

**MONITORING SISTEM PASOKAN BAHAN BAKAR MINYAK
PADA SPBU MELALUI JARINGAN INTERNET**

SKRIPSI

Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik

UNIVERSITAS BRAWIJAYA



Disusun Oleh :

ANDY KRISNA
0001063228 – 63

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2007**

**MONITORING SISTEM PASOKAN BAHAN BAKAR MINYAK
PADA SPBU MELALUI JARINGAN INTERNET**

SKRIPSI

Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik



Disusun Oleh :
ANDY KRISNA
0001063228 – 63

DOSEN PEMBIMBING:

Ir. Bambang Siswojo
NIP. 131 759 588

Ir. Nanang Sulistyanto
NIP. 132 090 389

**MONITORING SISTEM PASOKAN BAHAN BAKAR MINYAK
PADA SPBU MELALUI JARINGAN INTERNET**

Disusun Oleh :

ANDY KRISNA
0001063228 – 63

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal
9 Februari 2007

DOSEN PENGUJI :

Ir. Ponco Siwindarto, MS
NIP. 131 837 966

Ir. Nurussa'adah
NIP. 131 994 339

Ir. M. Julius ST., MS
NIP. 131 124 655

Ir. Wahyu Adi P., MS
NIP. 131 759 602

Mengetahui,
Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, M.Kom
NIP. 131 879 033

PENGANTAR

Puji dan syukur Penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan kasih dan anugerah-Nya selama ini sehingga Penulis dapat menyelesaikan tugas akhir yang berjudul “MONITORING SISTEM PASOKAN BAHAN BAKAR MINYAK PADA SPBU MELALUI JARINGAN INTERNET” sebagai syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Ucapan terima kasih Penulis sampaikan kepada yang terhormat:

1. Ibunda Hariyati dan ayahanda Mudjiyanto tercinta yang telah memberikan do'a, nasehat dan biaya kuliah sampai terselesainya skripsi ini.
2. Kakakku tercinta Iwan Hariyanto dan Mia Christiana yang telah memberi semangat untuk terus berusaha menyelesaikan skripsi ini.
3. Bapak Ir. Purwanto, MT selaku Ketua Jurusan Teknik Elektro.
4. Bapak Ir. Heri Purnomo selaku Sekretaris Jurusan Teknik Elektro.
5. Bapak Ir. Ponco Siwindarto, MS selaku Ketua Kelompok Pengajar Keahlian Elektronika.
6. Bapak Ir. Bambang Siswojo selaku pembimbing.
7. Bapak Ir. Nanang Sulistiyanto selaku pembimbing.
8. Seluruh dosen dan karyawan Jurusan Teknik Elektro.
9. Rekan-rekan mahasiswa yang telah banyak membantu.
10. Semua pihak yang tidak mungkin Penulis sebutkan satu per satu.

Pembuatan laporan tugas akhir ini telah Penulis wujudkan apa yang telah direncanakan dengan mengusahakan sebatas kemampuan dan pengetahuan Penulis, juga tidak menutup kemungkinan memperoleh masukan-masukan dari semua pihak demi terwujudnya tugas akhir ini.

Sebagai akhir koreksi, Penulis dengan lapang dada dan tangan terbuka menerima segala masukan-masukan demi pengembangan dan kesempurnaan alat serta laporan tugas akhir yang telah dibuat.

Semoga tugas akhir ini bermanfaat bagi kita semua khususnya bagi almamater
Jurusan Teknik Elektro Fakultas Teknik.

Malang, September 2006

Penulis



DAFTAR ISI

PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL	viii
DAFTAR LAMPIRAN	ix
ABSTRAK	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Ruang Lingkup	2
1.4. Tujuan dan Manfaat	3
1.5. Sistematika Penulisan	3
BAB II DASAR TEORI	4
2.1. Mikrokontroler	4
2.1.1. Organisasi Memori	7
2.1.1.1. Memori Program	8
2.1.1.2. Memori Data	9
2.1.2. Pewaktuan CPU	10
2.1.3. Interupsi	10
2.1.3.1. <i>Interrupt Enable</i>	11
2.1.3.2. <i>Interrupt Priority</i>	12
2.1.4. <i>Special Function Register</i>	13
2.1.5. <i>Reset</i>	15
2.1.6. <i>Timer/counter</i>	16
2.2. Sensor	19
2.3. <i>Optocoupler</i>	19
2.3.1. Prinsip Kerja <i>Optocoupler</i>	19
2.4. Tombol Masukan/ <i>Keypad</i>	20
2.5. LCD	21

2.5.1.	LCD Tipe M1632	21
2.6.	TCP/IP	23
2.7.	HTML (<i>Hyper Text Markup Language</i>)	24
2.8.	<i>Miniature Web Server Module</i>	24
BAB III	METODOLOGI	27
3.1.	Studi Literatur	27
3.2.	Perancangan Alat	27
3.3.	Pembuatan Alat	28
3.4.	Pengujian Alat	28
BAB IV	PERANCANGAN DAN PEMBUATAN ALAT	30
4.1.	Perancangan Sistem	30
4.2.	Rangkaian Sensor <i>Flowmeter</i>	31
4.3.	<i>Watermeter</i>	33
4.4.	<i>Driver</i>	34
4.5.	LCD (<i>Liquid Crystal Display</i>)	35
4.6.	Perencanaan Rangkaian <i>Keypad</i>	37
4.7.	Mikrokontroler AT89C51	37
4.8.	RS-232 <i>Driver</i>	39
4.9.	Perangkat Lunak	39
4.9.1.	Mikrokontroler	39
BAB V	PENGUJIAN SISTEM	42
5.1.	Pengujian Rangkaian <i>Optocoupler</i>	42
5.1.1.	Tujuan	42
5.1.2.	Alat Yang Digunakan	42
5.1.3.	Prosedur Pengujian	42
5.1.4.	Hasil dan Analisis	43
5.2.	Pengujian <i>Keypad</i> dan LCD	43
5.2.1.	Tujuan	43
5.2.2.	Alat Yang Digunakan	43
5.2.3.	Prosedur Pengujian	43
5.2.4.	Hasil dan Analisis	44

5.3. Pengujian Rangkaian Serial RS-232	45
5.3.1. Tujuan	45
5.3.2. Alat Yang Digunakan	45
5.3.3. Prosedur Pengujian	45
5.3.4. Hasil dan Analisis	45
5.4. Pengujian Rangkaian <i>Driver</i>	46
5.4.1. Tujuan	46
5.4.2. Alat Yang Digunakan	46
5.4.3. Prosedur Pengujian	47
5.4.4. Hasil dan Analisis	47
5.5. Pengujian Sistem Secara Keseluruhan	48
5.5.1. Tujuan	48
5.5.2. Alat Yang Digunakan	48
5.5.3. Prosedur Pengujian	48
5.5.4. Hasil dan Analisis	48
 BAB VI PENUTUP	 51
6.1. Kesimpulan	51
6.2. Saran	51
 DAFTAR PUSTAKA	 52
 LAMPIRAN	 53

ABSTRAK

ANDY KRISNA, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, September 2006, *Monitoring Sistem Pasokan Bahan Bakar Minyak Pada SPBU Melalui Jaringan Internet*, Dosen Pembimbing : Ir. Bambang Siswojo dan Ir. Nanang Sulistiyanto.

Kendaraan bermotor saat ini merupakan kebutuhan utama bagi sebagian besar orang. Seiring dengan terus bertambahnya jumlah kendaraan bermotor mengakibatkan kebutuhan akan bahan bakar minyak (BBM) juga meningkat.

Saat ini dalam pengelolaan SPBU beberapa pekerjaan masih dilakukan secara manual sehingga manajemen perusahaan berjalan kurang efisien. Kondisi ini tidak mampu mengatasi perkembangan kebutuhan di masa depan dimana SPBU harus beroperasi penuh selama 24 jam perhari dan masalah efisiensi waktu dan efektivitas kerja dimana nantinya hampir semua pelayanan SPBU terhadap konsumen dilakukan secara otomatis. Untuk itu sarana yang memadai seperti perlunya otomatisasi dan komputerisasi dalam manajemen perusahaan sangat dibutuhkan.

Pada prinsipnya kerja dari sistem ini adalah memonitor sistem pemasokan Bahan Bakar Minyak pada SPBU dengan cara melihat volume pada tangki SPBU yang ditampilkan pada layar monitor komputer yang bisa diakses melalui jaringan internet. Dimana sensor yang berfungsi untuk mengukur jumlah bahan bakar minyak yang dikeluarkan akan memberikan sinyal masukan ke mikrokontroler yang selanjutnya akan diproses oleh mikrokontroler. Status sensor akan memberikan data dan ditampilkan pada komputer yang bisa diakses melalui jaringan internet dengan menggunakan microweb server.

Apabila volume pada tangki sudah berada pada tahap pengisian, dengan melihat data yang dihasilkan maka secara otomatis Depo Pertamina akan mengirimkan bahan bakar minyak ke SPBU dan memberikan *password* pada sopir untuk proses pengisian. *Password* yang diberikan harus dimasukkan pada mikrokontroler melalui *keypad* yang akan ditampilkan pada LCD yang berfungsi untuk membuka *valve*. Setelah *valve* terbuka proses pengisian baru bisa dilaksanakan. Pemberian *password* pada sopir juga untuk mengetahui ketepatan waktu pengiriman dari Depo Pertamina ke SPBU yang dituju.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan bermotor saat ini merupakan kebutuhan utama bagi sebagian besar orang. Seiring dengan terus bertambahnya jumlah kendaraan bermotor mengakibatkan kebutuhan akan bahan bakar minyak (BBM) juga meningkat. Hal ini menciptakan lahan yang potensial untuk pengusaha SPBU, ditandai dengan makin banyaknya dibangun SPBU-SPBU baru.

Saat ini dalam pengelolaan SPBU beberapa pekerjaan masih dilakukan secara manual sehingga manajemen perusahaan berjalan kurang efisien. Kondisi ini tidak mampu mengatasi perkembangan kebutuhan di masa depan dimana SPBU harus beroperasi penuh selama 24 jam perhari dan masalah efisiensi waktu dan efektivitas kerja dimana nantinya hampir semua pelayanan SPBU terhadap konsumen dilakukan secara otomatis. Untuk itu sarana yang memadai seperti perlunya otomatisasi dan komputerisasi dalam manajemen perusahaan sangat dibutuhkan.

Menunjuk dari tulisan Tugas Akhir **Nur Kholis Umam, 2002 : Sistem Monitoring Volume Bahan Bakar SPBU**. Sistem tersebut menggunakan sensor ultrasonik untuk mengukur volume bahan bakar pada tangki SPBU secara otomatis dan tersentral yang dihubungkan secara serial pada personal komputer (PC) yang tersimpan pada suatu ruangan administrasi di SPBU. Kelemahannya adalah komputer tidak terhubung jaringan internet, sehingga proses pengiriman bahan bakar baru bisa dilakukan setelah pihak SPBU melakukan pemesanan ke Depo Pertamina karena proses monitoring hanya dapat dilakukan di SPBU saja.

Berdasarkan hal tersebut, dalam penelitian ini akan dirancang dan dibuat suatu sistem yang dapat memonitor transaksi bahan bakar yang dikeluarkan tangki SPBU dan pengisiannya secara otomatis dan tersentral melalui internet. Data monitoring yang dihasilkan dapat digunakan untuk mengetahui volume bahan bakar yang masih tersedia dan untuk memprediksi waktu pasok BBM dari Depo Pertamina menuju SPBU, sehingga keputusan untuk mendatangkan mobil pengisi BBM dapat dilakukan secara tepat dan masalah kelangkaan BBM juga dapat dihindari.

Manusia hanya berperan sebagai operator yang dapat dilakukan pada komputer, sedangkan komputer berperan sebagai pusat kontrol yang melakukan pemantauan dan pengendalian terhadap sistem. Agar pelaksanaan pemantauan dan pengendalian tidak terhambat oleh jarak, yang berarti bahwa operator dapat melakukan tugasnya dari jarak jauh atau tidak harus berada di lokasi, maka digunakan jaringan internet.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang diuraikan pada pendahuluan, maka rumusan masalah ditekankan pada:

1. Bagaimana merancang dan membuat rangkaian sensor yang dapat memberikan informasi tentang volume BBM pada tangki SPBU.
2. Bagaimana merancang sistem elektronik yang dapat diaplikasikan untuk memonitor volume BBM pada tangki SPBU.
3. Bagaimana merancang protokol komunikasi data serial.
4. Bagaimana membuat perangkat lunak pada mikrokontroler dan microweb untuk pengambilan data, pengiriman data ke unit pengolah data, dan penanganan perintah yang diterima dari unit pengolah data.
5. Bagaimana menggabungkan antara perangkat keras dan perangkat lunak sehingga dapat digunakan untuk memantau dan mengontrol volume BBM pada tangki SPBU.
6. Bagaimana melakukan pengujian alat secara per blok dan keseluruhan.

1.3 Ruang Lingkup

Setelah melihat permasalahan secara umum dan menyeluruh, maka penulis merasa perlu untuk membatasi permasalahan agar dapat mendapatkan kejelasan yang lebih terarah dan sesuai dengan apa yang diinginkan. Adapun batasan-batasan masalahnya adalah sebagai berikut :

1. Volume BBM yang dimonitor hanya satu tangki.
2. Tidak membahas sumber dari catu daya yang digunakan alat.
3. Tidak membahas kabel jaringan termasuk gejala-gejala yang terdapat didalamnya.
4. Tidak membahas bagian mekanik dari tangki.
5. Sistem yang dibuat berupa prototipe.
6. Tidak membahas perangkat keras dari microweb.
7. Tidak membahas *losses* pada bahan bakar yang terjadi pada saat pengiriman BBM dari Depo Pertamina menuju SPBU.

1.4 Tujuan dan Manfaat

Tujuan dari perancangan sistem ini adalah :

1. Merancang dan membuat suatu sistem yang dapat memonitor volume BBM pada tangki SPBU secara terpusat melalui jaringan internet.
2. Proses pengiriman BBM dari Depo Pertamina menuju SPBU bisa tepat waktu.

