

Sistem Informasi Penjadwalan Waktu Sholat Berbasis Mikrokontroler ATmega16

Entin Sutinah^{1,*}

¹ Manajemen Informatika; AMIK Bina Sarana Informatika Jakarta; Jl. R.S Fatmawati No.24 Pondok Labu Jakarta Selatan, Indonesia; email: entin.esh@bsi.ac.id

* Korespondensi: entin.esh@bsi.ac.id

Diterima: 18 Oktober 2016 ; Review: 24 Oktober 2016; Disetujui: 31 Oktober 2016

Cara Sitasi: Sutinah E. 2016. Sistem Informasi Penjadwalan Waktu Sholat Berbasis Mikrokontroler ATmega16. *Informatics for Educators and Professionals*. 1 (1): 37 – 50.

Abstrak: Sholat merupakan salah satu rukun Islam yang kedua, dimana sholat lima waktu ini wajib hukumnya atas setiap orang muslim yang beriman, namun pada umumnya masyarakat muslim di Indonesia lebih banyak mengetahui waktu sholat dengan sistem yang masih tradisional yaitu dengan cara mendengarkan suara adzan di Masjid, kentongan, atau suara bedug. Tujuan pembahasan jurnal ini menjelaskan pembuatan sistem informasi penjadwalan waktu sholat dengan menggunakan teknologi modern yaitu dengan menggunakan microcontroller ATmega16, sehingga umat muslim dapat dengan mudah mengetahui waktu sholat. Metode yang digunakan dalam penelitian ini menggunakan *hardware: Microcontroller ATmega16, LCD 16x2, Power Supply 5v dan Komputer. Software* yang digunakan yaitu Code Vision AVR dan Proteun, sedangkan teknik pengumpulan datanya dengan mengumpulkan literature, jurnal dan bacaan-bacaan yang berhubungan dengan topik penelitian. Hasil dari penelitian ini yaitu Masyarakat dapat dengan mudah mengetahui jadwal waktu sholat, Aplikasi Jadwal Shalat dapat ditampilkan di LCD 16x2, Aplikasi Jadwal Shalat akan membantu pengguna dalam mengetahui waktu Shalat, Aplikasi Jadwal Shalat dapat diterapkan di LCD jenis apapun dan dapat dipasang di Masjid-masjid supaya memudahkan dalam melihat jadwal waktu shalat.

Kata kunci: atmega16, jadwal, sistem, sholat, waktu

Abstract: *Prayer is one of the second pillars of Islam, where it must be done five times a day on every Mosle. In Indonesia, prayer time used traditional custom that is listening to the sound of the call to prayer at the mosque, bell sounded by drum. This journal purpose describes the preparation of the prayer time scheduling information system using modern technology in this case is by using ATmega16 microcontroller, so that Moslems can easily find out the prayer time. The method used in this study using hardware: AVR microcontroller ATmega16, 16x2 LCD, Power Supply 5v and computer. software used is Code Vision AVR, Proteun, while data collection technique by collecting literature, journals and readings related to the research topic. The results of this study hope that public can easily find out the time schedule of prayer, Applications Schedule Prayer can be displayed on the LCD 16x2, Applications Schedule Prayer will help people know the prayer times, Applications Schedule Prayer can be applied in LCD of any type and can be mounted in mosque in order to facilitate the view prayer schedule.*

Keywords: atmega16, pray, schedule, system, time

1. Pendahuluan

Sholat merupakan salah satu rukun Islam yang kedua, dimana sholat lima waktu ini wajib hukumnya atas setiap orang muslim yang beriman, seperti yang tercantum dalam Quran Surat An Nisa ayat 103 yaitu "Sesungguhnya salat itu adalah kewajiban yang ditentukan waktunya atas orang-orang yang beriman."

Seiring dengan kemajuan teknologi saat ini yang begitu pesat dan kebutuhan manusia semakin meningkat serta adanya rasa keingintahuan yang tinggi pada diri manusia, maka dari

sinilah ilmu semakin berkembang dan dengan ilmu terciptalah teknologi-teknologi yang begitu modern seperti yang kita rasakan saat ini, dengan adanya teknologi dapat membantu kemampuan berfikir manusia secara logis dalam memperoleh pengetahuan tentang fakta-fakta baik natural maupun sosial yang berlaku umum dan sistematis.

Pada umumnya masyarakat muslim di Indonesia lebih banyak mengetahui waktu sholat dengan sistem yang masih terdisonal yaitu dengan cara mendengarkan suara adzan di Masjid, kentongan, atau suara bedug. Maka dari itu tujuan pembahasan penelitian ini menjelaskan tentang pembuatan sistem informasi penjadwalan waktu sholat dengan menggunakan teknologi modern yaitu dengan menggunakan *Microcontroller* ATMEGA 16 sehingga umat muslim dapat dengan mudah mengetahui waktu sholat serta dapat bersaing dalam dunia teknologi informasi.

