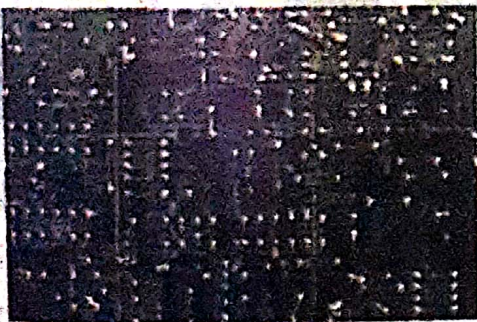




# Sinusoida

Jurnal Penelitian dan Kajian Ilmu Teknik



## Daftar Isi

Hal

- *Analisa Performansi Trafik Inter MSC Untuk Mengatasi High Occupancy Pada Sistem GSM*  
Mufti Gafar dan Aji Bayu ..... 1
- *Analisa Kinerja Antena Horn Pada Acces Point Extend Untuk Jaringan WLAN*  
Heru Abrianto dan Arum R ..... 10
- *Analysis Pemanfaatan UMTS 900 Sebagai Third Carrier Dalam Meningkatkan Kapasitas 3G*  
Edy Supriadi dan M. Iskandar ..... 19
- *Rancang Bangun Sistem Pengontrolan Suhu Dengan Masukan Fungsi Ramp dan Soak Berbasis PLC*  
Rachman Soleman dan Nur Faiz R ..... 29
- *Prototipe Pembersih Asap Rokok Pada Ruang Kerja Berbasis Mikrokontroller ATMEGA 8 Dengan Sensor MQ-2*  
Surya Alimsyah dan Wahyu Crestanto ..... 40
- *Rancang Bangun Sistem Monitoring produksi Menggunakan Mikrokontroller ATMEL AT89S52*  
Irmayani dan Hendra Parluhutan ..... 48
- *Studi Pengaturan Frekuensi Sistem Jawa Bali Dengan Load Frequency Control Menggunakan Software Sinaut Spectrum*  
Sugianto dan M. Ali Randy ..... 61
- *Analisa Performansi Jaringan UMTS Jawa Barat*  
M. Hamdani dan M. Reza. M ..... 76
- *Analisa Penambahan RNC pada Proses Load Balancing Dengan Metode Rehomeing*  
Ariman dan Osman RR ..... 76
- *Analisis Pengaruh Implementasi Refarming UMTS-900 Terhadap Kinerja Jaringan 2G*  
Syamsul El Yumin dan Siti RR ..... 100



# Sinusoida

Jurnal Penelitian Dan Pengkajian Elektro

## DAFTAR ISI

		Halaman
1.	Analisa Performansi Trafik Inter MSC Untuk Mengatasi High Occupancy Pada Sistem GSM <b>Mufl Gafar dan Aji Bayu</b> .....	1 - 9
2.	Analisa Kinerja Antena Horn Pada Access Point Extend Untuk Jaringan WLAN <b>Heru Abrianto dan Arum Rachmitasari</b> .....	10 - 18
3.	Analisis Pemanfaatan UMTS 900 Sebagai Third Carrier Dalam Meningkatkan Kapasitas 3G <b>Edy Supriyadi dan M. Iskandar</b> .....	19 - 28
4.	Rancang Bangun Sistem Pengontrolan Suhu Dengan Masukan Fungsi Ramp dan Soak Berbasis PLC <b>Rachman Soleman dan Nur Faiz</b> .....	29 - 39
5.	Prototipe Pembersih Asap Rokok Pada Ruang Kerja Berbasis Mikrokontroler ATMEGA-8 Dengan Sensor MQ-2 <b>Surya Alimsyah dan Wahyu Crestanto</b> .....	40 - 47
6.	Rancang Bangun Sistem Monitoring Produksi Menggunakan Mikrokontroler ATMEL AT89S52 <b>Irmayani dan Hendra Paruhutan</b> .....	48 - 60
7.	Studi Pengaturan Frekuensi Sistem Jawa Bali Dengan Load Frequency Control (Lfc) Menggunakan Software Sinaut Spectrum <b>Suglanto dan M. All Randy</b> .....	61 - 75
8.	Analisa Performansi Jaringan UMTS Area Jawa Barat <b>Mohammad Hamdani dan M. Reza Mubarrak</b> .....	76 - 85
9.	Analisis Penambahan RNC Pada Proses Load Balancing Dengan Metode Rehomeing <b>Ariman dan Osman Ramadhan Randy</b> .....	86 - 99
10.	Analisis Pengaruh Implementasi Refarming UMTS-900 Terhadap Kinerja Jaringan 2G <b>Syamsul El Yumin dan Siti Romlah Rezani</b> .....	100 - 112

## RANCANG BANGUN SISTEM MONITORING PRODUKSI MENGGUNAKAN MIKROKONTROLER ATMEL AT89S52

Irmayani dan Hendra Parluhutan  
Prodi Teknik Elektro, FTI-ISTN Jagakarsa, Jakarta 12640  
Email : [ir.irmayani@istn.ac.id](mailto:ir.irmayani@istn.ac.id)

### ABSTRAK

*Sistem Monitoring Produksi adalah suatu sistem yang berfungsi untuk memonitoring keadaan produksi lapangan dengan memberikan hasil perhitungan yang lebih akurat. Sistem ini menggunakan PC sebagai interface dengan user dan mikrokontroler yang berfungsi untuk pengambilan data dari sensor yang terpasang pada mesin. Beberapa aspek yang dimonitoring antara lain hasil produksi yang sedang berjalan dan down time mesin. Data-data yang dicatat oleh mikrokontroler disimpan ke dalam database yang dalam hal ini menggunakan Interbase sebagai server database.*

*Dalam penelitian ini akan digunakan standar komunikasi protokol modbus. Protokol Modbus merupakan aturan-aturan komunikasi data dengan teknik Master-Slave. Dalam komunikasi tersebut hanya terdapat satu Master dan satu atau beberapa Slave yang membentuk sebuah jaringan. Master dan slave ini dibentuk dengan menggunakan Mikrokontroler yang menggunakan Max 485 sebagai pembentuk sebuah jaringan RS-485.*

*Hasil pengujian menunjukkan bahwa sistem monitoring produksi yang dibuat menggunakan mikrokontroler dan fungsi 03 protokol modbus sebagai bahasa komunikasi antara master dan slave, berjalan dengan baik. Pengiriman query oleh master dengan Slave ID yang tidak didukung akan menghasilkan time-out error, sedangkan untuk kode fungsi yang tidak didukung sistem akan menghasilkan exception response.*

**Kata Kunci : Protokol Modbus, Master, Slave, RS485, HMI**

### I. PENDAHULUAN

Terinspirasi dengan kemajuan teknologi khususnya dalam bidang elektronika, penulis menerapkan kemajuan teknologi tersebut dalam bidang otomatisasi pemantauan hasil produksi di pabrik karena dalam memproduksi suatu barang/produk ada suatu target produksi yang harus dicapai dalam sebulan.

Dalam sehari-hari setiap pabrik akan mencatat hasil produksi yang dicapai dalam sehari. Data sehari-hari ini akan biasanya dicatat secara manual dengan cara melihat (display) secara langsung ke alat tersebut, baru setelah itu diinput ke komputer untuk diakumulasi akhir bulan apakah dalam setiap bulan tercapai target yang diinginkan. Namun sistem pencatatan secara manual ini banyak kekurangan karena bisa saja ada beberapa hari lupa mencatat jumlah produksi yang dihasilkan mesin.

