

BAB III

ANALISA MASALAH DAN PERANCANGAN

III.1. Analisa

Sub bab ini berisikan tentang analisa sistem yang akan dibangun. Sub bab ini membahas teknik pemecahan masalah yang menguraikan sebuah sistem menjadi bagian-bagian komponen dengan tujuan mempelajari seberapa baik bagian-bagian komponen tersebut bekerja dan berinteraksi.

III.1.1. Analisa Masalah

Tujuan dari fase analisis adalah memahami dengan sebenar-benarnya kebutuhan dari sistem baru dan mengembangkan sebuah sistem yang mawadahi kebutuhan tersebut. Untuk mempermudah analisis sistem dalam menentukan kebutuhan secara lengkap, maka penulis membagi kebutuhan sistem kedalam dua jenis yakni, kebutuhan fungsional dan kebutuhan nonfungsional.

III.1.1.1. Analisa Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem supaya alat bekerja dengan baik. Berikut kebutuhan fungsional yang terdapat pada sistem yang dibangun :

1. Mengimplementasikan penggunaan mikrokontroler ATmega8 sebagai Pengendali utama dalam membuat simulator ayunan otomatis.

2. Sensor pendeteksi air seni yang dirancang harus mampu melakukan pendeteksian terhadap adanya air seni. Jika sistem sensor menangkap akan adanya air seni, maka sistem sensor akan memberikan informasi ke sistem mikrokontroler, kemudian sistem mikrokontroler akan mengaktifkan alarm.
3. Sensor suara yang dirancang juga harus mampu melakukan pendeteksian terhadap suara tangisan bayi. Jika sistem sensor menangkap akan adanya suara tangisan bayi, maka sistem sensor akan memberikan informasi ke sistem mikrokontroler, kemudian sistem mikrokontroler akan menggerakkan motor DC.
4. Sistem mikrokontroler harus mampu berkomunikasi dengan sensor air dan suara untuk melakukan proses pembacaan data.

III.1.1.2. Analisa Kebutuhan Nonfungsional (Perangkat)

Kebutuhan ini adalah tipe kebutuhan yang berisi properti perilaku yang dimiliki oleh sistem. Berikut adalah kebutuhan nonfungsional yang dimiliki sistem :

III.1.1.2.1. Operasional

Aplikasi dibangun dengan menggunakan komponen sebagai berikut :

1. Paket *software Sketch Arduino 1.0.5*, sebagai *tools* pemrogram mikrokontroler yang digunakan.
2. Mikrokontroler *ATMega 8* sebagai *controller* utama.
3. *Analog Sound Sensor V2* sebagai sensor suara, sensor ini terdiri dari *microfon* digital, penguat *output*, dan modulator sigma.
4. *Motor DC*

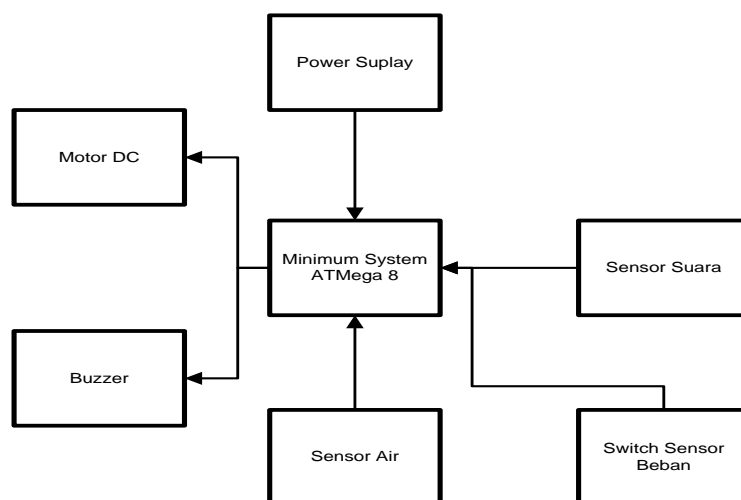
5. Sensor pendeteksi air seni dengan memanfaatkan sifat karakteristik transistor sebagai saklar.
6. Adaptor sebagai penyuplai daya.
7. Spesifikasi minimal komputer yang digunakan untuk melakukan *programming microcontroller* adalah *Processor Pentium IV 2,6 GHz*, Memori 512 MB, Kartu Grafik 128 MB dengan Sistem Operasi *Windows 7*.

