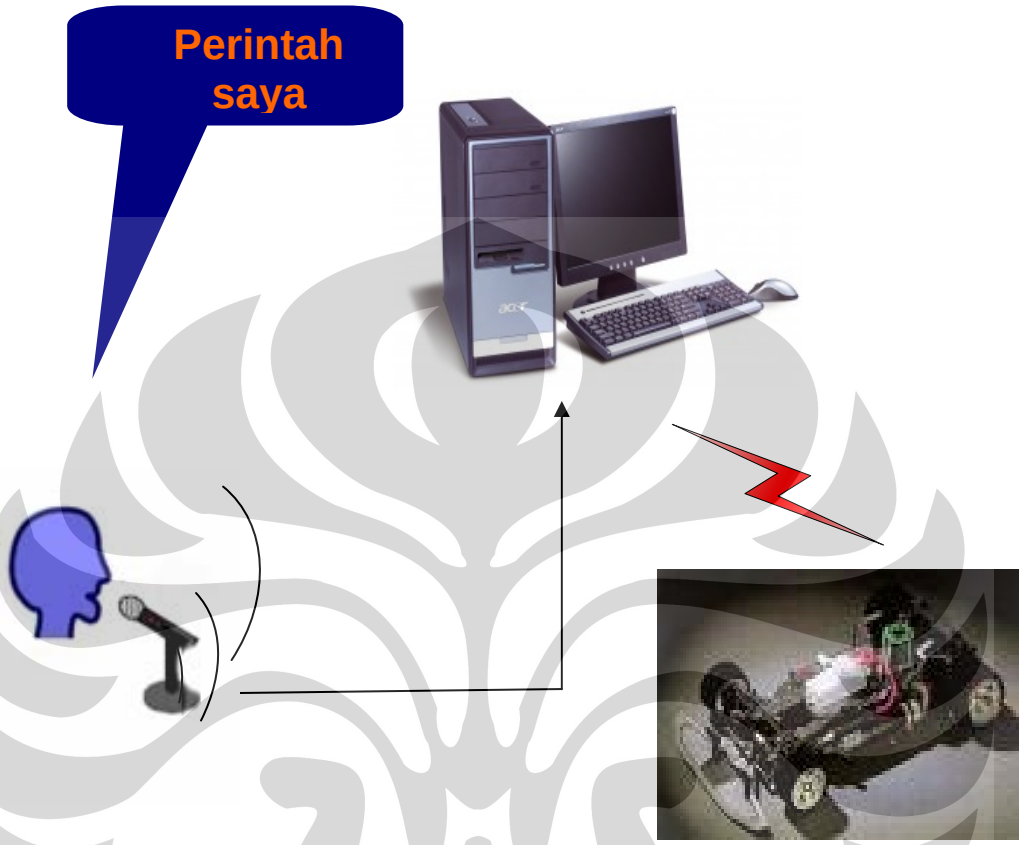
1.3 Batasan Masalah

- Dalam skripsi ini permasalahan yang dibahas hanya sebatas pembuatan dan pengenalan suara manusia yang telah ditentukan untuk mengendalikan gerakan robot mobil dengan ucapan suara (Kiri,Kanan,Depan,Belakang dan Stop). Sedangkan sistem koneksi yang digunakan untuk menggerakkan dari PC ke robot mobil menggunakan sistem Wireless (Bluetooth)
- Perancangan dan pembuatan sistem hanya sebatas pada software dan Hardware dengan berbagai metode yang dimplementasikan pada bahasa Matlab sehingga diperoleh output data yang diinginkan

1.4 Deskripsi Singkat

Pada prinsipnya alat ini berfungsi untuk menggerakkan sebuah miniatur mobil dengan menggunakan suara manusia yang sebelumnya telah disimpan dalam database komputer. Pengujian alat dimulai dengan menyebutkan kata kunci melalui sebuah microphone yang akan dicocokkan dengan data yang ada di dalam database. Apabila input yang diberikan cocok dengan data yang ada, maka perintah akan diteruskan kepada mikrokontroler yang terdapat pada alat peraga melalui bluetooth. Kemudian mikrokontroler akan menggerakkan keempat motor listrik yang terdapat pada alat peraga sesuai dengan perintah yang telah diberikan.

Adapun sistem kerja secara sederhana dapat dilihat pada blok diagram berikut:



Gambar 1.1. Sistem kerja pengendali miniatur robot mobil berbasis suara manusia

Sistem perancangan alat ini menggunakan perangkat keras meliputi beberapa bagian yaitu: bagian pengontrol utama yaitu Mikrokontroler sebagai otak pengendali, sinyal masukan yaitu berupa perintah suara manusia dan rangkaian penggerak yaitu pengendalian motor DC sebagai plant yang diproses untuk menggerakan model kendaraan sesuai dengan perintah yang diberikan oleh suara manusia dengan media Bluetooth sebagai perantara transfer datanya. Dengan mengintegrasikan sistem secara keseluruhan pada alat yang dibuat dengan beberapa metode diharapkan dapat berjalan sesuai dengan apa yang direncanakan. Sehingga performance dari system alat diatas akan menjadi sempurna dalam proses pengujian akhir dengan kendali suara manusia yang telah ditentukan.

1.5 Metode Penulisan

1. *Study Literatur*

Penulis menggunakan metode ini untuk memperoleh informasi yang berkaitan dengan penelitian yang dilakukan dengan mengacu kepada buku-buku pegangan, data *sheet*, internet, makalah-makalah dan lain-lain.

2. Perancangan Alat

Penulis berusaha untuk membuat suatu rancangan sistem mekanik serta *hardware* yang ingin dibuat di dalam penelitian, berdasarkan bahan-bahan yang ada untuk dapat dianalisa kembali.

3. Pembuatan Alat

Pembuatan alat dari rancangan yang telah diperoleh pada tahap sebelumnya, sesuai dengan tujuan yang diinginkan.

4. Pengujian Sistem

Melakukan pengujian terhadap sistem rangkaian secara keseluruhan, diharapkan dengan dilakukan proses tersebut akan menyempurnakan hasil yang diinginkan.

5. Pembuatan Program

Membuat *software* yang menggunakan *Matlab* dengan menggunakan metode *Hidden Markov Model*

6. Pengambilan Data

Setelah alat diuji secara keseluruhan sebagai suatu sistem sehingga dapat dilihat apakah sistem dapat bekerja dengan baik dan benar, sehingga penulis dapat melakukan pengambilan data.

7. Penulisan Penelitian

Dari hasil pengujian dan pengambilan data kemudian dilakukan suatu analisa sehingga dapat diambil suatu kesimpulan. Dengan adanya beberapa saran juga dapat kita ajukan sebagai bahan perbaikan untuk penelitian lebih lanjut.

BAB 2
TEORI DASAR

2.1 Suara Manusia

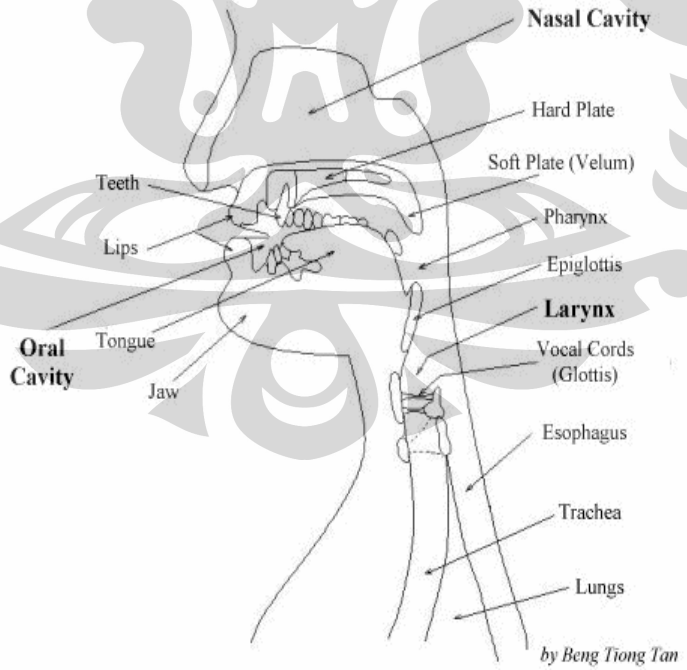
Suara adalah suatu sinyal yang sangat dipengaruhi oleh frekuensi dan merupakan bentuk sinyal diskrit yang sangat di pengaruhi oleh waktu. Demikian juga suara manusia dari mikropon mempunyai amplitudo dan panjang sinyal tertentu [3]. Proses terjadinya suara sendiri secara sederhana dapat di gambarkan sebagai berikut, *Vocal tract* yang berbentuk tabung resonansi dalam sistem speech production memiliki bagian utama *pharynx, nasal cavity* dan *oral cavity* [3]. Bentuk pada vocal tract memiliki variasi berdasarkan *soft plate (velum), tongues, lips* dan *jaw* yang secara keseluruhan disebut sebagai *articulators*. Proses pembentukan vocal tract untuk menghasilkan bentuk speech berbeda-beda disebut *articulation.* Setiap sinyal suara dapat dirubah menjadi suatu matrik dengan baris dan kolom tertentu sesuai dengan amplitudo dan panjang sinyal suara tersebut.

Pada umumnya suara seseorang sangat sulit dikenali walaupun orang tersebut sudah sangat dekat dengan kita. Hal ini disebabkan oleh suara manusia yang sangat variatif. Sebagai salah satu contoh ketika kita berbicara dengan seseorang di suatu telepon dan orang tersebut sedang mengalami gangguan pada tenggorokannya. Bila orang tersebut tidak menyebutkan siapa namanya maka akan sulit bagi kita untuk mengenal orang tersebut dengan reaksi yang cepat.

2.1.1 Proses Terjadinya Suara Manusia

Untuk membuat suatu pengenalan suara manusia di perlukan sedikit pemahaman bagaimana urutan proses terjadinya suara pada manusia, sebagai berikut, udara mengalir dari lungs (paru-paru) bergerak menuju trachea, sebuah tabung tersusun dari cincin cartilage, dan melalui larynx menuju vocal tract. Dalam hal ini *larynx* beraksi sebagai gate (pintu gerbang) antara lungs dan mouth (mulut). Ini tersusun atas *epiglottis, vocal cords* dan *false vocal cords* [3]. Ketiganya menutup saat menelan makanan sehingga paru-paru tidak kemasukan dan membuka kembali saat mengambil nafas normal. Fonem dalam bahasa

Inggris diklasifikasikan dalam terminologi *manner of articulation* dan *place of articulation.* Manner of articulation dikonsentrasikan pada aliran udara, maksudnya dalam hal ini adalah masalah lintasan dan tingkatan yang terjadi pada vokal yang dilewatkan. Manner of articulation dan voicing membagi fonem manjadi tiga kelas besar. Fonem yang memproduksi employ voicing dan semata-mata merangsang vocal tract pada glottis disebut *sonorants* (vowels, diphthongs, glides, liquids, dan nasals). Mereka memiliki sifat continuous, intense, dan periodic phonemes. *Voiced* sounds dihasilkan oleh tekanan pada udara yang mengalir melalui vocal cords sementara vocal cords ditekan untuk membuka dan menutup secara cepat untuk menghasilkan sederetan puffs periodic yang memiliki *fundamental frequency* (harmonisasi ke1) sama seperti frekuensi vibrasi vocal cord. Frekuensi vocal cord tergantung pada tingkat kepejalan, tension dan panjang vocal cords dan efek aliran udara yang dihasilkan dalam glottis, sebuah ruang diantara vocal cord – vocal cord. Komponen frekuensi ini tersusun dari sejumlah harmonisa dari frekuensi fundamental. Suatu sound yang dihasilkan tanpa vibration dalam vocal cord disebut *unvoiced* [3].



Gambar 2.1 Rongga mulut manusia

Sistem pengenalan ucapan manusia yang dilakukan oleh manusia adalah sangat baik, karena manusia memiliki *intelligent* dan *speech production* yang sudah baik. Pengenalan warna suara, frekuensi suara dari berbagai macam suara manusia yang dikenal direkam dan dikelompokkan dengan suatu hubungan asosiatif yang akurat, sehingga manusia tak pernah salah mengenal siapa yang punya suara, suara yang diucapkan dan dengan intonasinya atau yang disebut dengan amplitudo suara.Tool yang dirancang akan mengikuti pola pengenalan ucapan manusia dengan frekuensi suara, amplitudo suara dan algorithma *Hidden Markov Model (HMM)* yang merupakan urutan-urutan ucapan manusia dalam base data yang sudah ada di *mangement file-file* suara.

Alat ucap manusia mempunyai fungsi-fungsi dalam menghasilkan suatu ucapan yang dapat didengar dan dipahami oleh sesama manusia, seperti yang terlihat pada *gambar 2.1* yang terdiri dari paru-paru sebagai sumber tekanan udara *vocal cord* sebagai penentu huruf-huruf yang diucapkan, *vocal fold* sebagai penentu nada suara. Bahan baku bicara, yang disebut sumber suara, adalah suara yang terjadi akibat aliran udara menerjang pita suara. Nada kompleks ini terjadi atas frekuensi dasar (ditentukan oleh frekuensi getaran pita suara) dan sejumlah besar harmoni bernada tinggi yang disebut oveilone [3].

Saluran suara seperti laring, faring, hidung, dan mulut, merupakan resonator. Dengan kata lain, masing-masing melewatkan frekuensi tertentu secara paling baik. Empat atau lima frekuensi yang paling penting disebut forman. Semakin dekat suara yang dihasilkan pita suara dengan frekuensi forman, ia akan semakin keras terdengarnya. Suara terjadi apabila aliran udara dan paru secara periodic terputus-putus karena getaran pita suara yang dihasilkan sumber bunyi memiliki spectrum yang tersusun dari banyak harmoni yang disebut juga overtone. Ketika sumber bunyi bergerak melalui saluran suara, masing-masing harmoni berkurang proporsinya, tergantung jaraknya dan frekuensi forman. Frekuensi forman muncul berupa puncak-puncak spectrum suara yang keluar dari mulut.

2.1.2 Prinsip Dasar Speech Recognition

Semua metode dasar proses pengenalan suara terdiri dari dua fase operasi, pertama adalah proses *training*, pada proses ini sistem belajar dari referensi pola yang berupa perbedaan pola sinyal suara misal frase, kata, fonem yang akan mengisi vocabulari dari sistem. Setiap referensi di pelajari dari kata yang dikatakan yang kemudian disimpan dalam template dan telah mengalami metode untuk merata-rata dan karakteristik statistik dan parameter statistik. Yang kedua adalah proses *recognition* yaitu pada proses ini sistem akan diberikan inputan yang belum diketahui dan akan di identifikasi berdasarkan pola template yang telah didapatkan pada proses training. Pada umumnya suatu sistem pengenalan suara terdiri dari beberapa modul utama yaitu:

Signal processign frontend, accoustic modelling, language modelling, pada modul *signal processing frontend* digunakan untuk mengkonversi sinyal suara kedalam bentuk *sequence feature vector* yang akan digunakan pada saat klasifikasi. Kemudian *accoustic modeling* digunakan untuk memodelkan secara statistik hasil training yang telah dilakukan kedalam sebuah template, *language modelling* digunakan untuk memodelkan bentuk kata baik berupa kata, fonem, ataupun kalimat [13].

2.1.2.1 Tipe-Tipe Spech Recognition

Speech recognition dapat diklasifikasikan menjadi dua jenis, yaitu text independent speech recognition dan text dependent speech recognition.

2.1.2.2 Text Independent Speech Recognition

Sistem dapat mengenali suara atau kata dari banyak orang (tidak tergantung pada orang yang mengucapkannya). Untuk melakukan hal ini, sistem mengumpulkan berbagai macam suara orang dalam mengucapkan suatu kata atau frasa tertentu. Sehingga akan diperoleh referensi yang berlaku umum untuk semua pembicara. Referensi yang digunakan tidak tergantung pada jumlah orang yang dapat menggunakan sistem tersebut. Jika referensi pembicara untuk masing-

masing kata semakin banyak, maka sistem semakin baik dalam mengenali suara yang masuk.

2.1.2.3 Text Dependent Speech Recognition

Proses ini diterapkan untuk pembicara tunggal. Sistem mengenali suara yang diucapkan oleh orang tertentu. Pada tipe ini suara dari orang tertentu disimpan sebagai referensi yang menjadi acuan dalam proses pengenalan. Referensi bergantung pada jumlah pembicara yang dikenali oleh sistem.Proses ini cocok untuk mengenali orang yang berbicara dari suatu sinyal suara. Sistem lebih murah, lebih akurat, dan lebih mudah. Akan tetapi sistem tidak sefleksibel sistem *text independent* karena sulit beradaptasi dengan karakteristik pembicara baru.

Pengenalan identitas pembicara didasarkan pada pengucapan satu atau lebih frase yang spesifik, misalnya pengucapan huruf atau kata yang menjadi *password.* Jika dilihat dari klasifikasi *speech recognition*, sistem pengenalan suara manusia termasuk dalam sistem *text dependent* karena hanya dapat mengenali suara tertentu.

2.1.2.4 Speech Modeling

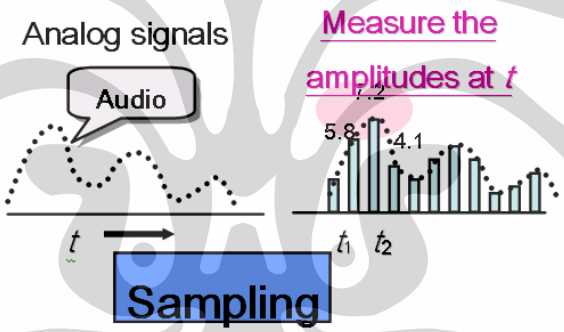
Sebuah ucapan mungkin merepresentasikan urutan vektor-vektor. Pengucapan dari orang yang sama, pada waktu yang berbeda akan menghasilkan urutan vektor-vektor yang berbeda. Tujuan pemodelan suara yaitu untuk membuat model dari variasi yang telah didapat dalam bentuk suara, dalam bentuk yang telah terekstraki. Terdapat dua pemodelan yang banyak digunakan pada *speech verification* dan *speech recognition;* yaitu model stochastic dan model template.

Model stochastic akan membuat proses *speech production* sebagai proses parametrik *random* dan menganggap parameter dari dasar proses stochastic tersebut dapat diperkirakan secara tepat. Model template memodelkan proses *speech production* dalam bentuk tanpa parameter dengan menjaga urutan vektor-vektor yang berasal dari berbagai pengucapan pada huruf yang sama dengan orang yang sama.

Model template mendominasi pengerjaan awal di *speech verification* dan *speech recognition* karena model template secara intuisi cenderung lebih baik. Pengerjaan dengan menggunakan model stochastic menunjukkan bahwa model ini lebih fleksibel dan oleh karena itu memberikan model yang lebih baik pada proses *speech production*. Model stochastic yang banyak digunakan untuk pemodelan adalah *Hidden Markov Model* (HMM).

2.2 Sampling

Sampling adalah suatu proses untuk membagi-bagi suatu sinyal kontinu dalam interval waktu yang telah ditentukan. *Sampling* dapat digunakan untuk membagi-bagi sinyal analog dalam fungsi waktu. Pada gambar 2.2 diperlihatkan bahwa sinyal analog akan dibagi menjadi waktu *sampling t* [10].



Gambar 2.2 Proses Sampling

Parameter-parameter yang menentukan hasil *sampling* adalah panjang interval yang digunakan. Frekuensi *sampling* yang digunakan mengikuti teorema Nyquist, bahwa frekuensi *sampling* harus dua kali frekuensi tertinggi dari sinyal *input* [10]. Kriteria Nyquist menyatakan :

$$f_s \geq 2xf_h \quad \text{.....(2.1)}$$

Dimana : f_s adalah frekuensi *sampling*

f_h adalah frekuensi tertinggi sinyal *input*

Suara manusia memiliki frekuensi dari 300 Hz – 3400 Hz. *Sampling* yang digunakan adalah 8000 Hz karena frekuensi tertinggi adalah 4000 Hz.

2.3 Ekstrasi

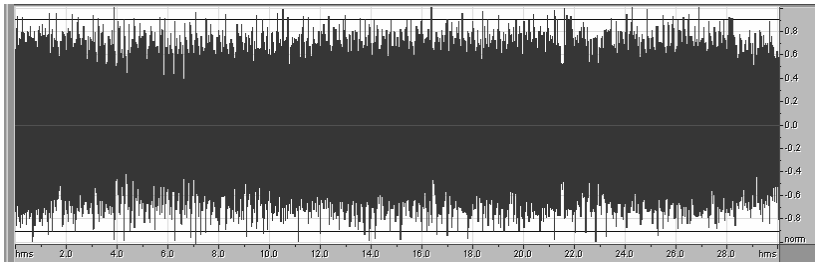
Sinyal *input* yang akan diproses secara digital terlebih dahulu di ekstraksi untuk mendapatkan suatu sinyal yang dapat dianalisa selanjutnya. Proses ekstraksi yang dilakukan adalah *pre-emphasis*, *frame blocking, windowing, Discrete Fourier Transform/Fast Fourier Transform* dan *Mel Frequency Cepstrum Coefficients* (MFCC) [9].

