TUGAS AKHIR

PENGUKUR BERAT DAN TINGGI BADAN DENGAN OUTPUT SUARA

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Strata-1
Teknik Elektro

Oleh

Nama : Yulfitra Yunis

No. Mahasiswa : 02 524 059

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011

LEMBAR PENGESAHAN PEMBIMBING

PENGUKUR BERAT DAN TINGGI BADAN DENGAN OUTPUT SUARA

TUGAS AKHIR



Disusun oleh:

Nama : Yulfitra Yunis

No. Mahasiswa : 02524059

Yogyakarta, Agustus 2011

Pembimbing I,	Pembimbing II
Dwi Anna Ratnawati,ST. M.Eng.	Ir.Hj.Budi Astuti, MT.

LEMBAR PENGESAHAN PENGUJI

PENGUKUR BERAT DAN TINGGI BADAN DENGAN OUTPUT SUARA

TUGAS AKHIR

Oleh :

Nama : Yulfitra Yunis

No. Mahasiswa : 02524059

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk Memeperoleh Gelar Sarjana Teknik Elektro

Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 16 Agustus 2011

Tim Penguji,

Ir.Hj.Budi Astuti, MT

Ketua

Wahyudi Budi Pramono,ST. M.Eng

Anggota I

Medilla Kusriyanto,ST. M.Eng

Anggota II

Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia

Eko Yuwono, ST., M.Sc.

HALAMAN PERSEMBAHAN

Kupersembahkan untuk…

Kedua orang tuaku,
yang selalu memberi pelajaran tentang makna kehidupan dan selalu mengingatkan untuk bersujud pada-Nya. Doa, kasih sayang, pengorbanan serta nasehat yang tiada hentinya mengalir untukku, membuatku yakin bisa menghadapi segalanya dengan ikhlas

Adik-adikku,
yang memberikan dorongan, semangat, doa dan kasih sayangnya

Om dan tanteku
yang memberikan semangat, doa, perhatian, dan nasehat yang tak henti-hentinya

MOTTO

"Jadikanlah sabar dan shalat sebagai penolongmu. Dan sesungguhnya yang demikian itu sungguh berat, kecuali bagi orang-orang yang khusyu"

(Al Baqarah : 45)

"Bertaqwalah kepada Allah, maka Allah akan mengajarimu ilmu"

(Al Baqarah : 282)

"Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia"

(HR. Tirmidzi)

"Lakukan segala hal yang baik dengan ikhlas niscaya kita akan menuai kebaikan"

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji bagi Allah, Rabb alam semesta. Shalawat dan salam semoga terlimpahkan kepada Rassulullah *Shallallahu 'Alaihi Wassallam*, keluarganya, sahabatnya dan pengikutnya hingga akhir zaman.

Sesungguhnya atas petunjuk, pertolongan dan bimbingan-Nya maka Tugas Akhir ini dapat terselesaikan dengan baik. Tugas Akhir ini merupakan salah satu syarat untuk menyelesaikan jenjang studi Strata 1 Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Keberhasilan terselesaikannya Tugas akhir ini tidak terlepas dari dukungan berbagai pihak. Oleh karena itu dengan rasa hormat dan terima kasih yang sebesar-besarnya penulis sampaikan kepada :

1. Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Ketua Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Ibuk Dwi Anna Ratnawati, ST. M.Eng selaku dosen pembimbing I yang telah memberikan bantuan dan arahannya dalam penyusunan tugas akhir ini.
4. Ibuk Ir.Hj.Budi Astuti, MT selaku dosen pembimbing II yang telah memberikan bantuan dan arahannya dalam penyusunan tugas akhir ini.

5. Segenap Dosen dan Karyawan Fakultas Teknologi Industri, khususnya jurusan Teknik Elektro atas segala dedikasinya dalam memberikan ilmu kepada penulis serta memberikan bantuan dalam segala hal.
6. Ayah, Ibu, Om, Tante, dan Adik - adik tercinta yang selalu memberikan perhatian, do'a dan dorongan semangat kepada penulis.
7. Teman – teman seperjuangan Teknik Elektro Universitas Islam Indonesia .
8. Keluarga Besar Koperasi Mahasiswa Universitas islam Indonesia.
9. Keluarga Besar Himpunan Anak Minangkabau Universitas Islam Indonesia.
10. Pihak – pihak lain yang tidak mungkin penulis sebutkan satu per satu, terima kasih atas perhatian dan dukungannya.

Semoga Allah membalas semua jasa-jasanya yang diberikan pada penulis. Harapan penulis semoga Tugas Akhir ini dapat bermanfaat bagi kita semua. Amin.

Wassalamu'alaikum Wr. Wb

Yogyakarta, Agustus 2011

Penulis

ABSTRAKSI

Dalam kehidupan sehari-hari, manusia perlu memperhatikan kesehatannya. Salah satu yang harus diperhatikan adalah mengenai masalah berat badan yang ideal atau tidak. Apabila berat badan tidak ideal, kekurangan atau kelebihan, maka akan menimbulkan penyakit dalam tubuh. Faktor yang mempengaruhi berat badan itu disebut ideal atau tidak adalah tinggi badan.

Apabila berat badan tidak ideal, kekurangan atau kelebihan, maka akan menimbulkan penyakit dalam tubuh. Oleh karena itu manusia perlu memantau berat badannya. Faktor yang mempengaruhi berat badan itu ideal adalah tinggi badan. Untuk mengukur tinggi badan seseorang, perlu posisi berdiri yang sempurna agar dapat diketahui tinggi badan yang sesuai. Dengan adanya hal tersebut maka pada proses pengukuran dibutuhkan bantuan orang lain untuk melihat hasil pengukuran tinggi dan berat badan yang dihasilkan pada LCD. Oleh karena itu, pada tugas akhir ini direalisasikan suatu pengembangan dari alat pengukur berat dan tinggi tersebut agar dapat mengeluarkan suara. Selain itu penyandang tunanetra juga dapat menggunakan alat ini. Alat ini menggunakan sensor ultrasonik jenis SFR02 untuk pengukur tinggi badan, potensio geser untuk pengukur berat badan, mikrokontroller ATMEGA16 untuk perhitungan, LCD untuk menampilkan hasil pengukuran, dan IC ISD 1420 untuk penyimpanan suara yang telah direkam sebelumnya.

Alat ini menampilkan hasil pengukuran berat badan, tinggi badan,serta dapat mengeluarkan suara hasil pengukuran tersebut. Pengguna alat ini akan berdiri dengan posisi yang sempurna dan dalam beberapa detik pengguna akan mendengar hasil pengukuran berupa berat badan, tinggi badan, dan ideal atau tidaknya berat badan pengguna. Tingkat kesalahan pada alat ini adalah 2,77 % untuk mengukur berat badan dan 1,5 % untuk mengukur tinggi badan (dilakukan dalam tujuh kali percobaan).

Kata kunci : berat, tinggi, LCD, ISD 1420, ATMEGA16, SFR02

DAFTAR ISI

JUDUL……..……..i

LEMBAR PENGESAHAN PEMBIMBING…….…..ii

LEMBAR PENGESAHAN PENGUJI…….. iii

HALAMAN PERSEMBAHAN ……..…….iv

MOTTO ……...……...v

KATA PENGANTAR ……..……....vi

ABSTRAKSI …….…….……viii

DAFTAR ISI ……..……..ix

DAFTAR GAMBAR ……..……xiii

DAFTAR TABEL …….…….…… xv

BAB I PENDAHULUAN.……...1

1.1 Latar Belakang …….…..……1

1.2 Rumusan Masalah ……..…….1

1.3 Batasan Masalah ……..……..….2

1.4 Tujuan Penelitian ……..…..2

1.5 Sistematika Penulisan ……..……...….3

BAB II TINJAUAN PUSTAKA ……..5

2.1 Ultrasonik ……..……...……5

2.1.1 Suara Ultrasonik ……..……..…….5

2.1.2 Sensor Ultrasonik ……....……...6

2.2 Mikrokontroller ATMEGA16 ……..……9

2.2.1 Konfigurasi Pin ATMEGA16 ……..……10

2.2.2 ADC Mikrokontroller ATMEGA 16 ……..…….12

2.3 Positive Voltage Regulator LM78XX ……..…..…15

2.4 Voice Record / Playback Strorange Device ISD 1420 ……..……...17

2.4.1 Struktur PIN …….……...18

2.4.2 Sistem Pengalamatan ……..…….....….21

2.5 Potensio Geser ……...…..……...21

BAB III PERANCANGAN ……...……23

3.1 Gambaran Umum Sistem ……...…….23

3.2 Perancangan Perangkat Keras ……..……..….24

3.2.1 Power Supply ……..……..…24

3.2.2 Rangkaian Sensor Berat ……...……24

3.2.3 Rangkaian SRF01 Sensor Jarak (Ultrasonik) …….……..25

3.2.4 Mikrokontroller ATMEGA16 ……...……...26

3.2.4.1 Osilator ……...……...26

3.2.4.2 Reset ……..……...26

3.2.5 Rangkaian ISD 1420 ……..…….27

3.2.6 Rangkaian LCD ……..…….……32

3.2.7 Rangkaian Tombol ……..……..…..27

3.3 Perancangan Perangkat Lunak ……...…….33

3.3.1 Setting Code Vision ……..……...33

3.3.2 Inisialisasi ……...……..……37

3.3.3 Program Utama ……...…….38

3.3.4 Subrutin Baca Tinggi ……..……39

3.3.5 Subrutin Baca Berat ……..……..…40

BAB IV PENGUJIAN DAN ANALISIS KINERJA SISTEM …….42

4.1 Metode Pengujian ……..…….42

4.2 Pengujian Fungsional ……...……...42

4.2.1 Pengujian Rangkaian Sensor Berat ……..……43

4.2.2 Rangkaian Ultrasonik SFR02 ……..…..44

4.2.3 Rangkaian Tombol ……...……44

4.2.4 Rangkaian Reset Mikrokontroller ……...…….45

4.2.5 Pengujian Power Supply ……...…...45

4.3 Pengujian Dan Pembahasan Sistem Secara Keseluruhan ……..…….46

4.3.1 Pengukuran Berat Badan ……..…47

4.3.2 Pengukuran Tinggi Badan ……..…..…….47

BAB V PENUTUP ……..…….49

5.1 Kesimpulan ……..…...……49

5.2 Saran ……..…….49

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Beberapa Jenis Sensor Ultrasonik ……..…..….6

Gambar 2.2 Prinsip Kerja sensor Ultrasonik ……...……..…6

Gambar 2.3 Pin ATMEGA16 …….....10

Gambar 2.4 ADCSRA ……..……...….12

Gambar 2.5 Register ADMUX ……..……...13

Gambar 2.6 Setting ADLAR ……..…….....13

Gambar 2.7 Register SFIOR ……..…….…….14

Gambar 2.8 Rangkaian dasar Regulator LM78XX ……..17

Gambar 2.9 Diagram Blok ISD 1420 ……...…17

Gambar 2.10 Diagram Pin ISD 1420 ……...…….18

Gambar 2.11 Potensio Geser ……...……..22

Gambar 3.1 Blok Diagram Sistem ……..…….23

Gambar 3.2 Rangkaian Power Supply ……...……...…24

Gambar 3.3 Rangkaian Sensor Berat ……..…….25

Gambar 3.4 Rangkaian Sensor Jarak SRF02 ……...26

Gambar 3.5 Rangkaian Minimum ATMEGA16 ……...…27

Gambar 3.6 Rangkaian Rekam ISD 1420 ……..28

Gambar 3.7 Rangkaian Hubungan ISD 1420 dengan Mikrokontroller …...……31

Gambar 3.8 Rangkaian LCD …… ….……..…32

Gambar 3.9 Rangkaian Tombol ……. 33

Gambar 3.10 Setting Code Vision untuk Pemilihan Mikrokontroller ……..... 34

Gambar 3.11 Setting Code Vision untuk Fungsi Port I/O Mikrokontroller……. 35

Gambar 3.12 Pngaturan Komunikasi Serial Mikrokontroller pada Code Vision..35

Gambar 3.13 Setting ADC pada Code Vision ……... 36

Gambar 3.14 Setting kode Vision untuk Penggunaan LCD …….. 37

Gambar 3.15 Flowchart Program Utama ……... 38

Gambar 3.16 Flowchart Subrutin Baca Tinggi …….. 39

Gambar 3.17 Flowchart Subrutin Baca Berat ……..…… 41

Gambar 4.1 Tampilan Awal pada LCD ……..… 46

DAFTAR TABEL

Tabel 2.1 Command SFR02 ……..…… 8

Tabel 2.2 Konfigurasi Pin ATMEGA16 ……..…...…… 11

Tabel 2.3 Pemilihan Sumber Picu ADC ……..... 14

Tabel 2.4 Karakteristik Elektrik IC Regulator Keluarga LM78XX ……....16

Tabel 3.1 Alamat dan Waktu Perekaman ke ISD 1420 …….. 30

Tabel 4.1 Hasil Pengukuran sensor Berat Badan …… 43

Tabel 4.2 Hasil Uji Rangkaian Ultrasonik …… .44

Tabel 4.3 Hasil Uji Rangkaian Tombol ……. 45

Tabel 4.4 Hasil Pengukuran Tegangan Power Supply …….……46

Tabel 4.5 Hasil Pengukuran Berat Badan ……. ...47

Tabel 4.6 Hasil Pengukuran Tinggi Badan ……... ..48

UNIVERSITAS ISLAM INDONESIA

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini perkembangan teknologi berkembang sangat pesat. Peralatan-peralatan yang dulunya menggunakan sistem analog, sekarang sudah mulai ditinggalkan dan digantikan dengan peralatan yang menggunakan sistem digital.

