

**ALAT PEMBERI PAKAN UNTUK IKAN SECARA OTOMATIS
BERBASISKAN MIKROKONTROLER AT89S8252**

SKRIPSI

Diajukan untuk memenuhi persyaratan

Memperoleh gelar Sarjana Teknik



Disusun Oleh :

RIZKY PERDANA KURNIAWAN

NIM. 9801060358-63

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2005

**ALAT PEMBERI PAKAN UNTUK IKAN SECARA OTOMATIS
BERBASISKAN MIKROKONTROLER AT89S8252**

SKRIPSI

Diajukan untuk memenuhi persyaratan

Memperoleh gelar Sarjana Teknik



Disusun Oleh :

RIZKY PERDANA KURNIAWAN

NIM. 9801060358-63

Menghentui dan menyetujui :
DOSEN PEMBIMBING SKRIPSI

Ir. Bambang Siswoyo

NIP. 131 759 588

Ir. Nanang Sulistyanto

NIP. 132 090 389

ALAT PEMBERI PAKAN UNTUK IKAN SECARA OTOMATIS BERBASISKAN MIKROKONTROLER AT89S8252

Disusun Oleh :

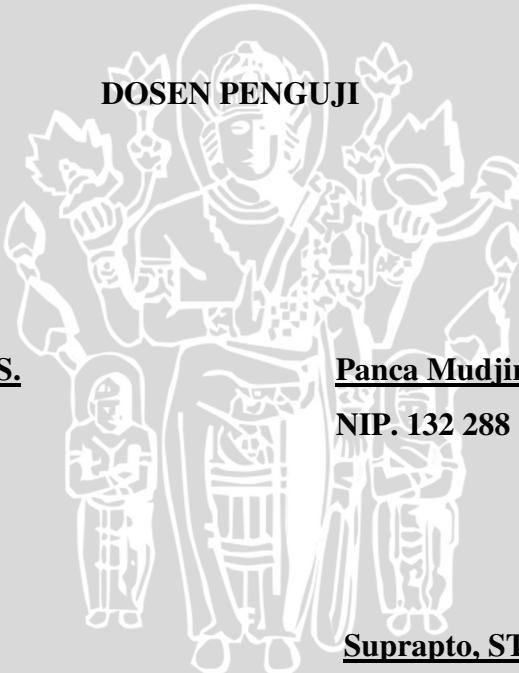
RIZKY PERDANA KURNIAWAN

NIM. 9801060358-63

Skripsi ini telah diuji dan dinyatakan lulus pada

Tanggal 30 Juli 2005

DOSEN PENGUJI



Ir. Ponco Siwindarto, MS.

NIP. 131 837 966

Panca Mudjiraharjo, ST., MT.

NIP. 132 288 163

Ir. Muhammad Aswin

NIP. 131 879 045

Suprapto, ST.,MT.

NIP. 132 149 320

Mengetahui

Ketua Jurusan Teknik Elektro

Ir. Purwanto, MT.

NIP. 131 574 847

PENGANTAR

Puji syukur kepada Allah SWT yang telah memberikan rahmat-Nya sehingga penyusunan skripsi ini dapat terselesaikan. Hanya karena pertolongan-Nya semata penulis mampu melewati segala kendala yang ada selama penyusunan skripsi ini. Skripsi berjudul “*Alat Pemberi Pakan untuk Ikan Secara Otomatis Berbasiskan Mikrokontroler AT89S8252*” ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.

Penulis menyadari selama penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, dorongan dan motivasi dari berbagai pihak. Oleh sebab itu, dengan segala kerendahan hati penulis menyampaikan terima kasih kepada :

- ❖ Ir. Purwanto, MT selaku Ketua Jurusan Teknik Elektro dan Ir. Hery Purnomo selaku Sekretaris Jurusan Teknik Elektro,
- ❖ Ir. Bambang Siswoyo selaku dosen pembimbing I dan Ir. Nanang Sulistiyanto selaku dosen pembimbing II,
- ❖ Mama, Dik Ema, Dik Reza dan Dik Nia untuk doa, kesabaran dan dukungannya selama studi hingga terselesaikannya skripsi ini,
- ❖ Rekan-rekan Arrijal untuk kebersamaannya selama ini,
- ❖ Semua pihak yang telah banyak membantu penulis selama ini.

Akhir kata, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan. Penulis berharap semoga skripsi ini dapat berguna bagi pengembangan ilmu dan teknologi.

Malang, Juli 2005

Penulis

ABSTRAKSI

RIZKY PERDANA KURNIAWAN, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Juli 2005, *Alat Pemberi Pakan Untuk Ikan Secara Otomatis Berbasiskan Mikrokontroler AT89S8252*, Dosen Pembimbing : Ir. Bambang Siswoyo dan Ir. Nanang Sulistiyanto.

Salah satu permasalahan utama dalam memelihara ikan hias ialah pertumbuhan dan kesehatan ikan yang dapat terganggu apabila waktu pemberian pakan tidak teratur dan jumlah pakan yang diberikan tidak tepat. Hal ini memerlukan perhatian khusus dari pemilik ikan tersebut. Untuk meringankan tugas pemilik ikan tersebut, dibuat alat pemberi pakan ikan otomatis yang dapat disesuaikan dengan kebutuhan ikan.

Alat ini terdiri dari empat bagian utama yaitu, komponen pewaktu *real time* (DS12887), pengendali utama (mikrokontroler AT89S8252), sistem mekanik pemberi pakan ikan, serta sistem antarmuka dengan pengguna (*keypad* dan LCD). Ketika waktu pemberian pakan telah tiba, pewaktu akan memberikan sinyal kepada mikrokontroler untuk melakukan pemberian pakan. Mikrokontroler kemudian memberikan perintah kepada sistem mekanik untuk memberikan pakan sesuai dengan takaran yang telah ditentukan. Jadwal pemberian pakan dan banyaknya pakan yang akan diberikan dapat di-*setting* melalui unit antarmuka.

Alat bekerja dengan baik dengan kemampuan untuk menimbang pakan mulai 1 gram sampai dengan 20 gram, dengan kenaikan massa setiap 1 gram. Dalam satu hari alat bisa memberikan pakan sampai dengan lima kali. Alat dapat menampung 1 kilogram pakan yang mencukupi untuk pemberian pakan minimal dalam 10 hari.

DAFTAR ISI

PENGANTAR	i
ABSTRAKSI	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	ix

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Sistematika Pembahasan.....	3

BAB II TINJAUAN PUSTAKA

2.1 Mikrokontroler AT89S8252	5
2.2 <i>RTC</i> Basis Pewaktuan DS12887.....	12
2.3 <i>Liquid Crystal Display (LCD)</i>	16
2.4 <i>Switching Transistor</i>	20
2.5 <i>ADC 0809</i>	21
2.6 <i>Encoder MM74C922</i>	24
2.7 Sensor <i>LDR (Light Dependent Resistor)</i>	25
2.8 Pemberian Pakan Untuk Ikan Hias	26

BAB III METODOLOGI

3.1	Studi Literatur	27
3.2	Perancangan Alat	27
3.3	Pengujian dan Analisis Alat.....	27
3.4	Prosedur yang Digunakan.....	28

BAB IV PERANCANGAN DAN PEMBUATAN ALAT

4.1	Gambaran Umum.....	29
4.2	Blok Diagram Rangkaian	29
4.3	Perangkat Keras	30
4.3.1	Mikrokontroler AT89S8252	30
4.3.2	Reset Mikrokontroler	32
4.3.3	Papan Tombol (Keypad)	33
4.3.4	RTC	34
4.3.5	LCD	34
4.3.6	Sensor LDR dan Pengkondisi Sinyal.....	36
4.3.7	ADC08909	40
4.3.8	Driver Solenoid.....	43
4.3.9	Rangkaian Mekanik Pemberi Pakan	45
4.4	Perancangan Perangkat Lunak.....	45
4.4.1	Format Penyimpanan Data dalam Memori EEPROM.....	45
4.4.2	Perancangan Diagram Alir Program.....	45

BAB V PENGUJIAN DAN ANALISIS

5.1 Pengujian Rangkaian Unit Masukan Keypad	56
5.2 Pengujian Tampilan LCD	57
5.3 Pengujian RTC.....	58
5.4 Pengujian ADC0809	58
5.5 Pengujian Sensor	60
5.6 Pengujian Rangkaian Pengkondisi Sinyal	61
5.7 Pengujian Error Timbangan.....	62
5.8 Pengujian Keseluruhan Sistem	62

BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan	64
6.2 Saran	64

DAFTAR PUSTAKA..........65**LAMPIRAN**

DAFTAR GAMBAR

Gambar 2.1	Konfigurasi pin AT89S8252.....	5
Gambar 2.2	Blok diagram core AT89S8252	6
Gambar 2.3	Koneksi master/slave SPI	11
Gambar 2.4	Konfigurasi penyematan IC DS12887.....	12
Gambar 2.5	Timing diagram proses baca pada sistem bus intel.....	13
Gambar 2.6	Timing diagram proses tulis pada sistem bus intel	13
Gambar 2.7	Timing diagram proses baca dan tulis pada sistem bus intel	14
Gambar 2.8	Peta alamat RAM <i>Real Time Clock</i> DS12887	16
Gambar 2.9	LCD dan konfigurasi pinnya.....	19
Gambar 2.10	Susunan pena ADC0809	22
Gambar 2.11	Diagram pewaktuan ADC0809.....	22
Gambar 2.12	Konfigurasi IC MM74C922.....	23
Gambar 2.13	Simbol LDR	24
Gambar 4.1	Diagram blok perancangan	28
Gambar 4.2	Rangkaian clock mikrokontroller	30
Gambar 4.3	Rangkaian mikrokontroler AT89S8252.....	30
Gambar 4.4	Rangkaian power on reset dan setaranya.....	31
Gambar 4.5	Susunan keypad	32
Gambar 4.6	Rangkaian RTC DS12887	33
Gambar 4.7	Rangkaian LCD	34
Gambar 4.8	LDR sebagai sensor pada timbangan.....	35
Gambar 4.9	LDR dan rangkaian pengkondisi sinyal.....	37
Gambar 4.10	Rangkaian ADC 0809	38
Gambar 4.11	Rangkaian clock ADC 0809	39

