

BAB III

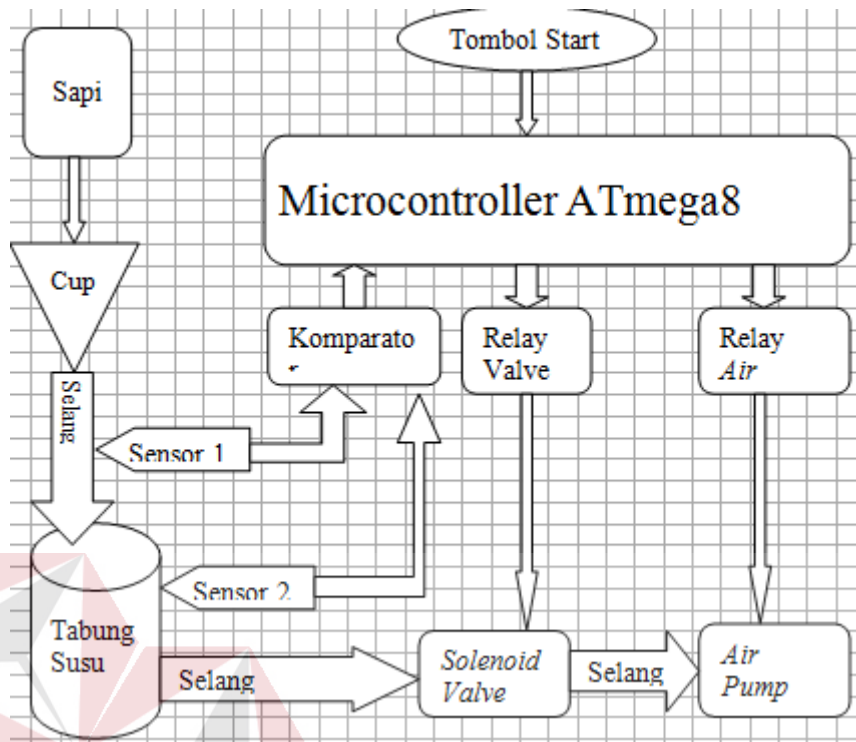
METODE PENELITIAN

Metodologi penelitian yang digunakan dalam perancangan sistem ini antara lain studi kepustakaan, meninjau tempat pemerahan susu sapi untuk mendapatkan dan mengumpulkan informasi berupa data-data literatur dan masing-masing komponen pembuatan alat ini, serta wawancara secara lisan dan informasi baik dari internet dan konsep-konsep teoritis dari buku penunjang yang berkaitan dengan penelitian.

Dari data-data yang diperoleh, selanjutnya dilakukan sebuah perancangan sistem yang terdiri dari perancangan dan pembuatan perangkat keras (*Hardware*), setelah desain *hardware* selesai dilakukan juga proses perancangan dan pembuatan perangkat lunak (*Software*) yang nantinya digunakan sebagai percobaan pada *hardware* maupun pada komputer.

Pada bagian perancangan perangkat keras dijelaskan berbagai macam tentang beberapa komponen yang digunakan untuk membangun *interface* alat ini khususnya desain mekanik mesin pemerah susu elektrik, *minimum sistem microcontroller* ATmega8, rangkaian *Relay*, rangkaian sensor photodiode, sensor LED. Sedangkan untuk menunjang komponen yang digunakan penulis menggunakan *software* CodeVisionAVR sebagai perancangan perangkat lunak.

Untuk pembuatan mesin pemerah susu Elektrik digunakan blok diagram secara keseluruhan seperti gambar 3.1.



Gambar 3.1. Blok Diagram Keseluruhan Sistem

Dari blok diagram pada gambar 3.1 ambung sapi dimasukkan ke dalam cup, udara dalam cup dibuang agar dapat merekat pada ambung. Selanjutnya *microcontroller* ATmega8 memberikan *input* pada *AirPump* untuk mengkosongkan udara didalam tabung agar proses pemerahan dapat dilakukan. Setelah udara dalam tabung telah habis maka otomatis ambung akan tersedot layaknya diperah oleh tangan manusia, sehingga cairan susu pada ambung akan keluar melewati selang sebelum menuju ke dalam tabung. Sensor 1 untuk mendeteksi aliran susu yang melewati selang apakah masih ada susu yang melewati selang lagi atau tidak, jika dalam waktu 10 detik tidak ada lagi susu yang melewati selang maka sensor 1 akan memberikan *input*. *Microcontroller* merespon *input* tersebut bahwa sudah tidak ada aliran susu yang melewati selang dan menonaktifkan valve. Sedangkan untuk sensor 2 untuk mendeteksi apakah

susu dalam tabung telah terpenuhi atau tidak, jika sudah terpenuhi maka sensor 2 akan memberikan *input* ke *microcontroller* untuk menonaktifkan valve dan *AirPump*.

3.1 Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras akan dibahas bagaimana komponen-komponen elektronika yang terhubung pada *hardware* dengan *microcontroller* agar elektronika pendukung dapat bekerja sesuai dengan sistem yang diharapkan.

3.1.1 Rangkaian *Microcontroller*

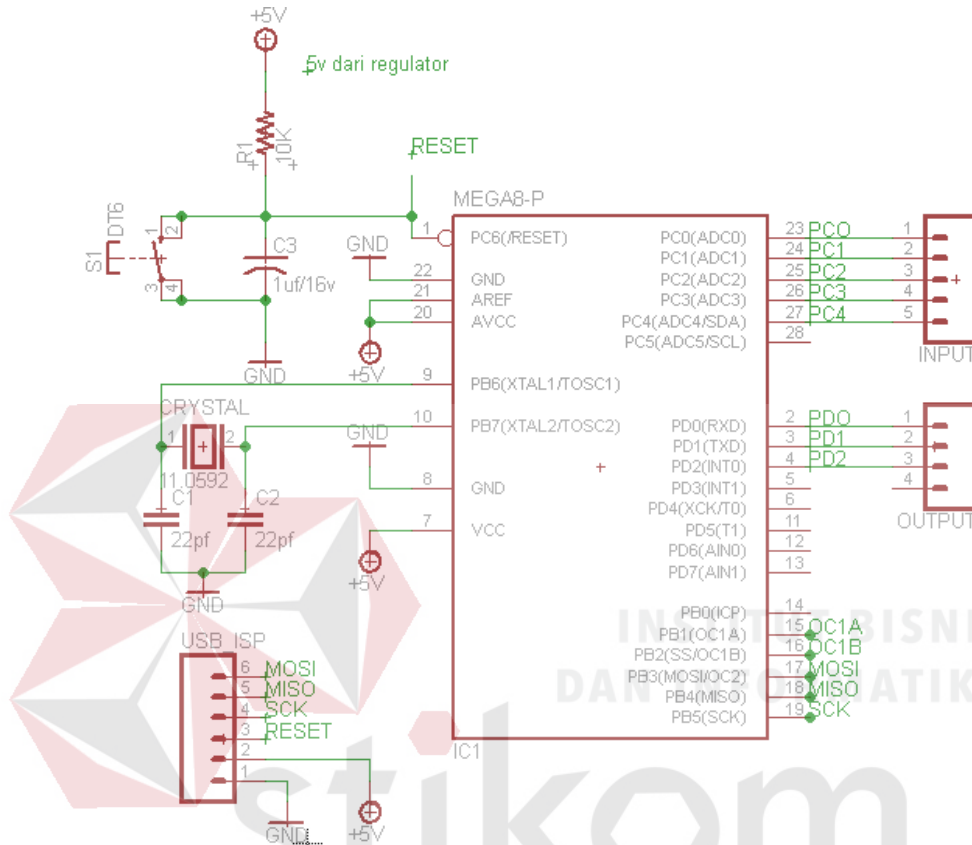
Pada penelitian ini dibuat piranti pengendali menggunakan *microcontroller* keluaran AVR, yaitu ATmega8. Untuk mengaktifkan atau menjalankan *microcontroller* ini diperlukan rangkaian *minimum system*. Rangkaian *minimum system* tersebut terdiri rangkaian *reset*. Rangkaian *oscillator*, rangkaian *Relay*, rangkaian *power supply* dan rangkaian *microcontroller*.