III.2. Perancangan Sistem

Sub bab ini berisikan tentang rancangan sistem yang akan dibangun, dalam hal ini perancangan terhadap sistem.

III.2.1 Blok Diagram Sistem

Pada tahap awal perancangan sistem yang dilakukan adalah perancangan diagram blok. Blok diagram merupakan penyederhanaan dari rangkaian yang menyatakan hubungan berurutan dari satu atau lebih rangkaian yang memiliki kesatuan kerja tersendiri. Diagram blok aplikasi yang dirancang adalah seperti terlihat pada gambar berikut ini :



Gambar III.1. Diagram Blok

Keterangan gambar :

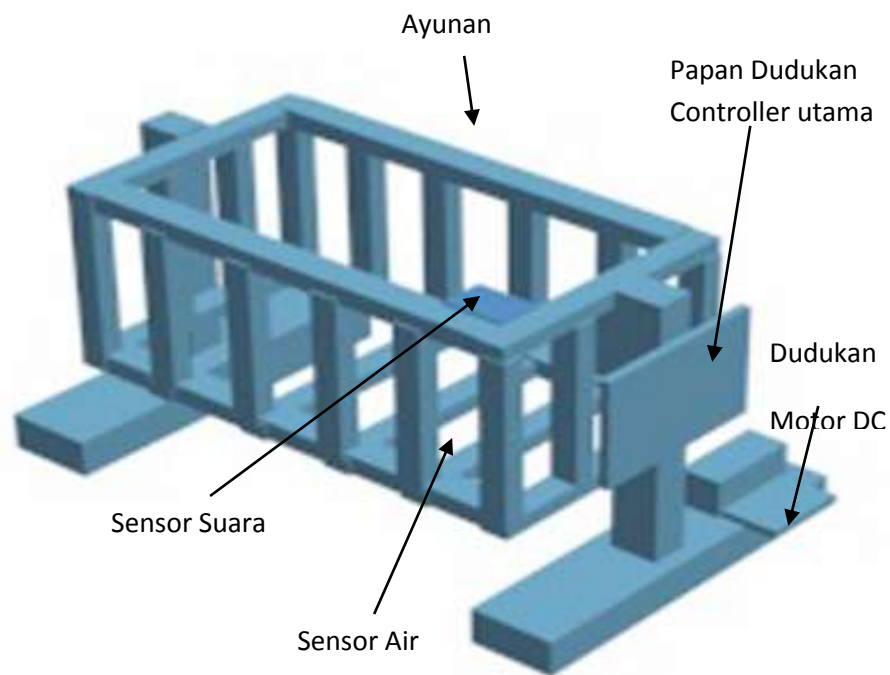
- Power suplay : Sebagai penyuplai daya dc
- Motor DC : Motor yang digunakan untuk menggerakkan ayunan
- Sistem minimum : Rangkaian utama yang dijadikan terminal kontrol
- Microcontroller : untuk menerima data dari sensor air dan sensor suara, ATmega 8 serta mengirimkan data kepada motor DC
- Swicth sensor : Sensor yang menandakan adanya bayi atau tidak di beban dalam ayunan, bila kondisi high akan memberi masukan ke micro
- Sensor suara : Merupakan suatu modul sensor yang menggunakan *Analog Sound Sensor V2* sebagai sensor suara, sensor ini terdiri dari mikrofon, penguat output, dan modulator sigma
- Sensor Air : Sensor pendeteksi akan adanya air seni bayi
- Buzzer : Sebagai penanda bayi buang air seni dan menangis

III.3 Perancangan Alat

III.3.1. Perancangan *Prototipe* Ayunan Bayi Otomatis

Perancangan *Prototipe* ayunan bayi otomatis ini dibuat bertujuan sebagai simulasi ayunan bayi otomatis mewakili ayunan bayi yang sebenarnya. *Prototipe* ayunan bayi otomatis ini dibuat dengan bentuk kotak persegi panjang dengan rongga-rongga di sekeliling ayunan. *Prototipe* ayunan bayi otomatis ini dibangun menggunakan bahan besi olo serta memiliki perangkat pendukung yakni sensor

air yang mampu untuk melakukan deteksi ketika bayi buang air kecil. Sensor suara yang memiliki fungsi untuk mendeteksi suara bayi ketika menangis dan motor DC yang berfungsi sebagai mesin penggerak ayunan bayi. *Prototipe* ayunan bayi otomatis dapat dilihat pada gambar berikut :