a. Mikrokontroler

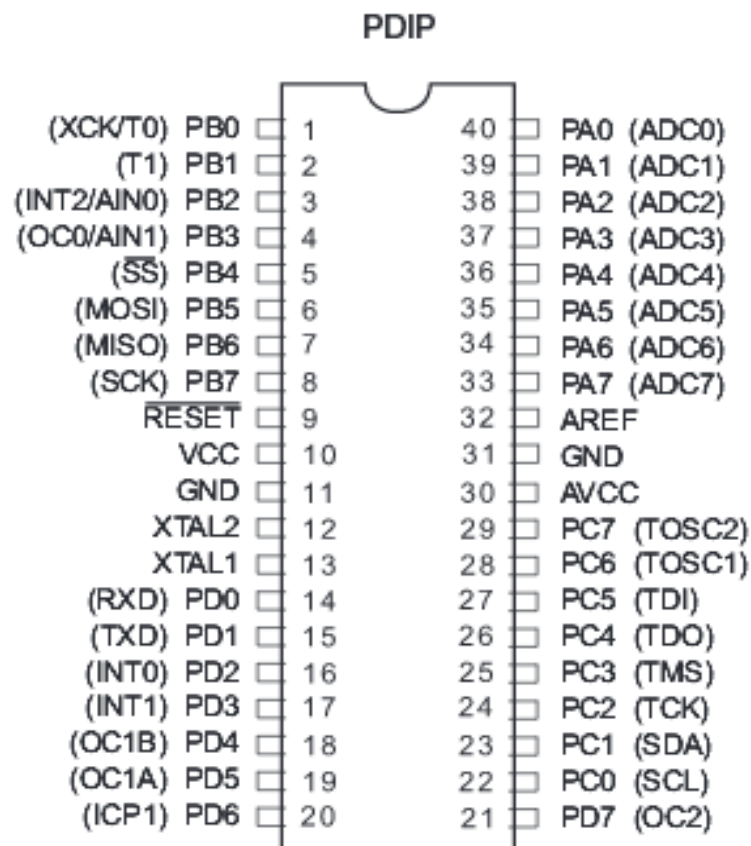
“Mikrokontroler adalah suatu chip yang dapat digunakan sebagai pengontrol utama sistem elektronika, misalnya sistem pengukur suhu digital (*termometer digital*), sistem keamanan rumah (*home remote system*), sistem kendali industri, robot penjinak bom dan lain-lain” Budiharto dan Rizal (2007: 27). Dalam chip mikrokontroler sudah tersedia prosesor, ROM, RAM, *Input* dan *Output* serta fasilitas pendukung lainnya.

1) Arsitektur ATMEGA16

Mikrokontroler ini memiliki arsitektur RISC (*Reduced Instruction Set Computing*) 8 bit dimana semua instruksi dikemas dalam kode 16 bit (16-bits *word*) dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*.

a) Konfigurasi PIN ATMEGA16

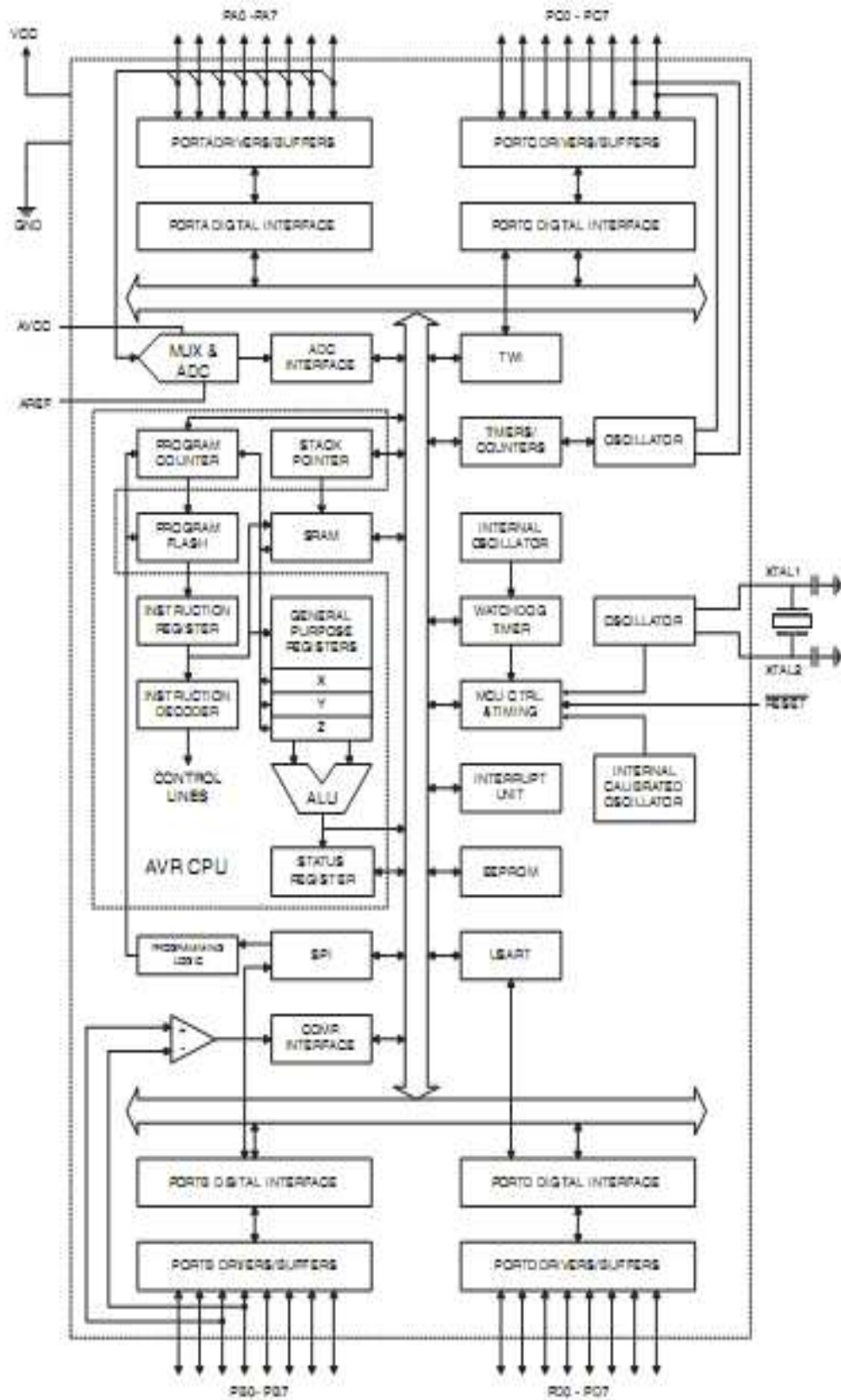
PIN ini memiliki fungsi masing-masing ada yang berupa input dan ada juga yang berupa output atau bisa juga berfungsi untuk kedua-duanya, tergantung pengaturan dari program yang dibuat.