Gambar 4.12	Rangkaian driver solenoid	41
Gambar 4.13	Konstruksi mekanik pemberi pakan ikan.....	42
Gambar 4.14	Diagram alir program utama.....	44
Gambar 4.15	Diagram alir subrutin pengecekan jadwal makan.....	45
Gambar 4.16	Diagram alir subrutin proses pemberian pakan	46
Gambar 4.17	Diagram alir subrutin pemilihan menu	47
Gambar 4.18	Diagram alir subrutin pengesetan jadwal.....	48
Gambar 4.19	Diagram alir subrutin pemilihan jadwal	48
Gambar 4.20	Diagram alir subrutin set jam pemberian pakan	49
Gambar 4.21	Diagram alir subrutin set menit pemberian pakan	49
Gambar 4.22	Diagram alir subrutin set banyaknya pakan yang diberikan.....	49
Gambar 4.23	Diagram alir subrutin set pengaktifan jadwal	50
Gambar 4.24	Diagram alir subrutin pengesetan waktu	51
Gambar 4.25	Diagram alir subrutin set jam.....	51
Gambar 4.26	Diagram alir subrutin set menit	52
Gambar 4.27	Diagram alir subrutin set detik.....	52
Gambar 4.28	Diagram alir subrutin kalibrasi	53
Gambar 4.29	Diagram alir subrutin baca timbangan.....	53
Gambar 5.1	Rangkaian pengujian unit masukan keypad.....	54
Gambar 5.2	Rangkaian pengujian LCD.....	55
Gambar 5.3	Hasil pengujian LCD	55
Gambar 5.4	Rangkaian pengujian RTC.....	56
Gambar 5.5	Rangkaian pengujian ADC	57
Gambar 5.6	Grafik Pengujian ADC.....	58
Gambar 5.7	Grafik hasil pengujian LDR.....	62

Gambar 5.8	Rangkaian pengujian rangkaian pengkondisi sinyal.....	62
Gambar 5.9	Pengujian sistem keseluruhan	63



DAFTAR TABEL

Tabel 2.1	Fungsi alternatif pin port 1	6
Tabel 2.2	Fungsi alternatif pin port 3	7
Tabel 2.3	Register untuk watchdog dan EEPROM	8
Tabel 2.4	<i>Periodic interrupt rate</i> dan Frekuensi SQW output.....	14
Tabel 2.5	Register seleksi	17
Tabel 2.6	Instruksi pemrograman LCD	18
Tabel 2.7	Fungsi masing-masing penyematan LCD	18
Tabel 2.8	Masukan jalur alamat untuk memilih kanal analog.....	20
Tabel 2.9	Data output <i>encoder</i> MM74C922.....	23
Tabel 4.1	Susunan penyematan LCD M1632	34
Tabel 4.2	Data keluaran LDR	35
Tabel 5.1	Hasil pengujian rangkaian keypad.....	53
Tabel 5.2	Hasil pengujian RTC	55
Tabel 5.3	Data pengujian ADC 0809	56
Tabel 5.4	Data pengujian rangkaian pengkondisi sinyal	58
Tabel 5.5	Hasil pengujian sistem secara keseluruhan	59



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Terdapat tiga permasalahan utama dalam memelihara ikan hias yang perlu diperhatikan oleh para pemilik ikan hias. Pertumbuhan ikan hias jenis tertentu dapat terganggu apabila waktu pemberian pakan dilakukan dengan tidak teratur dan jumlah takaran pakan yang diberikan tidak sesuai dengan kebutuhan ikan tersebut. Pemberian pakan yang berlebihan menyebabkan air pada akuarium cepat kotor sehingga bisa mengganggu kesehatan ikan hias tersebut. Pemeliharaan ikan hias yang membutuhkan perhatian khusus serta cukup banyak menyita waktu tersebut dapat memberikan permasalahan tersendiri bagi pemilik ikan hias yang memiliki kesibukan kerja yang padat.

Dari permasalahan-permasalahan di atas maka diperlukan alternatif solusi dalam hal pemberian pakan ikan hias. Di antaranya adalah dengan membayar pekerja untuk mengantikan tugas pemilik dalam hal pemberian pakan ikan hias. Solusi lain adalah dengan menggunakan alat otomatis yang bisa memberikan pakan secara teratur dan tepat waktu, serta dapat memberikan jumlah takaran secara tepat sesuai kebutuhan. Sebagai pengendali utama alat otomatis tersebut dapat menggunakan komputer. Alternatif lain untuk pengendali utama alat otomatis tersebut adalah dengan menggunakan mikrokontroler.

Dari beberapa alternatif solusi tersebut maka perlu dipilih satu solusi yang tepat untuk mengatasi permasalahan pemberian pakan pada ikan hias. Solusi yang dipilih adalah dengan menggunakan alat otomatis yang menggunakan mikrokontroler sebagai pengendali utama. Alasan memilih alat otomatis karena biaya operasional relatif lebih murah dibandingkan membayar pekerja untuk mengantikan tugas pemilik dalam hal pemberian pakan. Sebagai pengendali utama digunakan mikrokontroler karena biayanya jauh lebih murah dibandingkan menggunakan komputer, biaya tersebut terdiri dari harga sistem mikrokontroler yang jauh lebih murah dibandingkan komputer, serta biaya operasional sistem mikrokontroler yang jauh lebih murah karena konsumsi dayanya jauh lebih kecil dibandingkan komputer.

Ada beberapa permasalahan teknis yang perlu diperhatikan pada perancangan alat otomatis pemberi pakan ikan hias agar alat bisa bekerja dengan baik. Ketelitian dalam hal jumlah takaran harus cukup baik sehingga diperlukan sistem penimbang pakan yang dapat menimbang dengan tingkat ketelitian yang baik. Alat harus bisa memberikan pakan tepat waktu dan teratur sehingga diperlukan sistem pewaktu yang baik pula. Daya tampung alat minimal harus cukup untuk sekitar dua minggu agar pemilik tidak terlalu sering mengisi pakan pada penampung. Sistem antarmuka alat harus dirancang agar mudah dalam pengoperasiannya.

1.2 RUMUSAN MASALAH

Dari permasalahan yang telah disebutkan di atas, maka dalam perancangan ini rumusan masalah ditekankan pada:

1. Bagaimana membuat alat pemberi pakan yang bisa disetting dengan mudah oleh pengguna.
2. Bagaimana membuat peralatan yang dapat mengatur waktu pemberian pakan dan jumlah takaran secara tepat pada Alat Pemberi Pakan untuk Ikan Secara Otomatis.
3. Bagaimana membuat perangkat lunak sebagai pendukung pembuatan Alat Pemberi Pakan Ikan Secara Otomatis Berbasiskan Mikrokontroler AT89S8252.
4. Bagaimana membuat sistem mekanis yang dapat menampung jumlah pakan sesuai kebutuhan, menimbang jumlah pakan sekaligus memberikan pakan untuk ikan pada Alat Pemberi Pakan Ikan Secara Otomatis Berbasiskan Mikrokontroler AT89S8252

1.3 BATASAN MASALAH

Mengingat luasnya permasalahan yang ada dalam perancangan ini, pembahasan ditekankan pada:

1. Jenis mikrokontroller yang digunakan sebagai pengendali sistem adalah AT89S8252
2. Alat ini hanya dirancang khusus untuk ikan hias yang biasa dipelihara di akuarium.

3. Alat ini dirancang untuk mengantikan tugas pemelihara ikan hias dalam hal pemberian pakan secara rutin tanpa dipengaruhi oleh kondisi ikan ataupun akuarium tempat ikan berada setelah alat dioperasikan.
4. Jenis pakan yang digunakan hanya berupa pakan buatan berbentuk butiran kecil, tidak berupa pakan alami atau pakan hidup.
5. Tidak melakukan pembahasan pada perancangan mekanik.
6. Tidak membahas catu daya .
7. Diasumsikan catu daya *disupply* secara kontinyu.

1.4 TUJUAN

Tujuan dari skripsi ini adalah pembuatan Alat Pemberi Pakan Ikan Secara Otomatis Berbasiskan Mikrokontroler AT89S8252, yang dapat disetting waktu pemberian pakan, frekuensi pemberian pakan per harinya, serta massa pakan yang diberikan setiap proses pemberian pakan.

1.5 SISTEMATIKA PEMBAHASAN

Sistematika pembahasan yang akan digunakan dalam penulisan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Berisi latar belakang judul, rumusan masalah, batasan masalah, tujuan dan sistematika pembahasan.

BAB II : DASAR TEORI

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat. Pemberian Pakan Ikan Otomatis Berbasiskan AT89S8252.

BAB III: METODOLOGI PENELITIAN

Membahas metode penelitian yang digunakan, spesifikasi alat, perencanaan alat dan pengujian.

