

Perancangan Subrutin Master pada Sistem Waktu Perkuliahan

Riandini, Is Hendarto
Jurusan Teknik Elektro
Politeknik Negeri Jakarta
Kampus Baru UI Depok, Indonesia
riandini@gmail.com, is.hendarto@gmail.com

Diterima: 17 September 2015. Disetujui: 20 Oktober 2015. Dipublikasikan: November 2015

Abstrak - Subrutin master pada sistem waktu perkuliahan merupakan program penunjang sistem yang berbasis Mikrokontroler. Sistem waktu perkuliahan sendiri merupakan sebuah sistem yang diharapkan dapat menciptakan kedisiplinan waktu kuliah pada Program Studi Teknik Elektronika Industri, dimana sistem tersebut dikontrol dan dimonitor melalui PC dengan menggunakan LabVIEW8.5 yang dihubungkan juga ke Mikrokontroler ATmega8535 melalui komunikasi serial RS232. Perancangan subrutin master dimaksudkan agar Mikrokontroler dapat bekerja sendiri (*stand-alone*) yaitu tetap menampilkan jam dan mengontrol *loudspeaker* walaupun sudah tidak ada komunikasi dengan LabVIEW. Prinsip kerjanya adalah subrutin program master akan dieksekusi oleh Mikrokontroler apabila dalam waktu kurang dari 2 detik tidak ada komunikasi dari LabVIEW dan segera mengambil data terakhir dari RTC DS1307 yang telah di-set oleh LabVIEW sesaat sebelum komunikasinya terputus.

Kata Kunci: subrutin program master, *stand-alone*, mikrokontroler ATmega8535, RTC DS1307, LabVIEW.

I. PENDAHULUAN

Waktu adalah sesuatu yang berharga bagi manusia. Dengan mengatur waktu kita dapat menjalankan aktivitas dengan teratur. Di samping waktu yang terencana dengan baik, informasi yang akurat juga menunjang kita dalam menjalankan segala aktivitas. Dalam sistem perkuliahan Politeknik Negeri Jakarta khususnya Program Studi Teknik Elektronika Industri masih terdapat kendala yang berhubungan dengan waktu seperti jadwal kuliah yang masih konvensional yaitu ditulis pada selembar kertas dan ditempel pada ruang-ruang kelas serta tidak ada tanda untuk pergantian jam kuliah.

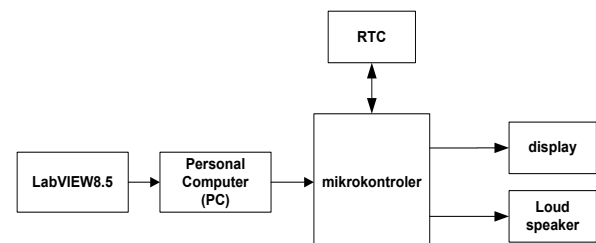
Maka dirancang sebuah sistem Sistem Waktu Perkuliahan yang diharapkan dapat membantu memudahkan dalam memonitor jadwal perkuliahan di Program Studi Elektronika Industri. Proses kontrol dan *monitoring* sistem ini dilakukan oleh program aplikasi LabVIEW 8.5 dan Mikrokontroler ATmega8535 yang saling berhubungan.

Mikrokontroler diprogram agar dapat bekerja sendiri (*stand-alone*) dengan tetap menampilkan *display* jam dan mengontrol *speaker*. Permasalahan mendasar dalam perancangan subrutin master pada sistem waktu perkuliahan ini adalah bagaimana mengeset dan mengambil data dari RTC dan menampilkan hasil pengolahan data pada *display seven-segment*. Terutama saat tidak ada komunikasi dengan LabVIEW yaitu Mikrokontroler akan tetap menampilkan jam pada *display* dengan mengambil data jam, menit dan detik dari RTC serta mengatur *on/off* dari indikator (*loud speaker*).

II. LANDASAN TEORI

Dalam mengkomunikasikan antara LabVIEW dengan Mikrokontroler dapat menggunakan komunikasi serial dan paralel, dan dalam sistem ini digunakan komunikasi serial RS232 [1].

Mikrokontroler mempunyai kemampuan untuk bekerja sendiri (*stand-alone*) dengan program yang ada di dalamnya walaupun sedang tidak ada koneksi dengan *device* lain, dalam hal ini adalah LabVIEW 8.5. Mikrokontroler secara otomatis akan mengambil alih tugas LabVIEW yaitu mengatur *on/off* dari indikator *Loud Speaker* [2]. Diagram blok sistem waktu perkuliahan ditunjukkan pada Gambar 1.