Gambar III.2. Rancangan Fisik Simulator Ayunan Otomatis

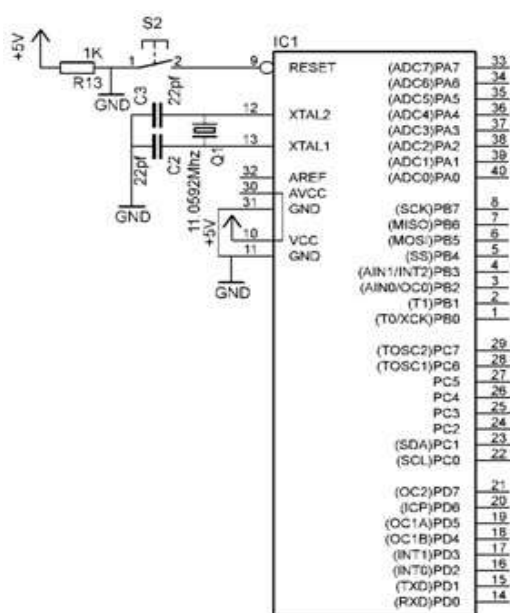
III.3.1.1. Analisa Rangkaian Sistem

Analisa rangkaian Sistem merupakan pengembangan dalam pemahaman terhadap sistem kerja alat yang merupakan satu kesatuan inti dari setiap perangkat-perangkat pendukungnya. Sistem dikatakan lengkap apabila dalam mencapai tujuan yang ingin dicapai terjadi interaksi antara sub sistem- sub sistem yang ada. Pada sub bab berikut akan dijelaskan mengenai analisa perancangan

masing-masing rangkaian yang mendukung tercapainya tujuan pembuatan alat disertai hasil pengukuran pada masing-masing rangkaian.

III.3.2. Sistem Minimum ATmega 8

AVR ATmega8 adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K *byte in-System Programmable Flash (datasheet)*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16 MIPS pada frekuensi 16MHz. Untuk ATmega8 dapat bekerja pada tegangan antara 4,5 – 5,5 V.



Gambar III.3. Sistem Minimum Mikrokontroler ATmega 8

Sistem minimum mikrokontroler adalah sistem elektronika yang terdiri dari komponen-komponen dasar yang dibutuhkan oleh suatu mikrokontroler untuk dapat berfungsi dengan baik. Pada umumnya, suatu mikrokontroler membutuhkan dua elemen (selain *power supply*) untuk berfungsi: Kristal Oscillator (XTAL), dan Rangkaian RESET. Analogi fungsi Kristal Oscillator adalah jantung pada tubuh

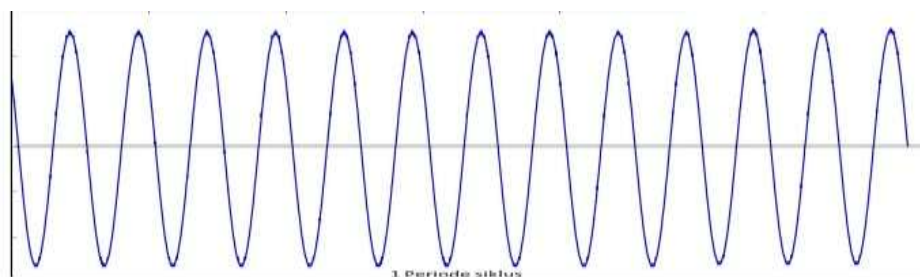
manusia. Perbedaannya, jantung memompa darah dan seluruh kandungannya, sedangkan XTAL memompa data. Dan fungsi rangkaian RESET adalah untuk membuat mikrokontroler memulai kembali pembacaan program, hal tersebut dibutuhkan pada saat mikrokontroler mengalami gangguan dalam meng-eksekusi program. Pada sistem minimum AVR khususnya ATMEGA8 terdapat elemen tambahan (optional), yaitu rangkaian pengendalian ADC: *AGND* (= *GND ADC*), *AVCC* (*VCC ADC*), dan *AREF* (= *Tegangan Referensi ADC*).