BAB IV: PERENCANAAN DAN REALISASI ALAT

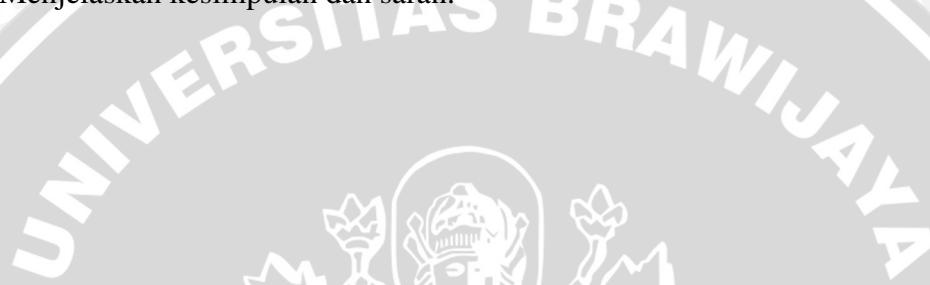
Menjelaskan perencanaan dan pembuatan Alat Pemberian Pakan Ikan Otomatis Berbasiskan AT89S8252

BAB V : PENGUJIAN DAN ANALISIS

Menjelaskan hasil pengujian dan analisa alat yang sudah direalisasikan.

BAB VI: PENUTUP

Menjelaskan kesimpulan dan saran.

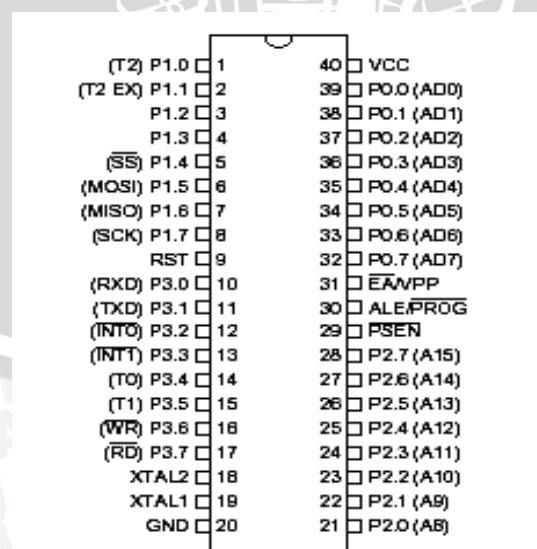


BAB II

TINJAUAN PUSTAKA

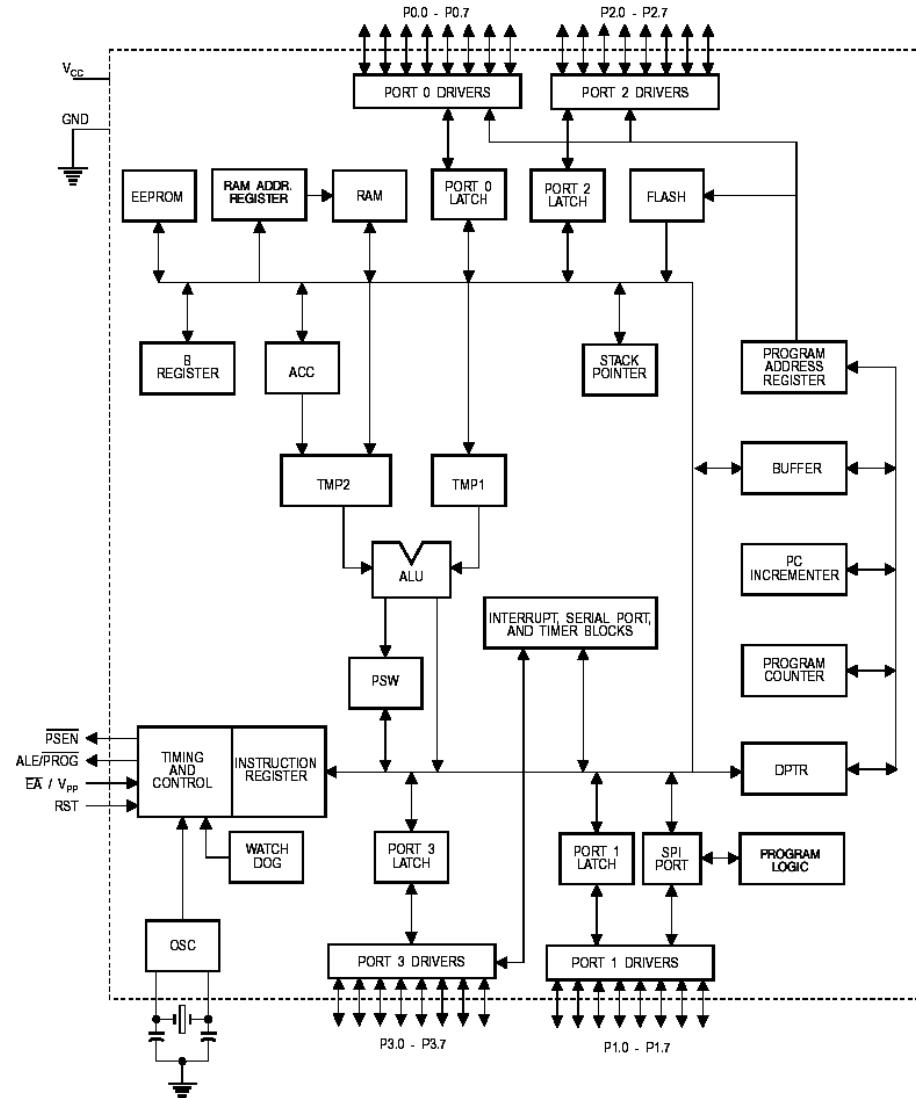
2.1 Mikrokontroler AT89S8252

Mikrokontroler yang dipilih adalah mikrokontroler AT89S8252 yang diproduksi oleh Atmel, yang dilengkapi dengan 8 kB *flash memory* yang bisa diprogram dan dihapus. Selain *flash memory*, AT89S8252 juga menyediakan 2 kB EEPROM, 256 byte RAM internal, 32 jalur I/O, tiga buah *timer/ counter* 16- bit, sebuah *watchdog timer* yang dapat diprogram, dua buah data pointer, enam buah vektor *interrupt*, *on-chip oscillator*, rangkaian clock dan sebuah port serial (*full- duplex*), yang sangat bermanfaat dalam melakukan komunikasi serial. AT89S8252 juga mendukung mode untuk menghemat pemakaian daya. Mode tersebut antara lain mode *idle* dan mode *power-down*. Pada mode *idle*, CPU berhenti bekerja sedangkan RAM, *timer/counter*, port serial, dan sistem *interrupt* tetap aktif. Sedangkan pada mode *power-down*, semua aktivitas mikrokontroler berhenti kecuali RAM sampai dideteksi adanya *interrupt* atau sinyal *reset* dari luar. Gambaran konfigurasi pin mikrokontroler dan diagram bloknya dapat dilihat pada Gambar 2.1.



Gambar 2.1. Konfigurasi Pin AT89S8252

Sumber: Atmel , 1997: 105



Gambar 2.2. Blok Diagram Core AT89S8252

Sumber: Atmel, 1997: 107

Blok diagram diatas mirip dengan diagram untuk core AT89S8252. Yang membedakan hanyalah adanya *Serial Peripheral Interface*, EEPROM, jumlah timer lebih banyak, dan *watchdog timer*.

2.1.1 Deskripsi Pin

VCC

Pin ini digunakan untuk mencatut mikrokontroler sebesar +5 V.

GND

Pin ini digunakan sebagai titik referensi untuk setiap pin pada mikrokontroler.

PORT 0

Port 0 merupakan sebuah port I/O *bi-directional* 8-bit. Sebagai port output setiap pin dapat menerima arus dari sebuah 8 buah input TTL. Pada saat port 0 diberi satu, tiap pin dapat digunakan sebagai input dengan *high impedance*. Port 0 juga digunakan sebagai bus *byte* terendah dari alamat atau data pada saat mikrokontroler mengakses memori eksternal. Pada mode ini, Port 0 mempunyai *pull-up* sudah tersedia di dalam mikrokontroler. Port 0 juga digunakan untuk menerima kode selama proses pemrograman dan mengeluarkan kode selama proses verifikasi. Pada saat verifikasi program, digunakan *pull-up* eksternal.

PORT 1

Port 1 merupakan port I/O *bi-direction* dengan *pull-up* internal. *Buffer* output Port 1 mampu menerima atau memberikan arus untuk 4 input TTL. Pada saat port 1 diberikan logika ‘1’, port 1 akan di *pull-high* dengan *pull-up* internal dan dapat digunakan sebagai input. Sebagai input, tiap pin pada port 1 (yang di *pull-low*) akan mengeluarkan arus I_{IL} karena adanya *pull-up* internal. Beberapa pin port 1 mempunyai fungsi tambahan. Pin P 1.0 dan P1.1 dapat digunakan sebagai pencacah dan *trigger* untuk *timer/ counter* 2. Sedangkan P.1.4, P.1.5, P.1.6 dan P.1.7 dapat digunakan untuk memilih port SPI, I/O data dan I/O *shift clock*. Port 1 juga digunakan untuk menerima alamat terendah selama proses pemrograman dan verifikasi.

Tabel 2.1. Fungsi Alternatif Pin Port 1

Pin Port	Fungsi Alternatif
P1.0	T2 (input pencacah eksternal untuk Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger dan kontrol arah)
P1.4	SS (Input pemilih port slave)
P1.5	MOSI (Master data output, slave data input) – pin untuk jalur SPI
P1.6	MISO (Master data input, slave data output) – pin untuk jalur SPI
P1.7	SCK (Master clock output, slave clock input) – pin untuk jalur SPI

Sumber: Atmel, 1997:108

PORT 2

Port 2 merupakan sebuah port I/FADE OUT 8-bit yang sifatnya *bi-directional* dengan *pull-up* internal. *Buffer* dari keluaran port 2 mampu menerima atau memberikan arus pada empat buah input TTL. Jika port 2 diberi logika ‘1’, maka setiap pin akan di *pull-high* dengan *pull-up* internal dan bisa digunakan sebagai input. Sebagai input, jika

port 2 di *pull-low* maka setiap pin mampu memberikan arus I_{IL} karena adanya *pull-up* internal.

