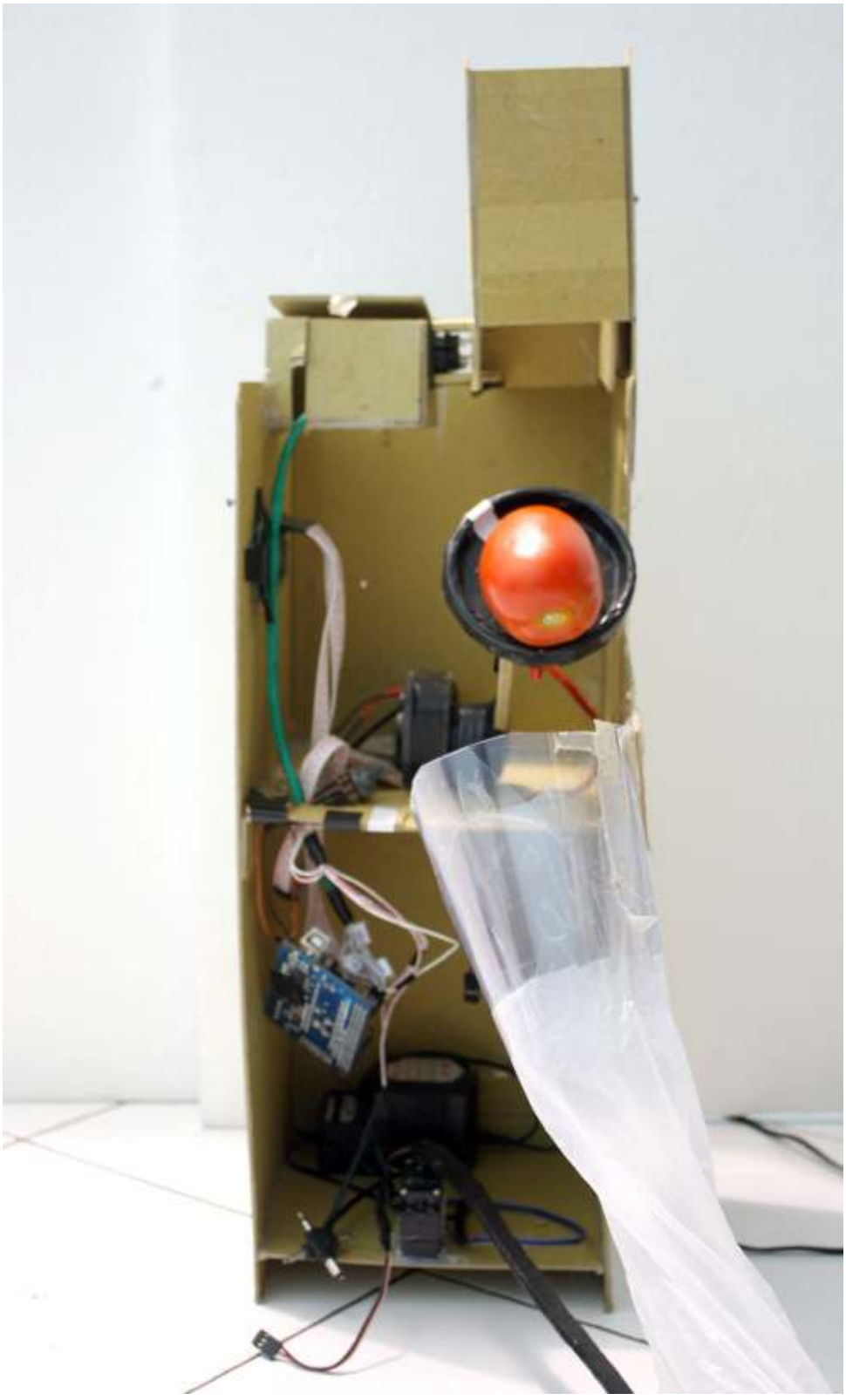
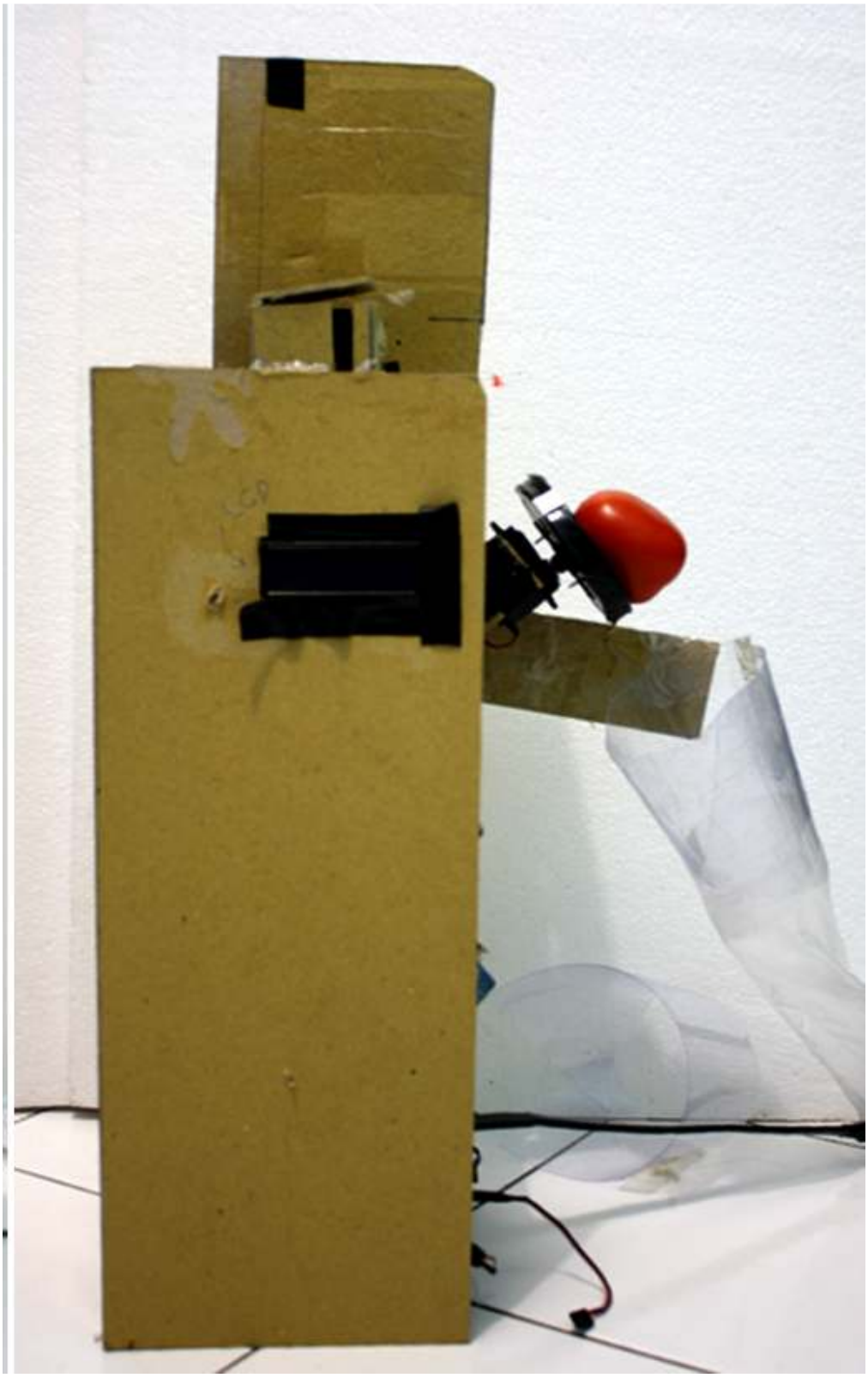
LAMPIRAN A
FOTO ALAT





LAMPIRAN B
LIST PROGRAM

```
/*****************************************************
This program was produced by the
CodeWizardAVR V2.04.4a Advanced
Automatic Program Generator
© Copyright 1998-2009 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project : ta
Version : 1.0
Date    : 27/05/2011
Author  : adps
Company : situmorang
Comments:


Chip type               : ATmega16
Program type            : Application
AVR Core Clock frequency: 11,059200 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 256
*****************************************************/

#include <mega16.h>
#include <stdio.h>
#include <delay.h>

#define redref (20383/255)
#define greenref (21000/255)
#define blueref (28865/255)

// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

// Declare your global variables here

int i,j;
unsigned char lcd_buffer[33];
unsigned int count, red, green,
blue,a=0,b=0,c=0,r1,g1,b1,r2,g2,b2,r3,g3,b3,rt,gt,bt;
float periode, frekuensi;
```

```
void red_filter()
{
    PORTA.5=0;
    PORTA.6=0;
    for(a=0; a<=5; a++)
    {
    while(PINB.1 == 0){}
    while(PINB.1 == 1){}
    }

    while(PINB.1 == 0)
    {
     count++;
    }

    while(PINB.1 == 1)
    {
     count++;
    }

    periode = ((float)count*0.000001440092165898617511 5207373271889);
    frekuensi = ((float)1/periode);
    red = (frekuensi/redref);

    count=0;

}

void green_filter()
{
    PORTA.5=1;
    PORTA.6=1;
    for(b=0; b<=5; b++)
    {
    while(PINB.1 == 0){}
    while(PINB.1 == 1){}
    }

    while(PINB.1 == 0)
    {
     count++;
    }

    while(PINB.1 == 1)
```

```
    {
     count++;
    }

    periode = ((float)count*0.000001440092165898617511520737327188 9);
    frekuensi = ((float)1/periode);
    green = (frekuensi/greenref);

    count=0;

}

void blue_filter()
{
    PORTA.5=0;
    PORTA.6=1;
    for(c=0; c<=5; c++)
    {
    while(PINB.1 == 0){}
