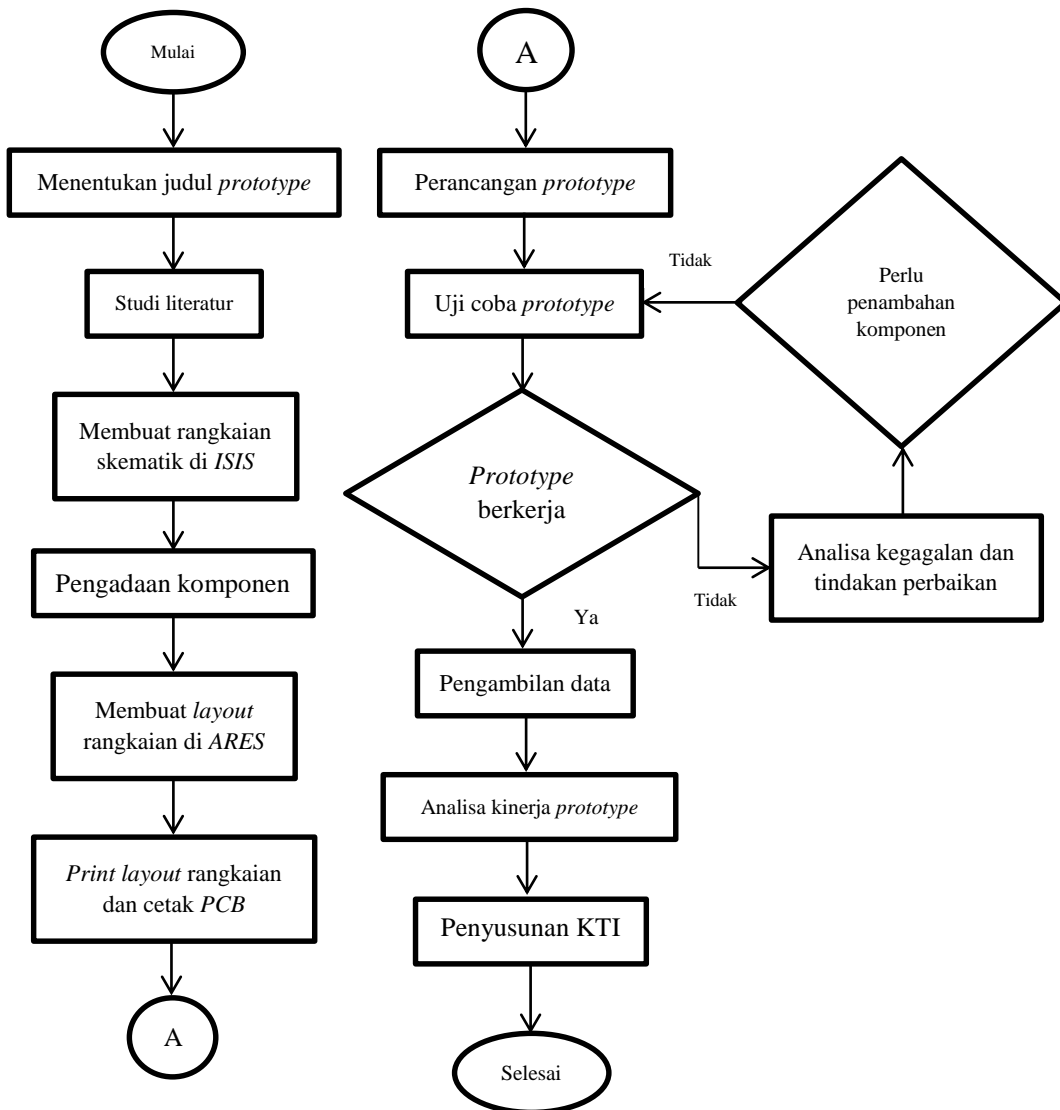


**BAB III**  
**METODOLOGI PENELITIAN**

**3.1. Perancangan Perangkat Keras**

**3.1.1. Diagram Blok KTI**

Diagram blok KTI ditunjukkan pada Gambar 3.1



**Gambar 3.1** Diagram blok KTI

### **3.1.1.1. Menentukan judul *prototype***

Dalam menentukan judul, sebelumnya penulis mencari referensi dengan membaca daftar judul Tugas Akhir beserta KTI yang ada di perpustakaan pusat UMY, perpustakaan Program Studi Teknik Elektromedik dan internet. Setelah mendapatkan judul, penulis mengkonsultasikannya dengan dosen pembimbing baik dosen pembimbing kampus maupun dosen pembimbing rumah sakit. Akhirnya setelah beberapa kali berganti judul, penulis mendapatkan judul atas rekomendasi dosen pembimbing rumah sakit untuk merancang timbangan berat badan bayi yang dilengkapi dengan *output* suara. Judul tersebut juga disetujui oleh dosen pembimbing kampus.

### **3.1.1.2. Studi literatur**

Penulis melakukan studi *literatur* dengan mencari sumber referensi yang berkaitan dengan timbangan berat badan bayi dan suara. Sumber referensi didapat dari buku, KTI, skripsi, jurnal maupun web.

### **3.1.1.3. Membuat rangkaian skematik di *ISIS***

Penulis menggunakan aplikasi *ISIS* yang terdapat dalam *PROTEUS* untuk membuat rangkaian skematik. Rangkaian skematik yang dibuat meliputi rangkaian *minimum system ATmega 16*, rangkaian penampil *LCD*, rangkaian *zeroing* dan rangkaian regulator 3,3 V. Sedangkan untuk HX711 dan *DFPlayer Mini* penulis hanya menghubungkan saja. Karena HX711 dan *DFPlayer Mini* sudah dalam bentuk modul. Walaupun dalam bentuk modul, rangkaian tersebut tetap penulis gambar sendiri di *ISIS* sebab *ISIS* tidak menyediakan *library* khusus

untuk HX711 dan *DFPLayer Mini*. Penulis menggambar rangkaian sesuai dengan yang ada dalam *datasheet* HX711 dan *DFPLayer Mini*.

#### **3.1.1.4. Pengadaan komponen**

Penulis melakukan pengadaan komponen dengan membuat daftar komponen-komponen yang dibutuhkan kemudian membelinya.

#### **3.1.1.5. Membuat *layout* rangkaian di *ARES***

Penulis membuat *layout* dari masing-masing rangkaian pada aplikasi *ARES* yang terdapat dalam *PROTEUS*. Dalam proses penggambaran *layout*, penulis menyesuaikan dengan yang ada pada rangkaian skematik. Sehingga jalur-jalurnya terhubung dengan benar.