POR T 3

Port 3 merupakan sebuah I/O *bi-directional* 8-bit yang mempunyai *pull-up* internal atau *buffer* keluaran port 3 mampu menerima/memberi arus kepada empat buah input TTL. Pada saat port 3 diberi logika ‘1’, setiap pin tersebut akan di *pull-up* secara internal dan dapat digunakan sebagai input. Sebagai input, jika port 3 di *pull-low*, setiap pin pada port tersebut akan memberikan arus I_{IL} karena adanya *pull-up*. Port 3 juga menyediakan fungsi tambahan seperti terlihat pada Tabel 2.2 berikut.

Tabel 2.2. Fungsi Alternatif Pin Port 3.

Pin Port	Fungsi Alternatif
P3.0	RXD (port input serial)
P3.1	TXD (port output serial)
P3.2	INT0 (interrupt eksternal 0)
P3.3	INT1 (interrupt eksternal 1)
P3.4	T0 (input eksternal timer 0)
P3.5	T1 (input eksternal timer 1)
P3.6	WR (strobe penulisan data eksternal)
P3.7	RD (strobe pembacaan data eksternal)

Sumber: Atmel, 1997:109

RST

Merupakan pin untuk memberikan sinyal *reset* pada mikrokontroler. *Reset* yang digunakan adalah sebuah pulsa dengan logika ‘1’ atau HV. Pulsa yang diberikan minimal sebesar 2 cycle mesin pada saat osilator bekerja atau aktif.

EA/VPP

EA merupakan sinyal *Enable* yang digunakan untuk mengakses memori eksternal. EA harus di-*starp* ke VCC supaya dapat mengeksekusi program internal. Pin ini juga menerima VPP sebesar 12 V selama pemrograman pararel.

XTAL1 dan XTAL2

XTAL1 merupakan input untuk amplifier osilator *inverting* dan input untuk rangkaian clock internal. XTAL2 merupakan keluaran dari amplifier osilator *inverting*.

2.1.2 Spesial Function Register

AT89S8252 mempunyai peta register yang memiliki fungsi khusus. Register ini disebut dengan istilah *Special Function Register* (SFR). Tidak semua alamat dalam memori ditempati oleh SFR. Oleh karena itu penulisan dan pembacaan pada alamat yang tidak ditempati ini merupakan sesuatu yang sia-sia.

Register kontrol untuk EEPROM

EEMEN dan EEMWE pada register WMCON digunakan untuk memilih 2 KB EEPROM dan untuk meng-*enable* penulisan ke EEPROM.

Tabel 2.3. Register Untuk Watchdog dan EEPROM.

PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
7	6	5	4	3	2	1	0

Sumber: Atmel, 1997:111

Register Dual data Pointer

Untuk memfasilitasi pengaksesan dua memori, EEPROM dan memori data eksternal, AT89S8252 dilengkapi dengan dua buah register Data Pointer 16-bit. DP0 pada SFR dengan lokasi 82H-83H dan DP1 pada 84H-85H. Bit DPS = 0 dalam SFR WMCON akan memilih DP0 dan DPS = 1 akan memilih DP1. Sebelum mengakses *Register Data Pointer*, bit DPS harus diinisialisasi ke nilai yang sesuai.

2.1.3 EEPROM dan RAM

AT89S8252 mempunyai 2 KB EEPROM internal untuk menyimpan data dan 256 byte RAM internal. 128 byte teratas RAM menduduki ruang memori yang sama secara pararel dengan SFR. Artinya 128 byte tersebut mempunyai alamat yang sama dengan SFR. Namun secara fisik, 128 byte teratas RAM terpisah dari SFR.

Pada saat sebuah instruksi mengakses sebuah lokasi diatas 7FH, mode pengalamatan yang digunakan dalam instruksi menentukan apakah CPU harus mengakses SFR atau 128 byte teratas RAM. Instruksi yang menggunakan pengalamatan langsung akan mengakses SFR, sedangkan pengalamatan tak langsung akan mengakses 128 byte teratas RAM. Daerah RAM ini juga digunakan sebagai ruang untuk *stack pointer*.

Penggunaan EEPROM harus diawali dengan mengatur bit EEMEN dalam register WMCON. Alamat EEPROM dimulai dari 000H sampai 7FFH. Jika EEMEN di-

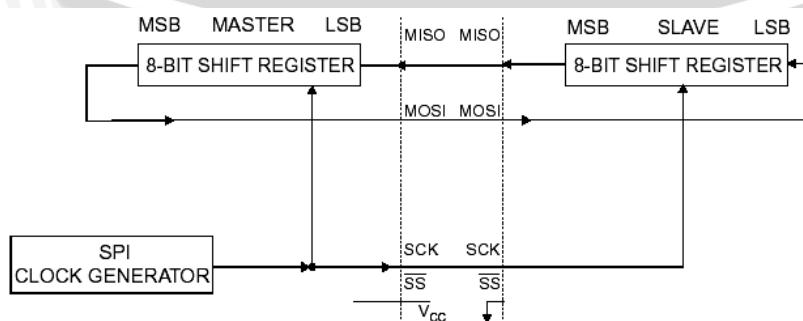
set ‘0’ dan EEMWE di-set ‘1’ maka mikrokontroler akan mengakses memori data eksternal. EEMWE harus di set ke ‘0’ jika tidak ada memori EEPROM yang digunakan. Umumnya penulisan ke EEPROM membutuhkan waktu 2,5 ms untuk setiap karakter . Status penulisan ini dapat dipantau melalui bit RDY/BSY (*read only*) yang terdapat dalam SFR WMCON. RDY/BSY = 0 berarti penulisan belum selesai, RDY/BSY = 1 berarti penulisan telah selesai dan siklus penulisan selanjutnya dapat diinisialisasi.

2.1.4 UART (Universal Asynchronous Transmitter/Receiver)

UART dikenal pula sebagai port serial. Port serial pada AT89S8252 dapat bekerja secara *full duplex* maupun *half duplex*. Pada mode *full duplex*, port serial dapat mengirim dan menerima data dalam waktu yang bersamaan. Port serial dapat mulai menerima data kedua sebelum port selesai menerima data pertama atau diistilahkan dengan sebutan *receive-buffered*. Meskipun demikian, jika data pertama belum juga dibaca ketika data kedua selesai diterima, berarti data pertama akan hilang. Pada mode *half-duplex*, pengiriman dan penerimaan data dilakukan secara bergantian. Register yang digunakan untuk menerima dan mengirim data port serial adalah SBUF. Penulisan data ke SBUF berarti mikrokontroler akan melakukan pengiriman dan penerimaan data melalui port TX/RX.

2.1.5 SPI (Serial Peripheral Interface)

SPI merupakan *interface* yang mampu melakukan komunikasi sinkron *full-duplex* dengan kecepatan tinggi antar AT89S8252 atau antara AT89S8252 dan perangkat lain yang membutuhkan kecepatan tinggi. Port ini dapat digunakan untuk sambungan *master/slave*. Sambungan *master/slave* antara CPU dengan SPI ditunjukkan oleh Gambar 2.3 :



Gambar 2.3. Koneksi Master/Slave SPI

Sumber: Atmel, 1997:119

2.1.6 Karakteristik Osilator.

XTAL1 dan XTAL2 masing-masing adalah sebuah input dan output dari sebuah amplifier inverting yang dapat digunakan sebagai osilator. Resonator yang digunakan dapat berupa kristal maupun keramik. Untuk memberikan clock pada AT89S8252 dari luar, XTAL1 harus terhubung ke rangkaian clock sedangkan XTAL2 dibiarkan dalam keadaan tidak tersambung. *Duty cycle* dari sumber clock luar diberikan oleh sebuah *flip-flop* pembagi-2.

2.1.7 Memprogram Flash dan EEPROM

AT89S8252 menyediakan memori *flash* yang dapat ditulis maupun dibaca sebesar 8 kB dan 2 kB EEPROM. AT89S8252 mendukung pemrograman secara pararel dan serial. Pemrograman secara pararel dilakukan pada tegangan tinggi sebesar 12 V dan secara serial dilakukan pada tegangan rendah sebesar 5 V. Pemrograman secara serial memberikan kemudahan kepada user karena tidak memerlukan rangkaian khusus, sedangkan pemrograman secara pararel kompatibel dengan *pararel-programmer* konvensional.

Pada mode serial, memori untuk kode dan data dipetakan oleh alamat yang berbeda. Sedangkan pada mode pararel, keduanya menempati memori yang bersebelahan yaitu 0000H – 1FFFH untuk kode dan 2000H – 27FFH untuk data. Pada mode serial, untuk memprogram ulang memori tidak perlu menghapus *chip* terlebih dahulu kecuali jika bit-bit lock digunakan pada pemrograman sebelumnya. Sedangkan pada mode pararel, untuk memprogram ulang memori yang sudah berisi kode, *chip* harus dihapus terlebih dahulu (memori *flash* dan EEPROM).

2.2 RTC Basis Pewaktuan (Real Time Clock) IC DS12887

DS12887 adalah serpih yang dipakai sebagai basis pewaktuan buatan Dallas Semiconductor. Terdiri atas 24 penyemat dengan konfigurasi penyemat seperti ditunjukkan dalam Gambar 2.4.