Sumber : Wardhana (2006:4)

Gambar 1. Konfigurasi PIN ATMEGA16

b) Blok Diagram Fungsional ATMEGA16
 Diagram ini mnggambarkan arsitektur IC ATMEGA16



Sumber : Wardhana (2006:2)

Gambar 2. Blok Diagram Fungsional ATMEGA16

b. Bahasa C

Bahasa C merupakan Bahasa pemrograman tingkat menengah yang merupakan perkembangan dari Bahasa B. bahasa C diciptakan oleh Dennis Ritchie pada tahun 1972. Bahasa C pertama kali dioperasikan pada sistem operasi Unix di Bell Telephone Laboratories USA.

c. LCD (*Liquid Cristal Display*) 16x2

"LCD adalah modul penampil yang banyak digunakan karena tampilannya lebih menarik, LCD yang paling banyak digunakan adalah LCD M1632 *refurbish* karena harganya yang relatif murah, LCD M1632 merupakan modul LCD dengan tampilan 16x2 (16 kolom dan 2 baris) dengan konsumsi daya rendah, modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD" Budiharto dan Rizal (2007: 51).

a) Daftar PIN LCD 16x2

Tabel 1. Daftar PIN LCD 16x2

Nama Pin	Keterangan	Port
VCC	+5V	Vcc
GND	0 V	GND
VEE	Tegangan kontras LCD	
RS	Register Select, 0=Input Instruksi, 1=Input Data	PD7
R/W	1= Read, 0=Write	PD5
E	Enable Clock	PD6
D4	Data Bus 4	PC4
D5	Data Bus 5	PC5
D6	Data Bus 6	PC6
D7	Data Bus 7	PC7
Anode	Tegangan Positif backlight	
Katode	Tegangan Negatif backlight	

Sumber: Budiharto dan Rizal (2007: 52)

2. Metode Penelitian

Dalam penelitian ini alat pendukung yang digunakan terdiri atas:

Hardware :

- Microcontroller ATmega 16
- LCD 16x2 merupakan modul penampil (*output*) dari hasil pemrosesan
- Power Supply* 5v sebagai sumber daya.
- Komputer Untuk pembuatan *code* program, simulasi dan *download* data ke mikrokontrol.

Software :

- CodeVisionAVR untuk pembuatan *source code* perhitungan jadwal sholat dengan menggunakan bahasa C yang nantinya hasil *compile* berupa file .HEX di *download* ke IC Mikrokontrol.
- Proteus merupakan simulator untuk melihat hasil kerja rangkaian dari *source code* program yang telah dibuat.

Adapun metodologi pengumpulan datanya dengan cara mengumpulkan literature, jurnal dan bacaan-bacaan yang berhubungan dengan topik penelitian.

3. Hasil dan Pembahasan

Penentuan Waktu Shalat Menggunakan Mikrokontroler ATmega16

a. Blok skema rangkaian:

