



UNIVERSITAS INDONESIA

Text to Voice with Sad Condition

SKRIPSI

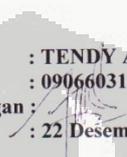
Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

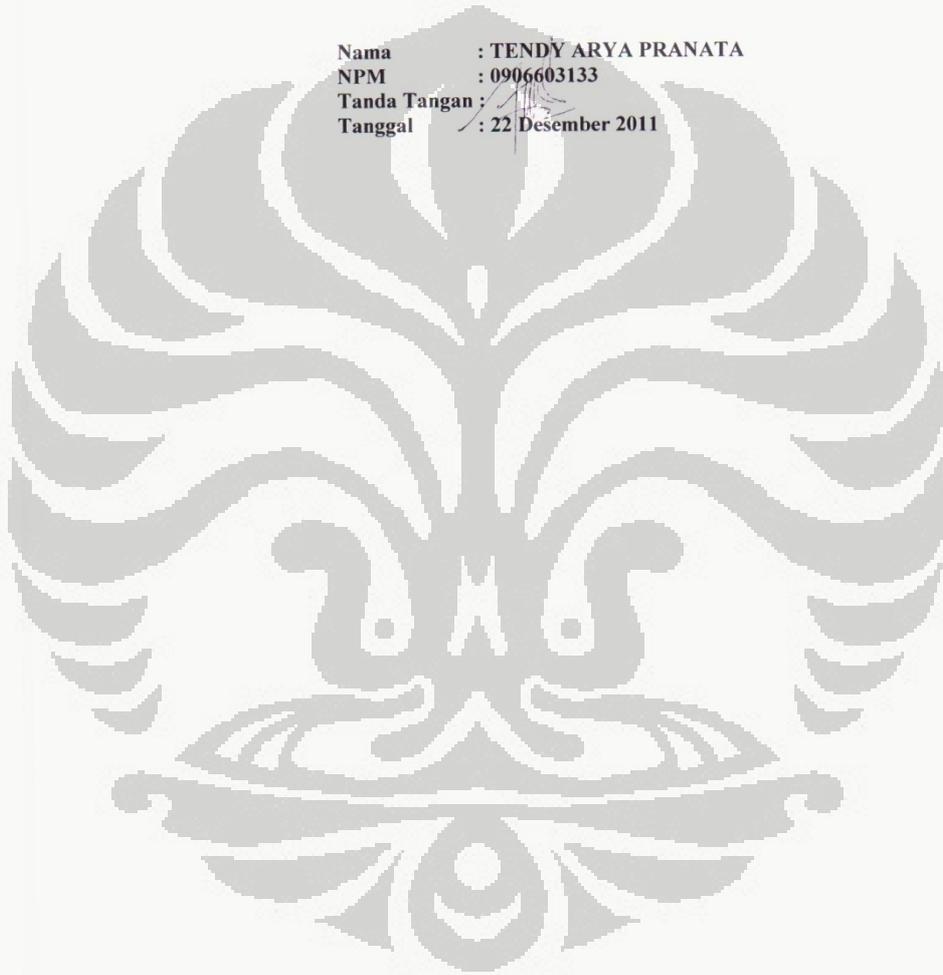
**Tendy Arya Pranata
0906603133**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
EKSTENSI
DEPOK
Desember 2011**

HALAMAN PERNYATAAN ORISINALITAS

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

**Nama : TENDY ARYA PRANATA
NPM : 0906603133
Tanda Tangan : 
Tanggal : 22 Desember 2011**



LEMBAR PERSETUJUAN

Skripsi dengan judul :

Text to Voice with Sad Condition

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam Presentasi skripsi.

Depok, 22 Desember 2011

Dosen Pembimbing,

Dr. Ir. Arman Djohan.D

NIP :194811131985031001

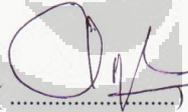
LEMBAR PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Tedy Arya Pranata
NPM : 0906603133
Program Studi : Teknik Elektro
Judul Skripsi : *Text to Voice with Sad Condition*

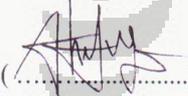
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. Ir. Arman Djohan Diponegoro

()

Penguji : Prof. Dr. Ir. Harry Sudibyo M.Sc.

()

Penguji : Dr. Ir. Purnomo Sidi Priambodo M.Sc.

()

Ditetapkan di : Depok

Tanggal

: 24 Januari 2012

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektronika pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Allah SWT. Atas segala nikmat yang diberikan sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Dr. Ir. Arman Djohan.D, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
3. Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
4. Tomy dan Achmad Gunawan sebagai sahabat dekat yang telah banyak membantu saya dalam menyelesaikan skripsi ini, serta rekan-rekan mahasiswa ekstensi teknik elektronika 2009.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 20 Desember 2011

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

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

Nama : TENDY ARYA PRANATA
NPM : 0906603133
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Text to Voice with Sad Condition

berserta perangkat yang ada. 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 : 20 Desember 2011

Yang menyatakan


(TENDY ARYA PRANATA)

Abstrak

Seiring perkembangan teknologi di dunia telekomunikasi, banyak penemuan dan inovasi layanan yang dibuat untuk mempermudah para konsumen dalam berkomunikasi.

Pada penelitian ini akan dibuat alat yang dapat merubah dari bentuk teks ke dalam bentuk suara ketika sedih dengan bantuan *device* dan program. Pada penelitian ini menggunakan program bahasa C, lalu pada bagian alatnya menggunakan mikrokontroler ATmega 128 dan modul suara. Program dibuat dalam Bahasa C kemudian diunduh ke dalam perangkat keras berupa mikrokontroler ATmega 128 dan dikonversi oleh modul suara yang sudah menyimpan database lalu dikeluarkan oleh output *speaker*.

Kata kunci : Bahasa C, Mikrokontroler ATmega 128, Modul Suara, Speaker.

Abstract

Along with the development of telecommunications technology in the world, many inventions and innovations made services to facilitate the consumer in communicating.

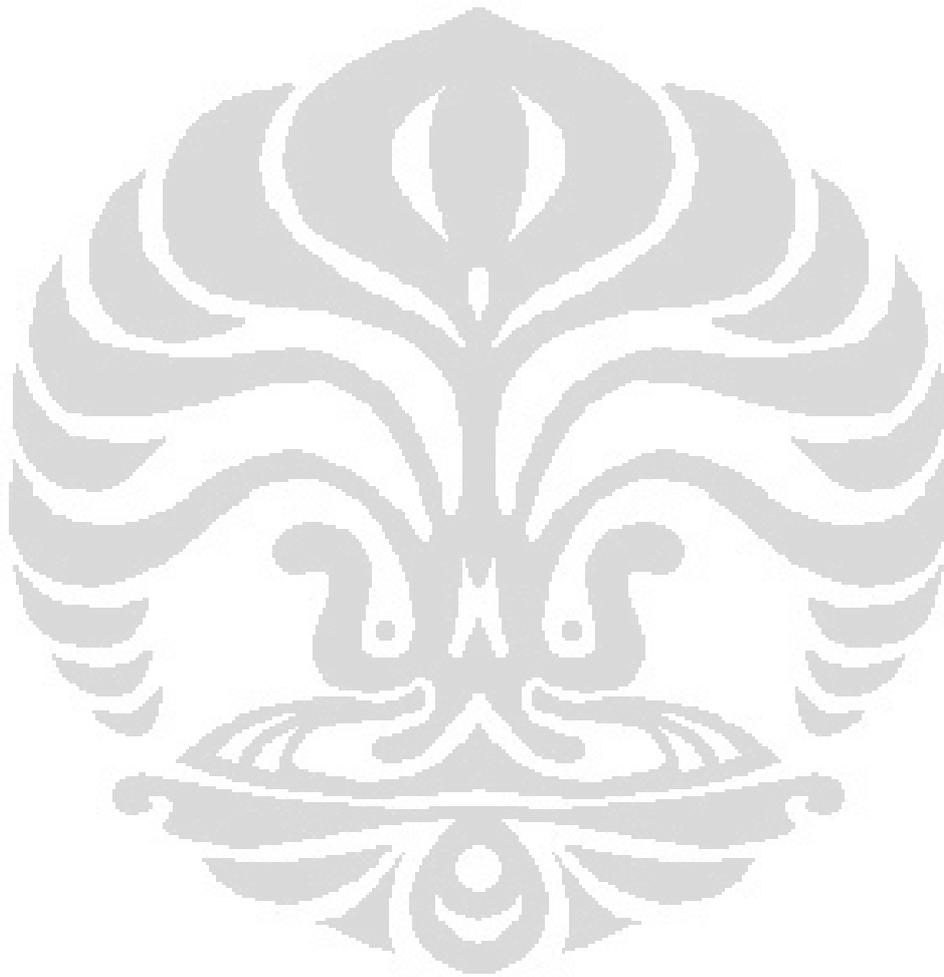
In this study will be made a tool that can change the form of text into the form when the sad voice with the help of devices and programs. In this study using the C language program, then on the appliance using a microcontroller ATmega 128 and sound modules. Programs created in C language and then downloaded into the hardware of the microcontroller ATmega 128 and converted by a voice module that already store the database and then released by the speaker output.