Solusi dari masalah tersebut maka dibuat suatu alat yang berfungsi untuk melakukan Monitoring Produksi Menggunakan Mikrokontroler ATMEL AT89S52".

Pembahasan meliputi :

1. Komponen Master dan Slave dibentuk dengan menggunakan mikrokontroler AT89S52.
2. Komunikasi antara mikrokontroler ( Master dan Slave ) menggunakan aturan protokol Modbus dasar.

3. Komunikasi antara mikrokontroler (Master dan Slave) menggunakan komunikasi serial RS485 secara *half-duplex*.
4. Antarmuka komunikasi serial RS232 (komputer) menggunakan IC MAX-232 dan RS485 (mikrokontroler) menggunakan IC MAX-485.
5. Sensor yang di pakai menggunakan sensor infrared.

### II. DASAR TEORI

#### 2.1 Sistem Monitoring Produksi

Sistem Monitoring Produksi merupakan suatu sistem yang berfungsi untuk memonitoring hasil produksi setiap mesin. Hal ini dilakukan untuk melihat kualitas, kuantitas dan efisiensi dari mesin produksi tersebut karena bila mesin produksi mempunyai tiga hal ini dalam level yang cukup rendah maka target-target hasil produksi yang ingin dicapai oleh pengusaha produksi bisa tidak tercapai.

Proses memonitoring ini dilakukan dengan mencatat dan menghitung secara manual oleh operator berapa hasil produksi yang dicapai dalam satu hari, setelah itu barulah data-data yang dimonitoring akan dicatat dalam laporan/report hasil produksi. Laporan/report tersebut adalah:

- **NEE (Net Equipment Effectiveness) Report**

NEE Report yang menampilkan laporan mengenai keefektifan produksi yang dihitung melalui berbagai faktor produksi. Besarnya nilai NEE didapat berdasarkan dua faktor dibawah ini:

## Program Studi Teknik Elektro - ISTN

- $UT = \text{Actual output (pcs)} / \text{standard output/hr (pcs)} \quad (2.1)$
- $POT = \text{Total waste (hrs)} + UT \quad (2.2)$

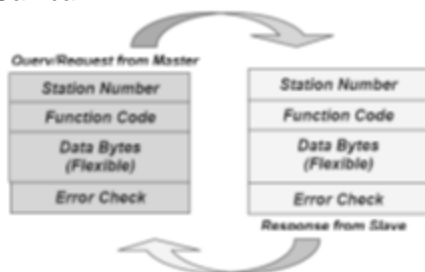
Maka nilai NEE adalah:

- $NEE = UT / POT$
- **Machine Log Event and Alarm Report**  
Machine Log Event and Alarm Report memberikan informasi mengenai kondisi setiap mesin. Kondisi-kondisi yang dimaksud adalah:
  1. Down Time  
Down time terjadi jika mesin berhenti beroperasi karena adanya kerusakan atau disebabkan oleh faktor lainnya.
  2. Production Time (Run Time)  
Production Time merupakan keadaan mesin yang sedang beroperasi dan mengeluarkan *output*.
  3. Cycle Time  
Cycle time adalah perkiraan jumlah produk yang dapat dihasilkan oleh mesin dalam satu jam. *Default* untuk Cycle Time jika dikosongkan adalah 1, yang berarti dalam 1 jam produk yang dihasilkan adalah 1 buah.

## 2.2 Protokol Modbus

Protokol Modbus merupakan aturan-aturan komunikasi data dengan teknik Master-Slave. Dalam komunikasi tersebut hanya terdapat satu Master dan satu atau beberapa Slave yang membentuk sebuah jaringan. Komunikasi Modbus selalu diawali dengan *query* dari Master, dan Slave memberikan respon dengan mengirimkan data atau melakukan aksi sesuai perintah dari Master. Master hanya melakukan satu komunikasi dalam satu waktu. Slave hanya akan melakukan komunikasi jika ada perintah (*query*) dari Master dan tidak bisa melakukan komunikasi dengan Slave yang lain.

Protokol Modbus membentuk sebuah format pesan untuk *query* Master dan respon Slave. Format pesan tersebut membentuk siklus pengiriman *query*-respon seperti yang ditunjukkan pada Gambar 2.1



Gambar 2.1 siklus pengiriman *query*-respon

Pada saat mengirimkan *query* ke Slave, Master menggunakan mode pengalamatan, yaitu:

- **Unicast mode.**  
Master mengirimkan *query* kepada satu Slave. Setelah menerima dan memproses

*query*, Slave akan memberikan jawaban berupa respon kepada Master.

### 2.2.1 Mode Transmisi Serial

Dalam jaringan Modbus terdapat 2 mode transmisi serial, yaitu mode RTU dan mode ASCII.

#### 2.2.1.1 Mode RTU (*Remote Terminal Unit*)

Format masing-masing *byte* (11 bit) dalam mode RTU adalah:

- **Coding system:** 8 bit biner, heksadesimal 0-9,A-F.
- **Bits per byte:**
  - 1 start bit.
  - 8 data bits, *Least Significant Bit* (LSB) dikirim pertama.
  - 1 bit untuk even/odd parity, no bit untuk no parity.
  - 1 stop bit jika menggunakan parity, 2 bits untuk no parity.
- **Error check field:** *Cyclical Redundancy Check* (CRC).

### 2.2.2 Modbus Message Framing

#### 2.2.2.1 RTU Framing

Frame pesan pada mode transmisi RTU ditunjukkan pada Tabel 2.1.

Tabel 2.1 RTU Framing

STA RT	ADDR ESS	FUNCTI ON	DAT A	CRC CHE CK	END
$T_{3,5}$ $\geq 3,5$ CHA R	8 BIT	8 BIT	$n \times 8$ BIT	16 BIT	$T_{3,5}$ $\geq 3,5$ CHA R

Dari Tabel 2.1, pada mode RTU *frame* pesan dipisahkan oleh *silent interval* paling sedikit waktu 3,5 karakter. Interval waktu ini disebut  $T_{3,5}$ . Seluruh karakter dalam *frame* pesan harus ditransmisikan secara bersambung. Interval antar karakter dalam *frame* pesan tidak boleh lebih besar dari waktu 1,5 karakter ( $T_{1,5}$ ). Jika interval antar karakter lebih besar dari  $T_{1,5}$ , maka *frame* pesan tersebut dinyatakan tidak lengkap dan akan diabaikan.

Dari Tabel 2.2 juga dapat dilihat bahwa *frame* pesan terdiri dari *Address Field*, *Function Field*, dan *Data Field*.

- **Address Field**

Masing-masing Slave harus mempunyai alamat yang berbeda dalam range 1 – 247 untuk pengalamatan individual.

- **Function Code**

*Function code* berfungsi untuk memberitahu Slave tentang perintah yang harus dikerjakan dan sebagai indikasi respon normal atau jenis *error* yang terjadi (*exception response*).