Manfaat dari perancangan ini adalah :

1. Dengan adanya sistem ini diharapkan proses monitoring dapat berlangsung dengan cepat dan masalah kelangkaan akan BBM bisa teratasi.
2. Data monitoring yang tersimpan pada komputer juga dapat digunakan sebagai data *cross check* terhadap data transaksi bahan bakar dan mampu menunjukkan perkembangan transaksi bahan bakar dari waktu ke waktu.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut :

- BAB I** : Menjelaskan latar belakang, permasalahan, perumusan masalah, pembatasan masalah, tujuan penulisan, metodologi serta sistematika penulisan.
- BAB II** : Menjelaskan tentang teori dasar serta komponen pendukung yang digunakan dalam perancangan alat..
- BAB III** : Menjelaskan tentang perencanaan setiap rangkaian elektronika dan perangkat lunak yang digunakan dalam Monitoring Sistem Pasokan Bahan Bakar Minyak Pada SPBU Melalui Jaringan Internet.
- BAB IV** : Menjelaskan tentang perancangan sistem, diagram blok rangkaian, prinsip kerja alat, perancangan *hardware* dan *software* serta realisasi alat.
- BAB V** : Menjelaskan hasil pengujian terhadap alat yang telah dibuat, baik setiap blok rangkaian maupun secara keseluruhan.
- BAB VI** : Menjelaskan kesimpulan yang dapat diambil dan saran-saran untuk pengembangan alat yang telah dibuat.
- Lampiran** : Berisi tentang perangkat lunak sistem mikrokontroler, gambar rangkaian, gambar alat dan data IC yang digunakan.

BAB II

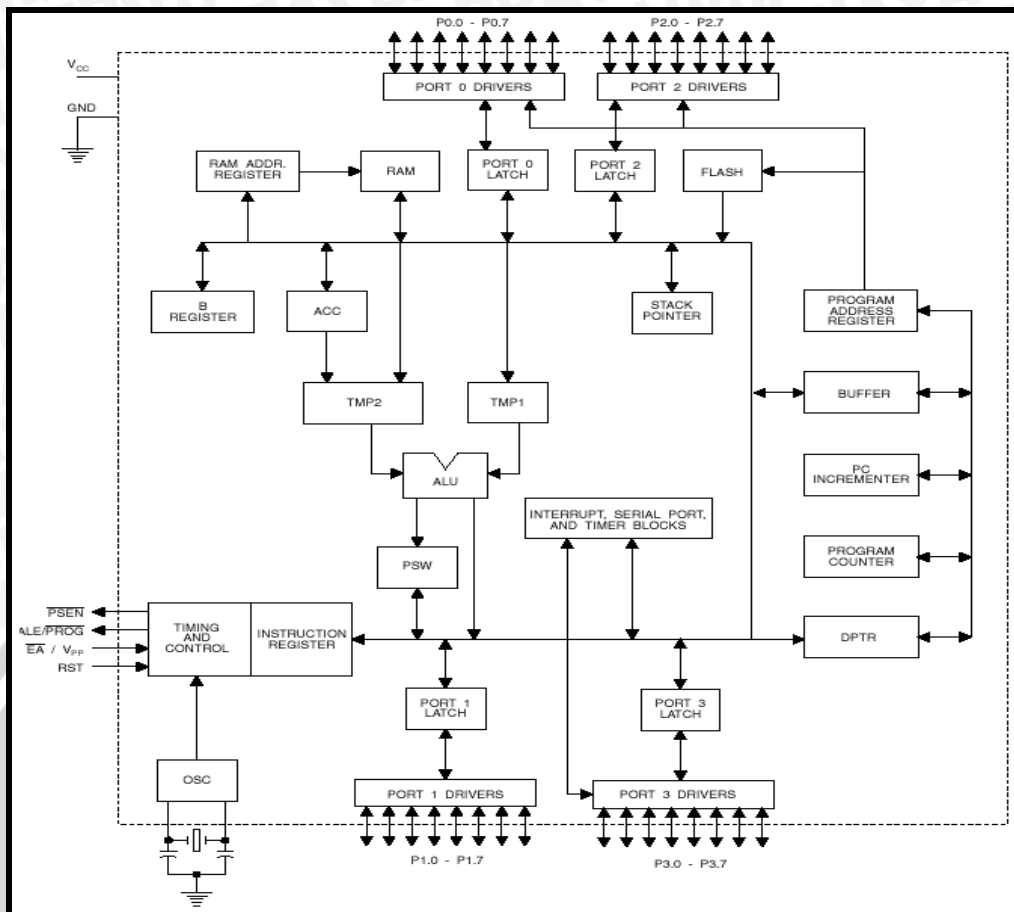
TEORI DASAR

Dalam bab berikut ini akan dijelaskan mengenai dasar teori, untuk dapat memahami alat yang akan dirancang. *Monitoring Sistem Pasokan Bahan Bakar Minyak Pada SPBU Melalui Jaringan Internet* ini, pada prinsipnya adalah proses perhitungan jumlah volume tangki yang dikeluarkan oleh sensor baling-baling kemudian akan ditampilkan pada layar monitor yang bias diakses melalui jaringan internet. Teori-teori pendukung yang akan dibahas pada bab ini adalah :

2.1 Mikrokontroler

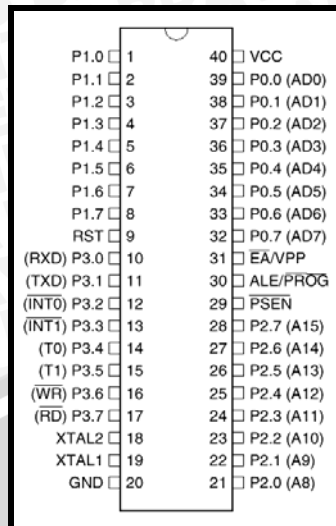
Sebagai pengontrol dari sistem ini memanfaatkan mikrokontroler yaitu IC AT89C51. Alasan penggunaan komponen ini adalah karena komponen ini dirasa cukup praktis dengan kelebihan dalam satu chip dapat menampung program internal sebanyak 4 Kb flash memori untuk operasi internal, 128 byte RAM, 32 jalur I/O, dua buah timer 16-bit, arsitektur interrupt dua tingkat lima jalur, full duplex port serial, rangkaian osilator, timer/counter internal dan dapat di program dengan bahasa assembly Intel.

Mikrokontroler AT89C51 mempunyai kemampuan instruksi dan konfigurasi pin dengan mikrokontroler MCS-51, mikrokontroler MCS-51 merupakan komponen yang berorientasi pada kontrol (*microcontroller*), dan oleh Intel diklasifikasikan dalam kelompok *embedded microcontroller*, yang berarti dapat diprogram ulang (*reprogrammable*). Blok diagram MCS-51 ditunjukkan dalam Gambar 2.1.



Gambar 2.2. Blok Diagram Mikrokontroler AT89C51.
 Sumber : Atmel, 1998 : 2

Masing-masing kaki dalam mikrokontroler AT89C51 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler AT89C51, perancangan aplikasi mikrokontroler AT89C51 akan lebih mudah merencanakan dan membuat sistem yang dirancang. AT89C51 mempunyai 40 pin, susunan masing-masing pin dapat dilihat pada Gambar 2.3.



Gambar 2.3. Susunan kaki MCS-51.
Sumber : Atmel, 1998 : 1

Fungsi penyemat (kaki) AT89C51 adalah:

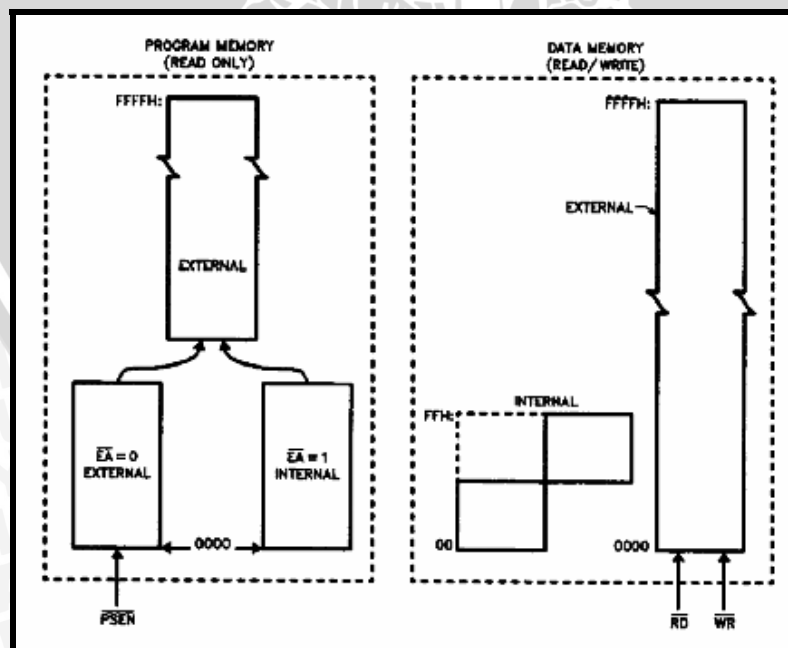
- Penyemat 1 sampai 8 (*Port 1*) merupakan port paralel 8-bit dua arah (*bidirectional*) yang dapat digunakan untuk berbagai keperluan (*general purpose*).
- Penyemat 9 (*Reset*) adalah masukan *reset* (aktif tinggi), pulsa transisi dari rendah ke tinggi (tepi naik) akan me-*reset* AT89C51.
- Penyemat 10 sampai 17 (*Port 3*) adalah port paralel 8-bit dua arah yang memiliki fungsi pengganti. Fungsi pengganti meliputi *RxD* (*Receive Data*), *TxD* (*Transmit Data*), $\overline{\text{INT}}_0$ (*Interrupt 0*), $\overline{\text{INT}}_1$ (*Interrupt 1*), *T0* (*Timer 0*), *T1* (*Timer 1*), $\overline{\text{WR}}$ (*Write*), dan $\overline{\text{RD}}$ (*Read*). Bila fungsi pengganti tidak dipakai, maka Penyemat-Penyemat ini dapat digunakan sebagai port paralel 8-bit serbaguna.
- Penyemat 18 (*XTAL 1*) adalah Penyemat masukan ke rangkaian osilator internal, dapat digunakan sebuah osilator kristal atau sumber osilator luar.
- Penyemat 19 (*XTAL 2*) adalah Penyemat keluaran ke rangkaian osilator internal, penyemat ini dipakai bila menggunakan osilator kristal.
- Penyemat 20 (*Ground*) dihubungkan ke *Vss* atau *ground*.
- Penyemat 21 sampai 28 (*Port 2*) adalah port paralel 2 (*P2*) selebar 8-bit dua arah (*bidirectional*). Port 2 ini mengirimkan byte alamat bila dilakukan pengaksesan memori eksternal.
- Penyemat 29 adalah Penyemat $\overline{\text{PSEN}}$ (*program store enable*) yang merupakan sinyal pengontrol yang membolehkan program memori eksternal masuk ke dalam bus selama proses pemberian/pengambilan instruksi (*fetching*).

- Penyemat 30 adalah Penyemat ALE (*Address Latch Enable*) yang digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
- Penyemat 31 adalah Penyemat \overline{EA} (*External Access Enable*) bila penyemat ini diberi logika tinggi (H), mikrokontroler akan melaksanakan instruksi dari ROM/EPROM ketika isi program *counter* kurang dari 4096, bila diberi logika rendah (L), mikrokontroler akan melaksanakan seluruh instruksi dari memori program luar.
- Penyemat 32 sampai 39 (Port 0) merupakan port paralel 8-bit *open drain* dua arah, bila digunakan untuk mengakses memori luar, port ini akan memultipleks alamat memori dengan data.
- Penyemat 40 (Vcc) dihubungkan ke Vcc (+5 volt).

2.1.1 Organisasi Memori

Semua mikrokontroler dalam keluarga MCS-51 memiliki pembagian ruang alamat (*address space*) untuk program dan data seperti yang ditunjukkan dalam Gambar 2.4. Pemisahan memori program dan memori data membolehkan memori data untuk diakses oleh alamat 8-bit. Sekalipun demikian, alamat data memori 16 bit dapat dihasilkan melalui register DPTR (*data pointer register*).

Memori program hanya dapat dibaca, tidak bisa ditulisi (karena disimpan dalam EPROM). Memori program sebesar 64k dapat dimasukkan dalam EPROM eksternal.



Gambar 2.4 Struktur memori Mikrokontroler MCS-51.
Sumber : Intel, 1998

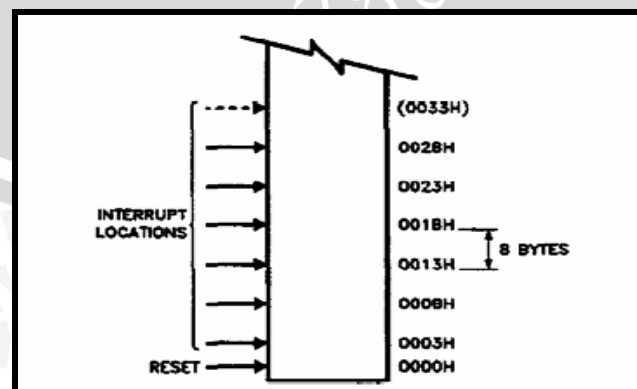
Sinyal yang membolehkan pembacaan dari memori program eksternal adalah dari Penyemat $\overline{\text{PSEN}}$ (*program store enable*). Memori data terletak pada ruang alamat terpisah dari memori program. RAM eksternal 64k dapat dialamati dalam ruang memori data eksternal. CPU menghasilkan sinyal *read* dan *write* selama menghubungi memori data eksternal.

Mikrokontroler AT89C51 memiliki 5 buah ruang alamat, yaitu:

1. Ruang alamat kode (*code address space*) sebanyak 64k, yang seluruhnya merupakan ruang alamat kode eksternal (*off-chip*).
2. Ruang alamat data internal yang dapat dialamati secara langsung, yang terdiri atas:
 - RAM (*Random Access Memori*) sebanyak 128 byte.
 - *Hardware register* sebanyak 128 byte.
 - EEPROM sebanyak 4k byte.
3. Ruang alamat data internal yang dialamati secara tidak langsung sebanyak 128 byte, seluruhnya diakses dengan pengalamatan tidak langsung.
4. Ruang alamat data eksternal sebanyak 64k byte (*off-chip*) yang dapat ditambahkan oleh pemakai.
5. Ruang alamat bit, dapat diakses dengan pengalamatan langsung.