Dalam perancangan perangkat keras *minimum system* ATmega8 terdapat beberapa rangkaian pendukung yaitu rangkaian *reset* dan rangkaian *oscillator*. Pada rangkaian *reset* menggunakan *manual reset*. Pada rangkaian *oscillator* menggunakan komponen Kristal 11.0592MHZ sebagai *clk (clock)*.

A. Rangkaian *Minimum System Microcontroller*

Untuk rangkaian *microcontroller* dibutuhkan sebuah rangkaian *minimum system* agar *microcontroller* tersebut dapat bekerja dengan baik. Rangkaian *minimum system* terdiri dari rangkaian *reset* dan rangkaian *oscillator*.

Pada pin VCC diberi masukan tegangan operasi berkisar antara 4,5 volt sampai dengan 5,5 volt. Pin RST mendapat masukan dari manual *reset*. Rangkaian *minimum system* dapat dilihat pada gambar 3.2 berikut



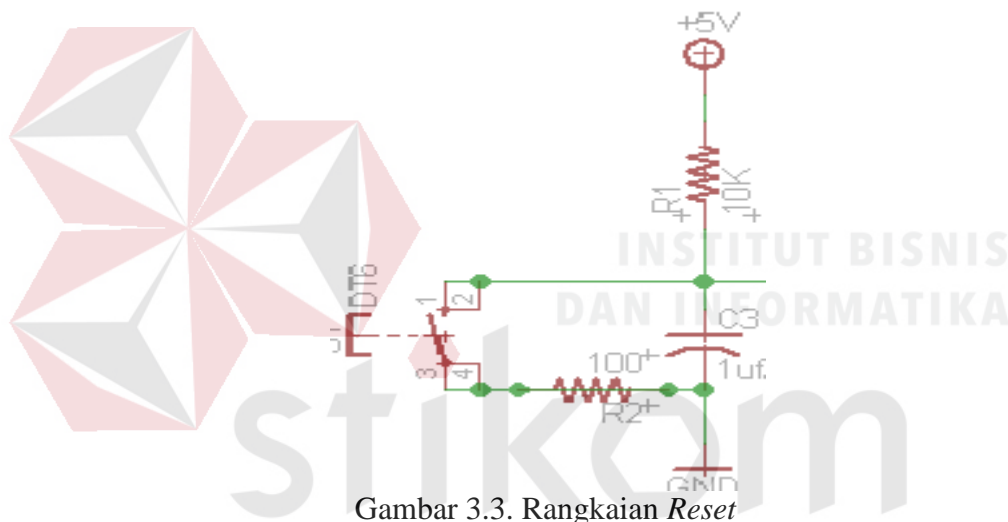
Gambar 3.2. Rangkaian *Minimum System*

Berdasarkan gambar 3.2 di atas pin vcc diberi tegangan operasi yang besarnya berkisar 5 volt sampai dengan 6,5 volt. *Port C*, digunakan sebagai *input* dari sensor *photodiode* sedangkan *port D* digunakan sebagai *output* dari *microcontroller* yang member perintah pada *valve* dan *air pump*. Pin XTAL1 dan XTAL2 merupakan pin *oscillator* bagi *microcontroller* ATmega8. Pin XTAL1 berfungsi sebagai *input* dan XTAL2 sebagai *output oscillator*. *Oscillator* ini bias berasal dari Kristal atau dari keramik *Resonator*. Seperti yang sudah terlihat di atas, pin XTAL1 dan XTAL2 dihubungkan dengan komponen XTAL sebesar

110592 MHz. pada proyek Tugas Akhir ini dibuat rangkaian *oscillator internal* yang terbuat dari Kristal. Nilai C1 dan C2 masing-masing 22pF.

B. Rangkaian Reset

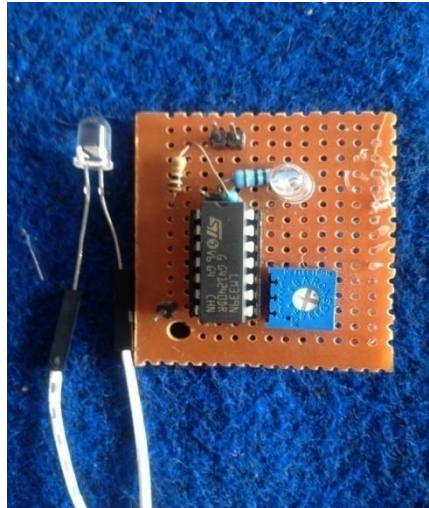
Reset pada *microcontroller* ATmega8 terjadi dengan adanya logika *High* “1” selama dua *cycle* pada kaki RST pada *microcontroller* ATmega8. Setelah kondisi pin RST kembali *low*, maka *microcontroller* akan menjalankan program dari alamat 0000H. dalam hal ini *reset* yang digunakan adalah manual *reset*. Rangkaian *reset* dapat dilihat pada gambar 3.3.