MOT	1	24	V _{CC}
NC	2	23	SQW
NC	3	22	NC
AD0	4	21	NC
AD1	5	20	NC
AD2	6	19	IRQ
AD3	7	18	RESET
AD4	8	17	DS
AD5	9	16	NC
AD6	10	15	R/W
AD7	11	14	AS
GND	12	13	CS

Gambar 2.4. Konfigurasi Penyemant IC DS12887

Sumber : Dallas Semiconductor, 2002 : 1

Keistimewaan IC DS12887 antara lain adalah:

- Osilator internal dan *time base* internal.
- Menghitung detik, menit, jam dalam sehari-hari.
- Menghitung hari dalam setiap minggu, tanggal, bulan dan tahun.
- Seratus tahun kelender.
- Mempunyai catu daya *back-up*.
- Dapat tetap beroperasi selama lebih dari 10 tahun tanpa kehadiran catu daya eksternal (karena adanya battry cadangan di dalam RTC).
- RAM internal sebesar 64 byte, terdiri atas 14 byte untuk *clock* dan register kontrol, sedangkan 50 byte lainnya dapat digunakan oleh pemakai.

Secara keseluruhan, fungsi penyemant-penyemant RTC DS12887 adalah sebagai berikut :

GND dan Vcc:

Merupakan penyemant catu daya. Vcc dihubungkan pada catu daya +5 volt dan GND pada *ground*.

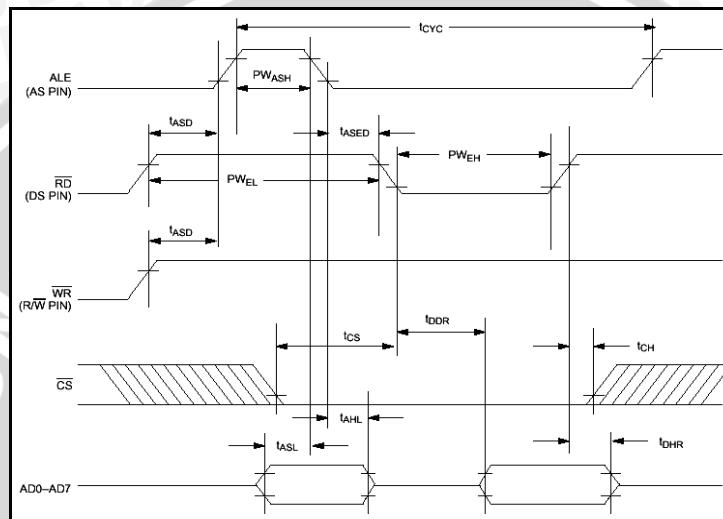
MOT — Motel :

Digunakan sebagai saklar pemilih mode diagram pewaktuan. Apabila dihubungkan pada Vcc berarti sistem diagram pewaktuan Motorola yang dipakai, jika dihubungkan dengan GND berarti sistem diagram pewaktuan Intel yang dipakai.

Sistem diagram pewaktu Intel dan Motorola ditunjukkan pada Gambar 4, 5 dan 6.

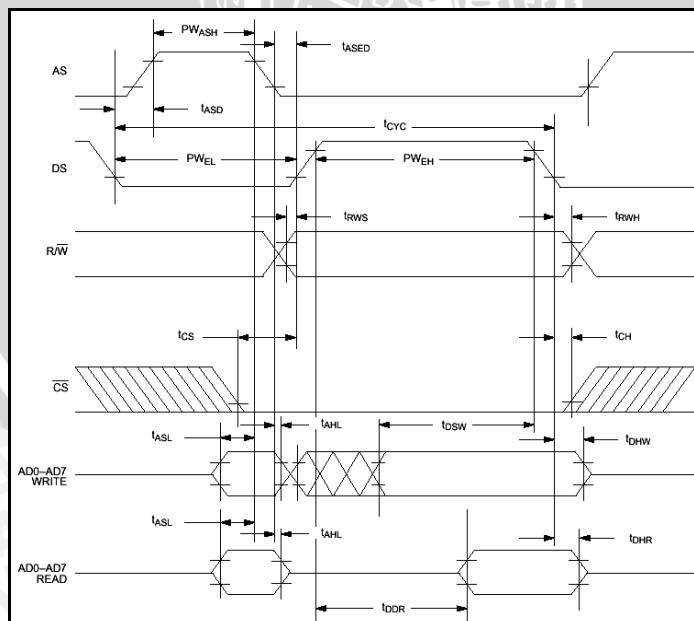
AS — Address Strobe Input :

Merupakan masukan bagi sinyal yang digunakan untuk memisahkan bus data dan bus alamat (ALE). Tepi turun AS/ALE akan menyebabkan alamat ditahan dalam DS1287. Tepi naik berikutnya yang terjadi pada AS akan meniadakan alamat tersebut tanpa memperhatikan apakah penyemat \overline{CS} aktif atau tidak.



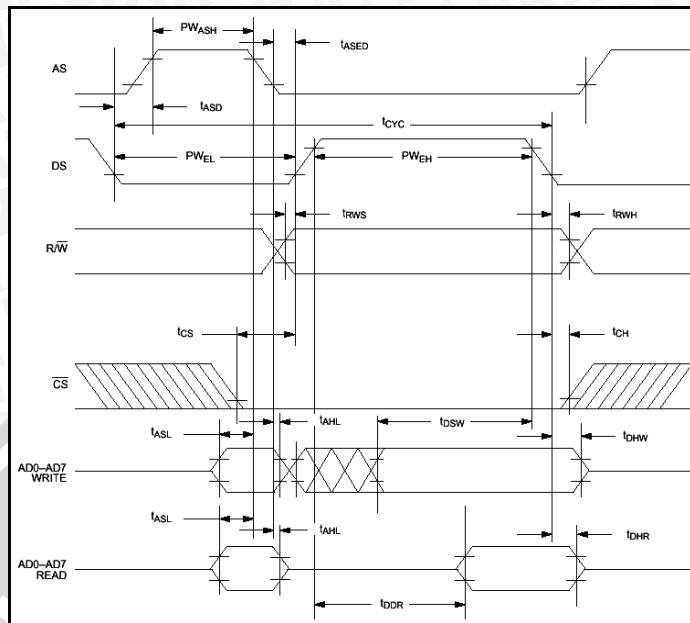
Gambar 2.5. Timing Diagram Proses Baca pada Distem Bus Intel

Sumber : Dallas Semiconductor, 2002 : 17



Gambar 2.6. Timing Diagram Proses Tulis pada Sistem Bus Intel

Sumber : Dallas Semiconductor, 2002 : 17



Gambar 2.7. Timing Diagram Proses Baca dan Tulis pada Sistem Bus Motorola

Sumber : Dallas Semiconductor, 2002 : 16

AD0–AD7 — *Multiplexed Bidirectional Address/Data Bus :*

Merupakan bus alamat/data dua arah yang ter-*multipleks*. Pengiriman data maupun alamat dari maupun ke RTC dilakukan melalui bus ini.

DS — *Data Strobe or Read Input :*

Penyemata DS dapat diartikan sama seperti sinyal *Output Enable* (\overline{OE}) pada komponen memori. Dihubungkan dengan sinyal \overline{RD} yang berasal dari mikrokontroler untuk melakukan proses membaca data pada RAM internal RTC.

R/ \overline{W} — *Read/Write Input :*

Penyemata R/ \overline{W} dapat diartikan sama seperti sinyal *Write Enable* (\overline{WE}) yang biasa dipakai pada komponen memori jenis RAM. Digunakan untuk proses penulisan data pada RAM internal RTC.

SQW — *Square Wave Output:*

Merupakan penyemata yang dapat dipilih untuk mengeluarkan satu dari 13 frekuensi keluaran yang tersedia. Besar frekuensi keluaran SQW dapat diubah dengan diprogram pada Register A. Untuk mengaktifkan atau me-nonaktifkan sinyal SQW dipilih lewat bit SQWE pada register B. Tabel 2.8. berikut ini menunjukkan cara mengeset register A agar dihasilkan keluaran frequensi tertentu pada pin SQW.

Tabel 2.4. Periodic Interupt Rate dan Frekuensi SQW Output

SELECT BIT REGISTER A				t_{PI} PERIODIC INTERUPT RATE	SQW OUTPUT FREQUENCY
RS3	RS2	RS1	RS0		
0	0	0	0	None	None
0	0	0	1	3.90625 ms	256 Hz
0	0	1	0	7.8125 ms	128 Hz
0	1	1	1	122.070 us	8.129 Hz
1	1	0	0	244.141 us	4.096 Hz
1	1	0	1	488.281 us	2.048 Hz
1	0	1	0	976.5625 us	1.024 Hz
1	0	1	1	1.953125 ms	512 Hz
0	0	0	0	3.90625 ms	256 Hz
0	1	0	1	7.8125 ms	128 Hz
0	1	1	0	15.625 ms	64 Hz
0	1	1	1	31.25 ms	32 Hz
1	0	0	0	62.5 ms	16 Hz
1	0	0	1	125 ms	8 Hz
1	0	1	0	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz

Sumber : Dallas Semiconductor, 2002 : 4

CS — Chip Select Input:

Merupakan masukan untuk mengaktifkan RTC. Sinyal \overline{CS} didapat dari dekoder alamat dengan alamat tertentu.