## Program Studi Teknik Elektro - ISTN

Pada sistem komunikasi Modbus, jumlah *function code* yang didukung bervariasi tergantung kontroler dan peralatan Slave yang digunakan. Beberapa kode fungsi berikut keterangannya ditunjukkan pada Tabel 2.2.

Tabel 2.2 Kode fungsi Modbus

Kode Fungsi	Fungsi	Aksi yang dikerjakan
1 = 01H	<i>Read Coil Status</i>	Membaca status ON/OFF coil (output digital).
2 = 02H	<i>Read Input Status</i>	Membaca status ON/OFF input digital.
3 = 03H	<i>Read Holding Registers</i>	Membaca nilai "Holding" Register
4 = 04H	<i>Read Input Registers</i>	Membaca nilai input analog.
5 = 05H	<i>Force Single Coil</i>	Mengset status satu coil pada keadaan ON/OFF.
6 = 06H	<i>Preset Single Register</i>	Mengset nilai "Holding" Register
15 = 0FH	<i>Force Multiple Coils</i>	Mengset status beberapa coil pada keadaan ON/OFF.
16 = 10H	<i>Preset Multiple Registers</i>	Mengset nilai beberapa "Holding" Register

- **Data Field**

*Data field* pada *query* berisi kode sebagai informasi tambahan pada *function code* tentang aksi yang harus dikerjakan Slave. Informasi tersebut bisa berupa alamat input-output, jumlah input-output, jumlah *byte* data, atau nilai data pengesetan. Jika tidak terjadi *error*, *data field* pada respon berisi data yang diminta. Sedangkan pada *exception response*, *data field* berisi *exception code*.

- **Error Checking Field**

Dalam *Error Checking Field* terdapat dua cara pengujian error dalam pengiriman sinyal. Untuk pengujian mode ASCII frame pesan menggunakan LRC (*Longitudinal Redundancy Check*) dan untuk pengujian mode RTU frame pesan menggunakan CRC (*Cyclical Redundancy Check*).

- **CRC (Cyclical Redundancy Check)**

Pada mode RTU, *error checking field* berisi sebuah nilai 16 bit (2 *byte*) yang didasarkan pada metode CRC. Prosedur perhitungan CRC adalah :

1. Inisialisasi nilai register 16 bit CRC dengan FFFF hex.
2. Eksklusif OR 8 bit data pesan pertama dengan *low order byte* register CRC, letakkan hasilnya di register CRC.
3. Geser kanan register CRC 1 bit ke arah LSB, dan MSB diisi dengan 0. Nilai LSB register CRC yang tergeser diperiksa.

4. Jika LSB tergeser adalah 0, ulangi langkah 3 (pergeseran yang lain). Jika LSB tergeser 1, eksklusif-OR register CRC dengan nilai A001 hex (1010 0000 0000 0001).
5. Ulangi langkah 3 dan 4 sampai delapan pergeseran. Setelah delapan pergeseran, proses 8 bit data pesan pertama selesai.
6. Ulangi langkah 2 - 5 untuk 8 bit data pesan berikutnya sampai semua data diproses.
7. Nilai akhir register CRC adalah nilai CRC.
8. Pada saat CRC ditempatkan di pesan, nilai CRC *low order byte* dikirimkan terlebih dahulu diikuti *high order byte*.

- **Exception Response**

Pada sebuah *exception response*, Slave mengembalikan kode fungsi dengan MSB (*Most Significant Bit*) diset 1 dan *data field* diisi dengan kode pengecualian (*exception code*). Hal ini dimaksudkan agar Master mengetahui *exception* yang terjadi. Beberapa *exception code* berikut keterangannya ditunjukkan pada Tabel 2.3.

Tabel 2.3 *Exception code* dalam *exception response*

Kode	Nama	Arti
01	<i>ILLEGAL FUNCTION</i>	Kode fungsi yang terdapat dalam <i>query</i> merupakan perintah ( <i>action</i> ) yang tidak diizinkan untuk Slave.
02	<i>ILLEGAL DATA ADDRESS</i>	Alamat data dalam <i>query</i> merupakan alamat yang tidak diizinkan untuk Slave.
03	<i>ILLEGAL DATA VALUE</i>	Nilai dalam <i>data field query</i> merupakan nilai yang tidak diizinkan untuk Slave.

## 2.3 Komunikasi Serial

Komunikasi serial adalah pengiriman data satu persatu secara berurutan dalam satu waktu. Oleh karena itu, komunikasi serial lebih lambat dari pada komunikasi paralel. Komunikasi serial dibagi menjadi dua macam, yakni komunikasi serial *sinkron* dan komunikasi serial *asinkron*. Komunikasi serial *sinkron* merupakan pengiriman data dengan mengikutsertakan sinyal *clock*, sedangkan komunikasi serial *asinkron* adalah pengiriman data tanpa disertai dengan sinyal *clock*.

### 2.3.1 RS-232

Standar sinyal serial RS232 juga memiliki ketentuan *level* tegangan sebagai berikut:

1. Logika '1' disebut mark terletak antara -3 volt hingga -25 volt.
2. Logika '0' disebut space terletak antara +3 volt hingga +25 volt.

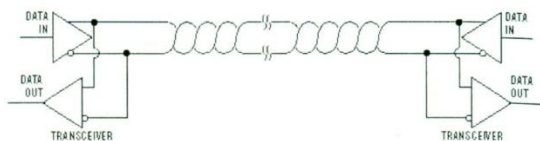
Program Studi Teknik Elektro - ISTN

3. Daerah tegangan antara -3 volt hingga + 3 volt adalah *invaled level*, yaitu daerah tegangan yang tidak memiliki *level* logika pasti sehingga harus dihindari. Demikian juga, *level* tegangan lebih negatif dari -25 volt. atau lebih positif dari + 25 volt juga menghindari karena tegangan tersebut dapat merusak *line* pengendali pada saluran RS232.

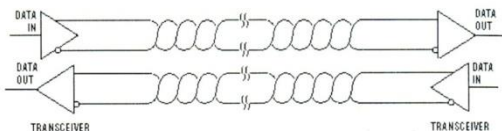
**2.3.2 RS-485**

Berbeda dengan standard RS-232 yang menggunakan ground bersama (*common ground*), RS-485 menggunakan 2 buah kabel untuk menyalurkan sinyal data dan kebalikannya dengan tidak menggunakan ground bersama. Sistem ini dinamakan dengan sinyal differensial atau *balanced*. Sementara RS-232 atau sistem ground bersama dinamakan sinyal *unbalanced* atau *single-ended*.

Standar RS-485 mendukung komunikasi dua arah bergantian (*half duplex*) dan dua arah penuh (*full duplex*). Sistem *half duplex* (gambar 2.3) memerlukan 2 buah kabel (untuk sinyal A dan sinyal B) dimana sinyal A dan B untuk *driver* atau penerima diparalel. Sedangkan sistem *full duplex* (gambar 2.4) memerlukan 4 kabel, masing-masing 2 untuk sinyal A dan sinyal B, driver dan penerima terpisah.