2.3.1 *Pre-emphasis*

Proses ini bertujuan untuk menghilangkan *background noise* yang menyertai *input* dari sinyal suara. Pada skripsi ini digunakan orde pertama dari *high pass Butterworth filter* dengan frekuensi *cut off* 200Hz dan frekuensi *sampling* 8000 Hz. Dengan frekuensi *cut off* 200 Hz, maka akan menghilangkan suara-suara yang terdapat pada frekuensi dibawah 200 Hz. Dengan menggunakan *high pass Butterworth filter*, maka diperoleh respon frekuensi yang halus. Filter ini akan memberikan respon frase yang minimum dan respon amplitudo datar (*flat*) yang maksimum.

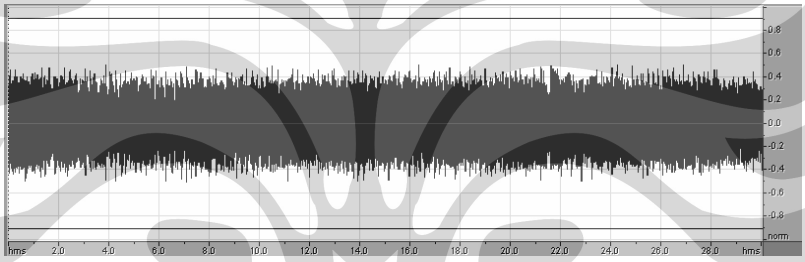
Proses normalisasi adalah proses agar didapatkan sinyal dengan ukuran level sinyal yang sesuai. Proses normalisasi akan mengurangi level sinyal yang terlalu berlebihan. Sinyal dengan level yang berlebihan akan menyebabkan parameter koefisien yang didapatkan tidak sesuai dengan yang diharapkan. Hal ini seperti terlalu besar atau terlalu kecil. Sehingga pada proses pengenalan akan menjadi lebih sulit.

Pada *Gambar 2.3* dapat dilihat bahwa sinyal *input* mempunyai level suara yang terlalu besar. Sinyal dinormalisasi sehingga *output* yang dihasilkan lebih memenuhi syarat [4].



Gambar 2.3 Sinyal wicara

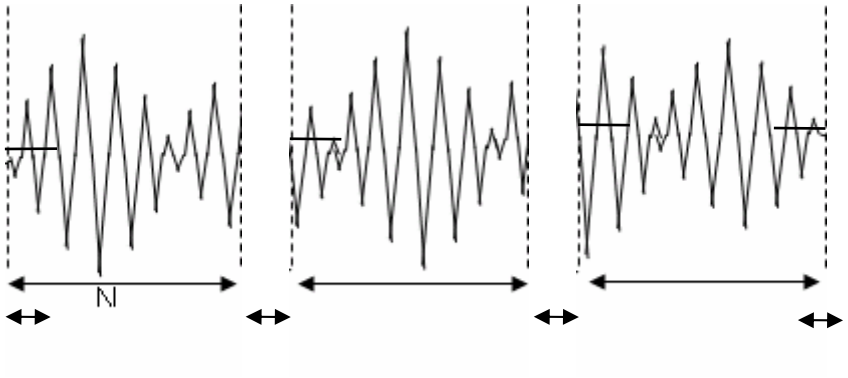
Pada Gambar 2.4 sinyal yang telah diratakan akan mempunyai magnitude yang lebih kecil dari sinyal aslinya [4]. Sinyal hasil normalisasi mempunyai *output* yang lebih memenuhi syarat. Sinyal ini tidak terdengar pecah atau terlalu keras.



Gambar 2.4 Sinyal hasil *pre-emphasis*

2.3.2 *Frame Blocking dan Windowing*

Sinyal kontinu akan dibagi-bagi dalam *frame*. Satu *frame* terdiri dari N *sample*. *Frame* kedua juga terdiri dari N *sample* [9]. Pada awal dan akhir *frame* pertama diberikan M *sample*. Pemberian *sample* tambahan juga dilakukan pada *frame* kedua. Pembagian *frame* dan pemberian *sample* tambahan akan dilakukan sampai pada *frame* terakhir. Proses *frame blocking* dicontohkan pada *Gambar 2.5*



Gambar 2.5 Proses *Frame Blocking*

Pemberian M *sample* pada awal dan akhir farme untuk mengurangi *error* saat proses. Kegunaan *sample* yang diberikan mirip dengan *guard band* pada proses *sampling.*

Adapun proses *windowing* akan meminimalisasikan sinyal yang telah di*frame-frame*kan sehingga sinyal akan nol pada permulaan dan akhir masing-masing *frame*.Sinyal yang baru tersebut seperti diberi *fade in* dan *fade out*. Jika *window* didefinisikan sebagai w(n), $0 \leq n \leq N-1$, dimana N adalah banyaknya *sample* pada masing-masing *frame*. Hasil *windowing* adalah sinyal yang dinyatakan dengan persamaan [12] :

$$y_1(n) = x_1(n)w(n), \quad 0 \leq n \leq N-1 \qquad \text{.....(2.2)}$$

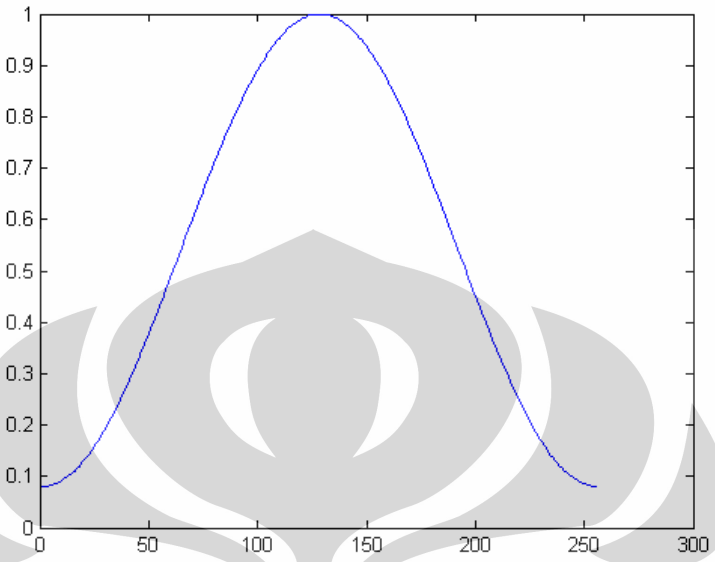
Dimana : $y_1(n)$ adalah sinyal hasil *windowing*

$x_1(n)$ adalah sinyal *input*

$w(n)$ adalah koefisien *windowing*

Pada skripsi ini menggunakan *Hamming windowing. Hamming Windowing* merupakan proses *windowing* yang cukup halus. Proses *Hamiing windowing* akan mengkonvolusi sinyal kontinu dengan sinyal kosinus [1] Gambar 2.6 adalah koefisien *Hamming windowing* dengan jumlah *sample* 256 titik. Persamaan *Hamming windowing* adalah [9] :

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), \qquad 0 \le n \le N\text{-}1 \qquad \text{.....(2.3)}$$



Gambar 2.6 koefisien *Hamming windowing* dengan jumlah *sample* 256 titik [1]

2.3.3 *Discrete Fourier Transform* dan *Fast Fourier Transfor*

Transformasi Fourier merupakan metode untuk mentransformasikan sinyal domain waktu menjadi sinyal domain frekuensi. Transformasi ini penting untuk analisis sinyal karena karakteristik sinyal domain frekuensi dapat diamati dengan lebih jelas dan dimanipulasi dengan lebih mudah dari pada sinyal domain waktu. Di domain frekuensi sinyal dapat direpesentasikan sebagai serangkaian nilai yang menunjukan banyaknya satuan sinyal yang berada di frekuensi tertentu. *Transform fourier* banyak digunakan untuk aplikas sains, misalnya, fisika, teori numerik, pemrosesan sinyal, statistik, akustik, optik,geometri dll.

Untuk melakukan *transformasi fourier* terhadap sinyal diskrit, digunakan *Discrete Fourier transform (DFT)* [9].

DFT menghasilkan serangkaian *N* buah nilai yang berindeks *k* di dalam domain frekuensi yang merupakan transformasi dari sinyal domain waktu yang berindeks *n*. Dari hasil tersebut, *X(k)* dan *X(N-k)* merupakan konjugasi kompleks. Karena magnitude dari konjugasi kompleks adalah sama, maka di dapatkan

$|X(k)| = |X(N-k)|$ Untuk k bernilai 0 sampai N/2. Dengan demikian, nilai hasil transformasi dalam domain frekuensi yang digunakan untuk analisis sinyal hanya nilai yang berindeks 0 samapai N/2 saja [1].

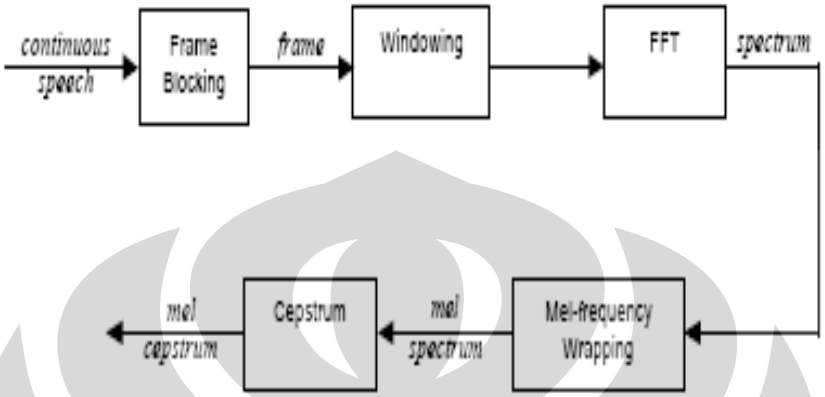
Untuk mengembalikan sinyal domain frekuensi ke domain waktu, digunakan persamaan transformasi inverse Persamaan DFT inverse didefinisikan sebagai berikut :

Fast Fourier Transform (FFT) dikembangkan oleh Cooley dan Tukey pada tahun 1965 [9]. Algoritma *FFT* merupakan penyederhanaan dari *DFT* yang memiliki persyaratan jumlah data harus merupakan bilangan $2n$ untuk n = 0,1,2,.... Waktu komputasi *DFT* memiliki kompleksitas N^2 sedangkan *FFT* memiliki kompleksitas $Np/2$ dengan $p=^2$ logN, sehingga *FFT* lebih cepat daripada *DFT* dengan rasio kecepatan *FFT* terhadap *DFT* [10] .

2.3.4 Mel Frequency Cepstrum Coefficients (MFCC)

Tingkat akurasi pengenalan suara yang diperoleh dengan menggunakan MFCC lebih tinggi dibanding metode ekstraksi fitur lainnya, seperti LPC (*Linear Predictive Coding*), LPCC (*Linear Predictive Cepstrum Coefficients*) dan lain-lain. MFCC didasarkan pada variasi batas *bandwidth* frekuensi pendengaran manusia MFCC memfilter secara linear pada frekuensi rendah dan secara logaritmik pada frekuensi tinggi untuk menangkap karakteristik penting dari sinyal suara. MFCC merupakan bentuk khusus dari *cepstrum* dan menggunakan skala *mel frequency*. Satu mel merupakan sebuah unit pengukuran (frekuensi) dari merasakan getaran pita suara (*perceived pitch*). Skala mel frekuensi memiliki spasi frekuensi linear untuk frekuensi dibawah 100 Hz dan spasi frekuensi logaritmik untuk frekuensi diatas 1000 Hz.

Skala mel frekuensi memungkinkan sistem untuk memperoleh persepsi pendengar yang sangat baik. Gambar 2.7 menunjukkan blok diagram struktur dari MFCC *processor input* sinyal [10]



Gambar 2.7 Blok diagram struktur dari MFCC *processor input* sinyal

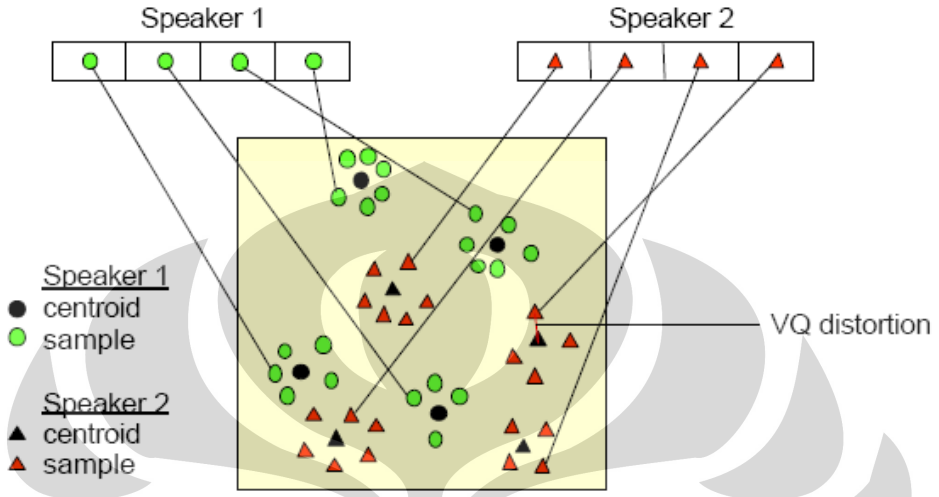
Persepsi manusia terhadap isi frekuensi dari suatu suara tidak mengikuti skala linear. Untuk itu masing-masing *tone* dengan sebuah frekuensi tertentu diukur dengan skala Hz, sedangkan sebuah *subject pitch* diukur pada skala yang disebut *mel.* Skala mel-frekuensi adalah pemetaan frekuensi secara linear untuk frekuensi dibawah 1 KHz dan logaritmik untuk frekuensi diatas 1 KHz. Hasil akhir dari proses MFCC adalah *mel cepstral coefficients.*

2.4 VECTOR QUANTIZATION

Vector quantization (VQ) adalah proses pemetaan vektor dari ruang vektor yang besar menjadi sebuah wilayah yang terbatas [7]. VQ akan sangat baik digunakan dalam *speech recognition* karena mengurangi kesalahan dan memiliki akurasi yang tinggi. VQ akan mengkompresi sinyal dalam domain frekuensi. Sinyal akan dikompresi sesuai dengan ukuran *codebook.*

Tiap wilayah disebut sebagai *cluster* dan dapat direpresentasikan oleh *centroid* yang disebut *codeword* [6]. Kumpulan dari *codeword* disebut *codebook.*Gambar 2.8 Menunjukkan diagram yang mengilustrasikan proses *recognition.* Pada gambar terdapat dua *user* dan dua dimensi. Lingkaran

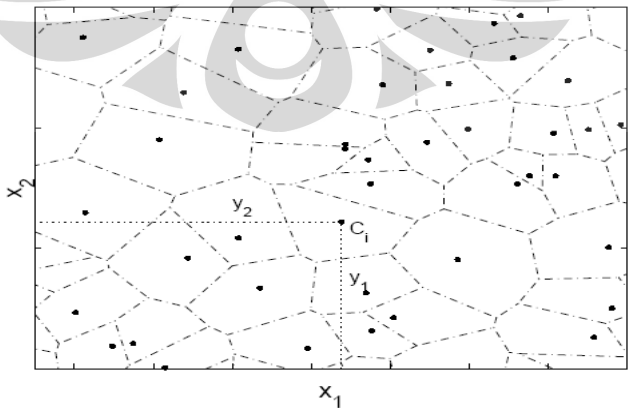
menunjukkan vektor dari *user* satu. Segitiga adalah *user* dua. Pada saat *training*, tiap *user* akan dikelompokkan dengan meng-*cluster* tiap-tiap vektornya. Jarak antara vektor yang dekat dengan *codeword* disebut sebagai *distortion* [6].



Gambar 2.8 Kumpulan *codebook*

VQ diinterpretasikan dengan skalar kuantisasi. Sinyal *input* akan dikuantisasi menjadi *codebook* C = $\{y_k \mid k = 1,...,N\}$. Sinyal *input* yang digunakan merupakan sebuah vektor yang harus dikodekan ke dalam ruang multidimensi. Gambar 2.9 adalah contoh ruang dua dimensi dari *codebook*. Pada gambar menunjukkan partisi dari ruang multidimensi sebuah *input* vektor yang dibagi menjadi *L* wilayah yang dapat dinotasikan sebagai P = $\{C_1, C_2, ..., C_L\}$ dimana [6] :

$$C_i = \{x \mid d(x, y_i) \le d(x, y_j), j \ne i\} \quad \text{.....(2.4)}$$



Gambar 2.9 *Codebook* secara multidimensi

Pada tahap *recognition*, sinyal *input* akan di vektor-kuantisasi menggunakan semua *trained codebook* dan selanjutnya dihitung total VQ *distortion*-nya. Total *distortion* yang paling kecil antara *codeword* dari salah satu sinyal dalam *database* dan VQ *codebook* dari sinyal *input* diambil sebagai hasil identifikasi. Dalam pembentukan *codebook* untuk iterasi guna meperbaiki VQ digunakan *General Lloyd Algoritm* (GLA) atau yang sering disebut dengan LBG *algorithm* [4]. LBG VQ *algorithm* tersebut dapat diimplementasikan dengan prosedur rekursif sebagai berikut :

1. Mendesign vektor *codebook* yang merupakan *centroid* dari keseluruhan vektor *training*
2. Melipatgandakan ukuran dari *codebook* dengan membagi masing-masing *codebook* C_n menurut aturan

$$C_n^+ = C_n(1+\varepsilon) \quad \text{.....(2.5)}$$

$$C_n^- = C_n(1+\varepsilon) \quad \text{.....(2.6)}$$

Dimana *n* bervariasi dari satu sampai dengan *current size codebook* dan ε adalah parameter *splitting* ($\varepsilon = 0.01$)

3. *Nearest Neighbour Search*

Mengelompokkan *training vector* yang mengumpul pada blok tertentu. Selanjutnya menentukan *codeword* dalam *current codebook* yang terdekat dan memberikan tanda vektor yaitu *cell* yang diasosiasikan dengan *codeword-codeword* yang terdekat [5].

4. *Centroid Update*

Menentukan *centroid* baru yang merupakan *codeword* yang baru pada masing-masing *cell* dengan menggunakan *training vector* pada *cell tersebut.*

5. Iterasi 1

Mengulang step 3 dan 4 sampai jarak rata-rata dibawah *present threshold*

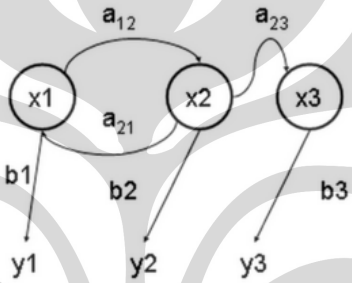
6. Iterasi 2

Mengulang step 2, 3, dan 4 sampai *codebook* berukuran *M.*

2.5 HIDDEN MARKOV MODEL (HMM)

2.5.1 Definisi Hidden Markov Model

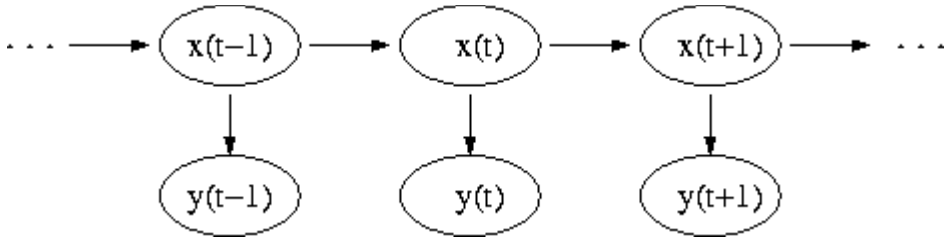
Hidden Markov Model (HMM) adalah suatu sistem yang dipermodelkan dengan proses Markov dengan parameter-parameter yang tidak diketahui [6]. Oleh karena itu harus dicari *hidden parameter* dari parameter *observable*. Parameter-parameter dari pemodelan dapat digunakan untuk melakukan analisa yang lebih lanjut. Contohnya adalah pola aplikasi pengenalan (*recognition*). Pada gambar 2.10 menunjukkan contoh sederhana HMM. *x* adalah *hidden states, y* adalah *observable outputs, a* adalah *transition probabilities,* dan *b* adalah *output probabilities* [8].