Gambar 1. Diagram Blok Sistem Waktu Perkuliahan

A. RTC DS1307

RTC DS1307 adalah serial RTC (*Real Time Clock*) keluaran MAXIM dari Dallas Semiconductor Co. RTC seri ini telah dilengkapi dengan *Binary-Coded Decimal* (BCD) untuk pembacaan kalender maupun jam serta memiliki 56 *byte* NV SRAM (*Non-*

volatile SRAM) sehingga apabila catu daya terputus, data yang terdapat di dalamnya tidak terhapus.[3] Didalam RTC tersebut telah disediakan data berupa detik, menit, jam, hari, tanggal, bulan, serta tahun. RTC DS1307 dapat bekerja dalam dua mode yaitu:

1. Mode *Slave* Penerima (Master Menulis Pada *Slave*)
2. Mode *Slave* Pengirim (Master Membaca Dari *Slave*)

B. Penampil Seven Segment

Penampil *seven-segment* memiliki 7 LEDs dalam satu paket penampil angka. Setiap LED dapat diaktifkan masing-masing untuk menunjukkan *segment* dari suatu *digit*. Secara umum *segment* ditandai dengan huruf a-g. Terdapat 2 tipe penampil *seven-segment*, yaitu *common anode* dan *common cathode*. [4]

III. ANALISIS DAN PERANCANGAN

A. Deskripsi Sistem

Perancangan Subrutin Program Mikrokontroler *Master* memiliki fungsi mengolah data dari RTC pada saat tidak ada koneksi dengan LabVIEW.

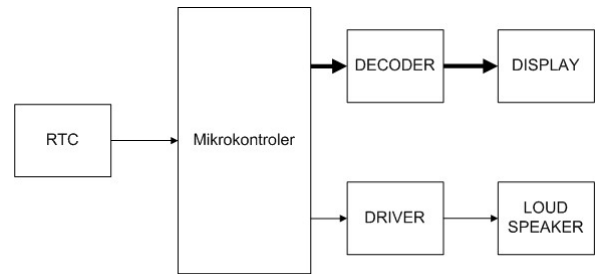
- Minimum Sistem: Minimum sistem yang digunakan adalah Mikrokontroler jenis ATmega8535 sebagai *slave* sekaligus master apabila PC off.
- Input Sistem: LabVIEW 8.5 dan RTC DS1307
- Output Sistem: 4 digit *seven-segment common anoda* 12VDC, 2 buah led 10 mm dan 2 buah *loud speaker*.

B. Rangkaian Pengatur

Rangkaian pengatur disini maksudnya adalah Mikrokontroler yang aktif tanpa ada komunikasi dengan LabVIEW. *Input* dari rangkaian pengatur didapat dari RTC DS1307 yang menyimpan data terakhir dari LabVIEW.

Mikrokontroler melakukan komunikasi dengan RTC menggunakan komunikasi I2C, Mikrokontroler sebagai master sedangkan RTC sebagai *slave*. Mikrokontroler bertugas hanya membaca data dari RTC dan ditampilkan pada *display seven-segment* yang sebelumnya melalui dekoder 74LS247 terlebih dahulu [5].

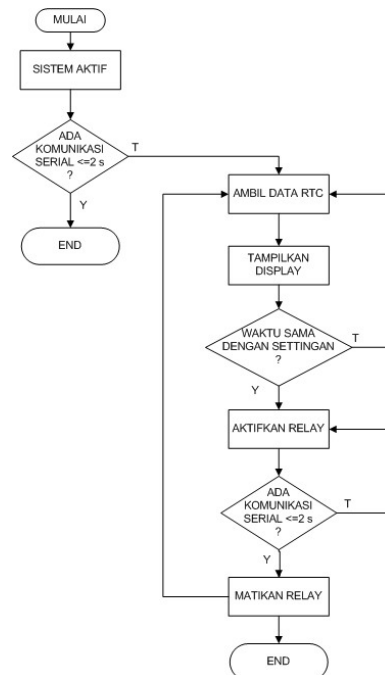
Selain membaca nilai data RTC, Mikrokontroler juga bertugas untuk menghidupkan dan mematikan indikator berupa *Loud Speaker*. Gambar 2 adalah diagram blok dari rangkaian pengatur.