#### **3.1.1.6. *Print layout* rangkaian dan cetak *PCB***

Penulis melakukan *print layout* masing-masing rangkaian dan mencetaknya di *PCB*. Penulis menggunakan pelarut *feriklorit* saat proses pencetakan *PCB*.

#### **3.1.1.7. Perancangan *prototype***

Penulis merancang *prototype* timbangan berat badan bayi dengan cara merangkai/menghubungkan masing-masing rangkaian menjadi satu keseluruhan yang utuh. Agar mengurangi tingkat kesalahan dalam perancangan *prototype*, penulis melakukan setiap langkah perancangan dengan hati-hati dan tetap berpanduan dengan rangkaian skematik yang telah dibuat dan *datasheet*. Sehingga *port-port* yang dihubungkan tidak salah serta agar sesuai dengan rangkaian yang ada dalam rangkaian skematik.

### **3.1.1.8. Uji coba *prototype***

Penulis melakukan uji coba *prototype*. Tujuan pengujian *prototype* yaitu untuk mengetahui sejauh mana *prototype* dapat berkerja. Jika *prototype* sudah berkerja dengan baik maka akan tampil berat badan bayi pada *LCD* dan terdengar suara dari *speaker*

### **3.1.1.9. Pengambilan data**

Penulis melakukan pengambilan data pada beban 2 Kg, 2,7 Kg dan 5 Kg sebanyak 6 kali pengukuran setiap data. Dari hasil pengambilan data dapat diketahui kinerja *prototype* apakah dapat berfungsi dengan baik atau tidak.

### **3.1.1.10. Analisa kinerja *prototype***

Penulis menganalisa kinerja *prototype* jika terjadi kegagalan/mengalami permasalahan. Cek satu per satu rangkaian, apakah sudah benar peletakan komponen-komponennya, pengecekan sumber tegangan yang masuk ataupun keluar rangkaian dan lain sebagainya sampai ditemukannya penyebab kegagalan atau permasalahan. Apabila diperlukan penambahan atau penggantian komponen, maka komponen yang akan diganti dibeli terlebih dahulu kemudian dipasang ke rangkaian yang dianalisa mengalami kegagalan/kerusakan. Setelah itu penulis melakukan pengujian kembali untuk memastikan *prototype* sudah dapat berkerja sesuai dengan yang diharapkan.

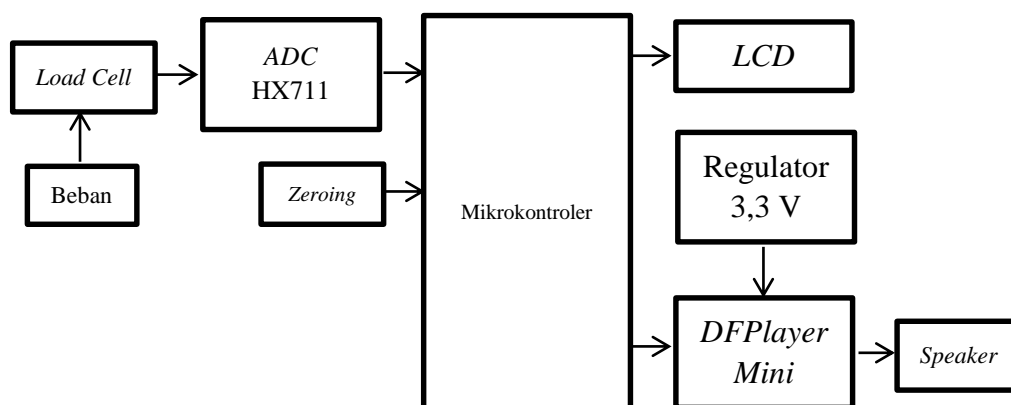
### **3.1.1.11. Penyusunan KTI**

Setelah *prototype* berfungsi dengan baik dan telah dilakukan pengujian, maka penulis menyusun KTI sampai selesai. Kemudian penulis

mengkonsultasikannya dengan dosen pembimbing kampus dan rumah sakit. Sehingga diharapkan dalam penyusunan KTI hasilnya sudah sesuai dengan harapan pembimbing serta mengurangi kesalahan penyusunan baik cara penulisan, metode dalam penelitian, pengambilan data, kesimpulan dan lain sebagainya.

### 3.1.2. Diagram Blok Sistem

Gambar 3.2 merupakan diagram blok modul yang dibuat penulis. Ketika saklar ditekan ke posisi *ON*, maka tegangan jala-jala PLN akan masuk dan memberikan *supply* tegangan ke semua rangkaian serta menghidupkan modul.



**Gambar 3.2** Diagram blok

Sebelum pengukuran berat badan bayi dimulai, sebaiknya *user* memastikan tidak ada beban yang terbaca pada modul dengan menekan tombol *zeroing*. Ini dimaksudkan agar saat pengukuran nanti, modul melakukan pembacaan mulai dari nol sehingga hasilnya merupakan angka *real* dari bayi dan bukan hasil akumulasi antara berat badan bayi dengan beban yang terbaca

sebelum pengukuran. Contoh beban yang terbaca sebelum pengukuran antara lain tempat bayi, bantal, selimut dan lain-lain.