    while(PINB.1 == 1){}
    }

    while(PINB.1 == 0)
    {
     count++;
    }

    while(PINB.1 == 1)
    {
     count++;
    }

    periode = ((float)count*0.000001440092165898617511520737327188 9);
    frekuensi = ((float)1/periode);
    blue = (frekuensi/blueref);

    count=0;

}

void servo1()
{
for(i=0;i<200;i++)
{
PORTB.7=1;
delay_us(600);
```

```
PORTB.7=0;
delay_us(19400);
}
}

 void servo2()
{
for(i=0;i<200;i++)
{
PORTB.7=1;
delay_us(1200);
PORTB.7=0;
delay_us(18800);
}
}

void servo3()
{
for(i=0;i<200;i++)
{
PORTB.7=1;
delay_us(1800);
PORTB.7=0;
delay_us(18200);
}
}

 void mateng()
{
for(i=0;i<200;i++)
{
PORTB.6=1;
delay_us(800);
PORTB.6=0;
delay_us(19200);
}
}

 void setengah()
{
for(i=0;i<200;i++)
{
PORTB.6=1;
delay_us(1200);
PORTB.6=0;
delay_us(18800);
```

```
}
}

void mentah()
{
for(i=0;i<200;i++)
{
PORTB.6=1;
delay_us(1500);
PORTB.6=0;
delay_us(18500);
}
}

void turun()
{
OCR1AL=30;
PORTD.0=1;
PORTD.1=0;
DDRA.7=0;

}

void naik()
{
OCR1AL=30;
PORTD.0=0;
PORTD.1=1;
DDRA.7=1;
}

void mati()
{
OCR1AL=100;
PORTD.0=1;
PORTD.1=1;
DDRA.7=0;
}

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A 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
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=Out Func6=Out Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=0 State6=0 State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xC0;

// 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=Out Func4=Out Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=0 State4=0 State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x30;

// 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: 11059,200 kHz
// Mode: Ph. correct PWM top=00FFh
// OC1A output: Inverted
// OC1B output: Inverted
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xF1;
TCCR1B=0x01;
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: Timer2 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;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);
count=0;
PORTA=0b10011000;