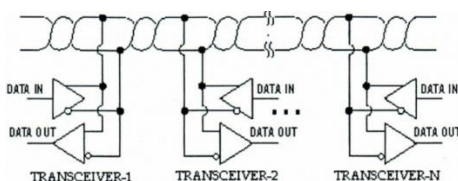


Gambar 2.3 Sistem Half Duplex RS-485



Gambar 2.4 Sistem Full Duplex RS-485

Dalam sistem jaringan seperti gambar 2.5, pada saat jaringan dalam keadaan *idle* hanya 1 pengirim dari sebuah transceiver yang terhubung ke jaringan, sedangkan transceiver yang lain terhubung ke jaringan melalui penerimanya. Pin pengendali pengirim dan penerima yang akan mengatur hal ini. Transceiver yang terhubung pengirim dinamakan *Master*, sedangkan transceiver-transceiver lain dinamakan *slave*.



Gambar 2.5 Jaringan RS-485 dengan system 2 kabel

**2.4 Borland Delphi 7.0**

Borland Delphi adalah software aplikasi pemrograman yang menggunakan bahasa pemrograman pascal.

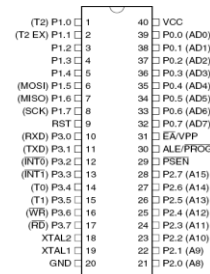
Borland Delphi memiliki banyak fasilitas untuk pemrograman misalnya fasilitas source completion, tool database dekstop, xml converter, Microsoft Office Component Connector, Corba, BDE dan masih banyak lagi.

**2.5 Mikrokontroler**

AT89S52 adalah mikrokontroler keluarga MCS-51 keluaran Intel. Mikrokontroler ini berdaya rendah namun berkemampuan tinggi dengan CPU (*Central Processing Unit*) kapasitas 8-bit.

**2.5.2 Konfigurasi PIN pada IC AT89S52**

Mikrokontroler AT89S52 memiliki 40 pin dengan 32 pin diantaranya digunakan sebagai *port* paralel. Satu *port* paralel terdiri dari 8 pin, sehingga jumlah *port* pada mikrokontroler AT89S52 adalah 4 *port*, yaitu *port* 0, *port* 1, *port* 2 dan *port* 3.



Gambar 2.6 Konfigurasi pin AT89S52

**2.5.4 Register Mikrokontroler AT89S52**

Adapun register-register tersebut adalah sebagai berikut :

1. Register A (Accumulator), yaitu register 8 bit yang berfungsi sebagai akumulator yang digunakan untuk operasi aritmetika atau logika dan berfungsi untuk menyimpan hasil operasi tersebut.
2. Register B, yaitu register 8 bit yang digunakan bersama dengan register A untuk operasi perkalian dan pembagian.
3. Register penyimpan program status (Program Status Word: PSW), yaitu register 8 bit yang berfungsi sebagai penanda status (status flag) sebagai akibat dari pelaksanaan suatu instruksi. Register ini meliputi bit-bit : carry (CY), auxiliary (AC), Zero Flag (F0), pemilih register bank (RS0 dan rs1), over flow (OV), dan parity flag (P). Flag CY, AC dan OV merupakan keluaran dari suatu proses aritmetika yang dilakukan di akumulator. Flag P merupakan parity dari isi register akumulator. Flag C digunakan juga sebagai akumulator operasi bit. RS0 dan RS1 digunakan untuk memilih register bank yang aktif pada saat itu.
4. Register penunjuk data (Data Pointer: DPTR), yaitu register 16 bit yang berfungsi untuk mengakses lokasi kode alamat dan lokasi alamat eksternal. DPTR terdiri dari 2 buah

Program Studi Teknik Elektro - ISTN

register, yaitu DPL byte rendah dan DPH byte tinggi.

5. Register pengontrol prioritas interupsi (Interrupt Priority: IP), yaitu register 8 bit yang berfungsi menentukan prioritas interupsi sesuai yang diinginkan. Jika salah satu bit pengatur prioritas interupsi diberi masukkan sinyal logika tinggi maka pena interupsi tersebut memiliki interupsi dengan prioritas tinggi.
6. Register pengatur interupsi (Interrupt Enable: IE), yaitu register 8-bit yang berfungsi sebagai pengatur interupsi.
7. Register pengatur ragam kerja pewaktu / pencacah (Timer / Counter Mode Register : TMOD), yaitu register 8 bit yang berfungsi untuk memilih mode waktu atau pencacah yang dilakukan mikrokontroler.
8. Register pengatur kerja pewaktu/pencacah (Timer/Counter: TCON), yaitu register 8-bit yang berfungsi untuk mengatur kerja pewaktu / pencacah.
9. Register pengatur komunikasi serial (Serial Port Control: SCON), yaitu register 8 bit yang berfungsi untuk mengatur ragam penerimaan atau pengiriman data serial melalui terminal serial.
10. Register penampung data serial (Serial Buffer: SBUF), yaitu register yang berfungsi sebagai penampungan sementara dari penerimaan atau pengiriman data serial dari atau ke terminal data serial.
11. Register pencacah program (Program Counter: PCON), yaitu register 16 bit yang berisi alamat dari instruksi berikutnya yang akan dilaksanakan oleh CPU.

**2.5.6 Baud rate**

Baud rate dalam system komunikasi serial tidak dapat dilepaskan fungsi dan pemahamannya, karena dengan Baud rate jumlah bit data serial yang dikirim dapat diketahui secara teoritis guna mengetahui validitas dan keakuratan data terkirim. Pada Port serial mikrokontroler , Baud rate dapat diatur pada Mode 1 dan Mode 3 , namun pada Mode 0 dan Mode 2, baud rate tersebut mempunyai kecepatan yang permanen yaitu untuk mode 0 adalah 1 / 12 dari frekwensi oscillator dan mode 2 adalah 1 / 64 dari frekwensi osilator.

Perhitungan guna mengatur Baud rate pada port serial dalam system komunikasi ini menggunakan rumus :

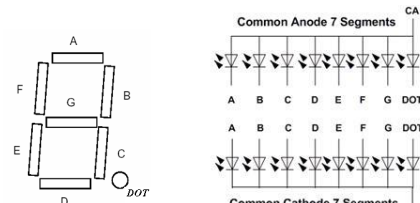
$$\text{Baud rate} = \frac{F_{\text{Oscillator}}}{12x(256 - TH1)x32} \quad (\text{SMOD} = 1)$$

Dalam pengiriman dan penerimaan serial memerlukan waktu. Besarnya waktu tersebut dapat dihitung dengan rumus:

$$T_{\text{serial}} = T_{\text{serial}} \times \frac{\text{Jumlah bit per karakter}}{\text{Baud Rate}}$$

**2.8 Display Seven Segmen**

Seven Segment adalah suatu segmen- segmen yang digunakan menampilkan angka. Seven segmen ini tersusun atas 7 batang LED yang disusun membentuk angka 8 dengan menggunakan huruf a-f yang disebut DOT MATRIKS. Setiap segmen ini terdiri dari 1 atau 2 Light Emitting Diode ( LED ).