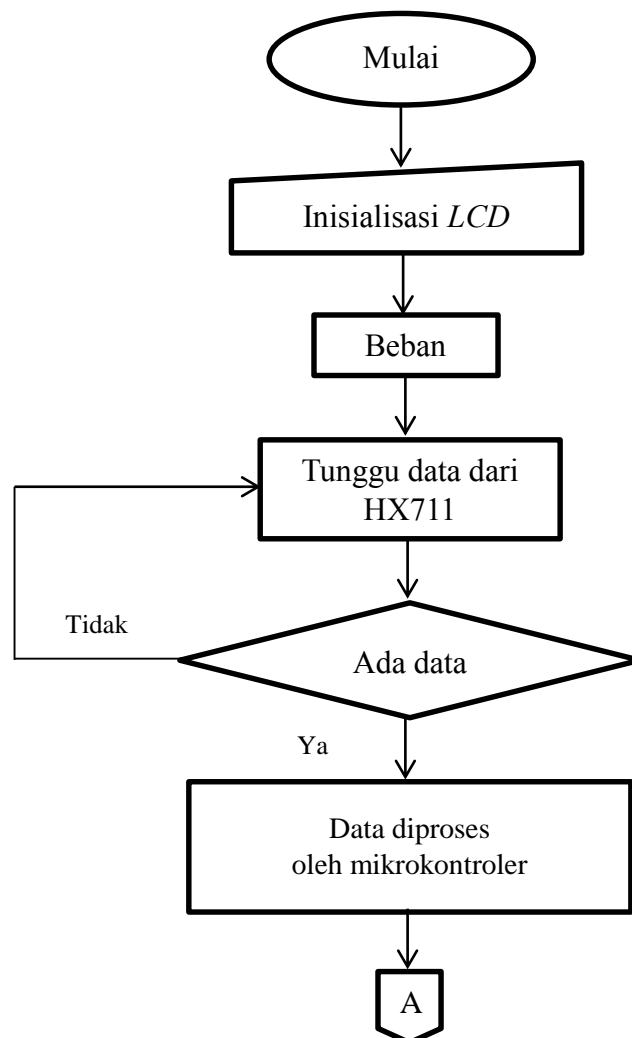
Saat beban mengenai *load cell*, maka *load cell* akan tertekan dan beban tersebut dapat dideteksi. Setelah itu, *load cell* akan membaca beban yang terukur berupa tegangan. Tegangan keluaran *load cell* masih terlalu kecil berkisar 1 mV sehingga tegangan tersebut perlu dikuatkan lagi agar dapat diolah oleh Mikrokontroler. Mikrokontroler juga mengolah rekaman suara yang ada pada modul *DFPlayer Mini*. Modul *DFPlayer Mini* mendapatkan *supply* tegangan 3,3 V dari rangkaian regulator 3,3 V.

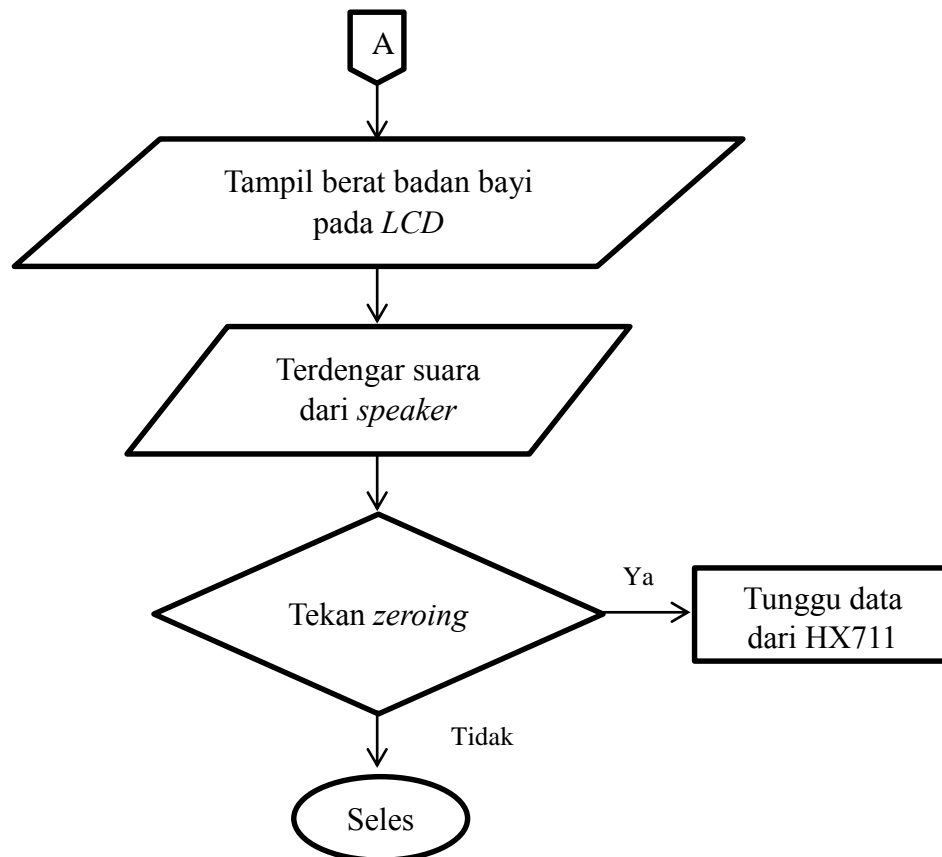
Rangkaian *ADC HX711* berfungsi sebagai penguat tegangan keluaran *load cell*. Data keluaran *load cell* yang masih berupa data analog akan diubah menjadi data digital oleh *ADC HX711* sehingga hasilnya ditampilkan ke *LCD*. Selang beberapa saat setelah hasil ditampilkan pada *LCD*, *speaker* akan aktif dan mengeluarkan suara sesuai dengan yang tertera pada *LCD*. Keluaran suara tersebut sebelumnya telah direkam dan disimpan ke *Micro SD Card*.

### **3.1.3. Diagram Alir**

Diagram alir merupakan diagram yang menjelaskan urutan kerja *prototype* dari awal sampai akhir. Adapun urutan kerja *prototype* dapat dilihat pada Gambar 3.3. Ketika *prototype* dinyalakan dengan menekan tombol *ON/OFF* ke posisi *ON*, maka *prototype* akan melakukan inisialisasi *LCD*. Adapun tampilan inisialisasi *LCD* antara lain yaitu salam, kalimat pembuka, nama universitas, nama mahasiswa, NIM, dan judul *prototype*. Pada saat beban diletakkan di atas tempat yang telah disediakan, maka beban tersebut akan memberikan tekanan ke *load*

*cell*. Kemudian *load cell* akan membaca beban yang terdeteksi. Tegangan keluaran *load cell* masuk ke rangkaian *ADC HX711* untuk dikuatkan. Tunggu data dari *HX711* sampai stabil. Data yang stabil akan dikirimkan sekaligus diproses oleh mikrokontroler. Setelah mikrokontroler memproses data tersebut maka data berupa berat akan ditampilkan pada layar *LCD*. Selang beberapa saat kemudian akan terdengar suara dari *speaker*, yaitu data berat badan bayi yang terukur sesuai dengan tampilan yang ada pada *LCD*.