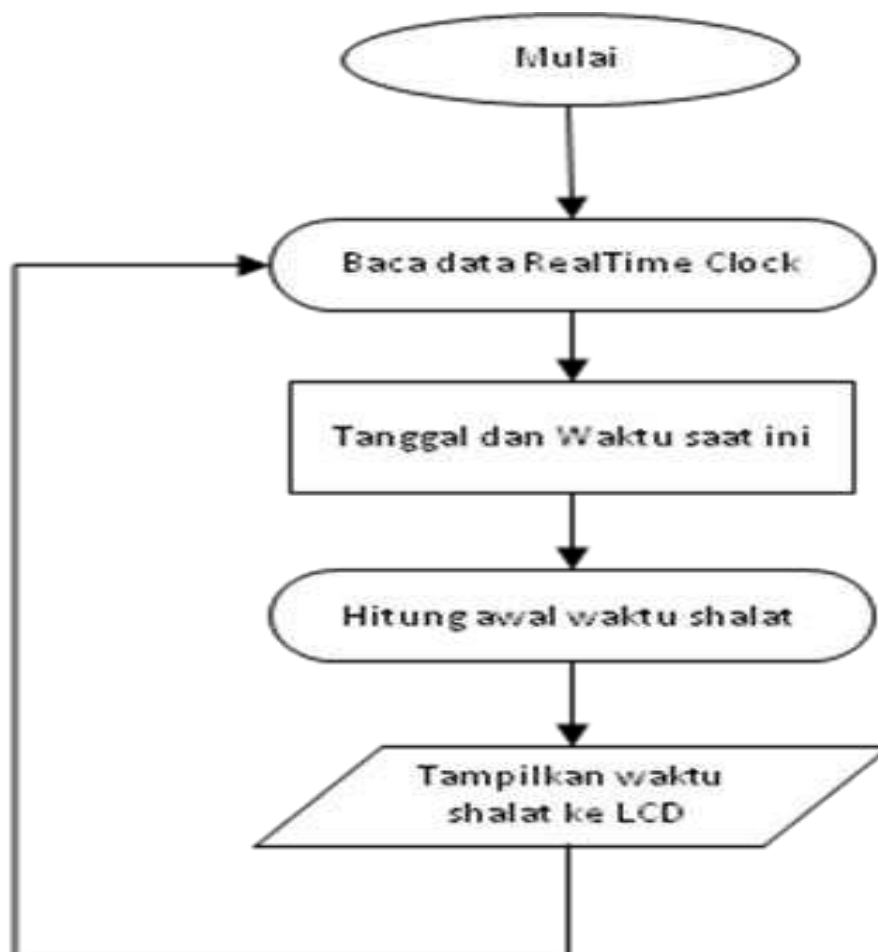


Gambar 3. Blok skema rangkaian

- 1) *Realtime Clock* (RTC) merupakan jam elektronik berupa bentuk chip IC dengan type DS1307 yang memiliki fungsi sebagai penghasil data waktu tahun, bulan, tanggal, jam, menit, dan detik.
- 2) ATmega 16 merupakan IC *processor* yang berfungsi menangani semua proses pengolahan data mulai dari penampilan data dari RTC kemudian melakukan perhitungan-perhitungan dan hasil dari semua pengolahan akan dikirim ke LCD untuk bisa dilihat.
- 3) LCD 16x2 merupakan modul peraga digital dengan 16 kolom dan 2 baris.

b. **Flow Diagram :**

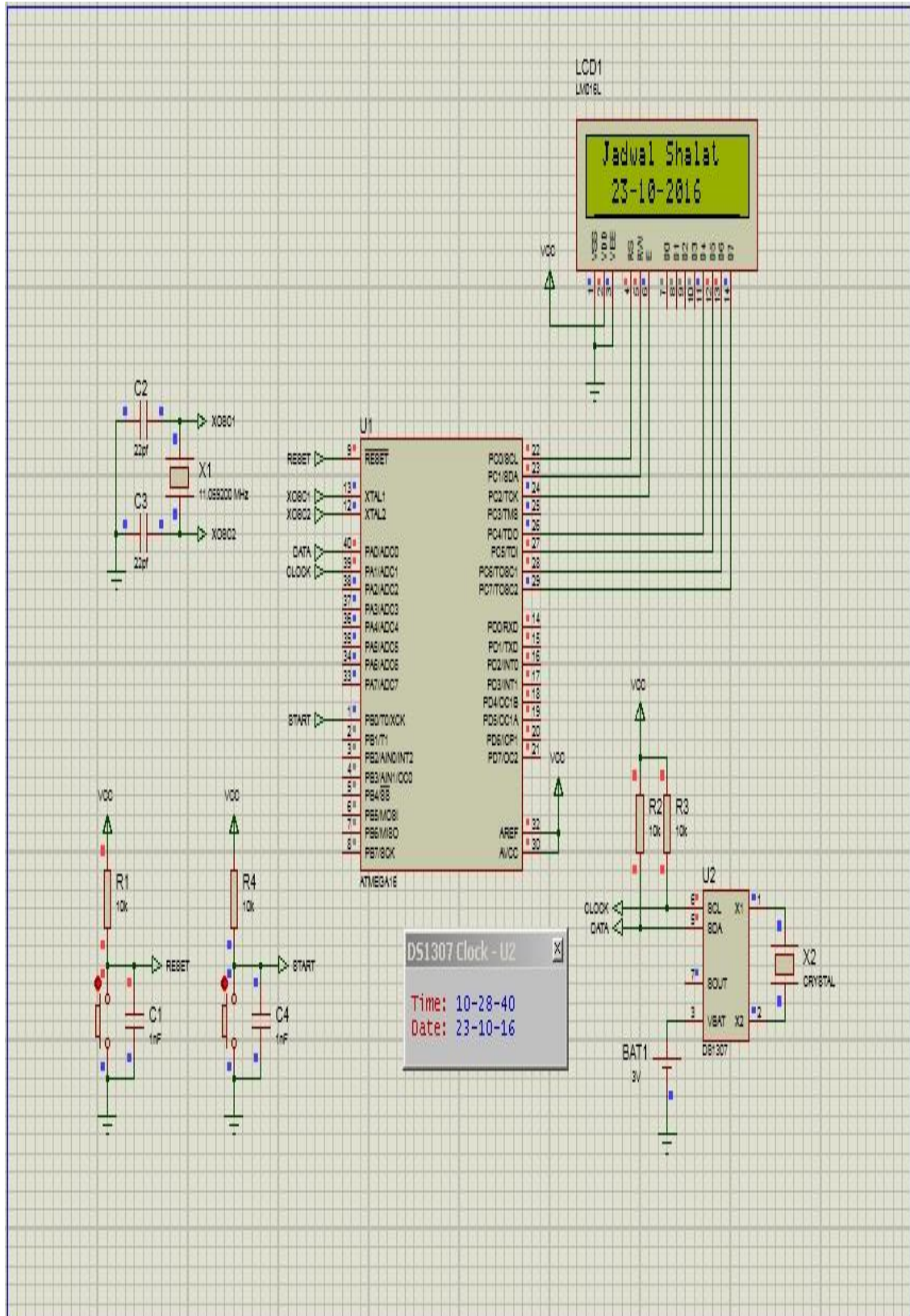
Flow diagram menggambarkan alur dari sistem yang akan dibuat .