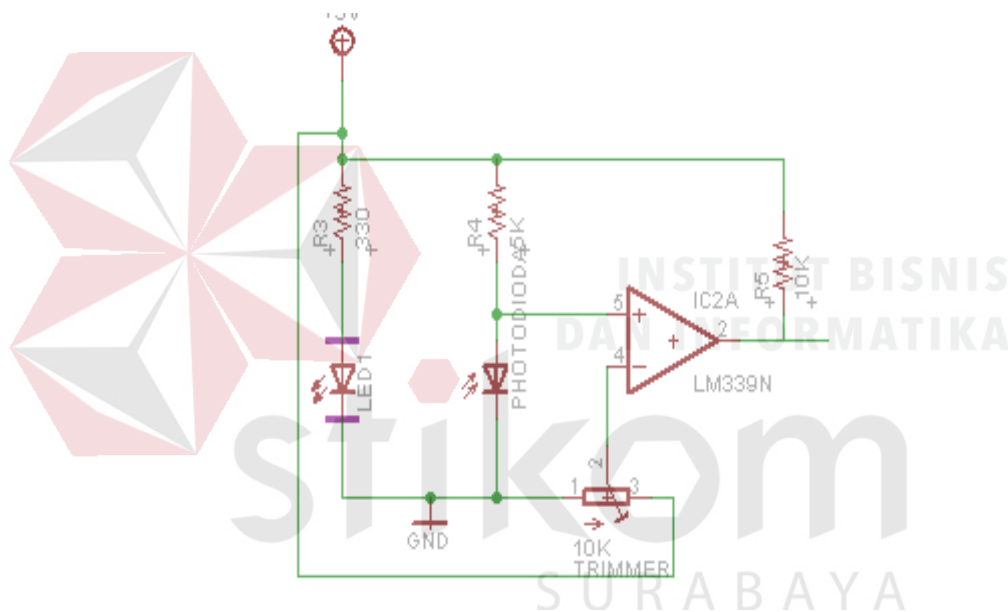
Gambar 3.3. Rangkaian *Reset*

C. Rangkaian Sensor Photodiode

satu buah komparator terdiri dari 2 *input*, yaitu *Vin* (*input* masukan dari sensor) dan *Vref* (tegangan referensi). Pada dasarnya, jika tegangan *Vin* lebih besar dari *Vref*, maka V0 akan mengeluarkan logika 1 berarti 5 volt. Sebaliknya, jika tegangan *Vin* lebih kecil daripada tegangan *Vref*, maka *output* V0 akan mengeluarkan logika 0 yang berarti 0 volt. *Output* pada komparator diberi *pull-up* agar nilai V0 menjadi logika 1 yang berate 5volt. Berikut adalah gambar 3.4.



Gambar 3.4 Rangkaian Sensor Photodiode

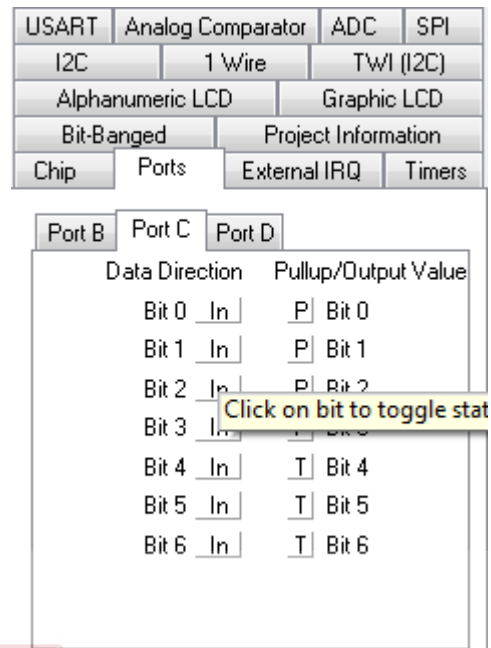


Gambar 3.5 Skematik Sensor Photodiode

D. Perancangan *Interface I/O*

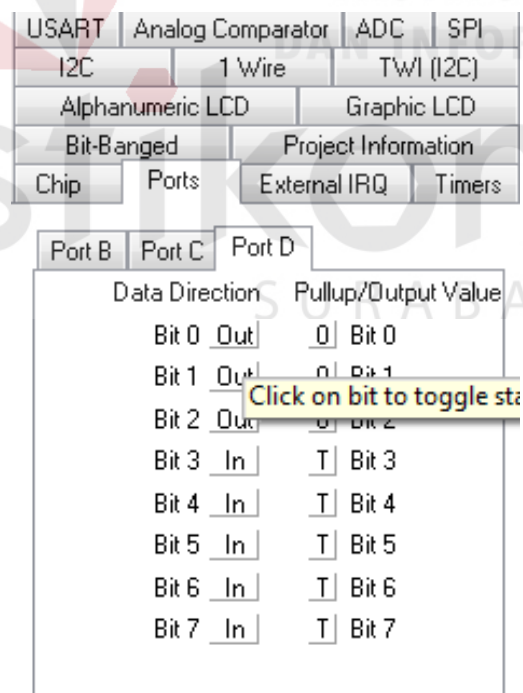
Rangkaian *I/O* dari *microcontroller* mempunyai control direksi yang tiap bitnya dapat dikonfigurasi secara individual, maka dalam perancangan *I/O* yang digunakan ada yang berupa operasi *port* ada pula yang dikonfigurasi tiap bitnya. Berikut ini akan diberikan konfigurasi dari *I/O microcontroller* tiap bit yang ada masing-masing *port* yang terdapat pada *microcontroller*.

1. *Port C*, digunakan untuk *input* sensor Photodiode dan tombol Start.



Gambar 3.6 PIN C Input

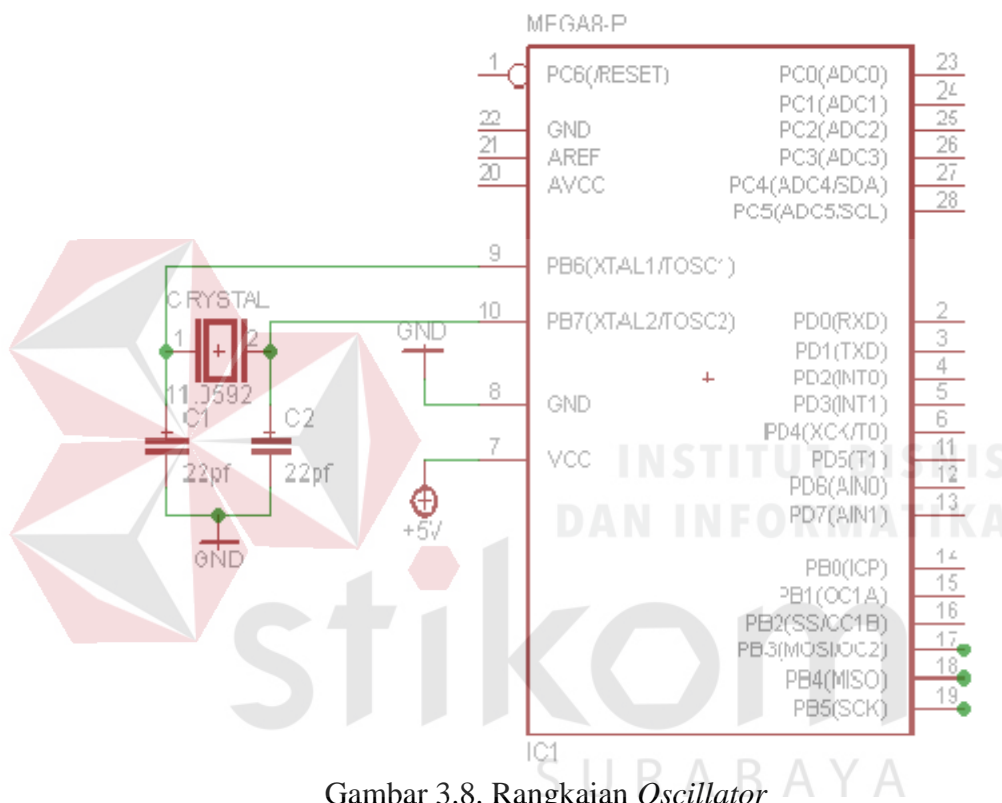
2. *Port D*, digunakan untuk *output* Relay, air pump.



