LAMPIRAN

LAMPIRAN A

PROGRAM INISIALISASI SERIAL

```
/************************************
 Implementasi dari UART
 ************************************/

#include "uart.h"
#include "io306x.h"[1]



void uart_init(void)
{
        //Serial yang digunakan adalah SCI1
        SCI1.SMR.BYTE = 0; // 8 bit , no parity , 1 stop bit
        //Seting BRR 9600
        SCI1.BRR = 64;
        //Enablekan transmit dan Receive
        SCI1.SCR.BYTE = 0x30;
        //Clear receive flag
        SCI1.SSR.BYTE &=0x80;
        //Enablekan Receive interrupt
        SCI1.SCR.BYTE |= 0x40;
}

void uart_wr(char data)
{
        while (!SCI1.SSR.BIT.TDRE);
        SCI1.TDR = data;
        SCI1.SSR.BIT.TDRE = 0;
}

void uart_tx(char * data)
{
        unsigned char i = 0;
        if (!data) return;

        while (data[i]!='\0'){
                uart_wr(data[i]);
                i++;
        }
}

/*
void uart_twl(unsigned char * data, unsigned char len)
{

}
*/
```

```
/*
unsigned char uart_rx(unsigned char buff[])
{
        unsigned char len = 0;;

        if (!SCI1.SSR.BIT.RDRF) {
                return 0;
                }
        else {
                for (len = 0 ; len < MAX_BUFF; len++) {
                        while (!SCI1.SSR.BIT.RDRF);
                        buff[len] = SCI1.RDR;
                        SCI1.SSR.BIT.RDRF = 0;
                        //buff++;
                        //len++;
                }

                return len;
        }
}
*/
char uart_rd(void)
{
        char rxdt;
        rxdt = SCI1.RDR;
        SCI1.SSR.BIT.RDRF = 0;
        return rxdt;
}
```

PROGRAM INISIALISASI DELAY DAN INTERRUPT

```
/****************************
 implementasi mydef.h
****************************/

#include "mydef.h"

void msdelay(unsigned long ms)
{
        unsigned long loop  = 5 * ms;

   while(loop){
        loop--;
   }
}

void set_interrupt_mask (unsigned char mask)
{
  asm ("mov %0l, r0l"::"r"(mask));
  asm ("and.b #01,r0l");
  asm ("rotr.b r0l");
  asm ("ldc r0l,ccr");
}
```

Berdasarkan pengukuran dengan oscilloscope, di dapat hasil 10 µs jika program kosong di loop 5 kali.