Gambar 2.10 *Hidden markov model*

Output dari suatu percobaan tidak bergantung pada kemungkinan *output* yang selanjutnya. Proses percobaan yang sebelumnya akan mempengaruhi *output* selanjutnya. Suatu proses akan dimulai dari suatu state dan menuju state yang lainnya. Setiap perubahan disebut *step*. Apabila dimulai dari suatu state x_i menuju x_j akan memiliki probabilitas a_{ij}. Probabilitas a_{ij} disebut sebagai *transition probabilities* [6].

Gambar 2.11 dibawah menunjukkan arsitektur HMM secara umum. Bentuk oval menunjukkan nilai variable *random*. Nilai *random x(t)* adalah nilai dari *hidden variable* pada waktu *t.* Tanda panah menunjukkan *conditional dependencies.* Diagram ini menunjukkan bahwa *hidden variable x(t)* (saat waktu *t*) hanya bergantung dari nilai *hidden varibale x(t-1*(saat waktu t-1). Hal ini disebut juga *Markov property*. Nilai dari *observed variable y(t)* bergantung dari nilai *hidden variable x(t)* [6]

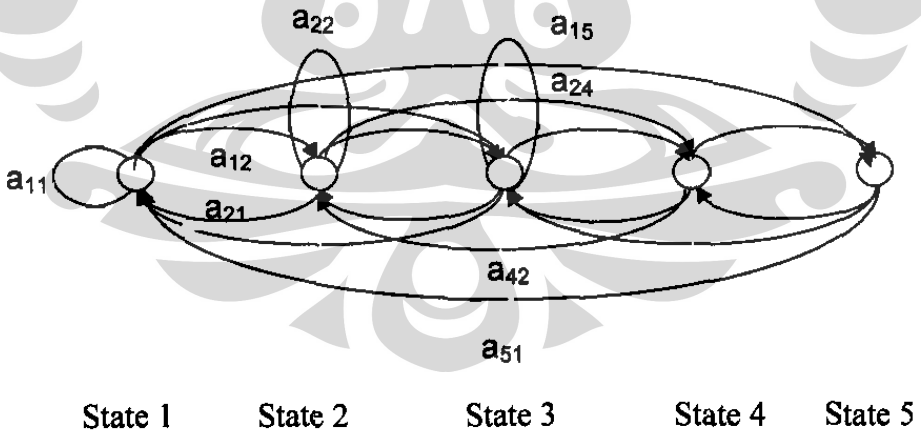


Gambar 2.11 Arsitektur HMM secara umum

Probabilitas dari urutan observer Y = y(0),y(1),…,y(L-1) dengan panjang L ditunjukkan dengan persamaan [4] :

$$P(Y) = \sum_{X} P(Y \mid X) P(X) \quad \text{.....(2.7)}$$

HMM akan diklasifikasikan dengan melihat bentuk matriks transisinya dari rantai Markov. Bentuk yang akan digunakan adalah bentuk *ergodic* atau bentuk state saling terhubung. a_{ij} adalah *state transition* dengan nilai antara 0 dan 1 ($0 < a_{ij} < 1$). Gambar 2.12 adalah contoh matriks transisi untuk model *ergodic* [6]



Gambar 2.12 Contoh matriks transisi untuk model *ergodic*

Dengan HMM dapat dituliskan sebagai $\lambda = (A, B, \Pi)$ maka dapat didefinisikan :

$$A = a_{ij} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \qquad B_i = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \qquad \Pi_i = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}$$

dimana :

A = Probabilitas dari transisi matriks

B = Probabilitas dari kemunculan state i

Π = Probabilitas dari awal pada state I

State 1 : *waveform* segment 1 (W1)

State 2 : *waveform* segment 2 (W2)

State 3 : *waveform* segment 4 (W4)

State 4 : *waveform* segment 5 (W5)

Jadi probabilitas dari observasi HMM :

Suara 1 → (w1, w2, w2, w1, w1) = $c_1 * a_{12} * b_2 * a_{22} * b_2 * a_{21} * b_1 * a_{11} * b_1$

Suara 2 → (w1, w2, w1, w3, w1) = $c_1 * a_{12} * b_2 * a_{21} * b_1 * a_{13} * b_3 * a_{31} * b_1$

⋮

Suara x → (w4, w5, w4, w5, w4) = $c_4 * a_{45} * b_5 * a_{54} * b_4 * a_{45} * b_5 * a_{54} * b_4$

Proses yang terjadi adalah :

1. Gelombang yang telah dibagi menjadi gelombang-gelombang kecil pada *frame blocking* akan dikenali melalui *codebook* yang dimiliki. Pada proses pencocokan

dengan *codebook* akan dihitung jarak dari tiap gelombang dengan *centroid-centroid.* Jarak terdekat akan menentukan urutan kode observasi

2. Gelombang yang telah dikenali berdasarkan *codebook* akan dicocokkan dengan nilai pada parameter HMM. Parameter HMM sesuai dengan urutan kode observasi.

Pada contoh diatas dapat diketahui bahwa suara 1 terbentuk dari gelombang w1, gelombang w2, gelombang w2, gelombang w1, gelombang w1. Tiap suara dibentuk oleh susunan gelombang yang berbeda-beda. Susunan-susunan gelombang tersebut memiliki probabilitas transisi yang bergantung terhadap perubahan gelombangnya. Jika gelombang dari n menuju m, maka probabilitas transisinya adalah a_{nm}. Selain itu juga memiliki probabilitas kemunculan dari state selanjutnya b_m.

2.5.2 Kendala pada HMM

1. Menghitung probabilitas dari urutan *output* pada parameter pemodelan,. Kendala ini dapat dipecahkan dengan *forward-backward procedure.*

2. Mencari urutan yang paling baik pada parameter pemodelan dari *hidden states* yang dapat menghasilkan urutan *output* yang diinginkan. Permasalahan ini dipecahkan dengan algoritma Viterbi.

3. Mencari *set of state transition* dan kemungkinan *output* yang paling baik pada *output sequences*.

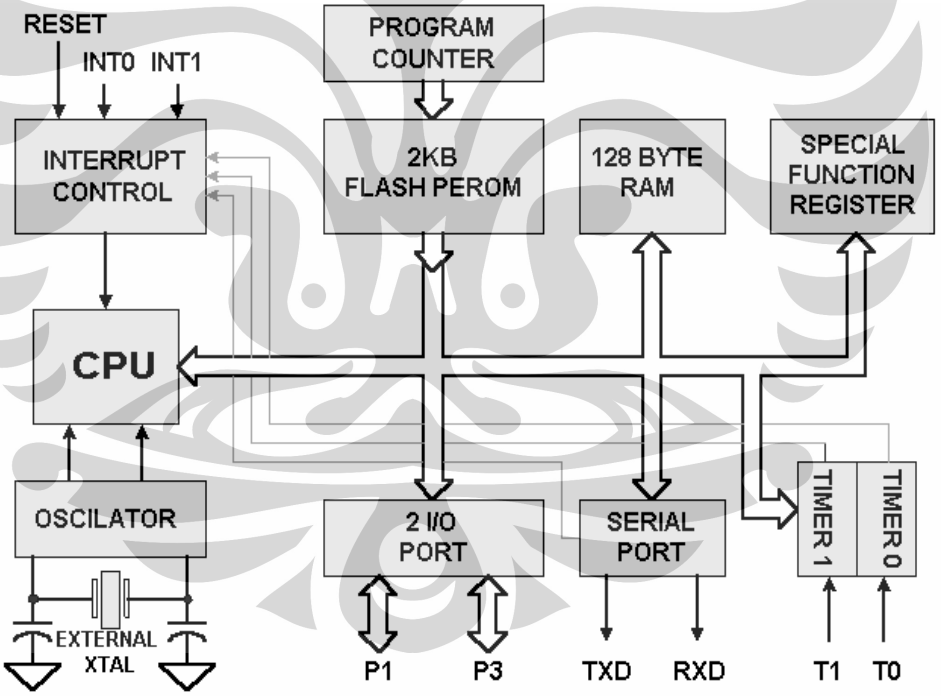
2.6 *MIKROKONTROLER AT89S51*

Dalam perancangan alat ini, digunakanlah IC mikrokontroler dengan jenis AT89S51 yang diproduksi pertama kali oleh intel. Pada intel, seri-seri mikrokontroler berarsitektur 8951 tergabung dalam satu keluarga mikrokontroler yaitu keluarga MSC-51. Seri mikrokontroler berarsitektur 8951, baik dari keluarga Intel MCS-51 maupun dari produsen lainnya, memiliki beragam tipe dan fasilitas, namun semuanya memiliki arsitektur yang sama, dan juga set instruksi yang relatif tidak berbeda. Dimana mikrokontroller jenis ini dibuat dengan teknologi CMOS sehingga akan mengkonsumsi daya yang rendah. Salah satu tipe mikrokontroller yang menjadi andalan dari keluarga MCS-51 adalah tipe 89S51.

Tipe ini banyak digunakan karena memiliki fasilitas On-Chip Flash memory. Mikrokontroler jenis 89S51 memiliki fasilitas-fasilitas pendukung, antara lain [16]:

- 4K *bytes* ROM
- 128 *bytes* RAM atau lebih
- 4 buah 8-bit I/O (Input/Output) port.
- 2 buah 16 bit timer, yaitu Timer 0 dan Timer 1
- 2 Jalur Interrupt, yaitu Interrupt 0 dan Interrupt 1
- Port Komunikasi serial, melalui pin TxD dan RxD.

IC mikrokontroler dikemas dalam bentuk yang berbeda, namun pada dasarnya fungsi kaki yang ada pada IC memiliki persamaan. Adapun blok diagram dari IC AT89S51 ini adalah [16]:



Gambar 2.13 *Blok Diagram Mikrokontroler AT89C51*

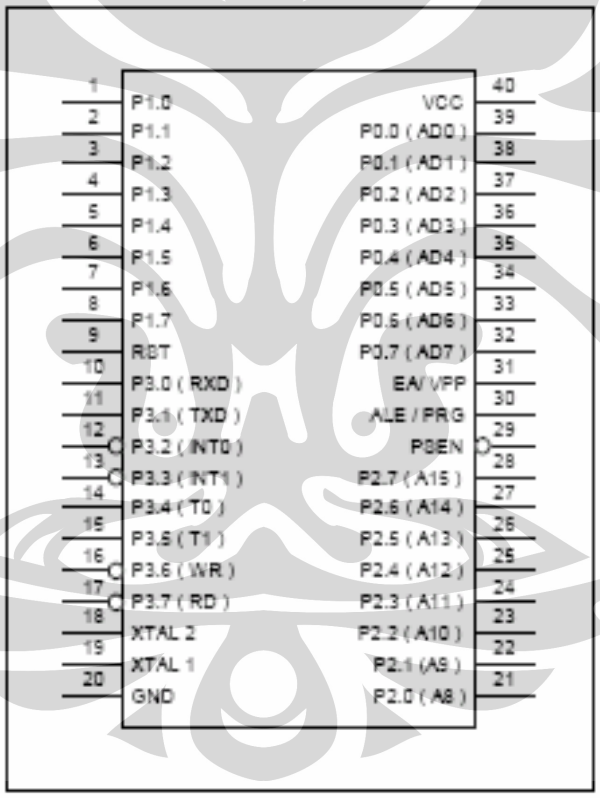
2.6.1 Organisasi Memori

Mikrokontroler AT89S51 memiliki pembagian ruang alamat untuk program dan data. Memori data diakses oleh alamat 8 bit, tetapi alamat data 16 bit

juga dapat dihasilkan mikrokontroler melalui regiater DPTR (*Data Pointer Register*). Alamat data dan program yang bisa dialamatkan oleh mikrokontroler adalah sebesar 64 kilo*byte* yaitu dari alamat 0000_H-$FFFF_H$.

2.6.2 Deskripsi Pin-Pin Pada Mikrokontroler 8952

Pada mikrokontroler dengan tipe 89S51 ini memiliki 4 port, yaitu port 0, port 1, port 2, dan port 3. Setiap port ini memiliki kaki atau pin yang berjumlah 8. Contohnya pada port 0 yang memiliki kaki atau pin yang dimulai dari P0.0 hingga P0.7. Berikut ini adalah penjelasan fungsi tiap-tiap kaki atau pin-pin yang ada pada seri mikrokontroler 89S51 [16].



Gambar 2.14
Pin-Pin Mikrokontroler AT89C52

Port 0

Port ini merupakan *dual-purpose* port (port yang memiliki dua kegunaan). Biasanya digunakan sebagai port I/O (*Input/Output*). Kegunaan lain sebagai saluran data dan setengah saluran-alamat. Port 0 ini terdapat pada pin 32-39.

Port 1

Port 1 pada IC mikrokontroler 89S51 hanya berfungsi sebagai port I/O. Port 1 pada IC mikrokontroler 89S51 ini terdapat pada pin 1-8

Port 2

Port 2 ini merupakan *dual-purpose* port. Pada desain minimum digunakan sebagai port I/O atau sebagai *High-Byte* pada *address* bus untuk sebuah desain yang menggunakan kode eksternal memori. Port 2 terdapat pada pin 21-28

Port 3

P3.0	RXD (Serial *Input* port)
P3.1	TXD (Serial *output* port)
P3.2	INT0 (External Interrupt 0)
P3.3	INT1 (External interrupt 1)
P3.4	T0 (Timer 0 external *input*)
P3.5	T1 (Timer 1 external *input*)
P3.6	WR (External data memory write strobe)
P3.7	RD (External data memory read strobe)

PSEN (Program Store Enable)

PSEN adalah kontrol sinyal yang digunakan untuk mengakses program memori eksternal. PSEN terdapat pada pin 29

ALE (Address Latch Enable)

Keluaran ALE menghasilkan pulsa-pulsa untuk mengancing byte rendah alamat selama mengakses memori eksternal. ALE terdapat pada pin 30.

EA (External Access)

Pada IC mikrokontroler tipe 8951 jika EA diberikan kondisi input (masukan) 1 maka akan menjalankan program memory internal saja. Sedangkan jika EA diberikan input 0 (ground) maka IC ini akan menjalankan program memori eksternal (PSEN akan bernilai 0). EA terdapat pada pin 31.

RST (Reset)

Reset terdapat pada pin 9 yang merupakan reset dari 89S51. jika pada pin ini diberikan masukan 1 selama minimal 2 machine cycle maka sistem akan di-reset dan register-register internal pada 89S51 akan berisi nilai default tertentu.

***On-Chip Oscillator* Input**

IC 89S51 memiliki *On-chip Oscillator* yang dapat bekerja jika di-drive menggunakan kristal. Tambahan kapasitor diperlukan untuk menstabilkan sistem. Nilai kristal yang biasa digunakan pada keluarga MCS-51 adalah 12 MHz. *On-chip oscillator* dihubungkan oleh kristal pada pin 18 dan 19 (XTAL 1 DAN XTAL 2).

Koneksi Power

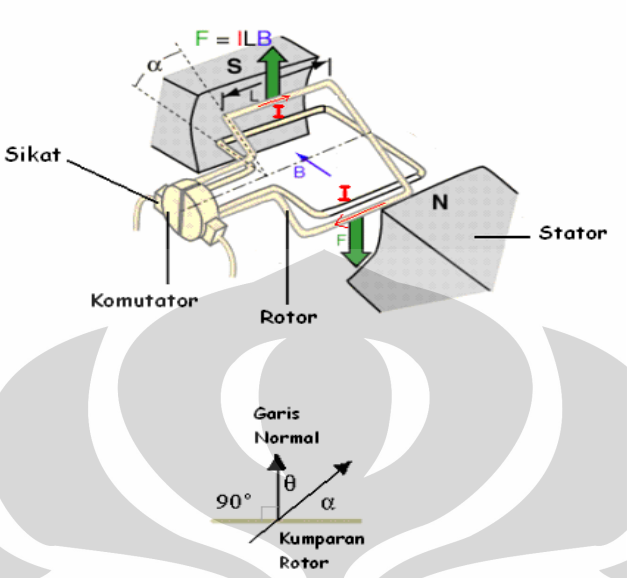
IC 8951 beroperasi pada tegangan 5 volt. Pin Vcc terdapat pada kaki 40 dan Vss (*Ground*) terdapat pada kaki 20.

2.7. *MOTOR DC*

Motor arus searah (motor DC) adalah mesin listrik yang mengubah daya masuk listrik menjadi daya keluar mekanik dan menggunakan arus listrik searah (DC). Pada dasarnya, motor DC terdiri dari 4 bagian yaitu :

- Bagian yang berputar disebut dengan rotor, yaitu kumparan jangkar
- Bagian yang diam disebut dengan stator, yaitu kumparan medan.
- Komutator, terbuat dari batangan tembaga yang dikeraskan dan diisolasi dengan bahan sejenis mika, serta berfungsi untuk mengumpulkan arus listrik induksi dari konduktor jangkar dan mengkonversikannya menjadi arus searah melalui sikat.

♣ Sikat, terbuat dari karbon, grafit, logam grafit, atau campuran karbon-grafit [17].



Gambar 2.15 Motor arus searah

2.7.1 Prinsip Kerja Motor Arus Searah

Prinsip kerja motor DC mempunyai prinsip dasar yang sama dengan motor stepper namun gerakannya bersifat kontinyu atau berkelanjutan. Berdasarkan gambar 2.15 di atas, maka prinsip kerja motor DC adalah :

- Jika sebuah kumparan yang berada dalam medan magnet dialiri arus listrik, maka pada tiap sisi dari kumparan tersebut akan timbul suatu gaya yang dinamakan *Gaya Lorentz*, di mana rumusnya adalah sebagai berikut [17] :

$$F = B.i.L.\sin\theta \quad \text{.....(2.8)}$$

Di mana :

F : gaya Lorentz (Newton)
B : medan magnet (Weber / m^2)
i : kuat arus (Ampere)
L : panjang kawat (Meter)
θ : sudut antara arus dengan medan magnet (Derajat)

Arah dari *Gaya Lorentz* tersebut dapat ditentukan dengan menggunakan *aturan tangan kiri*, yaitu:

"Bila arah garis gaya menusuk telapak tangan kanan dan jari-jari tangan kanan menunjukkan arus, maka ibu jari menunjukkan arah gaya yang timbul."

- Garis gaya yang terjadi pada sekeliling kumparan jangkar akan menimbulkan sebuah momen, di mana besarnya momen tersebut adalah [17] :

$$M = B.i.N.A.\frac{D}{2}\sin\theta \quad \text{……..…….....……} \quad (2.9)$$

Di mana :

M : momen yang timbul

B : medan magnet (Weber / m^2)

i : kuat arus listrik (Ampere)

N : jumlah lilitan (Lilitan)

A : luas penampang yang dilalui oleh medan magnetis (m^2)

θ : sudut antara arus listrik dengan medan magnet (derajat)

D : diameter rotor (Meter)

- Bila momen yang terjadi lebih besar daripada momen lawan, maka motor akan berputar.
- Jika motor berputar, maka kumparan-kumparan di dalam jangkar (sama seperti generator) akan terjadi GGL, tetapi arah dari GGL yang timbul ini berlawanan arah dengan arus yang masuk ke dalam motor, maka GGL yang terjadi disebut dengan GGL lawan *(back emf)*. Besarnya GGL lawan tersebut adalah [17] :

$$E = N\frac{d\Phi}{dt} \quad \text{……..…….....…..……....} \quad (2.10)$$

Di mana : E : besarnya GGL lawan (Newton)

N : jumlah lilitan (Lilitan)

dΦ/dt : besarnya perubahan fluks magnetik terhadap waktu

Untuk mendapatkan arus GGL lawan *(back emf)* ini, maka kita membutuhkan energi listrik. Daya listrik inilah yang akan diubah oleh motor DC menjadi daya mekanik

2.8 *BLUETOOTH*

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter). Bluetooth sendiri dapat berupa card yang bentuk dan fungsinya hampir sama dengan card yang digunakan untuk wireless local area network (WLAN) dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada bluetooth mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah [18].

Pada dasarnya bluetooth diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan kabel didalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi mobile wireless dengan biaya yang relatif rendah, konsumsi daya yang rendah, interoperability yang menjanjikan, mudah dalam pengoperasian dan mampu menyediakan layanan yang bermacam-macam.

Protokol bluetooth menggunakan sebuah kombinasi antara circuit switching dan packet switching. Bluetooth dapat mendukung sebuah kanal data asinkron, tiga kanal suara sinkron simultan atau sebuah kanal dimana secara bersamaan mendukung layanan data asinkron dan suara sinkron. Setiap kanal suara mendukung sebuah kanal suara sinkron 64 kb/s. Kanal asinkron dapat mendukung kecepatan maksimal 723,2 kb/s asimetris, dimana untuk arah sebaliknya dapat mendukung sampai dengan kecepatan 57,6 kb/s. Sedangkan untuk mode simetris dapat mendukung sampai dengan kecepatan 433,9 kb/s [18].