Gambar 3.7 PORT D Output.

E. Rangkaian Oscillator

Pin XTAL1 dan XTAL2 merupakan *pin oscillator* bagi *microcontroller* ATmega8. *Oscillator* ini berasal dari Kristal atau keramik *resonator*. Rangkaian *oscillator* dapat dilihat pada gambar 3.8.



Gambar 3.8. Rangkaian *Oscillator*

Pada proyek akhir ini dibuat rangkaian *oscillator internal* yang terbuat dari Kristal. Nilai C1 dan C2 masing-masing 22pF.

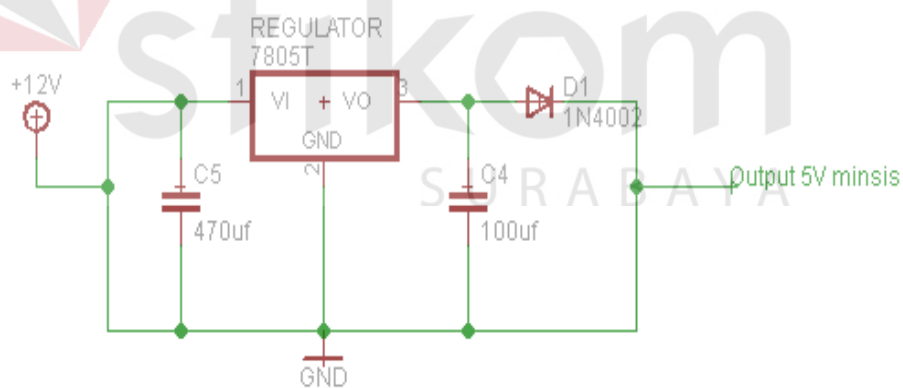
3.1.2 Rangkaian Pendukung

Pada proyek ini menggunakan beberapa komponen pendukung untuk membantu kinerja *input* dan *output*. Komponen pendukung ini memiliki rangkaian dan penjelasan sebagai berikut.

Regulator

Pada bagian regulator untuk proyek ini menggunakan 2 komponen regulator yaitu regulator untuk *output* 5volt dimana dalam proyek ini menggunakan IC 7805 yang bertujuan untuk menstabilkan tegangan dengan *output* 5volt.

Rangkaian ini berfungsi untuk catu daya. Catu daya merupakan pendukung utama bekerjanya suatu sistem. Catu daya yang biasanya digunakan untuk menyuplai tegangan sebesar 5volt adalah catu daya DC yang memiliki keluaran 5volt. Catu daya ini digunakan untuk mensuplay tegangan sebesar 5volt. IC7805 digunakan untuk menstabilkan tegangan searah. Kapasitor digunakan untuk mengurangi tegangan kejut saat pertama kali saklar catu daya dihidupkan. Sehingga keluaran IC regulator 7805 stabil sebesar 5volt DC. Rangkaian regulator terlihat pada gambar 3.9.



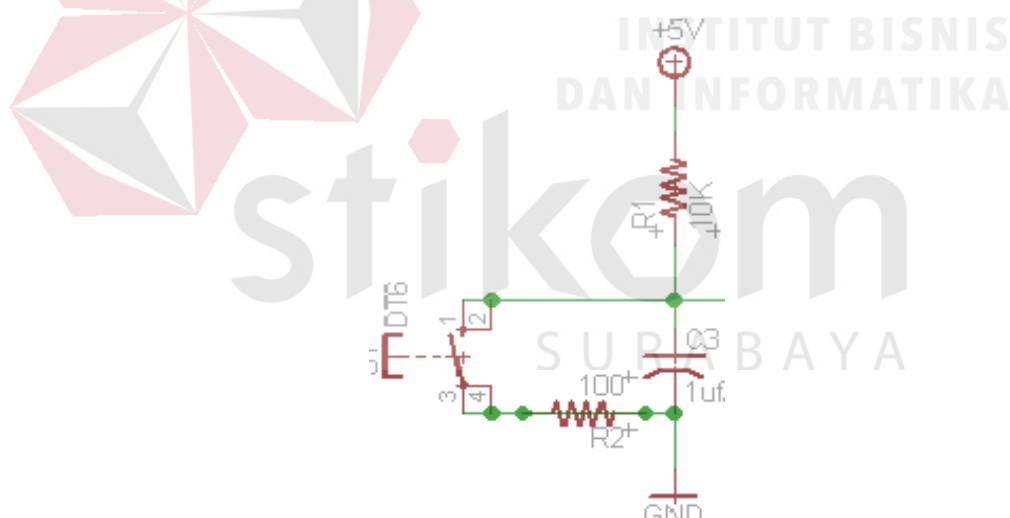
Gambar 3.9. Rangkaian Regulator

3.1.3 Rangkaian Input

pada proyek ini terdapat rangkaian masukan yaitu sensor photodiode dan *push button switch* untuk menunjang kinerja alat dalam proyek ini. Dimana penjelasan rangkaian sebagai berikut.

Push Button

Push button disini memiliki fungsi untuk memulai alat agar dapat bekerja dan berjalan sesuai yang diinginkan. Prinsip kerja *push button* adalah memiliki fungsi sama seperti saklar *push-on* yaitu akan terhubung pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus saat katup tidak ditekan. Berikut adalah rangkaian tombol 3.10.



Gambar 3.10. Rangkaian Tombol Start

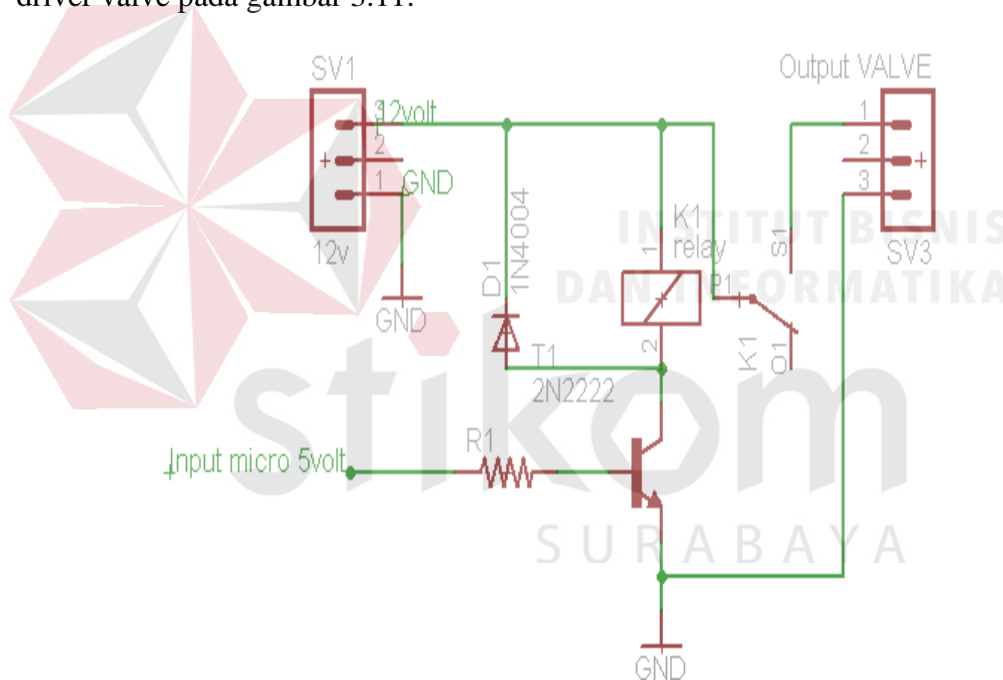
3.1.4 Rangkaian Output

A. Driver Valve

Dalam rangkaian valve, terdapat 1 buah relay 12v dan transistor 2n222. Relay adalah sebuah komponen yang terdiri dari kumparan berinti besi yang akan