Gambar 4. *Flow Diagram*

c. **Skema rangkain dan simulator**

Pembuatan skema rangkain perhitungan waktu sholat dengan menggunakan *software* Proteus dan *output* ditampilkan pada LCD 16x2.



Gambar 5. Skema rangkaian dan simulator dibuat pada *software* Proteus

d. Penentuan awal waktu sholat:

Dalam penentuan awal waktu sholat secara hisab ada beberapa parameter yang harus diketahui yaitu: LATITUDE (Koordinat lintang daerah), LONGITUDE (Koordinat bujur daerah), TZ (Timezone atau zona waktu sesuai daerah), H (Ketinggian daerah dari permukaan air laut), HaSUBUH (*Hour Angle* subuh), HaSYA (*Hour Angle* isya), D (Tanggal), M (bulan), Y (tahun), Delta (Sudut Deklinasi Matahari), ET (Equation of Time), JD12UT, JD12LT (Julian Day). Tanggal, Bulan, Tahun merupakan Parameter waktu yang akan dihitung dengan menggunakan kalender Gregorian. Tanggal, bulan dan tahun digunakan untuk menentukan waktu sholat pada tanggal tersebut.

Langkah selanjutnya menentukan Julian Day, dikarekan kalender yang digunakan saat ini adalah kalender masehi, maka dapat ditentukan rumus Julian day sebagai berikut :

$$JD12UT = 1720994,5 + INT(365,25*Y) + INT(30,6001(M + 1)) + LONGITUDE + D$$

$$JD12LT = JD12UT - (TZ/24)$$

Nilai JD tersebut, dihitung sudut tanggal T dengan rumus:

$$T = 2 * PI * (JD12LT - 2451545)/365.25$$

PI adalah konstanta yang bernilai 3,14159265359

Dari sudut tanggal T di atas, deklinasi Matahari (Delta) untuk satu tanggal tertentu dapat dihitung dengan menggunakan rumus berikut:

$$DELTA = (0.37877 + 23.264*\sin((57.297*T - 79.547)*PI/180) + 0.3812*\sin((2*57.297*T - 82.682)*PI/180) + 0.17132*\sin((3*57.297*T - 59.722)*PI/180))*PI/180$$

Equation of Time untuk satu tanggal tertentu dapat dihitung sebagai berikut, Pertama kali perlu dihitung dahulu Bujur rata-rata Matahari L0 yang dirumuskan:

$$U = (JD12LT - 2451545)/36525$$

$$L0 = (280.46607 + 36000.7698*U) * PI/180$$

L0 bersatuan derajat. Selanjutnya *Equation of Time* dapat dirumuskan sebagai berikut:

$$L02 = (-1*(1789 + 237*U)*\sin(L0) - (7146 - 62*U)*\cos(L0) + (9934 - 14*U)*\sin(2*L0) - (29 + 5*U)*\cos(2*L0) + (74 + L0*U)*\sin(3*L0) + (320 - 4*U)*\cos(3*L0) - 212*\sin(4*L0))/1000$$

Rumus terakhir untuk menentukan awal waktu sholat dan terbitnya Matahari adalah sebagai berikut:

- 1) Dzuhur = 12 + Z - B/15 - ET/60
- 2) Ashar = Dzuhur + (Hour Angle Ashar)/15
- 3) Maghrib = Dzuhur + (Hour Angle Maghrib)/15
- 4) Isya' = Dzuhur + (Hour Angle Isya')/15
- 5) Shubuh = Dzuhur - (Hour Angle Shubuh)/15
- 6) Terbit Matahari = Dzuhur - (Hour Angle Terbit Matahari)/15

Awal waktu sholat bergantung pada Hour Angle. Rumus Hour Angle (Ha) :

$$\cos(HA) = [\sin(\text{Altitude}) - \sin(\text{Lintang}) * \sin(\text{Delta})] / [\cos(\text{Lintang}) * \cos(\text{Delta})]$$

e. Impelementasi rumus perhitungan waktu sholat dengan menggunakan bahasa C

```
#include <mega16.h>
#include <mega16.h>
#include <lcd.h>
#include <delay.h>
#include <stdio.h>
#include <math.h>
#include <i2c.h>
#include <ds1307.h>
```

```
// I2C Bus functions
```

```
#asm
```

```
.equ __i2c_port=0x1B ;PORTA
```

```
.equ __sda_bit=0
```

```

.equ __scl_bit=1
#endasm

// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm

// Declare your global variables here
const unsigned char cpos = 10;
unsigned char buff[16];
//unsigned char h,m,s;
char week,day,month,year;
//char __hour,__min,__sec;

/*****/
float LATITUDE, LAT = -1*6.2830798;
float LONGITUDE = 106.9420819;
int TZ = 7;
int H = 50;
float HaSUBUH = 20;
float HaSYA = 18;
int TGL, BLN , THN ;

float JD12UT, JD12LT, DELTA, T, U, L0, L02, DELTA;
float __abs, __pw, __pwc, __jd, __t, __ttan, __u, __l0, __l02, __d, __w;
float __alt, __kalt, __alt, __alts, __cos, __ha;
int __v, __bln, __thn;
int __i1, __i2, __i3;
float __f1, __f2;

void init(){
    rtc_get_date(&week, &day ,&month ,&year);
    TGL = day;
    BLN = month;
    THN = 2000+year;
    LATITUDE= LAT * PI / 180;
    __bln = BLN <= 2 ? BLN+12 : BLN;
    __thn = BLN <= 2 ? THN-1 : THN;
    __i1 = THN/100;
    __i2 = __i1/4;
    __i3 = 2 + __i2 - __i1;
    __f1 = 365.25*(__thn);
    __f2 = 30.6001* ((int)(__bln + 1));
    JD12UT = 1720994.5 + __f1 + __f2 + __i3 + TGL + (12/24);
    JD12LT = JD12UT-(TZ/24);
    T = 2 * PI * (JD12LT - 2451545)/365.25;
    DELTA = (0.37877 + 23.264*sin((57.297*T - 79.547)*PI/180) + 0.3812*sin((2*57.297*T -
82.682)*PI/180) + 0.17132*sin((3*57.297*T - 59.722)*PI/180))*PI/180;
    U = (JD12LT - 2451545)/36525;
    L0 = (280.46607 + 36000.7698*U)* PI/180;
    L02 = (-1*(1789 + 237*U)*sin(L0) - (7146 - 62*U)*cos(L0) + (9934 - 14*U)*sin(2*L0) - (29 +
5*U)*cos(2*L0) + (74 + L0*U)*sin(3*L0) + (320 - 4*U)*cos(3*L0) - 212*sin(4*L0))/1000;
}

float Dzuhur(){
    __pw = ((float)12+TZ)-(LONGITUDE/15)-(L02/60);
    __jd = JD12LT + (__pw/24)-(12/24);
}