Keywords: C Language, Microcontroller ATmega 128, Voice Module, Speaker

DAFTAR ISI

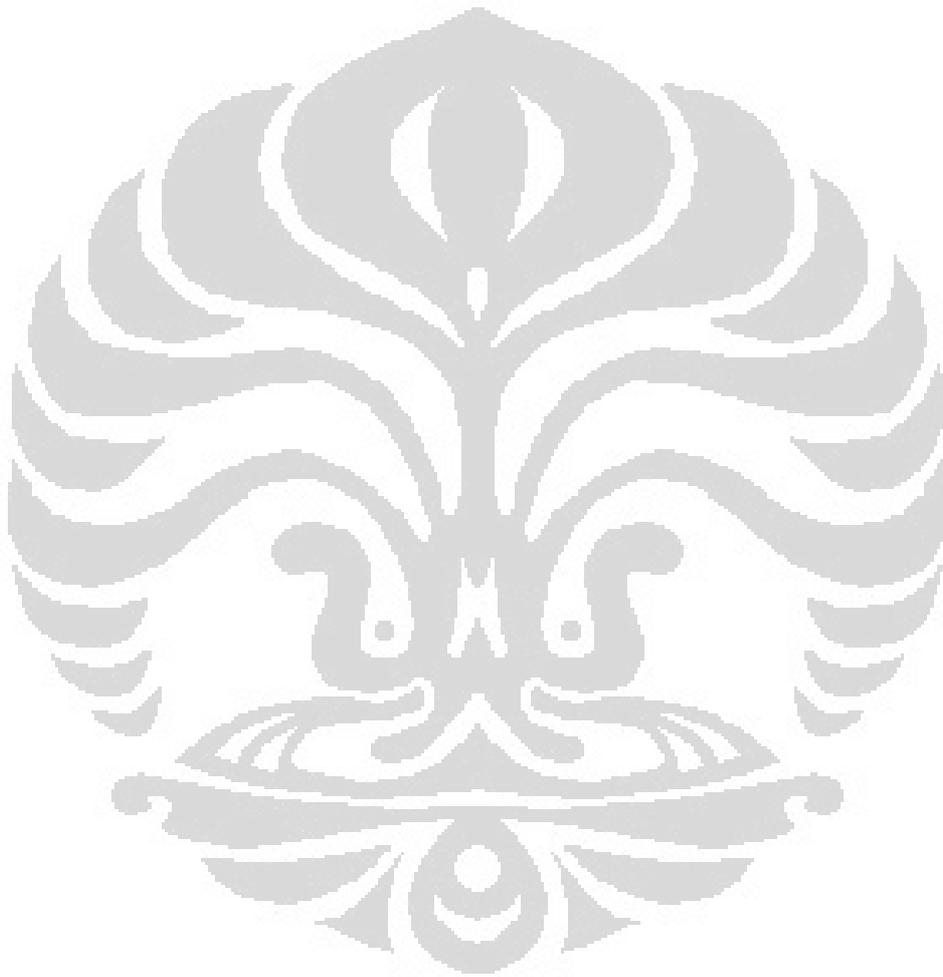
	Halaman
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PERSETUJUAN	iv
KATA PENGANTAR	v
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
ABSTRAK	vii
DAFTAR ISI	viii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
LAMPIRAN	xi
1. PENDAHULUAN	1
1.1.1. Latar Belakang	1
1.1.2. Batasan Masalah	1
1.1.3. Tujuanpenulisan.....	2
1.1.4. MetodePenulisan.....	2
1.1.5. Sistematika Penulisan	2
.....	2
2. LANDASAN TEORI	4
2.1. Mikrokontroller.....	4
2.2. Modul Suara	8
2.3. Bahasa C	9
2.3.1. Header File	12
2.3.2. Preprosesor Directive (#include)	12
2.3.3. Void	12
2.3.4. Main ()	13
2.3.5. Statement	13
3. RANCANG BANGUN ALAT & PROGRAM	14
3.1. Perancangan Alat	14
3.2. Perancangan Program	17
4. PENGUJIAN & ANALISA DATA	19
4.1. Deskripsi Pengujian	19
4.2. Pengujian Program.....	19
4.3. Analisa data hasil.....	23
5. KESIMPULAN DAN SARAN	26

5.1. Kesimpulan	26
5.2. Saran	26
DAFTAR PUSTAKA	27
LAMPIRAN	28



DAFTAR TABEL

	Halaman
1. Kata kunci dalam Bahasa C	9

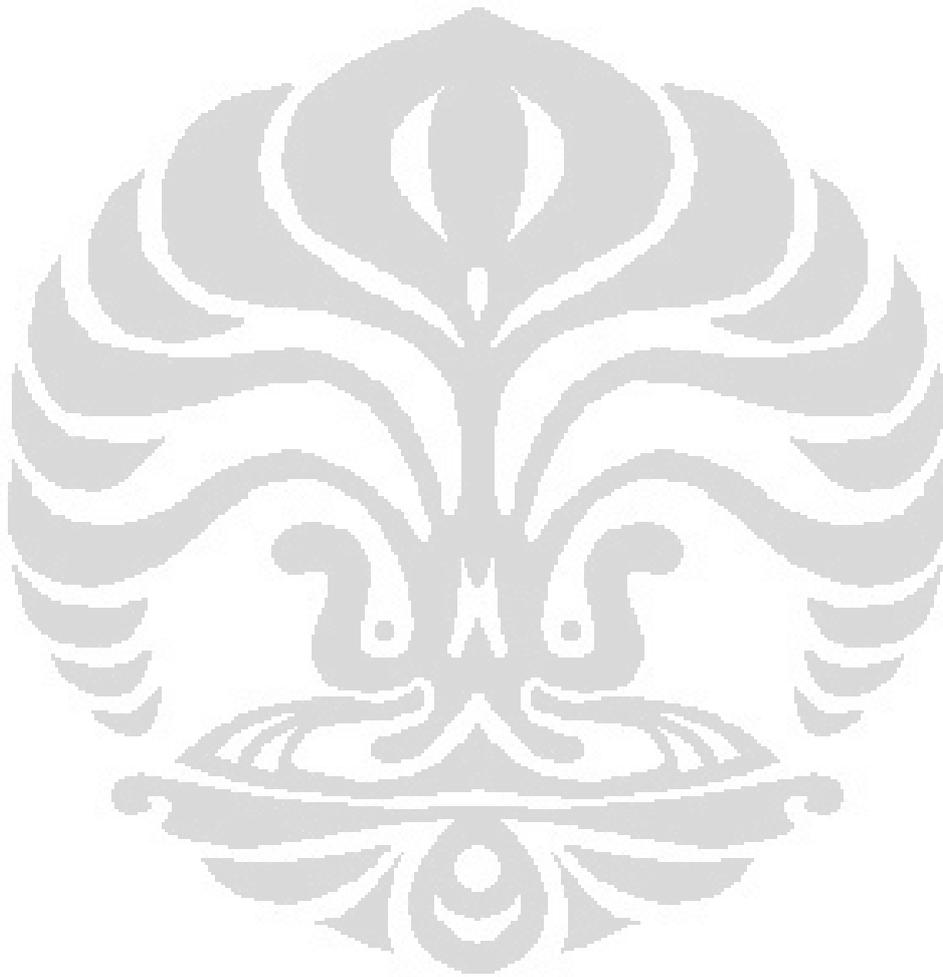


DAFTAR GAMBAR

	Halaman
1. Gambar ATmega 128 <i>Pin out</i>	7
2. Tata letak DT- AVR ATMEGA 128L BMS	9
3. Modul Suara.....	9
4. Sejarah Bahasa C	10
5. Mikrokontroler ATmega 128	15
6. Struktu & Pin Modul Suara	16
7. Diagram alir program.....	17
8. Data percobaan 1 dari mikrokontroler.....	20
9. Data percobaan 1 dari modul suara.....	20
10. Data percobaan 2 dari mikrokontroler.....	21
11. Data percobaan 2 dari modul suara.....	21
12. Data percobaan 3 dari mikrokontroler.....	21
13. Data percobaan 3 dari modul suara.....	22
14. Data percobaan 4 dari mikrokontroler.....	22
15. Data percobaan 4 dari modul suara.....	22
16. Data percobaan 5 dari mikrokontroler.....	23
17. Data percobaan 5 dari modul suara.....	23

DAFTAR LAMPIRAN

	Halaman
1. Data Kode ASCII	28
2. Database Kata	28



BAB I

PENDAHULUAN

1.1. Latar Belakang

Komunikasi melalui surat elektronik (*email*) atau *chat* banyak digunakan oleh masyarakat karena penggunaannya yang sangat sederhana dan biayanya relatif murah dibanding sistem komunikasi lain. Namun interaksi antara pengguna dan komputer sebagai perangkat elekomunikasi, merupakan masalah yang dihadapi oleh para tunanetra dan manula (seseorang yang indra penglihatannya tidak terlalu peka). Hal ini terjadi karena ukuran huruf atau angka pada ponsel cukup kecil sehingga sulit untuk membacanya. Sehubungan dengan hal di atas maka alat pada proyek akhir ini dapat mengkonversi isi kalimat menjadi suara. Sehingga untuk mengetahui isi text dapat dengan cara mendengarkan hasil konversi.

Alat ini memerlukan sebuah ruang penyimpanan database fonem yang cukup besar. Maka digunakan SD Card (*Secure Digital Card*) sebagai media penyimpanan database suara. Untuk mengambil isi teks dari komputer digunakan atmega128 dengan memanfaatkan komunikasi serialPerangkat Text to Voice ini memberikan kemudahan kepada konsumennya khususnya bagi para konsumen yang cinta dan hobi dengan perangkat komputer. Terlebih dengan maraknya perkembangan teknologi jaringan informasi saat ini, seperti jejaring sosial yang pada umumnya menggunakan komunikasi dua arah dengan menggunakan teks atau biasa disebut dengan *chat*.

Namun untuk perangkat *Text to Voice* ini dilengkapi kemampuan untuk membaca teks maupun karakter yang menunjukkan perasaan sedih (*Sad Condition*) yang ditunjukkan oleh penulis kepada pembacanya dengan outputnya berupa suara sedih.