2.1.1.1 Memori Program

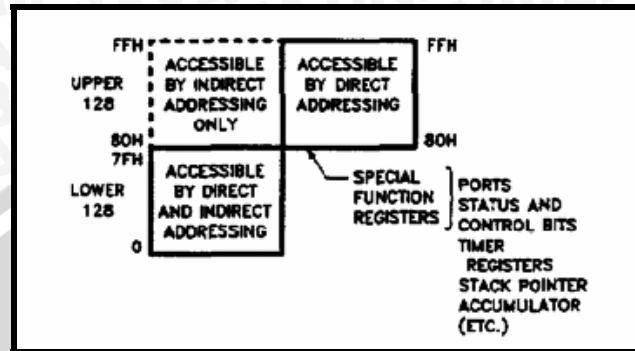
Gambar 2.5 memperlihatkan bagian bawah dari memori program. Setelah *reset* CPU memulai eksekusi dari lokasi 0000H. Setiap interupsi mempunyai lokasi tetap dalam memori program. Interupsi menyebabkan CPU melompat ke lokasi tersebut di mana pada lokasi tersebut terdapat sub-rutin yang harus dilaksanakan.



Gambar 2.5. Memori Program.
Sumber : Intel, 1998

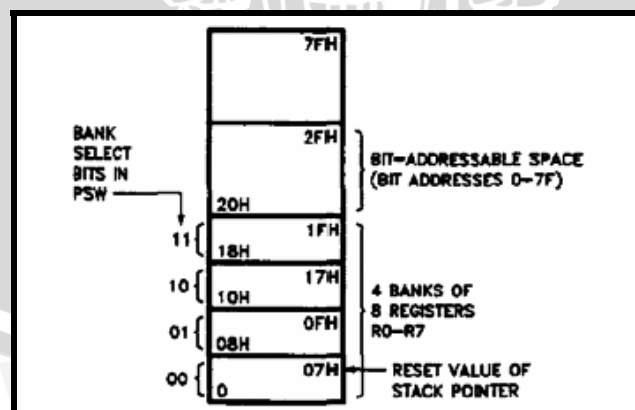
2.1.1.2 Memori Data

Memori data internal dipetakan seperti dalam Gambar 2.6. Ruang memorinya dibagi menjadi tiga blok, yaitu sebagai *lower 128*, *upper 128*, dan ruang SFR (*special function register*).



Gambar 2.6. Memori Data Internal.
Sumber : Intel, 1998

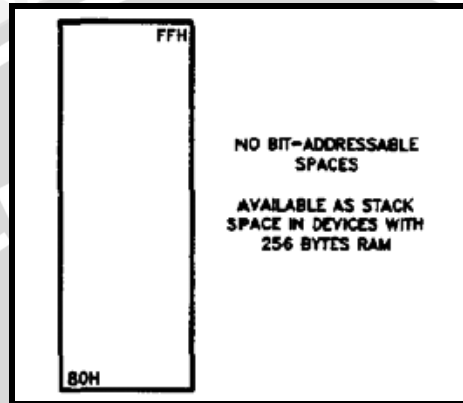
Bagian bawah dari 128 byte RAM dipetakan seperti terlihat dalam Gambar 2.7. Tiga puluh dua byte paling bawah dikelompokkan dalam 4 bank (masing-masing terdiri atas 8 register, yaitu R0 sampai R7). Dua bit dalam PSW (*program status word*) memilih register bank yang digunakan. Hal ini memungkinkan penggunaan yang lebih efisien dari ruang alamat kode, karena instruksi yang menggunakan register lebih singkat dari pada instruksi yang menggunakan pengalamatan langsung.



Gambar 2.7. Bagian bawah 128 byte RAM Internal.
Sumber : Intel, 1998

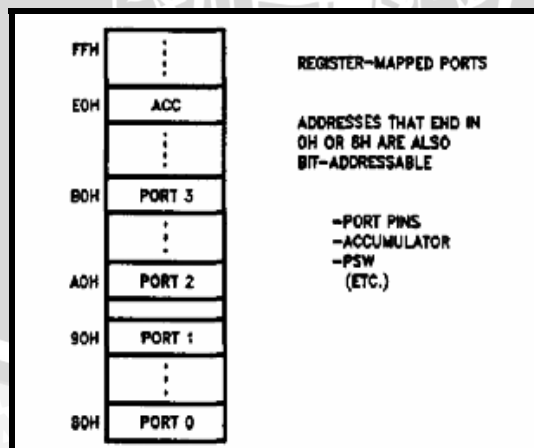
Enam belas byte di atas bank register (alamat 20H sampai 2FH) membentuk blok ruang memori untuk pengalaman bit. Alamat bit pada daerah ini adalah mulai 00H sampai 7FH

Semua byte pada bagian bawah 128 byte RAM internal dapat diakses oleh pengalaman langsung dan tak langsung. Sedangkan bagian atas 128 byte RAM internal ditunjukkan dalam Gambar 2.8 hanya dapat diakses dengan pengalaman tak langsung.



Gambar 2.8. Bagian Atas 128 Byte RAM Internal.
Sumber : Intel, 1998

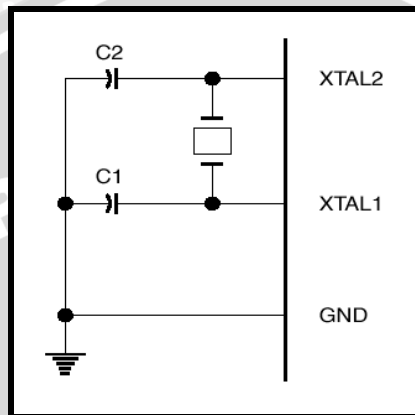
Gambar 2.9 menunjukkan ruang SFR (SFR space). SFR berisi penahan port (*port latch*), pewaktu (*timer*), pengontrol periferai, dan lain-lain. Register ini hanya dapat diakses oleh pengalaman langsung.



Gambar 2.9. Ruang Special Function Register.
Sumber : Intel, 1998

2.2.2 Pewaktuan CPU

Mikrokontroler AT89C51 memiliki osilator internal (*on chip oscillator*) yang dapat digunakan sebagai sumber clock bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal antara penyemat XTAL 1 dan penyemat XTAL 2 dan dua buah kapasitor ke ground seperti terlihat dalam Gambar 2.10. Dapat digunakan kristal dengan frekuensi antara 6 sampai 12 MHz, sedangkan untuk kapasitor dapat digunakan antara 27 pF sampai 33 pF [Atmel, 1998 : 4].



Gambar 2.10. Menggunakan Osilator Internal.
Sumber : Atmel, 1998 : 4

2.2.3 Interupsi

Pada saat CPU dalam mikrokontroler AT89C51 sedang melaksanakan suatu program, pelaksanaan program tersebut dapat dihentikan sementara dengan interupsi. Apabila CPU mendapatkan interupsi, *program counter* (PC) akan diisi alamat dari vektor interupsi. Kemudian CPU melaksanakan rutin pelayanan interupsi mulai dari alamat tersebut. Bila rutin pelayanan interupsi selesai dilaksanakan, maka CPU AT89C51 akan kembali melaksanakan program utama.

Terdapat beberapa saluran interupsi pada mikrokontroler AT89C51 yang dibedakan menjadi 2 jenis, yaitu:

1. Interupsi yang tak dapat dihalangi oleh perangkat lunak (*non maskable interrupt*), contoh interupsi jenis ini adalah *reset*.
2. Interupsi yang dapat dihalangi oleh perangkat lunak (*maskable interrupt*). Contoh interupsi jenis ini adalah \overline{INT}_0 dan \overline{INT}_1 (eksternal) serta *Timer/counter* 0, *Timer/counter* 1, dan interupsi dari port serial (internal).

Instruksi RETI (*return from interrupt routine*) harus digunakan untuk kembali dari layanan rutin interupsi. Instruksi ini dipakai agar saluran interupsi kembali dapat

dipakai. Alamat awal layanan rutin interupsi dari *setiap* sumber interupsi seperti terlihat dalam Tabel 2.1.

Tabel 2.1. Alamat layanan rutin interupsi.

Nama	Lokasi	Alat interupsi
Reset	00H	Power on reset
\overline{INT}_0	03H	\overline{INT}_0
Timer 0	0BH	Timer 0
\overline{INT}_1	13H	\overline{INT}_1
Timer 1	1BH	Timer 1
Sint	23H	Port I/O serial

Sumber : Malik, 1997: 16

Mikrokontroler AT89C51 menyediakan 5 sumber interupsi: 2 interupsi eksternal, 2 interupsi *timer*, dan 1 interupsi port serial. Interupsi eksternal \overline{INT}_0 dan \overline{INT}_1 masing-masing dapat diaktifkan berdasarkan level atau transisi, tergantung pada bit IT0 dan IT1 dalam TCON. Flag yang menghasilkan interupsi ini adalah bit dalam IE0 dan IE1 dari TCON. Interupsi *Timer 0* dan *Timer 1* dihasilkan oleh TF0 dan TF1. Interupsi port serial dihasilkan oleh logika OR dari R1 dan T1.

Terdapat 2 buah register yang mengontrol interupsi, yaitu IE (*interrupt enable*) dan IP (*interrupt priority*). Mikrokontroler AT89C51 tidak akan menanggapi permintaan interupsi jika suatu instruksi belum dilaksanakan secara lengkap.

2.2.3.1 *Interrupt Enable*

Setiap sumber interupsi dapat diaktifkan maupun dilumpuhkan secara individual dengan cara mengatur 1 bit di SFR yang bernama IE (*interrupt enable*). Bit-bit IE didefinisikan sebagai berikut:

MSB				LSB			
EA	—	—	ES	ET1	EX1	ET0	EX0

Simbol	Posisi	Fungsi
EA	IE.7	Melumpuhkan semua interupsi. Jika EA=0 tidak ada interupsi yang akan dilayani. Jika EA=1 setiap sumber interupsi dapat dijalankan atau dilumpuhkan secara individual.
—	IE.6	Kosong
—	IE.5	Kosong
ES	IE.4	Bit pembuat enable port serial
ET1	IE.3	Bit pembuat enable <i>timer</i> 1
EX1	IE.2	Bit pembuat enable /INT 1
ET0	IE.1	Bit pembuat enable <i>timer</i> 0
EX0	IE.0	Bit pembuat enable /INT 0

Jika akan mengaktifkan interupsi 0 ($\overline{\text{INT}}_0$), misalnya, nilai yang harus diberikan ke IE adalah 81H (yaitu memberikan logika 1 ke EA dan EX0).

2.2.3.2 Interrupt Priority

Setiap sumber interupsi dapat diprogram secara individual (sendiri-sendiri) menjadi satu atau dua tingkat prioritas dengan mengatur bit pada SFR yang bernama IP (*interrupt priority*). Interupsi dengan prioritas rendah (*low priority*) dapat diinterupsi oleh interupsi yang memiliki prioritas lebih tinggi (*high priority*), akan tetapi interupsi yang memiliki prioritas lebih rendah tidak dapat menginterupsi interupsi yang memiliki prioritas lebih tinggi. Interupsi yang memiliki prioritas tertinggi tidak dapat diinterupsi oleh interupsi lainnya.

Apabila terjadi dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan, permintaan interupsi dengan prioritas tertinggi yang akan dilayani. Jika permintaan interupsi dengan prioritas yang sama diterima bersamaan, akan dilakukan *polling* untuk menentukan mana yang akan dilayani.

Bit-bit pada IP adalah sebagai berikut:



Priority bit = 1 menandakan prioritas tinggi

Priority bit = 0 menandakan prioritas rendah

Simbol	Posisi	Fungsi
—	IP.7	Kosong
—	IP.6	Kosong
—	IP.5	Kosong
PS	IP.4	Bit prioritas interupsi port serial
PT1	IP.3	Bit prioritas interupsi <i>timer</i> 1
PX1	IP.2	Bit prioritas interupsi /INT 1
PT0	IP.1	Bit prioritas interupsi <i>timer</i> 0
PX0	IP.0	Bit prioritas interupsi /INT 0

2.2.4 Special Function Register

SFR berisi register-register dengan fungsi tertentu. Masing-masing register seperti ditunjukkan dalam Tabel 2.2 yang meliputi simbol, nama, dan alamatnya serta keadaannya dalam nilai biner pada saat terjadi *power-on reset*.

Beberapa fungsi SFR yang penting, yaitu:

1. *Accumulator* (ACC) merupakan register penting dalam operasi penambahan dan pengurangan.
2. Register B merupakan register khusus yang berfungsi melayani operasi perkalian dan operasi pembagian.
3. *Program Status Word* (PSW) berisi beberapa bit status yang menggambarkan keadaan CPU pada saat itu. PSW terdiri atas bit *carry*, *auxiliary carry*, dua bit pemilih *bank*, bendera *overflow*, bit paritas dan dua *flag* yang dapat didefinisikan sendiri oleh pemakai.
4. *Stack Pointer* (SP) merupakan register 8-bit. Register SP dapat diletakkan pada alamat manapun pada RAM internal. Isi register ini ditambah sebelum data

disimpan, selama instruksi *PUSH* dan *CALL*. Pada saat *reset*, register SP diinisialisasi pada alamat 07H, sehingga *stack* akan dimulai pada lokasi 08H.