III.3.3. Analog to Digital Converter (ADC)

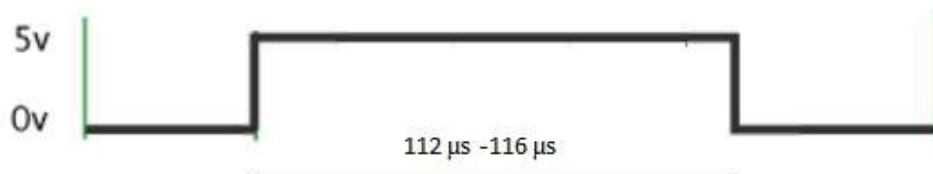
ADC digunakan untuk mengkonversi sinyal analog menjadi data digital. AVR ATMega8 telah dilengkapi ADC didalam chip, sehingga tidak diperlukan lagi ADC tambahan. Resolusi ADC selalu dinyatakan sebagai jumlah bit – bit dalam kode keluaran digital. Misalnya, ADC dengan n-bit memiliki 2^n kode digital. Disini digunakan ATMega8 model PDIP yang memiliki ADC internal dengan resolusi 8 bit dilengkapi dengan 6 channel masukan ADC0 – ADC5 terletak pada pin 23 – 28 (PC0 – PC5). Resolusi ADC menentukan ketelitian nilai hasil konversi ADC, contoh ADC 8 bit memiliki output 8 bit data digital menghasilkan bilangan 0 sampai 255 sedangkan 10 bit menghasilkan bilangan 0 sampai 1024. Sebagaimana komponen ADC pada umumnya, ADC dalam AVR juga membutuhkan tegangan referensi dapat berupa AVCC, AREF atau 2,56V. Tegangan referensi AVCC dan 2.56V berasal dari dalam AVR, sedangkan tegangan referensi AREF berasal dari luar AVR yang dimasukkan melalui pin AREF.

Rekomendasi dari pabrikan mengatakan bahwa kecepatan detak yang direkomendasikan adalah antara 50 kHz sampai dengan 200 kHz pada resolusi maksimum 10-bit. Kecepatan yang lebih tinggi daripada 200 kHz akan menurunkan akurasi. Mikrokontroler Arduino ATmega8 menggunakan clock sebesar 16 MHz, dan umumnya ADC akan memiliki sumber clock yang sama dengan mikrokontroler. Nilai ini jauh lebih besar dari yang dibutuhkan oleh ADC, yaitu maksimum hanya 200 kHz. Oleh karena itu, terdapat suatu **prescaler** pada mikrokontroler. Prescaler ini yang akan membagi nilai clock sumber yang digunakan oleh mikrokontroler.

Berdasarkan pembagian di atas, jika kita menggunakan prescaler sebesar 128, maka kecepatan ADC sebesar 125 kHz. Karena ADC membutuhkan waktu 13 detak dalam sekali konversi, maka banyaknya sampel yang bisa dikonversi oleh ADC harus dibagi lagi sebanyak 13 dari kecepatan ADC, yaitu $125/13 = 9.6$ kSa/s (kilo sampel per detik). Dari hasil perhitungan tersebut dapat disimpulkan bahwa mikrokontroler arduino Atmega8 memiliki kemampuan untuk melakukan deteksi sensor hingga 9.6 kSa.



Gambar III.4. Periode Siklus Dalam Mendeteksi Nilai ADC



Gambar III.5. Waktu Rata-rata Pengujian 1 Periode Siklus Deteksi Nilai ADC Mikrokontroler Arduino Atmega8

Gambar III.4. dan III.5. diatas menunjukkan bagaimana Mikrokontroler melakukan pembacaan nilai ADC dari sensor. Dalam melakukan pembacaan nilai ADC mikrokontroler memerlukan 13 detik dalam sekali konversi nilai deteksi yang terbaca pada sensor. Pada gambar III.5. ditunjukkan bahwa dalam melakukan satu kali konversi ADC memerlukan waktu antara $112 \mu\text{s} - 116 \mu\text{s}$. nilai waktu ini bukanlah ketetapan nilai mutlak akan tetapi nilai yang didapat dalam pengujian deteksi sensor suara (analog sensor sound V2) dan dapat berubah tergantung keadaan pengujian alat dan penggunaan komponen perangkat elektronik dari rangkaian alat yang dirancang.

Mikrokontroler memiliki kemampuan untuk menerima, mengolah, dan memberikan output signal digital. Sebagai contoh bila mikrokontroler dengan catu daya 5 volt, mikrokontroler tersebut dapat mengenali tegangan sebesar 0 volt sebagai signal digital *low*, atau tegangan 5 volt sebagai signal digital *high*. Sedangkan untuk mengenali nilai tegangan antara 0 volt hingga 5 volt, diperlukan feature khusus, yakni Analog-to-Digital Converter. Analog-to-Digital Converter atau biasa disebut ADC, memungkinkan mikrokontroler untuk mengenali suatu nilai analog melalui suatu pendekatan digital.