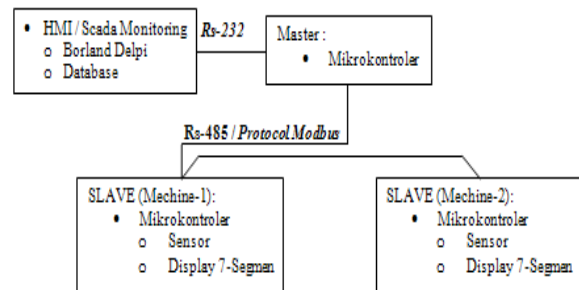


Gambar 2.7 Bagan Seven Segmen

**III. PERENCANAAN dan REALISASI**

**3.1 Perancangan Sistem**

Perancangan sistem monitoring produksi sesuai dengan gambar 3.1. Dari setiap blok ini mempunyai fungsi masing-masing. HMI (*Human Machine Interface*) digunakan sebagai antarmuka dalam proses *monitoring*. Master berfungsi sebagai pemberi data kepada HMI dan pengambil data dari setiap slave sedangkan Slave berfungsi sebagai counter hasil produk dan memberikan data hasil produk kepada Master. Komunikasi master dan slave ini menggunakan standar Protokol Modbus mode RTU.



Gambar 3.1 Diagram blok sistem monitoring produksi

Cara kerja sistem ini adalah:

1. Setelah sistem dalam keadaan ON, master akan mengirim data awal pertama ke HMI selama dua detik.
2. Kemudian setelah dua detik master akan mengambil data dari slave-1 dengan menggunakan standar komunikasi protokol modbus.
3. Bila selama proses pengambilan data tidak ada respon dari slave-1 maka master akan langsung menghentikan proses pengambilan data lalu memulai pengiriman data ke HMI selama dua detik. Bila mendapatkan respon dari slave-1, master akan mengecek semua data yang diterima dari slave-1 lalu

Program Studi Teknik Elektro - ISTN

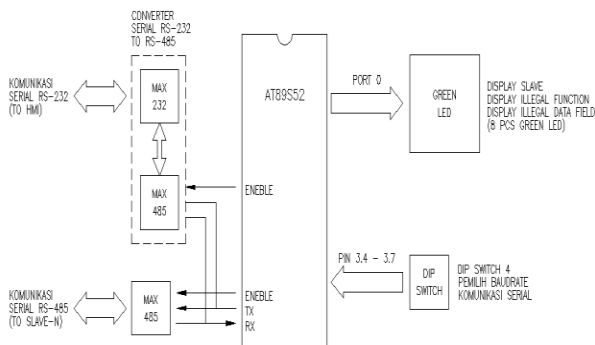
menyimpan data tersebut kedalam memori sementara sebelum dikirim ke HMI. (Sesuai dengan aturan dalam protokol modbus).

4. Setelah pengambilan data dari slave-1, master akan mengirim data kembali ke HMI selama dua detik.
5. Kemudian setelah dua detik, master akan mengambil data dari slave berikutnya yaitu slave-2.
6. Bila selama proses pengambilan data tidak ada respon dari slave-2 maka master akan langsung menghentikan proses pengambilan data lalu memulai pengiriman data ke HMI selama dua detik. Bila mendapatkan respon dari slave-2, master akan mengecek semua data yang diterima dari slave-1 lalu menyimpan data tersebut kedalam memori sementara sebelum dikirim ke HMI. (Sesuai dengan aturan dalam protokol modbus).
7. Setelah pengambilan data dari slave-2, master akan mengirim data kembali ke HMI selama dua detik. Begitu seterusnya.

**3.2 Perancangan Perangkat Keras**

**3.2.1 Rangkaian Master**

Perancangan Rangkaian Master sistem monitoring produksi ini diperlihatkan blok diagram pada Gambar 3.2.

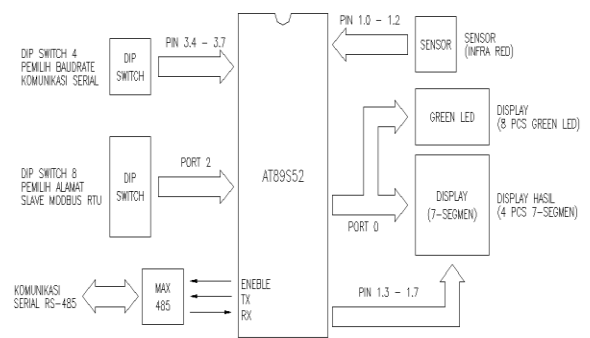


Gambar 3.2 Blok Diagram Master

Gambar 3.2 rangkaian master terdiri atas rangkaian utama sebagai pengolah data, Konverter RS232 to RS485, Komukasi Serial RS-485 ke Slave-n, Display Led yang mengindikasikan Slave, *illegal function*, *illegal data field* dan pemilihan Baudrate Komunikasi Serial.

**3.2.2 Rangkaian Slave**

Perancangan Rangkaian Slave sistem monitoring produksi ini diperlihatkan blok diagram pada Gambar 3.3.



Gambar 3.3 Blok Diagram Slave

Berdasarkan gambar 3.3 rangkaian slave terdiri atas rangkaian utama sebagai pengolah data, Komukasi Serial RS-485 ke Slave-n, 4 Pcs Seven Segment untuk Display hasil counter produk (alternatif indikasi data 8 Pcs Led), sensor infrared sebagai counter produk, pemilihan alamat slave, dan pemilihan Baudrate Komunikasi Serial.

**3.3 Perancangan Protokol Modbus**

Protokol Modbus merupakan aturan komunikasi dalam transmisi data antara Master dan Slave. Fungsi dasar Modbus memiliki beberapa fungsi yaitu fungsi 05, 06, 15, dan 16 sebagai fungsi pengesetan dan ditambahkan fungsi 01, 02, 03, dan 04 sebagai monitoring proses pengendalian. Fungsi dasar Modbus yang akan diaplikasikan dalam sistem PMS Berbasis Protokol Modbus ini fungsi 03.

Dalam perancangan protokol modbus, Slave sistem PMS berbasis Protokol Modbus ini memiliki beberapa register data untuk penempatan Data-data yang dibutuhkan dalam pengesetan dan monitoring proses pengendalian.

Register data yang akan digunakan dalam sistem PMS ini,

- Address 0000 = untuk data High Actual Produksi
- Address 0001 = untuk data Low Actual Produksi

Dalam sistem monitoring produksi berbasis Protokol Modbus ini digunakan protokol Modbus mode RTU dan cek *error CRC (Cyclical Redundancy Check)*.

**3.4 Perancangan Perangkat Lunak**

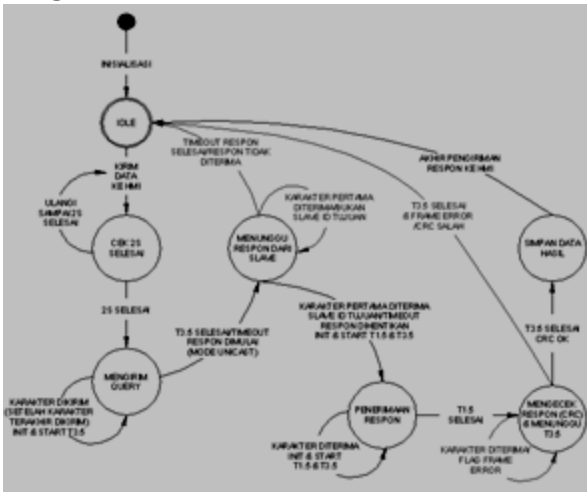
Program dalam sistem produk monitoring system dibagi menjadi 2 yaitu program Master dan Slave pada mikrokontroler AT89S52 dan program HMI (*Human Machine Interface*) pada komputer. Program pada mikrokontroler dibuat menggunakan bahasa *assembler*, sedangkan program di dalam komputer menggunakan Borland Delphi 7.0.