1.2. Batasan Masalah

Batasan masalah untuk skripsi ini yaitu membahas rancangan awal pembentukan seperti program / perangkat lunak yang digunakan yaitu Bahasa C lalu membahas pula pembentukan rancang bangun dari alat yang digunakan untuk

mengaplikasikan program berupa Mikrokontroller ATmega 128 serta modul suara yang digunakan untuk mengkonversi ke bentuk suara.

1.3. Tujuan Penulisan

Tujuan penulisan skripsi ini untuk memberikan informasi pengetahuan tentang teknologi konversi teks menjadi suara, alat yang digunakan, cara penggunaannya, serta mampu menganalisa dan mengembangkannya.

1.4. Metode Penulisan

Dalam penyusunan Laporan Skripsi ini, digunakan beberapa metode, antara lain :

1. Metode Study Literatur.

Mengambil dan mengumpulkan teori-teori dasar serta teori pendukung dari berbagai sumber, buku-buku referensi dan situs-situs dari internet tentang apa-apa yang menunjang dalam analisa ini.

2. Metode Konsultasi.

Melakukan konsultasi dengan pembimbing skripsi serta teman-teman mahasiswa yang dapat memberikan masukan dan ide-ide.

1.5. Sistematika Penulisan

BAB I PENDAHULUAN

Berisi latar belakang batasan masalah, tujuan penulisan, metode penulisan, serta sistematika penulisan.

BAB II LANDASAN TEORI

Berisi paparan tentang teori pemrograman bahasa C, mikrokontroller ATMEGA 128 dan modul suara.

BAB III RANCANG BANGUN ALAT & PROGRAM

Berisi penjelasan tentang perancangan perangkat lunak bahasa C dan perancangan alat yang digunakan untuk mampu mengkonversi dari teks menjadi suara yaitu modul suara.

BAB IV PENGUJIAN & ANALISA DATA

Berisi tentang percobaan program, data/sampel pengubahan teks menjadi suara.

BAB V KESIMPULAN & SARAN

Berisi tentang kesimpulan dari hasil uji coba dan analisa pada bab sebelumnya serta saran untuk menjadi masukan yang berguna bagi pengembangan teknik ini.

BAB II

LANDASAN TEORI

2.1 Mikrokontroller

Mikrokontroller, sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi kebutuhan pasar (*market need*) dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara masal (dalam jumlah banyak) membuat harganya menjadi lebih murah (dibandingkan mikroprosesor). Sebagai kebutuhan pasar, mikrokontroller hadir untuk memenuhi selera industri dan para konsumen akan kebutuhan dan keinginan alat-alat bantu bahkan mainan yang lebih baik dan canggih.

Tidak seperti sistem komputer, yang mampu menanganiberbagai macam program aplikasi (misalnya pengolah kata, pengolah angkadan lain sebagainya), mikrokontroller hanya bisa digunakan untuk suatu aplikasi tertentu saja (hanya satu program saja yang bisa disimpan). Perbedaan lainnya terletak pada perbandingan RAM dan ROM. Pada sistem komputer perbandingan RAM dan ROM-nya besar, artinya program-program pengguna disimpan dalam ruang RAM yang relatif besar, sedangkan rutin-rutin antarmuka perangkat keras disimpan dalam ruang ROM yang kecil. Sedangkan pada Mikrokontroller, perbandingan ROM dan RAM-nya yang besar, artinya program kontrol disimpan dalam ROM (bisa *Masked ROM* atau *Flash PEROM*) yang ukurannya relatif lebih besar, sedangkan RAM digunakan sebagai tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroller yang bersangkutan.

Adapun kelebihan dari mikrokontroller adalah sebagai berikut :

- A. Penggerak pada mikrokontoler menggunakan bahasa pemograman *assembly* dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem (bahasa *assembly* ini mudah dimengerti karena menggunakan bahasa *assembly* aplikasi dimana parameter input dan output langsung bisa diakses tanpa menggunakan banyak perintah). Desain bahasa *assembly* ini tidak menggunakan begitu banyak syarat penulisan bahasa pemrograman seperti huruf besar dan huruf kecil untuk bahasa *assembly* tetap diwajibkan.
- B. Mikrokontroller tersusun dalam satu *chip* dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan kontrol sistem sehingga mikrokontroller dapat dikatakan sebagai komputer mini yang dapat bekerja secara inovatif sesuai dengan kebutuhan sistem.
- C. Sistem *running* bersifat berdiri sendiri tanpa tergantung dengan komputer sedangkan parameter komputer hanya digunakan untuk *download* perintah instruksi atau program. Langkah-langkah untuk *download* komputer dengan mikrokontroller sangat mudah digunakan karena tidak menggunakan banyak perintah.
- D. Pada mikrokontroller tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.
- E. Harga untuk memperoleh alat ini lebih murah dan mudah didapat.

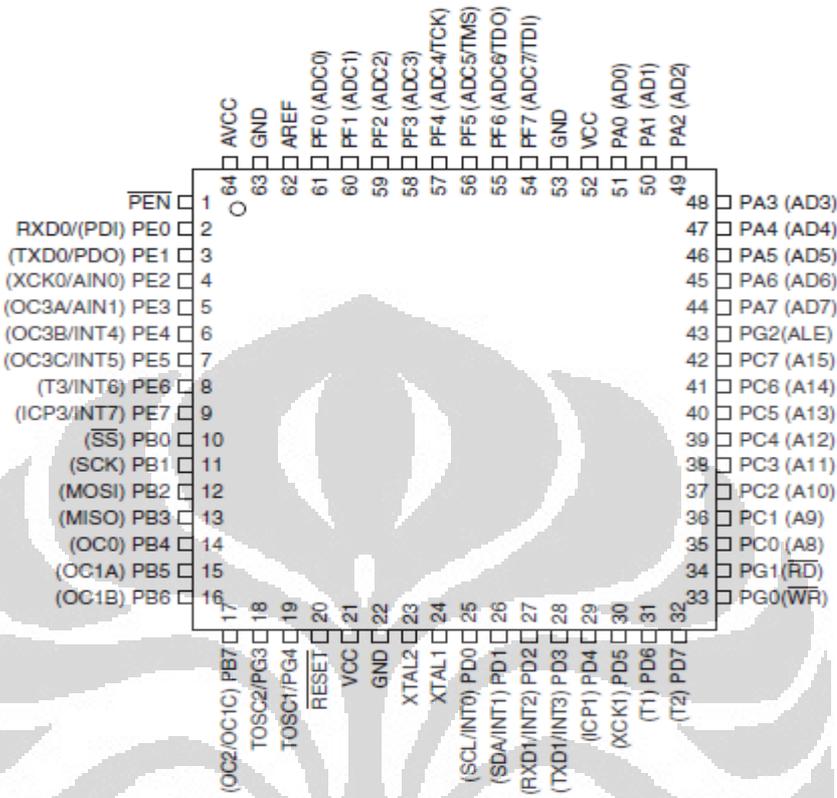
Mikrokontroller ATmega 128 merupakan mikrokontroller keluarga AVR yang mempunyai kapasitas flash memori 128KB. AVR (*Alf and Vegard's Risc Processor*) merupakan seri mikrokontroller CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Secara umum, AVR dapat terbagi menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga AT-Mega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, bisa dikatakan hampir sama. Semua jenis AVR dilengkapi dengan *flash* memori sebagai memori program. Kapasitas dari *flash* memori ini berbeda antara *chip* yang satu dengan *chip* yang lain. Tergantung dari jenis IC yang

digunakan. Untuk *flash* memori yang paling kecil adalah 1 kbytes (ATtiny11, ATtiny12, dan ATtiny15) dan paling besar adalah 128 kbytes (AT-Mega128). Mikrokontroler AVR ATmega-128 memiliki spesifikasi sebagai berikut:

- a. Saluran I/O sebanyak 56 buah, yaitu Port A, Port B, Port C, Port D, Port E, Port F dan Port G.
- b. ADC 10 bit sebanyak 8 saluran.
- c. 2 buah *Timer/Counter* 8 bit dan 2 buah *Timer/Counter* 16 bit.
- d. Dua buah PWM 8 bit.
- e. *Watchdog Timer* dengan osilator internal.
- f. Internal SRAM sebesar 4 kbyte.
- g. Memori *flash* sebesar 128 kBytes.
- h. Interupsi Eksternal.
- i. Port antarmuka SPI.
- j. EEPROM sebesar 4 kbyte.
- k. Real time counter.
- l. 2 buah Port USART untuk komunikasi serial.
- m. Enam kanal PWM.
- n. Tegangan operasi sekitar 4,5 V sampai dengan 5,5V.

Pemrograman mikrokontroler merupakan dasar dari prinsip pengendalian kerja robot, dimana orientasi dari penerapan mikrokontroler ialah untuk mengendalikan suatu sistem berdasarkan informasi input yang diterima, lalu diproses oleh mikrokontroler, dan dilakukan aksi pada bagian output sesuai program yang telah ditentukan sebelumnya. Mikrokontroler merupakan pengontrol utama perangkat elektronika saat ini, termasuk robot tentunya. Mikrokontroler yang terkenal dan mudah didapatkan di Indonesia saat ini ialah 89S51, AVR ATmega 8535, ATmega16, Atmega32 dan Atmega128. Beberapa merek lain yang terkenal misalnya PIC 16F877 dan Basic Stamp 2. Mikrokontroler memiliki beberapa Port yang dapat digunakan sebagai I/O (input/Output). Gambar Berikut ini merupakan susunan kaki standar 40 pin DIP mikrokontroler AVR ATmega 8535. Sedangkan ATmega128 memiliki

kemampuan yang lebih besar, yaitu 128 KB ISP *Flash Memory*, 4K EEPROM, 4K internal SRAM dan memiliki 5 port I/O.