```



```

__t = 2*PI*(__jd - 2451545)/365.25;
__u = (JD12LT - 2451545)/36525;
__l0 = (280.46607 + 36000.7698*__u)*PI/180;
__d = (0.37877 + 23.264*sin((57.297*__l0 - 79.547)*PI/180) + 0.3812*sin((2*57.297*__t -
82.682)*PI/180) + 0.17132*sin((3*57.297*__t - 59.722)*PI/180))*PI/180;
__l02 = (-1*(1789 + 237*__u)*sin(__l0) - (7146 - 62*__u)*cos(__l0) + (9934 -
14*__u)*sin(2*__l0) - (29 + 5*__u)*cos(2*__l0) + (74 + 10*__u)*sin(3*__l0) + (320 -
4*__u)*cos(3*__l0) - 212*sin(4*__l0))/1000;
__w = ((float)12+TZ)-(LONGITUDE/15)-(__l02/60);
return __w;
}

```

```

float Ashar() {
__abs = LATITUDE - DELTA;
__abs = __abs<0 ? __abs-1*__abs : __abs;

__pwc = (sin(atan(1/(1+tan(__abs))))-
sin(DELTA)*sin(LATITUDE))/(cos(DELTA)*cos(LATITUDE));
__pw = acos(__pwc)*180/PI;
__jd = JD12LT+(__pw/360);
__t = 2*PI*(__jd-2451545)/365.25;
__u = (__jd-2451545)/36525;
__l0 = (280.46607 + 36000.7698*__u)*PI/180;
__d = (0.37877+23.264*sin((57.297*__t-79.547)*PI/180)+0.3812*sin((2*57.297*__t-
82.682)*PI/180)+0.17132*sin((3*57.297*__t-59.722)*PI/180))*PI/180;
__l02 = (-1*(1789 + 237*__u)*sin(__l0) - (7146 - 62*__u)*cos(__l0) + (9934 -
14*__u)*sin(2*__l0) - (29 + 5*__u)*cos(2*__l0) + (74 + 10*__u)*sin(3*__l0) + (320 -
4*__u)*cos(3*__l0) - 212*sin(4*__l0))/1000;

__abs = LATITUDE - __d;
__abs = __abs<0 ? __abs-1*__abs : __abs;

__alt = atan(1/(1+tan(__abs)))*180/PI;
__kalt = 1/(60*tan((__alt+7.31/(__alt+4.4))*PI/180));
__alts = (__alt-__kalt)*PI/180;
__cos = (sin(__alts)-sin(__d)*sin(LATITUDE))/(cos(__d)*cos(LATITUDE));
__ha = acos(__cos)*180/PI;
__w = ((float)12+TZ)-LONGITUDE/15-__l02/60+__ha/15;
return __w;
}

```

```

float Maghrib() {
__pwc = (sin((-0.8333-0.0347*pow(H,0.5))*PI/180)-
sin(DELTA)*sin(LATITUDE))/(cos(DELTA)*cos(LATITUDE));
__pw = acos((sin((-0.8333-0.0347*pow(H,0.5))*PI/180)-
sin(DELTA)*sin(LATITUDE))/(cos(DELTA)*cos(LATITUDE)))*180/PI;
__jd = JD12LT + __pw/360;
__t = 2*PI*(__jd-2451545)/365.25;
__u = (__jd-2451545)/36525;
__l0 = (280.46607 + 36000.7698*__u)*PI/180;
__d = (0.37877+23.264*sin((57.297*__t-79.547)*PI/180)+0.3812*sin((2*57.297*__t-
82.682)*PI/180)+0.17132*sin((3*57.297*__t-59.722)*PI/180))*PI/180;
__l02 = (-1*(1789 + 237*__u)*sin(__l0) - (7146 - 62*__u)*cos(__l0) + (9934 -
14*__u)*sin(2*__l0) - (29 + 5*__u)*cos(2*__l0) + (74 + 10*__u)*sin(3*__l0) + (320 -
4*__u)*cos(3*__l0) - 212*sin(4*__l0))/1000;
__cos = (sin((-0.8333-0.0347*pow(H,0.5))*PI/180)-
sin(__d)*sin(LATITUDE))/(cos(__d)*cos(LATITUDE));
__ha = acos(__cos)*180/PI;