Manusia tidak henti-hentinya mengkaji dan meneliti teknologi elektronika termasuk dalam bidang analisis dan pengukuran. Dalam dunia kedokteran kita mengenal sebuah alat untuk mengukur berat dan tinggi badan yang masih berupa analog. Dokter harus mendekati orang yang mengukur berat atau tingginya yang terkadang merepotkan.

Dengan latar belakang permasalahan seperti diatas maka dibuat tugas akhir mengenai pengukur berat dan tinggi badan dengan output suara yang keluar dari alat tersebut. Alat ini juga dapat digunakan oleh orang cacat, misalnya orang buta. Dengan alat ini dia dapat mengetahui tinggi dan berat badannya dengan mendengar suara yang dikeluarkan alat tersebut.

1.2. Rumusan Masalah

Untuk mewujudkan tercapainya pembuatan alat yang dapat digunakan dalam kehidupan sehari-hari, dapat diidentifikasikan sebagai berikut:

1. Bagaimana merancang dan membuat pengukur berat badan dan tinggi badan dengan output suara.
2. Bagaimana unjuk kerja pengukur berat badan dan tinggi badan dengan output suara.
3. Bagaimana efektifitas penggunaan sensor berat badan dan sensor tinggi badan yang digunakan.

1.3. Batasan Masalah

Agar dalam penulisan penelitian ini lebih terarah, maka pembahasan penulisan ini dibatasi pada ruang lingkup pembahasan sebagai berikut:

1. Sistem menggunakan mikrokontroller ATMEGA16 sebagai pemroses utama.
2. Sistem berfungsi untuk mengukur berat dan tinggi badan seseorang dengan tampilan digital dan suara.
3. Sistem menggunakan sensor berat jenis modifikasi timbangan badan analog untuk mengukur berat badan, dimana sensor ini merupakan jenis transduser yang akan langsung menghasilkan nilai tegangan dari perubahan tekanan yang akan membebani sensor tersebut.
4. Sistem menggunakan tranduser ultrasonik untuk mengukur tinggi badan.

1.4. Tujuan Penelitian

Tujuan dan manfaat yang akan dicapai dalam penulisan penelitian ini adalah sebagai berikut:

1. Dapat mengukur dan merealisasikan suatu peralatan yang dapat mengukur berat dan tinggi badan seseorang dengan berbasis mikrokontroller ATMEGA16 dan perangkat-perangkat pendukung lainnya.
2. Mengetahui unjuk kerja dari alat pengukur berat badan dan pengukur tinggi badan dengan output suara.

1.5 Sistematika Penulisan

Dari hasil penelitian yang telah dilaksanakan, sistematika penulisan laporannya adalah sebagai berikut:

BAB I PENDAHULUAN

Berisi tentang Latar Belakang Masalah, Maksud dan Tujuan Masalah, Batasan Masalah, Perumusan Masalah, Metode Penelitian, serta Sistematika Penulisan pada Laporan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Berisi tentang penelitian yang telah dilakukan sebelumnya dan penjelasan secara terperinci mengenai teori-teori yang digunakan sebagai landasan untuk pemecahan masalah.

BAB III PERANCANGAN SISTEM

Berisi pembahasan tentang perancangan perangkat keras yang meliputi sensor berat badan dan tinggi badan, *mikrokontroler* sebagai kendali, rangkaian penyimpan suara, dan LCD.

BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN

Bagian ini menjelaskan tentang pengujian dan analisa terhadap hasil pengamatan dari sistem yang telah dibuat.

BAB V PENUTUP

Membahas tentang kesimpulan dan saran-saran yang menunjang agar alat tersebut dapat berkerja lebih optimal, sehingga tugas akhir ini dapat dikembangkan lebih lanjut, dengan harapan dapat digunakan untuk mendukung penerapannya dalam kehidupan masyarakat.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

2.1 Ultrasonik

2.1.1 Suara Ultrasonik

Suara merupakan salah satu hal yang tidak dapat dipisahkan dari manusia, karena manusia mempunyai indera pendengaran yang berfungsi untuk mengetahui suara disekitarnya. Suara termasuk gelombang longitudinal, yaitu gelombang yang arah getarnya searah dengan perambatannya. Suara juga termasuk gelombang mekanik, yaitu gelombang yang memerlukan media perambatan.

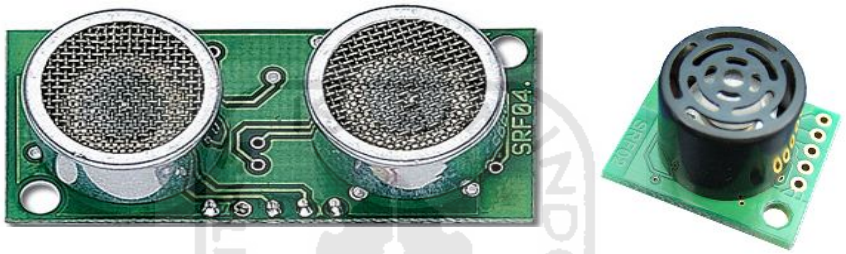
Menurut frekuensinya suara dapat dibedakan menjadi tiga, yaitu :

1. Suara infrasonik : frekuensi kurang dari 20 KHz
2. Suara audio : frekuensi antara 20 Hz – 20 Khz (bisa didengar manusia).
3. Suara ultrasonik : frekuensi diatas 20 KHz.

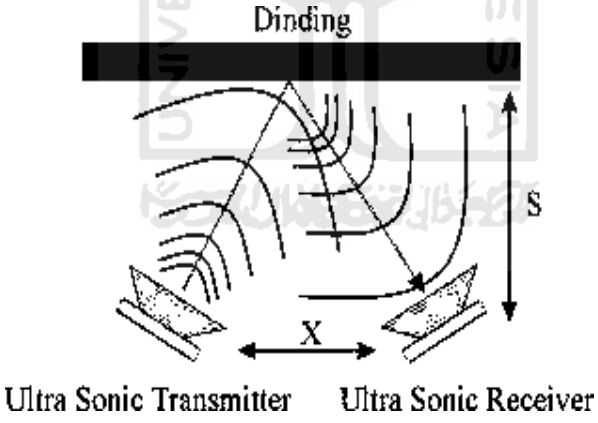
Suara ultrasonik merupakan suara dengan frekuensi diatas 20 KHz, tepatnya 40 Khz yang harus dipancarkan *transmitter*. Frekuensi ini tidak dapat didengar manusia sehingga dapat dimanfaatkan tanpa mengganggu indera pendengaran.

2.1.2 Sensor Ultrasonik

Sensor merupakan alat yang mengubah besaran kuantitatif yang ada dilingkungan menjadi besaran-besaran elektris yang dapat dibaca oleh rangkaian elektronik. Prinsip yang digunakan pada sensor ultrasonik ini adalah berupa pantulan suara, getaran suara 40 KHz dihasilkan dan dipancarkan oleh *transmitter* dan pantulan suara tersebut diterima oleh *receiver*.



Gambar 2.1 Beberapa jenis sensor ultrasonik



Gambar 2.2 Prinsip kerja sensor ultrasonik

Jeda waktu antara pemancaran getaran suara ultrasonik dan penerimaan getaran suara pantul sebanding dengan jarak antara sensor dan penghalang. Jika jarak

antara *transmitter* dan *reciever* cukup dekat, sudut pantulnya kecil, sehingga jarak ultrasonik dengan dinding (S) tidak terpengaruh sudut pantul. Jarak dapat diketahui dengan persamaan berikut :

$$S = \frac{k \cdot t}{2} \qquad \text{……..,,} \quad (2.1)$$

dengan S = jarak (m)

t = waktu (s)

k = konstanta kecepatan suara di udara (340 m/s)

SRF02 adalah kit sensor ultrasonik yang mudah digunakan, Feature SRF02 sebagai berikut.

- Modul sensor ultrasonik pengukur jarak dengan antarmuka I2C dan UART
- Catu daya 5 volt / 4 mA
- Frek kerja 40 kHz
- Range pengukuran 15 cm – 6 meter
- 1 Port UART dapat mengendalikan 16 modul
- Tanpa perlu kalibrasi, keluaran dapat dipilh satuannya (uS, cm, inch)

Command yang digunakan dapat dilihat pada tabel berikut :

Tabel 2.1 Command SRF02

Command		Action
Decimal	Hex	
80	0x50	Real Ranging Mode - Result in inches
81	0x51	Real Ranging Mode - Result in centimeters
82	0x52	Real Ranging Mode - Result in micro-seconds
83	0x53	Real Ranging Mode - Result in inches, automatically Tx range back to controller as soon as ranging is complete.
84	0x54	Real Ranging Mode - Result in centimeters, automatically Tx range back to controller as soon as ranging is complete.
85	0x55	Real Ranging Mode - Result in micro-seconds, automatically Tx range back to controller as soon as ranging is complete.
86	0x56	Fake Ranging Mode - Result in inches
87	0x57	Fake Ranging Mode - Result in centimeters
88	0x58	Fake Ranging Mode - Result in micro-seconds
89	0x59	Fake Ranging Mode - Result in inches, automatically Tx range back to controller as soon as ranging is complete.
90	0x5A	Fake Ranging Mode - Result in centimeters, automatically Tx range back to controller as soon as ranging is complete.
91	0x5B	Fake Ranging Mode - Result in micro-seconds, automatically Tx range back to controller as soon as ranging is complete.
92	0x5C	Transmit an 8 cycle 40khz burst - no ranging takes place
93	0x5D	Get software version - sends a single byte back to the controller
94	0x5E	Get Range, returns two bytes (high byte first) from the most recent ranging.
95	0x5F	Get Minimum, returns two bytes (high byte first) of the closest range measurable - see Autotune section

96	0x60	Force Autotune Restart - same as power-up. You can ignore this command.
160	0xA0	1st in sequence to change I2C address
165	0xA5	3rd in sequence to change I2C address
170	0xAA	2nd in sequence to change I2C address

2.2 Mikrokontroller ATMEGA16

Mikrokontroller AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-*bits word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*. Mikrokontroller tipe ini memiliki fitur tambahan yang tidak dimiliki oleh AT89S51/52 yaitu adanya fitur seperti ADC 10 bit, PWM, kemapuan timer dan kemampuan lain yang menjadikan mikrokontroller ini lebih canggih.