Tabel 2.2. *Special Function Register.*

Simbol	Nama	Alamat	Power-on Reset
ACC	Akumulator	E0H	0000 0000
B	Register B	F0H	0000 0000
PSW	Program Status Word	D0H	0000 0000
SP	Stack Pointer	81H	0000 0111
DPTR	Data Pointer 16 bit		
DPL	Data Pointer byte rendah	82H	0000 0000
DPH	Data Pointer byte tinggi	83H	0000 0000
P0	Port 0	80H	1111 1111
P1	Port 1	90H	1111 1111
P2	Port 2	A0H	1111 1111
P3	Port 3	B0H	1111 1111
IP	Interrupt Priority Control	B8H	xxx0 0000
IE	Interrupt Enable Control	A8H	0xx0 0000
TMOD	Timer/counter Mode Control	89H	0000 0000
TCON	Timer/counter Control	88H	0000 0000
TH0	Timer/counter 0 high byte	8CH	0000 0000
TL0	Timer/counter 0 low byte	8AH	0000 0000
TH1	Timer/counter 1 high byte	8DH	0000 0000
TL1	Timer/counter 1 low byte	8BH	0000 0000
SCON	Serial Control	98H	0000 0000
SBUF	Serial Data Buffer	99H	xxxx xxxx
PCON	Power Control	87H	0xxx xxxx

Sumber : Malik, 1997: 19-20

Data Pointer (DPTR) terdiri dari dua register, yaitu untuk byte tinggi (*Data Pointer High*, DPH) dan untuk byte rendah (*Data Pointer Low*, DPL). Fungsinya dimaksudkan untuk menahan alamat 16-bit. DPTR dapat dimanipulasi sebagai register 16-bit atau sebagai dua buah register 8-bit.

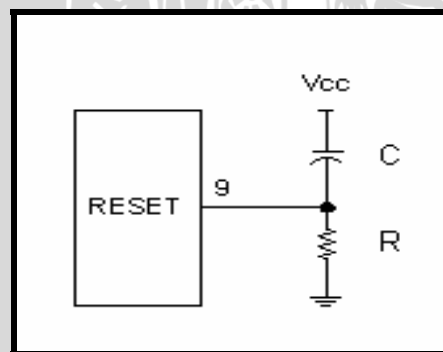
Port 0 sampai Port 3 merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada port 0, 1, 2, dan 3. Masing-masing register ini dapat dialamati per-bit maupun per-byte.

Control Register: terdiri atas register yang mempunyai fungsi kontrol. Terdapat dua register khusus untuk mengontrol sistem interupsi, yaitu register IP (*Interrupt Priority*) dan register IE (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/counter* terdapat register khusus, yaitu register TMOD (*Timer/counter Mode Control*) dan register TCON (*Timer/counter Control*), serta untuk pelayanan port serial menggunakan register SCON (*Serial Port Control*).

2.2.5 Reset

Untuk me-*reset* rangkaian mikrokontroler, digunakan rangkaian *power-on reset*. Rangkaian ini akan me-*reset* mikrokontroler secara otomatis *setiap* kali catu daya dihidupkan. Gambar 2.11 menunjukkan rangkaian *power-on reset*.

Prinsip kerja rangkaian *reset* adalah: saat catu daya diaktifkan, rangkaian *reset* akan menahan logika tinggi pada penyemat RST untuk jangka waktu tertentu. Jangka waktu tersebut ditentukan oleh pengosongan muatan pada kondensator. Untuk memastikan keabsahan *reset*, logika tinggi tersebut harus ditahan untuk waktu yang lebih lama dari dua siklus mesin ditambah waktu mulai hidup (*start-on*) osilator.



Gambar 2.11. Rangkaian *Power-on Reset*.
Sumber : Malik, 1997: 8

2.2.6 Timer/counter

Pada mikrokontroler AT89C51 terdapat dua buah *timer/counter* 16 bit yang dapat diatur melalui perangkat lunak, yaitu *timer/counter* 0 dan *timer/counter* 1.

Apabila *timer/counter* diaktifkan pada frekuensi kerja mikrokontroler 12 MHz, *timer/counter* akan melakukan perhitungan waktu sekali *setiap* 1 mikrodetik secara *independent*, tidak tergantung pada pelaksanaan suatu instruksi. Satu siklus pencacahan waktu berpadanan dengan satu siklus pelaksanaan instruksi, sedangkan satu siklus diselenggarakan dalam waktu 1 mikrodetik. Bila dimisalkan suatu urutan instruksi telah selesai dilaksanakan dalam waktu 5 mikrodetik, pada saat itu pula *timer/counter* telah menunjukkan perioda waktu 5 mikrodetik.

Apabila perioda waktu tertentu telah dilampaui, *timer/counter* segera menginterupsi mikrokontroler untuk memberitahukan bahwa perhitungan perioda waktu telah selesai dilaksanakan. Perioda waktu *timer/counter* secara umum ditentukan oleh persamaan berikut:

☞ Sebagai *timer/counter* 8 bit:

$$T = (255 - TLx) * 1 \mu s \dots\dots\dots(2-1)$$

Dimana TLx adalah isi register TL0 atau TL1.

☞ Sebagai *timer/counter* 16 bit:

$$T = (65535 - THx * TLx) * 1 \mu s \dots\dots\dots(2-2)$$

THx = isi register TH0 atau TH1

TLx = isi register TLO atau TL1

Pengontrol kerja *timer/counter* adalah register *timer control* (TCON). Adapun definisi dari bit-bit pada *timer control* adalah sebagai berikut:

MSB				LSB			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Simbol	Posisi	Fungsi
TF1	TCON.7	<i>Timer 1 overflow flag</i> . Di- <i>set</i> oleh perangkat keras saat <i>timer/counter</i> menghasilkan limpahan (<i>overflow</i>).
TR1	TCON.6	Bit untuk menjalankan <i>Timer 1</i> . Di- <i>set/clear</i> oleh <i>software</i> untuk membuat <i>timer on</i> atau <i>off</i> .
TF0	TCON.5	<i>Timer 0 overflow flag</i> . Di- <i>set</i> oleh perangkat keras.
TR0	TCON.4	Bit untuk menjalankan <i>timer 0</i> . Di- <i>set/clear</i> oleh <i>software</i> untuk membuat <i>timer on</i> atau <i>off</i> .
IE1	TCON.3	<i>External interrupt 1 edge flag</i> .

IT1	TCON.2	<i>Interrupt 1 type control bit. Set/clear oleh software untuk menspesifikasikan sisi turun/level rendah trigger dari interupsi eksternal.</i>
IE0	TCON.1	<i>External interrupt 0 edge flag.</i>
IT0	TCON.0	<i>Interrupt 0 type control bit.</i>

Pengontrol pemilihan mode operasi *timer/counter* adalah register *timer mode* (TMOD) yang mana definisi bit-bitnya adalah sebagai berikut:

MSB				LSB			
GATE	C/T	M1	M0	GATE	C/T	M1	M0

Keterangan:

GATE: Saat GATE = 1, *Timer/counter* x akan berjalan ketika TRx dalam TCON di-set 1 (*timer* dikontrol *software*).

C/T: Pilih fungsi *timer* atau *counter*. *Clear* (0) untuk operasi *timer* dengan masukan dari sistem clock internal. *Set* (1) untuk operasi *counter* dengan masukan dari pena T0 atau T1.

M1: Bit pemilih mode.

M0: Bit pemilih mode.

Kombinasi M0 dan M1 adalah sebagai berikut:

M1	M0	Mode	Operasi
0	0	0	<i>Timer</i> 13 bit
0	1	1	<i>Timer/counter</i> 16 bit
1	0	2	<i>Timer auto reload</i> 8 bit (pengisian otomatis) TL0 adalah <i>timer/counter</i> 8 bit yang dikontrol oleh kontrol bit standar <i>Timer</i> 0
1	1	3	TH0 adalah <i>timer</i> 8 bit dan dikontrol oleh kontrol bit <i>Timer</i> 1

Keterangan:

☞ **Mode 0:**

Dalam mode ini register *timer* disusun sebagai register 13 bit. Setelah semua perhitungan selesai, mikrokontroler akan men-set *Timer Interrupt Flag* (TF1). Dengan



membuat $GATE=1$, *timer* dapat dikontrol oleh masukan luar \overline{INT}_1 , untuk fasilitas pengukuran lebar pulsa.

➤ **Mode 1:**

Mode 1 sama dengan mode 0 kecuali register *timer* akan bekerja dalam 16 bit.

➤ **Mode 2:**

Mode 2 menyusun register *timer* sebagai 8 bit *counter*. Limpahan (*overflow*) dari TL1 tidak hanya men-*set* TF1 tetapi juga mengisi TL1 dengan isi TH1 yang diatur secara *software*. Pengisian ini tidak mengubah TH1.

➤ **Mode 3:**

Timer 1 dalam mode 3 semata-mata memegang hitungan. Efeknya sama seperti men-*set* $TR1=0$. *Timer* 0 dalam mode 3 menetapkan TL0 dan TH0 sebagai dua *counter* terpisah. TL0 menggunakan kontrol bit *timer* 0 yaitu C/T, GATE, TR0, \overline{INT}_0 , dan TF0. TH0 ditetapkan sebagai fungsi *timer*.

Mode 3 diperlukan untuk aplikasi yang membutuhkan *timer/counter* ekstra 8 bit. Dengan *timer* 0 dalam mode 3, mikrokontroler 8031 seperti memiliki 3 *timer/counter*. Saat *timer* 0 dalam mode 3, *timer* 1 dapat dihidupkan atau dimatikan, atau dapat digunakan oleh port serial sebagai pembangkit baud rate.

➤ **Menyetel *Timer***

Tabel 2.3 sampai 2.6 memberikan beberapa nilai bagi TMOD yang dapat digunakan untuk menyetel *timer* dalam mode yang berbeda.

Diasumsikan hanya satu *timer* yang digunakan. Jika diinginkan untuk menjalankan *timer* 0 dan *timer* 1 secara bersamaan dalam beberapa mode, maka nilai TMOD dalam *timer* 0 (Tabel 2.3) harus di-OR-kan dengan nilai TMOD dalam *timer* 1 (Tabel 2.5). Sebagai contoh, jika diinginkan untuk menjalankan *timer* 0 dalam mode 1 (kontrol eksternal) dan *timer* 1 dalam mode 2 *counter* (kontrol internal), maka nilai yang harus diisikan pada TMOD adalah 69H (nilai 09H dari Tabel 2.3 di-OR-kan dengan 60H dari Tabel 2.6).

Pada kontrol internal, *timer* dihidup-matikan dengan men-*set* atau men-*clear* bit TR0 (untuk *timer* 0) dan TR1 (untuk *timer* 1). Hal ini dilakukan lewat kontrol secara *software*. Sedangkan, pada kontrol eksternal, *timer* dihidup-matikan dengan memberikan logika 0 pada pena \overline{INT}_0 (untuk *timer* 0) dan \overline{INT}_1 (untuk *timer* 1). Hal ini dilakukan lewat kontrol secara *hardware*.

Tabel 2.3. Fungsi *Timer 0*.

Mode	Fungsi <i>Timer 0</i>	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit <i>timer</i>	00H	08H
1	16 bit <i>timer</i>	01H	09H
2	8 bit <i>auto reload</i>	02H	0AH
3	Dua 8 bit <i>timer</i>	03H	0BH

Sumber : Malik, 1997 : 24

Tabel 2.4. Fungsi *Counter 0*.

Mode	Fungsi <i>Counter 0</i>	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit <i>timer</i>	04H	0CH
1	16 bit <i>timer</i>	05H	0DH
2	8 bit <i>auto reload</i>	06H	0EH
3	Satu 8 bit <i>counter</i>	07H	0FH

Sumber : Malik, 1997 : 24

Tabel 2.5. Fungsi *Timer 1*.

Mode	Fungsi <i>Timer 1</i>	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit <i>timer</i>	00H	80H
1	16 bit <i>timer</i>	10H	90H
2	8 bit <i>auto reload</i>	20H	A0H

Sumber : Malik, 1997 : 24

Tabel 2.6. Fungsi *Counter 1*.

Mode	Fungsi <i>Timer 0</i>	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit <i>timer</i>	40H	C0H
1	16 bit <i>timer</i>	50H	D0H
2	8 bit <i>auto reload</i>	60H	E0H

Sumber : Malik, 1997 : 24

2.3 Sensor

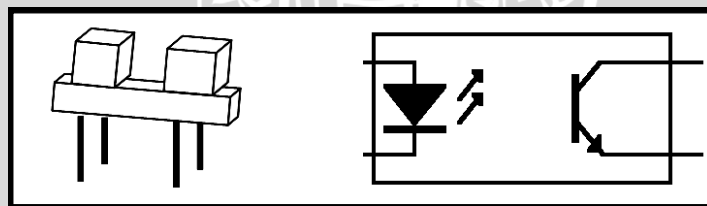
Sensor merupakan suatu peralatan yang digunakan untuk mendeteksi suatu hal tanpa adanya kontak fisik. Akhir-akhir ini, sensor banyak digunakan untuk memantau secara terus-menerus perubahan yang terjadi pada sesuatu yang dipantau. Pada perancangan ini sensor yang digunakan adalah baling-baling dengan memanfaatkan casing pada meteran PDAM.

2.4 Optocoupler

Optocoupler adalah alat yang dipakai untuk mengkopel cahaya dari suatu sumber ke detektor tanpa adanya perantara. Oleh karena itu piranti ini sering disebut dengan *optoisolator/optocoupler*.

Sinyal listrik (arus) pada *input* diubah menjadi sinyal optik dengan menggunakan sumber cahaya (biasanya LED). Sinyal optik tersebut akan diterima oleh detektor untuk diubah kembali menjadi sinyal listrik.

Umumnya *optocoupler* dipakai untuk mengisolasi sinyal listrik yang ada pada *input* dan *output* sehingga dapat digunakan transmisi sinyal antar rangkaian. Pada perancangan ini *optocoupler* digunakan untuk menghitung banyaknya putaran pada baling-baling sebagai sinyal masukan. Bentuk dan diagram rangkaian *optocoupler* ditunjukkan dalam Gambar 2.12.



Gambar 2.12. *Optocoupler* dan Diagram Rangkaiannya.
Sumber : Motorola Semiconductor Technical Data 7:2.

2.4.1 Prinsip kerja *Optocoupler*

Optocoupler memiliki sebuah LED yang dihubungkan dengan masukan dan sebuah transistor yang dihubungkan dengan keluaran.