Sebuah perangkat yang memiliki teknologi wireless bluetooth akan mempunyai kemampuan untuk melakukan pertukaran informasi dengan jarak jangkauan

sampai dengan 10 meter (~30 feet). Sistem bluetooth menyediakan layanan komunikasi point to point maupun komunikasi point to multipoint.

Sistem bluetooth terdiri dari sebuah radio transceiver, baseband link controller dan sebuah link manager. Baseband link controller menghubungkan perangkat keras radio ke base band processing dan layer protokol fisik. Link manager melakukan aktivitas-aktivitas protokol tingkat tinggi seperti melakukan link setup, autentikasi dan konfigurasi.

2.8.1 Karakteristik Radio

Berikut beberapa karaketristik radio bluetooth sesuai dengan dokumen Bluetooth SIG yang dirangkum dalam tabel 2.1 [18].

Tabel 2.1 Karakteristik Radio Bluetooth

Parameter	Spesifikasi
Transmitter :	
Frekuensi	ISM band, 2400 - 2483.5 MHz (mayoritas), untuk beberapa negara mempunyai batasan frekuensi sendiri (lihat tabel 2), spasi kanal 1 MHz.
Maximum Output Power	Power class 1 : 100 mW (20 dBm)Power class 2 : 2.5 mW (4 dBm)Power class 3 : 1 mW (0 dBm)
Modulasi	GFSK (Gaussian Frequency Shift Keying), Bandwidth Time : 0,5; Modulation Index : 0.28 sampai dengan 0.35.
Out of band Spurious Emission	30 MHz - 1 GHz : -36 dBm (operation mode), -57 dBm (idle mode)1 GHz – 12.75 GHz: -30 dBm (operation mode), -47 dBm (idle mode)1.8 GHz – 1.9 GHz: -47 dBm (operation mode), -47 dBm (idle mode)5.15 GHz –5.3 GHz: -47 dBm (operation mode), -47 dBm (idle mode)
Receiver :	
Actual Sensitivity Level	-70 dBm pada BER 0,1%.
Spurious Emission	30 MHz - 1 GHz : -57 dBm1 GHz – 12.75 GHz : -47 dBm

Max. usable level	-20 dBm, BER : 0,1%

2.8.2 Pita Frekuensi dan Kanal RF

Bluetooth beroperasi dalam pita frekuensi 2,4 GHz ISM, walaupun secara global alokasi frekuensi bluetooth telah tersedia, namun untuk berbagai negara pengalokasian frekuensi secara tepat dan lebar pita frekuensi yang digunakan berbeda.

Batas frekuensi serta kanal RF yang digunakan oleh beberapa negara dapat dilihat pada table 2.2 [18]

Tabel 2.2 Batas frekuensi dan kanal RF bluetooth

Negara	Range Frekuensi	Kanal RF	
Eropa *) dan USA	2400 – 2483,5 MHz	f = 2402 + k MHz	k = 0,…,78
Jepang	2471 – 2497 MHz	f = 2473 + k MHz	k = 0,…,22
Spanyol	2445 – 2475 MHz	f = 2449 + k MHz	k = 0,…,22
Perancis	2446,5 – 2483,5 MHz	f = 2454 + k MHz	k = 0,…,22

***) Kecuali Spanyol dan Perancis**

2.9 *MIKROFON*

Mikrofon adalah sebuah alat yang mengubah suara menjadi sinyal elektrik. Mikrofon banyak terdapat dalam alat-alat yang digunakan dalam kehidupan sehari-hari, contohnya : telepon, alat perekam suara, alat bantu pendengaran, dalam siaran radio atau televisi, dan sebagainya.

BAB 3
PERANCANGAN SISTEM

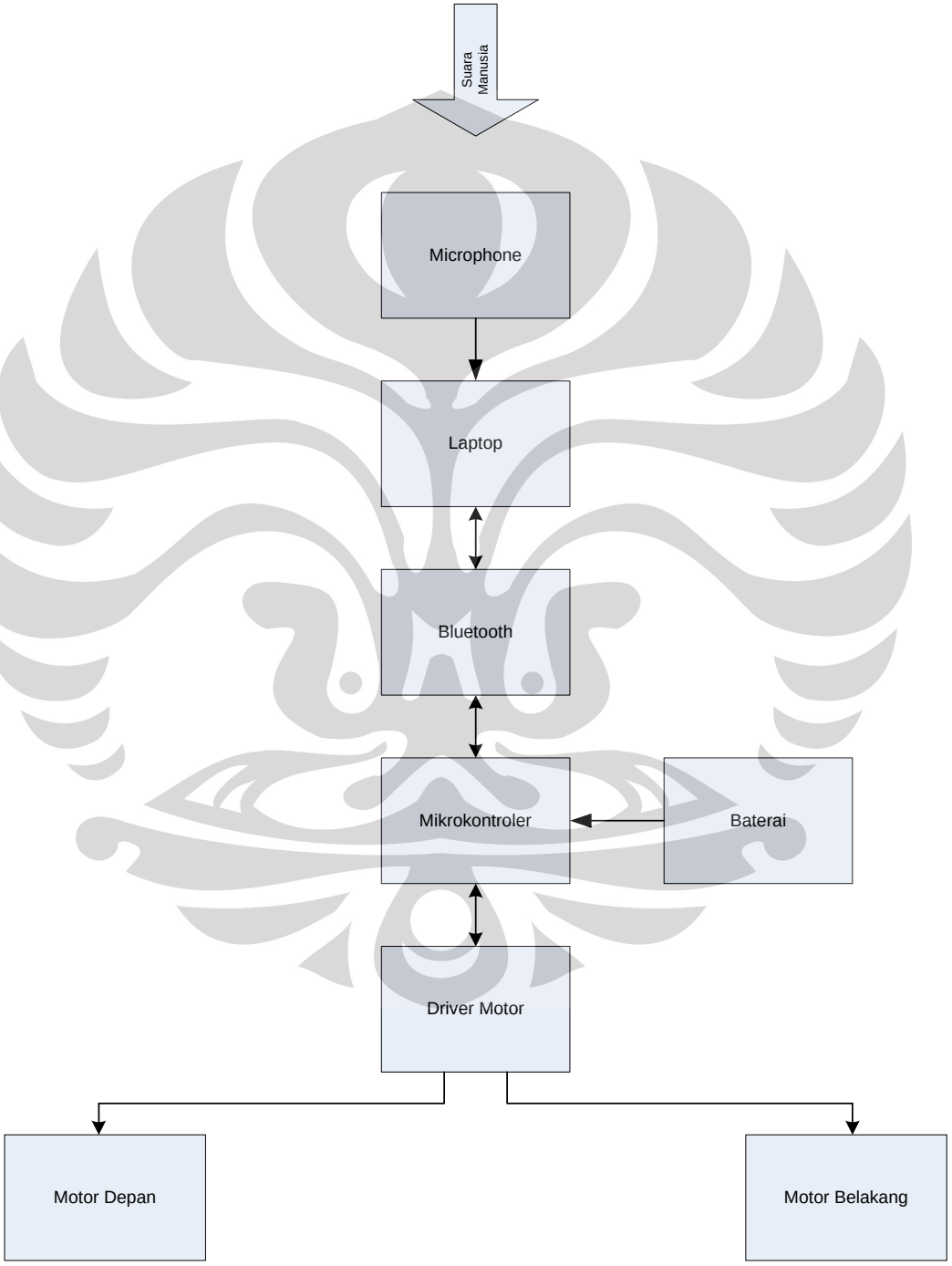
3.1 Gambaran Umum

Pada bab ini akan dibahas mengenai fungsi alat dan blok diagram secara keseluruhan serta cara kerjanya secara rinci dalam tiap-tiap sub bab. Pada prinsipnya alat ini berfungsi untuk menggerakkan sebuah miniatur robot mobil dengan menggunakan suara manusia sebagai pengendalinya yang sebelumnya telah disimpan dalam database komputer. Pemakaian alat dimulai dengan menyebutkan kata kunci melalui sebuah microphone yang akan dicocokkan dengan data yang ada di dalam database. Apabila input yang diberikan cocok dengan data yang ada, maka perintah akan diteruskan kepada mikrokontroler yang terdapat pada alat peraga melalui bluetooth. Kemudian mikrokontroler akan menggerakkan kedua motor listrik yang terdapat pada alat peraga sesuai dengan perintah yang telah diberikan.

Dengan diperlukanya sebuah sistem kontrol untuk setiap operasi tertentu, misalnya mesin produksi maupun sistem control yang lain maka di perlukan juga suatu sistem pengendali yang bisa berfungsi dengan baik. Sehingga performance yang diharapkan pada setiap pengaturan yang diinginkan akan sesuai dengan apa yang telah dirancang sesuai program yang dibuat, demikian juga pada sistem yang dibuat oleh penulis yaitu untuk dapat menggerakkan miniatur robot mobil dengan menggunakan suara manusia. Perancangan pada bab ini menggunakan perangkat keras meliputi beberapa bagian yaitu: bagian pengontrol utama yaitu Mikrokontroler sebagai otak pengendali, sinyal masukan yaitu berupa perintah suara manusia dan rangkaian penggerak yaitu pengendalian motor DC sebagai plant yang diproses untuk menggerakan model kendaraan sesuai dengan perintah yang diberikan oleh suara manusia.

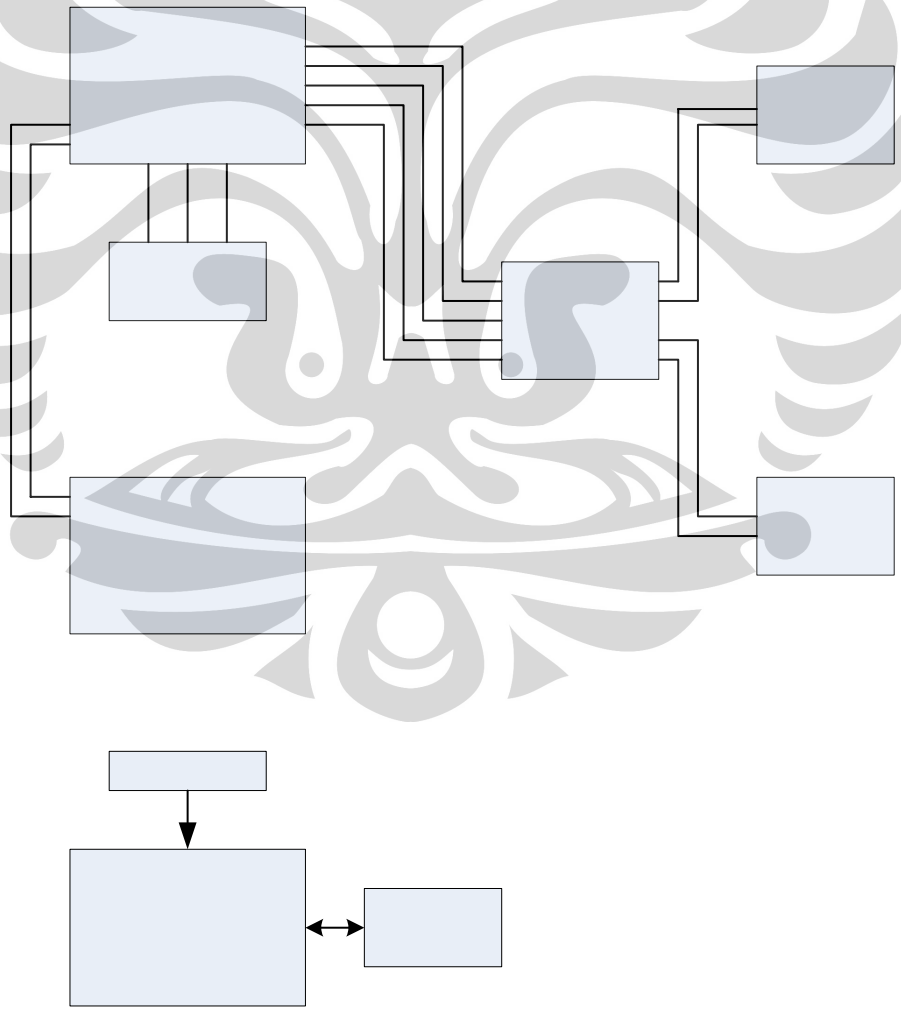
3.2 Diagram Blok Rangkaian

Pada perancangan ini dibuat diagram blok terlebih dahulu untuk mengambarkan sistem dari alat secara keseluruhan. Dari setiap blok kemudian dibuat rangkaian sesuai fungsinya seperti gambar 3.1 dibawah ini.



Gambar 3.1 Blok diagram rangkaian

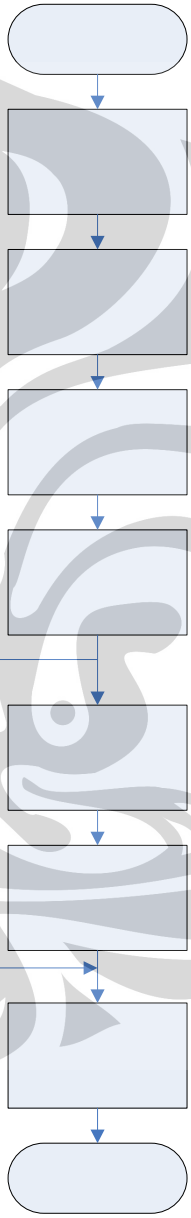
Berdasarkan gambar blok diagram di atas, fungsi kerja dari masing-masing blok adalah suara manusia diinputkan ke dalam sebuah sistem melalui microphone, kemudian suara tersebut akan dicocokkan untuk dikenali dengan suara yang terdapat dalam database di dalam laptop. Apabila suara tersebut cocok dengan database maka perintah akan diteruskan kedalam mikrokontroler melalaui bluetooth yang terdapat pada laptop dan juga pada alat peraga. Selanjutnya mikrokontroler akan memproses perintah tersebut yang kemudian akan diteruskan ke dalam driver motor. Driver motor digunakan untuk menggerakan dan meningkatkan arus dan tegangan yang akan dialirkan kepada kedua motor yang akan digunakan untuk menggerakan miniatur robot mobil.



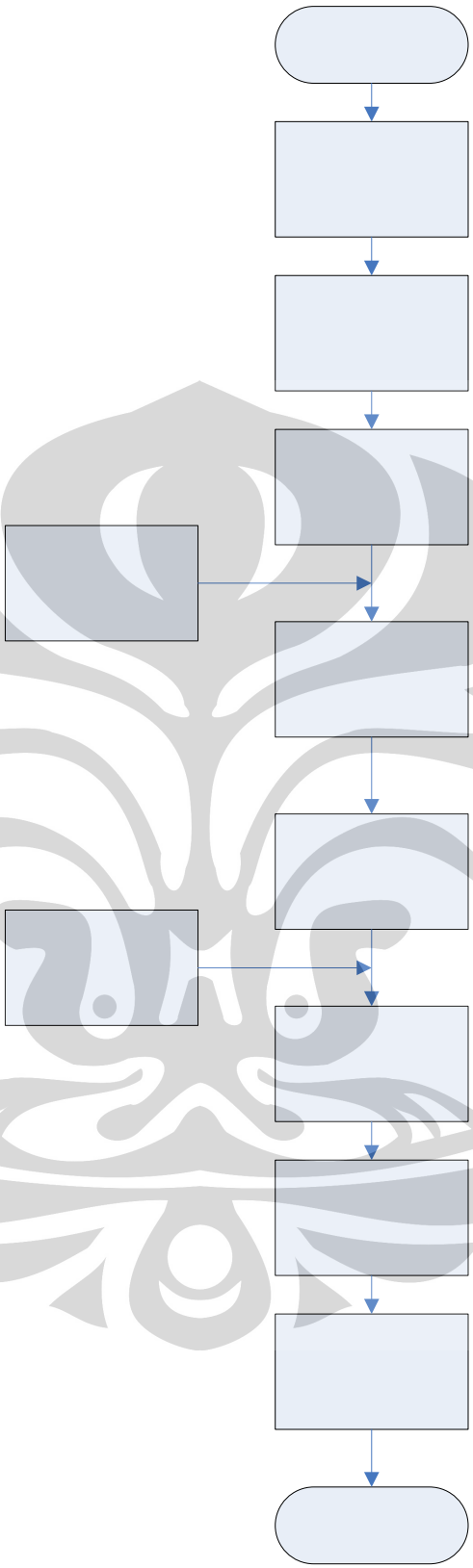
gambar 3.2 Rangkaian sistem pengendali miniatur robot mobil

3.3 Proses Pembentukan Database dan Recognition

Dalam proses yang dilakuakan terdiri dari dua tahap yaitu pembentukan *database* dan *recognition.* Berikut ini *flowchart* dalam pembentukan database dan *recognition*.



gambar 3.3 Flowchart pembentukan *database*



Database
(*codebook*)

gambar 3.4 Flowchart recognition

Sistem kerja yang dirancang untuk aplikasi ini adalah sebagai berikut :

1. Pada aplikasi ini setiap *user* akan merekam suaranya dengan menggunakan aplikasi signal getter(signal_getter.m) pada MATLAB. Pada proses *training* suara *user* diambil untuk diproses pada MATLAB. Proses pada MATLAB untuk mendapatkan *database codebook* dan *database* parameter HMM. Pada proses *training* suara *user* diambil untuk dikenali.
2. Pada aplikasi ini setelah *database* telah dibuat. *User* dapat memulai proses pengenalan suara dengan cara menekan tombol Get Signal pada program voice_recognition.m. Hasil proses pengenalan suara akan menggerakkan kendaraan sesuai dengan perintah yang diucapkan.
3. Pada saat berhasil dikenali kendaraan akan bergerak sesuai dengan urutan suara yang telah di-*training*. Urutan pergerakan kendaraan tersebut adalah :

Kendaraan bergerak depan adalah "Depan"
Kendaraan bergerak belakang adalah "Belakang"
Kendaraan bergerak ke kanan adalah "Kanan"
Kendaraan bergerak ke kiri adalah "Kiri"
Kendaraan berhenti bergerak adalah "Stop"

3.4 ALGORITMA SPEECH RECOGNITION

Secara garis besar algoritma dari *speech recognition* akan menggunakan pemprosesan secara digital. Teknologi *speech recognition* digunakan untuk mengetahui ucapan dari *user*. *Speech recognition* dapat mengetahui informasi yang diucapkan *user* dan dijadikan sebagai *input* sistem lainnya. *Speech recognition* akan menguji dari vokal karakteristiknya dengan akses yang dimiliki. Sistem yang telah dibuat dapat mengenali vokal karakteristik baik dari yang telah dilatih maupun yang belum dilatih.

3.4.1 Algoritma *Training Hidden Markov Model* Menggunakan MATLAB

Prosedur dalam men-*training* sistem yang akan dimulai dari mengambil suara yang akan di-*training*. Suara yang diambil adalah sepuluh suara dari tiap label yang akan di-*training.* Berarti terdapat 5 *database* suara.

Pada saat pendeklarasian, suatu individu akan diminta mengulang angka dan phrasa-phrasa. Phrasa yang pendek akan sulit dalam pengenalan. Phrasa yang panjang akan mengurangi keakuratan. Oleh karena itu individu akan mengulang beberapa phrasa. Oleh karena waktu yang diberikan hanya 2 detik, maka untuk satu kata direkam dalam phrasa yang berbeda-beda.

Suara dari *user* akan direkam ke dalam PC menggunakan program signal_getter.m dengan format **.wav*. Suara yang direkam akan disesuaikan durasinya selama 2 detik. Suara langsung terekam ke PC. Frekuensi *sampling* yang digunakan adalah 11025 Hz. Sinyal yang dihasilkan telah dinormalisasi pada level 1 volt sampai -1 volt. Setiap pengambilan data suara terdapat 22050 *sample* dijadikan *input* untuk menghasilkan *codebook* dan men-*training* state dari HMM. Proses *training* akan dilakukan pada MATLAB dengan melakukan tiga tahapan; yaitu labelisasi, membuat *codebook* dan men-*training* HMM. Data yang diambil adalah lima suara yaitu 'Depan', 'Belakang', 'Kanan', 'Kiri' dan 'Stop' dari satu *user*. Hasil *training* pada MATLAB akan digunakan sebagai *database*. Pada awalnya akan ditangkap suara dari *user* terlebih dahulu. Suara dapat ditangkap dengan menggunakan banyak alat, misalnya telepon ataupun *microphone*. Hasil dari *speech recognition* tergantung dari kualitas sinyal *audio*-nya. Alat yang digunakan adalah alat yang memiliki *noise* sangat sedikit.