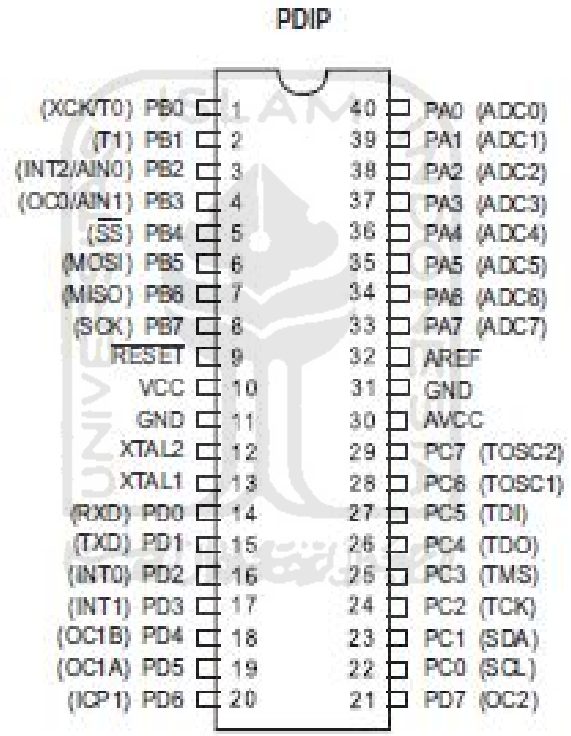
Fitur dasar yang dimiliki *Mikrokontroler* AVR ATMEGA16 adalah sebagai berikut :

1. Port I/O 32 jalur (Port A, Port B, Port C, Port D masing-masing 8 bit).
2. ADC 10 bit 8 channel.
3. 3 buah timer/counter.
4. 32 register dalam CPU
5. Watchdog Timer dengan osilator internal.
6. Flash PEROM 16 kbyte.
7. EEPROM 512 bytes.
8. SRAM 1 kbytes.

9. Interupsi Eksternal dan Internal.
10. Interfacing dengan komparator analog.
11. Port USART untuk komunikasi serial.

2.2.1 Konfigurasi PIN ATMEGA16

Mikrokontroler AVR ATMEGA16 terdiri dari 40 Pin, yang konfigurasinya sebagai berikut :



Gambar 2.3 Pin ATMEGA16
Sumber :Atmel Semiconductor, Datasheet ATMEGA16

Fungsi dari masing-masing pin Mikrokontroller AVR ATMEGA16 adalah sebagai berikut :

Tabel 2.2 Konfigurasi Pin ATMEGA16

No Pin	Nama	Fungsi
1	PB0 (XCK/TO)	Port B.0 / Timer-Counter 0 dan clock eksternal untuk USART (XCK)
2	PB1 (T1)	Port B.1 / Timer-Counter 1
3	PB2 (INT2/AIN0)	Port B.2 / Input (+) Analog komparator (AIN0) dan interupsi eksternal 2 (INT2)
4	PB3 (OC0/AIN1)	Port B.3 / Input (-) Analog komparator (AIN1) dan output pembanding timer/counter (OC0)
5	PB4 (SS)	Port B.4 / SPI Slave Select Input (SS)
6	PB5 (MOSI)	Port B.5 / SPI Bus Master Out Slave In
7	PB6 (MISO)	Port B.6 / SPI Bus Master In Slave Out
8	PB7 (SCK)	Port B.7 / sinyal clock serial SPI
9	RESET	Me-reset Mikrokontroller
10	VCC	Catu daya (+)
11	GND	Sinyal ground terhadap catu daya
12 – 13	XTAL 2 - XTAL 1	Sinyal input clock eksternal (kristal)
14	PD0 (RXD)	Port D.0 / penerima data serial
15	PD1 (TXD)	Port D.1 / pengirim data serial
16	PD2 (INT0)	Port D.2 / Interupsi eksternal 0
17	PD3 (INT1)	Port D.3 / Interupsi eksternal 1
18	PD4 (OC1B)	Port D.4 / Pembanding Timer-Counter 1
19	PD5 (OC1A)	Port D.5 / Pembanding Timer-Counter 1
20	PD6 (ICP1)	Port D.6 / Timer-Counter 1 Input
21	PD7 (OC2)	Port D.7 / Pembanding Timer-Counter 2
22	PC0 (SCL)	Port C.0 / Serial bus clock line
23	PC1 (SDA)	Port C.0 / Serial bus data input-output
24 – 27	PC2 – PC5	Port C.0
28	PC6 (TOSC1)	Port C.0 / Timer osilator 1
29	PC7 (TOSC2)	Port C.0 / Timer osilator 2
30	AVCC	Tegangan ADC
31	GND	Sinyal ground ADC
32	AREFF	Tegangan referensi ADC
33 – 40	PA0 (ADC0) – PA7 (ADC7)	Port A.0 – Port A.7 dan input untuk ADC (8 channel : ADC0 – ADC7)

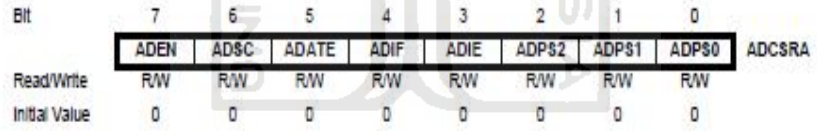
2.2.2 ADC Mikrokontroller ATMEGA16

Mikrokontroller TMEGA 16 mempunyai 8 input ADC. Secara umum, proses inisialisasi ADC meliputi proses penentuan clock, tegangan referensi, format output data, dan mode pembacaan. Register yang perlu diset nilainya adalah ADMUX

(ADC *Multiplxer Selection Register*), ADCSRA (ADC *Control and Status Register*), dan SFIOR (*Special Function IO Register*). ADMUX merupakan register 8 bit yang berfungsi menentukan tegangan referensi ADC, format data output, dan saluran ADC yang digunakan.

Register yang digunakan untuk setting ADC adalah :

1. ADC Control and Status Register (ADCSRA)



Gambar 2.4 ADC Control and Status Register (ADCSRA)

ADEN : 1 = ADC enable, 0 = ADC disabled

ADSC : 1 = Mulai konversi, 0 = konversi belum terjadi

ADATE : 1 = Auto trigger diaktifkan, trigger berasal dari sinyal yang dipilih (set pada register SFIOR bit ADTS). ADC akan start konfersi pada edge positif sinyal trigger.

ADIF : diset kesatu, jika konfersi ADC selesai dan data register terupdate

ADIE : diset 1, jika bit I dalam register SREG diset

ADPS (0..2) : bit pengatur clock ADC, factor pembagi 0-7 = 2,4,8,16,32,64,128

2. ADC Multiplexer (ADMUX)

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.5 Register ADMUX

REFS 0,1 : Pemilihan Teganganreferensi ADC

00 : Vref = Aref

01 : Vref = Avcc dengan eksternal kapasitor pada AREF

10 : vref = reversed

11 : vref = internal 2.56 voltdengan eksternal kapasitor pada AREF

ADLAR : *Untuk setting format data hasl koversi ADC,* Default = 0

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	

ADLAR = 1

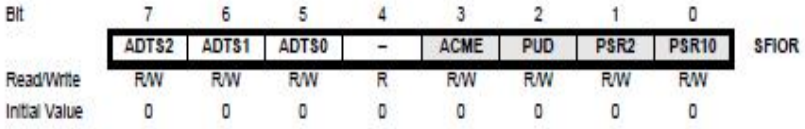
Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	

Gambar 2.6 Setting ADLAR

MUX0-4 : Pemilihan chanel ADC yang digunakan

3. Spesial Function IO Register (SFIOR)

SFIOR merupakan register 8 bit pengatur sumber picu ADC, yaitu dari internal atau eksternal



Gambar 2.7.Register SFIOR

ADTS(0-2) : Pemilihan trigger (pengatur picu) untuk konversi ADC, bit2 ini akan berfungsi jika bit ADATE pada register ADCSRA bernilai 1. Konfigurasi bit ADTS (0-2) dapat dilihat pada Tabel 2.3

Tabel 2.3 Pemilihan Sumber Picu ADC

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

ADHSM : 1, ADC high speed mode enable

BIT ACME, PUD, PSR2 dan PSR10 tidak diaktifkan

Untuk perhitungan konversi yaitu:

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}} \quad \text{.....(2.2)}$$

Dimana Vin ialah tegangan pada inputnya yang dipilih dab Vref tegangan referensi. Jika hasil ADC = 000h, maka menunjukkan tegangan input sebesar 0 V, jika hasil ADC 002h

2.3 *Positive Voltage Regulator* LM78XX

Regulator seri LM78XX adalah regulator dengan tiga terminal. Dapat diperoleh dengan berbagai tegangan tetap, sehingga memungkinkan untuk penggunaan yang fleksibel. Tegangan yang diperoleh dari regulator ini memungkinkan untuk dipakai dalam sistem-sistem logika, instrumrntasi, HiFi dan kelengkapan elektronik lainnya. Meskipun semula dirancang sebagai regulator tegangan tetap, namun akan dapat diperoleh tegangan dan arus yang dapat diatur dengan tambahan komponen eksternal.

Jenis IC regulator ini memiliki kelebihan-kelibihan teknis antara lain:

a. Arus keluaran lebih dari 1,5A

b. Toleransi tegangan keluaran adalah ± 5% (LM 78XXAC) dan ± 10% (LM 78LXXC).

c. Tidak memerlukan komponen eksternal

d. Jenis tegangan keluaran dari IC yang tersedia bervariasi (5V; 5,2V; 6V; 8V; 8,5V; 9V 12V; 15V;18V dan 24V)

e. Memiliki proteksi untuk panas berlebih

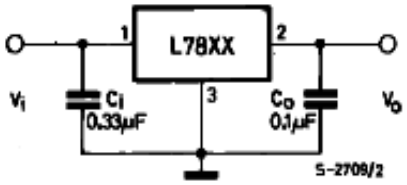
f. Pembatas arus hubung singkat

g. Pengaman daerah aman bagi transistor keluaran

Tabel 2.4 Karakteristik elektrik IC regulator keluarga 78XX

Type	V_{out} (V)	I_{out} (A)			V_{in} (V)	
		78XXC	78LXX	78MXX	Min.	Max.
7805	5	1	0,1	0,5	7,5	20
7806	6	1	0,1	0,5	8,6	21
7808	8	1	0,1	0,5	10,6	23
7809	9	1	0,1	0,5	11,7	24
7812	12	1	0,1	0,5	14,8	27
7815	15	1	0,1	0,5	18	30
7818	18	1	0,1	0,5	21	33
1824	24	1	0,1	0,5	27,3	38

Sedangkan gambar berikut menunjukkan rangkaian dasar dari regulator 78XX.

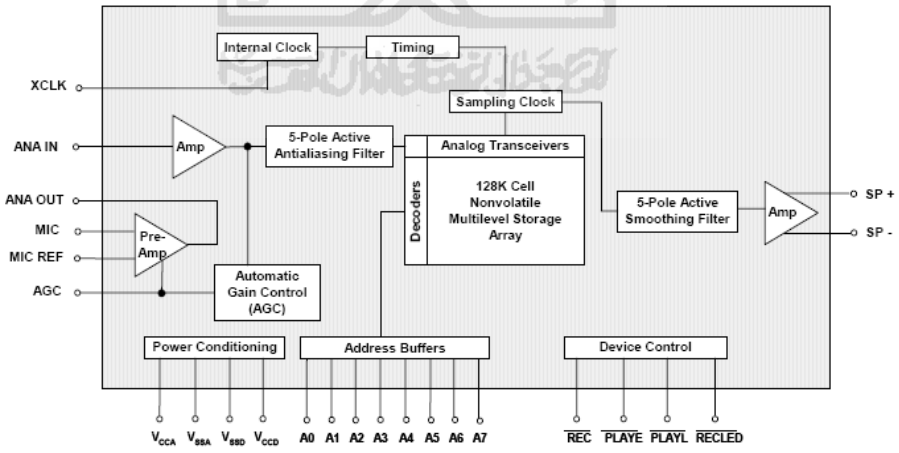


Gambar 2.8 Rangkaian Dasar Regulator 78XX

Selain Regulator untuk tegangan positif 78XX terdapat juga Regulator untuk Tegangan Negatif dengan seri komersial 79XX.