Gambar 2. Diagram Blok Sistem Rangkaian Pengatur

C. Perancangan Algoritma

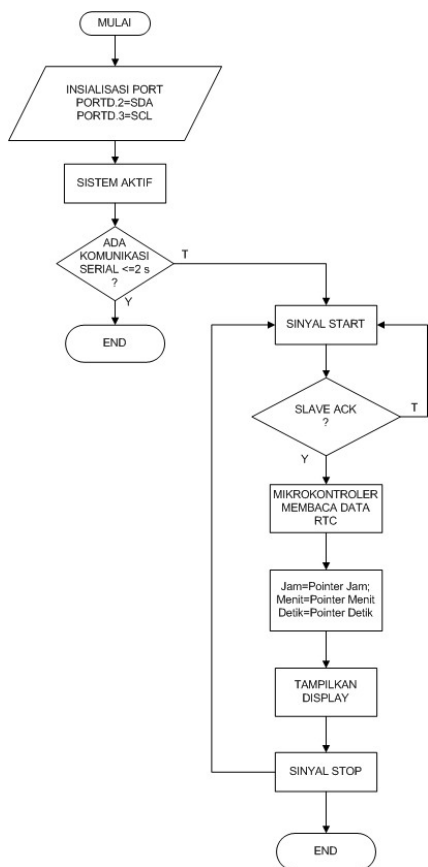
Pada awal sistem diaktifkan, sistem akan memastikan apakah ada komunikasi dengan LabVIEW atau tidak selama 2 detik. Apabila tidak ada maka Mikrokontroler akan mengambil data dari RTC. Data pada RTC merupakan data terakhir yang dikirim oleh LabVIEW kepada Mikrokontroler dan di-set pada RTC. Kemudian Mikrokontroler akan menampilkan data jam dan menit pada *display seven-segment* dan mengecek apakah waktu pada RTC sudah sama dengan waktu untuk membunyikan *loud speaker*[6]. Apabila sudah sama maka Mikrokontroler akan mengaktifkan *driver relay* dc. Namun apabila saat Mikrokontroler sedang mengaktifkan *relay* dan kemudian ada *interrupt* dari LabVIEW, maka Mikrokontroler akan menon-aktifkan *relay* sehingga *loud speaker* tidak aktif. Kondisi itu akan terus berulang hingga sistem dinon-aktifkan. Aktif *relay* hanya pada hari Senin hingga Jumat sedangkan Sabtu dan Minggu *relay* tidak akan aktif. Algoritma dari sub-rutin tampak seperti di Gambar 3.



Gambar 3. Flowchart Subrutin

D. Perancangan Pengambilan Data RTC

Pengambilan data RTC menggunakan komunikasi I2C, Mikrokontroler sebagai *device master* dan RTC sebagai *device slave*. Jalur SDA dan SCL menggunakan 2 bit pada Mikrokontroler yaitu pada PORTD.2 dan PORTD.3. Saat sistem aktif dan tidak ada komunikasi, Mikrokontroler membangkitkan sinyal *start* pada jalur SDA dan RTC membalas dengan memberikan sinyal *acknowledge*. Mikrokontroler membaca nilai dari alamat detik, menit, jam serta hari untuk dibaca dan RTC mengirimkan data-data yang diminta Mikrokontroler tersebut. Alamat-alamat tersebut disimpan oleh RTC dalam variabel jam, menit, detik serta hari. Saat sistem mati dan dihidupkan kembali, Mikrokontroler akan melakukan hal tersebut kembali, sedangkan pada saat sistem mati, RTC akan tetap aktif karena disediakan baterai *backup* 3V sehingga saat Mikrokontroler membaca kembali, maka data waktu yang dikirim RTC tidak di-*set* dari ulang melainkan dilanjutkan dengan data yang terus dihitung oleh RTC. Nilai jam dari RTC langsung ditampilkan pada display tanpa memasukkannya kembali ke dalam variabel lain. Setelah Mikrokontroler selesai membaca data, Mikrokontroler mengirimkan sinyal *stop* untuk mengakhiri pengambilan data. *Flowchart* dari perancangan ini dapat dilihat pada Gambar 4.