menghasilkan elektromagnet ketika kumparannya dialiri oleh arus listrik. Elektromagnet ini kemudian menarik mekanisme kontak yang akan menghubungkan kontak (*Normally open*) dan membuka kontak (*Normally Closed*).

Prinsip kerja dari transistor NPN adalah, arus akan mengalir dari kolektor ke emitter jika basisnya dihubungkan ke *Ground*. Arus yang mengalir dari basis harus lebih kecil daripada arus yang mengalir dari kolektor ke emitter. Fungsinya adalah sebagai saklar elektronik yang akan mengalirkan arus listrik. Berikut rangkaian driver valve pada gambar 3.11.

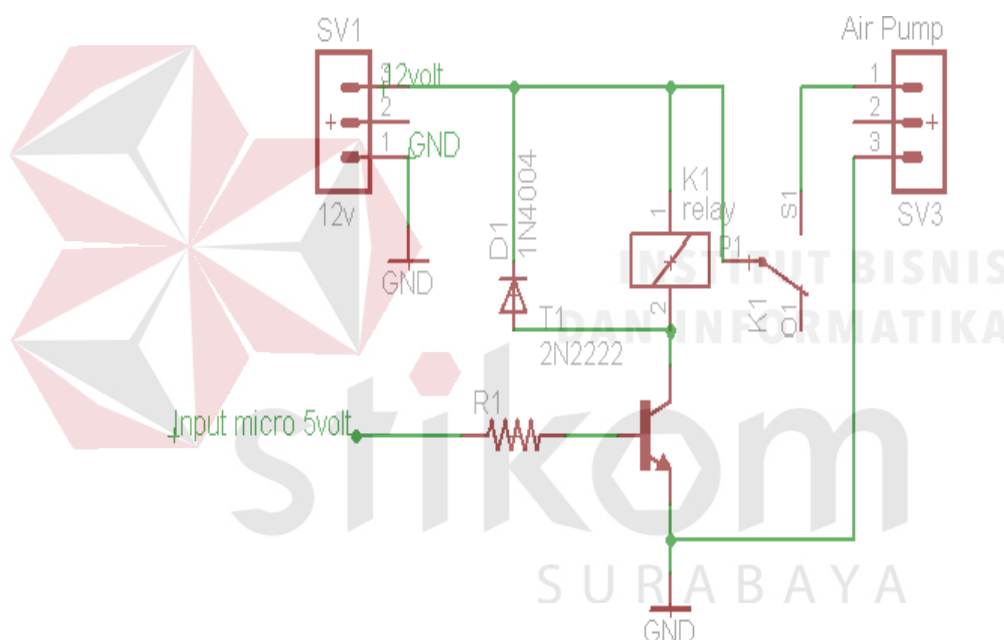


Gambar 3.11. Rangkaian Driver Valve

Pada rangkaian driver valve digunakan sebagai pengendali valve *PORTD.0* dan *PORTD.1* sebagai *output* dari *microcontroller*, apabila *PORTD.0* dan *PORTD.1* memberikan *input* logika *low* maka relay akan memutus aliran tegangan yang menghubungkan valve dengan sumber tegangan. Dan apabila *inputan* logika *high* maka relay akan menyambungkan aliran tegangan yang menghubungkan dengan sumber tegangan. Kemudian valve akan terbuka.

B. Driver Air Pump

Dalam rangkaian *Air Pump*, terdapat 1 buah relay 12v dan transistor 2n2222. Relay adalah sebuah komponen yang terdiri dari kumparan berinti besi yang akan menghasilkan elektromagnet ketika kumparannya dialiri oleh arus listrik. Elektromagnet ini kemudian menarik mekanisme kontak yang akan menghubungkan kontak (*Normally open*) dan membuka kontak (*Normally Closed*). Berikut gambar 3.12 rangkaian Driver *Air Pump*.