Data suara yang telah direkam akan menjadi sebuah matrik (22050 x 1). Selanjutnya akan digunakan MATLAB untuk melakukan proses *training.* Pada tiap label akan ditentukan banyaknya pengulangan. Banyaknya pengulangan sesuai dengan jumlah *training.* Pada pembuatan *database* tiap label akan dilakukan *training* dengan data suara yang berbeda-beda. Tiap waktu ucapan yang diambil akan berbeda-beda. Hal ini tergantung dari kondisi *user.* Fungsi ini akan menghasilkan lima label (*label1,label2,label3,label4,label5*) karena hanya

terdapat lima suara yang di-*training*. Label yang terbentuk adalah :

1. *Depan*
2. *Belakang*
3. *Kanan*
4. *Kiri*
5. *stop*

Proses *training* yang dilakukan hanya menggunakan 10 data pada masing-masing label . Adapun urutan kerja dari pembuatan label adalah :

1. Masukkan nama label sesuai dengan nama *file* yang tersimpan
2. Menentukan jumlah *training*. Jumlah *training* akan menentukan banyaknya data yang diambil tiap label.
3. Nilai indeks label akan bertambah dan ulangi langkah ke-1 sampai semua label masuk

Tabel 3.1 menunjukkan hasil dari *labelisasi* yang tersimpan dalam label 1 sampai label 5.

Tabel 3.1 Nama *file labelisasi*

Kata	Nama *file*	10 *training*
Depan		Depan1 – Depan10
Belakang		Belakang1 – Belakang10
Kanan		Kanan1 – Kanan10
Kiri		Kiri1 – Kiri10
Stop		Stop1 – Stop10

Data yang telah disusun dalam label akan diekstraksi untuk menghasilkan data yang dapat diolah untuk men-*training* parameter Hidden Markov. Proses ekstraksi yang akan dilakukan adalah *pre-emphasis, frame blocking, windowing, Discrete Fourier Transform/Fast Fourier Transform*, dan *Mel Frequency Cepstrum Coefficients* (MFCC). Pada proses ini akan dihasilkan sinyal dengan domain frekuensi [10].

Data yang dimiliki akan dibagi-bagi dalam *frame-frame*.Satu *frame* akan memiliki 256 data. Data suara yang terekam adalah 100 dari 256 data dalam satu

frame. 152 data yang lain adalah data kosong untuk mencegah terjadinya *error* dalam pengolahan data. Data yang telah di-*frame*-kan akan di *windowing* menggunakan *Hamming window*.

Kemudian semua data *sample* akan dilakukan proses FFT. Proses FFT yang dilakukan adalah FFT 256 *point* karena data satu *frame* adalah 256 data. Hasil FFT akan dibagi dengan 1,000. Pembagian ini agar hasil dari FFT tidak terlalu besar. Hasil FFT yang terlalu besar disebabkan oleh data hasil rekaman pada level suara -2500 – 2500. Kemudian akan dilakukan proses MFCC untuk mendapatkan *mel cepstral coefficients* dengan listing program sebagai berikut.

```
function [spec,logSpec,f,Pcoeff1,coeffs1,coeffs2]=mfcc3(signal,fs)
spec=fft(signal);
logSpec=log(abs(spec(1:floor((length(spec)+1)/2))));
nbpts=length(logSpec);

i=1;
freq(i)=1;
f(i)=1;
while freq(i)<nbpts
  i=i+1;
  f(i)=(exp(log(2)*(150*(i-1))/1000)-1)*1000;
  freq(i)=floor((exp(log(2)*(150*(i-1))/1000)-1)*1000*2/fs*nbpts);
end

freq(i+1)=floor((exp(log(2)*(150*i)/1000)-1)*1000*2/fs*nbpts);
 NbFilters=length(freq)-2;
filters=zeros(NbFilters,freq(end)-freq(end-1));
for k=1:NbFilters

  filters(k,1:freq(k+2)-freq(k))=triang(freq(k+2)-freq(k))';

end

ZerosVect=zeros(freq(end)-1-length(logSpec),1);
```

```
logSpec=[logSpec;ZerosVect];
spec=[spec(1:floor((length(spec)+1)/2));ZerosVect];

for j=1:NbFilters
  Pcoeff1(j)=sum(filters(j,1:freq(j+2)-freq(j))'.*logSpec(freq(j):freq(j+2)-1));
  Pcoeff2(j)=sum(filters(j,1:freq(j+2)-freq(j))'.*abs(spec(freq(j):freq(j+2)-1)));
end

NbCoeffs=floor(NbFilters/2)+1;
MAX=max(10*log10(Pcoeff2));
for k=0:NbCoeffs-1

  coeffs1(k+1)=1/NbFilters*sum(Pcoeff1.*cos(2*pi*k/NbFilters*[0:NbFilters-
1]));

coeffs2(k+1)=1/NbFilters*sum((10*log10(Pcoeff2)-
MAX+50).*cos(2*pi*k/NbFilters*[0:NbFilters-1]));

end
```

Selanjutnya adalah proses pembentukan *database*. Proses *database* terdiri 2 proses yaitu *database codebook* dan *database Hidden Markov Model*.

3.4.2 Pembuatan Codebook

Pada proses pembentukan *database codebook*, data hasil ekstraksi akan digunakan untuk mencari *codebook*. Salah satu metode yang digunakan dalam proses pembuatan *codebook* yaitu dengan metode algoritma LBG. Proses algoritma LBG akan mencari *centroid* untuk menentukan *codeword* yang baru. Proses ini akan dilakukan berulang kali agar didapatkan nilai *distortion* yang terkecil. Prosedur yang digunakan disebut *nearest neighbour search.* Lisitng program membentuk *database codebook* adalah sebagai berikut.

```
function make_codebook(filename,M,iteration);
[names,speech]=crear_speech;
```

```
Code=VQ_training(speech,M,iteration);
eval(['save ' filename ' Code names']);
pause(2)
menu;
```

Ukuran *codebook* yang digunakan akan menggunakan ukuran 32. Iterasi pada metode GLA adalah sebanyak sepuluh kali. Selain itu perlu diketahui jumlah label yang akan dibuat *codebook*-nya. Hasil dari fungsi ini adalah matriks *Code* dengan ukuran 32 x 9. Tahapan dalam melakukan *training* HMM dapat dijelaskan sebagai berikut :

1. Dicari nilai *codebook* dengan *error splitting* 0.00000001
2. Dengan algoritma LBG dicari nilai distorsi yang paling kecil dengan prosedur *nearest neighbour search.*
3. Menyimpan *codebook* sesuai dengan nama yang telah ditentukan

3.4.3 *Membuat Hidden Markov Model*

Pada proses pembentukan *database Hidden Markov Model* data hasil ekstraksi akan digunakan untuk menentukan urutan observasi. Urutan observasi sesuai dengan pembagian hasil ekstraksi pada *codebook.* Urutan observasi akan digunakan untuk mencari matriks-matriks HMM. Lisitng program mencari urutan observasi adalah sebagai berikut.

```
function make_HMM(model_file,codebook,iteration)

load(codebook);
[M,a]=size(Code);
SAVING_OBSERVATION_new(codebook,'fileobservation');
TRAIN_PART_demo(model_file,'fileobservation',iteration,M);

pause(2)

menu;
```

Hasil dari *training* HMM adalah matrik-matrik *Ai, Bi* dan *p0i* untuk setiap *label(i)* (*i = 1, 2, 3, 4,5*).

dimana*Ai* : matrik transisi

B : Probabilitas matrik dari observasi

p0i : vektor dari *initial probability*

Setelah mengetahui urutan observasi akan ditentukan parameter HMM-nya. Proses menggunakan algoritma *Baum Welch.* Pada proses ini proses HMM dilakukan secara *foward* dan *backward.* Listing program membentuk *database* HMM adalah sebagai berikut.

HMM_BACKWARD.M COTENT

```
function beta=HMM_backward(A,B,p0,O,c);

 T=length(O);
[n,m]=size(B);
beta_temp=ones(n,1);
beta(:,T)=c(T)*beta_temp;

for t=(T-1):-1:1,

  beta_temp= A*(beta(:,t+1).*B(:,O(t+1)));

  beta(:,t)=c(t)*beta_temp;

end;
```

HMM_FORWARD.M COTENT

```
function [alfa,c]=HMM_forward(A,B,p0,O);

 T=length(O);
alfa_temp=p0.*B(:,O(1));
c(1)=1/sum(alfa_temp);
alfa(:,1)=c(1)*alfa_temp;
```

```
for t=2:T,

 alfa_temp=B(:,O(t)).*(A'*alfa(:,t-1));
 c(t)=1/sum(alfa_temp);
 alfa(:,t)=c(t)*alfa_temp;
end;
```

Jumlah iterasi yang paling optimum digunakan untuk algoritma *Baum Welch* adalah sepuluh [1]. Tahapan dalam melakukan *training* HMM dapat dijelaskan sebagai berikut :

1. Menentukan urutan observasi dari *file codebook.*
2. Inisialisasi matrik A, B dan ∏ dengan nilai acak.
3. Dengan algoritma *Baum Welch,* dilakukan pelatihan untuk menentukan nilai HMM yang sebenarnya.
4. Menghitung probabilitas observasi HMM untuk setiap data
5. Menyimpan nilai probabilitas HMM sesuai dengan nama yang telah ditentukan

Kedua *database* yang didapatkan akan digunakan pada proses pengenalan suara.

3.5 Algoritma Untuk Mengenali Suara *User*

Pengenalan suara akan dicoba pada MATLAB yang merupakan program yang sudah *user friendly.* Proses pada pengenalan suara mempunyai proses yang mirip dengan proses *training.* Akan tetapi pada proses pengenalan suara semuanya dilakukan pada MATLAB secara *real time.* Prosedur pengenalan suara akan dimulai dengan merekam suara *user* pada MATLAB. Suara *user* akan diproses oleh MATLAB. Hasil dari pengenalan suara akan ditunjukkan oleh pergerakan kendaraan. Urutan pergerakan kendaaan tersebut adalah :

Kendaraan bergerak Depan adalah “Depan”
Kendaraan bergerak Belakang adalah “Belakang”
Kendaraan bergerak ke Kanan adalah “Kanan”
Kendaraan bergerak ke Kiri adalah “Kiri”
Kendaraan berhenti bergerak adalah “Stop”

Proses pengenalan suara akan menangkap suara *user* hanya sekali. Suara diambil dengan *sampling* 11025 Hz selama 2 detik. Berarti akan didapat *sample* sebanyak 22050 data.

Selanjutnya data yang dimiliki akan diekstraksi untuk mendapatkan sinyal pada domain frekuensi. Sinyal akan dibagi dalam *frame-frame.* Pembagian *frame* sama dengan pada *training.* Satu *frame* terdiri dari 256 data dan terdapat 100 data suara pada satu *frame.* Pemberian data *overlapping* untuk mencegah terjadiya kesalahan pada proses ekstraksi.

Selanjutnya data yang telah dibagi dalam *frame* akan di-*windowing*. Pada proses ini koefisien *Hamming windowing* akan ditentukan terlebih dahulu. Kemudian Koefisien *Hamming windowing* dikalikan dengan masing-masing data pada satu *frame.*

Data yang telah di-*windowing* akan dicari FFT-nya. Hasil dari FFT akan dibagi dengan 1,000 agar tidak terlalu besar. Hasil dari FFT terlalu besar karena sinyal *input* memiliki level suara yang besar.

Selanjutnya adalah membuat vektor dari 150 mel frekuensi. Vektor ini akan digunakan untuk mendapatkan *mel cepstral coefficients.* Kemudian membuat matriks *filter bank*. Matriks ini dibuat berdasarkan matriks *trianggular.* Setelah itu menghitung *mel cepstral coefficients* . Hasil dari ekstraksi adalah matriks yang berisi *mel cepstral coefficients* dari tiap *frame.* Proses selanjutnya adalah menentukan urutan observasi dari sinyal suara yang akan dikenali. Matriks hasil ekstraksi akan dihitung distorsinya dengan *codebook* yang dimiliki. Apabila didapatkan distorsi yang paling kecil, maka dicari indeks dari *codebook* yang sesuai. Indeks yang didapatkan merupakan urutan observasi pada suara yang akan dikenali.

Proses selanjutnya adalah mencari nilai *probabilitas* dari suara yang akan dikenali menggunakan urutan observasi yang didapatkan. Prosedur HMM yang digunakan adalah *HMM foward.* Hasil yang didapatkan adalah sebuah urutan probabilitas dari urutan observasi sinyal yang akan dikenali. Selanjutnya urutan probabilitas tersebut dijumlah dan dicari hasil perhitungan *log*-nya.

Proses ini dilakukan untuk kelima sinyal yang telah dilatih ditambah dengan keadaan bahwa sinyal tersebut tidak dikenali sebagai suara apapun. Hasil

yang didapatkan adalah kelima *log of probability.* Kelima *log of probability* tersebut dibandingkan untuk mendapatkan kesimpulan suara yang dikenali. *Log of probability* yang paling besar merupakan sinyal yang dikenali.

Urutan proses yang terjadi adalah :

1. Suara diambil secara *real-time* ke MATLAB dengan durasi 2 detik dan frekuensi *sampling* 11025 Hz.
2. Hasilnya adalah 22050 *sample* data.
3. Nilai FFT dari suara akan dibandingkan dengan nilai *codeword* dan *database.*
4. Selanjutnya didapatkan urutan kode observasi yang diinginkan.
5. Dicari nilai dari HMM *probability* dan dibandingkan dengan nilai HMM *probability label lainnya.*
6. Nilai *log of probability* paling besar merupakan suara yang dikenali

Suara yang dikenali akan ditunjukkan oleh kendaraan yang bergerak. Hasil pengidentifikasian memberikan langkah-langkah selanjutnya. Langkah-langkah selanjutnya tergantung dari *output* yang diinginkan sesuai dengan identitas *user*. Kemampuan mengenali tergantung antara alat pengenalnya. Sehingga dapat dilakukan penambahan pada alat pengenalnya.

BAB IV
HASIL UJI COBA DAN ANALISA

Salah satu tujuan pengujian ini akan mencoba mengenali suara manusia yang sebelumnya telah tersimpan dalam database dengan tujuan untuk menggerakkan miniatur robot mobil dengan lima jenis suara. Adapun blok-blok yang akan diuji adalah :

1. Pengujian program pengenalan suara (*Speech Recognition)*.
2. Pengujian perangkat komunikasi
3. Pengujian pergerakan model kendaraan.
4. Pengujian alat secara keseluruhan.

Dalam pengambilan data dilakukan dalam beberapa tahap yaitu pengamatan tampilan pada program MATLAB, pengamatan melalui LCD dan pengamatan pada pergerakan model kendaraan.

4.1 Pengujian Program Pengenalan Suara *(Speech Recognition)*

1. Tujuan : Mengamati dan mengetahui nilai keakurasian dari program pengenalan suara yang telah dibuat dengan menggunakan program MATLAB.
2. Jalan percobaan : *User* mengucapkan kelima kata kunci yang telah disimpan di dalam database MATLAB sebanyak 10 kali dengan menekan tombol “Get Signal”. Kata kunci yang diucapkan oleh pengguna akan ditampilkan di layar komputer oleh program MATLAB.

3. Hasil Pengujian :

Tabel 4.1 Hasil Uji coba dengan repetisi 10 kali

No.	Depan	Belakang	Kiri	Kanan	Stop
	Teridentifikasi				
1	Belakang	Belakang	Kiri	Kanan	Stop
2	Depan	Depan	Depan	Kanan	Depan
3	Depan	Belakang	Kiri	Kanan	Stop
4	Kiri	Belakang	Kiri	Kiri	Stop
5	Depan	Belakang	Kanan	Kanan	Stop
6	Depan	Belakang	Depan	Kiri	Kiri
7	Depan	Belakang	Kiri	Depan	Stop
8	Depan	Belakang	Kiri	Kanan	Stop
9	Depan	Belakang	Stop	Kanan	Stop
10	Kanan	Belakang	Kiri	Depan	Stop

Ket: warna biru berarti keluaran tidak sesuai dengan perintah

Hasil pengujian : Tabel 4.2 Akurasi per label

Kata kunci	Jumlah Pengujian	Akurasi
Depan	10	70 %
Belakang	10	90 %
Kiri	10	60 %
Kanan	10	60 %
Stop	10	80 %

4. Analisa : dari hasil uji coba dapat diambil kesimpulan bahwa keakuratan pengenalan kata kunci dari program MATLAB yang telah dibuat adalah cukup baik karena semua nilai akurasi berada di atas 50%.

4.2 Pengujian Perangkat Komunikasi

1. Tujuan : untuk mengetahui tingkat keakuratan sistem perangkat komunikasi dalam proses pengiriman data menggunakan bluetooth.
2. Jalannya Ujicoba : Perangkat bluetooth melakukan komunikasi dengan menggunakan Hyper Terminal dengan setingan port sebagai berikut :

- Port : COM22
- Baudrate : 9600 bps
- Data Bits : 8
- Parity : none
- Stop Bits : 1
- Flow Control : none

Sedangkan setingan bluetooth pada model kendaraan adalah sebagai berikut :

- Port : COM22
- Baudrate : 9600 bps
- Data Bits : 8
- Parity : none
- Stop Bits : 1
- Flow Control : none

Kondisi pengujian adalah sebagai berikut: jarak yang digunakan adalah kurang lebih 10m, antara kedua device tersebut tidak ada penghalang.

Metode pengujian pengguna akan melakukan pengiriman data melalui Hyper Terminal dan pengujian langsung pada model kendaraan dengan variasi jarak. Data yang akan dikirim adalah sebagai berikut :

- 123456789, yang akan dilakukan pengujian sebanyak 5 kali.

- abcdefghi, yang akan dilakukan pengujian sebanyak 5 kali.
- ABCDEFGHI, yang akan dilakukan pengujian sebanyak 5 kali.
- 000011110, yang akan dilakukan pengujian sebanyak 5 kali.
- Pengujian dengan variasi jarak pada model kendaraan.

3. Hasil Pengujian :

Tabel 4.3 Hasil Pengujian Perangkat Komunikasi

Jumlah Pengujian	Data yang dikirim	Data tampilan LCD
5 kali	123456789	123456789
5 kali	abcdefghi	Abcdefghi
5 kali	ABCDEFGHI	ABCDEFGHI
5 kali	000011110	000011110

Tabel 4.4 Hasil Pengujian Perangkat Komunikasi Dengan Variasi Jarak

Uji Coba Bluetooth	
Jarak (Meter)	Tingkat Keberhasilan
1	Baik
2	Baik
3	Baik
4	Baik
5	Baik
6	Baik
7	Baik
8	Baik
9	Baik
10	Baik

4. Analisa : Dari hasil pengamatan dapat diambil kesimpulan bahwa tingkat keakuratan pengiriman data dari perangkat komunikasi yang digunakan adalah sangat baik.

4.3 Pengujian Pergerakan Model Kendaraan

1. Tujuan : Untuk mengetahui pergerakan robot mobil.
2. Jalan percobaan : Pengguna mengirimkan perintah kepada model kendaraan dengan cara mengetikkan perintah tersebut melalui program Hyper Terminal, dengan perintah-perintah sebagai berikut:
   - "1" untuk perintah maju
   - "2" untuk perintah mundur
   - "3" untuk perintah kiri
   - "4" untuk perintah kanan
   - "5" untuk perintah berhenti

   Semua perintah dilakukan sebanyak 5 kali

3. Hasil Pengujian :

Tabel 4.5 Hasil Pengujian Pergerakan Model Kendaraan

Jumlah Pengujian	Data yang dikirim	Pergerakan model kendaraan
5 kali	1	Maju
5 kali	2	Mundur
5 kali	3	Belok kanan
5 kali	4	Belok kiri
5 kali	5	Berhenti

4. Analisa : Dari hasil pengamatan dapat diambil kesimpulan bahwa pergerakan model kendaraan adalah baik.

4.4 Pengujian Alat Secara Keseluruhan

1. Tujuan : Mengamati atau mengetahui hasil uji coba secara keseluruhan.
2. Alur percobaan : Pengguna mengucapkan kelima kata kunci yang telah disimpan di dalam database MATLAB sebanyak 10 kali .dengan menekan tombol "Voice Recognition". Kata kunci yang

diucapkan oleh pengguna akan ditunjukkan dari pergerakan model kendaraan.

3. Hasil Pengujian :

Tabel 4.6 Hasil Uji coba keseluruhan dengan repetisi 10 kali

	Depan	Belakang	Kiri	Kanan	Stop
	Teridentifikasi				
1	Belakang	Mundur	Depan	Kanan	Depan
2	Kiri	Maju	Mundur	Stop	Berhenti
3	Maju	Mundur	Kiri	Maju	Kiri
4	Maju	Mundur	Kiri	Kanan	Berhenti
5	Maju	Mundur	Belakang	Kanan	Berhenti
6	Maju	Mundur	Kanan	Mundur	Berhenti
7	Belakang	Mundur	Depan	Kanan	Depan
8	Kiri	Mundur	Kiri	Kanan	Berhenti
9	Maju	Mundur	Kiri	Kanan	Berhenti
10	Maju	Maju	Kiri	Maju	Berhenti

Ket: warna biru berarti keluaran tidak sesuai dengan perintah

Setelah seluruh uji coba dapat dihitung persen akurasi keseluruhan dari sistem.