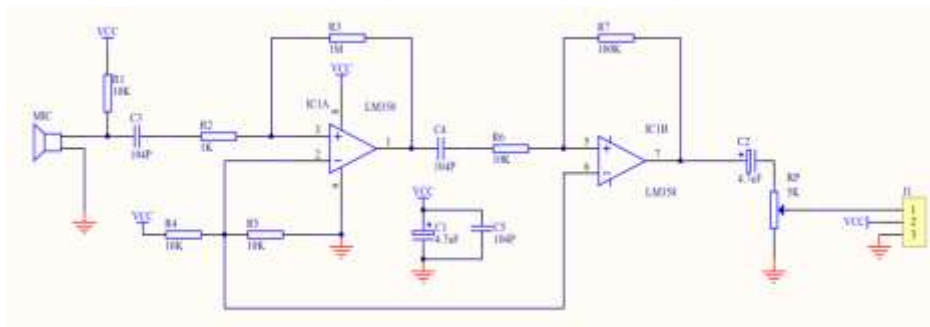
Nilai ADC menunjukkan ratio perbandingan dengan tegangan yang terbaca. Berikut persamaannya ialah nilai ADC terukur ialah nilai ADC

maksimum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber. Nilai ADC tergantung dengan tegangan yang menjadi catu daya sistem mikrokontroler. Untuk *board* Arduino biasa menggunakan sumber tegangan 5V. Berikut adalah cara mencari nilai ADC, dengan menerapkan persamaan yang ada dan tegangan terbaca sebesar 2,12 volt pada board Arduino Uno.

1. Board Arduino Uno memiliki resolusi 10 bit, dengan nilai terbesar 1023
2. Tegangan sumber 5 volt dan tegangan terbaca ialah 2,12 volt
3. Nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber
4. Sehingga diperoleh nilai ADC sebesar 434

Dengan demikian diperoleh nilai ADC sebesar 434 dari tegangan terukur 2,12 volt. Untuk setiap unit ADC tersebut memiliki perbandingan tegangan sebesar 4,9 mV. Selain dipengaruhi oleh besarnya nilai resolusi ADC, tepat tidaknya pengukuran nilai ADC juga dipengaruhi oleh clock speed ADC tersebut. Untuk board Arduino Uno sendiri clock speed ADC maksimum yang disarankan ialah 200 kHz. Nilai clock speed 200 kHz tersebut berdasarkan spesifikasi internal DAC (Digital to Analog Converter) pada rangkaian pengubahnya.