Prinsip kerja dari *optocoupler* adalah:

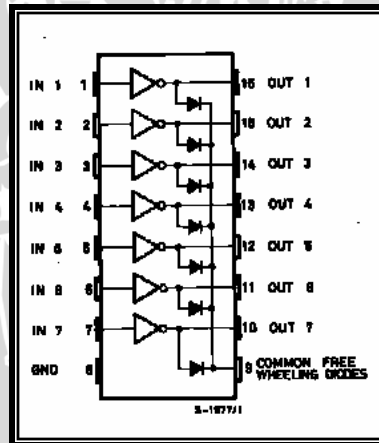
- Jika antara transistor dan LED terhalang oleh suatu benda maka transistor akan off sehingga *output* dari kolektor akan berlogika *high*.

- Sebaliknya jika antara transistor dan LED tidak terhalang oleh suatu benda maka transistor akan on sehingga outputnya akan berlogika *low*.

Ketika suatu benda memotong cahaya yang dipancarkan oleh LED, sehingga akan mengalir arus yang melewati resistor *output*. Karena tegangan *output* sama dengan tegangan suplai dikurangi tegangan yang melewati resistor. Sehingga ketika masukan berubah, intensitas cahaya berubah maka tegangan pada keluaran juga berubah.

2.5 ULN2004

ULN2001A, ULN2002A, ULN2003 dan ULN2004A adalah larik darlington yang memiliki kemampuan tegangan tinggi, arus tinggi dan masing-masing berisi tujuh buah *open collector* darlington dengan *common emitter*. Mampu mengalirkan arus sebesar 500mA dan dapat menahan arus puncak sebesar 600mA. Dilengkapi dengan diode peredam untuk mencatu beban induktif. Komponen ini sangat berguna mencatu berbagai macam jenis beban antara lain solenoid, motor DC dan lain sebagainya. Untuk ULN2004 ini sudah dikemas dalam bentuk IC, jadi dalam penggunaannya lebih mudah dan sederhana. Pada perancangan ini ULN2004 berfungsi untuk membuka dan menutup *valve*. Diagram rangkaian ULN2004 ditunjukkan dalam Gambar 2.13.

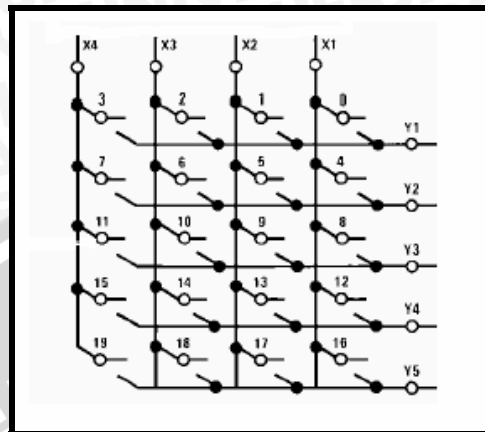


Gambar 2.13. Diagram rangkaian ULN2004.
Sumber : Data Sheet ULN2003.

2.6 Tombol Masukan / Keypad

Tombol masukan merupakan rangkaian yang digunakan untuk memberikan data masukan, yang mana data tersebut diberikan melalui penekanan tombol/*tuts* yang terdapat pada papan masukan itu sendiri. Ketika tombol masukan ditekan, rangkaian enkoder akan menerjemahkan penekanan pada salah satu tombol tersebut ke dalam bentuk biner. Sehingga jumlah jalur yang akan masuk ke rangkaian berikutnya dapat

diperkecil. Kode-kode biner ini yang akan dikirimkan ke unit pengolah. Diagram rangkaian ditunjukkan dalam Gambar 2.14.

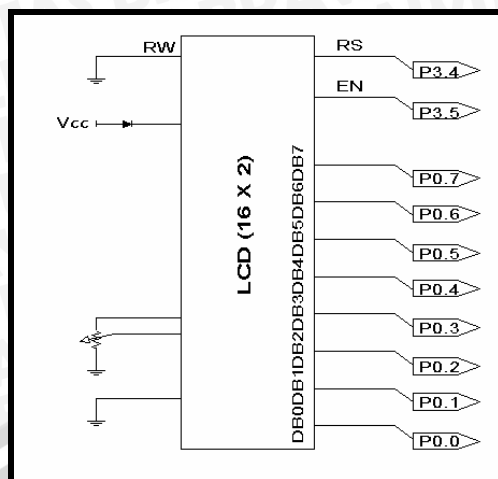


Gambar 2.14. Susunan keypad matrik 4 x 4.
Sumber : Fairchild Data Sheets.

2.7 Rangkaian LCD

LCD adalah merupakan komponen optoelektronik yaitu komponen-komponen yang dikerjakan atau dipengaruhi oleh sinar (*optolistrik*), komponen-komponen pembangkit cahaya (*light emitting*) dan komponen-komponen yang mempengaruhi akan mengubah sinar. LCD terbuat dari bahan kristal cair yang merupakan suatu kompon organik yang mempunyai sifat optik seperti benda padat meskipun bahan tetap cair. Contoh kompon itu adalah *Kolesteril nonanoat* dan *P-azoxyanisole*.

Sel kristal cair terdiri dari selapis bahan kristal cair yang diapit antara gelas tipis dengan elektroda lapisan logam transparan yang diendapkan (*deposited*) pada bagian dalam gelas. Kedua keping gelas juga transparan. Sel ini disebut sel tipe transmitif. Bila hanya sebuah lapisan gelas yang transparan sedang yang lain mempunyai lapisan reflektif, sel ini disebut tipe reflektif. kedua tipe ini. Bila tidak diaktifkan, tipe transmitif maka sel itu meneruskan sinar dari belakang atau dari pinggir dalam garis lurus. Dalam hal ini sel tidak nampak cemerlang. Bila diaktifkan, sinar yang datang dipendarkan ke depan dan difusi sel nampak cemerlang dibawah cahaya kamar yang terang.



Gambar 2.15. Rangkaian LCD.

Sumber : El-TECH Electronics, LCD Module User Manual, 1987.

Untuk display dalam bentuk tampilan tulisan yang digunakan dalam perencanaan alat ini adalah LCD 16 x 2 digit ditunjukkan dalam Gambar 2.15. Untuk penggunaan LCD terlebih dahulu harus diinisialisasikan terlebih dahulu menurut instruksi yang terdapat di LCD. Display difungsikan sebagai alamat yang dihubungkan dengan bus data, dan dengan bantuan *software* maka dapat ditampilkan karakter yang diinginkan pada display, dengan mengontrol pin E, R/W, dan RS.

2.8 TCP/IP

Dalam arti yang sederhana TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah nama keluarga protokol jaringan. Protokol adalah sekelompok aturan yang harus diikuti oleh perusahaan-perusahaan dan produk-produk *software* agar dapat saling berkomunikasi antara satu dengan yang lainnya. Pada awalnya internet memang sudah dibuat dengan menggunakan TCP/IP yang memungkinkan sistem apapun yang terhubung ke dalamnya bisa berkomunikasi dengan sistem lain tanpa mempedulikan bagaimana masing-masing sistem bekerja. Istilah TCP/IP mengacu kepada seluruh keluarga protokol yang dirancang untuk mentransfer informasi sepanjang jaringan. Dan dirancang untuk menjadi komponen perangkat lunak dari suatu jaringan. TCP/IP merupakan dua protokol yang berbeda, dimana TCP bertanggung jawab memecah informasi ke dalam beberapa paket, sedangkan IP bertanggung jawab untuk mentransfer paket-paket tersebut sesuai tujuannya. Kemudian TCP yang ada pada tujuan akan menyatukan kembali paket-paket tersebut menjadi untai yang sebenarnya. Layanan

repository.ub.ac.id

dalam TCP/IP yang berbeda dikelompokkan menurut fungsi-fungsinya. Protokol-protokol transport mengendalikan pergerakan data antara dua mesin, dan mencakup :

1. TCP (*Transmission Control Protokol*) Protokol ini bersifat *Connection-based*, artinya kedua mesin pengirim dan penerima tersambung dan berkomunikasi satu sama lain sepanjang waktu.
2. UDP (*User Datagram Protokol*) Protokol ini bersifat *Unconnectionless* atau tanpa koneksi, dimana data dikirim tanpa kedua mesin penerima dan pengirim saling berhubungan. Dan kita juga tidak dapat mengetahui apakah data/pesan tersebut telah atau belum atau tidak sampai ke penerima. Di samping itu ada pula protokol-protokol *routing* untuk menangani pengalamatan (*addressing*) data dan menentukan jalur terbaik untuk mencapai tujuan. Protokol-protokol tersebut juga bertanggung jawab memecah informasi yang berukuran besar dan menyusunnya kembali pada tujuan. Protokol-protokol tersebut antara lain :
 - IP (*Internet Protokol*) menangani transmisi data yang sebenarnya.
 - ICMP (*Internet Control Message Control Protokol*) menangani informasi status untuk IP, seperti *error* dan perubahan-perubahan dalam perangkat keras jaringan yang mempengaruhi *routing* (penentuan jalur).
 - RIP (*Routing Information Protokol*) dan OSPF (*Open Shortest-Path First*), yaitu satu dari berbagai protokol yang menentukan metode *routing* terbaik untuk menyampaikan data.

2.9 World Wide Web

Pada awalnya internet adalah sebuah proyek yang dimaksudkan untuk menghubungkan para ilmuwan dan peneliti di Amerika, namun saat ini telah tumbuh menjadi media komunikasi global yang dipakai semua orang dimuka bumi. Pertumbuhan ini membawa beberapa masalah penting mendasar, di antaranya kenyataan bahwa internet tidak diciptakan pada jaman *graphical user interface* (GUI) seperti saat ini. Internet dimulai pada masa di mana orang masih menggunakan alat-alat akses yang tidak *user-friendly* yaitu terminal dengan berbasis teks serta perintah-perintah *command line* yang panjang-panjang serta susah diingat, sangat berbeda dengan apa yang ada pada saat ini yang hanya mengklikkan tombol *mouse* pada layar grafik berwarna. Popularitas Internet mulai berkembang pesat setelah standar baru diperkenalkan kepada masyarakat yaitu HTTP (*Hypertext Transfer Protokol*) dan HTML (*Hypertext Markup Language*) sehingga pengaksesan internet melalui protocol TCP/IP menjadi lebih mudah dari

UNIVERSITAS BRAWIJAYA

sebelumnya. Dengan standar baru tersebut maka informasi di internet dapat disajikan secara visual dan lebih menarik. Pemunculan HTTP dan HTML membuat orang mengenal istilah baru dalam dunia internet yang sangat populer yang diidentikkan dengan internet itu sendiri yaitu *World Wide Web* (WWW) atau WEB.



BAB III

METODOLOGI

Dalam metodologi akan diuraikan cara-cara yang dilakukan dalam perencanaan dan pembuatan “*Monitoring Sistem Pasokan Bahan Bakar Minyak Pada SPBU Melalui Jaringan Internet*” serta penyusunan laporannya. Adapun metodologi penelitian skripsi yang digunakan adalah sebagai berikut :

1. Studi Literatur.
2. Perancangan Alat.
3. Pembuatan Alat.
4. Pengujian Alat.
5. Pengambilan Kesimpulan.

3.1 Studi Literatur

Pada tahapan ini dilakukan kegiatan mempelajari berbagai literatur yang menunjang dan berhubungan dengan perancangan dan pembuatan alat ini. Jenis data yang digunakan dalam desain ini meliputi data sekunder dan data primer. Data sekunder adalah data yang diperoleh dari *data sheet book* masing-masing komponen, sedangkan data primer diperoleh dari hasil pengujian.

3.2 Perancangan Alat

Berdasarkan studi literatur, tahap selanjutnya adalah perencanaan alat. Hal-hal yang dilakukan dalam perencanaan yaitu penentuan spesifikasi sistem yang dirancang, penyusunan blok diagram sistem untuk mempermudah pemahaman mengenai alur kerja alat yang akan dibuat serta pembuatan skema rangkaian. Pemilihan komponen perangkat keras berdasarkan dalam komponen yang umum dan mudah didapatkan di pasaran lokal, sedangkan untuk perancangan perangkat lunak dimulai dengan membuat diagram alir. Perencanaan perangkat lunak meliputi pengaturan protokol komunikasi, pembacaan data dari mikrokontroler yang akan dikirim ke PC untuk ditampilkan secara visual untuk kemudahan operator menjalankan alat ini.

3.3 Pembuatan Alat

Dalam pembuatan alat dilakukan sesuai dengan perencanaan. Tahap pembuatan alat dimulai dengan pembuatan unit rangkaian per blok pada PCB. Selanjutnya tiap blok rangkaian dilakukan pengujian awal. Kemudian dilakukan penggabungan tiap blok menjadi suatu blok rangkaian sesuai dengan perencanaan. Untuk proses pembuatan PCB dibuat dengan menggunakan perangkat lunak Protel 98 PCB. Setelah pemasangan komponen tiap blok selesai, dilakukan penggabungan antara blok rangkaian PCB. Sedangkan pembuatan perangkat lunak dengan mengimplementasikan alur program yang telah direncanakan yaitu dengan menggunakan bahasa *Visual Basic* dan *assembly*.

3.4 Pengujian Alat

Adapun tujuan pengujian alat adalah untuk memastikan bahwa sistem yang dibuat telah berjalan, yaitu dengan menguji rangkaian perblok dan keseluruhan rangkaian. Metode pengujian alat adalah sebagai berikut:

Pengujian rangkaian sensor.

Bertujuan untuk mengetahui apakah rangkaian sensor dengan menggunakan baling-baling bisa bekerja sesuai dengan perencanaan. Metode pengujian dengan melakukan praktek untuk mengetahui jumlah volume yang telah dikeluarkan, yang mana jumlah volume yang dikeluarkan akan ditampilkan pada monitor komputer.

Pengujian rangkaian *driver*.

Bertujuan untuk mengetahui apakah rangkaian *driver* dengan menggunakan ULN2004 bisa bekerja sesuai dengan perencanaan. Metode pengujian dengan memasukkan *password* melalui *keypad* dan ditampilkan pada *LCD*, dan *password* sebelumnya diset terlebih dahulu dengan menggunakan *Personal Computer* (PC).

Pengujian perangkat lunak.

Bertujuan untuk mengetahui apakah program dapat berjalan sesuai dengan perencanaan. Metode pengujian dengan menggunakan perangkat lunak *Visual Basic*.

Pengujian hubungan komunikasi serial antara komputer dengan mikrokontroler.