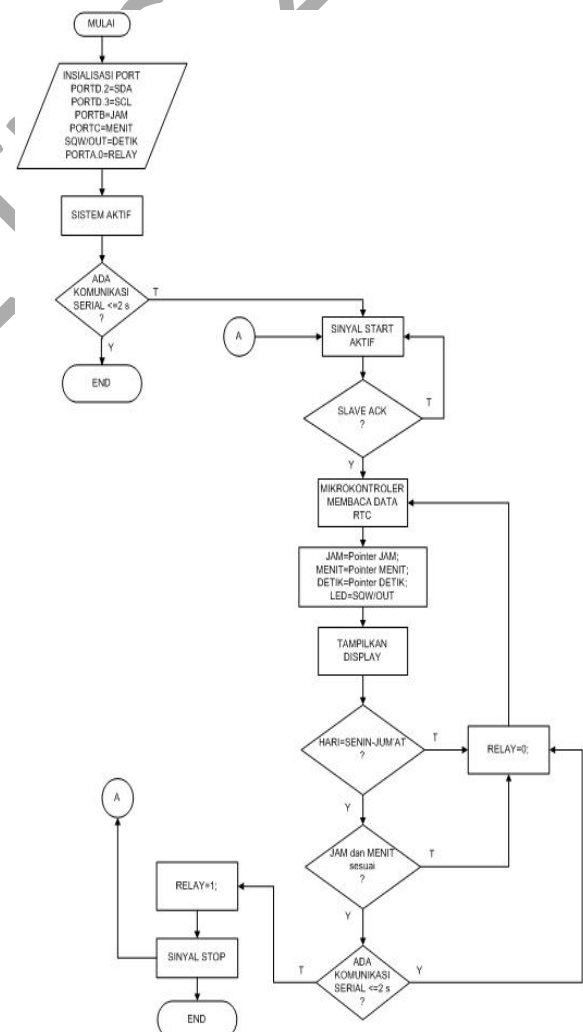


Gambar 4. *Flowchart* Pengambilan Data RTC

E. Perancangan Display dan Indikator

Data pada *display* di dapat dari nilai data jam, menit, detik dan hari dari RTC. Data yang di tampilkan pada *display* hanya nilai jam dan menit, sedangkan untuk detik tidak ditampilkan. Untuk tampilan detik digunakan 2 buah led yang diparalel dan mendapatkan masukan dari RTC DS1307 yang mempunyai SQW/OUT. Pin SQW/OUT dapat diprogram dari Mikrokontroler untuk mengatur keluaran frekuensinya. Dalam sistem ini, pin SQW/OUT di-*set* dengan keluaran 1Hz dengan *duty cycle* 50%.

Untuk indikator, aktif dan tidaknya bergantung dari PORTA.0 yang merupakan *input* data untuk *driver relay* dc. *Driver* aktif apabila menerima logika tinggi serta bergantung dari hari dan jam yang diatur pada program. Selama pengaktifan *relay*, apabila terdapat *interrupt* dari komunikasi LabVIEW, maka secara otomatis *relay* akan *off*. Gambar 5 adalah diagram alur untuk proses penampilan *display* dari Mikrokontroler.



Gambar 5. *Flowchart* Penampil Display dan Indikator

IV. HASIL DAN PEMBAHASAN

Pengujian dilaksanakan untuk menguji dan mengetahui hasil perancangan sistem berjalan sesuai dengan yang diinginkan atau tidak. Apabila terjadi kesalahan atau penyimpangan pada sistem, maka akan dianalisa dan dicarikan solusi untuk mengatasinya sehingga sistem tersebut dapat berjalan sesuai dengan yang diinginkan. Pengujian menitikberatkan pada pembacaan data dari RTC saat tidak ada komunikasi.

A. Tujuan Pengujian

- Mengetahui data setelah tidak ada komunikasi dengan LabVIEW.
- Mengetahui apakah data akan dilanjutkan dengan data terakhir saat tidak ada koneksi dengan LabVIEW.

Target Pengujian: Data terakhir yang diterima oleh Mikrokontroler.