Tabel 4.7 Hasil akurasi keseluruhan sistem

Repetisi	*Codebook*	Akurasi
10	32	64 %

Sedang untuk hasil akurasi tiap label diperlihatkan pada Tabel 4.8 :

Hasil pengujian : Tabel 4.8 Akurasi keseluruhan per label

Kata kunci	Jumlah Pengujian	Akurasi
Depan	10	60 %
Belakang	10	80 %
Kiri	10	50 %
Kanan	10	60 %

Stop	10	70 %

4. Analisa : Berdasarkan pengamatan dapat diambil kesimpulan bahwa keakuratan pengenalan kata kunci dari program MATLAB dan ketepatan dari pergerakan model kendaraan yang telah dibuat adalah cukup baik karena semua nilai akurasi berada di atas 50%.

BAB 5
SARAN DAN KESIMPULAN

5.1 Kesimpulan

Berdasarkan pada hasil pengujian dan analisa terhadap hasil yang didapatkan, maka dapat di diambil suatu hasil kesimpulan yaitu :

- Pengoperasian sistem kontrol penggerak miniatur robot mobil dapat dilakukan dengan menggunakan perintah suara manusia.
- Metode HMM dapat digunakan untuk aplikasi speech recognition.
- Sistem dependent speaker memiliki tingkat akurasi yang tinggi dalam pengenalan sistem suara yang akan diuji kebenaranya.
- Pengintegrasian sistem yang dibuat belum menunjukkan dari hasil yang diinginkan.
- Kesalahan pengenalan yang terjadi diakibatkan adanya perbedaan yang terlalu besar antara sinyal suara yang hendak dikenali dengan sinyal suara yang dilatihkan, hal ini dapat diatasi dengan menambahkan/memperbanyak berbagai variasi pola kata pada saat pelatihan dengan demikian sistem jaringan lebih diperkaya pengetahuannya.

5.2 Saran

Untuk mendapatkan hasil yang lebih baik dikemudian hari dapat mempertimbangkan saran sebagai berikut dalam pendesainan alat selanjutnya:

- Suara masukan yang diucapkan oleh pengucap harus memiliki intonasi yang sama dengan standartd yang ada dalam database
- Pengurangan noise harus ditingkatkan dengan memperhatikan sistem secara keseluruhan agar pengambila data awal dan akhir menjadi akurat.

DAFTAR ACUAN

[1] B. T. Tan and M. Fu. “coustic-phonetic analysis of the speech signals”. Internal report. Department of Electrical and Computer Engineering, University of Newcastle, Australia, 1994.

[2] Dmitri Asonov, Rakesh Agrawal: Keyboard Acoustic Emanations. Proceedings of the 2004 IEEE Symposium on Security and Privacy, 2004.

[3] Arry Akhmad Arman, ”Teknologi Pemrosesan Bahasa Alami sebagai Teknologi Kunci untuk Meningkatkan Cara Interaksi antara Manusia dengan Mesin”. Fakultas Teknologi Industri – ITB, 23 Agustus 2004

[4] Li Zhuang, Feng Zhou, J.D.Tygar: Keyboard Acoustic Emanations Revisited. To appear in Proceedings of the 12th ACM Conference on Computer and Communications Security, November 2005.

[5] Microsoft Corporation, Cambridge University Engineering Department, *HTK Book* (2001-2005). htkbook.pdf.

[6] C. Karlof, *et al.*, *Hidden MarkovModel Cryptanalysis*, Department of Computer Science, niversity of California, Berkeley, USA, 2003.

[7] *http://htk.eng.cam.ac.uk/* Juni 2007.

[8] *http://www.relisoft.com/Science/Physics/fft.html.*

[9] *http://en.wikipedia.org/wiki/Fast_Fourier_Transform.*

[10] *http://www.speech.cs.smu.edu/comp.speech/Section6/ Q6.6.html#intro.*

[11] Huda Miftahul, Bima, “Pelatihan Tcl/Snack”, PENS – ITS, Surabaya, 2005.

[12] Ambardar Ashok, “Analog and Digital Signal Processing – 2nd ed”, Brooks/Cole Publising Company, 1999.

[13] John. G. Proakis, Dimitris. G. Monolakis, “Digital Signal Processing: principles, algorithms, and application”, Prentice Hall, Inc, New Jersey, 1995.

[14] Lawrence Rabiner, Biing Hwang Juan, “Fundamentals of Speech Recognition”, Prentice Hall International Inc, 1993.

[15] Sadaoki Furui. “Digital Speech Processing Synthesis, and Recognition”, Marcell Dekker, Inc, New York, 1989.

[16] Datasheets *“Mikrokontroller AT89S52”*

[17] Datasheets *“Driver Motor L2898N”*

[18] Datasheets *“Bluetooth EB 500 Series”*

LAMPIRAN

EXTRATION.M

```
function F=extraction(signal,length_frame,overlap,fs)

% F=extraction(signal,length_frame,overlap,fs)
% fs: frequency sample

fprintf('Extracting Features\n');
nmax_frame=floor(length(signal)/length_frame);
sp=[zeros(1,overlap) signal' zeros(1,overlap)]';
frame=0;
for i=overlap+1:length_frame:nmax_frame*length_frame,
   frame=frame+1;
   sframe=sp((i-overlap):(i+length_frame-1+overlap));

[spec,logSpec,f,Pcoeff1,F1(frame,:),F2(frame,:)]=mfcc3(sframe.*ham
ming(length(sframe)),fs);
end;
fprintf('Done!\n');
F=F2;
```

HMM_FORWARD.M

```
function [alfa,c]=HMM_forward(A,B,p0,O);

% A transition matrix (n,n)
% B the observation symbol probability matrix (n,m)
% p0 initial state distribution vector [n,1]
% O observation sequence vector (1,T)
% c the scaling variable, vector (1,T)
% alfa the scaled alfa variable, matrix (n,T)

T=length(O);
%t=1
alfa_temp=p0.*B(:,O(1));
c(1)=1/sum(alfa_temp);
alfa(:,1)=c(1)*alfa_temp;

for t=2:T,
  alfa_temp=B(:,O(t)).*(A'*alfa(:,t-1));
  c(t)=1/sum(alfa_temp);
  alfa(:,t)=c(t)*alfa_temp;
end;
```

HMM_PROBABILITY.M

```
function logp=HMM_probability(c);

% c, the scaling variable, vector (1,T)
% logp the logarithm of the probability (1,1)
logp=-sum(log(c));
```

MFCC3.M

```
function [spec,logSpec,f,Pcoeff1,coeffs1,coeffs2]=mfcc3(signal,fs)
%function [spec,logSpec,f,Pcoeff1,coeffs1,coeffs2]=mfcc(signal,fs)

% function [ coeffs1, coeffs2 ] = mfcc ( signal, fs )
%
% This function computes the mel frequency cepstral coefficients
(mfcc) of a signal
% with 2 methods.
%
% Parameters :
% Input : - signal, fs : the signal and its sampling frequency.
% Output : - coeffs1 : mfcc coefficients calculated by taking
%                      P1(i)=sum(log|S(k)|*filterBank(k)) as
power coefficients
%                      and 1/N*sum(P(i)*cos(2*pi*k*n/N)) as mfcc
coefficients
%          - coeffs2 : mfcc coefficients calculated by taking
%                      P1(i)=sum(|S(k)|*filterBank(k)) as power
coefficients
%                      and 1/N*sum(log(P(i))*cos(2*pi*k*n/N)) as
mfcc coefficients
%

% Spectrum (fft) of the signal
spec=fft(signal);
logSpec=log(abs(spec(1:floor((length(spec)+1)/2))));
nbpts=length(logSpec);

% Creation of the filter bank
%% creation of a vector of 150 mels spaced frequencies
i=1;
freq(i)=1;
f(i)=1;
while freq(i)<nbpts
   i=i+1;
   f(i)=(exp(log(2)*(150*(i-1))/1000)-1)*1000;
   freq(i)=floor((exp(log(2)*(150*(i-1))/1000)-
1)*1000*2/fs*nbpts);
end
freq(i+1)=floor((exp(log(2)*(150*i)/1000)-1)*1000*2/fs*nbpts);

%% creation of the triangular filters
NbFilters=length(freq)-2;
filters=zeros(NbFilters,freq(end)-freq(end-1));
for k=1:NbFilters
   filters(k,1:freq(k+2)-freq(k))=triang(freq(k+2)-freq(k))';
end


% Adding zeros to the spectrum to match the filter bank's length
ZerosVect=zeros(freq(end)-1-length(logSpec),1);
logSpec=[logSpec;ZerosVect];
spec=[spec(1:floor((length(spec)+1)/2));ZerosVect];

% Computation of the power coefficients
for j=1:NbFilters
```

```
   Pcoeff1(j)=sum(filters(j,1:freq(j+2)-
freq(j))'.*logSpec(freq(j):freq(j+2)-1));
   Pcoeff2(j)=sum(filters(j,1:freq(j+2)-
freq(j))'.*abs(spec(freq(j):freq(j+2)-1)));
end

% Computation of the mel cepstral coefficients
NbCoeffs=floor(NbFilters/2)+1;
MAX=max(10*log10(Pcoeff2));
for k=0:NbCoeffs-1

coeffs1(k+1)=1/NbFilters*sum(Pcoeff1.*cos(2*pi*k/NbFilters*[0:NbFi
lters-1]));
   coeffs2(k+1)=1/NbFilters*sum((10*log10(Pcoeff2)-
MAX+50).*cos(2*pi*k/NbFilters*[0:NbFilters-1]));
end
```

Voice Recognition . M

```
function varargout = voice_recognition(varargin)
% VOICE_RECOGNITION M-file for voice_recognition.fig
%      VOICE_RECOGNITION, by itself, creates a new
VOICE_RECOGNITION or raises the existing
%      singleton*.
%
%      H = VOICE_RECOGNITION returns the handle to a new
VOICE_RECOGNITION or the handle to
%      the existing singleton*.
%
%      VOICE_RECOGNITION('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in VOICE_RECOGNITION.M with the
given input arguments.
%
%      VOICE_RECOGNITION('Property','Value',...) creates a new
VOICE_RECOGNITION or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before voice_recognition_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to
voice_recognition_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
voice_recognition

% Last Modified by GUIDE v2.5 19-Aug-2008 18:01:19

% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
@voice_recognition_OpeningFcn, ...
                   'gui_OutputFcn',  @voice_recognition_OutputFcn,
...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before voice_recognition is made visible.
function voice_recognition_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to voice_recognition (see
VARARGIN)
% Choose default command line output for voice_recognition
handles.output = hObject;



% Update handles structure
guidata(hObject, handles);

% UIWAIT makes voice_recognition wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = voice_recognition_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
% --- Executes on button press in recognition_pushbutton.
function recognition_pushbutton_Callback(hObject, eventdata,
handles)
% hObject    handle to recognition_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

AI = analoginput('winsound');

%2. Add channels - Add one channel to AI.
chan = addchannel(AI,1);

%3. Configure property values - Assign values to the basic setup
properties, and
%create the variables blocksize and Fs, which are used for
subsequent analysis.
%The actual sampling rate is retrieved since it may be set by the
engine to a
%value that differs from the specified value.
duration = 2; %2 second acquisition
set(AI,'SampleRate',11025);
ActualRate = get(AI,'SampleRate');
set(AI,'SamplesPerTrigger',duration*ActualRate);
set(AI,'TriggerType','Manual');
blocksize = get(AI,'SamplesPerTrigger');
Fs = ActualRate;
t=0:(1/Fs):duration-(1/Fs);


%4. Acquire data - Start AI, issue a manual trigger, and extract
all data from
%the engine. Before trigger is issued, you should begin inputting
data from the
%tuning fork into the sound card.
start(AI);
trigger(AI);
signal = getdata(AI);

axes(handles.signal_axes);
plot(t,signal);
grid on;
ylabel('Magnitude');
xlabel('Time (Sec)');
title('Signal Components');

%5. Clean up - When you no longer need AI, you should remove it
from memory and
%from the MATLAB workspace.
delete(AI);
clear AI;


model = 'model_DSP';
book = 'DSP_codebook';
```

```
fs=11025;

%signal extraction
eval(['load' ' ' book]);
F=extraction(signal,100,78,fs);
%VQ
[O,Y,Distortion]=vq(F,Code);
%decision
eval(['load' ' ' model]);
%model 1
number=5-available;
P=-inf*ones(1,5);
if number>=1
[alfa,c]=HMM_forward(A1,B1,p01,O);
P(1)=HMM_probability(c);
end;
%model 2
if number>=2
[alfa,c]=HMM_forward(A2,B2,p02,O);
P(2)=HMM_probability(c);
end;
%model 3
if number>=3
[alfa,c]=HMM_forward(A3,B3,p03,O);
P(3)=HMM_probability(c);
end;
%model 4
if number>=4
[alfa,c]=HMM_forward(A4,B4,p04,O);
P(4)=HMM_probability(c);
end;
%model 5
if number>=5
[alfa,c]=HMM_forward(A5,B5,p05,O);
P(5)=HMM_probability(c);
end;
%choose the maximum
[val,index]=max(P);
measure=val-mean(P);
[Y,I]=sort(-P);
fprintf('Label   Log of Probability\n')
fprintf('-------------------------\n')
for j=1:5,
   fprintf(' %i      %f\n',I(j), -Y(j));

end;

fprintf('\n');
nam=names{index};
fprintf('The voice is %s\n',nam);

namstring=findobj(allchild(0),'Tag','wordtext');
set(handles.command_text,'String',nam(3:length(nam)));

fprintf(handles.ser,[nam(1)]);
```

```
% --- Executes during object creation, after setting all
properties.
function recognition_pushbutton_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to recognition_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called


% ------------------------------------------------------------
----
function file_menu_Callback(hObject, eventdata, handles)
% hObject    handle to file_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)


% ------------------------------------------------------------
----
function bluetooth_menu_Callback(hObject, eventdata, handles)
% hObject    handle to bluetooth_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% ------------------------------------------------------------
----
function setting_bluetooth_menu_Callback(hObject, eventdata,
handles)
% hObject    handle to setting_bluetooth_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
connection = instrfind();
if (length(connection) ~= 0 )
    delete(connection);
    clear connection;
end;
port = 'COM22';
ser = serial(port);
fopen(ser);

handles.ser = ser;

guidata(hObject,handles);

% ------------------------------------------------------------
----
function exit_menu_Callback(hObject, eventdata, handles)
% hObject    handle to exit_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% --- Executes on button press in stop_pushbutton.
function stop_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to stop_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

fprintf(handles.ser,['5']);



% --- Executes during object creation, after setting all
properties.
function stop_pushbutton_CreateFcn(hObject, eventdata, handles)
% hObject    handle to stop_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

VQ.M

```
function [O,Y,Distortion]=vq(X,C)
% [O,Y,Distortion]=vq(X,C);
% O :observation sequences discrete values
% Y :quantized values
% Distortion: Average Distortion
% X NxL vectored signal(one row of the matrix is L features of the
one frame)
% C MxL codebook
[N,L]=size(X);
[M,L]=size(C);
Distortion=0;
for t=1:N,
    for c=1:M,
        dist(c)=norm(X(t,:)-C(c,:)); %you can use only the square
distance without use square root
    end;
    [val,O(t)]=min(dist);
    Distortion=Distortion+val^2;
    Y(t,:)=C(O(t),:);
end;
Distortion=Distortion/N;
```

Program Peegerakan Motor

```
$crystal = 11059200
$baud = 9600

Dim A1 As Byte
Dim A2 As Byte

Wait 1
Reset P3.3                              'EB500-MODE
Waitms 250
Print "dis" ; Chr(13);
Bitwait P3.2 , Reset                    'EB500-STATUS
Print "set name dance" ; Chr(13);
Waitms 250
Print "set security off" ; Chr(13);
Waitms 250
Print "con 11:11:11:11:11:11" ; Chr(13);
Bitwait P3.2 , Reset                    'EB500-STATUS
Print "xxx";

Do
A2 = Waitkey
Select Case A2
Case "1":                               ' depan
Set P1.2                                ' kanan
Reset P1.3

Do
Loop Until P3.6 = 0 Or P3.5 = 0
If P3.6 = 0 Then
Reset P1.2                              ' kiri
Set P1.3

Do
Loop Until P3.5 = 0
Set P1.2                                ' stop kiri kanan
Set P1.3
Elseif P3.5 = 0 Then
Set P1.2                                ' stop kiri kanan
Set P1.3
End If


Reset P1.0                              ' maju
Set P1.1
Wait 3
Set P1.0
Set P1.1
Case "2":                               ' belakang
Set P1.2                                ' kanan
Reset P1.3

Do
Loop Until P3.6 = 0 Or P3.5 = 0
If P3.6 = 0 Then
Reset P1.2                              ' kiri
Set P1.3
```

```
Do
Loop Until P3.5 = 0
Set P1.2                        ' stop kiri kanan
Set P1.3
Elseif P3.5 = 0 Then
Set P1.2                        ' stop kiri kanan
Set P1.3
End If


Set P1.0                         ' mundur
Reset P1.1
Wait 3
Set P1.0                         ' stop
Set P1.1
Case "3":                        ' kanan
Set P1.2                         ' kanan
Reset P1.3

Do
Loop Until P3.6 = 0
Set P1.2                         ' stop kiri kanan
Set P1.3

Reset P1.0                       ' maju
Set P1.1
Wait 3
Set P1.0                         ' stop
Set P1.1
Case "4":                        ' kiri
Reset P1.2                       ' kiri
Set P1.3

Do
Loop Until P3.4 = 0
Set P1.2                         ' stop kiri kanan
Set P1.3

Reset P1.0                       ' maju
Set P1.1
Wait 3
Set P1.0                         ' stop
Set P1.1
Case "5":                        ' stop
Set P1.0                         ' stop
Set P1.1
Set P1.2                         ' kanan
Reset P1.3

Do
Loop Until P3.6 = 0 Or P3.5 = 0
If P3.6 = 0 Then
Reset P1.2                      ' kiri
Set P1.3
Do
Loop Until P3.5 = 0
```

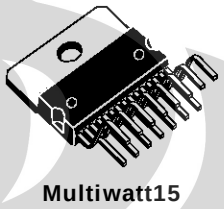
```
Set P1.2                    ' stop kiri kanan
Set P1.3
Elseif P3.5 = 0 Then
Set P1.2                    ' stop kiri kanan
Set P1.3
End If
End Select
Loop