**3.4.1 Program Master**

Perancangan pembuatan program Master dilakukan dengan menggunakan diagram keadaan (*state diagram*) seperti terlihat pada Gambar 3.4.

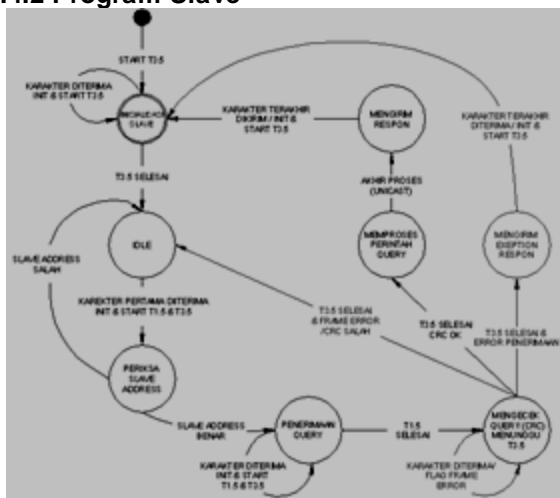


Program Studi Teknik Elektro - ISTN



Gambar 3.4 Diagram keadaan program Master

3.4.2 Program Slave



Gambar 3.5 Diagram keadaan program Slave

3.4.3 Program pada Komputer

Program pada komputer dibangun dengan menggunakan bahasa pemrograman Borland Delphi 7.0. Program tersebut membentuk sebuah HMI (*Human Machine Interface*). HMI berfungsi untuk memberikan perintah kepada Master untuk melakukan komunikasi dengan Slave.



Gambar 3.6 Diagram keadaan HMI / Scada Monitoring

3.4.3.1 HMI / Scada Monitoring

Dalam perancangan HMI / Scada Monitoring akan terdiri dari beberapa form. Masing-masing form tersebut adalah:

- **Form Scada Monitoring**

Form ini merupakan form utama yang berfungsi untuk menampilkan data-data yang dimonitoring. Beberapa aspek yang dimonitoring antara lain hasil produksi yang sedang berjalan, dan *down time* mesin. Data-data yang dicatat oleh mikrokontroler disimpan ke dalam database yang dalam hal ini menggunakan Interbase sebagai server database.

IV. PENGUJIAN dan ANALISA DATA

4.1 Pengujian Rangkaian Master

Pengujian yang dilakukan pada rangkaian master ini ada dua tahap yaitu pengukuran dan analisa level tegangan komponen utama yang terdapat pada rangkaian master ini dan komunikasi serial antara master dengan PC.

4.1.1 Pengukuran dan Analisa Level Tegangan

- **Mikrokontroler**

Pengukuran level tegangan pada setiap port yang ada pada mikrokontroler dengan menggunakan multimeter digital.

- o Port 0

Tabel 4.1 Hasil pengukuran Port 0

Port	Hasil Pengukuran	Logik
0.0	3.22 V	1
0.1	3.22 V	1
0.2	3.23 V	1
0.3	3.21 V	1
0.4	3.22 V	1
0.5	3.22 V	1
0.6	3.22 V	1
0.7	3.21 V	1

Berdasarkan tabel 4.1 logik Port 0 menunjukan logik 1, hal ini disebabkan mikrokontroler pada port 0 diprogram logik 1 untuk membuat LED dalam keadaan off.

- o Port 1

Tabel 4.2 Hasil pengukuran Port 1

Port	Hasil Pengukuran	Logik
1.0	4.31 V	1
1.1	4.31 V	1
1.2	4.31 V	1
1.3	4.31 V	1
1.4	4.31 V	1
1.5	4.31 V	1
1.6	4.31 V	1
1.7	4.31 V	1

Berdasarkan tabel 4.2 logik Port 1 menunjukan logik 1, hal ini disebabkan mikrokontroler pada port 1 logik 1 karena tidak digunakan untuk apa-apa (default).

## Program Studi Teknik Elektro - ISTN

## ○ Port 2

Tabel 4.3 Hasil pengukuran Port 2

Port	Hasil Pengukuran	Logik
2.0	0 V	0
2.1	0 V	0
2.2	4.35 V	1
2.3	4.35 V	1
2.4	4.35 V	1
2.5	0 V	0
2.6	4.31 V	1
2.7	4.31 V	1

Berdasarkan tabel 4.3 logik Port 2 menunjukkan logik 0 dan logik 1, hal ini disebabkan mikrokontroler pada port 2 diprogram logik 1 dan logik 0. Untuk P2.0 dan P2.1 diprogram logik 0 karena untuk mengatur Max 485 (U4 gambar 3.5) mode penerimaan, untuk P2.2 dan P2.3 diprogram logik 1 karena mengatur Max 485 (U5 gambar 3.5) mode pengiriman, untuk P2.4 dan P2.5 diprogram logik 1 dan 0 karena mengatur Max 485 dalam keadaan *idle*, untuk P2.6 dan P2.7 logik 1 karena tidak digunakan untuk apa-apa (default).

## ○ Port 3

Tabel 4.4 Hasil pengukuran Port 3

Port	Hasil Pengukuran	Logik
3.0	4.35 V	1
3.1	4.35 V	1
3.2	4.35 V	1
3.3	4.35 V	1
3.4	0 V	0
3.5	0 V	0
3.6	0 V	0
3.7	0 V	0

Berdasarkan tabel 4.4 logik Port 3 menunjukkan logik 0 dan logik 1, hal ini disebabkan mikrokontroler pada port 3 diprogram untuk komunikasi serial untuk P3.0 dan P3.1, untuk P3.2 dan P3.3 logik 1 karena tidak digunakan untuk apa-apa (default), untuk P3.4 – P3.5 logik 0 karena digunakan untuk menyetting baudrate komunikasi serial 9600bps.

• **Max-232**

Pengukuran level tegangan pada setiap pin IC Max 232 dengan menggunakan multimeter digital.

Tabel 4.5 Hasil pengukuran IC Max 232

No.Pin	Hasil Pengukuran	Keterangan
1	6.4 V	C1+
2	8.4 V	V+
3	2.2 V	C1-
4	4.3 V	C2+
5	-3.8 V	C2-
6	-8 V	V-
7	-8 V	T2Out
8	0 V	R2In
9	4.5 V	R2Out
10	0 V	T2In
11	0 V	T1In

Berdasarkan data Tabel 4.5 dapat disimpulkan bahwa IC Max 232 berfungsi baik. Hal ini disebabkan karena level tagangan yang diukur sesuai karakteristik level tegangan pada datasheet.

• **Max-485**

Pengukuran level tegangan pada setiap pin IC Max 485 dengan menggunakan multimeter digital. Setiap pin IC Max 485 ini juga terhubung pada port mikrokontroler.