III.3.4. Sensor Suara

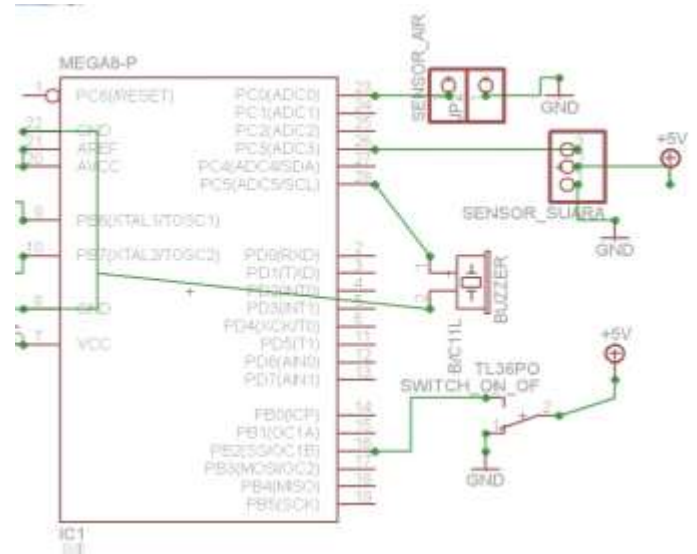


Gambar III.6. Rangkaian Sensor Suara

Sebagai sensor suara yang digunakan untuk mendeteksi suara tangisan bayi yang tidur di dalam ayunan digunakan *Analog Sound Sensor V2*. *Analog Sound Sensor V2* adalah sensor suara analog produksi *DFRobot*. Rentang tegangan kerja dari sensor ini adalah 3,3 V – 5 V. Sensor ini memiliki *electret condenser microphone DG09767CD* yang berfungsi sebagai *microphone* untuk menangkap suara. Secara teori, bila menggunakan sebuah *microphone* untuk menangkap suara, maka *microphone* akan mengubah getaran menjadi sinyal listrik. Kemudian sinyal listrik yang masih kecil ini perlu di perkuat misalnya dengan transistor. Hasilnya dimasukkan ke kapasitor dan resistor, sehingga kalau mendapatkan sinyal listrik besar, kapasitor terisi dan tegangannya akan tinggi. Berikut ini merupakan penggalan program deteksi sensor suara.

```
int analogPin = A3;
int lang = 0;
void setup()
{ Serial.begin(9600);}
void loop()
{
  lang = analogRead(analogPin);
  delay(100);
  Serial.println(lang);
}
```

III.3.5 Sensor Air

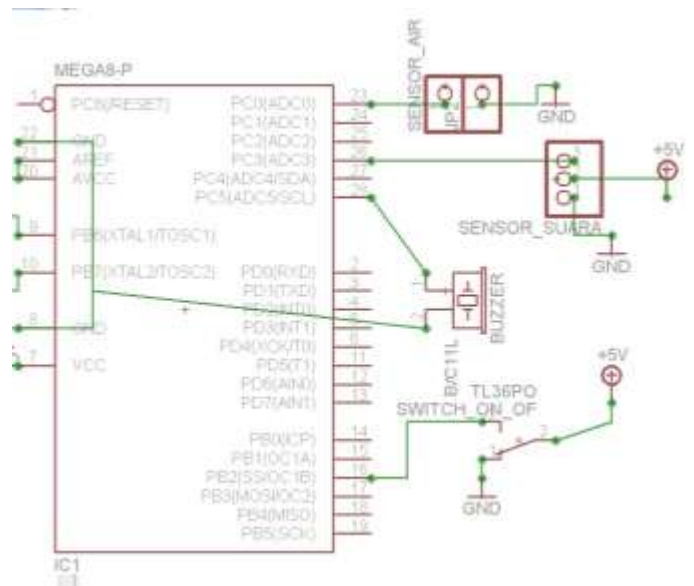


Gambar III.7. Sensor Air

Ketika air mengenai media penghantar arus listrik (dalam hal ini papan PCB yang sudah didesain sedemikian rupa), data ini dikirimkan ke mikrokontroler dan digunakan mikrokontroler sebagai parameter untuk mengaktifkan *buzzer*. Berikut ini merupakan penggalan program deteksi sensor suara.

```
int analogPin = A0;
int lang = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  lang = analogRead(analogPin);
  delay(100);
  Serial.println(lang);
}
```

II.3.6. Sensor Pendeteksi Bayi Menggunakan Switch



Gambar III.8. Sensor Saklar Pendeteksi Bayi

Ketika bayi diletakkan diatas ayunan bayi maka berat bayi akan menekan sensor saklar switch sehingga akan membuat kondisi switch bernilai high pada kaki pin B2 mikrokontroler ATmega8, data ini dikirimkan ke mikrokontroler dan digunakan mikrokontroler sebagai parameter untuk mengaktifkan sensor suara dan air serta motor DC sebagai penggerak ayunan bayi otomatis. Sebaliknya jika bayi diangkat maka saklar switch akan bernilai low. Ketika dalam kondisi low maka mikrokontroler akan menonaktifkan sensor suara, air dan motor DC sehingga ayunan bayi akan berada dalam kondisi standby. Berikut ini merupakan penggalan program deteksi keberadaan bayi.

```

int motorrelayA = 11, motorrelayB = 12, buzzer = A5;

int switch_bayi = 10;

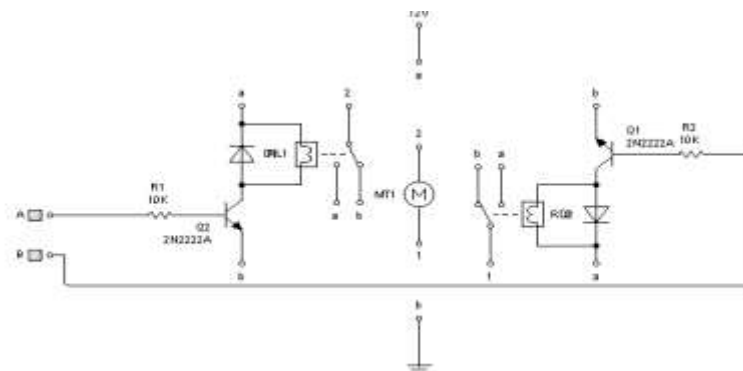
void setup() {
    pinMode(switch_bayi, INPUT);
    Serial.begin(9600);
    pinMode(buzzer, OUTPUT);
    pinMode(motorrelayA, OUTPUT);
    pinMode(motorrelayB, OUTPUT);
}

void loop(){
    int keadaanTegangan = digitalRead(switch_bayi);
    //----- disini logika utk liat switch bayi -----
    if (keadaanTegangan == HIGH)
    {
        Serial.println("Ada Bayi");

        // Pengaturan Nilai Dan Delay Goyangan Ayunan
        digitalWrite(motorrelayA, HIGH);
    }
}