IRQ — Interrupt Request Output:

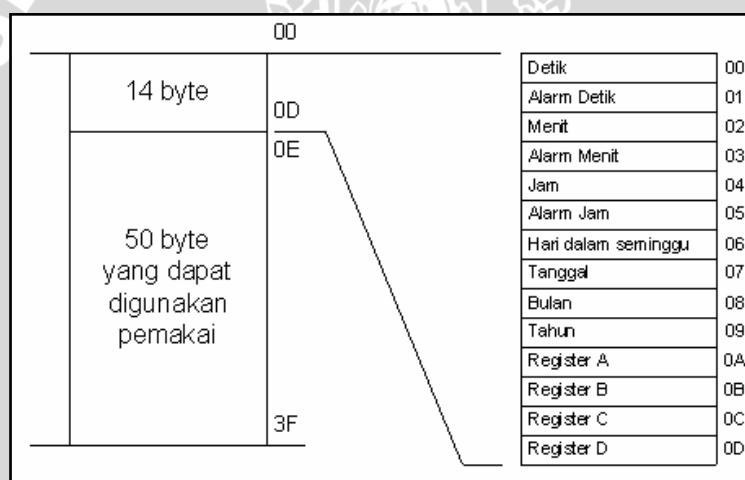
Sinyal \overline{IRQ} merupakan sinyal aktif rendah, yang dapat digunakan untuk menginterupsi mikrokontroler. Keluaran \overline{IRQ} tetap rendah selama status bit yang menyebabkan interupsi ada. Untuk me-reset \overline{IRQ} , mikrokontroler memberikan program pada register C RTC. Saat tidak terdapat interupsi, penyematan ini akan dalam kondisi impedansi tinggi (*high-impedance*).

RESET — Reset Input:

Sinyal \overline{RESET} diberikan dengan memberikan logika rendah selama waktu yang dispesifikasikan, dan tidak berpengaruh pada unjuk kerja *clock*, kalender, dan fungsi RAM. Namun sinyal \overline{RESET} dapat mengakibatkan beberapa *flag* di-reset menjadi nol.

Untuk menginisialisasi RTC, yaitu men-set waktu yang ada di RAM RTC saat kali pertama RTC diaktifkan, digunakan sinyal \overline{WR} (*write*) yang dihubungkan pada penyematan R/\overline{W} , data yang diinginkan pada alamat yang bersesuaian. Sedangkan untuk membaca data dari RAM internal RTC digunakan sinyal \overline{RD} (*read*) yang dihubungkan pada penyematan DS (*data strobe*). Operasi pembacaan dan penulisan pada RAM internal RTC sama seperti operasi baca dan tulis pada komponen memori jenis RAM.

RTC mempunyai RAM internal sebesar 64 byte, yang berisi data-data mengenai waktu yang sedang berjalan, seperti: detik, menit, jam, hari, tanggal, bulan, tahun, serta beberapa register. Secara otomatis RTC akan mengganti data dalam RAM internal sesuai dengan perhitungannya. Jika diinginkan mengambil data waktu, maka dibaca pada RAM internal sesuai dengan alamat yang dimaksud. Peta alamat RAM internal RTC DS12887 ditunjukkan dalam Gambar 2.8



Gambar 2.8 Peta Alamat RAM Internal Real Time Clock DS12887

Sumber : Dallas Semiconductor, 2002 : 6

2.3 Liquid Crystal Display (LCD) Tipe M1632

LCD M1632 merupakan suatu jenis tampilan yang menggunakan liquid crystal dalam menampilkan suatu karakter secara dot matrik. LCD yang akan digunakan memiliki spesifikasi sebagai berikut :

1. LCD terdiri dari 32 karakter dengan dua baris dengan display dot matrik 5 x 7.
2. Karakter generator ROM dengan 192 tipe karakter.
3. Karakter generator RAM dengan 8 bit karakter.
4. 80x8 bit display data RAM.
5. Dapat di-interface-kan ke MCU dengan system 8 atau 4 bit data.

6. Dilengkapi fungsi tambahan seperti : *display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift.*
7. Internal Data.
8. Internal otomatis *reset* pada saat *power on*.
9. Tegangan 5 volt PSU tunggal.

LCD ini mempunyai konsumsi daya yang relative rendah dan terdapat sebuah kontroler CMOS di dalamnya. Kontroler tersebut sebagai pembangkit dari karakter ROM/RAM dan display data RAM. Semua fungsi tampilan diatur oleh suatu instruksi dan modul LCD dapat dengan mudah diantarmukakan dengan mikrokontroler. Masukan yang diperlukan untuk mengendalikan modul ini berupa bus data yang masih termutiflex dengan bus alamat serta 3 bit sinyal kontrol. Sementara pengendalian dot matrik LCD dilakukan secara internal oleh kontroler yang sudah ada pada modul LCD.

Dasar-dasar pengoperasian LCD ini terdiri atas pengoperasian dasar pada register, *busy flag, address counter, display data RAM*.

1. Register

Kontroler dari LCD mempunyai dua buah register 8 bit yaitu register instruksi (IR) dan register data (DR). IR menyimpan instruksi seperti *display clear, cursor shift, and display data RAM (DDRAM)* serta *character generator RAM (CGRAM)*. DR menyimpan data untuk ditulis di DDRAM atau CGRAM ataupun membaca data dari DDRAM atau CGRAM. Ketika data ditulis ke DDRAM atau CGRAM maka DR secara otomatis menulis data ke DDRAM atau CGRAM. Ketika data pada DDRAM atau CGRAM akan dibaca maka alamat data ditulis pada IR sedangkan data akan dimasukkan melalui DR dan mikrokontroler membaca data dari DR.

2. *Busy Flag*

Busy flag menunjukkan bahwa modul siap menerima instruksi selanjutnya. Sebagaimana yang terlihat pada table 2.4, register seleksi sinyal akan melalui DB7 jika RS=0 dan R/W=1. jika bernilai 1 maka modul LCD sedang melakukan kerja internal dan instruksi tidak akan diterima. Oleh karena itu status dari *flag* harus diperiksa sebelum melaksanakan instruksi selanjutnya.

Tabel 2.5. Register seleksi

RS	R/W	OPERASI
0	0	Seleksi IR. IR Write Display Clear
0	1	Busy Flag (DB7) @ counter (DB0-DB7) Read
1	0	Seleksi DR. DR Write
1	1	Seleksi DR. DR Write

Sumber : Seiko Instruments, 1987: 8

3. Address Counter

Address Counter menunjukkan lokasi memori dalam modul LCD. Pemilihan lokasi alamat itu diberikan lewat register instruksi (IR). Ketika data dibaca atau ditulis dari DDRAM atau CGRAM maka *address counter* secara otomatis menaikkan atau menurunkan alamat tergantung dari *entry mode set*.

4. Display Data RAM (DDRAM)

Pada LCD masing-masing pin mempunyai range alamat tersendiri. Alamat itu diekspresikan dalam bilangan hexadecimal. Untuk line 1 range alamat berkisar antara 00H-0FH sedangkan untuk line 2 range alamat berkisar antara 40H-4FH.

5. Character Generator ROM (CGROM)

CGROM mempunyai tipe dot matrik 5x7. Dimana pada LCD telah tersedia ROM sebagai pembangkit karakter dalam kode ASCII.

6. Character Generator RAM (CGRAM)

CGRAM berfungsi untuk pembuatan karakter tersendiri melalui program.

LCD sendiri bekerja dengan sistem pemisahan antara instruksi dengan pemberian data. Menggunakan pin RS sebuah LCD dapat mengetahui bila data yang masuk merupakan sebuah instruksi ataukah data karakter. Selain itu sebuah LCD mempunyai kemampuan membaca dan menulis. Walaupun sangat jarang dipakai, LCD dapat digunakan untuk mengetahui posisi karakter pada posisi tertentu. Secara umum biasanya sebuah LCD hanya digunakan pada proses menulis (Nalwan ,2004: 10).

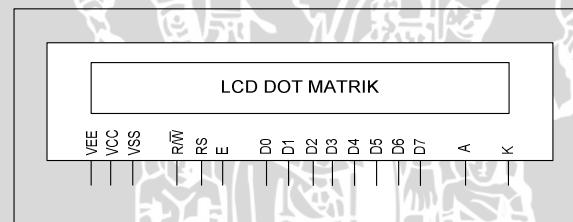
Instruksi-instruksi yang digunakan untuk pemrograman LCD dapat dilihat dalam Tabel 2.6.

Tabel 2.6. Instruksi Pemrograman LCD

No	Instruksi	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0
1	Display Clear	0	0	0	0	0	0	0	0	0	1
2	Cursor Home	0	0	0	0	0	0	0	0	1	0
3	Entry Mode Set	0	0	0	0	0	0	0	1	1/0	S
4	Display ON/OFF	0	0	0	0	0	0	1	D	C	B
5	Cursor Display Shift	0	0	0	0	0	1	S/C	R/L	0	0
6	Function Set	0	0	0	0	1	DL	N	F	0	0
7	CGRAM Address Set	0	0	0	1						Alamat Karakter
8	DDRAM Address Set	0	0	1							Tampilan Alamat Data
9	BF/Address Set	0	1	BF							Alamat Bus
10	Penulisan ke CG RAM Atau DDRAM	1	0								Byte karakter
11	Pembacaan CGRAM Atau DDRAM	1	1								Byte karakter

Sumber : Seiko Instruments., 1987: 16

Untuk lebih jelas mengenai gambaran sebuah LCD dengan susunan pin pinnya dapat dilihat dalam Gambar 2.9

**Gambar 2.9** LCD dan Konfigurasi Pinnnya

Sumber : Seiko Instruments, 1987 : 2

Fungsi dari masing-masing penyematan LCD M1632 dapat dilihat dalam Tabel 2.7 berikut :

Tabel 2.7 Fungsi Masing-Masing Penyematan LCD

Nama Penyematan	Fungsi
D0-D7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan LCD
Enable (E)	Sinyal operasi awal. Sinyal ini akan mengaktifkan data tulis atau baca
R/W	Sinyal seleksi tulis dan baca : 0 = tulis 1 = baca
RS	Sinyal pemilih register internal : 0 = instruksi register (tulis) 1 = data register (tulis dan baca)
V _{EE}	Untuk mengendalikan kecerahan LCD dengan mengubah-ubah nilai resistor variable yang dihubungkan padanya
V _{CC}	Catu daya +5V
V _{SS}	Terminal ground