```

```
while (1)
    {

If(PINA.0==0)
{
    delay_ms(500);

    red_filter();
    r1=red;
    delay_ms(50);
    green_filter();
    g1=green;
    delay_ms(50);
    blue_filter();
    b1=blue;
    delay_ms(50);
    lcd_gotoxy(0,0);
    lcd_clear();
    sprintf(lcd_buffer,"data sample 1-R:%d G:%d B:%d",red,green,blue);
    lcd_puts(lcd_buffer);
    a=0; b=0; c=0;

    delay_ms(10);
    servo2();
    delay_ms(100);


    red_filter();
    r2=red;
    delay_ms(50);
    green_filter();
    g2=green;
    delay_ms(50);
    blue_filter();
    b2=blue;
    delay_ms(50);
    lcd_gotoxy(0,0);
    lcd_clear();
    sprintf(lcd_buffer,"data sample 2-R:%d G:%d B:%d",red,green,blue);
    lcd_puts(lcd_buffer);
    a=0; b=0; c=0;

    delay_ms(10);
```

```
servo3();
delay_ms(100);

red_filter();
r3=red;
delay_ms(50);
green_filter();
g3=green;
delay_ms(50);
blue_filter();
b3=blue;
delay_ms(50);
lcd_gotoxy(0,0);
lcd_clear();
sprintf(lcd_buffer,"data sample 3-R:%d G:%d B:%d",red,green,blue);
lcd_puts(lcd_buffer);
a=0; b=0; c=0;

delay_ms(10);
servo1();
delay_ms(10);

rt=((r1+r2+r3)/3);                //rata rata data yg disampling
bt=((b1+b2+b3)/3);
gt=((g1+g2+g3)/3);

delay_ms(150);
lcd_gotoxy(0,0);
lcd_clear();
sprintf(lcd_buffer,"data rata rata-R:%d G:%d B:%d",rt,gt,bt);
lcd_puts(lcd_buffer);
delay_ms(1000);
a=0; b=0; c=0;


if(rt >=50&& rt<=30)    //proses pemilihan kematengan
{  lcd_gotoxy(0,0);
lcd_clear();
lcd_puts("mateng");
mateng();

}

 if(gt=>45 && (rt-gt)<=5)    //proses pemilihan kematengan
{  lcd_gotoxy(0,0);
lcd_clear();
```

```
    lcd_puts("mentah");
    mentah();
    }

    else
    {  lcd_gotoxy(0,0);
    lcd_clear();
    lcd_puts("setengah");
    setengah();
    };


    };

    j=3000;                         // motor DC
    j=j+135;
    naik();
    delay_ms(3000);
    turun();
    delay_ms(j);
    mati();
    delay_ms(3000);

      }

   else
   {
lcd_gotoxy(0,0);
     lcd_puts("masukan buah tomat");



       }
   }
```

LAMPIRAN C
DATA SHEET