```

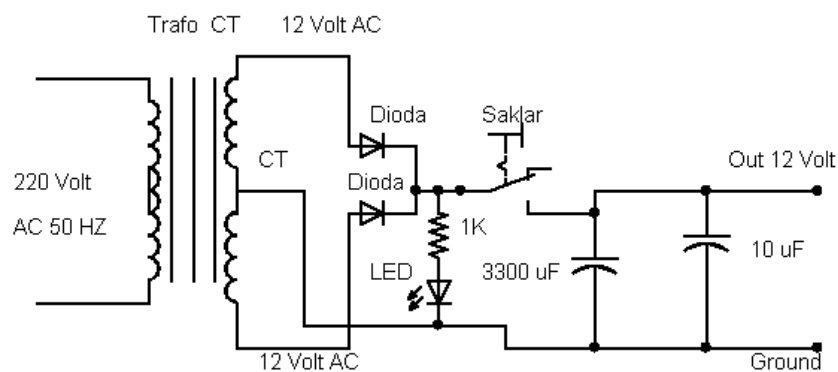
III.3.7. H-Bridge Relay



Gambar III.9. Rangkaian H-Bridge Relay

Rangkaian ini digunakan untuk menggerakkan motor *DC* yang digunakan sebagai motor penggerak ayunan. Rangkaian *H-Bridge* pada dasarnya merupakan rangkaian yang tersusun oleh sekumpulan transistor yang berfungsi sebagai *driver* motor. Khususnya motor dengan kebutuhan arus dan tegangan yang cukup besar. Namun di rangkaian ini digunakan *relay* sebagai driver penggerak motornya. Prinsip kerja rangkaian driver motor menggunakan *relay* tidak jauh berbeda dengan prinsip kerja rangkaian *H-Bridge* menggunakan *transistor*. *Relay* pada rangkaian *driver* motor ini berfungsi sebagai saklar otomatis yang akan mengatur perputaran dari motor.