**Gambar 3.3** Diagram alir

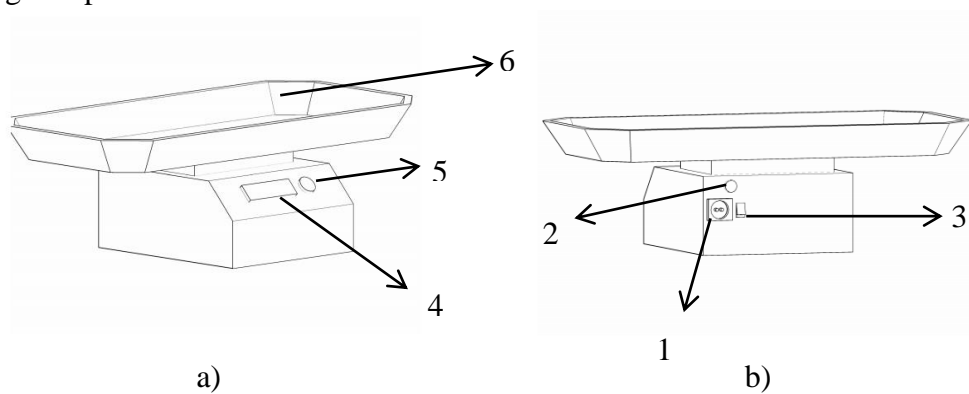
Jika tombol *zeroing* ditekan, maka proses akan kembali ke tunggu data dari HX711 dan berlanjut kembali sampai tampil *LCD* dan terdengar suara dari *speaker*. Sedangkan jika tombol *zeroing* tidak ditekan, maka proses pengukuran selesai. Tombol *zeroing* berfungsi untuk meng-nolkan beban yang terbaca agar beban tersebut tidak mempengaruhi hasil pembacaan yang sebenarnya.

#### 3.1.4. Diagram Mekanik Sistem

Gambar 3.4 merupakan rancangan mekanik timbangan berat badan bayi yang dibuat penulis. Adapun dalam perancangannya penulis menggunakan bahan yang terbuat dari besi dan akrilik. Besi digunakan untuk kerangka *prototype* sedangkan akrilik digunakan sebagai penutup kerangka tersebut. Penulis memilih



bahan besi untuk kerangka *prototype* karena besi memiliki sifat yang kuat sehingga dapat menopang beban maksimal *prototype* yang dibuat. Sedangkan untuk penutup kerangkanya, penulis memilih bahan akrilik karena akrilik memiliki sifat yang lentur dan tidak keras sehingga mudah dibentuk sesuai dengan keinginan penulis.



**Gambar 3.4** Diagram mekanik: a)Tampak depan ;b)Tampak belakang

Keterangan pada Gambar 3.4 adalah sebagai berikut:

1. Kabel AC
2. Fuse
3. Saklar *ON/OFF*
4. LCD
5. Tombol *zeroing*
6. Tempat bayi

### 3.1.5. Spesifikasi *Prototype*

Adapun spesifikasi dari *prototype* timbangan berat badan bayi dengan *output* suara antara lain:

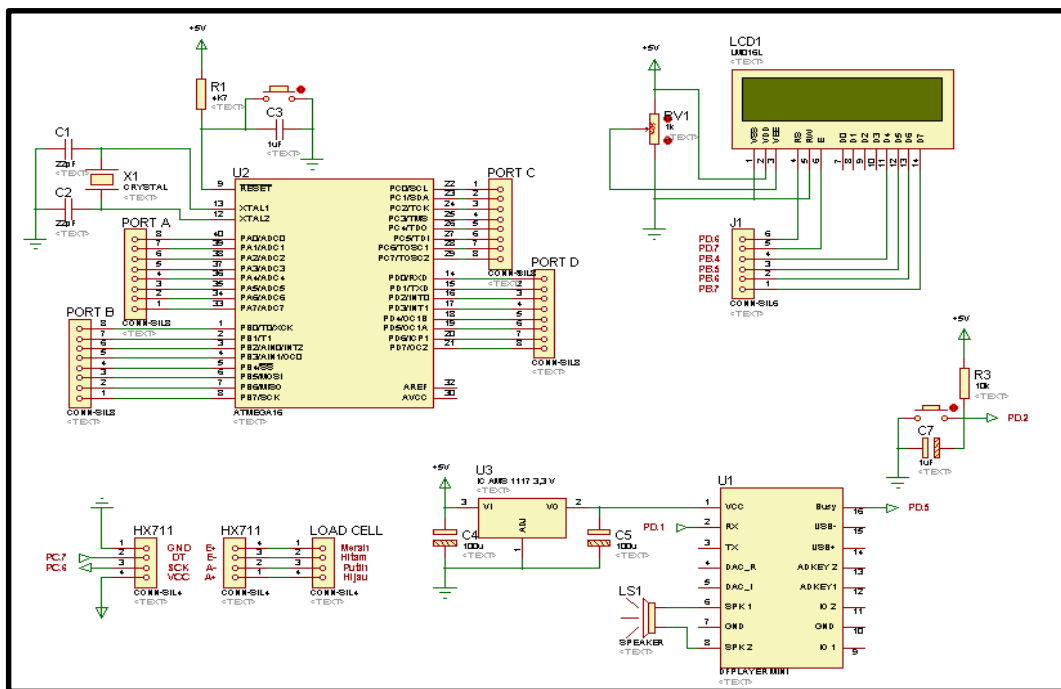
1. Nama : Timbangan Berat Badan Bayi Dengan *Output* Suara

Berbasis Mikrokontroler ATmega 16

2. Jenis : Timbangan Bayi Digital
3. Kapasitas : 20 Kg
4. Display : LCD karakter 16x2
5. Dimensi : 22 cm x 21 cm x 11 cm
6. Sensor : Load Cell

3.1.6. Rangkaian Keseluruhan

Seperti terlihat pada Gambar 3.5, *prototype* tersusun dari beberapa rangkaian antara lain rangkaian *minimum system* ATmega 16, rangkaian penampil LCD, rangkaian ADC HX711, rangkaian *zeroing*, rangkaian regulator 3,3 V dan modul *DFPlayer Mini*. Rangkaian *minimum system* ATmega 16 berfungsi sebagai otak atau pengendali utama dari semua rangkaian.