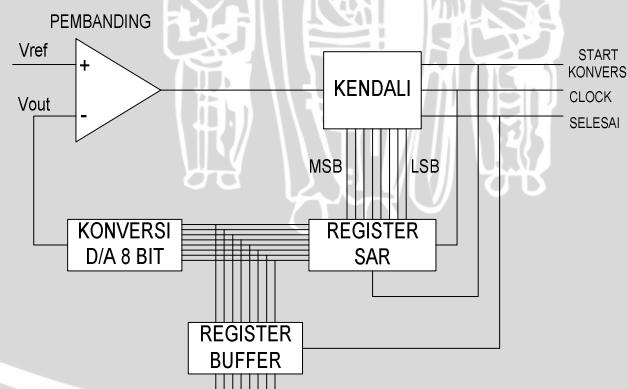
Sumber : Seiko Instruments, 1987: 7

Bila arus basis melebihi IB(sat), arus kolektor tidak dapat bertambah lagi karena diode kolektor tidak lagi berfungsi (tidak mendapat prategangan balik). Dengan kata lain, IC(sat) merupakan harga maksimum dari arus kolektor yang dapat dihasilkan oleh rangkaian dengan prategangan basis pada tegangan catu dan hambatan kolektor tertentu.

Sebagaimana telah disinggung sebelumnya, semua titik yang terletak di antara titik putus dan titik jenuh merupakan titik-titik operasi daerah aktif dari transistor. Dalam daerah operasi aktif ini, diode emiter dalam prategangan maju dan diode kolektor mendapat prategangan balik. Titik ini disebut titik tenang (*quiescent*) karena operasi pada titik ini berlangsung tanpa sinyal masukan ac.

2.5 ADC0809

ADC0809 adalah komponen akuisisi data CMOS monolit dengan *analog-to-digital converter* 8-bit, multiplekser 8-kanal dan kontrol logika. A/D converter 8-bit ini menggunakan *successive approximation* sebagai metode konversinya. *Successive approximation* conversion atau pendekatan bertingkat yang memiliki waktu konversi jauh lebih singkat dan tidak tergantung pada nilai masukan analognya atau sinyal yang akan diubah. Dalam Gambar 2.10, memperlihatkan diagram blok ADC tersebut.

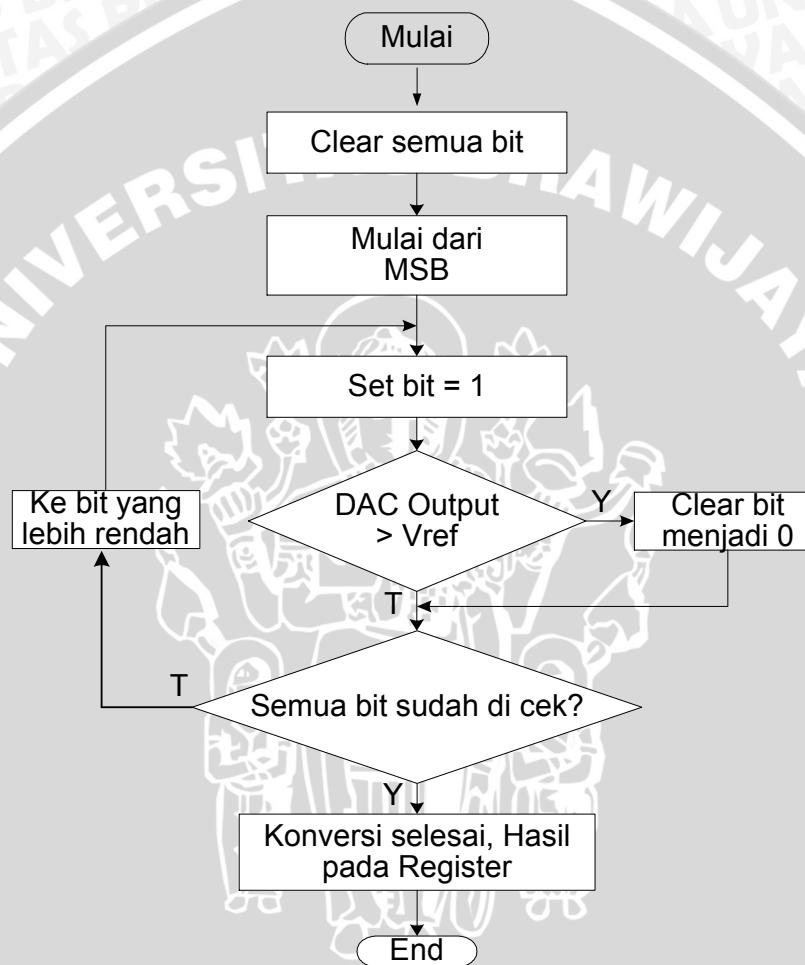


Gambar 2.10. Diagram Blok ADC

Sumber: National Semiconductor, 1995: 59

Secara singkat prinsip kerja dari konverter A/D adalah semua bit-bit diset kemudian diuji, dan bilamana perlu sesuai dengan kondisi yang telah ditentukan. Dengan rangkaian yang paling cepat, konversi akan diselesaikan sesudah 8 clock, dan keluaran D/A merupakan nilai analog yang ekivalen dengan nilai register SAR.

Apabila konversi telah dilaksanakan, rangkaian kembali mengirim sinyal selesai konversi yang berlogika rendah. Sisi turun sinyal ini akan menghasilkan data digital yang ekivalen ke dalam register buffer. Dengan demikian, keluaran digital akan tetap tersimpan sekalipun akan di mulai siklus konversi yang baru. Diagram alir berikut merupakan penjelasan singkat mengenai proses kerja ADC.



Gambar 2.11. Diagram Alir Proses Konversi ADC

Sumber: National Semiconductor, 1995: 60

Waktu konversi selama 100 μ s sangat cepat untuk pengukuran dengan perubahan lambat. Devais ini tidak membutuhkan pengaturan nol dan skala penuh serta memiliki *interface* yang mudah. Resolusi 8-bit mencukupi untuk aplikasi pengukuran yang tidak memerlukan ketelitian tinggi.

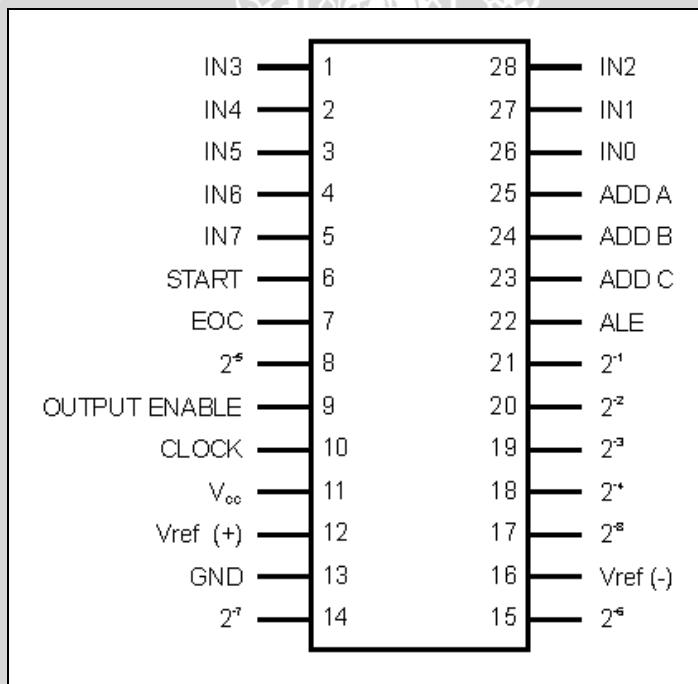
Tabel 2.8 menunjukkan keadaan masukan untuk jalur alamat untuk memilih kanal masukan analog. Alamat di-*latch* pada dekoder pada transisi dari rendah ke tinggi pada sinyal *Address Latch Enable* (ALE).

Tabel 2.8 Masukan Jalur Alamat untuk Memilih Kanal Analog

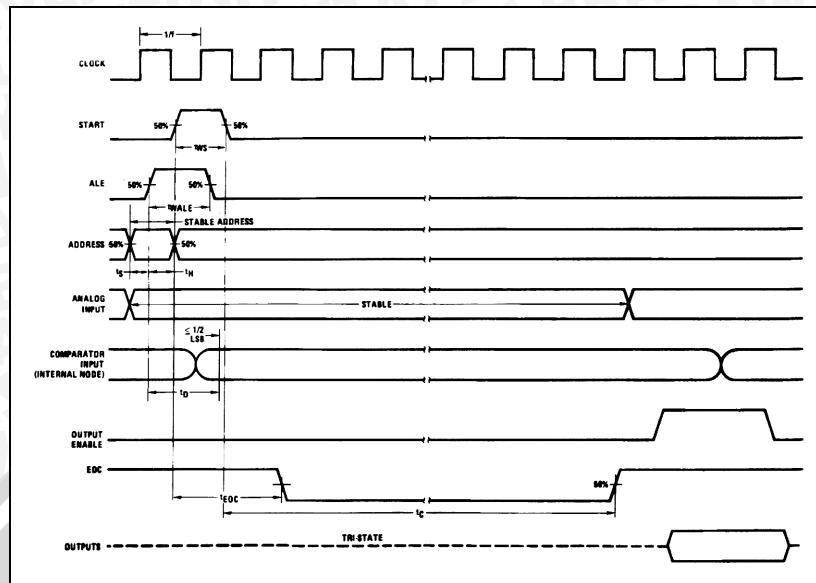
Kanal Analog Yang Dipilih	Jalur Alamat		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

Sumber: National Semiconductor, 1995: 54.