Tabel 4.6 Hasil pengukuran IC Max 485

No.Pin	Hasil Pengukuran			Keterangan
	U3	U4	U5	
1	3.67 V	4.45 V	4.36 V	R
2	0 V	4.45 V	4.36 V	RE
3	0 V	4.45 V	0 V	DE
4	4.6 V	3.58 V	3.58 V	D
5	0 V	0 V	0 V	Gnd
6	3.36 V	3.36 V	1.80 V	A
7	0.85 V	0.85 V	1.74 V	B
8	4.6 V	4.6 V	4.6 V	Vcc

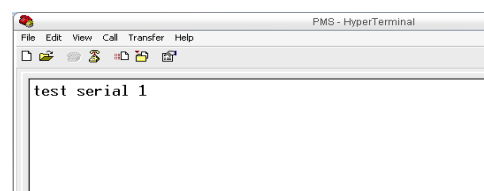
Berdasarkan data Tabel 4.6 dapat disimpulkan bahwa IC Max 485 berfungsi baik. Perbedaan Level tegangan pada setiap IC Max 485 (U3, U4, dan U5) disebabkan karena setiap IC Max 485 diprogram dengan mode yang berbeda-beda. Misal untuk U3 pada mode penerimaan, U4 pada mode pengiriman dan U5 pada keadaan *idle*/mengambang.

## 4.1.2 Komunikasi Serial dengan PC

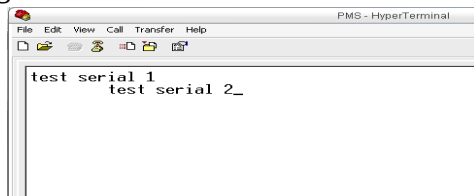
Pengujian rangkaian dilakukan dengan menghubungkan mikrokontroler ke rangkaian RS232, rangkaian RS 232 ke RS485, *keyboard* dan kemudian dihubungkan ke *PC*. Selanjutnya membuat program pada Bahasa assambler dan melihat pada *hyperterminal*. Program yang diberikan pada mikrokontroler.

Berdasarkan program yang diberikan didapatkan hasil pengujian sebagai berikut :

- Jika diketik 'test serial 1' pada *keyboard* maka akan keluar tampilan sebagai berikut :

Gambar 4.1 Pada *Hyperterminal* 1

- Jika diketik 'test serial 2' pada *keyboard* maka akan keluar tampilan sebagai berikut :

Gambar 4.2 Pada *Hyperterminal* 2

Dari tampilan Gambar 4.1 dan Gambar 4.2 maka dapat dilihat bahwa komunikasi serial dapat dilakukan antara mikrokontroler dan PC.

## 4.2 Pengujian Rangkaian Slave

Pengujian yang dilakukan pada rangkaian Slave ini ada tiga tahap yaitu pengukuran dan analisa level tegangan, pengujian hardware dan pengujian komunikasi serial.

### 4.2.1 Pengukuran dan Analisa Level Tegangan

Pengukuran dan analisa level tegangan pada komponen utama yaitu Mikrokontroler dan Max 485 dilakukan setelah program *assambler* untuk slave telah didownload pada mikrokontroler.

- **Mikrokontroler**

Pengukuran level tegangan pada setiap port yang ada pada mikrokontroler dengan menggunakan multimeter digital.

- Port 0

Tabel 4.7 Hasil pengukuran Port 0

Port	Hasil Pengukuran	Logik
0.0	2.36 V	1
0.1	2.80 V	1
0.2	2.36 V	1
0.3	2.36 V	1
0.4	2.36 V	1
0.5	2.80 V	1
0.6	2.36 V	1
0.7	2.36 V	1

Berdasarkan tabel 4.7 logik Port 0 menunjukkan logik 1, hal ini disebabkan mikrokontroler pada port 0 diprogram logik 1.

- Port 1

Tabel 4.8 Hasil pengukuran Port 1

Port	Hasil Pengukuran	Logik
1.0	0.19 V	1
1.1	4.37 V	1
1.2	4.37 V	1
1.3	4.37 V	1
1.4	3.36 V	1
1.5	3.36 V	1
1.6	3.36 V	1
1.7	3.36 V	1

Berdasarkan tabel 4.8 logik Port 1 menunjukkan logik 0 dan 1, hal ini disebabkan mikrokontroler pada port 1 diprogram berbeda-beda. Pada P1.0 diset untuk input data dari sensor, P1.1 – P1.3 diset default, dan P1.4 – P1.7 untuk

register geser untuk menghidupkan seven segmen.

- Port 2

Tabel 4.9 Hasil pengukuran Port 2

Port	Hasil Pengukuran	Logik
2.0	0 V	0
2.1	0 V	0
2.2	0 V	0
2.3	0 V	0
2.4	0 V	0
2.5	0 V	0
2.6	0 V	0
2.7	0 V	0

Berdasarkan tabel 4.9 logik Port 2 menunjukkan logik 0, hal ini disebabkan karena port 2 digunakan untuk setting alamat slave. Dalam pengukuran ini slave-0.

- Port 3

Tabel 4.10 Hasil pengukuran Port 3

Port	Hasil Pengukuran	Logik
3.0	4.35 V	1
3.1	0 V	1
3.2	0 V	1
3.3	0 V	1
3.4	0 V	0
3.5	0 V	0
3.6	0 V	0
3.7	0 V	0

Berdasarkan tabel 4.10 logik Port 3 menunjukkan logik 0 dan logik 1, hal ini disebabkan mikrokontroler pada port 3 diprogram untuk komunikasi serial untuk P3.0 dan P3.1, untuk P3.2 dan P3.3 logik 0 karena digunakan untuk mengatur IC Max 485 pada mode penerima, untuk P3.4 – P3.5 logik 0 karena digunakan untuk menyetting baudrate komunikasi serial 9600bps.

- **Max 485**

Pengukuran level tegangan pada setiap pin IC Max 485 dengan menggunakan multimeter digital. Setiap pin IC Max 485 ini juga terhubung pada port mikrokontroler.

Tabel 4.11 Hasil pengukuran IC Max 485

No.Pin	Hasil Pengukuran	Keterangan
1	4.24 V	R
2	0 V	RE
3	0 V	DE
4	4.24 V	D
5	0 V	Gnd
6	0.31 V	A
7	0.31 V	B
8	4.6 V	Vcc

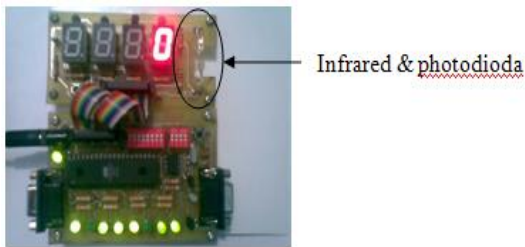
Berdasarkan data Tabel 4.11 dapat disimpulkan bahwa IC Max 485 berfungsi baik. IC Max 485 ini diprogram untuk mode penerima, hal ini dapat dilihat pada pin 2 dan pin 3 yang diberikan level tegangan 0 V.

**4.2.2 Pengujian Hardware**

• **Seven segmen dan sensor**

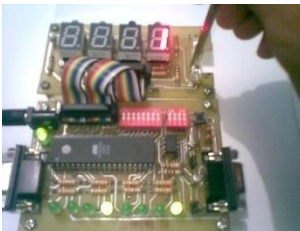
Tahap ini dengan menerapkan program sederhana untuk mencoba fungsi infrared dan photodiode lalu mendisplaykan ke seven segment. Tahap kedua dengan menerapkan program sederhana untuk melihat komunikasi antara master dan slave. Pengujian tahap pertama dilakukan dengan menghalangi infrared dan photodiode, bila sudah terlepas maka display akan menunjukkan angka 1 dan bila diulangi maka display akan menunjukkan angka 2, begitu seterusnya.