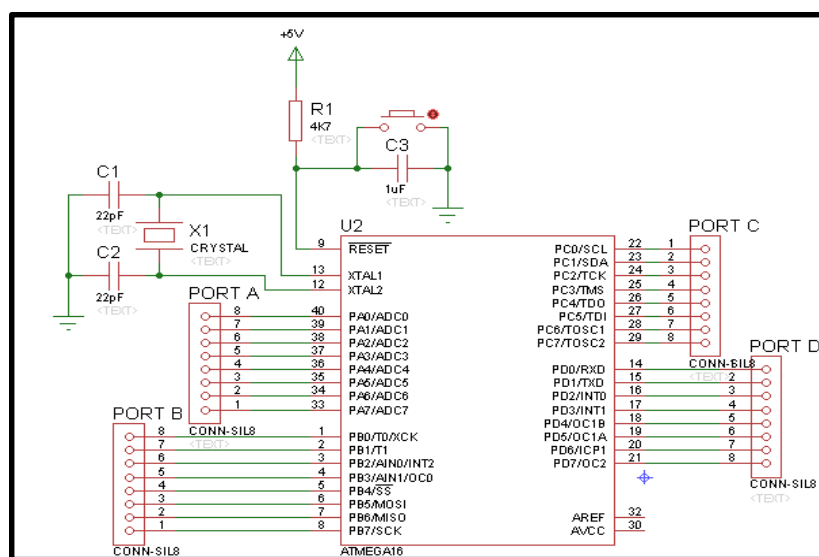


Gambar 3.5 Rangkaian keseluruhan *prototype*

Rangkaian penampil *LCD* berfungsi untuk menampilkan keluaran *prototype* berupa karakter serta jumlah berat badan bayi yang terukur. Rangkaian *ADC HX711* berfungsi sebagai pengubah data analog menjadi data digital. Selain itu, rangkaian *ADC HX711* juga digunakan sebagai penguat keluaran *Load Cell*. Rangkaian *zeroing* berfungsi untuk meng-nolkan beban yang terukur sebelum bayi ditimbang. Rangkaian regulator 3,3 V berfungsi untuk mengubah tegangan dari 5 V menjadi 3,3 V yang selanjutnya tegangan tersebut digunakan untuk memberikan *supply* ke modul *DFPlayer Mini*. Modul *DFPlayer Mini* berfungsi sebagai media untuk menyimpan *file* suara yang sudah direkam dan dimasukkan ke *Micro SD Card*.

### 3.1.7. Rangkaian Minimum System ATmega 16

Rangkaian *minimum system ATmega 16* seperti yang terlihat pada Gambar 3.6. adalah rangkaian yang berfungsi sebagai otak atau pengendali utama *prototype* yang dibuat.



**Gambar 3.6** Rangkaian skematik *minimum system ATmega 16*

Rangkaian tersebut membutuhkan *supply* tegangan sebesar 5 V DC dan membutuhkan sambungan *MISO*, *MOSI*, *SCK* sebagai jalur transfer data. Terdapat tombol reset yang digunakan untuk mereset program ketika terjadi masalah pada saat pengoperasian sehingga program akan kembali ke awal.

### 3.1.7.1. Alat

Berikut ini adalah alat yang dibutuhkan dalam pembuatan *Minimum System ATmega16*. Daftar alat dapat dilihat pada Tabel 3.1.

**Tabel 3.1** Daftar alat yang digunakan

No.	Nama Alat	Jumlah
1.	Solder	1 buah
2.	Atraktor	1 buah
3.	Multimeter	1 buah
4.	Tang Potong	1 buah
5.	Adaptor	1 buah
6.	Bor	1 buah
7.	Gergaji	1 buah

### 3.1.7.2. Bahan

Berikut ini adalah bahan yang dibutuhkan dalam pembuatan *Minimum System ATmega16*. Daftar bahan dapat dilihat pada Tabel 3.2.

**Tabel 3.2** Daftar bahan yang digunakan

No.	Nama Bahan	Jumlah
1.	<i>PCB</i>	1 buah
2.	Pelarut	1 buah
3.	Amplas	1 buah
4.	<i>Pin Sisir</i>	3 buah

5.	IC ATmega 16	1 buah
6.	Soket IC 40 pin	1 buah
7.	Kapasitor 22 pF	2 buah
8.	Kapasitor 100 nF	1 buah
9.	Resistor 4K7	1 buah
10.	Crystal 11 MHz	1 buah
11.	Push Button	1 buah
12.	Kabel konektor	Secukupnya

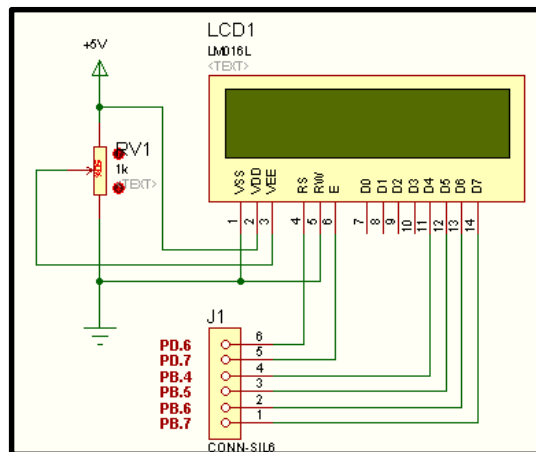
### 3.1.8. Rangkaian Penampil LCD

Rangkaian penampil LCD berfungsi untuk menampilkan hasil pengukuran berat badan bayi. Hasil tampilannya dapat dilihat pada Gambar 3.7.



**Gambar 3.7** Tampilan LCD

Seperti terlihat pada Gambar 3.8, rangkaian penampil LCD memiliki 14 pin yang terdiri dari VSS, VDD, VEE, RS, RW, E dan D0-D14. Pin 1 VSS dihubungkan dengan pin 5 RW. Pin 2 VDD dihubungkan ke +5V DC sebagai supply tegangan. Pin 3 VEE dihubungkan ke trimpot 20 K untuk mengatur tingkat kecerahan LCD. Pin 4 RS dihubungkan ke port D.6 sebagai selektor register. Pin 6 E dihubungkan ke port D.7 sebagai Enable Clock LCD. Pin 11-14 (D0-D4) dihubungkan ke port B.4, B.5, B.6, B.7 sebagai data yang dipakai untuk dikirimkan ke LCD.



**Gambar 3.8** Rangkaian skematik penampil LCD

### 3.1.8.1. Alat

Berikut ini adalah alat yang dibutuhkan dalam pembuatan rangkaian penampil LCD. Daftar alat dapat dilihat pada Tabel 3.3.

**Tabel 3.3** Daftar alat yang digunakan

No.	Nama Alat	Jumlah
1.	Solder	1 buah
2.	Atraktor	1 buah
3.	Multimeter	1 buah
4.	Tang Potong	1 buah
5.	Bor	1 buah
6.	Gergaji	1 buah

### 3.1.8.2. Bahan

Berikut ini adalah bahan yang dibutuhkan dalam pembuatan rangkaian penampil LCD. Daftar bahan dapat dilihat pada Tabel 3.4.