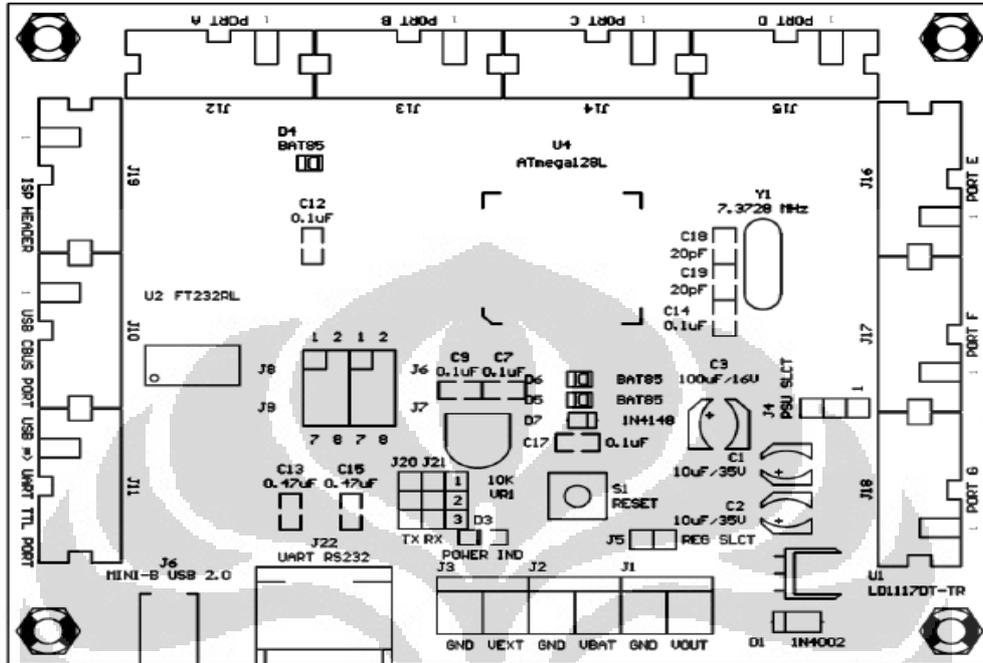


Gambar 2.1 Gambar ATmega 128 pin out

Pada mikrokontroler ATMEGA128 menggunakan serial komunikasi USART. Sehingga Data Rx berfungsi sebagai pengirim dan penerima data sedangkan Tx sebagai output clock. Output dari serial pada mikrokontroler ATMEGA 128 berupa tegangan TTL 0 – 5 Volt.

Dalam komunikasi data serial, data dikirim dalam bentuk pulsa kontinyu yang disebut bit. Data dikirim bit per bit secara berurutan melalui kanal komunikasi yang telah ditentukan. Penerima juga menerima data dalam bentuk bit-bit yang sama. DT-AVR ATmega128L BOOTLOADER MICRO SYSTEM (BMS) merupakan sebuah modul single chip berbasis mikrokontroler ATmega128L. DT-AVR ATmega128L BMS dilengkapi dengan program bootloader sehingga tidak membutuhkan divais programmer. Dengan menggunakan bootloader pada DT-AVRATmega128L BMS, pengguna dapat menggunakan jalur UART sebagai jalur komunikasi dengan komputer, sekaligus

menggunakannya untuk melakukan remote programming jika ada perbaikan program (update).

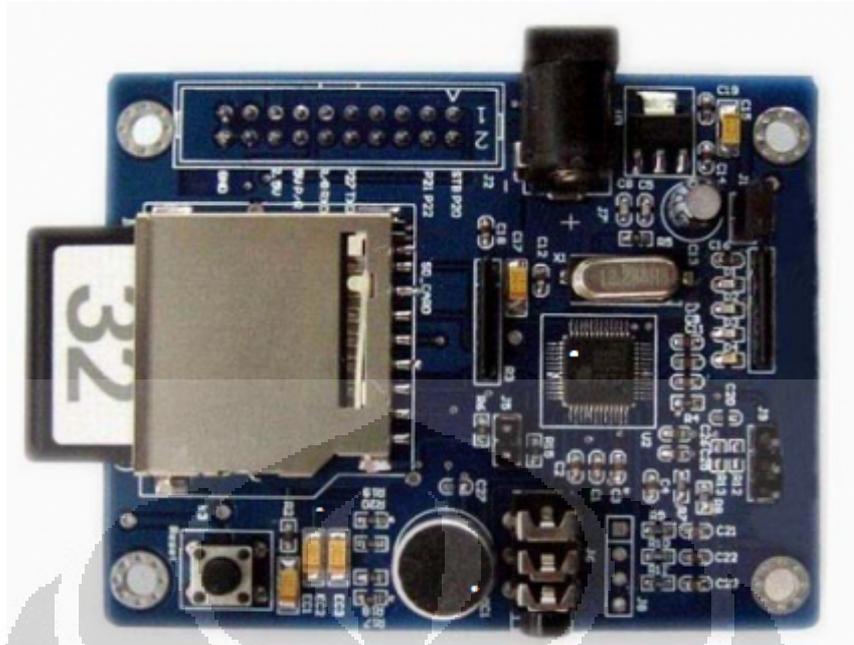


Gambar 2.2 Tata letak DT-AVR ATMEGA128L BMS

2.2 Modul Suara

Rangkaian modul suara yang digunakan dengan tipe TDR025 ini berfungsi untuk menyimpan database suara yang direkam dari mikrofon dan membutuhkan catuan tegangan 5volt DC. Berikut spesifikasi untuk modul suara TDR025 :

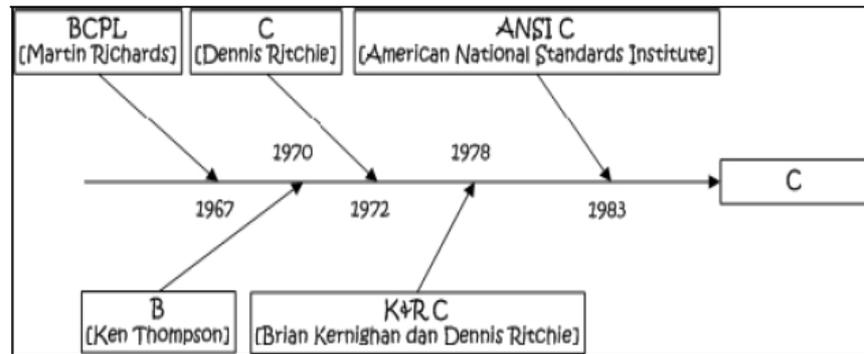
- A. Support file MP3, WMA, WAV ,MIDI.
- B. Support mikrokontroler.
- C. Antarmuka paralel, i2c, serial.
- D. File di simpan di MMC /SDCARD maksimum 2GB.
- E. 32 Mb SD CARD dapat merekam lebih dari 2 jam.
- F. Mikrofon pada modul untuk merekam.



Gambar 2.3 Modul Suara

2.3 Bahasa C

Bahasa C dilahirkan di Bell Telephone Laboratories (atau sering disebut sebagai Bell Labs saja). Sulit membayangkan dunia modern saat ini jika tidak ada pengaruh dari Bell Labs. Pada tahun 1947, transistor ditemukan di Bell Labs, dan juga sistem operasi yang sekarang banyak digunakan di dalam komputer korporat (UNIX) juga dibuat di sana. Untuk beberapa tahun, ada sebuah bahasa pemrograman yang sangat dekat dengan sistem operasi UNIX, yang disebut dengan bahasa C, yang didesain oleh Dennis Ritchie dan Brian Kernighan. Bahasa C disebut demikian mengingat bahasa tersebut adalah turunan dari bahasa B, dan bahasa B merupakan pemendekan dari Basic CPL, sementara CPL sendiri adalah sebuah bahasa pemrograman yang merupakan singkatan dari *Combined Programming Language*.



Gambar 2.4 Sejarah Bahasa C

Meskipun populer, bahasa C terkesan lebih rumit jika dibandingkan dengan bahasa pemrograman lainnya, khususnya jika digunakan oleh para pemrogram pemula. Ketika ALGOL dan banyak turunan ALGOL menggunakan kata-kata yang mudah diingat seperti BEGIN dan END untuk membatasi sebuah seksi dalam instruksi program, bahasa C malah menggunakan tanda kurung keriting ({ dan }). Beberapa operasi juga disederhanakan, bahkan banyak singkatannya yang cukup membingungkan para programmer pemula. Contoh yang sering digunakan adalah printf, dan scanf. Meskipun demikian, program-program yang ditulis dalam bahasa C seringkali lebih efisien dibandingkan dengan bahasa pemrograman lainnya, yang artinya program-program dalam bahasa C diterjemahkan ke dalam machine code dalam jumlah yang relatif lebih sedikit jika dibandingkan dengan program yang ditulis dalam bahasa pemrograman lainnya. Salah satu bagian yang paling canggih dari bahasa C adalah bahwa bahasa C memiliki "pointer", selain tentunya "variabel" dan "konstanta". Istilah Pointer dalam bahasa pemrograman merujuk kepada alamat-alamat memori yang mengizinkan para programmer untuk melakukan beberapa tugas secara jauh *lebih efisien, dengan melibatkan bit, byte, dan word memori*. Karenanya, banyak orang menyebut bahasa C sebagai "*High-level Assembly language*, atau bahasa rakitan tingkat tinggi.