• **Angka 0**



Gambar 4.3 Rangkaian slave menunjukkan 0

• **Angka 1**

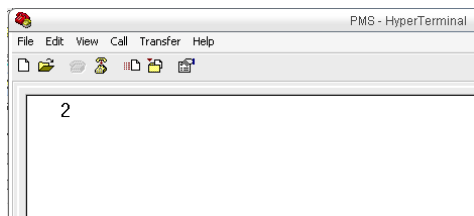


Gambar 4.4 Rangkaian slave menunjukkan 1

**4.2.3 Pengujian Komunikasi Serial**

Untuk pengujian tahap ini yaitu slave akan mengirim hasil yang didisplaykan di slave menuju ke master lalu master akan mengirimkan hasilnya ke *hyperterminal*. Bila komunikasi ok maka *hyperterminal* akan menunjukkan angka yang sama dengan angka yang ditunjukkan di slave. Pada Master dan slave akan diberikan program .

• **Pada Hyperterminal**

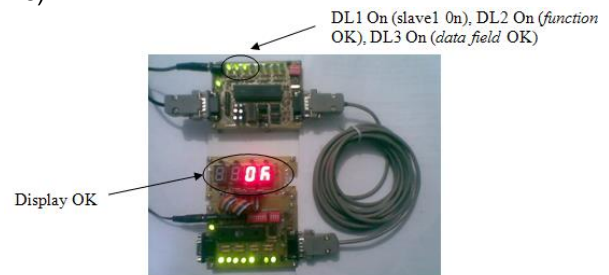


Gambar 4.5 *Hyperterminal* menunjukkan 2

Dari Gambar 4.5 dan gambar 4.6 terlihat bahwa komunikasi antara master dan slave berfungsi dengan baik.

**4.3 Pengujian komunikasi dengan Protokol Modbus**

Pengujian yang dilakukan dalam menguji komunikasi antara master dan slave menggunakan standar komunikasi Protokol Modbus. Pada masing-masing master dan slave akan diberikan program *assambler* protokol modbus (dapat dilihat pada lampiran listing program mikrokontroler nomor 5). Setelah diberikan program tersebut, kedua-duanya juga akan diberikan program *assambler* pengecekan *function* dan *data field*. Bila komunikasi "OK" maka display led (DL2, DL3, DL6 dan DL7) yang ada pada Master akan terus menyala sedangkan pada rangkaian slave akan mendisplaykan OK (hal ini terlihat pada gambar 4.6).



Gambar 4.6 Komunikasi antara master dan slave OK

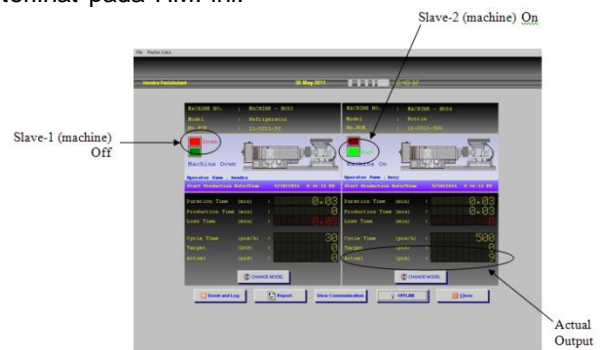
**4.4 Pengujian HMI / Scada Monitoring**

Pengujian HMI / Scada Monitoring ada dua tahap yang dilakukan yaitu pengujian fungsi-fungsi tombol yang ada pada form Scada Monitoring dan pengujian penerimaan data dari master.

**4.4.1 Pengujian Komunikasi HMI / Scada Monitoring dengan Master**

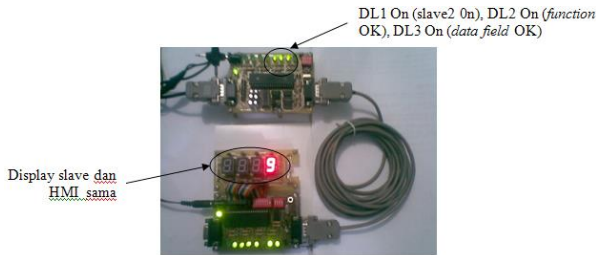
Pengujian yang dilakukan dalam menguji komunikasi antara master dan HMI. Bila master telah mengambil data dari slave (misal pada pengujian dilakukan di slave-2) maka data tersebut akan dikirim ke HMI sehingga apa yang didisplaykan di slave akan sama dengan yang didisplaykan HMI (actual).

Bila master tidak mendapatkan data dari slave (misal pada pengujian slave-1 dalam keadaan off) maka data 0 akan dikirim ke HMI. Indikasi on-off nya sebuah mesin juga dapat terlihat pada HMI ini.



Gambar 4.7 Tampilan pada HMI

**DAFTAR PUSTAKA**



Gambar 4.8 Tampilan pada Slave

Dari serangkaian pengujian yang telah dilakukan pada master, slave, dan HMI / Scada Monitoring. Dapat dilihat bahwa semua tahap pengujian yang telah dilakukan, HMI ini berjalan dengan baik. Pada tahap pertama, semua fungsi-fungsi tombol berfungsi dengan baik. Pada tahap kedua, komunikasi antara master dan slave yang menggunakan standar komunikasi Protokol Modbus berjalan dengan baik, indikasinya dapat dilihat pada 3 buah led yang ada pada rangkaian master yang On. Komunikasi antara master dan HMI dapat berjalan dengan baik, indikasinya dapat dilihat bahwa HMI akan menampilkan actual produk apa yang ditampilkan di slave.

**V. KESIMPULAN**

Berdasarkan hasil pengujian dan analisa yang dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Rangkaian Master dan slave berfungsi dengan baik.
2. Sistem Monitoring Produksi yang dibuat mendukung fungsi protokol Modbus 03, hasil pengujian menunjukkan fungsi 03 dapat berjalan dengan baik.
3. Pengiriman data *query* dengan kode fungsi yang tidak didukung akan menghasilkan *exception response*, yang dapat terlihat pada Led rangkaian master. *Exeption response* yang digunakan adalah *illegal function* dan *illegal data field*.

1. Usman, *Teknik Antarmuka + Pemograman Mikrokontroler AT89S52*, Yogyakarta: ANDI, 2008.
2. Budiharto, Widodo, *Perancangan Sistem dan Aplikasi Mikrokontroler*, Jakarta: PT. Elex Media Komputindo, 2005.
3. Setiawan, Rachmad, *Mikrokontroler MCS-51*, Yogyakarta: Graha Ilmu, 2006.
4. Nalwan, Paulus N, *Panduan Praktis Teknik Antarmuka dan Pemograman Mikrokontroler AT89C51*, Jakarta: PT. Elex Media Komputindo, 2003.
5. Bahri, Kusnassriyanto Saiful, & Wawan Sjachriyanto, *Teknik Pemrograman Delhi Edisi Revisi*, Bandung: INFORMATIKA, 2008.
6. Wahana Komputer, *Aplikasi Cerdas Menggunakan Delphi*, Yogyakarta: ANDI, 2009.