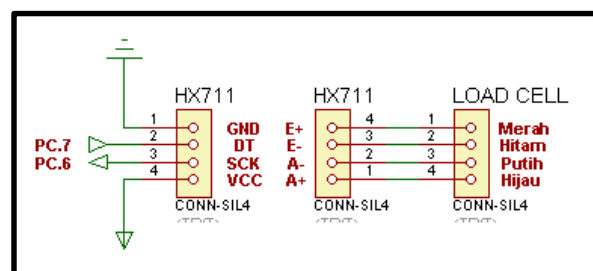
**Tabel 3.4** Daftar bahan yang digunakan

No.	Nama Bahan	Jumlah
1.	PCB	1 buah

2.	Pelarut	1 buah
3.	Amplas	1 buah
4.	Pin Sisir	3 buah
5.	LCD 16x2	1 buah
6.	Trimpot 20K	1 buah
7.	Kabel konektor	Secukupnya

### 3.1.9. Rangkaian ADC HX711

Rangkaian ADC HX711 merupakan rangkaian yang berfungsi sebagai *Analog to Digital Converter (ADC)* dan juga untuk menguatkan tegangan *load cell*. Besar penguatan yang dipakai yaitu 128 kali. *Channel* yang dipakai pada *prototype* yaitu *channel A*. *Pin data (DT)* HX711 dihubungkan ke *port C.7* sebagai *input* mikrokontroler. Sedangkan *pin serial clock (SCK)* HX711 dihubungkan ke *PORT C.6* sebagai *output* mikrokontroler. Pin E+, E-, A-, A+ HX711 dihubungkan ke kabel *load cell* dengan urutan merah, hitam, putih dan hijau. Sehingga pemasangannya seperti terlihat pada Gambar 3.9.



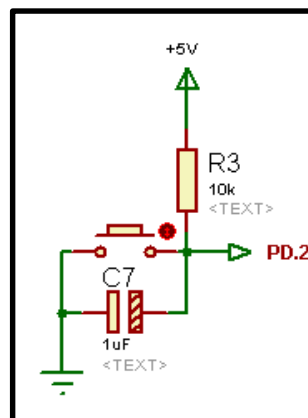
**Gambar 3.9** Rangkaian skematik ADC HX711 dan *load cell*

### 3.1.10. Rangkaian Zeroing

Rangkaian *zeroing* merupakan rangkaian yang berfungsi untuk menolkan beban yang terbaca sebelum pengukuran. Salah satu contoh beban yang

terbaca sebelum pengukuran yaitu alas/selimit bayi. Jadi dengan menekan tombol *zeroing*, angka yang tertampil pada *LCD* akan menjadi nol. Dengan adanya tombol tersebut akan memudahkan *user* untuk membaca hasil pengukuran. Karena jika tidak ada tombol tersebut dan *display* tetap menunjukkan hasil sebelum pengukuran maka untuk mendapatkan hasil pengukuran yang akurat, *user* harus mengurangi hasil akhir pengukuran dengan hasil yang terbaca sebelum pengukuran agar didapatkan hasil pengukuran yang sebenarnya.

Pada Gambar 3.10 merupakan rangkaian skematik *zeroing*. Fungsi resistor 10k yaitu untuk menghambat aliran arus agar tidak konslet. Sedangkan kapasitor 1 uF yang terhubung ke *push button* digunakan sebagai penstabil dan juga sebagai anti *bouncing* (anti pantul). Artinya ketika tombol *push button* ditekan sekali, mikrokontroler tidak membaca penekanan berkali-kali.



**Gambar 3.10** Rangkaian skematik *zeroing*

### 3.1.10.1. Alat

Berikut ini adalah alat yang dibutuhkan dalam pembuatan rangkaian *zeroing*. Daftar alat dapat dilihat pada Tabel 3.5.



**Tabel 3.5** Daftar alat yang digunakan

No.	Nama Alat	Jumlah
1.	Solder	1 buah
2.	Atraktor	1 buah
3.	Multimeter	1 buah
4.	Tang Potong	1 buah

**3.1.10.2. Bahan**

Berikut ini adalah bahan yang dibutuhkan dalam pembuatan rangkaian *zeroing*. Daftar bahan dapat dilihat pada Tabel 3.6.

**Tabel 3.6** Daftar bahan yang digunakan

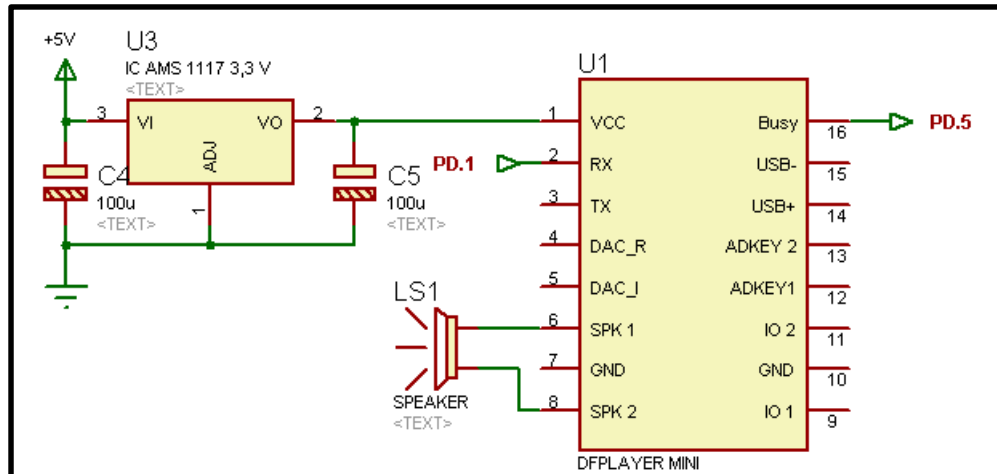
No.	Nama Bahan	Jumlah
1.	<i>PCB</i> Bolong	1 buah
2.	Amplas	1 buah
3.	Kabel konektor	Secukupnya
4.	Kapasitor 1uF/50V	1 buah
5.	Resistor 10K	1 buah
6.	<i>Push Button</i>	1 buah

**3.1.11. Rangkaian Regulator 3,3 V dan Modul *DFPlayer Mini***

Rangkaian regulator 3,3 V merupakan rangkaian yang digunakan untuk menurunkan tegangan *input* dari 5 V DC ke tegangan 3,3 V DC, kapasitor 100 uF pada rangkaian digunakan untuk penstabil listrik sebelum dan sesudah diregulasi. Rangkaian regulator 3,3 V dapat dilihat pada Gambar 3.11.