III.3.8. Rangkaian Power Supply



Gambar III.10. Rangkaian Power Supply

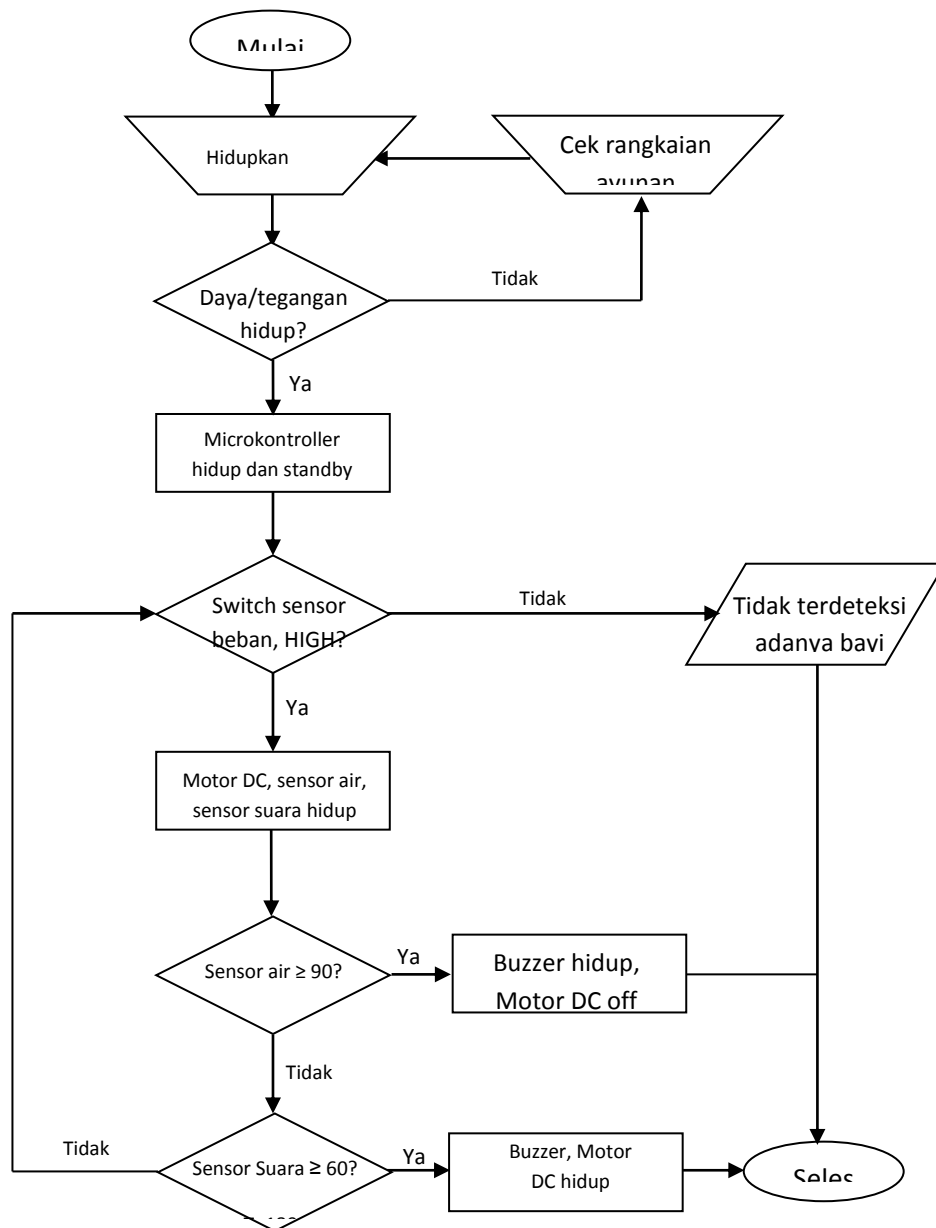
Rangkaian PSU yang dibuat terdiri dari satu keluaran, yaitu (+) 12 volt. Keluaran (+) 12 volt digunakan untuk suplai tegangan ke seluruh rangkaian. Trafo merupakan trafo *stepdown* yang berfungsi untuk menurunkan tegangan dari 220 volt AC menjadi 12 volt AC. Kemudian 12 volt AC akan disearahkan dengan menggunakan 2 buah dioda, selanjutnya 12 volt DC akan diratakan oleh kapasitor 3300 μF dan 10 μF yang dirangkai secara paralel.

III.3.9. Flowchart Simulator Ayunan Otomatis

Agar dapat melihat struktur jalannya program maka dibuat *flowchart* (diagram alur). *Flowchart* digunakan sebagai dasar acuan dalam membuat program. Struktur program akan lebih mudah dibuat/didesain. Selain itu juga jika terdapat kesalahan akan lebih mudah untuk mendeteksi letak kesalahannya serta untuk lebih memudahkan dalam menambahkan instruksi-instruksi baru pada program jika nantinya terjadi pengembangan pada struktur programnya.

Prinsip kerja alat ini diawali dengan dihidupkan saklar *power* secara manual yang akan mengalirkan catu daya dari adaptor, kemudian mikrokontroler akan aktif. Jika mikrokontroler mendeteksi adanya beban, yang ditandai dengan Switch sensor beban bernilai *High*, maka motor DC, sensor suara dan sensor air aktif, yang dalam artian menunggu masukan pendeteksi. Dan sebaliknya bila switch sensor beban bernilai *low*, maka motor DC, sensor suara dan sensor air tidak aktif.

Bila mikrokontroler mendeteksi adanya masukan dari sensor air yang bernilai lebih besar atau sama dengan 90 (*high*), maka motor DC dimatikan yang kemudian alarm dihidupkan untuk menandakan bayi buang air seni. Kemudian apakah bayi ada atau diangkat, pada umumnya saat bayi buang air seni, orang tua akan mengangkatnya untuk mengganti pakaian, yang berarti switch sensor akan bernilai *low*. Bila sensor suara bernilai lebih besar atau sama dengan 60 (*high*) selama 5 detik, motor DC hidup dan alarm akan hidup yang menandakan bayi menangis. Berikut ini adalah *Flowchart* yang dapat dilihat pada Gambar III.11.



Gambar III.11. Flowchart Simulator Ayunan Otomatis