PROGRAM LENGAN ROBOT PADA MIKROKONTROLLER DENGAN KOMUNIKASI SERIAL

```
/* Test uart sederhana */

#include "io306x.h"[1]
#include "mydef.h"
#include "uart.h"
#include "inthandler.h"[1]

//************************************
//Variabel untuk mengelola PWM

#define PWM1    P1DR.BIT.B0
#define PWM2    P1DR.BIT.B1
#define PWM3    P1DR.BIT.B2
#define PWM4    P1DR.BIT.B3
#define PWM5    P1DR.BIT.B4

#define MAX_POS         200
#define MIN_POS         38
#define INCREAMENT      1
#define waktu 200


unsigned char pwm1val;
unsigned char pwm2val;
unsigned char pwm3val;
unsigned char pwm4val;
unsigned char pwm5val;

unsigned char pwm1cur;
unsigned char pwm2cur;
unsigned char pwm3cur;
unsigned char pwm4cur;
unsigned char pwm5cur;

unsigned char rxdata = 0;
unsigned char i;


void pwm1task(void);
void pwm2task(void);
void pwm3task(void);
void pwm4task(void);
void pwm5task(void);

//***********************************************


int main ()
{

    P1DDR = 0xFF;
    P1DR.BYTE = 0;

    uart_init();
    set_interrupt_mask(0);
    uart_tx("Inisialisasi Broo\r\n");

    pwm1val = 139;
    pwm2val = 139;
    pwm3val = 135;
    pwm4val = 139;
    pwm5val = 139;

    pwm1cur =  MIN_POS;
    pwm2cur =  MIN_POS;
    pwm3cur =  MIN_POS;
    pwm4cur =  MIN_POS;
    pwm5cur =  MIN_POS;


    while (1){
        //Looping PWM forever
        pwm1task();
        msdelay(210);
        pwm2task();
        msdelay(210);
        pwm3task();
        msdelay(210);
        pwm4task();
        msdelay(210);
        pwm5task();
        msdelay(210);
            }

    return 0;
}

//Interrupt handler
void INT_RXI1(void)
{
    /*add your code here*/
    rxdata = uart_rd();

    if ((rxdata&0xE0) == 0x00){
```

```
                pwm1val = (((rxdata&0x1F)*10) + MIN_POS);
                return;
        }
        if ((rxdata&0xE0) == 0x20){
                pwm2val = (((rxdata&0x1F)*10) + MIN_POS);
                return;
        }

        if ((rxdata&0xE0) == 0x40){
                pwm3val = (((rxdata&0x1F)*10) + MIN_POS);
                return;
        }

        if ((rxdata&0xE0) == 0x60){
                pwm4val = (((rxdata&0x1F)*10) + MIN_POS);
                return;
        }
        if ((rxdata&0xE0) == 0x80){
                pwm5val = (((rxdata&0x1F)*10) + MIN_POS);
                return;
        }
        /*
        if ((rxdata&0xE0) == 0xA0){
                pwm6val = (((rxdata&0x1F)*5) + MIN_POS);
                return;
        }*/

        //uart_wr(rxdata);
}

//PWM Task
void pwm1task(void)
{
        if (pwm1cur == pwm1val){
                PWM1 = 1;
                msdelay(pwm1cur);
                PWM1 = 0;
        }
        else {
                if (pwm1cur <= pwm1val){
                        for ( i = pwm1cur; i <= pwm1val ; i += INCREAMENT){
                                PWM1 = 1;
                                msdelay(i);
                                PWM1 = 0;
                                msdelay(waktu);
                        }
                        pwm1cur = pwm1val;
                        return;
                }
                if (pwm1cur >= pwm1val){
                        for ( i = pwm1cur; i <= pwm1val ; i -= INCREAMENT){
                                PWM1 = 1;
                                msdelay(i);
                                PWM1 = 0;
                msdelay(waktu);
            }
            pwm1cur = pwm1val;
            return;
        }
        //pwm1cur = pwm1val;
    }
}
void pwm2task(void)
{
    if (pwm2cur == pwm2val){
        PWM2 = 1;
        msdelay(pwm2cur);
        PWM2 = 0;
    }
    else {
        if (pwm2cur <= pwm2val){
            for ( i = pwm2cur; i <= pwm2val ; i += INCREAMENT){
                PWM2 = 1;
                msdelay(i);
                PWM2 = 0;
                msdelay(waktu);
            }
            pwm2cur = pwm2val;
            return;
        }
        if (pwm2cur >= pwm2val){
            for ( i = pwm2cur; i <= pwm2val ; i -= INCREAMENT){
                PWM2 = 1;
                msdelay(i);
                PWM2 = 0;
                msdelay(waktu);
            }
            pwm2cur = pwm2val;
            return;
        }
        //pwm2cur = pwm2val;
    }
}

void pwm3task(void)
{
    if (pwm3cur == pwm3val){
        PWM3 = 1;
        msdelay(pwm3cur);
        PWM3 = 0;
    }
    else {
        if (pwm3cur <= pwm3val){
            for ( i = pwm3cur; i <= pwm3val ; i += INCREAMENT){
                PWM3 = 1;
                msdelay(i);
                PWM3 = 0;
                msdelay(waktu);
            }
            pwm3cur = pwm3val;
            return;
        }
        if (pwm3cur >= pwm3val){
            for ( i = pwm3cur; i <= pwm3val ; i -= INCREAMENT){
                PWM3 = 1;
                msdelay(i);
                PWM3 = 0;
                msdelay(waktu);
            }
            pwm3cur = pwm3val;
            return;
        }
        //pwm3cur = pwm3val;
    }
}

void pwm4task(void)
{
    if (pwm4cur == pwm4val){
        PWM4 = 1;
        msdelay(pwm4cur);
        PWM4 = 0;
    }
    else {
        if (pwm4cur <= pwm4val){
            for ( i = pwm4cur; i <= pwm4val ; i += INCREAMENT){
                PWM4 = 1;
                msdelay(i);
                PWM4 = 0;
                msdelay(waktu);
            }
            pwm4cur = pwm4val;
            return;
        }
        if (pwm4cur >= pwm4val){
            for ( i = pwm4cur; i <= pwm4val ; i -= INCREAMENT){
                PWM4 = 1;
                msdelay(i);
                PWM4 = 0;
                msdelay(waktu);
            }
            pwm4cur = pwm4val;
            return;
        }
        //pwm4cur = pwm4val;
    }
}

void pwm5task(void)
{
    if (pwm5cur == pwm5val){
        PWM5 = 1;
        msdelay(pwm5cur);
        PWM5 = 0;
    }
    else {
        if (pwm5cur <= pwm5val){
            for ( i = pwm5cur; i <= pwm5val ; i += INCREAMENT){
                PWM5 = 1;
                msdelay(i);
                PWM5 = 0;
                msdelay(waktu);
            }
            pwm5cur = pwm5val;
            return;
        }
        if (pwm5cur >= pwm5val){
            for ( i = pwm5cur; i <= pwm5val ; i -= INCREAMENT){
                PWM5 = 1;
                msdelay(i);
                PWM5 = 0;
                msdelay(waktu);
            }
            pwm5cur = pwm5val;
            return;
        }
        //pwm5cur = pwm5val;
    }
}
```