2.4 *Voice Record/Playback Storage Device* ISD 1420

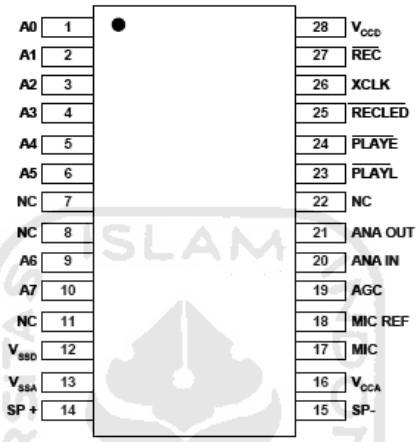
Information Storage Device (ISD) adalah rangkaian terpadu serpih tunggal yang mempunyai persamaan dengan CMOS LSI. ISD seri 1420 *Chip Corder* merupakan peralatan yang dirancang untuk merekam dan memutar ulang suara dalam satu *chip*



Gambar 2.9 Diagram Blok ISD1420

IC ISD 25120 mempunyai beberapa perlengkapan didalamnya, antara lain osilator internal, mikropon *pre-amplifier*, *automatic gain control*, tapis perata dan *speaker amplifier* (penguat speaker).

2.4.1 Struktur PIN



Gambar 2.10 Diagram pin ISD1420

Seri ISD1420 menyediakan kualitas yang tinggi, *single-chip*, dalam aplikasi merekam dan putar ulang pesan untuk waktu 20 detik.

Konfigurasi pin sebagai berikut:

a. Ax (*Address/Mode Inputs*). Alamat/mode masukan mempunyai dua fungsi tergantung pada logika dari dua *Most Significant Bits* (MBS) yang berdapat pada pin alamat (A6 dan A7). Jika salah satu atau kedua MBS berlogika rendah, seluruh pin masukan diterjemahkan sebagai bit alamat dan sigunakan sebagai awal alamat pada saat siklus rekam atau putar ulang. Pin alamat hanya

sebagai pin masukan dan bukan merupakan keluaran dari alamat internal selama operasi. Alamat masukan dikunci oleh tebing turun dari logika PlayE.

b. AUX IN (*Auxillary Input*). Masukan *auxillary* dikuatkan melalui pin keluaran *amplifier* dan pin keluaran speaker ketika CE berkondisi tinggi, P/R tinggi dan putar ulang pada saat ini tidak aktif atau jika komponen dalam kondisi putar ulang *overflow*.

c. V_{SSA}, V_{SSD} (*Ground*). Komponen ISD1420 dilengkapi dengan *ground* analog dan *ground* digital. Pin-pin tersebut harus dihubungkan terpisah melalui sebuah bagian impedansi rendah ke *ground* catu daya.

d. SP+/SP- (*Speaker Output*). Semua komponen dalam ISD1420 terdapat sebuah *chip driver speaker,* yang mampu men-*driver* 50 mW dalan 16Ω dari AUX IN (12,2 mW dari memori).

e. VCCA,VCCD (Sumber Tegangan). Untuk mengurangi *noise*, rangkaian analog dan digital pada komponen ISD1420 digunakan sumber tegangan yang terpisah. Jalur sumber tegangan yang keluar ke pin dibedakan. Jika hanya menggunakan sebuah sumber tegangan, maka harus di-*couple* dengan kapasitor.

f. MIC (*Microphone*). Pin *microphone* memindahkan sinyal masukan ke dalam *chip preamplifier*. Rangkaian *Automatic Gain Control* (AGC) di dalam *chip* mengontrol penguatan *preamplifier* dari -15 hingga 24dB. *Microphone* luar harus dikopeling dengan kapasitor ke dalam pin *mikrophone* ini.

g. MIC REF (*Microphone Reference*). Masukan MIC REF adalah masukan *inverting* ke penguat *microphone.*

h. AGC (*Automatic Gain Control*). Kegunaan dari AGC adalah untuk menambah atau mengurangi secara otomatis penguatan (*gain*) dari *preamplifier*, dan juga meluaskan batas dari sinyal masukan yang dapat digunakan oleh mikropon tanpa terjadi distorsi. AGC ini dapat secara dinamis meluaskan batas dari suara yang terekam dari suara bisikan sampai suara yang keras. Untuk menggunakan fasilitas AGC ini, resistor dan kapasitor luar (eksternal) harus dihubungkan secara paralel antara pin AGC dengan ground.

i. ANA IN (*Analog Input*). Kapasitor eksternal (luar) yang menghubungkan antara ANA IN dan ANA OUT. Nilai dari kapasitor luar, dengan impedansi masukan 3 KΩ dari ANA IN, dapat dipilih untuk memberikan keadaan *cutoff* pada frekuensi rendah.

j. RECLED. Sinyal ini berlogika rendah pada akhir dari memori IC, mengindikasikan bahwa komponen telah penuh dan pesan telah *overflow*. Pada saat *play* memberikan sinyal EOM (*End Of Message*). Sebuah tanda akan dimasukkan secara otomatis pada akhir setiap pesan yang direkam. Tanda ini akan ada sampai akhir pesan yang direkam. EOM menguluarkan pulsa rendah untuk sebuah periode dari akhir setiap pesan.

k. PLAY L. Pin masukan PLAY L dikondisikan rendah untuk operasi putar ulang.

l. REC. operasi rekam diberi logika low pada pin REC

m. PLAY E. operasi putar ulang dengan masukan transisi dari high ke Low pada kaki pin PLAY E

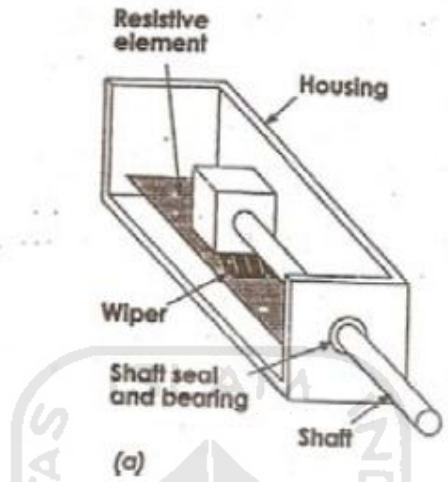
n. XCLK (External Clock). Pin masukan clock eksternal mempunyai sebuah perlengkapan pull-down internal. Perlengkapan ini dikonfigurasi pada pabrik dengan suatu pengambilan contoh clock internal frekuensi tengah hingga ± 1% dari spesifikasi.

2.4.2 Sistem Pengalamatan

ISD1420 ini dapat merekam suara dengan durasi waktu selama 20 detik. Memiliki pin alamat sebanyak 8 buah (A_0,A_1,A_2,A_3,A_4,A_5,A_6 dan A_7). Ketika semua alamat terpakai, maka dapat memiliki sebanyak 2^8=256 alamat. Tetapi dalam prakteknya, IC seri ini memiliki 160 tempat alamat. Masing-masing alamat memiiki durasi waktu selama 125 ms.

2.5 Potensio Geser

Potensio geser merupakan salah satu jenis resistor variable yaitu resistor yang dapat diubah nilai tahanannya. Salah satu jenis potensiometer adalah potensiometer geser, yaitu potensiometer yang nilai tahanannya dapat diubah dengan cara menggeser knop geser yang ada pada potensiometer tersebut. Gambar 2.11 memperlihatkan sebuah potensiometer geser, yang memiliki sebuah pita film, disebut sebagai jalur (track), yang terbuat dari karbon. Jalur ini terbuat dari bahan keramik yang bersifat konduktif. Ujung-ujung jalur terhubung ke dua buah terminal potensiometer.

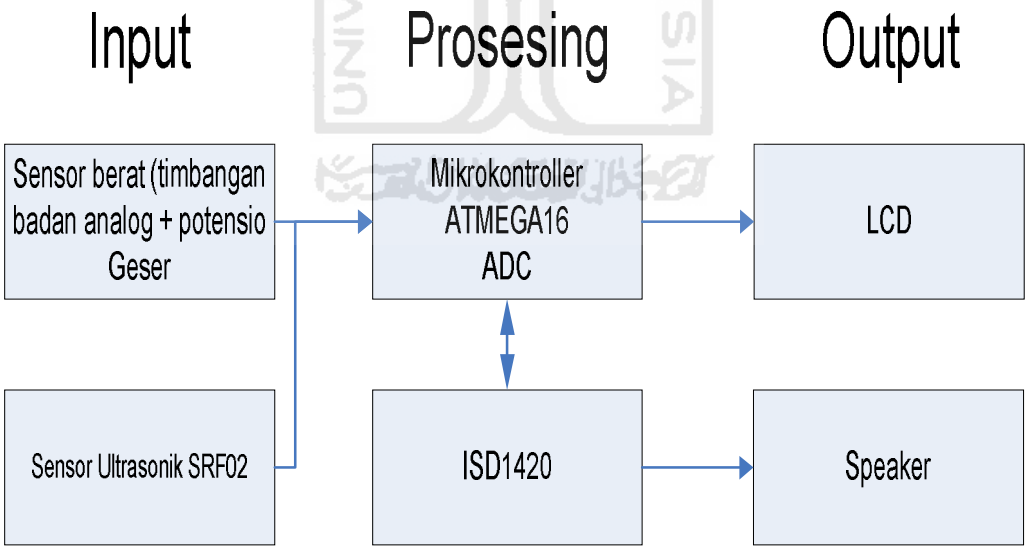


Gambar 2.11 Potensiometer Geser

BAB III
PERANCANGAN

4.1 Gambaran Umum Sistem

Perangkat keras yang akan dibangun adalah Pengukur Berat Badan dan Tinggi Badan Dengan Output Suara . Input untuk system yang akan dibuat ini berupa sensor Ultrasonik dan sensor berat modifikasi timbangan badan analog ditambah potensio geser. Untuk pengolahnnya digunakan mikrokontroler ATMEGA16 dan ISD1420 sedangkan untuk outputnya berupa data suara dan tampilan berat badan dan tinggi badan di LCD. Diagram blok Pengukur berat badan dan tinggi badan dengan output suara dapat dilihat pada Gambar 3.1.

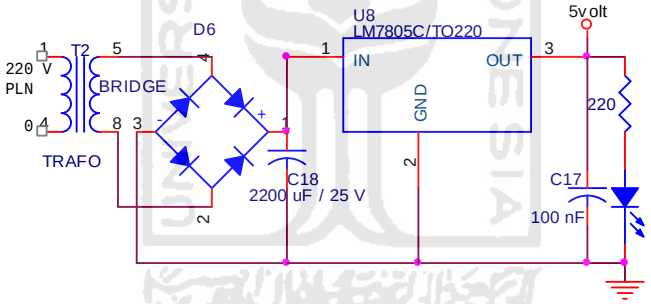


Gambar 3.1 Blok Diagram Pengukur Berat dan Tinggi Badan dengan Output Suara

3.2 Perancangan Perangkat Keras

3.2.1 *Power Supply*

Power supply yang digunakan dalam pembuatan Pengukur berat badan dan tinggi badan dengan output suara , berupa catu daya 5 Volt. Trafo yang digunakan adalah trafo 1 A. Fungsi trafo dalam rangkaian ini untuk menurunkan tegangan dari 220 V AC ke 12 Volt AC. Untuk merubah tegangan AC ke DC menggunakan dioda bridge 1 A, dan untuk menurunkan tegangan 12 V DC ke 5 Volt DC menggunakan regulator LM7805, mikrokontroller membutuhkan tegangan/catu daya 5 Volt ± 10%, Rangkaian power supply Pengukur Berat Badan dan Tinggi Badan Dengan Output Suara ditunjukkan pada Gambar 3.2

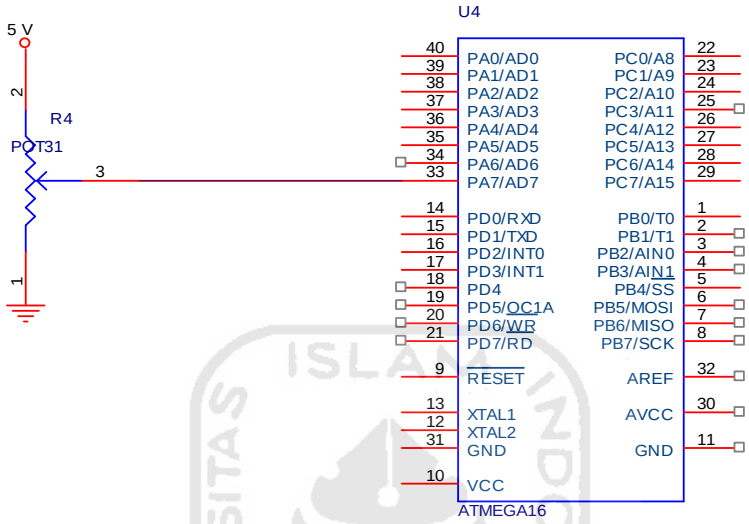


Gambar 3.2 Rangkaian *Power Supply*

3.2.2 Rangkaian Sensor Berat

Rangkaian sensor berat badan menggunakan timbangan berat badan analog ditambah dengan potensio geser / variabel resistor. Keluaran dari sensor berat badan berupa variabel tegangan dari 0V-5V.