Tegangan 3,3 V digunakan untuk *supply* tegangan pada modul *DFPlayer Mini* sehingga modul dapat bekerja. Pin 6 dan 8 modul *DFPlayer Mini* dihubungkan ke *speaker* sebagai media keluaran suaranya. Untuk penyimpanan

file suara digunakan *Micro SD Card* 4 GB yang dipasangkan ke modul *DFPlayer Mini*.



**Gambar 3.11** Rangkaian skematik regulator 3,3 V dan modul *DFPlayer Mini*

#### 3.1.11.1. Alat

Berikut ini adalah alat yang dibutuhkan dalam pembuatan rangkaian regulator 3,3 V dan modul *DFPlayer Mini*. Daftar alat dapat dilihat pada Tabel 3.7.

**Tabel 3.7** Daftar alat yang digunakan

No.	Nama Alat	Jumlah
1.	Solder	1 buah
2.	Atraktor	1 buah
3.	Multimeter	1 buah
4.	Tang Potong	1 buah

#### 3.1.11.2. Bahan

Berikut ini adalah bahan yang dibutuhkan dalam pembuatan rangkaian regulator 3,3 V dan modul *DFPlayer Mini*. Daftar bahan dapat dilihat pada Tabel 3.8.

**Tabel 3.8** Daftar bahan yang digunakan

No.	Nama Bahan	Jumlah
1.	Amplas	1 buah
2.	Kabel konektor	Secukupnya
3.	Kapasitor 100uF/16V	2 buah
4.	Resistor 1K	1 buah
5.	<i>Ic</i> AMS 1117 3,3 V	1 buah
6.	Modul <i>DFPlayer Mini</i>	1 buah
7.	<i>Micro SD Card</i> 4 GB	1 buah
8.	<i>Speaker</i>	1 buah

### 3.2. Perancangan Perangkat Lunak (*Software*)

#### 3.2.1. Algoritma

Dalam perancangan program pada *prototype* timbangan berat badan bayi dengan output suara, penulis menggunakan *software AVR Studio 4*. Bahasa yang digunakan pada AVR studio 4 adalah bahasa *assembly*. Algoritma pembacaan beban dan pengolahan suara sebagai berikut:

1. Memanggil *file* yang akan disertakan. Seperti yang terlihat pada *Listing* program 3.1 merupakan instruksi yang digunakan untuk menyertakan *file*.

```
.include    = "m16adef.inc"
#include = "macro.avr"
```

*Listing 3.1 Library*

2. Inisialisasi *LCD*. Seperti yang terlihat pada *Listing 3.2* merupakan instruksi yang digunakan untuk tampilan awal *LCD*.

```
;tampilan awal LCD
lcd32 (salam1)
rcall delay2
lcd32 (salam2)
```

```

rcall delay2
lcd32 (salam3)
rcall delay2
lcd32 (salam4)
rcall delay2
lcd32 (salam5)
rcall delay2
lcd32 (salam6)
rcall delay2
lcd32 (salam7)

ldi          track,17
ldi          cmd,play
call         tx_send

rcall delay2
lcd32 (salam8)
rcall delay2
lcd32 (salam9)
rcall delay2

```

*Listing 3.2 Program tampilan LCD*

3. Program pembacaan ADC seperti yang terlihat pada *Listing 3.3*.

```

Call  hx_capture
clr   ounce_ratusan
clr   ounce_puluhan

```

*Listing 3.3 Program ambil data HX711*

Ketika ADC HX711 sudah stabil, maka ADC HX711 akan memberitahukan mikrokontroler bahwa data sudah siap untuk diambil.

4. Program *auto tare* seperti yang terlihat pada *Listing 3.4*.

```

auto_tare:
lds      bebas,raw_byte0
sts      min_byte0,bebas
lds      bebas,raw_byte1
sts      min_byte1,bebas
lds      bebas,raw_byte2
sts      min_byte2,bebas

lds      bebas,raw_disp0
sts      min_disp0,bebas
lds      bebas,raw_disp1
sts      min_disp1,bebas
lds      bebas,raw_disp2

```

```

sts     min_disp2,bebas
lds     bebas,raw_disp3
sts     min_disp3,bebas
lds     bebas,raw_disp4
sts     min_disp4,bebas
lds     bebas,raw_disp5
sts     min_disp5,bebas

clr     min_hold

```

Listing 3.4 Program *auto tare*

*Auto tare* merupakan suatu kondisi ketika *prototype* sudah siap untuk melakukan pengukuran maka akan memberitahukan bahwa *prototype* akan ke tampilan nol. Artinya, awal dinyalakan setelah inisialisasi tanpa harus menekan tombol *zeroing*, *prototype* akan menunjukkan tampilan angka nol. Hal ini dilakukan dengan menyimpan nilai *variable* “*raw*” ke *variable* “*min*”. Maka hasilnya nilai “*raw = min*”. maka “*Raw-min = 0*”. Sehingga nilainya akan menjadi nol. Nilai *variable* “*min*” akan di *update* menjadi nilai “*min*” yang baru setiap kali menekan tombol *zeroing*.

##### 5. Program perhitungan setelah *tare* seperti terlihat pada Listing 3.5.

```

after_tared:
lds     tared0,raw_byte0
lds     tared1,raw_byte1
lds     tared2,raw_byte2
lds     bebas,min_byte0
lds     bebas1,min_byte1
lds     bebas2,min_byte2
clc
;sub    tared2,bebas2
sub     tared1,bebas1
sbc     tared0,bebas

mov     bebas,tared0
swap   bebas
andi   bebas,$0F
cpi    bebas,10
brsh   more1
ldi    bebas1,'0'

```

```

add     bebas,bebas1
sts     tared_disp0,bebas
rjmp   tared_dig2
more1:
ldi     bebas1,55
add     bebas,bebas1
sts     tared_disp0,bebas

```

*Listing 3.5* Program perhitungan setelah *auto tare*

Disebut juga *variable* “*tared*”. *Variable* “*tared*” merupakan suatu kondisi setelah pembacaan HX711, nilai *raw* (mentah) akan lebih besar daripada *variable* “*min*”, ketika ada beban pada sensor. Nilai ini yang akan dimasukkan ke *variable* “*tared*”.