Kepopuleran bahasa C membuat versi-versi dari bahasa ini banyak dibuat untuk komputer mikro. Untuk membuat versi-versi tersebut standar, ANSI (*American National Standards Institute*) kemudian menetapkan standar ANSI untuk bahasa C. Standar ANSI ini didasarkan dari standar UNIX yang diperluas.

Standar ANSI menetapkan sebanyak 32 buah kata-kata kunci (keyword) standar. Ke 32 kata kunci ini ditunjukkan pada **Tabel 1.1** berikut ini :

Tabel 1.1 Kata kunci dalam bahasa C

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Suatu *source* program C baru dapat dijalankan setelah melalui tahap kompilasi dan penggabungan. Tahap kompilasi dimaksudkan untuk memeriksa source-program sesuai dengan kaidah-kaidah yang berlaku di dalam bahasa pemrograman C. Tahap kompilasi akan menghasilkan *relocatable object file*. File-file objek tersebut kemudian digabung dengan perpustakaan-fungsi yang sesuai, untuk menghasilkan suatu *executable-program*. Perintah-perintah penting bahasa C dalam pengoperasian program antara lain :

- a. F1 = Help (Menu pertolongan)
- b. F2 = Untuk menyimpan program (Save)
- c. F3 = Untuk membuka program (Open)
- d. ALT + F3 = Untuk menutup program (Close)
- e. F5 = Melebarkan editor (Maximize)
- f. ALT + F5 = Untuk melihat hasil running program
- g. ALT + F9 = Mengkompile (Compile)
- h. F9 = Mengecek error program (Make)
- i. CTRL + F9 = Untuk menjalankan program (Run)
- j. F10 = Mengaktifkan menu
- k. ALT + X = Keluar dari Turbo C++

Kelebihan Bahasa C yaitu:

- a. Bahasa C tersedia hampir di semua jenis komputer.
- b. Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis komputer.
- c. Bahasa C hanya menyediakan sedikit kata-kata kunci. Hanya terdapat 32 kata kunci.
- d. Proses executable program bahasa C lebih cepat.
- e. Dukungan pustaka yang banyak.
- f. C adalah bahasa yang terstruktur.
- g. Bahasa C termasuk bahasa tingkat menengah.

Kekurangan Bahasa C:

- a. Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
- b. Bagi pemula pada umumnya akan kesulitan menggunakan pointer.

Struktur pemrograman Bahasa C akan dijelaskan pada sub bab berikut:.

2.3.1 *Header File*

Adalah berkas yang berisi prototype fungsi, definisi konstanta dan definisi variable. Fungsi adalah kumpulan kode C yang diberi nama dan ketika nama tersebut dipanggil maka kumpulan kode tersebut dijalankan.

Contoh :

```
stdio.h
math.h
conio.h
```

2.3.2 *Preprosesor Directive (#include)*

Preprosesor directive adalah bagian yang berisi pengikutsertaan file atau berkas berkas fungsi maupun pendefinisian konstanta.

Contoh:

```
#include <stdio.h>
#include phi 3.14
```

2.3.3 *Void*

Adalah fungsi yang mengikutinya tidak memiliki nilai kembalian (return)

2.3.4 *Main* ()

Fungsi main () adalah fungsi yang pertama kali dijalankan ketika program dieksekusi, tanpa fungsi main suatu program tidak dapat dieksekusi namun dapat dikompilasi.

2.3.5 *Statement*

Statement adalah instruksi atau perintah kepada suatu program ketika program itu dieksekusi untuk menjalankan suatu aksi. Setiap statement diakhiri dengan titik-koma (;).



BAB III

Rancang Bangun Alat & Program

3.1 Perancangan Alat

Perancangan alat yang digunakan untuk membuat perangkat *Text to Voice* ditentukan berdasarkan blok diagram yang dibentuk terlebih dahulu agar lebih mudah menentukannya, seperti pada **Gambar 3.1** dibawah ini :

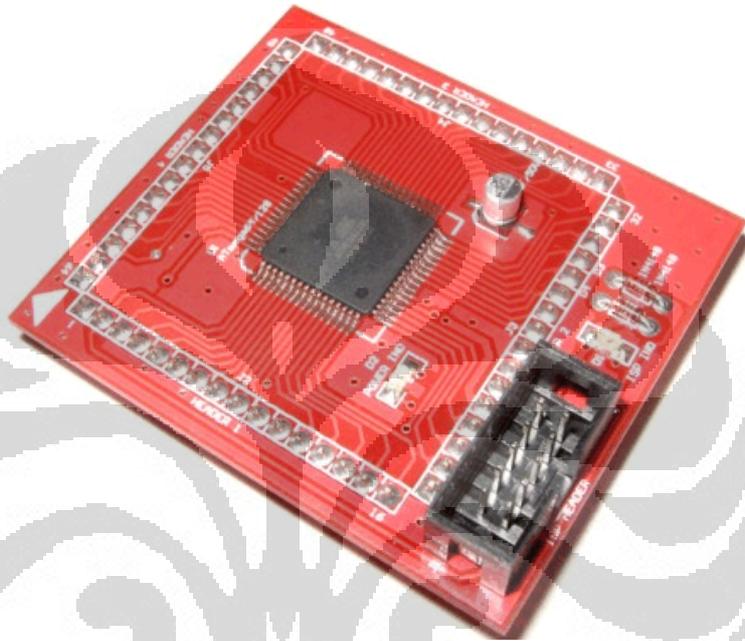


Gambar 3.1 Blok diagram Alat

Pada blok pertama, komputer berfungsi sebagai alat yang digunakan untuk merancang program *Text to Voice with Sad Condition* dengan mempergunakan bahasa C dan sekaligus sebagai inputan berupa teks yang diketik atau diterima untuk selanjutnya di terjemahkan. Komputer memberikan dan menghubungkan informasi input serta program kepada blok yang selanjutnya dengan mempergunakan komunikasi serial.

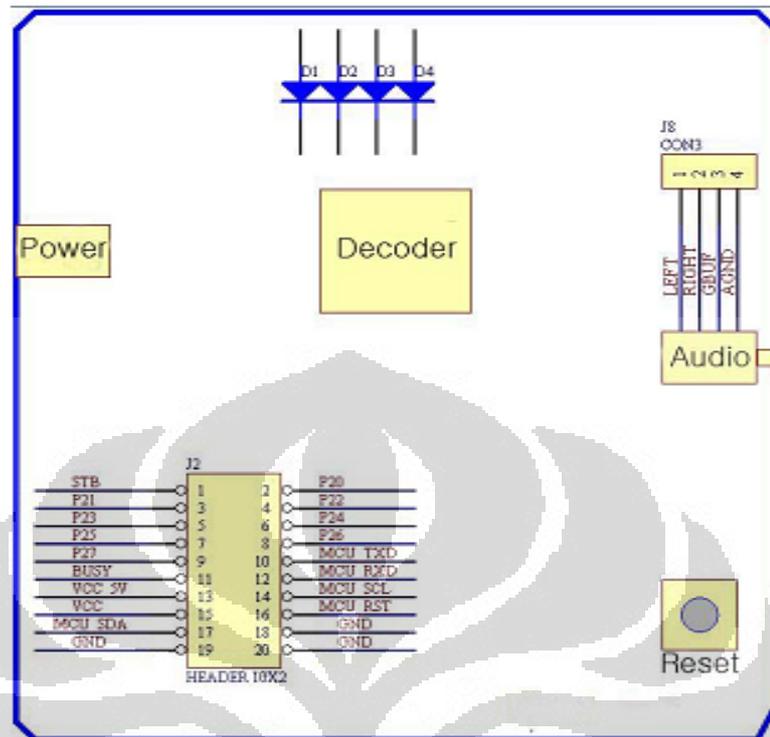
Pada blok kedua, mikrokontroller berfungsi sebagai media untuk menyimpan program yang telah dirancang pada komputer sekaligus menerima inputan yang dikirim oleh blok pertama. Mikrokontroller yang digunakan yaitu ATMEGA128 CPU Modul (128kB *Flash Memory* & 4kB *EEPROM*), yang juga mampu memberikan catu daya tegangan yang dimilikinya untuk kemudian digunakan oleh modul suara. DT-AVR ATMEGA128 CPU Modul merupakan modul mikrokontroller berbasis mikrokontroller AVR ATmega64L atau ATmega128L yang memiliki kemampuan dan konektor untuk *In-System Programming* (ISP).