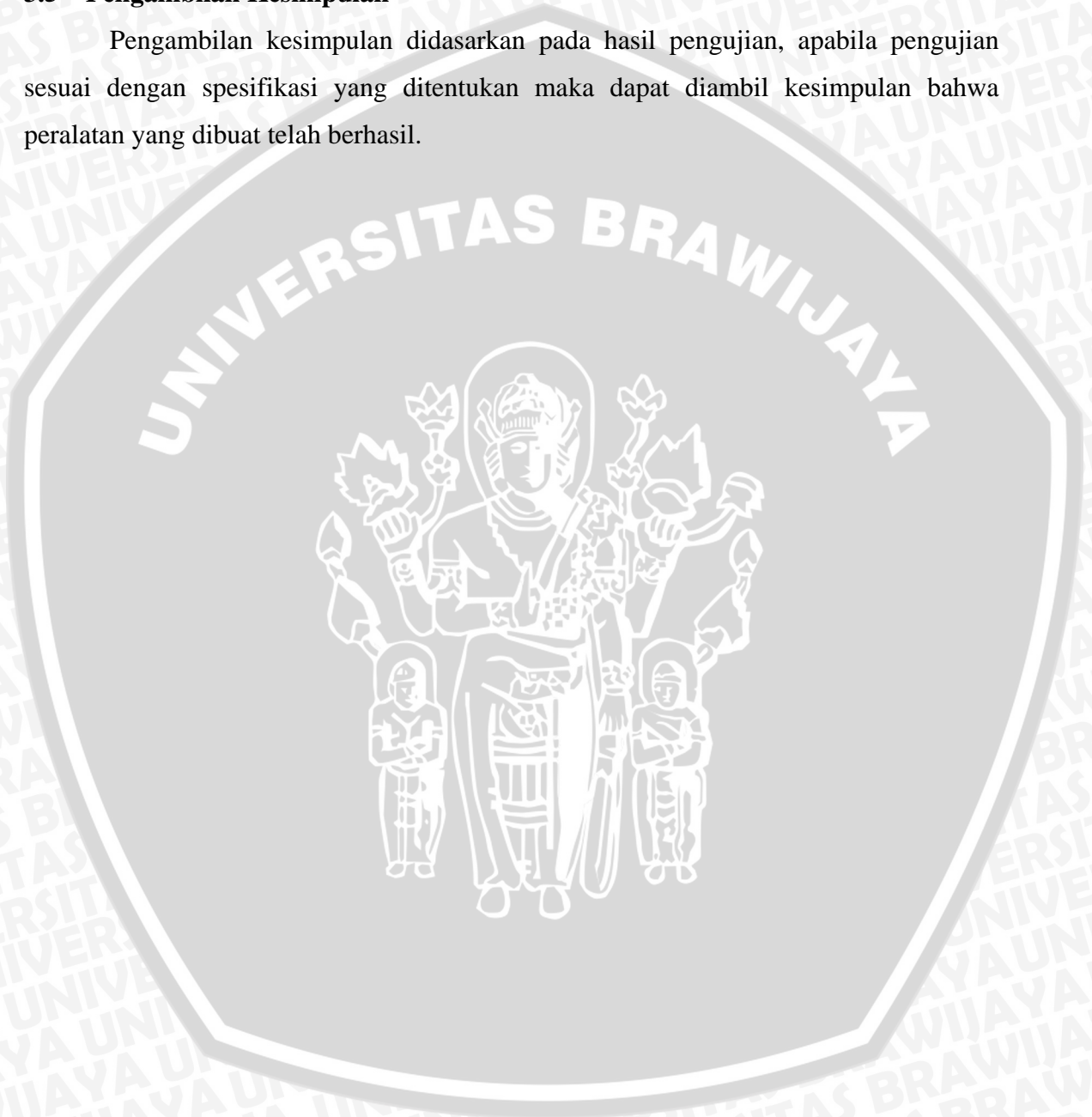
Bertujuan untuk mengetahui adanya komunikasi antara komputer dan mikrokontroler.

Pengujian keseluruhan sistem

Bertujuan untuk mengetahui apakah sistem dapat bekerja sesuai dengan perencanaan. Metode yang digunakan adalah dengan mengaplikasikan alat secara keseluruhan.

3.5 Pengambilan Kesimpulan

Pengambilan kesimpulan didasarkan pada hasil pengujian, apabila pengujian sesuai dengan spesifikasi yang ditentukan maka dapat diambil kesimpulan bahwa peralatan yang dibuat telah berhasil.



BAB IV

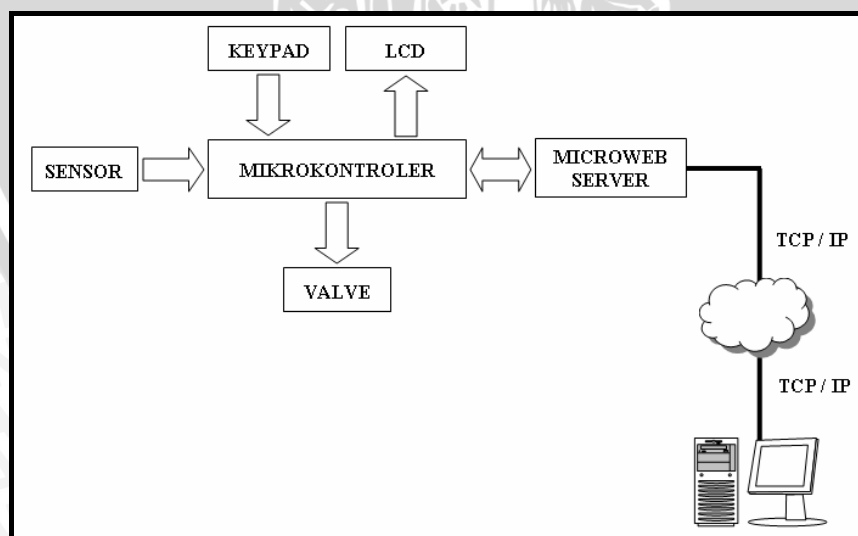
PERANCANGAN DAN PEMBUATAN ALAT

Dalam bab ini dijelaskan tentang perancangan *Monitoring Sistem Pasokan Bahan Bakar Minyak Pada SPBU Melalui Jaringan Internet* beserta prinsip kerja dari keseluruhan sistem. Perancangan sistem meliputi perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*). Perancangan perangkat keras terdiri atas mekanik sensor baling-baling dan sistem mikrokontroler AT89C51. Sedangkan perancangan perangkat lunak dibuat dengan menggunakan *kompiler bahasa Assembler*. Secara garis besar pokok bahasan pada bab ini adalah :

- Perancangan perangkat keras (*hardware*).
- Perancangan perangkat lunak (*software*).

4.1 Blok Diagram Rangkaian.

Blok diagram rangkaian merupakan salah satu bagian terpenting dalam perancangan sistem ini, karena dapat diketahui prinsip kerja keseluruhan rangkaian. Keseluruhan blok diagram rangkaian tersebut menunjukkan gambaran umum sistem yang dapat difungsikan atau sistem yang bekerja sesuai dengan perancangan. Blok diagram sistem secara keseluruhan dapat dilihat dalam Gambar 4.1.



Gambar 4.1. Blok Diagram Sistem
Sumber : Perancangan

4.2 Prinsip Kerja Sistem.

Pada prinsipnya kerja dari sistem ini adalah memonitor sistem pemasokan Bahan Bakar Minyak pada SPBU dengan cara melihat volume pada tangki SPBU yang ditampilkan pada layar monitor komputer yang bisa diakses melalui jaringan internet. Dimana sensor yang berfungsi untuk mengukur jumlah bahan bakar minyak yang dikeluarkan akan memberikan sinyal masukan ke mikrokontroler yang selanjutnya akan diproses oleh mikrokontroler. Status sensor akan memberikan data dan ditampilkan pada komputer yang bisa diakses melalui jaringan internet dengan menggunakan microweb server.

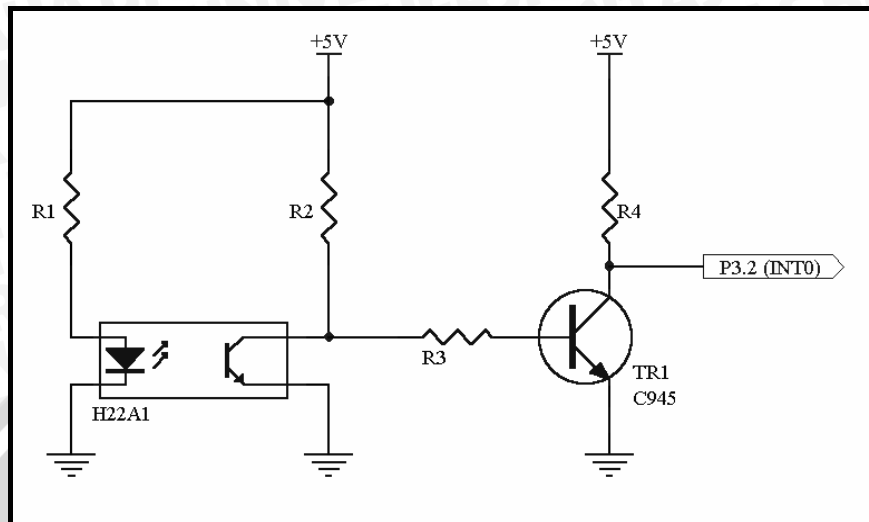
Apabila volume pada tangki sudah berada pada tahap pengisian, dengan melihat data yang dihasilkan maka secara otomatis Depo Pertamina akan mengirimkan bahan bakar minyak ke SPBU dan memberikan *password* pada sopir untuk proses pengisian. *Password* yang diberikan harus dimasukkan pada mikrokontroler melalui *keypad* yang akan ditampilkan pada LCD yang berfungsi untuk membuka *valve*. Setelah *valve* terbuka proses pengisian baru bisa dilaksanakan. Pemberian *password* pada sopir juga untuk mengetahui ketepatan waktu pengiriman dari Depo Pertamina ke SPBU yang dituju.

4.3 Rangkaian Sensor *Flowmeter*.

Sensor pemakaian minyak bekerja secara elektromekanis. Sensor mekanik yang dipergunakan adalah meter air standar. Pada meter tersebut terdapat baling-baling yang akan berputar jika minyak mengalir pada pipa. Poros baling-baling tersebut dihubungkan ke suatu piringan bercelah. Piringan bercelah tersebut ditempatkan di antara sebuah LED (*Light Emitting Diode*) sebagai pembangkit cahaya dan fototransistor sebagai sensor pada sebuah *optocoupler*.

Optocoupler ini menggunakan *Optocoupler* tipe U jenis H22A1. Ketika minyak mengalir, baling-baling dalam pipa berputar. Bersamaan dengan itu piringan celah juga berputar karena memiliki poros yang sama. Jika cahaya LED melewati celah dan mengenai fototransistor maka fototransistor akan saturasi, sehingga arus mengalir dari kolektor ke emitor. Pada kondisi ini fototransistor hubung singkat dan keluaran *optocoupler* akan *low*. Sebaliknya ketika cahaya LED terhalang piringan tidak ada arus basis pada fototransistor, sehingga tidak ada arus antara kolektor dan emitor. Pada kondisi ini keluaran *optocoupler* menjadi *high*. Untuk memperkuat arus dan memperbaiki bentuk pulsa, maka keluaran *optocoupler* dihubungkan ke rangkaian *switching*.

Keluaran rangkaian *switching* yaitu pada kolektor TR1 dihubungkan ke Port 3.2 (INT0) mikrokontroler.



Gambar 4.2. Rangkaian Sensor *Flowmeter*
Sumber : Perancangan

Untuk mendapatkan arus pada kolektor I_C sebesar 3,8 mA, maka LED harus diberi tegangan maju I_F sebesar 30 mA. Sehingga nilai resistor R_1 dapat diperoleh :

$$\begin{aligned} R_1 &= V_{CC} / I_F \\ &= 5 / (30 \cdot 10^{-3}) \\ &= 166,67 \Omega \end{aligned}$$

Sesuai standar nilai yang ada dipasaran maka nilai R_1 yang mendekati sebesar 180 Ω . Agar transistor bisa bekerja dengan baik, maka arus yang mengalir pada R_2 hendaknya lebih kecil dari arus kolektor I_C . Sehubungan dengan hal tersebut I_{R2} dapat ditentukan sebesar 5 mA, maka nilai resistor R_2 yang harus dipergunakan sebesar :

$$\begin{aligned} R_2 &= (V_{CC} - V_{CE \text{ sat}}) / I_{R2} \\ &= (5 - 0,25) / 5 \text{ mA} \\ &= 950 \Omega \end{aligned}$$

Sesuai standar nilai yang ada dipasaran maka nilai R_2 yang mendekati sebesar 1 K Ω . Pada saat fototransistor *cut off*, arus akan mengalir menuju basis pada transistor C945 melalui resistor basis R_3 . Agar transistor TR2 bekerja dengan baik, arus yang masuk ke basis tidak melebihi nilai maksimumnya yaitu sebesar 20 mA. Maka ditetapkan nilai

arus kerja I_{R3} sebesar $100 \mu\text{A}$ sehingga nilai R_3 yang harus dipergunakan adalah sebesar :

$$\begin{aligned}R_2 + R_3 &= (V_{CC} - V_{BE(C945)}) / I_{R3} \\1000 + R_3 &= (5 - 0,62) / (100 \cdot 10^{-6}) \\R_3 &= 43800 - 1000 \\&= 42800 \Omega\end{aligned}$$

Sesuai standar nilai yang ada dipasaran maka nilai R_3 yang mendekati sebesar $39 \text{K}\Omega$, sehingga nilai I_{R3} yang sebenarnya adalah :

$$\begin{aligned}I_{R3} = I_{B2} &= (V_{CC} - V_{BE(C945)}) / (R_2 + R_3) \\&= (5 - 0,62) / (1000 + 39000) \\&= 4,38 / 40000 \\&= 109,5 \mu\text{A}\end{aligned}$$

Maka arus yang dapat dilewati pada saat transistor dalam keadaan saturasi adalah :

$$\begin{aligned}I_C &= I_{B2} \cdot h_{FE} \\&= (109,5 \cdot 10^{-6}) \cdot 50 \\&= 5,5 \text{ mA}\end{aligned}$$