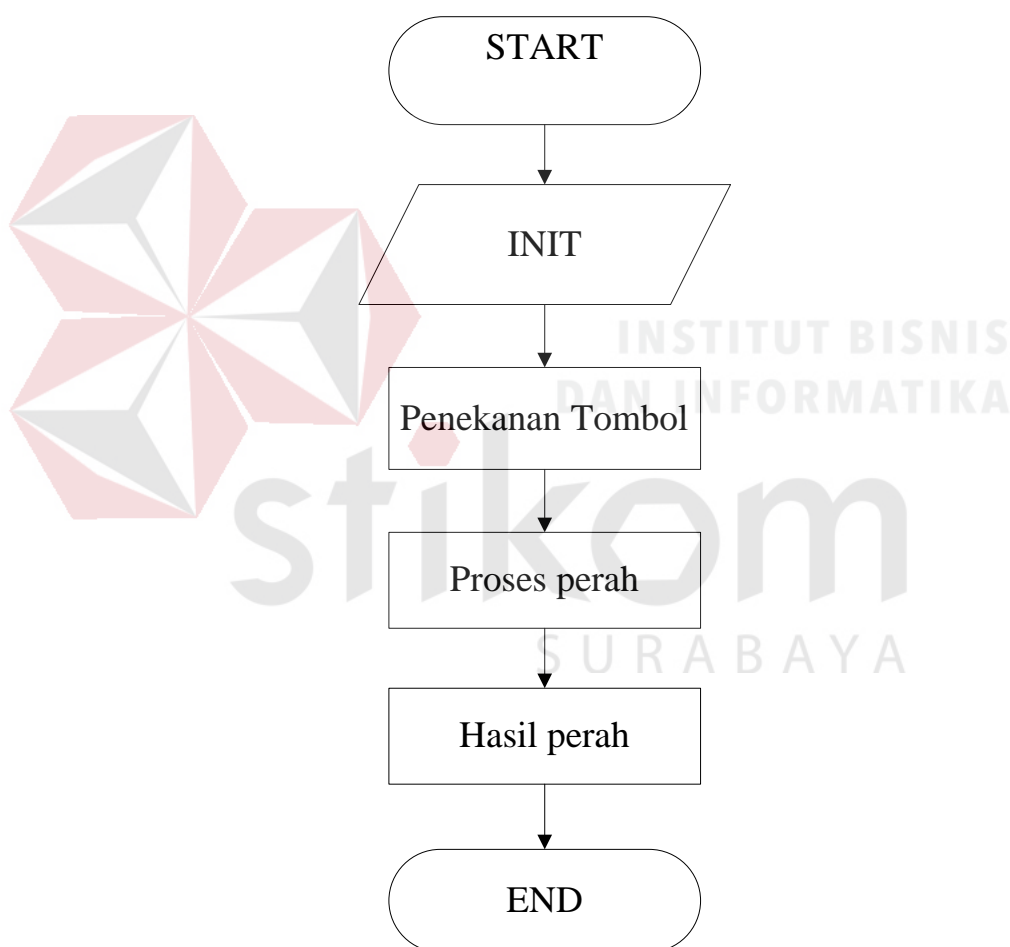


Gambar 3.12 Rangkaian Driver *Air Pump*

Pada rangkaian driver *air pump* digunakan sebagai pengendali *air pump* PORTD.2 sebagai *output* dari *microcontroller*, apabila PORTD.2 memberikan *air pump* logika *low* maka relay akan memutus aliran tegangan yang menghubungkan *air pump* dengan sumber tegangan. Dan apabila *inputan* logika *high* maka relay akan menyambungkan aliran tegangan yang menghubungkan dengan sumber tegangan. Kemudian *air pump* bekerja.

3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak bertujuan untuk memperoleh hasil dan waktu dalam proses pemerahan susu. Perancangan perangkat lunak terbagi dalam beberapa *device* sistem antara lain, program sensor, timer, counter dan program untuk penggerak komponen-komponen pembantu lainnya seperti, relay untuk valve dan *air pump*. Diagram alir perangkat lunak secara umum dapat dilihat pada gambar 3.13 berikut.

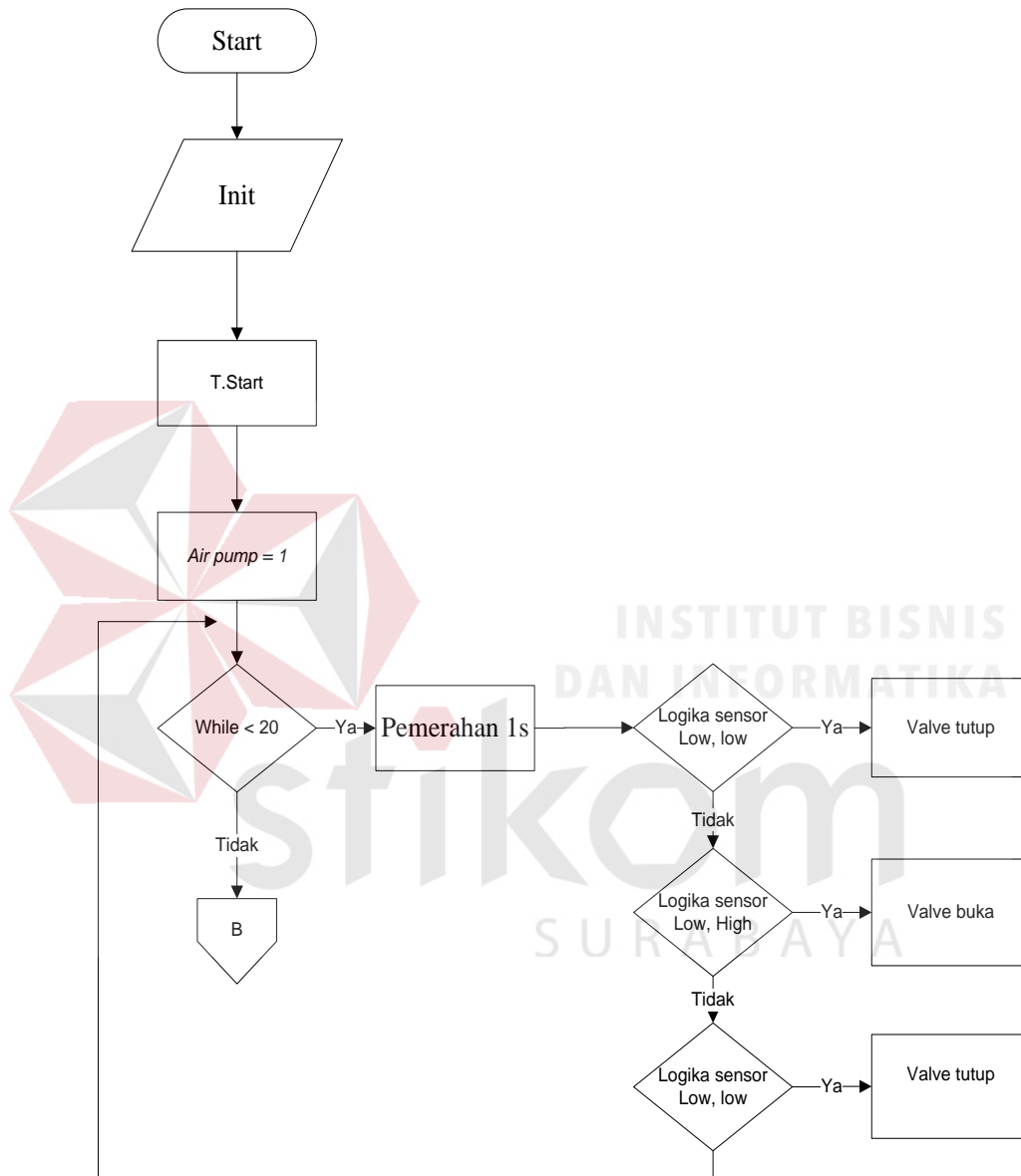


Gambar 3.13. Diagram Alir Program Secara Umum

Gambar diatas merupakan aliran proses saat start yaitu penekanan tombol start untuk menjalankan alat pemerah susu melalui *microcontroller*. Saat penekanan terjadi *microcontroller* akan mendeteksi penekanan tombol.

3.2.1 Program *Microcontroller*

A. Program saat penekanan tombol.



Gambar 3.14. Diagram Alir Program Interval 1s.