Modul ini dapat dihubungkan ke *DT-COMBO BASE BOARD* seri dan digunakan untuk aplikasi sederhana hingga aplikasi yang cukup kompleks. Catu daya tegangan yang dibutuhkan oleh Mikrokontroler yaitu VCC 2,7-5,5volt DC. Setelah mikrokontroler menerima inputan dari komputer, alat ini bertugas meneruskannya kepada blok modul suara yang kemudian diproses menjadi bentuk suara. Berikut pada **Gambar 3.2** adalah mikrokontroler ATMega 128 :



Gambar 3.2 Mikrokontroler ATMEGA 128

Pada blok ketiga, modul suara menyimpan database berupa suara yang direkam dengan format MP3, WMA, WAV, MIDI. Pada modul ini terdapat mic yang berfungsi untuk merekam suara. Database suara tersebut disimpan pada *memory card / SD Card* dan maksimal kapasitasnya adalah 2GB. Modul dapat dikontrol oleh MCU atau PLC controller untuk memuaskan banyak aplikasi. Modul suara membutuhkan catu daya tegangan DC sebesar 5 volt dalam hal ini catu daya diberikan dari mikrokontroler sebesar 5 volt, sehingga tidak memerlukan rangkaian catu daya sendiri. Ketika modul suara menerima perintah atau inputan dari mikrokontroler, alat ini bertugas mencari database yang dimilikinya dan kemudian dikonversi dari signal digital ke signal analog lalu outputnya diteruskan kepada speaker. Rangkaian modul suara ditunjukkan oleh **Gambar 3.3** berikut ini :



Gambar 3.3 Struktur & Pin Modul Suara

Pada **Gambar 3.3** pin strobe digunakan untuk *trigger*, sedangkan p20-p27 digunakan untuk data paralel 8 bit. MCU TXD merupakan sinyal serial transmit, level TTL. Lalu pada pin busy ini akan aktif low jika file mp3 sedang di jalankan. MCU RST, pin reset untuk modul suara, minimal low 10 ms. Pada pin D4 sebagai LED indikator daya, D1 sebagai LED indikator *memory card*. Setelah *power* dalam kondisi *on* dan LED ini pun menyala, berarti *memory card* sudah dalam kondisi siap. Namun, jika lampu LED ini tidak menyala maka harus di cek kembali kondisi *memory card* tersebut agar file rekaman bisa dibaca ketika program sedang berjalan. D2 sebagai LED indikator MCU. Jika MCU sedang bekerja maka LED akan menyala. D3 sebagai indikator *BUSY*, ketika pin *BUSY* dalam kondisi *low*, the LED akan menyala, berarti sistem sedang bekerja dan tidak dapat menerima perintah baru. Sedangkan ketika pin *Busy* dalam kondisi High, berarti sistem sedang tidak bekerja dan LED akan mati.

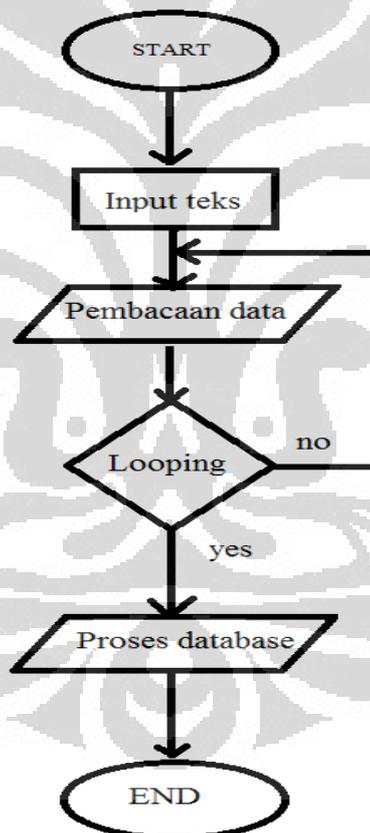
Fungsi dekoder adalah alat yang digunakan untuk mengembalikan suatu informasi yang telah diacak. Pada mikroprosesor Dekoder merupakan rangkaian logika yang digunakan untuk mengkodekan. Dekoder diaplikasikan pada bus alamat (*address bus*) sehingga berbagai rangkaian pendukung mikroprosesor (*input/output*) dapat

dialamati. Dekoder dapat disusun dari gerbang-gerbang dasar (*Not, AND, OR*) atau berupa IC dekode dalam satu kemasan.

Lalu pada blok terakhir, speaker bertugas sebagai media output suara yang diterima dari hasil yang sudah diproses oleh modul suara. Speaker yang digunakan adalah speaker stereo yang umumnya dipakai pada perangkat PC atau komputer.

3.2 Perancangan Program

Pada pembentukan program *text to voice with sad condition*, dibutuhkan rancangan diagram alirnya agar konsep program yang diinginkan terbentuk. Berikut **Gambar 3.4** adalah diagram alir untuk program *text to voice with sad condition* :

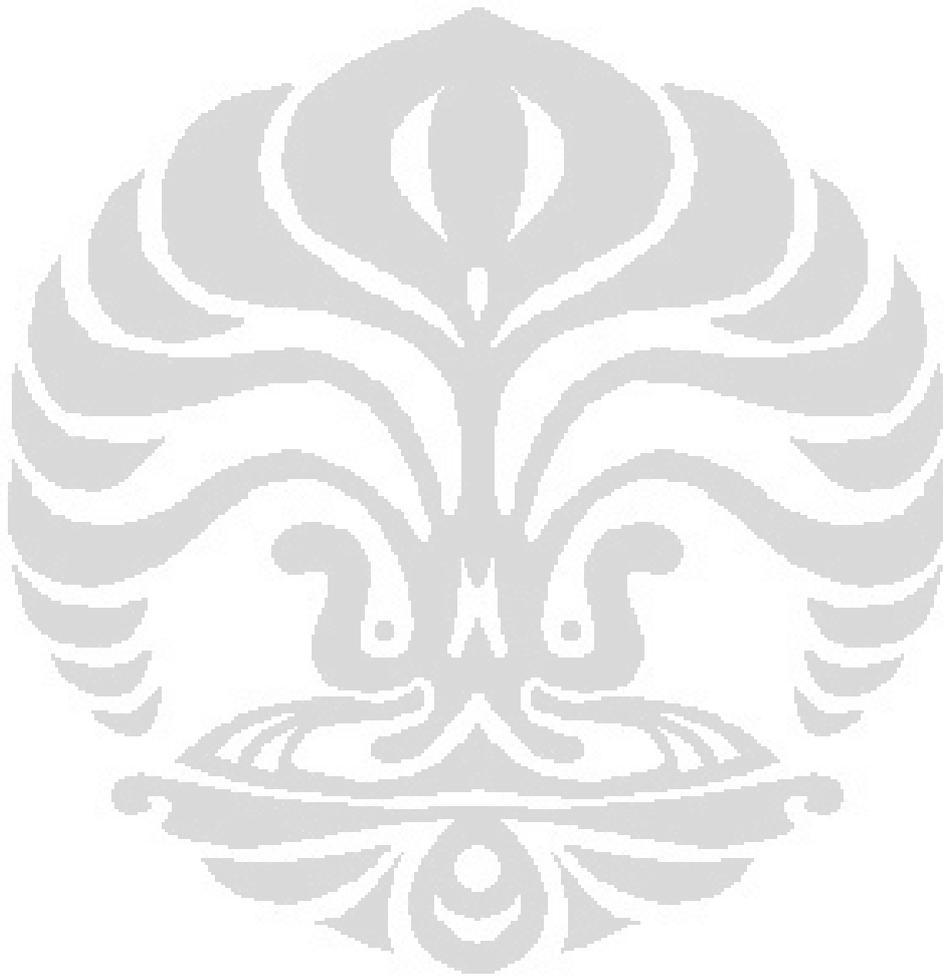


Gambar 3.4 Diagram Alir Program

Pada diagram alir awal, inputan teks dimasukkan secara manual oleh user atau diketik melalui PC atau Laptop. Media tampilan untuk inputannya bisa berupa *Hyperterminal* atau *Delphi*. Kemudian data inputan dibaca oleh media mikrokontroller yang sudah dimasukan program bahasa C, dan diteruskan ke

modul suara sebagai media penyimpan database untuk mencari file yang dimasukkan.

Jika file yang diinput berhasil ditemukan pada database, maka file tersebut akan diproses dan kemudian akan dikonversi ke bentuk suara. Namun jika file tidak ditemukan dalam database, maka akan kembali ke menu input untuk melanjutkan inputan teks yang berikutnya. Setelah penentuan flow chart terbentuk, barulah diaplikasikan ke dalam program bahasa C.



BAB IV

Pengujian dan analisa data

4.1 Deskripsi Pengujian

Nama Pengujian : Pengujian program konversi *Text to Voice with Sad Condition*

Tujuan Pengujian : Mengetahui apakah konversi teks ke suara telah bekerja sesuai dengan program yang telah direncanakan

Target Pengujian : Menguji apakah fungsi Mikrokontroller dan modul suara bekerja sesuai dengan perencanaan pada program. Mendapatkan data lengkap dan dapat dianalisa.

Data Lingkungan Pengujian

- a. Lokasi : Laboratorium Elektronika
- b. Tanggal Pelaksanaan : 22 Desember 2011 s.d 23 Desember 2011
- c. Pelaksana : Tendy Arya Pranata

4.2 Pengujian Program

Pengambilan data pada percobaan pengujian program yaitu data sinkronisasi yang terjadi antara mikrokontroller sebagai media aplikasi dari program dan modul suara sebagai penyimpan database dan konversi teks. Pengujian dilakukan sebanyak 5 kali dengan memasukkan data teks pada perangkat lunak komunikasi serial (XCTU) berupa kalimat yang diinput manual oleh user. Kemudian dari kelima inputan teks tersebut, mikrokontroller akan mencoba memproses data yang dimasukkan oleh user sesuai dengan ketentuan program yang telah dibentuk. Lalu mikrokontroller mengirimkan perintah atau *command* dengan menggunakan kode ASCII kepada modul suara dengan maksud agar modul suara mencari database yang diminta oleh user dan kemudian memberikan respon apakah data tersebut ditemukan dan tidak ditemukan atau urutan dari database yang sudah ada tidak sesuai dengan output yang diinginkan oleh user tersebut.