```

```

__w = ((float)12+TZ)-(LONGITUDE/15)-(__l02/60)+(__ha/15);
return __w;
}

float Isya() {
__f1 = HaISYA * PI/180;
__pwc = (sin(-1*HaISYA)-sin(__d)*sin(LATITUDE))/(cos(__d)*cos(LATITUDE));
__pw = acos(__pwc)*180/PI;
__jd = JD12LT + __pw/360;
__ttan = 2*PI*(__jd-2451545)/365.25;
__u = (__jd-2451545)/36525;
__l0 = (280.46607 + 36000.7698*__u)*PI/180;
__d = (0.37877+23.264*sin((57.297*__ttan-79.547)*PI/180)+0.3812*sin((2*57.297*__ttan-
82.682)*PI/180)+0.17132*sin((3*57.297*__ttan-59.722)*PI/180))*PI/180;
__l02 = (-1*(1789 + 237*__u)*sin(__l0) - (7146 - 62*__u)*cos(__l0) + (9934 -
14*__u)*sin(2*__l0) - (29 + 5*__u)*cos(2*__l0) + (74 + 10*__u)*sin(3*__l0) + (320 -
4*__u)*cos(3*__l0) - 212*sin(4*__l0))/1000;
__cos = (sin(-1*__f1)-sin(__d)*sin(LATITUDE))/(cos(__d)*cos(LATITUDE));
__ha = acos(__cos)*180/PI;
__w = ((float)12+TZ)-LONGITUDE/15-__l02/60+__ha/15;
return __w;
}

float Subuh() {
__f1 = HaSUBUH * PI/180;
__pwc = (sin(-1*__f1)-sin(DELTA)*sin(LATITUDE))/(cos(DELTA)*cos(LATITUDE));
__pw = acos(__pwc)*180/PI;
__jd = JD12LT-__pw/360;
__ttan = 2*PI*(__jd-2451545)/365.25;
__u = (__jd-2451545)/36525;
__l0 = (280.46607 + 36000.7698*__u)*PI/180;
__d = (0.37877+23.264*sin((57.297*__ttan-
79.547)*PI/180)+0.3812*sin((2*57.297*__ttan-82.682)*PI/180)+0.17132*sin((3*57.297*__ttan-
59.722)*PI/180))*PI/180;
__l02 = (-1*(1789 + 237*__u)*sin(__l0) - (7146 - 62*__u)*cos(__l0) + (9934 -
14*__u)*sin(2*__l0) - (29 + 5*__u)*cos(2*__l0) + (74 + 10*__u)*sin(3*__l0) + (320 -
4*__u)*cos(3*__l0) - 212*sin(4*__l0))/1000;
__cos = (sin(-1*__f1)-sin(__d)*sin(LATITUDE))/(cos(__d)*cos(LATITUDE));
__ha = acos(__cos)*180/PI;
__w = ((float)12+TZ)-LONGITUDE/15-__l02/60-__ha/15;
return __w;
}

char *RadToDate(float data) {
__i1 = data;
__i2 = (data - __i1) * 60;

sprintf(buff, "%c", "");
if (__i1<10 && __i2<10) {
    sprintf(buff, "0%d:0%d", __i1, __i2);
}
else if (__i1>9 && __i2<10) {
    sprintf(buff, "%d:0%d", __i1, __i2);
}
else if (__i1<10 && __i2>9) {
    sprintf(buff, "0%d:%d", __i1, __i2);
}
else {

```

```

    sprintf(buff, "%d:%d", __i1, __i2);
}

return buff;
}

void TampilJadwal() {
    __v++;
    lcd_clear();
    lcd_gotoxy(0,0);

    switch(__v) {
        case 1 :{
            lcd_gotoxy(0,0);
            lcd_puts(" Jadwal Shalat ");
            lcd_gotoxy(0,1);
            sprintf(buff, " %d-%d-%d", TGL, BLN, THN);
            lcd_puts(buff);
            break;
        }
        case 2 :{
            __f1 = Subuh();
            lcd_gotoxy(0,0);
            lcd_puts("Subuh :");
            lcd_gotoxy(cpos,0);
            lcd_puts(RadToDate(__f1));

            __f1 = Dzuhur();
            lcd_gotoxy(0,1);
            lcd_puts("Dzuhur :");
            lcd_gotoxy(cpos,1);
            lcd_puts(RadToDate(__f1));
            break;
        }
        case 3 :{
            __f1 = Ashar();
            lcd_gotoxy(0,0);
            lcd_puts("Ashar :");
            lcd_gotoxy(cpos,0);
            lcd_puts(RadToDate(__f1));

            __f1 = Maghrib();
            lcd_gotoxy(0,1);
            lcd_puts("Maghrib :");
            lcd_gotoxy(cpos,1);
            lcd_puts(RadToDate(__f1));
            break;
        }
        case 4 :{
            __f1 = Isya();
            lcd_gotoxy(0,0);
            lcd_puts("Isya :");
            lcd_gotoxy(cpos,0);
            lcd_puts(RadToDate(__f1));
            break;
        }
        case 5 :{
            lcd_gotoxy(0,0);