Pada gambar 3.14 adalah diaman saat pemerahan awal dengan jeda perah istirahat sekitar 1s, yang berguna untuk merangsang susu agar mudah diperah dalam jangka waktu yang lebih lama, sehingga dalam jangka waktu yang lama alat ini dapat berganti mode dengan perah istirahat yang sedikit melambat dan

memberi kenyamanan pada hewan yang sedang diperah. Berikut potongan program awal pemerahan.

```

while(PINC.5 == 1);
while(PINC.5 == 0)
{
    delay_ms(10);
    if(PINC.5 == 0)
    {
        TCCR1A = TCCR1B = 0x0D;
        valve1 = 1;
    }
}

flag1 = flag2 = 0;
while(ulang < 20)
{
    motor = 1;
    delay_ms(1000);
    motor = 0;
    delay_ms(1000);

    if(count1 <= 10 && sensor_akanan == 0 && sensor_bkanan == 0)
    {
        valve1 = 1;
        flag1 = 0;
    }
    else if(count1 <= 10 && sensor_akanan == 0 && sensor_bkanan == 1)
    {
        valve1 = 0;
        TCCR1A = 0;
        flag1 = 1;
    }
    else if(count1 > 10 && sensor_akanan == 0 && sensor_bkanan == 0)
    {
        valve1 = 0;
        flag1 = 1;
        TCCR1A = 0x00;
    }

    //=====sensor kiri=====
    if((count2 < 10 && sensor_akiri == 1) && sensor_bkiri == 0)
    {
        valve2 = 0;
        count2 = 0;
    }
    else if((count2 < 10 && sensor_akiri == 1) && sensor_bkiri == 1)
    {
        valve2 = 1;
        flag2 = 1;
    }
    else if((count2 < 10 && sensor_akiri == 0) && sensor_bkiri == 1)
    {
        valve2 = 1;
        flag2 = 1;
    }
}

```

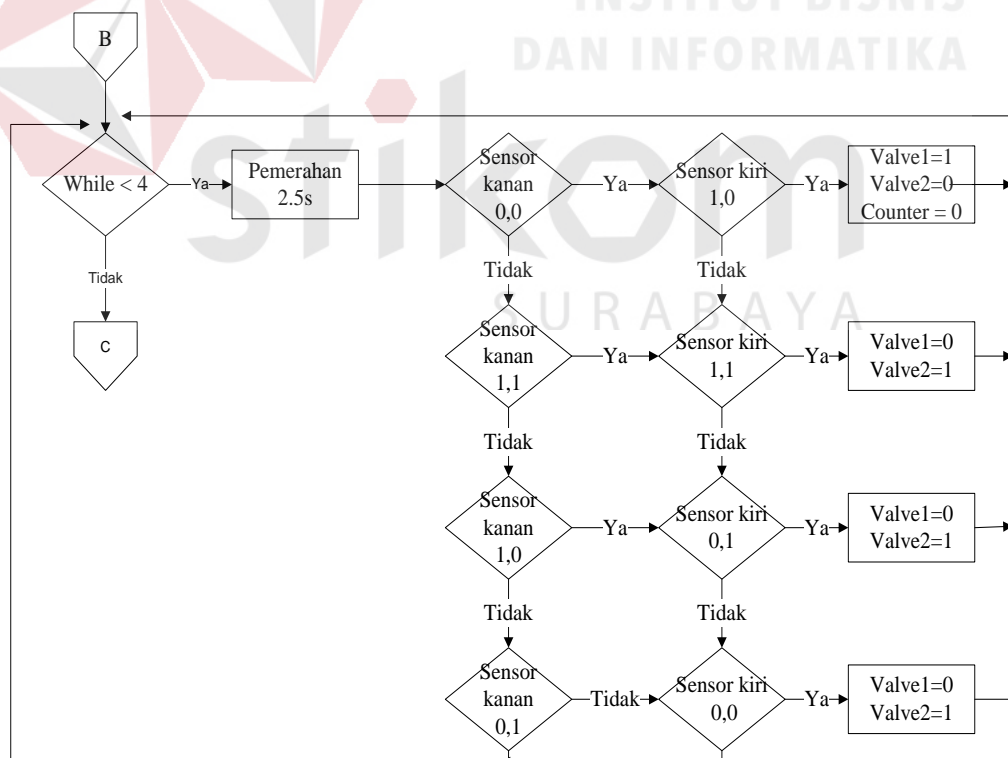
```

}
else if((count2 > 10 && sensor_akiri == 0) && sensor_bkiri == 0)
{
    valve2 = 1;
    flag2 = 1;
}
flag2 = 1;
if(flag1 == 1 && flag2 == 1)
{
    motor = 0;
    flag1 = flag2 = 0;
    TCCR1A = TCCR1B = 0x00;
    while(1);
}
else
{
    motor = 1;
}
ulang++;
}

```

B. Program perah 2,5s.

Proses ini berlangsung setelah proses awal, proses ini lebih relax dibanding proses awal, berikut gambar 3.15



Gambar 3.15. Pemerahan Dengan Interval 2,5s.

Pada proses pertengahan dengan jeda interval perah istirahat sekitar 2,5s, sehingga pemerahannya menjadi semakin relax daripada saat proses awal dengan jeda 1s. dan berikut adalah potongan program saat pemerahan ini.

```

while (1)
{
// Place your code here
  ulang = 0;
  while(ulang < 4)
  {
    motor = 1;
    delay_ms(2500);
    motor = 0;
    delay_ms(1000);

    if(count1 < 10 && sensor_akanan == 1 && sensor_bkanan == 1)
    {
      valve1 = 1;
      count1 = 0;
      flag1 = 0;
    }
    else if(count1 < 10 && sensor_akanan == 1 && sensor_bkanan == 0)
    {
      valve1 = 0;
      TCCR1A = 0x00;
      flag1 = 1;
    }
    else if(count1 < 10 && sensor_akanan == 0 && sensor_bkanan == 0)
    {
      valve1 = 0;
      TCCR1A = 0x00;
      flag1 = 1;
    }
    else if(count1 > 10 && sensor_akanan == 0 && sensor_bkanan == 1)
    {
      valve1 = 0;
      TCCR1A = 0x00;
      flag1 = 1;
    }

    //=====sensor =====
    if((count2 < 10 && sensor_akiri == 1) && sensor_bkiri == 0)
    {
      valve2 = 0;
      count2 = 0;
      flag2 = 0;
    }
    else if((count2 < 10 && sensor_akiri == 1) && sensor_bkiri == 1)
    {

```

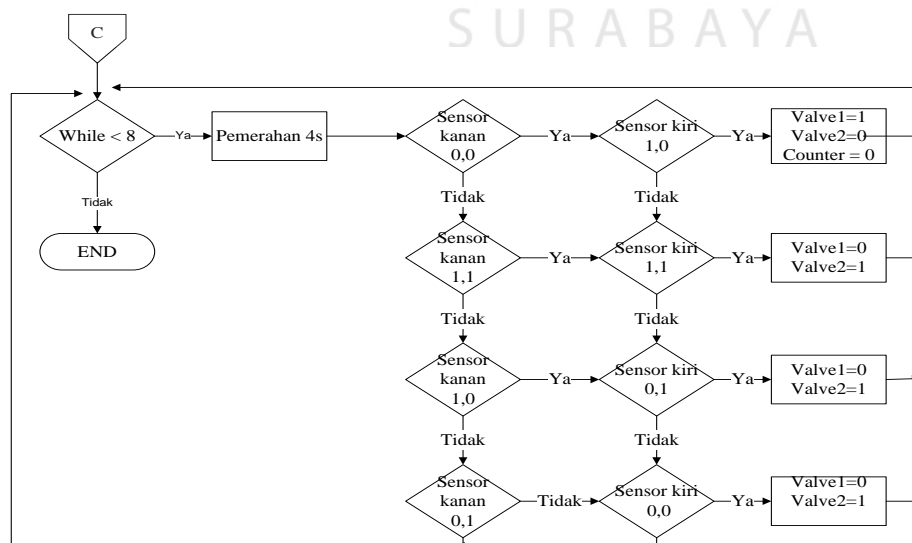
```

valve2 = 1;
  flag2 = 1;
}
else if((count2 < 10 && sensor_akiri == 0) && sensor_bkiri == 1)
{
  valve2 = 1;
  flag2 = 1;
}
else if((count2 > 10 && sensor_akiri == 0) && sensor_bkiri == 0)
{
  valve2 = 1;
  flag2 = 1;
}
flag2 = 1;
if(flag1 == 1 && flag2 == 1)
{
  motor = 0;
  flag1 = flag2 = 0;
  TCCR1A = TCCR1B = 0x00;
  while(1);
}
else
{
  motor = 1;
}
}
ulang++;
}

```

C. Program perah 4s.

Proses ini berlangsung setelah proses interval 2,5s, proses ini memerah lebih lama dibanding istirahat. Karena delay istirahat hanya 1s dan proses ini yang akan membuat pemerahan menjadi lebih santai. Berikut gambar 3.16.