Berikut adalah beberapa percobaan yang diambil pada komunikasi serialnya :

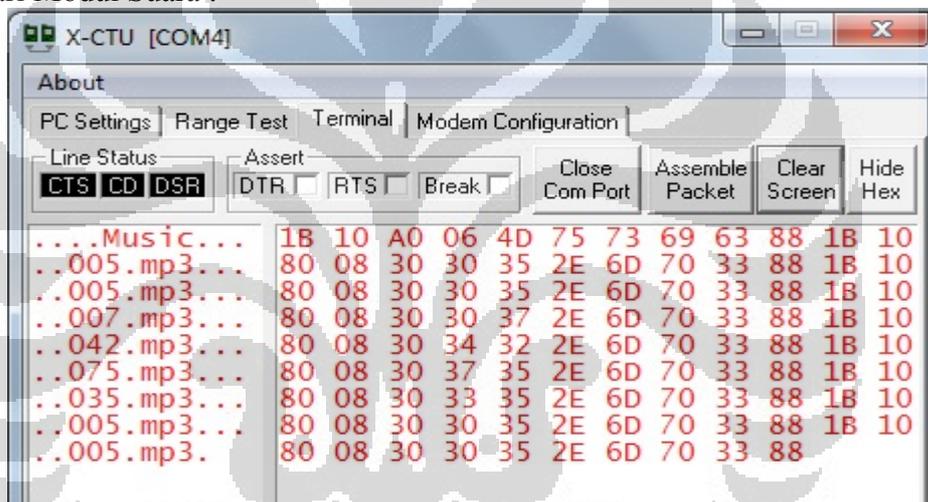
Percobaan 1

Data dikirim dari Mikrokontroler berupa teks “Baca buku saja”:

```
-----  
HARDWARE TEXT TO SPEECH.  
general purpose.  
-----  
masukan kata tanpa spasi : baca buku saja  
fonem : ba.  
urutan : 5.  
fonem : ca.  
urutan : 10.  
fonem : bu.  
urutan : 7.  
fonem : ku.  
urutan : 42.  
fonem : sa.  
urutan : 75.  
fonem : ja.  
urutan : 35.  
fonem : ba.  
urutan : 5.  
fonem : ca.  
urutan : 10.  
-----
```

Gambar 4.1 Data percobaan 1

Data dari Modul Suara :



Gambar 4.2 Data percobaan 1 dari modul suara

Percobaan 2

Data input dari mikrokontroler berupa teks “Kami suka bola”:

```
-----  
HARDWARE TEXT TO SPEECH.  
general purpose.  
-----  
masukan kata tanpa spasi : kami suka bola  
fonem : ka.  
urutan : 40.  
fonem : mi.  
urutan : 51.  
fonem : su.  
urutan : 77.  
fonem : ka.  
urutan : 40.  
fonem : bo.  
urutan : 9.  
fonem : la.  
urutan : 45.  
-----
```

Gambar 4.3 Data percobaan 2 dari mikrokontroler

Data dari Modul Suara :

```

..Music..... A0 06 4D 75 73 69 63 88 1B 10 80 08
040.mp3..... 30 34 30 2E 6D 70 33 88 1B 10 80 08
051.mp3..... 30 35 31 2E 6D 70 33 88 1B 10 80 08
077.mp3..... 30 37 37 2E 6D 70 33 88 1B 10 80 08
040.mp3..... 30 34 30 2E 6D 70 33 88 1B 10 80 08
009.mp3..... 30 30 39 2E 6D 70 33 88 1B 10 80 08
045.mp3.|    30 34 35 2E 6D 70 33 88

```

Gambar 4.4 Data percobaan 2 dari modul suara

Percobaan 3

Data input dari mikrokontroller berupa teks “Beli sepatu baru di toko”:

```

HARDWARE TEXT TO SPEECH.
general purpose.
-----
masukan kata tanpa spasi : beli sepatu baru di
toko
fonem : be.
urutan : 8.
fonem : li.
urutan : 46.
fonem : se.
urutan : 78.
fonem : pa.
urutan : 60.
fonem : tu.
urutan : 82.
fonem : ba.
urutan : 5.
fonem : ru.
urutan : 72.
fonem : di.
urutan : 16.
fonem : to.
urutan : 84.
fonem : ko.
urutan : 44.
-----

```

Gambar 4.5 Data percobaan 3 dari mikrokontroller

Data pada Modul Suara :

```

....Music... 1B 10 A0 06 4D 75 73 69 63 88 1B 10
..008.mp3... 80 08 30 30 38 2E 6D 70 33 88 1B 10
..046.mp3... 80 08 30 34 36 2E 6D 70 33 88 1B 10
..078.mp3... 80 08 30 37 38 2E 6D 70 33 88 1B 10
..060.mp3... 80 08 30 36 30 2E 6D 70 33 88 1B 10
..082.mp3... 80 08 30 38 32 2E 6D 70 33 88 1B 10
..005.mp3... 80 08 30 30 35 2E 6D 70 33 88 1B 10
..072.mp3... 80 08 30 37 32 2E 6D 70 33 88 1B 10
..016.mp3... 80 08 30 31 36 2E 6D 70 33 88 1B 10
..084.mp3... 80 08 30 38 34 2E 6D 70 33 88 1B 10
..044.mp3.| 80 08 30 34 34 2E 6D 70 33 88

```

Gambar 4.6 Data percobaan 3 dari modul suara

Percobaan 4

Data inputan dari mikrokontroller berupa teks “Kota Solo”:

```

-----
HARDWARE TEXT TO SPEECH.
general purpose.
-----
masukan kata tanpa spasi : kota solo
fonem : ko.
urutan : 44.
fonem : ta.
urutan : 80.
fonem : so.
urutan : 79.
fonem : lo.
urutan : 49.
fonem : ko.
urutan : 44.
fonem : ta.
urutan : 80.
fonem : so.
urutan : 79.
fonem : lo.
urutan : 49.
-----

```

Gambar 4.7 Data percobaan 4 dari mikrokontroller

Data dari modul suara :

....Music...	1B	10	A0	06	4D	75	73	69	63	88	1B	10
..044.mp3...	80	08	30	34	34	2E	6D	70	33	88	1B	10
..080.mp3...	80	08	30	38	30	2E	6D	70	33	88	1B	10
..079.mp3...	80	08	30	37	39	2E	6D	70	33	88	1B	10
..049.mp3...	80	08	30	34	39	2E	6D	70	33	88	1B	10
..044.mp3...	80	08	30	34	34	2E	6D	70	33	88	1B	10
..080.mp3...	80	08	30	38	30	2E	6D	70	33	88	1B	10
..079.mp3...	80	08	30	37	39	2E	6D	70	33	88	1B	10
..049.mp3.	80	08	30	34	39	2E	6D	70	33	88		

Gambar 4.8 Data percobaan 4 dari modul suara

Percobaan 5

Data dikirim dari Mikrokontroller berupa teks “Lari-lari di pagi hari”:

```

-----
HARDWARE TEXT TO SPEECH.
general purpose.
-----
masukan kata tanpa spasi : lari lari di pagi
hari
fonem : la.
urutan : 45.
fonem : ri.
urutan : 71.
fonem : la.
urutan : 45.
fonem : ri.
urutan : 71.
fonem : di.
urutan : 16.
fonem : pa.
urutan : 60.
fonem : gi.
urutan : 26.
fonem : ha.
urutan : 30.
fonem : ri.
urutan : 71.
-----

```

Gambar 4.9 Data percobaan 5 dari mikrokontroller

Data pada Modul Suara :

```

...Music... 1B 10 A0 06 4D 75 73 69 63 88 1B 10
..045.mp3... 80 08 30 34 35 2E 6D 70 33 88 1B 10
..071.mp3... 80 08 30 37 31 2E 6D 70 33 88 1B 10
..045.mp3... 80 08 30 34 35 2E 6D 70 33 88 1B 10
..071.mp3... 80 08 30 37 31 2E 6D 70 33 88 1B 10
..016.mp3... 80 08 30 31 36 2E 6D 70 33 88 1B 10
..060.mp3... 80 08 30 36 30 2E 6D 70 33 88 1B 10
..026.mp3... 80 08 30 32 36 2E 6D 70 33 88 1B 10
..030.mp3... 80 08 30 33 30 2E 6D 70 33 88 1B 10
..071.mp3..| 80 08 30 37 31 2E 6D 70 33 88

```

Gambar 4.10 Data percobaan 5 dari modul suara

4.3 Analisa data hasil

Ketika mikrokontroller berinteraksi dengan modul suara pada saat menjalankan program terjadi komunikasi pada serial antara keduanya. Data dibawah ini menunjukkan perintah atau *command* yang dikirimkan oleh mikrokontroller serta respon atau feedback dari modul suara dengan menggunakan kode ASCII.

Command : 0

Command : 1B 10 00 01 88

Feedback : 4F 4B 00 01 88

Pada perintah 0, serial dari mikrokontroller mengirimkan kode untuk mengetahui apakah koneksi untuk keduanya berhasil atau tidak. Lalu pada modul suara akan merespon kode tersebut, jika yang dikirim adalah 4F 4B 00 01 88, berarti koneksi sudah sukses. Namun jika yang dikirim kodenya berbeda, atau tidak dikirim sama sekali berarti koneksi antara keduanya gagal dan harus dicari kesalahan koneksi tersebut.