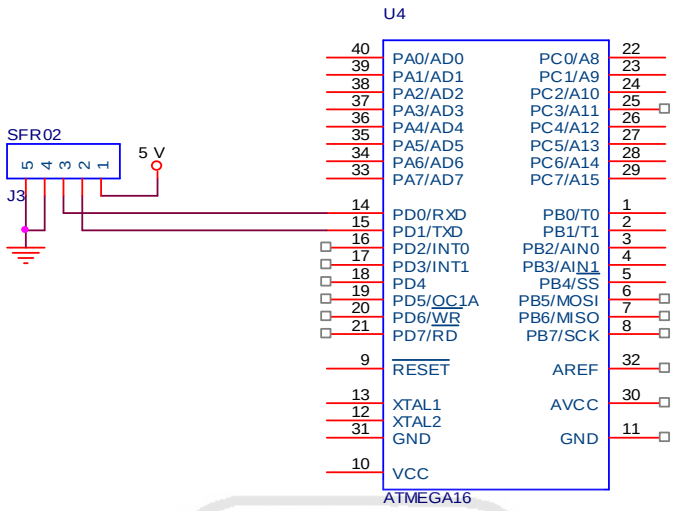
Keluaran dari sensor berat dibaca oleh ADC mikrokontroller ATMEGA16 channel 7. Rangkaian sensor berat badan dapat dilihat pada Gambar 3.3



Gambar 3.3 Rangkaian Sensor Berat Badan

3.2.3 Rangkaian SRF02 sensor jarak (ultrasonik)

Rangkaian sensor jarak menggunakan modul ultrasonik dengan merk SRF02. Keluaran dari sensor SRF02 berupa jarak terbaca dalam cm yang di kirim menggunakan interface serial. Rangkaian sensor berat badan dapat dilihat pada Gambar 3.4.



Gambar 3.4 Rangkaian Sensor Jarak SRF02

3.2.4 Mikrokontroler ATMEGA16

3.2.4.1 Osilator

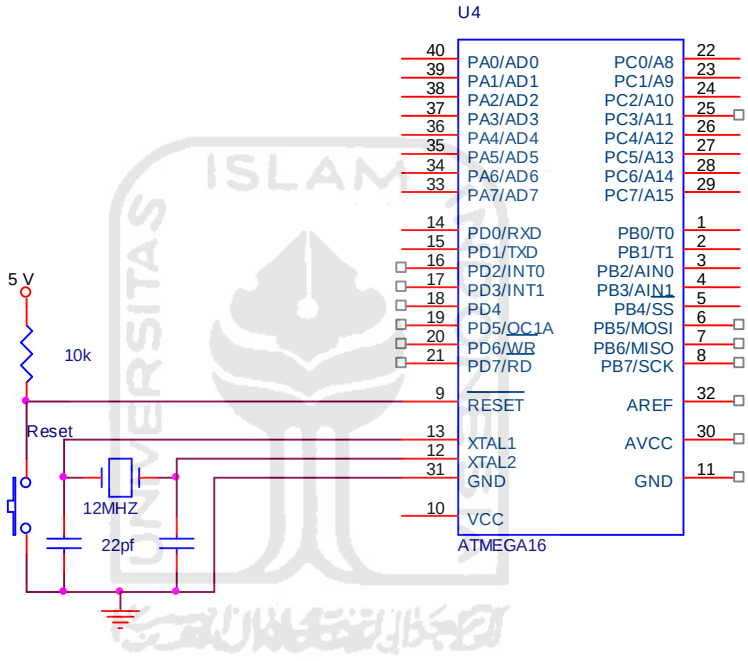
Pada rangkaian osilator ini digunakan kristal 12 MHz seperti ditunjukkan Gambar 16. Menurut *datasheet* crystal yang bisa digunakan untuk mikrokontroller ATMEGA16 adalah 0 – 16 MHz dan dua kapasitor 22 pF. Waktu yang dibutuhkan untuk melakukan eksekusi sebuah instruksi dinamakan waktu siklus instruksi (instruction cycles time) yang nilainya tergantung pada kristal yang digunakan.

3.2.4.2 Reset

Rangkaian *reset* digunakan untuk menghentikan kerja mikrokontroler dengan kembali ke alamat 0000/reset. Rangkaian *reset* dapat dilihat pada Gambar 16. Untuk mereset mikrokontroler ATMEGA16 yaitu dengan memberikan logika Low pada pin reset (pin 9) mikrokontroller ATMEGA16, logika low ini dibuat

minimal 50 ns. Keadaan yang dapat membuat mikrokontroller masuk kedalam kondisi reset adalah:

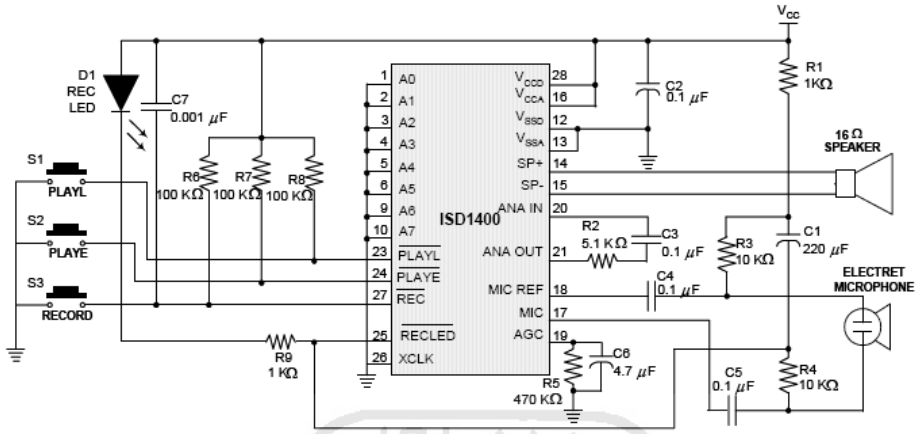
1. Pada saat Power On
2. Saat reset eksternal terjadi, yaitu ketika pin reset diaktifkan
3. Pada saat watchdog timer mencapai nilai maksimum (overflow)



Gambar 3.5 Rangkaian Minimum ATMEGA16

3.2.5 Rangkaian ISD1420

Piranti yang menghasilkan suara adalah rangkaian ISD1420. Rangkaian putar ulang dan rekam ditunjukkan pada Gambar 3.6.



Gambar 3.6 Rangkaian Rekam ISD1420

Terdapat 17 pesan yang direkam dalam IC ISD1420 sesuai dengan jumlah kata-kata untuk meyebutkan berat badan dan tinggi badan. Untuk dapat merekam 17 pesan, maka setiap pesan memiliki waktu:

$$\frac{20\,\text{detik}}{17} = 1{,}17 \text{ detik}$$

Karena ISD1420 memiliki resolusi 0,125detik, maka waktu yang dipakai adalah 1,1 detik untuk setiap pesan. Resolusi didapat dari durasi ISD1420 dibagi dengan jumlah alamat yang dapat dialamati. ISD1420 dapat dialamati dengan 160 alamat.

$$\frac{20\,\text{detik}}{160} = 0{,}125\,\text{detik}$$

Waktu yang diperlukan untuk merekam 17 pesan adalah:

$$1{,}1 \times 17 = 18{,}7\,\text{detik}$$

Sehingga terdapat sisa waktu sebanyak:

$$20 - 18{,}7 = 1{,}3 \,\text{detik}$$

Rangkaian pada Gambar 3.6 digunakan untuk saat pengisian / perekaman suara secara manual. Untuk melalukan perekaman sebuah pesan. Tahap-tahap yang harus dilakukan untuk proses rekam (*record*) adalah :

1. Tentukan alamat.
2. Tekan tombol REC, lepaskan jika telah selesai merekam.

Waktu penekanan tombol REC adalah selama 1,1 detik, yaitu lebih kecil dari durasi setiap alamat. Hal ini dilakukan agar suara yang terekam pada ISD1420 tidak terjadi penumpukan. Untuk membuat pin REC berada pada posisi LOW selama 1 detik, diwakilkan pada mikrokontroler yang telah diprogram khusus sebelumnya.

Sedangkan proses putar ulang (*play*) dapat dilakukan menurut langkah berikut:

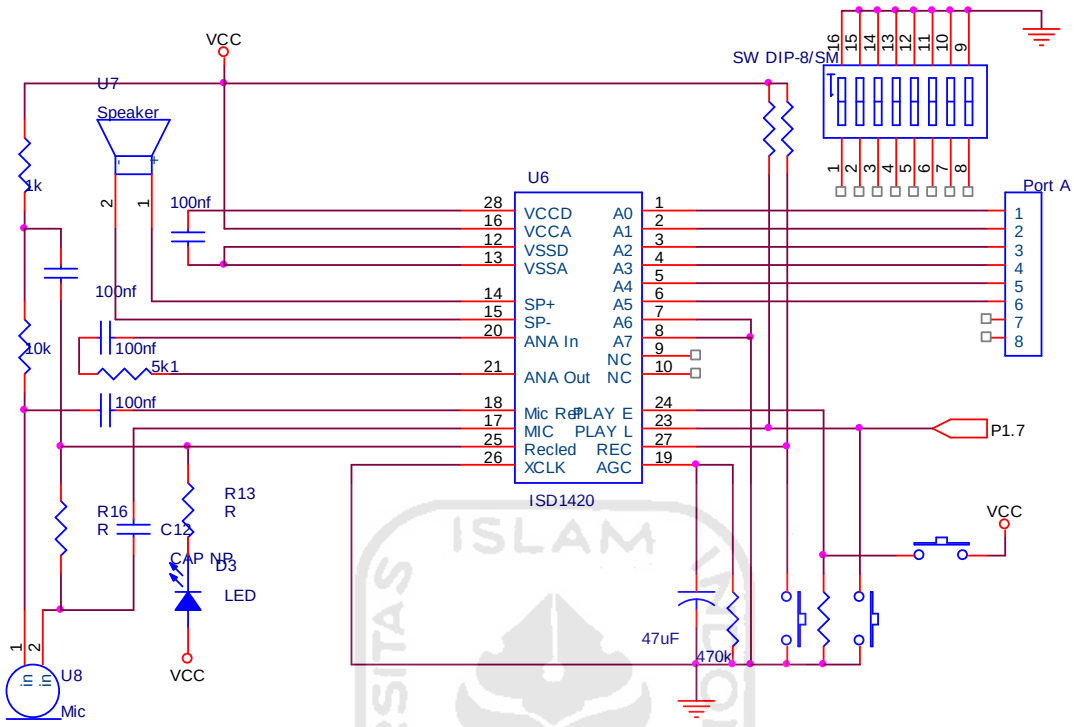
1. Tentukan alamat,
2. Berikan *trigger* LOW pada PLAY E.

Dengan langkah putar ulang (*play*) yang dilakukan diatas, maka suara yang ada alamat dipilih akan dikeluarkan sampai selesai. Pada tabel berikut akan dijelaskan untuk masing-masing suara yang telah direkam.

Tabel 3.1 Alamat dan waktu perekaman ke ISD1420

NO	Suara	Alamat(biner A7-A0)
1	**Satu**	0000 0000
2	**Dua**	0000 0101
3	**Tiga**	0000 1010
4	**Empat**	0000 1111
5	**Lima**	0001 0100
6	**Enam**	0001 0111
7	**Tujuh**	0001 1010
8	**Delapan**	0001 1110
9	**Sembilan**	0010 0010
10	**Se**	0010 0101
11	**Puluh**	0010 0111
12	**Ratus**	0010 1010
13	**Belas**	0010 1101
14	**Tinggi**	0011 0001
15	**Berat**	0011 0100
16	**Cm**	0011 0111
17	**Kg**	0011 1011

Mikrokontroler mengendalikan ISD hanya pada waktu pemutaran ulang. Rangkaian pemutaran ulang dengan kendali mikrokontroler ditunjukkan pada Gambar 3.7.