Konfigurasi Pengujian: Konfigurasi pengujian difokuskan pada pembuatan program saja.

```
#include <mega8535.h>
#include <delay.h>

//Subroutine slave
.....
.....

//subroutine master
void master(){
    PORTC=(((kiriman/10)<<4)|(kiriman%10
));
    kiriman++;
    if(kiriman>9){
        kiriman=0;
    }
    delay_ms(500);
}

// Declare your global variables here
void main(void)
{
    // Port C initialization
    // Func7=Out Func6=Out Func5=Out Func4=Out
    Func3=Out Func2=Out Func1=Out Func0=Out
    // State7=0 State6=0 State5=0 State4=0 State3=0
    State2=0 State1=0 State0=0
    PORTC=0x00;
    DDRC=0xFF;

    // USART initialization
    .....
    .....

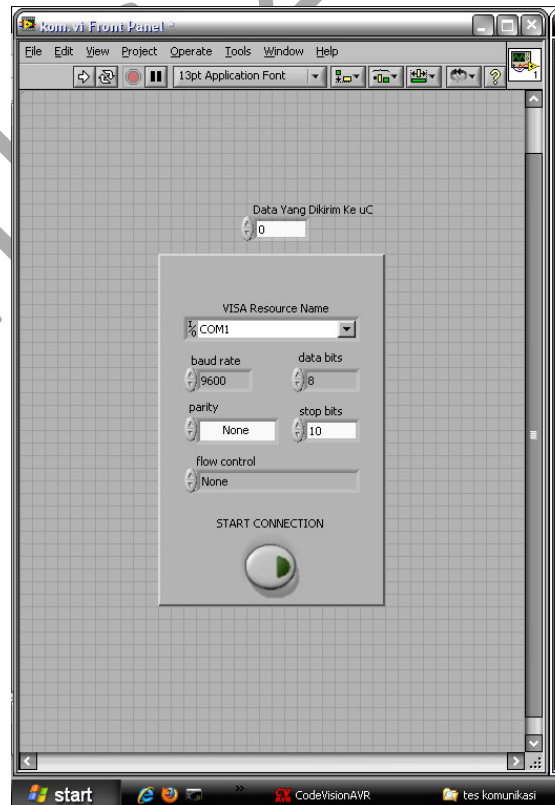
    // Global enable interrupts
    #asm("sei")

    while (1)
    {
        // Place your code here
        master();
    };
}
```



Gambar 6. Tampilan Display Sistem (Kiri: Saat Terkoneksi dengan LabVIEW, Kanan: Saat Stand-Alone)

Gambar 7 menunjukkan data hasil pengujian.



Gambar 7. Tampilan Labview saat Tidak Ada Komunikasi

B. Analisa Data Hasil Pengujian

Pada Gambar 6, terlihat bahwa apabila Mikrokontroler tidak terkoneksi dengan LabVIEW, maka data terakhir yang diterima Mikrokontroler akan dilanjutkan menjadi *up-counter*. Dari hasil pengujian data terakhir yang dikirim oleh LabVIEW adalah 0, dan saat komunikasinya diputus, Mikrokontroler

melanjutkan dengan meng-up-counter menjadi 1 sampai 9.

Selain itu dapat dilihat dari Gambar 7 bahwa *display* sistem saat terkoneksi dengan LabVIEW dan saat *stand-alone* tetap aktif.

V. KESIMPULAN

Berdasarkan tujuan dari perancangan subrutin master pada sistem waktu perkuliahan, dapat disimpulkan bahwa subrutin ini mampu mengeksekusi Mikrokontroler agar dapat bekerja sendiri (*stand-alone*) atau dengan kata lain dapat mengontrol dirinya sendiri. Selain itu Mikrokontroler tersebut tetap dapat mengendalikan agar *display* menampilkan jam dan mengatur aktif atau tidaknya *speaker* walaupun tanpa ada komunikasi dengan LabVIEW. Hal ini dilakukan dengan cara mengambil data yang diperlukan dari *register-register* alamat yang terdapat pada RTC DS1307 menggunakan komunikasi serial I2C.

REFERENSI

- [1] Abidin, Zainal. 2009. *Eksperimen Mikrokontroler*. Surabaya
- [2] Hendi Handian Rachmat dkk. 2011. *Pengembangan Sistem Remote Control untuk Setting Waktu pada Sistem Automatic Time Switch (ATS) Berbasis Real Time Clock (RTC) DS1307 untuk Saklar Lampu*. Jurnal Reayasa © LPPM Itenas | No.1 | Vol. XV Institut Teknologi Nasional Januari – Maret 2011
- [3] Heryanto, M. Ary dan Adi P. Wisnu. 2008. *Pemrograman Bahasa C untuk Mikrokontroler ATMEGA8535*. ANDI: Yogyakarta.
- [4] I Gusti Agung Putu Raka Agung dkk. 2013. *Rancang Bangun Bel Sekolah Otomatis Berbasis Mikrokontroler Avr Atmega8*, Teknologi Elektro, Vol.10 No 2 Juli –Desember 2011.
- [5] Texas Instruments Incorporated, SN5446A, '47A, '48, SN54LS47, 'LS48, 'LS49, SN7446A, '47A, '48, SN74LS47, 'LS48, 'LS49, BCD-TO-SEVENSEGMENT DECODERS/DRIVERS, 1999
- [6] Datasheet DS 1307, <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>, 22 April 2011