```

```

        lcd_puts(" - Wassallam -");
        break;
    }
    default:{
        __v = 0;
        __f1 = 0;
        init();
        break;
    }
}

if (__v>0) {
    delay_ms(250);
}
}

void Menu() {
    if (PINB.0 ==0) {
        lcd_gotoxy(0,0);
        lcd_puts("PINB Hits");
        while(PINB.0 == 0){
        }
    }
    else {
        lcd_clear();
    }
}

/*****/
void main(void) {
    PORTA=0x00;DDRA=0xff;
    PORTB=0x00;DDRB=0x00;
    PORTC=0x00;DDRC=0xff;
    PORTD=0xff;DDRD=0x00;

    ACSR=0x80;
    SFIOR=0x00;
    // I2C Bus initialization
    i2c_init();
    rtc_init(0,0,0);
    // LCD module initialization
    lcd_init(16);
    lcd_clear();

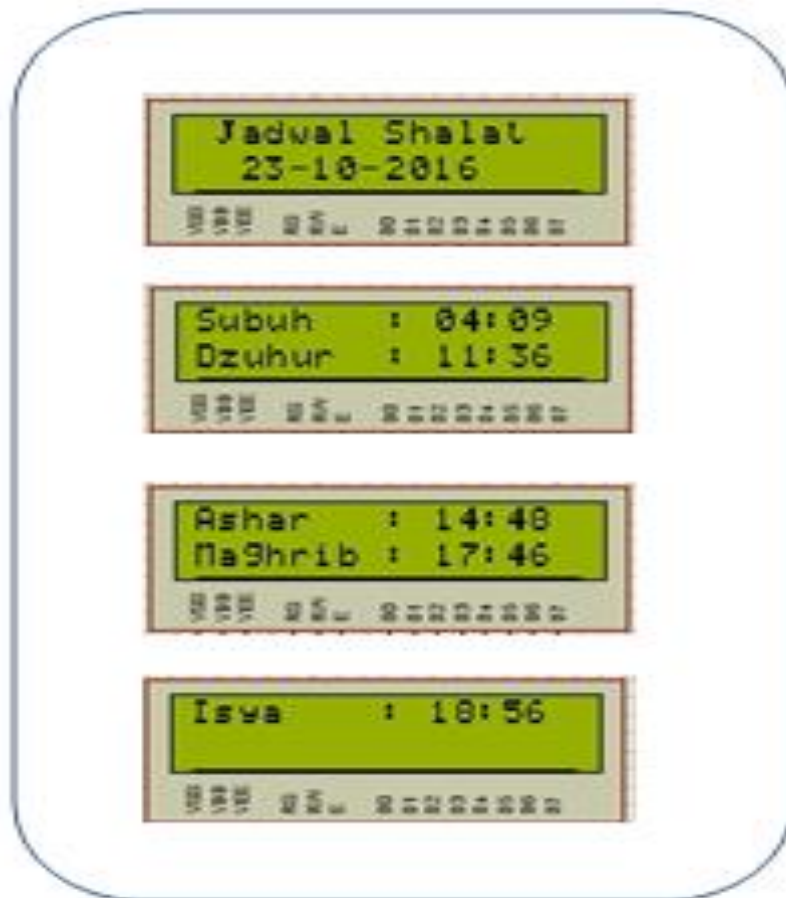
    init();
    __v = 0;

    while (1) {
        TampilJadwal();
        delay_ms(1000);
    };
}

```

f. Hasil perhitungan waktu sholat ditampilkan pada LCD 16x2

Coding Program dengan menggunakan bahasa C dapat ditampilkan *output*nya pada LCD 16x2



Gambar 6. *Output* jadwal waktu Sholat

4. Kesimpulan

Dari seluruh penjelasan dan uraian yang telah dituliskan pada pembahasan, maka kesimpulan yang dapat diambil adalah, dengan dibuatkannya sistem informasi waktu sholat dengan menggunakan mikrokontroler Atmega16 maka:

- Masyarakat dapat dengan mudah mengetahui jadwal waktu sholat dan bersiap menunaikan sholat tanpa harus mendengarkan suara Adzan di Masjid
- Simulasi *output* sistem informasi jadwal waktu sholat ditampilkan di LCD 16x2
- Sistem informasi jadwal waktu sholat akan membantu pengguna khususnya masyarakat muslim dimanapun berada dalam mengetahui waktu Shalat
- Sistem informasi jadwal waktu sholat dapat diterapkan di LCD jenis apapun dan dapat dipasang di masjid-masjid maupun di jalanan umum supaya memudahkan dalam melihat jadwal waktu sholat.

Referensi

- Atmel. 2010. Microcontroller with 16K Bytes In-System Programmable Flash. <http://www.atmel.com/images/doc2466.pdf> : 2-3.
- Widodo B, Rizal G. 2007. Belajar Sendiri 12 Proyek Mikrokontroler Untuk Pemula. Jakarta : PT.Elex Media Komputindo : 27, 51, 52, 126.
- Dwi H, Raharjo R. 2005. Visual Downloader untuk Microcontroller AT89C2051. Yogyakarta: Andi Offset: 59-75.

- Muhammad ID, Mintarsih F, Fahrianto F. 2013. Implementasi Algoritma Meeus Dalam Penentuan Waktu Shalat Dan Pencarian Masjid Terdekat. *Studia Informatika: Jurnal Sistem Informasi*, p-ISSN 1979-0767 8 (1):1-10.
- Maspeotra, Nabhan S. 2010. Perhitungan awal waktu shalat. *Orientasi Hisab Rukyat Ditjen Badilang Mahkamah Agung RI*: 5-14
- Riyadi S. 2011. Penentuan Awal – Akhir Waktu Sholat. *Orientasi Hisab-Rukyah bagi Takmir dan Utusan Pesantren se-Solo Raya. Kemenag Surakarta*.1-11.
- Wardhana L. 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi Hardware Dan Aplikasi*. Yogyakarta: Andi Offset : 1-4.