End
```

SGS-THOMSON MICROELECTRONICS

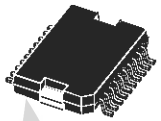
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



Multiwatt15



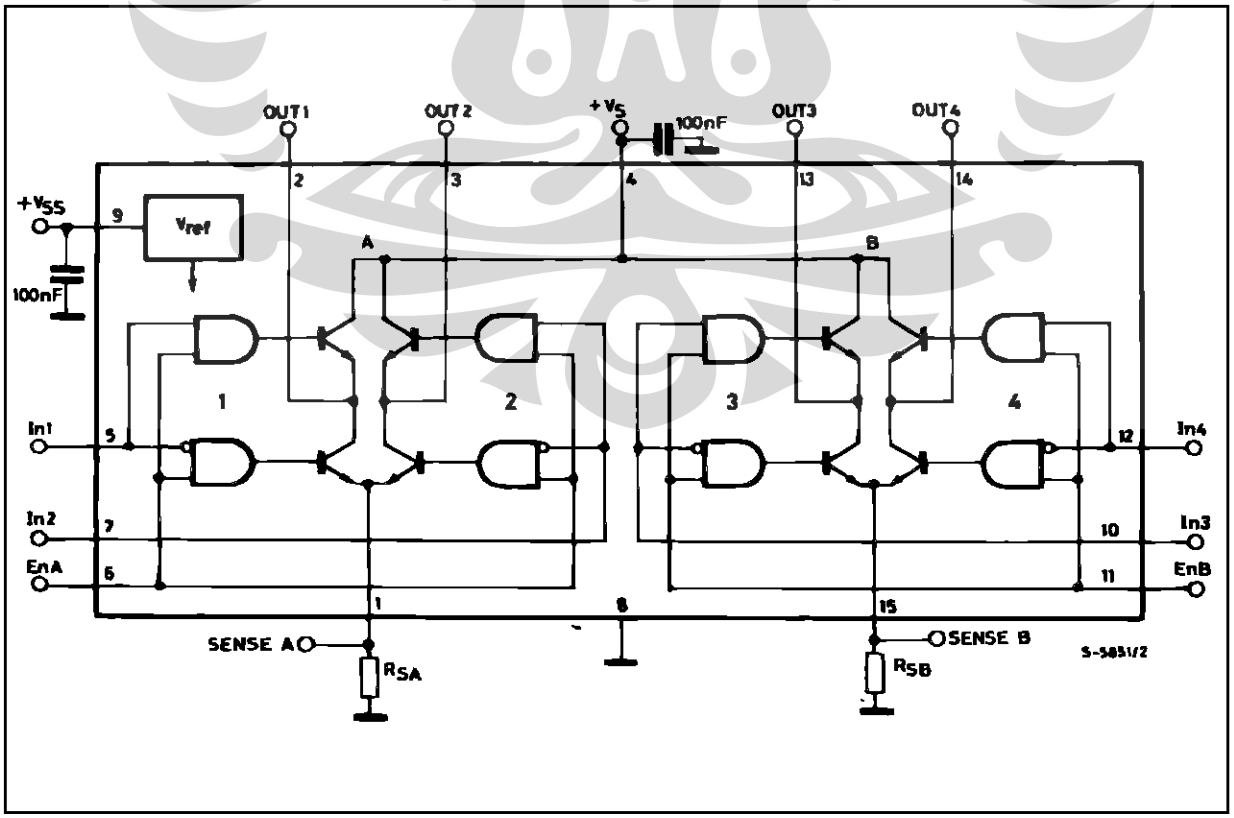
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

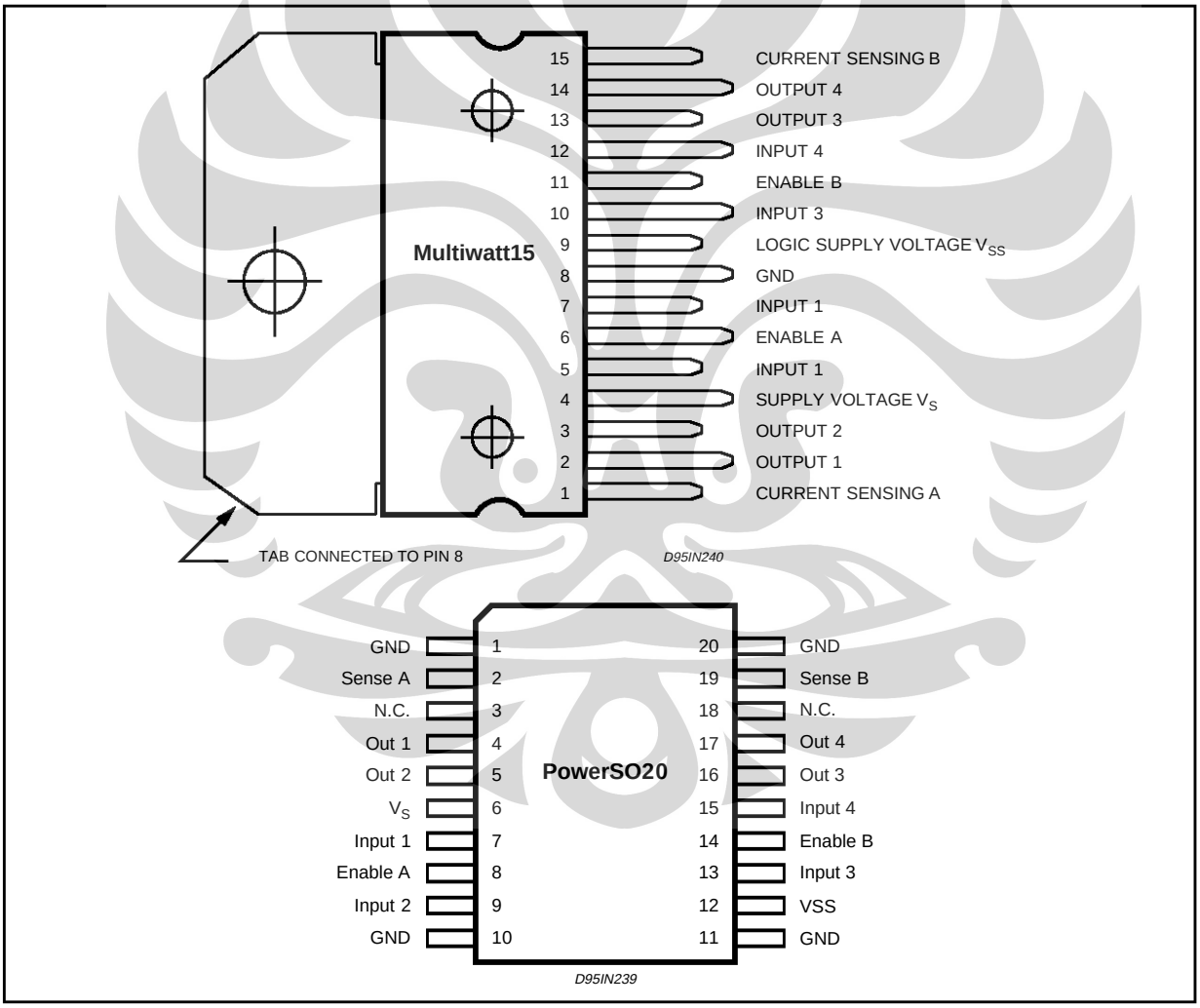
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition		V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)			4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0	V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L	V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0	V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L	V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)			–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)			2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L				–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V			30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)			–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)			2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L				–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V			30	100	µA
$V_{CEsat\ (H)}$	Source Saturation Voltage	I_L = 1A I_L = 2A			1.35 2	1.7 2.7	V V
$V_{CEsat\ (L)}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)			1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)				3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)			–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		µs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		µs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		µs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		µs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		µs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		µs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		µs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		µs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		µs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		µs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		µs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		µs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		µs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		µs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		µs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		µs
fc (V_{en})	Commutation Frequency	I_L = 2A		1		KHz

1) 1)Sensing voltage can be –1 V for t ≤ 50 µsec; in steady state V_{sens} min ≥ – 0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.
5) PIN 1 and PIN 15 connected to GND.

Figure 1 : Typical Saturation Voltage vs. Output Current.

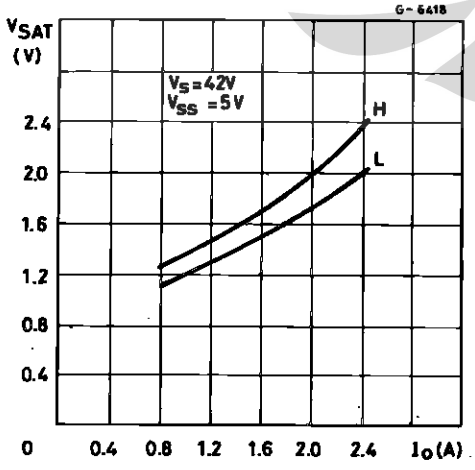
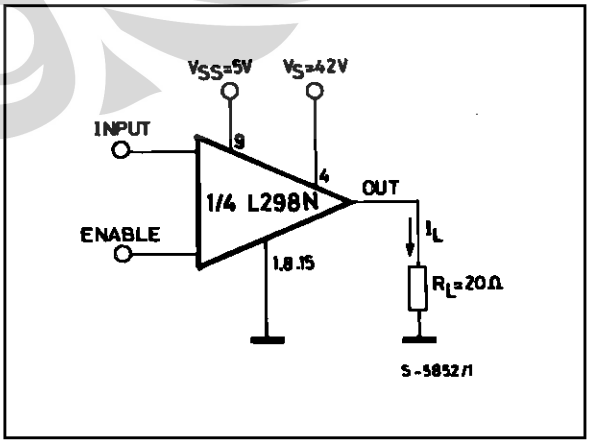


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

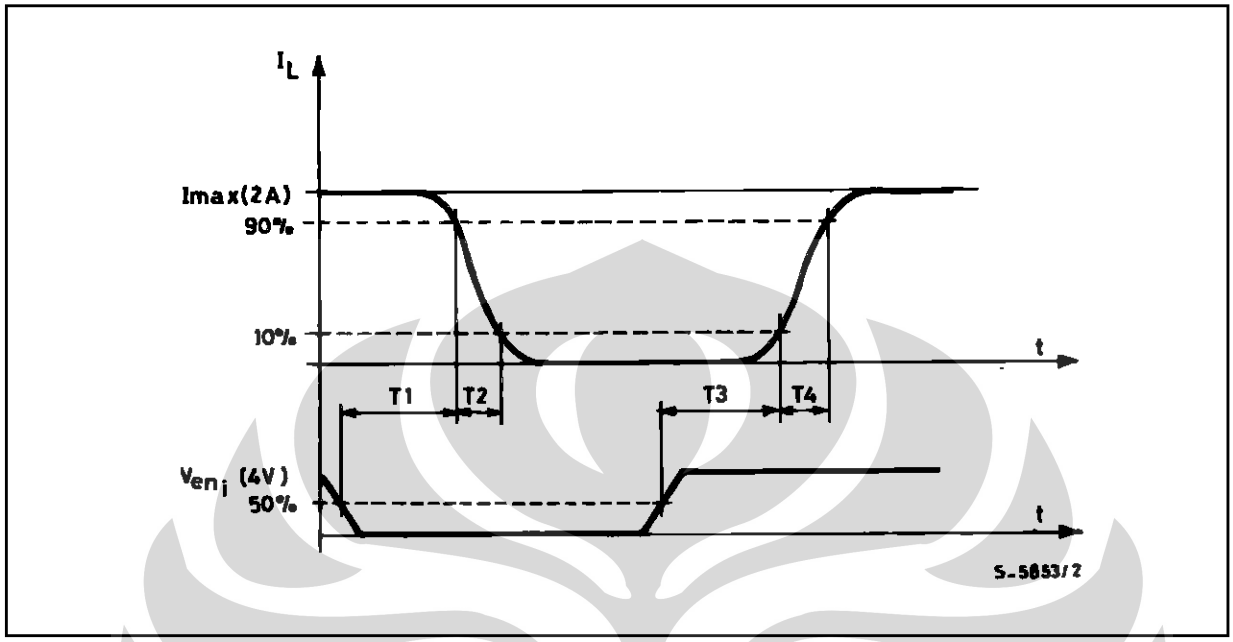
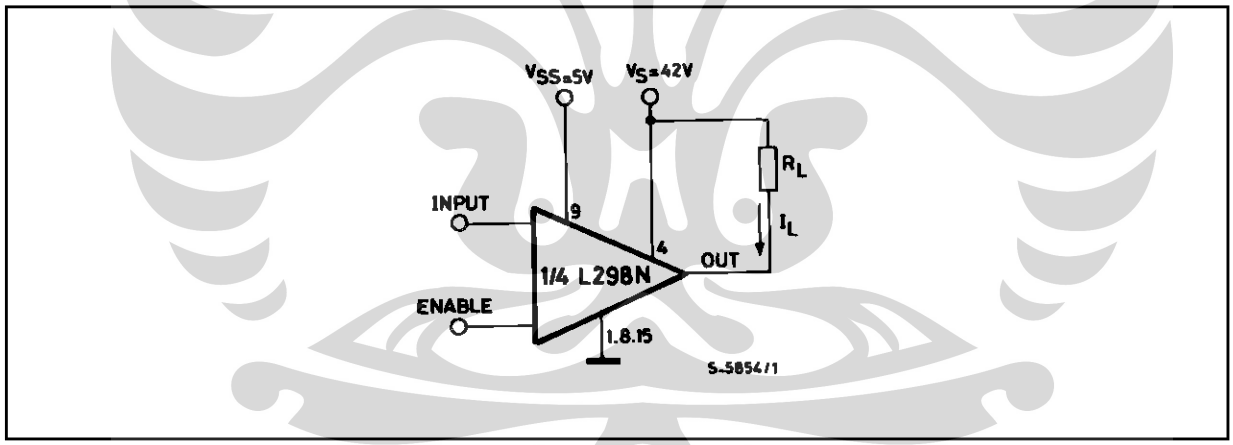


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

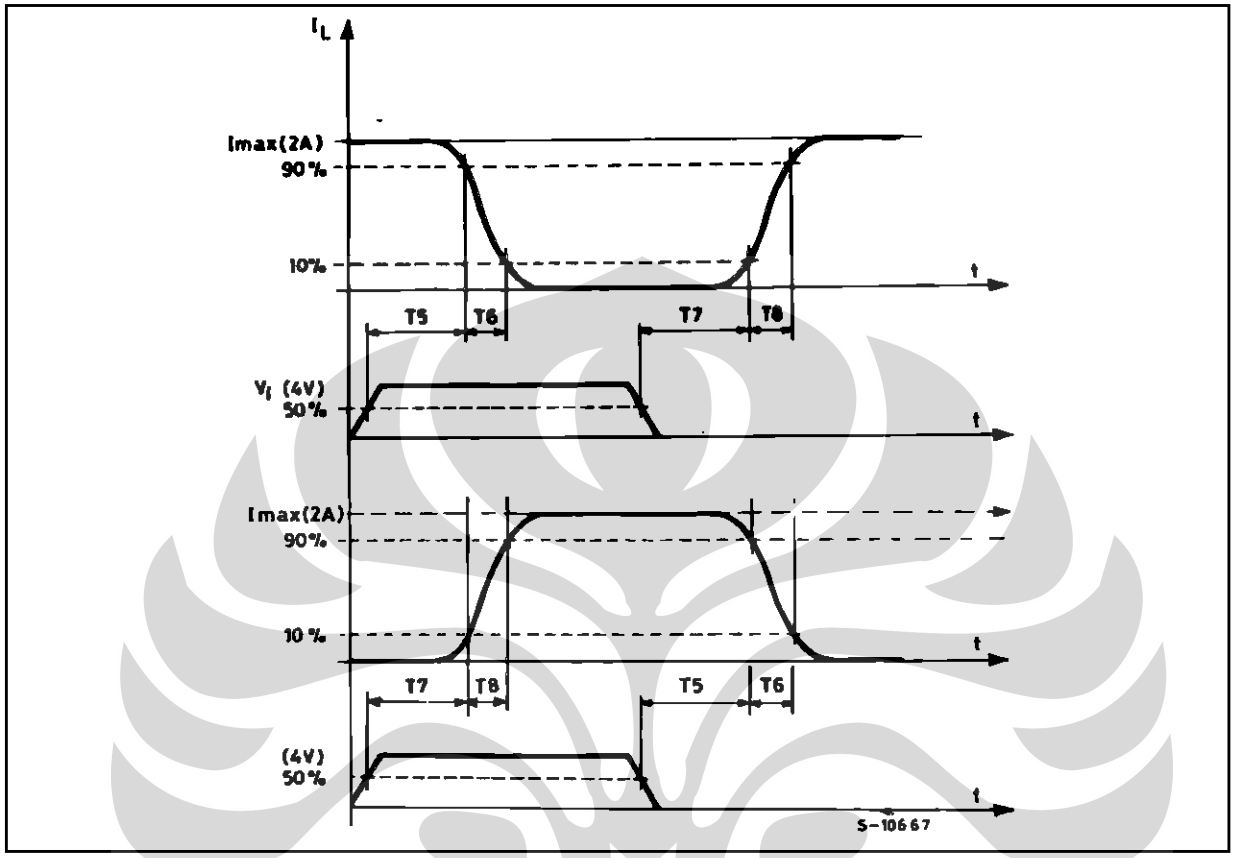
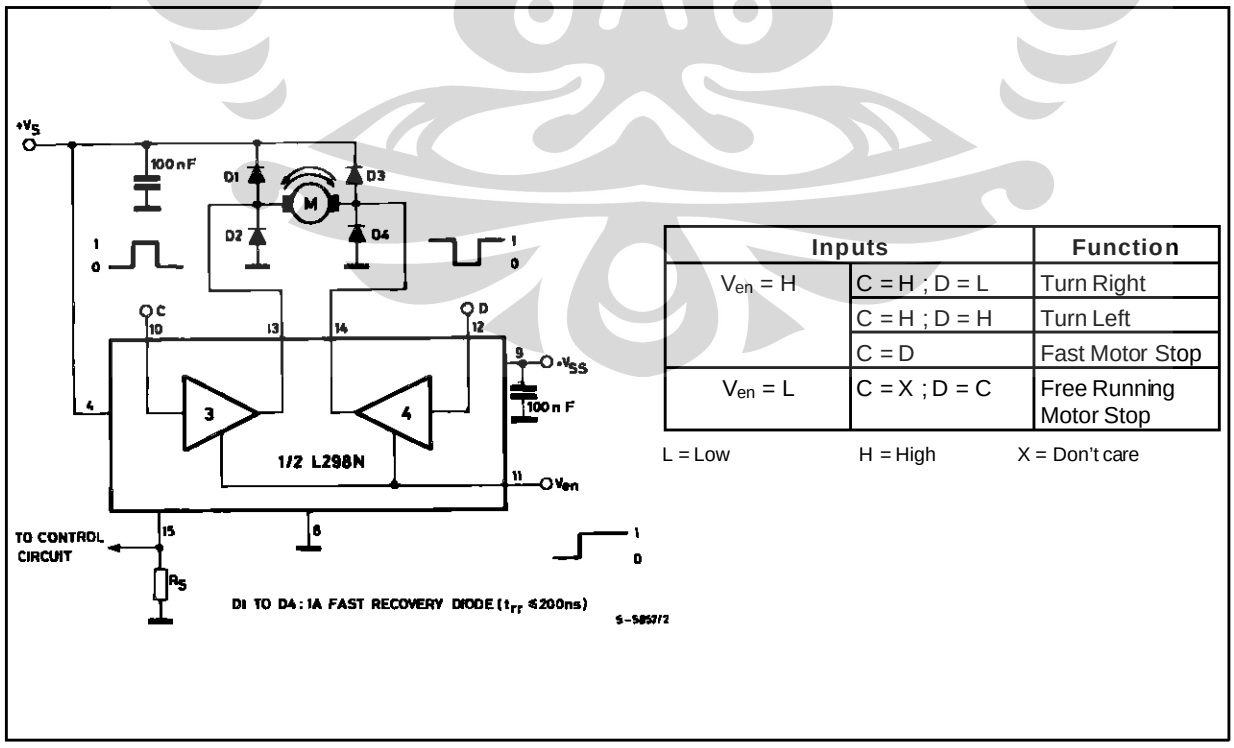


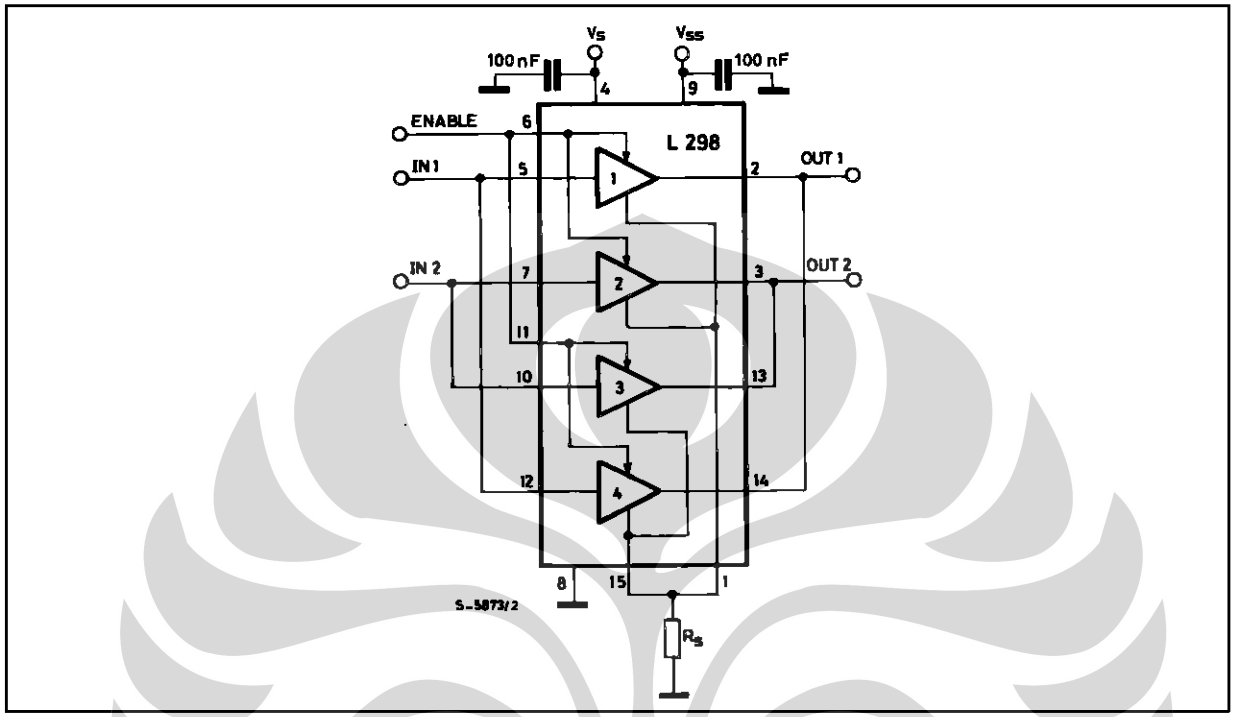
Figure 6 : Bidirectional DC Motor Control.



Inputs		Function
V_{en} = H	C = H ; D = L	Turn Right
	C = H ; D = H	Turn Left
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = C	Free Running Motor Stop

L = Low H = High X = Don't care

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}.) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements (trr ≤ 200 nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

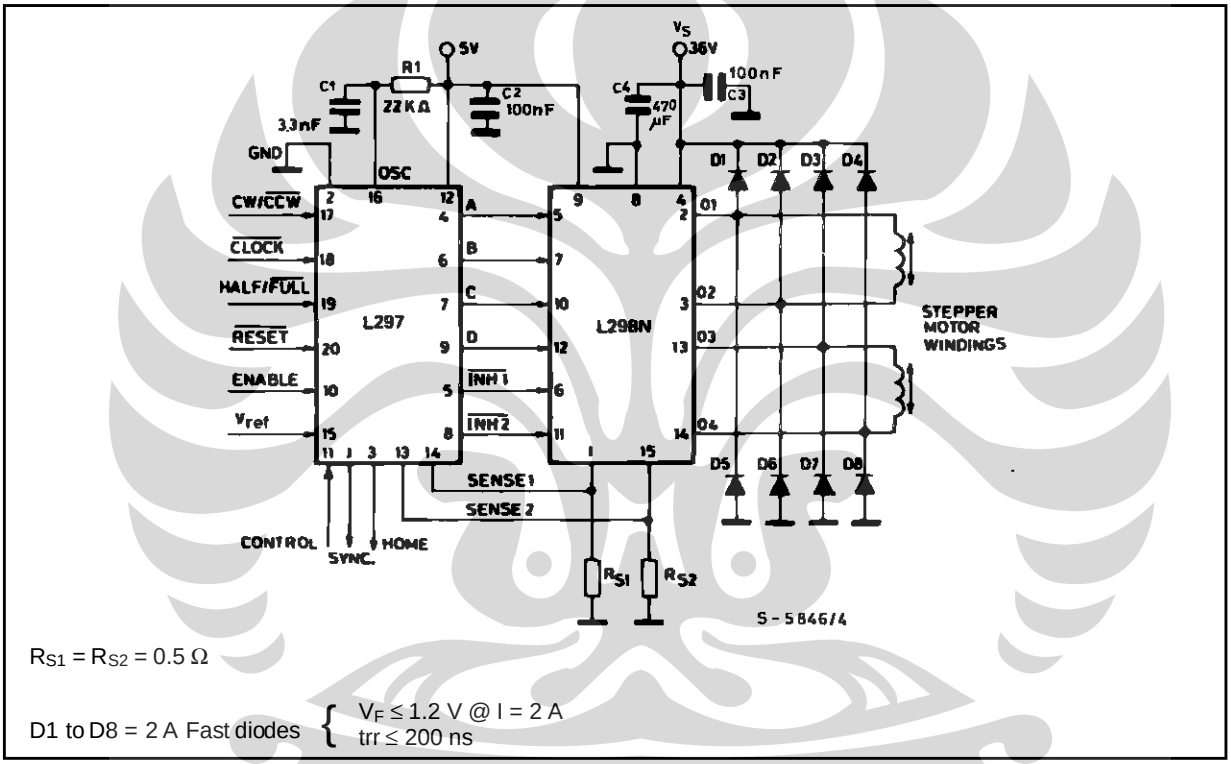
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



$R_{S1} = R_{S2} = 0.5\ \Omega$

D1 to D8 = 2 A Fast diodes { $V_F \leq 1.2$ V @ I = 2 A ; trr ≤ 200 ns

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

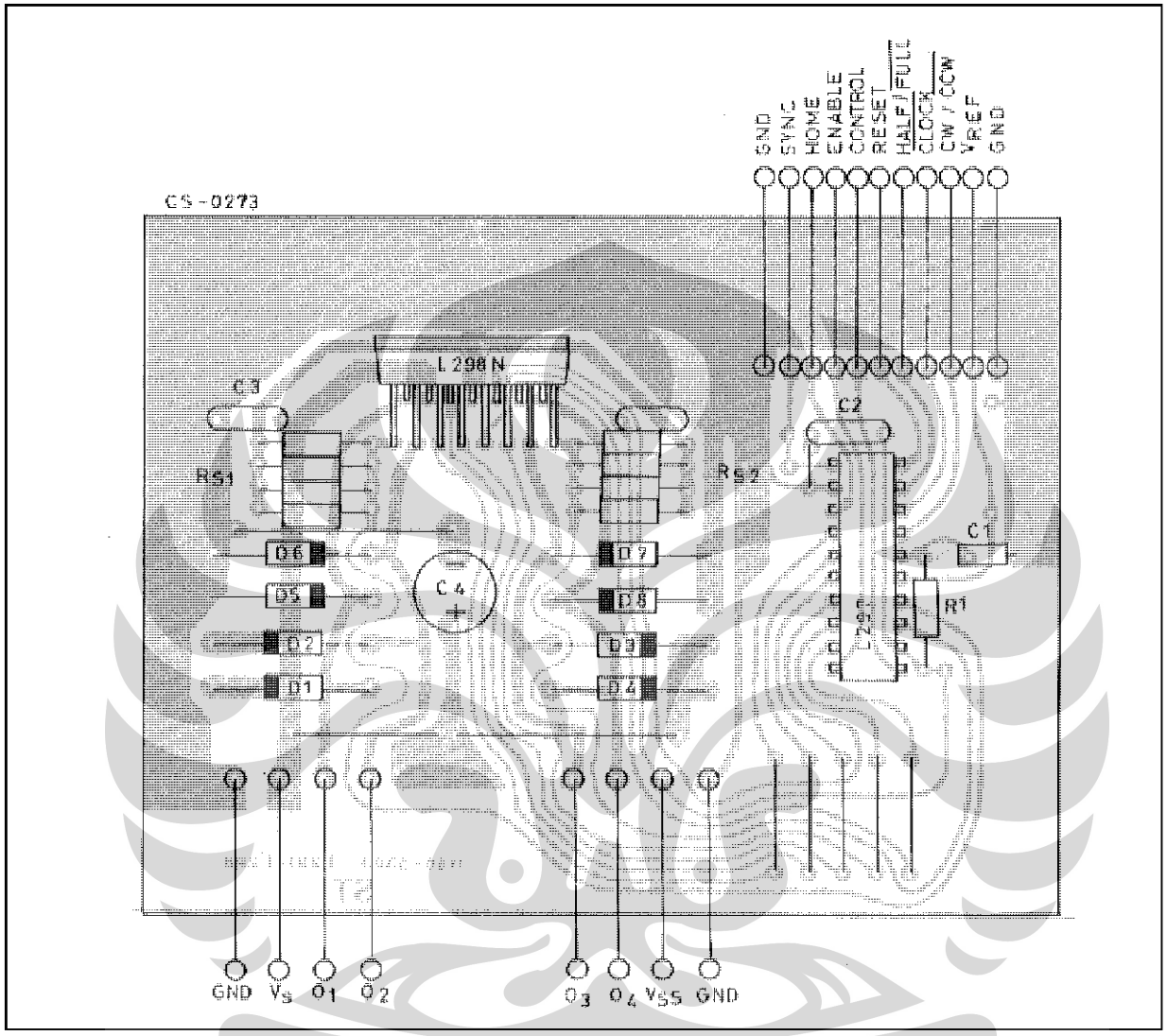
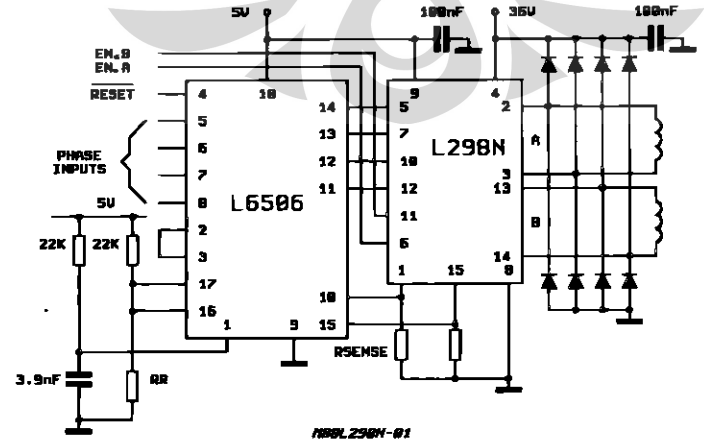


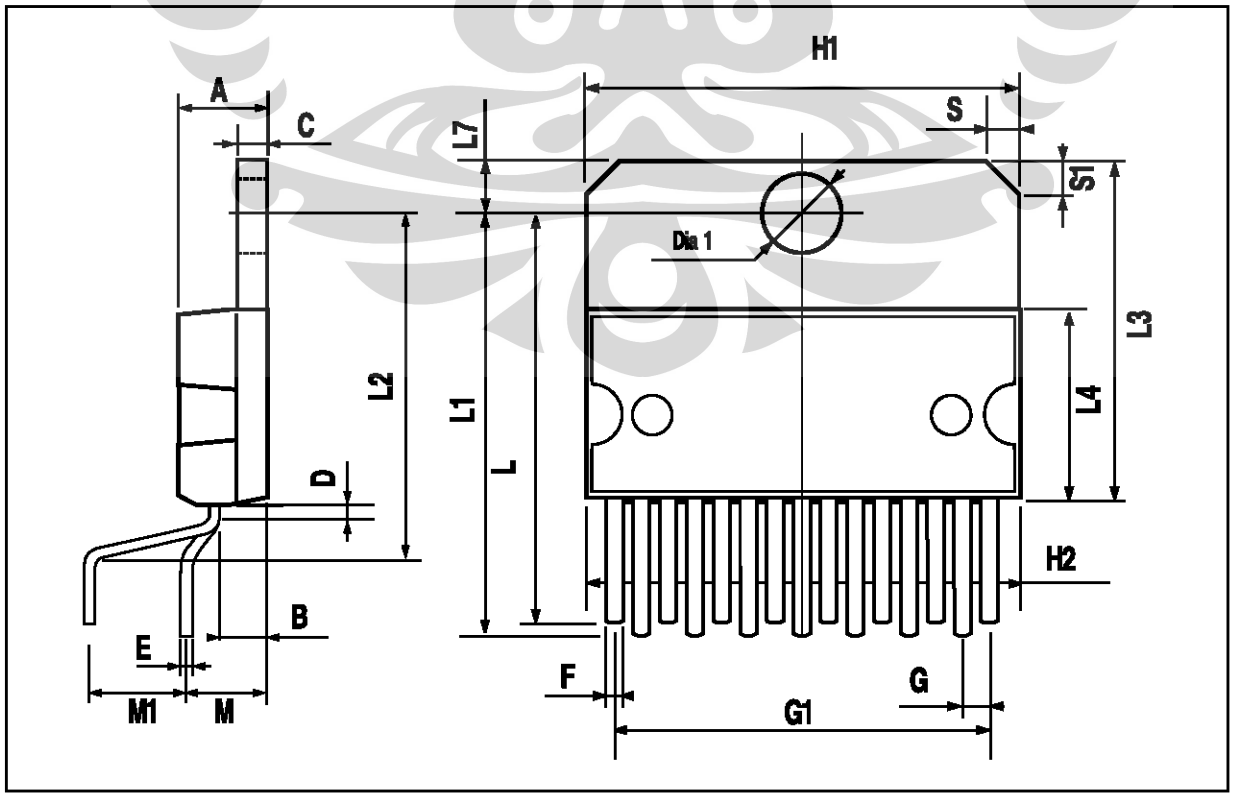
Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



R_R and R_{sense} depend from the load current

MULTIWATT15 (VERTICAL) PACKAGE MECHANICAL DATA

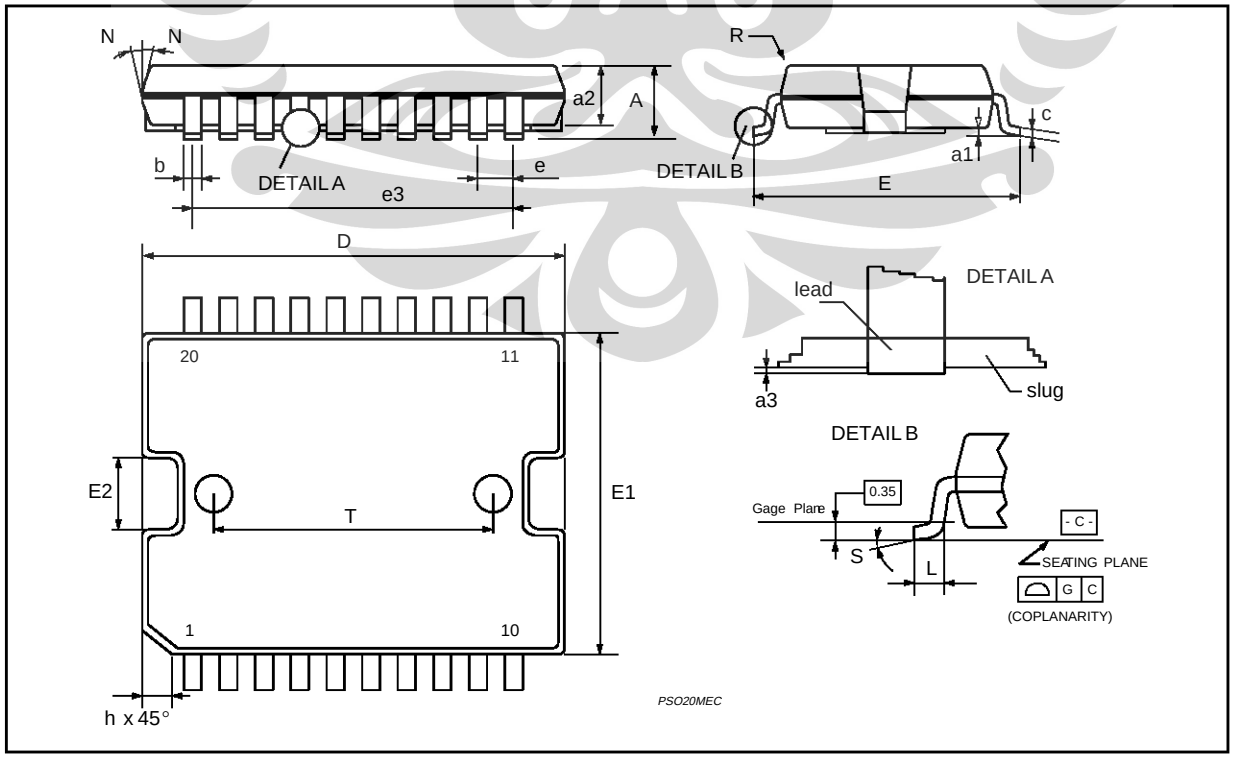
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L	22.1		22.6	0.870		0.890