Command : A0

Contoh : 1B 10 A0 06 **4D 75 73 69 63** 88

Feedback : 4F 4B 00 01 88

Feedback : 45 52 A0 01 88

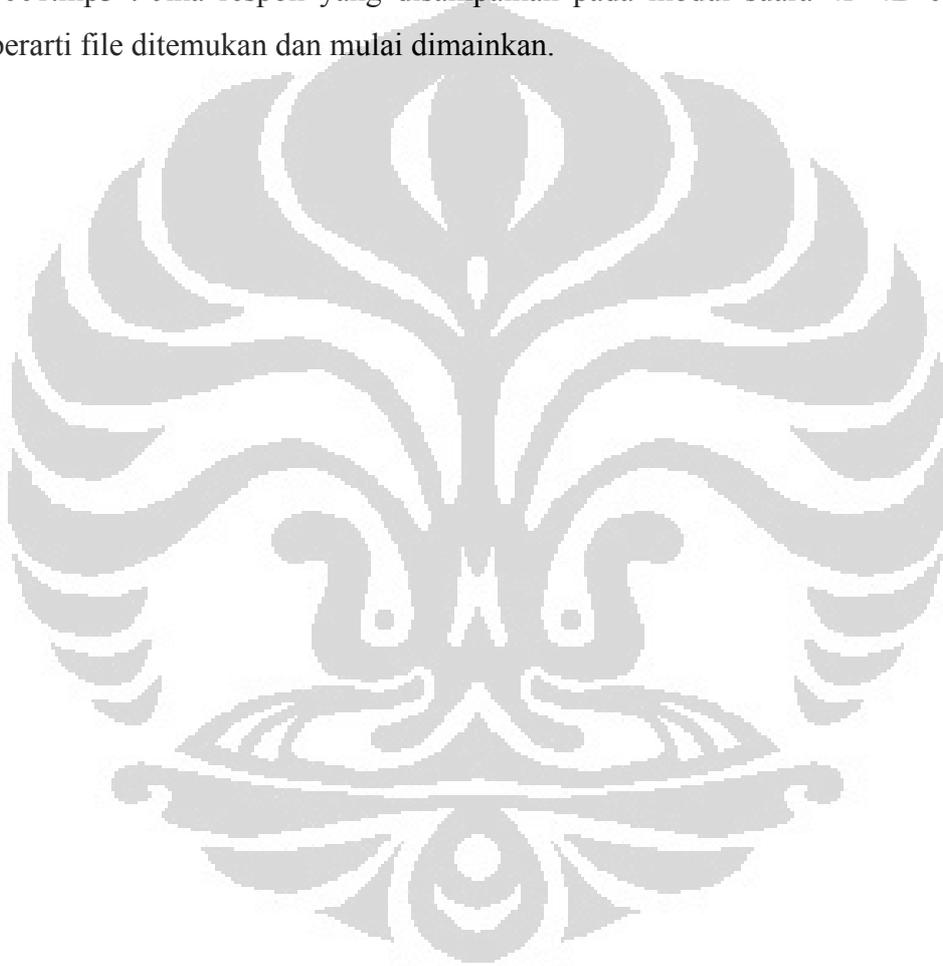
Perintah diatas berarti kode untuk mengakses folder dengan nama “*Music*” di *memory card*. Hal ini diketahui pada huruf dan angka yang dicetak tebal dengan bahasa ASCII yang berarti “*Music*”. Respon pertama menunjukkan masuk ke dalam folder berhasil, sedangkan untuk respon yang kedua berarti masuk ke dalam folder gagal. Jika gagal, maka kesalahan bisa saja terjadi pada penulisan nama folder atau kode ASCII, ataupun format memory card yang harus terlebih dahulu diubah kedalam format FAT.

Command 80 (Memutar spesifik *file* pada folder)

Contoh : 1B 10 80 08 **30 30 31 2E 6D 70 33** 88

Feedback : 4F 4B 00 01 88

Pada perintah 80 diatas berarti perintah untuk memutar file dengan menunjuk nomor secara spesifik “001.mp3” di dalam folder. Hal ini diketahui pada huruf dan angka yang dicetak tebal dengan bahasa ASCII yang berarti “001.mp3”. Jika respon yang disampaikan pada modul suara 4F 4B 00 01 88 berarti file ditemukan dan mulai dimainkan.



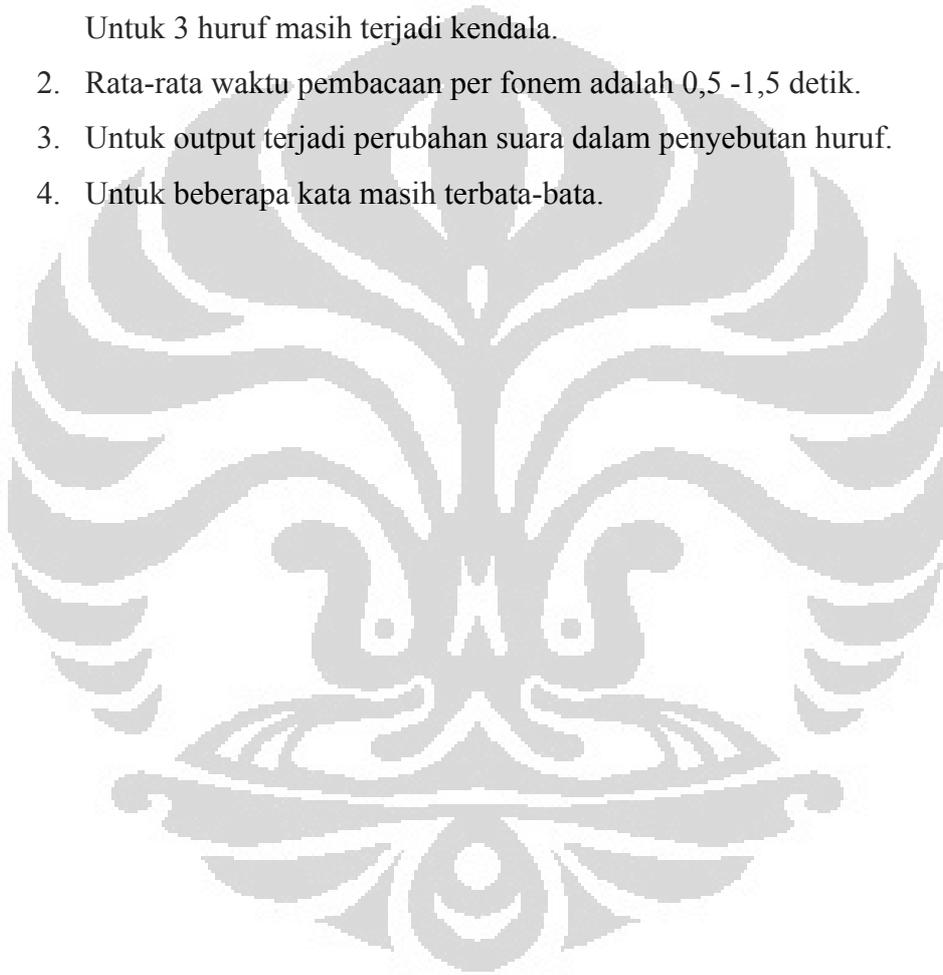
BAB V

Kesimpulan

5.1 Kesimpulan

Setelah melakukan perencanaan dan pembuatan sistem kemudian dilakukan pengujian dan analisisnya, maka dapat diambil beberapa kesimpulan yaitu sebagai berikut:

1. Fonem yang terbaca dengan baik hanyalah yang terdiri dari 1 atau 2 huruf.
Untuk 3 huruf masih terjadi kendala.
2. Rata-rata waktu pembacaan per fonem adalah 0,5 -1,5 detik.
3. Untuk output terjadi perubahan suara dalam penyebutan huruf.
4. Untuk beberapa kata masih terbata-bata.



Lampiran

Daftar kode ASCII

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	sp	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	
011 1101	075	61	3D	=	101 1101	135	93	5D]	111 1101	175	125	7D	}
011 1110	076	62	3E	>	101 1110	136	94	5E	^	111 1110	176	126	7E	~
011 1111	077	63	3F	?	101 1111	137	95	5F	_					

Tabel Data Base Kata

Urutan	Fonem	
0000'	A	
001'	I	
002'	U	
003'	E	
004'	O	
005'	B	A
006'	B	I
007'	B	U
008'	B	E
009'	B	O
010'	C	A
011'	C	I
012'	C	U
013'	C	E
014'	C	O
015'	D	A
017'	D	I
018'	D	U
019'	D	E
020'	D	O
021'	F	A
022'	F	I
023'	F	U
024'	F	E
025'	F	O
026'	G	A
027'	G	I
028'	G	U
029'	G	E
030'	G	O
031'	H	A
032'	H	I
033'	H	U
034'	H	E
035'	H	O
036'	J	A
037'	J	I
038'	J	U
039'	J	E
040'	J	O
041'	K	A
042'	K	I
043'	K	U

044'	K	E
045'	K	O
046'	L	A
047'	L	I
048'	L	U
049'	L	E
050'	L	O
051'	M	A
052'	M	I
053'	M	U
054'	M	E
055'	M	O
056'	N	A
057'	N	I
058'	N	U
059'	N	E
060'	N	O
061'	P	A
062'	P	I
063'	P	U
064'	P	E
065'	P	O
066'	R	A
067'	R	I
068'	R	U
069'	R	E
070'	R	O
071'	S	A
072'	S	I
073'	S	U
074'	S	E
075'	S	O
076'	T	A
077'	T	I
078'	T	U
079'	T	E
080'	T	O
081'	V	A
082'	V	I
083'	V	U
084'	V	E
085'	V	O
086'	W	A
087'	W	I
088'	W	U

089'	W	E
090'	W	O
091'	X	A
092'	X	I
093'	X	U
094'	X	E
095'	X	O
096'	Y	A
097'	Y	I
098'	Y	U
099'	Y	E
100	Y	O
101	Z	A
102	Z	I
103	Z	U
104	Z	E
105	Z	O
106	B	
107	C	
108	D	
109	F	
110	G	

111	H	
112	J	
113	K	
114	L	
115	M	
116	N	
117	P	
118	Q	
119	R	
120	S	
121	T	
122	V	
123	W	
124	X	
125	Y	
126	Z	
127	NY	A
128	NY	I
129	NY	U
130	NY	E
131	NY	O