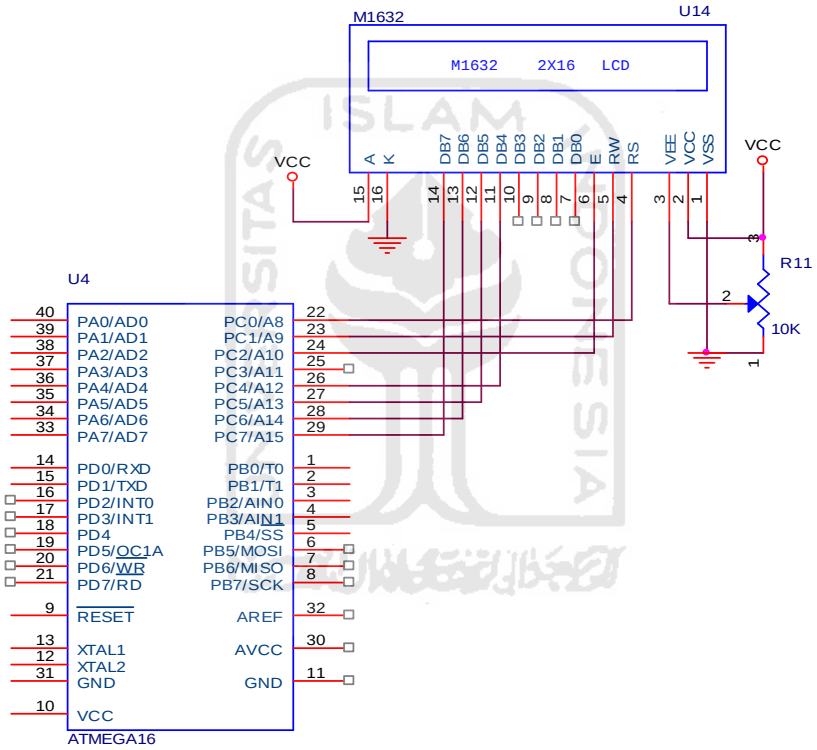


Gambar 3.7 Rangkaian Putar Ulang dan Hubungan ISD 1420 dengan Mikrokontroller

Hal yang harus dilakukan pada saat pemutaran ulang yaitu dengan menentukan alamat pesan kemudian memberi *trigger* LOW pada kaki PLAY E ISD1420. Ketika pada akhir setiap suara/pesan yang terekam, maka keluaran pada RECLED akan menghasilkan keluaran *trigger* LOW.

3.2.6 Rangkaian LCD

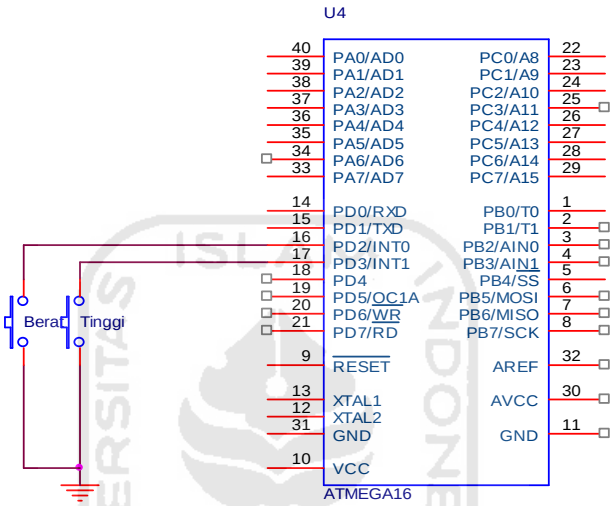
Untuk menampilkan data berat badan dan tinggi badan sistem menggunakan tampilan LCD 16 x 2. LCD yang digunakan adalah M1632 buatan *Hitachi*. Untuk koneksi LCD ke mikrokontroler dibutuhkan data bus dan kontrol RS, RW, dan E. Untuk data bus digunakan PC4-PC7, sedangkan untuk RS = PC.0, RW= PC.1, dan E= PC.2. Rangkaian LCD dapat dilihat pada Gambar 3.8.



Gambar 3.8 Rangkaian LCD

3.2.7 Rangkaian Tombol

Rangkaian tombol dapat dilihat pada Gambar 20. Ketika tombol ditekan akan terhubung dengan ground. Input mikrokontroller saat tombol ditekan berlogika 0 dan saat tombol tidak ditekan berlogika 1 high berasal dari pull up internal port mikrokontroller ATMEGA16 diaktifkan.



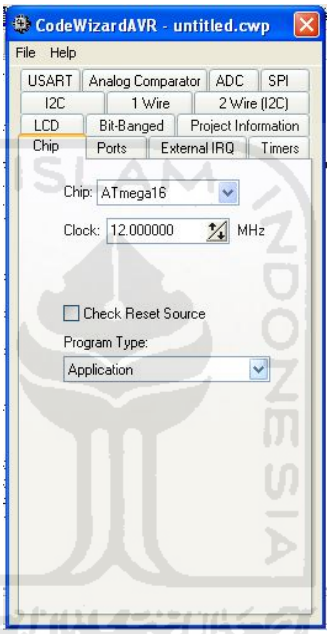
Gambar 3.9 Rangkaian Tombol

3.3 Perancangan Perangkat Lunak

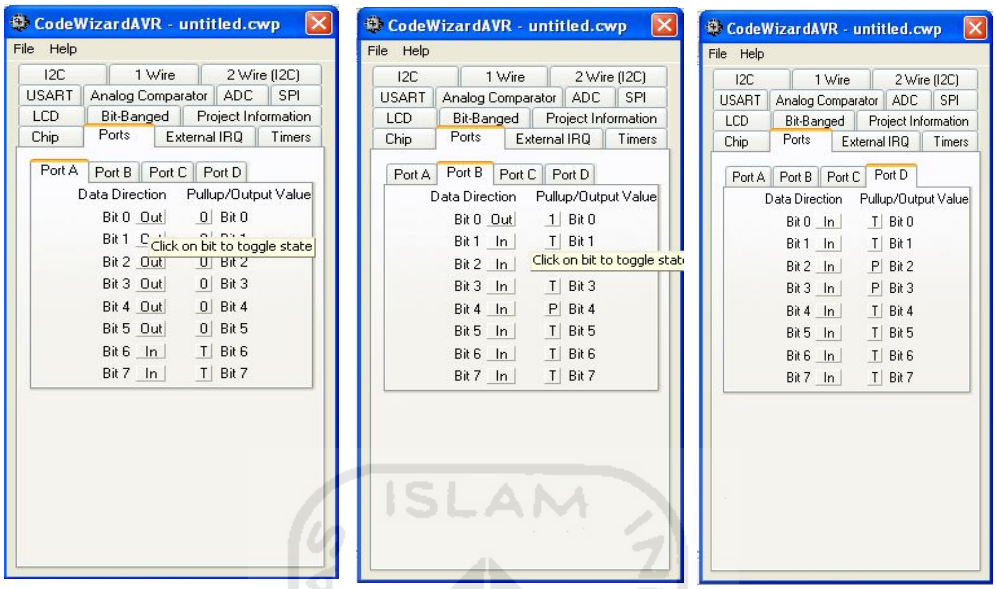
Agar sistem bekerja dengan sebagaimana mestinya, maka diperlukan perangkat lunak yang mengatur kerja dari keseluruhan rangkaian. Pertama-tama yang dibuat adalah diagram alir (*flowchart*) dan kemudian dilakukan pembuatan program. Pembuatan program ditulis dengan bahasa C menggunakan tool *codevision* AVR, dan program tersebut disimpan dalam memori flash mikrokontroller ATMEGA16. Pada mikrokontroller ATMEGA16 terdapat memori program sebesar 16 kbyte flash, EEPROM 512 byte dan memori data 512 byte RAM.

3.3.1 Setting CodeVision

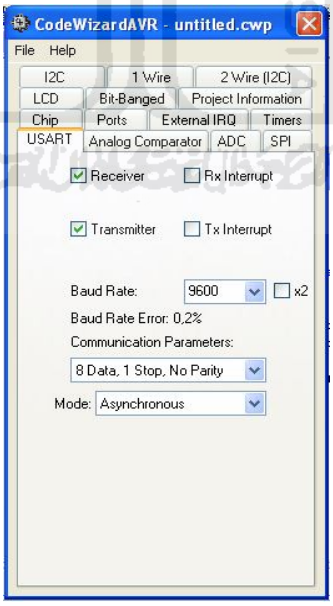
Setting awal code vision dengan menentukan jenis IC Mikrokontroller yang digunakan. Mikrokontroller yang digunakan adalah ATMEGA16, crystal yang digunakan 12 MHz. Pengaturan Code Vision untuk pemilihan jenis mikrokontroller dapat dilihat pada Gambar 3.10



Gambar 3.10 Setting Codevision untuk pemilihan jenis mikrokontroller

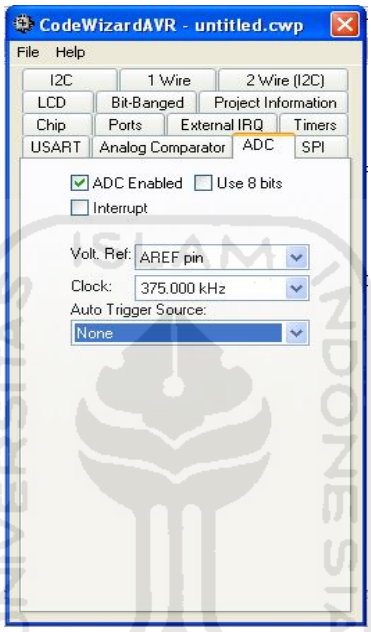


Gambar 3.11 Setting Codevision untuk fungsi masing masing PORT I/O mikrokontroler



Gambar 3.12 Pengaturan Komunikasi Serial Mikokontroller pada Code Vision

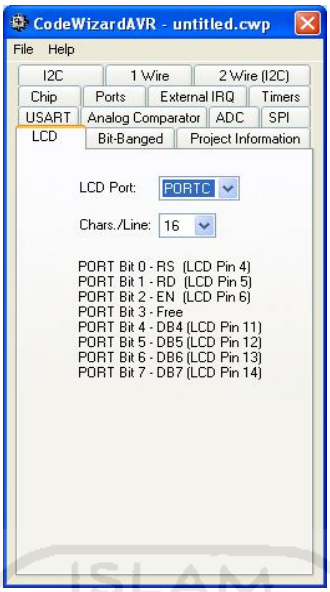
Komunikasi serial yang digunakan menggunakan kecepatan transfer 9600 bps, tanpa menggunakan interupsi ketika ada data serial masuk atau pun ketika berhasil mengirim data. Pengaturan Code Vision untuk pengaturan komunikasi serial dapat dilihat pada Gambar 3.12



Gambar 3.13 Setting ADC pada CodeVision

Untuk pengaturan ADC dengan mencentang ADC enable. Clock dapat dipilih dari beberapa pilihan, pada pmebuatan system ini dipilih yang paling tinggi yaitu 375.000kHz Pengaturan Code Vision untuk pengaturan ADC dapat dilihat pada Gambar 3.13.

Pengaturan Code Vision untuk penggunaan LCD dapat dilihat pada Gambar 3.14. LCD dihubungkan ke mikrokontroller melalui PORT C.