L1	22		22.5	0.866		0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.2	4.3	4.6	0.165	0.169	0.181
M1	4.5	5.08	5.3	0.177	0.200	0.209
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152



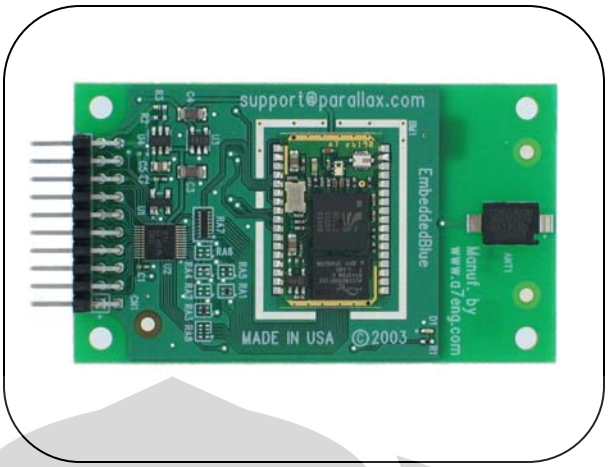
PowerSO20 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.60			0.1417
a1	0.10		0.30	0.0039		0.0118
a2			3.30			0.1299
a3	0		0.10	0		0.0039
b	0.40		0.53	0.0157		0.0209
c	0.23		0.32	0.009		0.0126
D (1)	15.80		16.00	0.6220		0.6299
E	13.90		14.50	0.5472		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.90		11.10	0.4291		0.437
E2			2.90			0.1141
G	0		0.10	0		0.0039
h			1.10			
L	0.80		1.10	0.0314		0.0433
N	10° (max.)					
S	8° (max.)					
T		10.0			0.3937	

(1) "D and E1" do not include mold flash or protrusions
- Mold flash or protrusions shall not exceed 0.15mm (0.006")



Wireless Technology
bridging your world)))™



EmbeddedBlue™ eb500-SER OEM Bluetooth® Serial Adapter

Features and Benefits

- Simple serial UART communications and control
- Seamless connectivity with other Bluetooth device
- Fully embedded solution with no Bluetooth host stack required
- 2.4GHz FHSS (Frequency Hopping Spread Spectrum) technology ensures high reliability and is robust to interference
- Compatible with Parallax® AppMod header for simple integration
- Low current consumption for long battery life
- Complete with sample applications and source code

Description

The eb500-SER implements all components of the Bluetooth stack on board so that additional host processor code is not required. Once a connection to another Bluetooth device has been established, the link has the appearance of a cabled serial connection eliminating the need for special wireless protocol knowledge.

Simple UART communication facilitates the interface between the host processor and the eb500-SER. This UART interface may be used to discover, connect, and communicate with other Bluetooth devices. An LED indicator for connection status is provided as a standard feature.

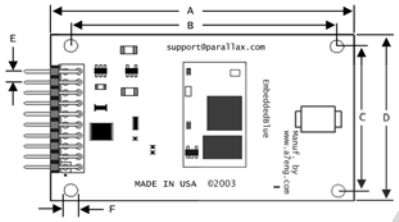
Applications

The EmbeddedBlue eb500-SER Bluetooth serial adapter is ideal for enabling your BASIC Stamp with a widely supported industry standard wireless protocol. Monitoring and control applications will benefit from an integrated implementation of the serial port profile for seamless connectivity with desktop computers, PDAs, and cellular phones. A focus on low current consumption makes the eb500-SER ideal for use in standalone battery powered devices common to robotics and remote data capture.

www.a7eng.com

Specifications and Ordering Information

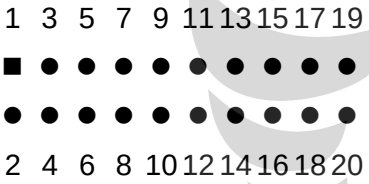
Dimensions



Note: controlling measurement is inches

	inches	mm
A	2.75	69.85
B	2.40	60.96
C	1.30	33.02
D	1.60	40.64
E	0.10	2.54
F	0.125	3.20

Pinout



1	VSS	2	VSS
3	TX	4	RX
5	NC	6	NC
7	NC	8	STATUS
9	MODE	10	NC
11	NC	12	NC
13	NC	14	NC
15	NC	16	NC
17	NC	18	NC
19	NC	20	VCC

Specifications

Transmit Power	+6dBm (max)
Receiver Sensitivity	-85dBm
Operating Temperature	-15° to 70°C
Supply Power	5 to 12VDC
Current Consumption	115.2kbps data transfer: 35mA 9.6kbps data transfer: 25mA idle connection: 8mA no connection: 3mA
Interfaces	5V TTL UART or RS232 serial with optional eb600 adapter Baud rates of 9.6k – 230.4k
Connectors	10x2 AppMod compatible 20 pin header with 0.1” spacing
Antenna	Internal surface mount
Bluetooth Support	Version 1.2 compliant with profiles GAP, SDP and SPP
Firmware	Upgradeable with A7 Engineering utility software
Size	75.85mm x 40.64mm x 8.9mm

Ordering Information

Part Number	Description
eb500-SER	eb500-SER adapter

Manufactured by:
A7 Engineering, Inc.
12127 Kirkham Road, Suite 101
Poway, CA 92064
Tel: (858) 391-1960
Fax: (619) 956-0082
Web: www.a7eng.com
E-mail: sales@a7eng.com

Distributed by:
Parallax, Inc.
599 Menlo Drive, Suite 100
Rocklin, CA 95765
Tel: (888) 512-1024
Fax: (916) 624-8003
Web: www.parallax.com
E-mail: sales@parallax.com

bridging your world)))™

A7-PB-eb500-SER
Revised March 20, 2007

www.a7eng.com