PROGRAM LENGAN ROBOT DENGAN METODE OTOMATIS

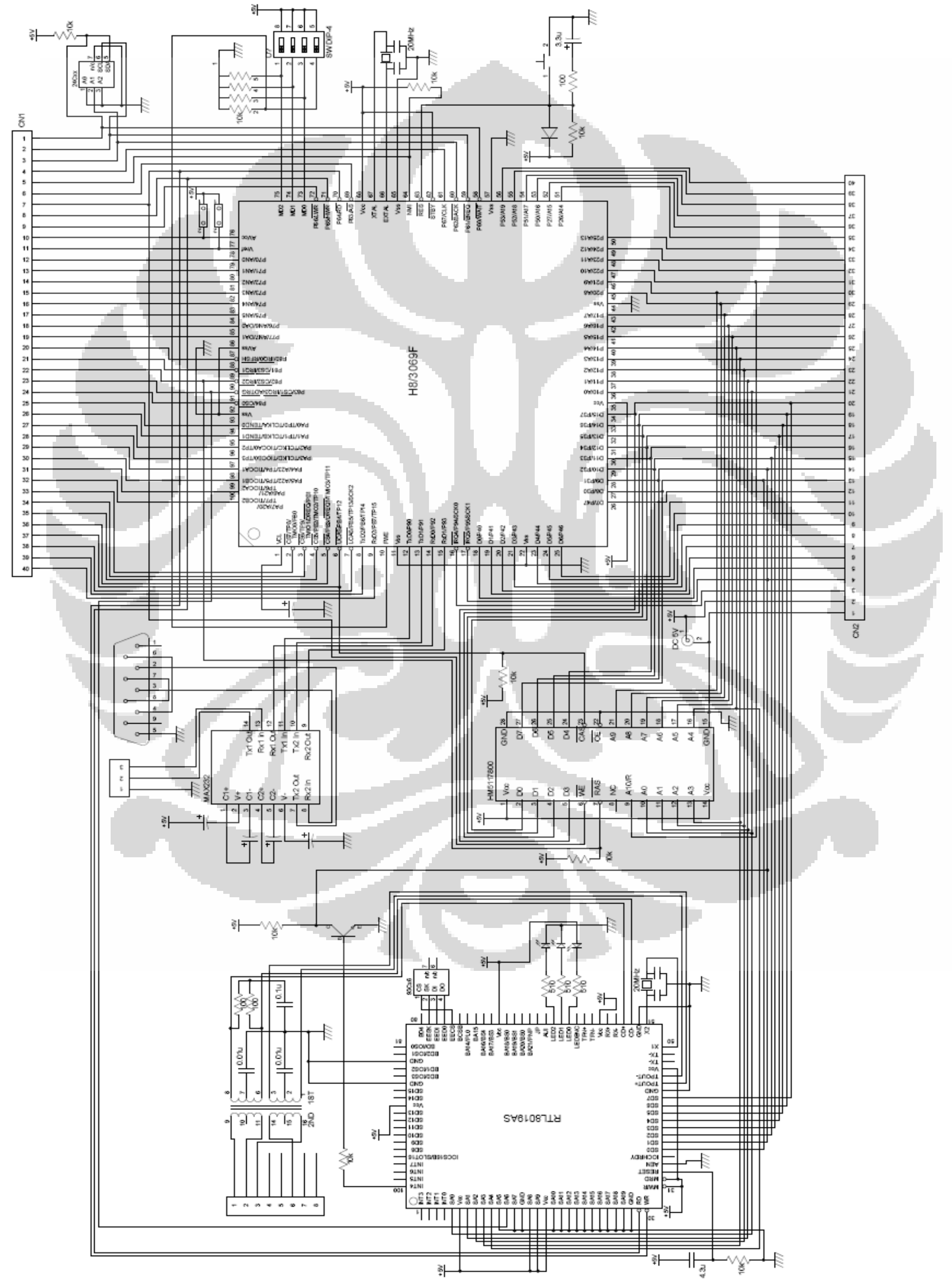
```
/***************************************************
   Latihan PWM (Pulse Wave Modulation)
***************************************************/
#include "io306x.h"[1]
#define waktu 1400
void delay(unsigned long ms)
{
   unsigned long loop  = 5 * ms;
      while(loop){
          loop--;
   }
   }
void gerak(int arah, int loop, int derajat, int data)
{
int a, b, i, j;
switch(arah) {
case 1:
{
        for(a=1;a<=loop;a++)
        {
                for(i=50;i<=derajat;i++)
                {
                        P1DR.BYTE = data;
                        delay(i);
                        P1DR.BYTE = 0;
                        delay(waktu);
                }[10]
        }
break;
}
case 0:
{
        for(b=1;b<=loop;b++)
        {
                for(j=derajat;j>=50;j--)
                {
                        P1DR.BYTE = data;
                        delay(j);
                        P1DR.BYTE = 0;
                        delay(waktu);
                }[10]
        }
break;
}
}
}
int main()
{
   P1DDR = 0xFF;
        P2DDR = 0x00;
//  fungsi(arah, loop, derajat, data);

        gerak(1, 1, 232, 1);
        gerak(0, 1, 232, 1);
        gerak(0, 1, 139, 7);

   return 0;
} [2]
```

LAMPIRAN B

RANGKAIAN SISTEM MINIMUM *H8/3069F*



Gambar B.1Rangkaian system minimum H8/3069F[1]