#### 6. Program pembacaan beban seperti terlihat pada *Listing 3.6*.

```

ounce_to_BCD:
;konversi ounce ke ASCII BCD

mov     fbin,ounce
call   bin2bcd8
ldi     bebas3,'0'
mov     ounce_satuan,bebas3
add     ounce_satuan,tBCDL
cpi     tBCDH,10                ;bandingkan hasil konversi
high dengan 10,
brsh   rekonversi              ;jika => maka konversi ulang tBCDH
mov     ounce_puluhan,bebas3    ;jika < maka data digunakan,
dipindah ke register avg64 ASCII
add     ounce_puluhan,tBCDH
mov     ounce_ratusan,bebas3
rjmp   konversi_end
rekonversi:                    ;konversi ke2 tBCDH
mov     fbin,tbcdH
call   bin2bcd8
mov     ounce_puluhan,bebas3    ; hasil dikonversi ke
ASCII dan di

add     ounce_puluhan,tBCDL     ;distribusikan ke
register yang sesuai
mov     ounce_ratusan,bebas3
add     ounce_ratusan,tBCDH
konversi_end:

```

```

;-----batas minimal timbang 0 ons

cp          ounce,nol
breq       ke_looper
rjmp       barrel_shifter_init2
ke_looper:
jmp        to_looper
direct_looper:
jmp        looper

```

*Listing 3.6* Program pembacaan beban

Ditentukan berat beban yang akan digunakan untuk kalibrasi yaitu 1 Kg. Satu Kg sama dengan 10 ons. Karena nilai terkecil pengukuran adalah ons, maka 1 Kg ada 10 ons. Dalam hal ini ada 10 langkah (*stepper*).

Ketika beban 1 Kg diberikan ke timbangan, maka nilai *variable* “*tared*” dicatat. Hasil *varibel* “*tared*” yang dicatat dimasukkan sebagai konstanta *max*. Lalu konstanta *max* akan dibagi dengan konstanta *step* (10), menghasilkan konstanta *stepper*. Setengah konstanta *stepper* adalah konstanta *half stepper*.

Ketika dilakukan pengukuran terhadap sebuah beban yang belum diketahui beratnya, maka *variable* “*tared*” dibagi dengan konstanta *stepper* menghasilkan *variable* “*multiplier*”. *Variable* “*multiplier*” ini cikal bakal hasil penimbangan dalam ons.

*Variable* “*multiplier*” akan kembali dikalikan dengan konstanta *stepper*. Lalu ditambah dengan konstanta *half stepper* menghasilkan *variabel* “*limit*”. *Variabel* “*limit*” akan dibandingkan dengan *variabel* “*tared*” . Jika *tared* => *limit*, maka *multiplier* + 1 = ons. Jika *tared* < *limit* maka *multiplier* = ons.

7. Program pengolahan suara seperti terlihat pada *Listing 3.7*.

```

after_display2:
lock:
ngomong:

```

```

ldi          bebas,$ff
sts          status,bebas

;-----
ldi          track,18
ldi          cmd,play
call        tx_send
audio_wait1:
sbis        pind,busy
rjmp        audio_wait1

;-----

digit1cek:
mov         track,fix_ratusan
subi        track,'0'
cp          track,nol
breq        digit2cek; digit pertama = 0
cpi         track,1 ; digit pertama 1
breq        belasan

;digit pertama 2
inc         track
ldi         cmd,play
call        tx_send
audio_wait20:
sbis        pind,busy
rjmp        audio_wait20

ldi         track,15
ldi         cmd,play
call        tx_send
audio_wait20up:
sbis        pind,busy
rjmp        audio_wait20up
rjmp        digit2cek

;-----digit pertama tidak nol-----
;cek digit2
belasan:
mov         track,fix_puluhan
subi        track,'0'
cpi         track,0
breq        sepuluh

cpi         track,1
breq        sebelas

;belas belas
inc         track

```



```
ldi      cmd,play
call     tx_send

audio_wait12:
sbis    pind,busy
rjmp    audio_wait12

ldi     track,13
ldi     cmd,play
call    tx_send
audio_wait12up:
sbis    pind,busy
rjmp    audio_wait12up
rjmp    say_koma

sepuluh:
ldi     track,11
ldi     cmd,play
call    tx_send
audio_wait10:
sbis    pind,busy
rjmp    audio_wait10
rjmp    say_koma

sebelas:
ldi     track,12
ldi     cmd,play
call    tx_send
audio_wait11:
sbis    pind,busy
rjmp    audio_wait11
rjmp    say_koma

;-----
digit2cek:
mov     track,fix_puluhan
subi    track,'0'
inc     track
ldi     cmd,play
call    tx_send
audio_wait3:
sbis    pind,busy
rjmp    audio_wait3

;-----
say_koma:
ldi     track,14
```

```

ldi  cmd,play
call tx_send
audio_wait4:
sbis pind,busy
rjmp audio_wait4
;-----

say_ons:

mov  track,fix_satuan
subi track,'0'
inc  track
ldi  cmd,play
call tx_send
audio_wait5:
sbis pind,busy
rjmp audio_wait5
;-----

say_kilo:
ldi  track,16
ldi  cmd,play
call tx_send
audio_wait6:
sbis pind,busy
rjmp audio_wait6

;-----

jmp  looper

to_looper:
sts  status,nol
clr  avg64_stat
clr  fix_satuan
rjmp looper

```

*Listing 3.7* Program suara

Ketika beban sudah dibaca mikrokontroler, maka mikrokontroler akan memerintahkan modul *DFPlayer mini* untuk mengeluarkan suara. Sistem pemanggilan suaranya satu-satu yaitu dengan mengecek per digit. Jika digit pertama sudah selesai maka baru pindah ke digit berikutnya sampai digit terakhir. Setelah semua digit dicek, baru dapat diputuskan untuk mengeluarkan suaranya.

Suara yang terdengar dari *speaker* akan sama dengan hasil yang tertera pada *LCD*. Sehingga kemungkinan kesalahannya sangat kecil.

### **3.3 Standar Operational Procedure (SOP)**

Standar pengoperasian *prototype* timbangan berat badan bayi dengan *output* suara antara lain sebagai berikut:

1. Siapkan *prototype* timbangan berat badan bayi.
2. Tempatkan *prototype* di tempat yang datar dan kokoh.
3. Hubungkan *prototype* ke sumber tegangan yaitu jala-jala PLN. Kemudian nyalakan *prototype* dengan menekan saklar *ON/OFF* ke posisi *ON*.
4. Tunggu sampai proses inisialisasi *LCD* selesai dan *prototype* siap digunakan.
5. Letakkan beban yang akan ditimbang pada tempat yang telah disediakan, lalu baca hasil penimbangan.
6. Setelah selesai digunakan, matikan *prototype* dengan menekan saklar *ON/OFF* ke posisi *OFF*.