Pada saat transistor C945 *cut off*, arus akan mengalir menuju port P3.2 melalui resistor kolektor R_4 . Agar mikrokontroler bekerja dengan baik, arus yang masuk ke port tidak melebihi nilai maksimumnya yaitu sebesar (I_{IL}) 10 mA . Maka ditetapkan nilai arus kerja I_{R4} sebesar 5 mA sehingga nilai R_4 yang harus dipergunakan adalah sebesar :

$$\begin{aligned}R_4 &= V_{CC} / I_{R4} \\&= 5 / (5 \cdot 10^{-3}) \\&= 1000 \Omega\end{aligned}$$

4.4 Driver

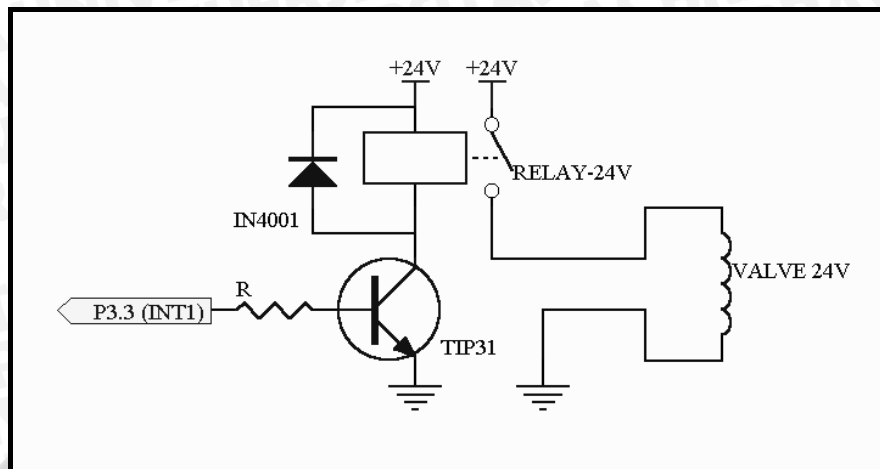
Untuk menggerakkan sebuah relay diperlukan rangkaian *driver*, karena arus keluaran dari AT89C51 terlalu kecil. Dari data sheet diketahui bahwa arus maksimum keluaran adalah :

I_{OL} (Arus keluaran pada logika 0) = $1,6 \text{ mA}$

I_{OH} (Arus keluaran pada logika 1) = $-60 \mu\text{A}$

V_{OH} minimum = $2,4 \text{ V}$

Rangkaian *driver* valve ditunjukkan dalam Gambar 4.3.



Gambar 4.3. Rangkaian *Driver* Valve
Sumber : Perancangan

Dari percobaan diketahui bahwa $R_{\text{relay}} = 530 \Omega$, sehingga :

$$I_{\text{relay}} = 24 \text{ V} / 530 \Omega = 45 \text{ mA}$$

$$I_{\text{C}} = I_{\text{relay}} = 45 \text{ mA}$$

Jadi, untuk menggerakkan relay dibutuhkan arus minimal 45 mA. Transistor dibutuhkan untuk menguatkan arus, transistor yang digunakan adalah transistor NPN TIP31 yang mempunyai $h_{fe \text{ typical}} = 65$ dan $V_{\text{BE}} = 0,7 \text{ V}$. Dari data yang diperoleh maka R_{b} dapat ditentukan dengan menggunakan persamaan :

$$h_{fe} = \frac{I_{\text{C}}}{I_{\text{b}}}, I_{\text{b}} = \frac{I_{\text{C}}}{h_{fe}} = \frac{45 \text{ mA}}{65} = 0,69 \text{ mA}$$

$$R_{\text{b}} = \frac{V_{\text{OH}} - V_{\text{BE}}}{I_{\text{b}}} = \frac{2,4 - 0,7}{0,69 \cdot 10^{-3}} = 2,46 \cdot 10^3 \Omega = 2460 \Omega$$

$$\approx 2\text{K}2\Omega$$

Sebuah dioda dipasang paralel dengan relay untuk menghubungkan arus tersimpan pada induktor relay.

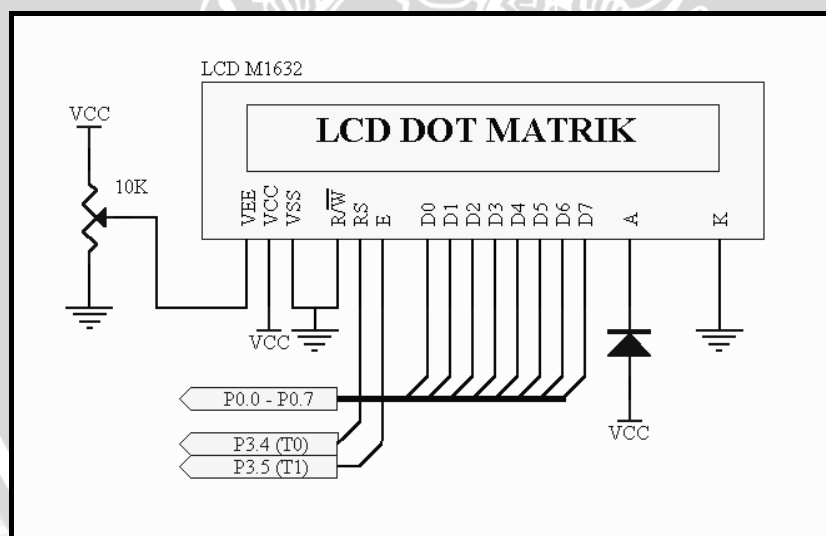
4.5 LCD (*Liquid Crystal Display*)

Tampilan dalam perancangan ini digunakan suatu tampilan berupa LCD (*Liquid Crystal Display*), yang mana LCD ini memudahkan untuk melihat secara visual tampilan dari hasil komposisi bahan bakar dan udara. Dalam perancangan ini digunakan LCD untuk tampilan 2x16 karakter.

Tampilan dalam perancangan ini dipergunakan LCD *type* M1632. LCD ini memiliki ciri-ciri sebagai berikut :

- a. LCD ini terdiri atas 32 karakter dengan 2 baris masing-masing 16 karakter dengan *display* dot matrik 5x7.
- b. Karakter generator ROM dengan 192 tipe karakter.
- c. Karakter generator RAM dengan 8 bit karakter.
- d. 80x8 bit *display* data RAM.
- e. Dapat dihubungkan ke MCU 8 atau 4.
- f. Dilengkapi fungsi tambahan; *display clear*, *cursor home*, *display on/off*, *cursor on/ off*, *display character blink*, *cursor shift*, *display shift*.
- g. Internal data.
- h. Internal otomatis, reset pada saat *power-on*.
- i. Tegangan + 5 volt PSU tunggal.

Gambar hubungan antara LCD dengan mikrokontroler dapat dilihat dalam Gambar 4.5.



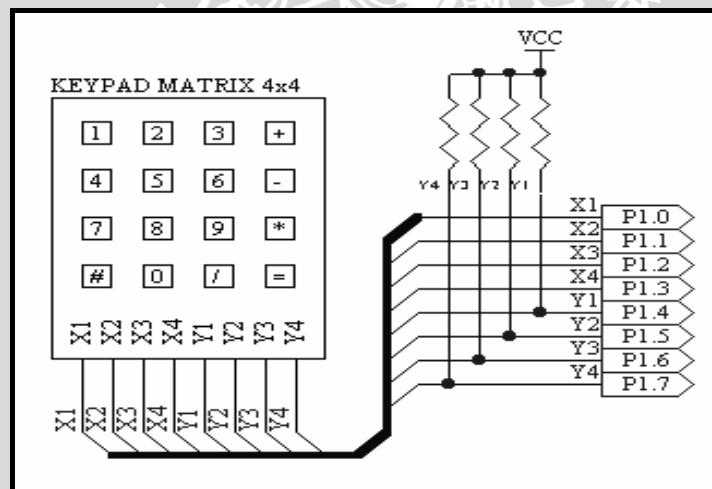
Gambar 4.4. Rangkaian LCD
Sumber : Perancangan

Liquid Crystal Display ini mempunyai konsumsi daya yang relatif rendah dan terdapat sebuah kontroler CMOS di dalamnya. Kontroler tersebut sebagai pembangkit dari karakter ROM/ RAM dan *display* data RAM. Semua fungsi tampilan di kontrol oleh suatu instruksi dan modul LCD dapat dengan mudah untuk diinterfacekan dengan

mikrokontroler. Masukan yang diperlukan untuk mengendalikan modul ini berupa bus data yang masih termultiplek dengan bus alamat serta 3 bit sinyal kontrol. Sementara pengendalian dot matrik LCD dilakukan secara internal oleh kontroler yang sudah ada pada modul LCD.

4.6 Perencanaan Rangkaian Keypad

Keypad yang digunakan berupa *keypad* matriks 4 x 4 yang terdiri dari 4 kolom dan 4 baris. *Keypad* matrik ini bekerja dengan menggunakan prinsip *scanning* pada baris dan kolom. Kondisi logika pada port *keypad* adalah logika 1 pada setiap bitnya. Saat salah satu tombol dari *keypad* ditekan, baris dan kolom yang berhubungan akan terhubung ke ground sehingga kondisi logika baris dan kolom tersebut akan berlogika 0. Pada perancangan, X1 sampai X4 pada *keypad* dihubungkan ke P1.0 sampai P1.3 pada mikrokontroler dan sementara Y1 sampai Y4 pada *keypad* dihubungkan ke P1.4 sampai P1.7 pada mikrokontroler. Perencanaan rangkaian *keypad* dapat dilihat dalam Gambar 4.2.

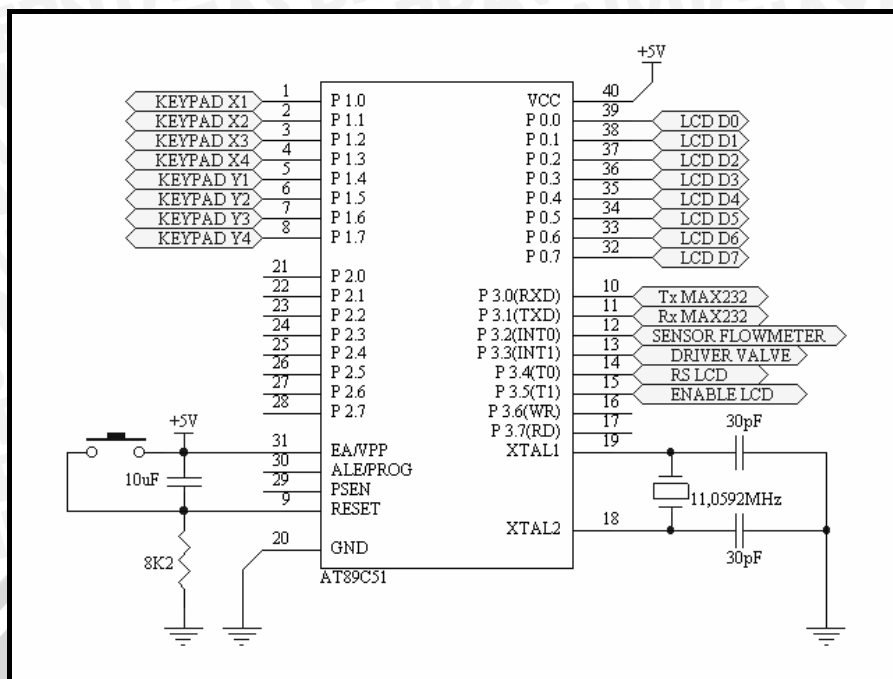


Gambar 4.5. Rangkaian Keypad
Sumber : Perancangan

4.7 Mikrokontroler AT89C51

Mikrokontroler pada perancangan ini di desain Sebagai otak dari pengolah data dan pengontrol alat, *port-port* AT89C51 dihubungkan pada rangkaian pendukung, penyemat-penyemat mikrokontroler yang digunakan seperti ditunjukkan dalam Gambar 4.7 dan berikut konfigurasi pin mikrokontroler yang digunakan dalam perancangan :

1. P0.0 – P0.7 (*Port 0*) digunakan sebagai penampilan data pada LCD (*Liquid Crystal Display*).
2. P3.4 (T0) digunakan sebagai data RS LCD.
3. P3.5 (T1) digunakan sebagai kontrol *enable signal* LCD.
4. P1.0 – P1.7 (*Port 1*) digunakan sebagai masukan dari *keypad*.
5. P3.2 (INT0) digunakan sebagai masukan dari rangkaian *optocoupler*.
6. P3.3 (INT1) digunakan sebagai output dari mikrokontroler yang dihubungkan sebagai input dari rangkaian *driver* untuk membuka/menutup *valve*.
7. P3.0 – P3.1 (*Port 3*) merupakan *port I/O* dimana pin yang digunakan adalah :
 - a. P3.0 (TxD) digunakan untuk *port Tx serial output*, *port* ini digunakan oleh mikrokontroler untuk mengirim data pada komputer (data dari komputer yang telah diubah level tegangannya menjadi level TTL oleh rangkaian RS 232).
 - b. P3.1 (RxD) digunakan sebagai *port Rx serial input*, *port* ini digunakan oleh mikrokontroler untuk menerima data dari komputer (data dari komputer yang telah diubah level tegangannya menjadi level TTL oleh rangkaian RS 232).
6. X1 dan X2 digunakan sebagai input dari rangkaian osilator kristal. Rangkaian osilator kristal terdiri dari kristal osilator 11,0592 MHz, kapasitor C1 dan C2 yang masing-masing bernilai 30 pF, akan membangkitkan pulsa *clock* yang menjadi penggerak bagi seluruh operasi internal MCU (aplication Note AT89C51 In-Circuit programing).
- 7 VCC dihubungkan dengan tegangan sebesar +5V sesuai dengan tegangan operasi *chip* tunggal yang diizinkan dalam *data sheet*.
- 8 GND dihubungkan ke *ground* catu daya.
- 9 Reset digunakan untuk mereset program kontrol MCU, dimana MCU memiliki masukan aktif *high*.