Gambar 3.14 Setting Codevision untuk Penggunaan LCD

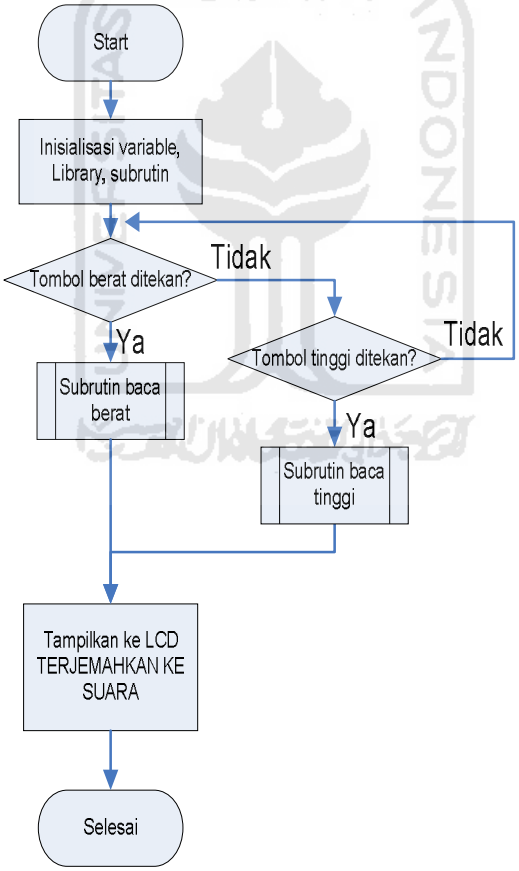
3.3.2 Inisialisasi

Inisialisasi digunakan untuk mendeklarasikan variabel, function, dan library yang akan digunakan. Deklarasi variabel dan library yang digunakan dapat dilihat pada listing program berikut:

```
#include <mega16.h>
#include <lcd.h>
#include <stdio.h>
#include <delay.h>
unsigned int baca_berat();
void voice (unsigned char i);
unsigned int baca_tinggi();
void show (int);
unsigned int temp, vc0, vc1, vc2,tinggi,berat;
unsigned char voice_code[]
={0x00,0x00,0x05,0x0a,0x0f,0x14,0x17,0x1a,0x1e,0x22,0x25,0x27,0x2a,0x2d,0x31,0x34,0x37,0x3b};
```

3.3.3 Program Utama

Program utama berisi program yang mendeteksi tombol perintah pengambilan data berat badan dan tinggi badan. Perintah pembacaan tinggi badan ketika tombol terhubung, PORTD 4 terhubung dengan *ground*. Sedangkan perintah pembacaan berat badan ketika tombol terhubung PORTD 4 juga terhubung ke *ground*. Setelah diperoleh data tinggi dan data berat, maka akan ditampilkan ke LCD dan diterjemahkan ke suara. Flowchart program utama dapat dilihat pada Gambar 3.15.



Gambar 3.15 Flowchart Program Utama

3.3.4 Subrutin Baca Tinggi

Subrutin Baca tinggi berisi program mengambil data tinggi dari sensor ultrasonik SRF02 Menggunakan *interface* serial. *Command* yang digunakan untuk perintah pembacaan adalah decimal 00 dan 84. Kemudian ditunggu data jarak yang diperoleh. Data jarak yang diperoleh berisi 2 byte. Flowchart Subrutin baca tinggi dapat dilihat pada Gambar 3.16.



Gambar 3.16 Flowchart Subrutin Baca Tinggi

Listing subrutin program baca tinggi sebagai berikut.

```
unsigned int baca_tinggi (void)
{
    unsigned int jarakup,jaraklow,jarak,tempjarak;
    putchar(00);
    putchar(84);
    jarakup=getchar();
    jaraklow=getchar();
    if (jarakup==1)
      {tempjarak=jaraklow + 256;};
    if (jarakup==0)
      {tempjarak=jaraklow;};
    jarak = 200 - tempjarak;
    return jarak;
}
```

Sensor jarak dipasang 200 cm. Tinggi badan diperoleh dari jarak pemasangan sensor badan dikurangi jarak yang terbaca pada dari sensor ultrasonik.

3.3.5 Subrutin Baca Berat

Subrutin Baca berat berisi program mengambil data berat badan dari sensor berat menggunakan ADC. ADC pada mikrokontroller ATMEGA16 mempunyai resolusi 10 bit, atau ((2 pangkat 10)-1) = 1024. Dengan digital *output proporsional to analog input* (Qout) sebesar

Qout = Vin / Vref

Qout= Vin / Vcc,

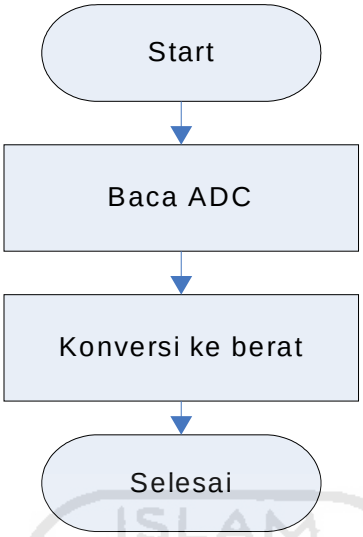
Untuk Vref ADC yang digunakan 5 Volt. Hal ini dengan Vref = Vcc = 5 Volt, jadi misalkan keluaran dari ADC adalah 01H. maka

Vin= (nilai keluaran ADC/1024)x Vref,

Vin = (1/1024) x 5 Volt

Vin= 0,004882 Volt atau 4,882 mVolt.

Dari hasil pembacaan ADC kemudian diterjemahkan ke Berat badan. Flowchart Subrutin baca berat dapat dilihat pada Gambar 3.17.



Gambar 3.17 Flowchart Subrutin baca berat

Berikut ini listing subrutin baca berat;

```
unsigned int baca_berat (void)
        {
        int temp1,temp2,count;
        temp1=(510 - read_adc(7));
        temp2=((temp1/11)*5);
        count=temp2;
        return count;
```

BAB IV
PENGUJIAN DAN ANALISIS KINERJA SISTEM

4.2 Metode Pengujian

Berdasarkan spesifikasi sistem yang telah dijelaskan sebelumnya, selanjutnya dilakukan pengujian terhadap sistem menggunakan beberapa metode pengujian. Tujuan pengujian ini untuk membuktikan apakah sistem yang diimplementasikan telah memenuhi spesifikasi yang telah direncanakan sebelumnya. Hasil pengujian akan dimanfaatkan untuk menyempurnakan kinerja sistem dan sekaligus digunakan dalam pengembangan lebih lanjut.

Metode pengujian dipilih berdasarkan fungsi operasional dan beberapa parameter yang ingin diketahui dari sistem tersebut. Data yang diperoleh dari metode pengujian yang dipilih tersebut dapat memberikan informasi yang cukup untuk keperluan penyempurnaan sistem.

Dalam penelitian ini dipilih dua macam metode pengujian, yaitu pengujian fungsional dan pengujian kinerja sistem. Pengujian fungsional digunakan untuk membuktikan apakah sistem yang diimplementasikan dapat memenuhi persyaratan fungsi operasional seperti yang direncanakan.

Pengujian kinerja sistem dimaksudkan untuk memperoleh beberapa parameter yang dapat menunjukkan kemampuan dan kehandalan sistem dalam menjalankan fungsi operasionalnya.

4.3 Pengujian Fungsional

Sebagaimana dijelaskan sebelumnya, pengujian fungsional bertujuan untuk memeriksa fungsi operasional sistem yang di implementasikan apakah telah

sesuai dengan spesifikasi yang direncanakan dan sistem menjalankan fungsinya sesuai dengan tujuan pengembangannya.

Ada dua macam metode pengujian fungsional yang dilakukan. Metode pertama adalah pengujian fungsional bagian demi bagian, sedangkan yang kedua adalah pengujian sistem secara keseluruhan.

4.2.1 Pengujian Rangkaian Sensor Berat Badan

Sensor yang digunakan berupa modifikasi timbangan analog dengan penambahan potensio geser. Jumlah sensor yang digunakan berjumlah 1 buah. Pengujian rangkaian sensor dengan mendeteksi keluaran tegangan yang terbaca oleh ADC Mikrokontroller dengan perubahan berat, membaca berat badan analog pada timbangan..

ADC Mikrokontroller melakukan konversi data analog menjadi data 10 bit data digital. Tegangan Vref = 5 Volt, seandainya data adc terbaca 01h

Vin= (nilai keluaran ADC/1024)x Vref,

Vin = (1/1024) x 5 Volt

Vin= 0,004882 Volt atau 4,882 mVolt.

Dari hasil pembacaan ADC kemudian diterjemahkan ke berat badan. Hasil pengujian ditampilkan pada Tabel 4.1.

Tabel 4.1 Hasil pengukuran sensor berat badan

No	Timbangan analog	Keluaran ADC
1	0	515
2	15	480
3	50	396
4	65	353
5	83	316

Dari hasil pengujian keluaran sensor berat badan dapat disimpulkan bahwa Kenaikan tegangan keluaran sensor berat badan berbanding lurus terhadap berat badan.

4.2.2 Rangkaian Ultrasonic SRF02

Pengujian rangkaian ultrasonik SRF02 dilakukan dengan mengukur kemampuan pembacaan jarak. Hasil dari pengujian dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Uji Rangkaian Ultrasonik

No	Pengukuran Dengan Penggaris	Pengukuran Dengan SRF02	Error (%)
1	35	35	0
2	50	47	6
3	60	61	1,6
4	80	80	0
5	95	92	3,15
6	110	108	1,8
7	125	121	3,2
Rata-rata kesalahan			2,25

Hasil pengujian pada Tabel 4.2 dapat dilihat hasil pengukuran dengan penggaris dan sensor ultrasonik SRF02 menunjukkan hasil memiliki sedikit perbedaan.

4.2.3 Rangkaian Tombol

Pengujian rangkaian tombol dilakukan dengan mengukur tegangan keluaran tombol saat tidak ditekan saat tombol ditekan. Hasil Pengujian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Uji Rangkaian Tombol

No	Tombol	Kondisi tombol (Volt)	
		Ditekan	Tidak ditekan
1	Berat	0	4,92
2	Tinggi	0	4,92

4.2.4 Rangkaian Reset Mikrokontroller

Rangkaian reset berfungsi menghentikan kerja CPU dan kemudian mengulang dari awal (program *counter* ke alamat 0000). Saat catu daya dihidupkan rangkaian reset menunda kerja dari CPU hingga tegangan stabil (*power on reset*). Reset pada mikrokontroler ATMEGA16 adalah aktif tinggi.

Pengujian dilakukan dengan menghidupkan mikrokontroler mengamati kerja mikrokontroler. Dari hasil pengamatan dapat disimpulkan reset mikrokontroler dapat berfungsi dengan baik, *power on reset* dapat berfungsi dengan baik, dengan tanda mikrokontroler dapat langsung bekerja ketika power dihidupkan. Begitu juga ketika dilakukan reset pada mikrokontroler melalui tombol maka mikrokontroler dapat melakukan reset.

4.2.5 Pengujian Rangkaian Power Supply

Pengujian Rangkaian power supply dilakukan dengan mengukur tegangan keluaran dari rangkaian power supply. Hasil pengukuran dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Pengukuran Tegangan Power Supply

No	Sumber Power Supply	Tegangan Terbaca (V)
1	5 Volt	4,94
2	5Volt	4,91
3	5 Volt	4,92
4	5Volt	4,91

Dari hasil pengukuran dapat dilihat keluaran tegangan memenuhi syarat yaitu tidak lebih atau kurang dari 5 % tegangan yang dikehendaki.

4.3. Pengujian dan Pembahasan Sistem Keseluruhan

Untuk memulai menjalankan alat ini yaitu dengan menghubungkan kabel power ke tegangan 220 volt PLN. Tampilan LCD pertama saat power dihidupkan dapat dilihat pada Gambar. 4.1.



Gambar 4.1 Tampilan Awal Pada LCD

Pengujian selanjutnya dengan mengukur berat badan dan tinggi badan yang hasilnya dapat di lihat pada LCD dan dapat didengarkan.