Gambar 3.16. Pemerahan Dengan Interval 4s.

Berikut adalah potongan program dari proses pemerahan 4s.

```

ulang = 0;
while(ulang < 8)
{
    motor = 1;
    delay_ms(4000);
    motor = 0;
    delay_ms(1000);

    if(count1 < 10 && sensor_akanan == 1 && sensor_bkanan == 1)
    {

        valve1 = 1;
        count1 = 0;
        flag1 = 0;

    }
    else if(count1 < 10 && sensor_akanan == 1 && sensor_bkanan == 0)
    {

        valve1 = 0;
        TCCR1A = 0x00;
        flag1 = 1;

    }
    else if(count1 < 10 && sensor_akanan == 0 && sensor_bkanan == 0)
    {

        valve1 = 0;
        TCCR1A = 0x00;
        flag1 = 1;

    }
    else if(count1 > 10 && sensor_akanan == 0 && sensor_bkanan == 1)
    {

        valve1 = 0;
        TCCR1A = 0x00;
        flag1 = 1;

    }

    //=====sensor kiri=====
    if((count2 < 10 && sensor_akiri == 1) && sensor_bkiri == 0)
    {
        valve2 = 0;
        count2 = 0;
        flag2 = 0;
    }
    else if((count2 < 10 && sensor_akiri == 1) && sensor_bkiri == 1)
    {
        valve2 = 1;
        flag2 = 1;
    }
    else if((count2 < 10 && sensor_akiri == 0) && sensor_bkiri == 1)
    {
        valve2 = 1;
    }

```

```

flag2 = 1;
}
else if((count2 > 10 && sensor_akiri == 0) && sensor_bkiri == 0)
{
valve2 = 1;
flag2 = 1;
}
flag2 = 1;
if(flag1 == 1 && flag2 == 1)
{
motor = 0;
flag1 = flag2 = 0;
TCCR1A = TCCR1B = 0x00;
while(1);
}
else
{
motor = 1;
}
ulang++;
}
}
}

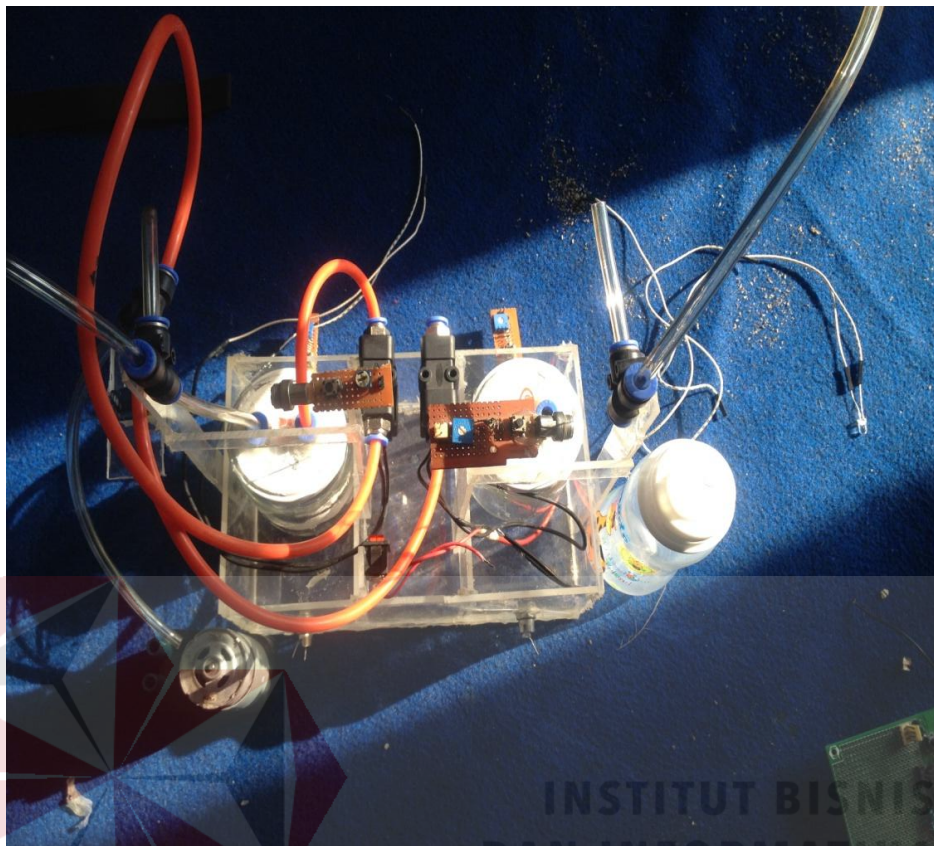
```

3.3 Perancangan Mekanik

Perancangan mekanik menggunakan tabung suntik sebagai tempat untuk menyedot dan *air pump* sebagai penyedot. Disini valve berguna sebagai pemutus dan penyambung aliran angin, sedangkan botol sebagai wadah dari susu yang akan diperah, dapat dilihat pada gambar 3.17.



Gambar 3.17. Alat Tampak Depan



Gambar 3.18. Alat Tampak Atas.

3.4 Metode pengujian dan Evaluasi sistem

Penelitian ini pengujian akan dilakukan pada alat pemerah susu elektrik. Mulai dari menghubungkan tegangan 12volt ke *microcontroller*, *relay* dan tegangan 5volt untuk sensor-sensornya. Pengujian paling akhir adalah pengujian sistem secara keseluruhan, sehingga alat dapat berjalan sesuai dengan apa yang kita harapkan.

3.4.1 Pengujian dan Evaluasi Sensor

Pengujian ini dilaukan untuk mengetahui apakah sensor-sensor dapat bekerja dengan baik ketika member *inputan* kepada *microcontroller*. Jika berhasil

maka *microcontroller* akan merespon *input* tersebut dan memberikan *output* kepada valve dan *airpump*.

3.4.2 Evaluasi Keseluruhan Sistem

Mengevaluasi apakah seluruh sistem dapat bekerja dengan baik dan tidak ada kendala, sehingga sistem bekerja sesuai harapan.