Gambar 4.6. Rangkaian Mikrokontroler AT89C51
 Sumber : Perancangan

4.8 RS 232 Driver

Komponen IC ICL232 merupakan komponen yang biasa digunakan untuk komunikasi RS 232 seperti yang ditunjukkan dalam Gambar 4.7. Untuk keluaran dari *port* komputer, tegangan +3 V sampai +12 V (*space*) akan dirubah menjadi 0 V atau logika rendah, sedang tegangan -3 V sampai -12 V (*mark*) dirubah menjadi 5 Volt atau logika tinggi. Sedangkan untuk masukan ke *port* komputer, tegangan masukan rendah 0 V dirubah menjadi -10 V dan tegangan masukan tinggi 5 V dirubah menjadi +10 V. Penggunaan kapasitor C_1 dan C_2 adalah sebagai pengali dua tegangan dari +5 V menjadi +10 V, C_3 dan C_4 sebagai pembalik tegangan +10 V menjadi -10 V, keempat kapasitor ini bekerja dengan rangkaian *internal*. Catu daya untuk rangkaian komunikasi serial ini sebesar 5 V.

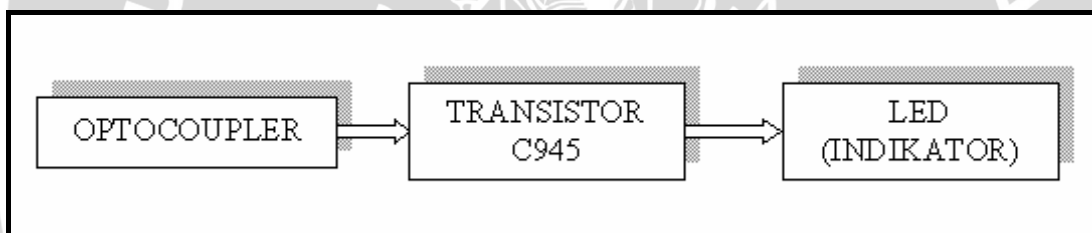
BAB V

PENGUJIAN SISTEM

Dalam bab ini membahas pengujian alat yang telah dirancang dan telah direalisasikan. Pengujian dilakukan dengan tujuan untuk mengetahui bahwa alat yang dirancang telah bekerja sesuai dengan spesifikasi yang diinginkan.

5.1 Pengujian Rangkaian *Optocoupler*

Tujuan pengujian rangkaian optocoupler adalah untuk mengetahui apakah rangkaian optocoupler tersebut dapat bekerja dengan baik. Pengujian pada bagian ini untuk mengukur tegangan keluaran yang dihasilkan oleh optocoupler pada saat rangkaian tersebut bekerja dan pada saat tidak bekerja. Diagram blok pengujian rangkaian *optocoupler* ditunjukkan dalam Gambar 5.1.



Gambar 5.1. Diagram Blok Pengujian Rangkaian *Optocoupler*
Sumber : Pengujian

Tabel 5.1. Hasil Pengujian Rangkaian *Optocoupler*

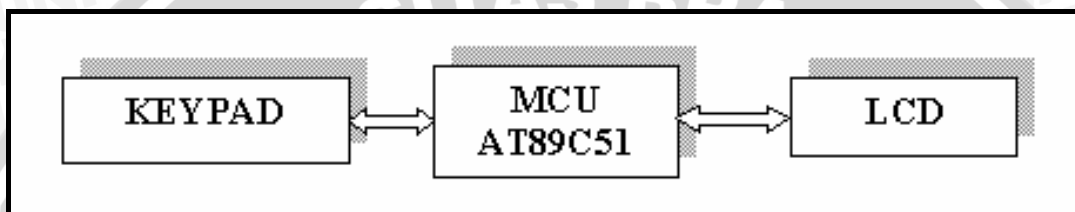
Keadaan	Vout	Led Indikator	Hasil
Tdk Ada Penghalang	3,95 V	Nyala	Bekerja (logika 1)
Ada Penghalang	0,14 V	Mati	Bekerja (logika 0)

Sumber : Pengujian

Dari hasil pengujian yang ditunjukkan dalam Tabel 5.1, sensor *flowmeter* bekerja dengan baik sesuai dengan perencanaan yang menghasilkan level tegangan untuk logika 0 (0,14V) dan logika 1 (3,95V).

5.2 Pengujian Keypad dan LCD

Tujuan pengujian Keypad dan LCD adalah untuk mengetahui apakah rangkaian keypad dan rangkaian LCD dapat bekerja dengan baik. Program dibuat untuk mengetahui data yang dikirim dari penekanan keypad telah sesuai atau tidak dan ditampilkan pada display LCD. Pengujian rangkaian keypad dan rangkaian LCD digabungkan dengan tujuan untuk efisiensi. Jadi pengujian rangkaian keypad sekaligus pengujian rangkaian LCD. Bila ada penekanan tombol pada keypad akan ditampilkan pada display LCD. Data hasil pengujian Keypad dapat dilihat dalam Tabel 5.2. Diagram blok pengujian keypad dan LCD dapat dilihat dalam Gambar 5.2.



Gambar 5.2. Diagram Blok Pengujian Keypad dan LCD
Sumber : Pengujian

Data dikirimkan melalui keypad matrik baris dan kolom. Persambungan antara baris dan kolom akan menghasilkan data keluaran tertentu. Untuk pemrosesan keypad dilakukan dengan menggunakan software. Data keluaran dari keypad dapat dilihat di dalam LCD. Bagian data menghasilkan data dari kombinasi hubungan antara baris dan kolom sedangkan bagian status akan mengidentifikasi adanya hubungan antara baris dan kolom yang valid. Keluaran masing-masing bagian adalah aktif tinggi.

Tabel 5.2. Hasil Pengujian Keypad

Tombol	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	Tampil di LCD
1	1	1	1	0	1	1	1	0	1
2	1	1	1	0	1	1	0	1	2
3	1	1	1	0	1	0	1	1	3
COR	1	1	1	0	0	1	1	1	C
4	1	1	0	1	1	1	1	0	4
5	1	1	0	1	1	1	0	1	5
6	1	1	0	1	1	0	1	1	6
MEN	1	1	0	1	0	1	1	1	D
7	1	0	1	1	1	1	1	0	7
8	1	0	1	1	1	1	0	1	8

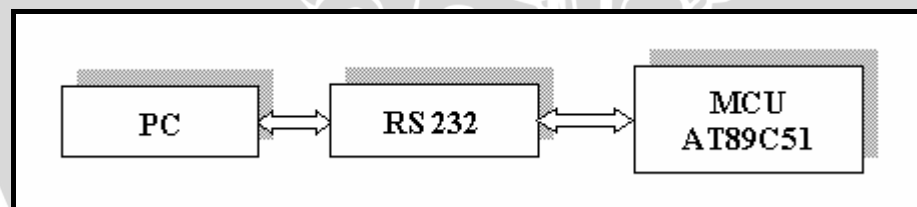
9	1	0	1	1	1	0	1	1	9
↑	1	0	1	1	0	1	1	1	E
CAN	0	1	1	1	1	1	1	0	A
0	0	1	1	1	1	1	0	1	0
ENT	0	1	1	1	1	0	1	1	B
↓	0	1	1	1	0	1	1	1	F

Sumber : Pengujian.

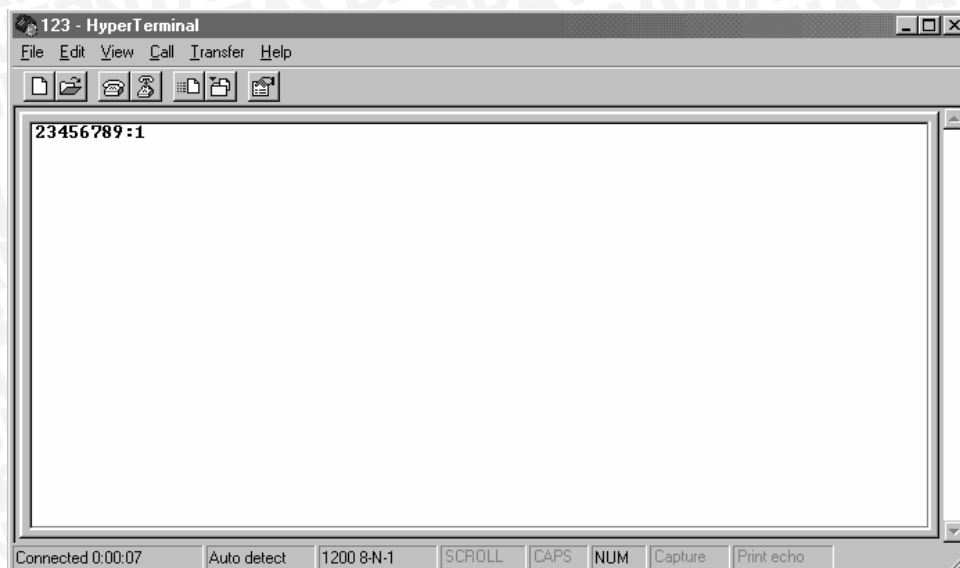
Berdasarkan hasil pengujian *keypad* pada Tabel 5.2 didapatkan bahwa *keypad* telah berfungsi dengan baik.

5.3 Pengujian Rangkaian Serial RS-232

Pengujian ini untuk mengetahui kerja IC serial MAX232 sebagai rangkaian antar muka antara komputer dengan sistem mikrokontroler AT89C51. Rangkaian pengujian serial RS-232 dapat dilihat dalam Gambar 5.3. Pengujian dilakukan dengan mengirimkan data berupa penekanan tombol di *keyboard* PC melalui port serial komputer (COM1) kemudian data tersebut akan dikembalikan lagi oleh mikrokontroler kepada komputer. Pengujian menggunakan program **Hyper-terminal** yang ada di OS Windows 98. Gambar **Hyper-terminal** hasil pengujian dapat dilihat dalam Gambar 5.3.



Gambar 5.3. Diagram Blok Pengujian RS-232
Sumber : Pengujian



Gambar 5.4. Hasil Pengujian RS-232 di Hyper-terminal
 Sumber : Pengujian

Tabel 5.3. Pengujian Serial

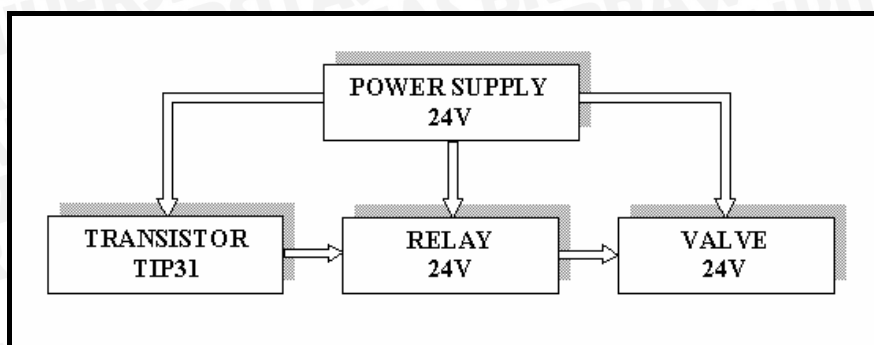
Karakter dikirim	Karakter diterima
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	:
0	1

Sumber : Pengujian

Dalam Gambar 5.4, karakter “23456789:1” tampil. Berdasarkan Tabel 5.3 karakter yang dikirim di-*increment*. Berdasarkan Gambar 5.4 dan Table 5.3, rangkaian serial RS-232 telah bekerja dengan baik.

5.4 Pengujian Rangkaian *Driver*

Tujuan pengujian rangkaian *driver* adalah untuk mengetahui apakah rangkaian driver tersebut dapat bekerja dengan baik. Pengujian pada bagian ini untuk mengukur tegangan keluaran yang dihasilkan oleh rangkaian *driver* pada saat rangkaian tersebut bekerja dan pada saat tidak bekerja. Diagram blok pengujian rangkaian *driver* ditunjukkan dalam Gambar 5.5.



Gambar 5.5. Diagram Blok Pengujian Rangkaian *Driver*
 Sumber : Pengujian

Tabel 5.4. Hasil Pengujian Rangkaian *Driver*

Keadaan	Vout	Hasil
Ada V Input dari MK	24 V	Bekerja (logika 1)
Tidak Ada V Input dari MK	0 V	Bekerja (logika 0)

Sumber : Pengujian

Dari hasil pengujian yang ditunjukkan dalam Tabel 5.4, rangkaian *driver* bekerja dengan baik sesuai dengan perencanaan yang menghasilkan level tegangan untuk logika 0 (0V) dan logika 1 (24V).

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan perancangan dan pengujian alat, maka dapat ditarik kesimpulan sebagai berikut:

1. Pada perancangan ini sistem komunikasi antara perangkat keras dan *PC* menggunakan komunikasi serial dengan menggunakan standar RS 232.
2. Dalam komputer (PC) menggunakan *operating system* windows dan untuk pemrogramannya menggunakan *compiler Basic*.
3. Dari hasil pengujian yang meliputi pengujian perangkat keras dan perangkat lunak diketahui bahwa alat dapat beroperasi sesuai dengan perencanaan.

6.2. Saran

Ada beberapa hal yang perlu diperhatikan dalam pengembangan alat ini dikemudian hari. Meskipun alat ini sudah dapat bekerja dengan baik dan sesuai dengan sistem dan spesifikasi yang direncanakan, namun masih ada beberapa hal yang perlu ditingkatkan diantaranya adalah :

1. Untuk pengembangan lebih lanjut penggunaan *microweb* cukup satu dan hanya dipusat, sehingga pihak SPBU tidak perlu memasang *microweb* tersebut dan cukup mengupload data saja.



DAFTAR PUSTAKA

Atmel. 1998. *8 Bit Microcontroller with 2K Bytes Flash AT 89C51*. USA:
<http://www.atmel.com>

Compulogic. 2005. *Picoserv Miniatur Web Server User Documentation*. United Kingdom. <http://www.compulogic.com>

Nalwan, Paulus Andi. 2003. *Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*. Jakarta: Elex Media Komputindo.

Putra, Agfianto Eko. 2002. *Teknik Antarmuka Komputer: Konsep dan Aplikasi*. Yogyakarta: Graha Ilmu.

<http://www.elektroindonesia.com/elektro/komp30.html>