4.3.1 Pengukuran Berat Badan

Untuk mengukur berat badan, menggunakan timbangan yang telah dimodifikasi dengan menambahkan potensio geser. Sebelum Mengukur berat badan, terlebih dahulu harus menyamakan angka yang terdapat pada timbangan dengan tampilan pada LCD. Hasil dari pengukuran berat badan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil Pengukuran Berat Badan

No	Nama	Berat Badan		Suara	Error (%)
		Timbangan	Alat		
1	Marwan	35	35	ok	0
2	Buk Karti	48	50	ok	4,16
3	Uul	48	50	ok	4,16
4	Joko	65	65	ok	0
5	Rosyid	50	50	ok	0
6	Anggar	62	65	ok	4.8
7	Aris	47	50	ok	6,3
Rata- kesalahan					2,77

4.3.2 Pengukuran Tinggi Badan

Untuk mengukur tinggi badan, posisi sensor ultrasonik diletakkan pada jarak 2 meter dari permukaan. Ketepatan posisi sensor ultrasonik menentukan hasil dari objek yang diukur. Hasil dari pengukuran tinggi badan dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil Pengukuran Tinggi Badan

No	Nama	Tinggi Badan		Suara	Error (%)
		Manual	Alat		
1	Marwan	146	148	ok	1,4
2	Buk Karti	157	155	ok	1,3
3	Uul	159	163	ok	2,5
4	Joko	165	165	ok	0
5	Rosyid	165	163	ok	1,2
6	Anggar	170	174	ok	2,3
7	Aris	160	157	ok	1,9
Rata- kesalahan					1,5

Dari hasil pengujian dan pengamatan pada proses diatas maka dapat di simpulkan sistem dapat bekerja sesuai dengan yang diharapkan. Kelemahan dari alat ini adalah pada saat mengukur berat badan hanya bisa menampilkan berat dalam jumlah kelipatan 5 (lima). Apabila hasil pengukuran bukan kelipatan 5 (lima) maka akan dibulatkan menjadi kelipatan 5 (lima).

Berdasarkan hasil pengujian diatas maka dapat dilihat perbandingan pengukuran nilai tekanan darah menggunakan tensimeter buatan dan tensimeter analog. Dari pengukuran tersebut terdapat beberapa perbedaan nilai yang diperoleh pada setiap pengukuran.

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan perancangan, pembuatan, pengujian alat, dan pembahasan, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Hasil dari pengukuran antara alat dengan secara manual memiliki nilai yang yang berbeda, yang menyebabkan kesalahan dalam pengukuran
2. Posisi letak dari alat tersebut mmpengaruhi hasil dari pengukuran terutama untuk mengukur tinggi badan.
3. Untuk mengukur berat badan, antara timbangan manual dengan alat digital harus dikalibrasi terlebih dahulu agar hasil yang dicapai dapat lebih baik.
4. Persentase kesalahan pada alat ini dalam tujuh kali percobaan adalah untuk berat badan sebesar 2,77 % dan untuk tinggi badan sebesar1,5 %.
5. Kesalahan yang paling menonjol adalah ketika mengukur berat badan, karena sensor yang digunakan potensio geser bukan sensor berat.

5.2.Saran-saran

Dari keseluruhan proses perancangan, pembuatan dan pengujian tugas akhir ini, penulis merasa masih ada kelemahan-kelemahan yang perlu dibenahi untuk kesempurnaanya lebih lanjut, antara lain:

1. Penelitian ini dapat dikembangkan lagi untuk mencapai hasil yang lebih baik, diantaranya yaitu dibagian sensor berat dengan menggunakan load cell.

2. Wadah dari sensor ultrasonik kurang bisa dioptimalkan dalam pemakaian alat ini, diharapkan untuk kedepannya dapat menggunakan wadah yang bisa mengoptimalkan kerja dari ultrasonik.

DAFTAR PUSTAKA

Andrianto, Heri, 2008, "*Pemrograman Mikrokontroller AVR ATMEGA8535 menggunakan bahasa C (CodeVisison AVR) "* Informatika, Bandung

Atmel, 20011,"*ATMEGA162"*, http://www.atmel.com/dyn/resources/prod_documents/doc2513.pdf

Malvino, Albert Paul, 2003, "*Prinsip-prinsip Elektronika*" Terjemahan Alb Joko Sutoso, Salemba Teknik, Jakarta

Nurdiansyah, Yanuar, 2005. *Termometer Badan Dengan Output Suara Berbasis Mikrokontroller AT89S51*. Skripsi, tidak diterbitkan. Jogjakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Wicaksono, Arie, 2006. *Pengukur Tinggi dan Berat Badan Berbasis Mikrokontroller*. Skripsi, tidak diterbitkan. Jogjakarta : Fakultas Teknik Industri Universitas Islam Indonesia.

Woolard, Barry. 2003. *Elektronika Praktis*. Pradya Paramita. Jakarta

Lampiran

Lampiran 1

Listing Program

```
This program was produced by the
CodeWizardAVR V2.03.4 Standard
Automatic Program Generator
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com
Project :
Version :
Date    : 8/4/2011
Author  :
Company :
Comments:

Chip type           : ATmega16
Program type        : Application
Clock frequency     : 12.000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
// Standard Input/Output functions
#include <stdio.h>
#include <delay.h>
#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
// Declare your global variables here
//void keypad (void);
//void cleard(void);
unsigned int baca_berat();
void voice (unsigned char i);
unsigned int baca_tinggi();
void show (int);
unsigned int temp, vc0, vc1, vc2,tinggi,berat;
```

```
unsigned char voice_code[]
={0x00,0x00,0x05,0x0a,0x0f,0x14,0x17,0x1a,0x1e,0x22,0x25,0x27,0x2a,0x2d,0x31,0x34,0x37,0
x3b};
                //          1   , 2  ,3   ,4   , 5  , 6  , 7  , 8  , 9  ,se  ,pluh,rtus,blas,tggi,bert ,cm ,kg
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=T State6=T State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTA=0x00;
DDRA=0x3F;
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=Out Func0=Out
// State7=T State6=T State5=T State4=T State3=T State2=1 State1=1 State0=1
PORTB=0x07;
DDRB=0x07;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=P State3=P State2=T State1=T State0=T
PORTD=0x18;
DDRD=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
```

```
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// ADC initialization
// ADC Clock frequency: 375.000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x85;

// LCD module initialization
lcd_init(16);
   lcd_gotoxy(0,1);
            lcd_putsf("Tinggi:     Cm");
            lcd_gotoxy(0,0);
            lcd_putsf("Berat :     Kg");
while (1)
     {
     // Place your code here
                        //delay_ms (1000);
       if (PIND.3==0)
```

```
{delay_ms(100);
berat = baca_berat();
                lcd_gotoxy(8,0);
                show (berat);
if (berat>0)
 {voice(0x34);    //berat  <=== Suara
 vc0 = berat / 100;
        temp = berat % 100;
        vc1 = temp / 10;
        vc2 = temp % 10;
                                if (vc0 == 1)
                                        {
                                                voice (0x25); // se
                                                voice (0x2a); // ratus
                                        }
                                if ((vc0 > 1) && (vc0 < 10))
                                        {
                                                voice (voice_code[vc0]); //
                                                voice (0x2a); // ratus
                                        }
                                if ((vc1 >= 2) && (vc1 < 10))
                                        {
                                                voice ( voice_code[vc1]);
                                                voice (0x27); // puluh
                                                if (( vc2 > 0 ) && ( vc2 < 10))
                                                        {
                                                                voice ( voice_code[vc2]);
                                                        }
                                        }
                                if (vc1 == 1)
                                        {
                                                if (vc2 == 0)
                                                        {
                                                                voice (0x25); // se
                                                                voice (0x27); // puluh
                                                        }
                                                if (vc2 == 1)
                                                        {
                                                                voice (0x25);       // se
                                                                voice (0x2d);       // belas
                                                        }
                                                if (( vc2 > 1) && (vc2 < 10))
                                                        {
                                                                voice (voice_code[vc2]);
                                                                voice (0x2d); // belas
                                                        }
                                                }
                                if ((vc1 == 0) && (vc1 < 10))
                                        {
                                                voice (voice_code[vc2]);
                                        }
      voice (0x3b);             // cm suara
                }
}
                //delay_ms (1000);
```

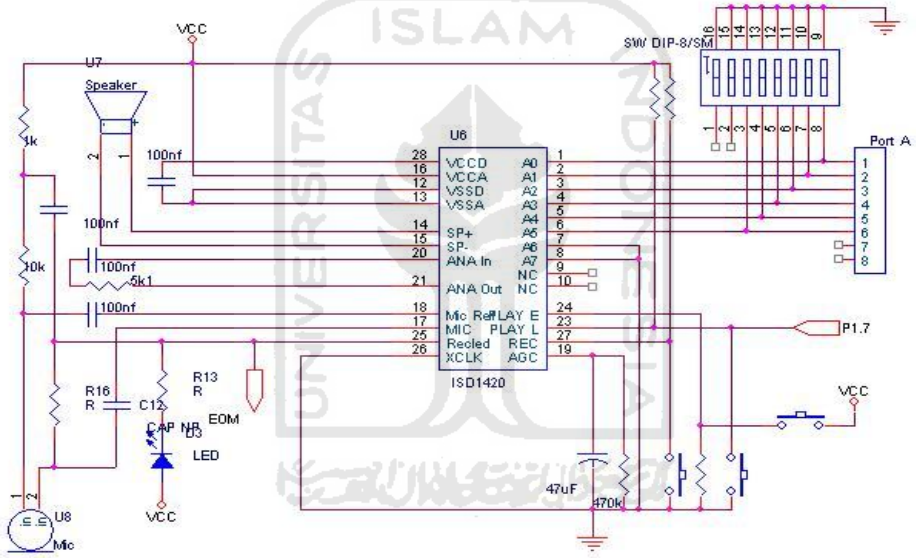
```
if (PIND.4==0)
{
delay_ms(100);
tinggi = baca_tinggi();
                lcd_gotoxy(8,1);
show (tinggi);
voice(0x31);    //tinggi  <=== Suara
vc0 = tinggi / 100;
        temp = tinggi % 100;
        vc1 = temp / 10;
        vc2 = temp % 10;
                                        if (vc0 == 1)
                                                {
                                                        voice (0x25); // se
                                                        voice (0x2a); // ratus
                                                }
                                        if ((vc0 > 1) && (vc0 < 10))
                                                {
                                                        voice (voice_code[vc0]); //
                                                        voice (0x2a); // ratus
                                                }
                                        if ((vc1 >= 2) && (vc1 < 10))
                                                {
                                                        voice ( voice_code[vc1]);
                                                        voice (0x27); // puluh
                                                        if (( vc2 > 0 ) && ( vc2 < 10))
                                                                {
                                                                        voice ( voice_code[vc2]);
                                                                }
                                                }
                                        if (vc1 == 1)
                                                {
                                                        if (vc2 == 0)
                                                                {
                                                                        voice (0x25); // se
                                                                        voice (0x27); // puluh
                                                                }
                                                        if (vc2 == 1)
                                                                {
                                                                        voice (0x25);       // se
                                                                        voice (0x2d);       // belas
                                                                }
                                                        if (( vc2 > 1) && (vc2 < 10))
                                                                {
                                                                        voice (voice_code[vc2]);
                                                                        voice (0x2d); // belas
                                                                }
                                                }
                                        if ((vc1 == 0) && (vc1 < 10))
                                                {
                                                        voice (voice_code[vc2]);
                                                }
                                        voice (0x37);                   // cm suara
}
};
```

```
}
void show (int a)
        {   int dgt0,dgt1,dgt2;

    dgt0 = a / 100;
                a %= 100;
                dgt1 = a / 10;
                dgt2 = a % 10;
                lcd_putchar(dgt0 + 0x30);
                lcd_putchar(dgt1 + 0x30);
        lcd_putchar(dgt2 + 0x30);
                delay_ms(50);
        }
void voice (unsigned char i)
        {
        PORTA = i;
        delay_us(5);
        PORTB.0=0;   //1
        delay_us(50);
        PORTB.0=1;   //1
        while (PINB.4==1);
        while (PINB.4==0);
        delay_ms (75);
        }
unsigned int baca_tinggi (void)
        {
    unsigned int jarakup,jaraklow,jarak,tempjarak;
    putchar(00);
    putchar(84);
    jarakup=getchar();
    jaraklow=getchar();
    if (jarakup==1)
      {tempjarak=jaraklow + 256;};
    if (jarakup==0)
      {tempjarak=jaraklow;};
    jarak = 200 - tempjarak;
    return jarak;
        }
unsigned int baca_berat (void)
        {
                int temp1,temp2,count;
      temp1=(510 - read_adc(7));
      temp2=((temp1/11)*5);
      count=temp2;
      return count;
        }
```

Lampiran 2

Rangkaian ISD 1420



Rangkaian Mikrokontroller dan Power Supply