ADC ini beroperasi dalam mode *handshake* sehingga memerlukan sinyal-sinyal *handshake* untuk bekerja. Proses konversi akan dilakukan jika mendapat sinyal *start* dan kemudian ADC memberikan sinyal EOC (*End of Conversion*) setelah proses konversi selesai. Gambar 2.10 menunjukkan diagram pewaktuan ADC 0809.

**Gambar 2.10** Susunan Pena ADC0809.

Sumber: National Semiconductor, 1995: 56.



Gambar 2.11 Diagram Pewaktuan ADC 0809

Sumber: National Semiconductor, 1995: 56.

Untuk menentukan nilai tegangan untuk tiap step pada keluaran ADC maka digunakan persamaan sebagai berikut :

$$V = \frac{V_{ref}}{2^n - 1} \quad \dots \dots \dots \quad (2.2)$$

dimana : V = nilai tegangan untuk tiap step pada keluaran ADC,

V_{ref} = tegangan referensi yang digunakan oleh ADC,

N = jumlah bit pada ADC.

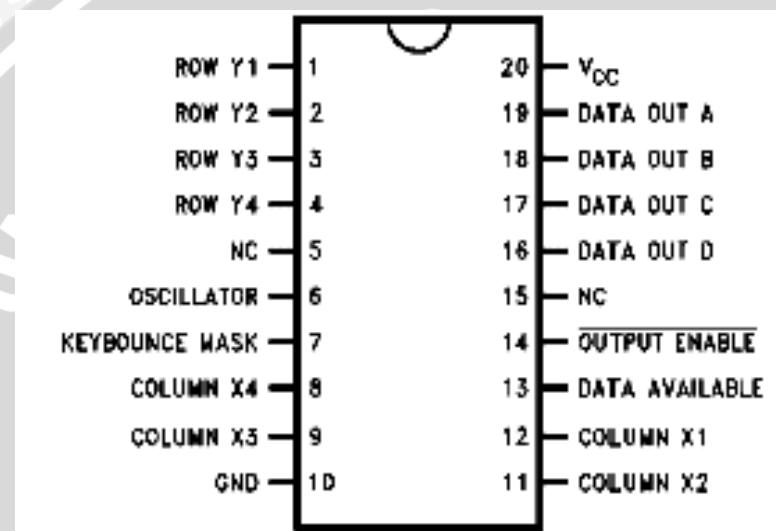
2.6 Enkoder MM74C922

MM74C922 adalah IC encoder. IC ini mempunyai *on-chip pull-up devices* yang bias digunakan untuk switch yang mempunyai resistansi di atas 50k. Tidak memerlukan dioda pada switch array untuk menghilangkan *ghost switches*. Internal debounce circuit hanya memerlukan sebuah kapasitor eksternal. Data keluaran yang dihasilkan menjadi level high ketika entri keyboard yang valid telah dibuat. Data output kembali berlevel tegangan rendah ketika tombol enter dilepas, walaupun tombol lain ditekan. Register internal menyimpan data dari tombol yang terakhir ditekan walupun setelah itu tombol dilepas. Keluaran 3-state menyediakan kemudahan ekspansi dan operasi bus.

Spesifikasi :

- Resistansi *switch* maksimum 50 kΩ
- *On* atau *off chip clock*

- *On-chip row pull-up devices*
- *key roll-over*
- Eliminasi *keybounce* dengan kapasitor tunggal
- Register yang menyimpan keluaran dari tombol yang terakhir ditekan
- Keluaran 3-STATE yang kompatibel dengan LPTTL
- *Range* tegangan *power supply* yang lebar yaitu 3 V to 15 V
- Konsumsi daya yang rendah



Gambar 2.12 Konfigurasi IC MM74C922

Sumber : Fairchild Semiconductor , 2001: 2

Tabel 2.9. Data Output Enkoder MM74C922

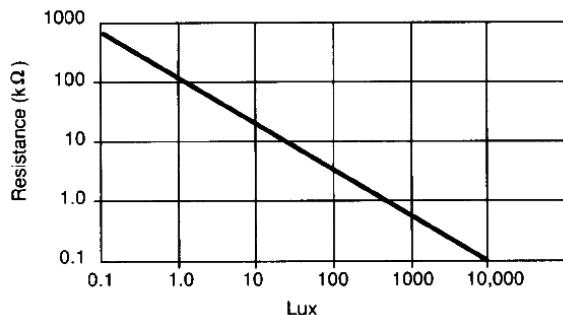
Posisi Switch	0 Y1,X1	1 Y1,X2	2 Y1,X3	3 Y1,X4	4 Y2,X1	5 Y2,X2	6 Y2,X3	7 Y2,X4	8 Y3,X1	9 Y3,X2	10 Y3,X3	11 Y3,X4
D												
A A	0	1	0	1	0	1	0	1	0	1	0	1
T B	0	0	1	1	0	0	1	1	0	0	1	1
A C	0	0	0	0	1	1	1	1	0	0	0	0
O D	0	0	0	0	0	0	0	0	1	1	1	1
U E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0
T												
Posisi Switch	12 Y4,X1	13 Y4,X2	14 Y4,X3	15 Y4,X4	16 Y5(Note 1),X1	17 Y5(Note 1),X2	18 Y5(Note 1),X3	19 Y5(Note 1),X4				
D												
A A	0	1	0	1	0	1	0	0	1	0	1	1
T B	0	0	1	1	0	0	1	1	0	1	1	1
A C	1	1	1	1	0	0	0	0	0	0	0	0
O D	1	1	1	1	0	0	0	0	0	0	0	0
U E (Note 1)	0	0	0	0	1	1	1	1	1	1	1	1
T												

Sumber : Fairchild Semiconductor , 2001: 2

2.7 Sensor LDR (Light Dependent Resistors)

LDR (*Light Dependent Resistors*) mengandung bahan *Cadmium Shulphide* yang mana nilai resistansinya berubah berdasarkan level cahaya. Apabila intensitas cahaya

semakin tinggi atau cahaya semakin terang, maka nilai resistansinya akan semakin menurun. Karakteristik LDR tersebut dapat dilihat pada Gambar 2.13



Gambar 2.13. Grafik Karakteristik LDR

Sumber : RS Components, 1997: 1

2.8 Pemberian pakan untuk ikan hias

Dalam memberikan pakan bagi ikan hias banyak persoalan yang akan timbul, diberi makanan yang berlebihan akan dapat mencemarkan lingkungannya, diberi makanan yang kurang akan menghambat pertumbuhan, bahkan dapat menyebabkan kematian. Oleh karena itu jangan memberikan makanan yang berlebihan dan ukurannya pun harus disesuaikan dengan bukaan mulut ikan, supaya makanan tersebut dapat dilahap semuanya. Dalam memberi pakan, yang harus diperhatikan secara cermat ialah jumlah pakan yang harus diberikan dan gizi makanan yang terkandung dalam pakan tersebut.

Salah satu faktor yang menunjang keberhasilan pemeliharaan ikan ialah penyediaan makanan secara cukup dan kontinyu, terutama makanan yang dapat diberikan untuk berbagai tingkatan umur serta ukuran ikan.

Makanan untuk ikan jangan diberikan dengan ditaburkan sembarangan, tetapi harus dilepaskan ke dalam air secara perlahan-lahan pada posisi samping tempat ikan berada. Bila memungkinkan , ikan dapat diberi makanan setiap hari secara rutin dan pada waktu yang sama. Misalnya pada jam 7 pagi dan jam 5 sore. Perlakuan semacam ini dapat merangsang naluri ikan agar terlatih untuk mengetahui kapan dan di mana pakan diberikan. Oleh sebab itu, usahakan memberi makanan pada titik sudut dan waktu yang sama, misalnya makanan selalu diberikan pada ujung sebelah kanan atau ujung sebelah kiri akuarium (Effendy, 1990: 39).

BAB III

METODOLOGI

3.1 Studi Literatur

Studi literatur dilakukan untuk mendapatkan data-data yang mencakup prinsip kerja dan landasan teori yang terkait dengan komponen yang akan digunakan dalam perancangan dan pembuatan alat ini, yaitu meliputi karakteristik dan prinsip kerja mikrokontroller AT89S8252, IC DS12887, dan modul tampilan LCD .

Data-data tersebut diperoleh dari sumber-sumber referensi yang relevan, baik dari buku-buku, majalah-majalah, ataupun sumber-sumber informasi dari internet.

3.2 Perancangan Alat

Perancangan alat ini disesuaikan dengan fungsi dari komponen-komponen yang akan digunakan sesuai dengan literatur yang dipelajari sehingga diperoleh rangkaian elektronika yang dapat direalisasikan.

Adapun pada perancangan alat ini dibagi ke dalam 3 perancangan, yaitu : perancangan mekanik, *hardware* dan *software*.

- Merancang mekanik, yaitu pembuatan bak penampung pakan, katup pembuka, dan timbangan pakan.
- Merancang *hardware*, yaitu blok diagram rangkaian secara keseluruhan dan rangkaian per blok yang meliputi : mikrokontroler AT89S8252, rangkaian penggerak, rangkaian sensor LDR, keypad dan rangkaian LCD.
- Merancang *software* dengan membuat algoritma program.

3.3 Pengujian dan Analisis Alat

Pengujian alat ini meliputi pengujian *hardware* dan *software* :

- Pengujian *hardware* meliputi pengujian perblok dan keseluruhan berupa pengujian sistem kontrol pada mikrokontroler AT89S8252, rangkaian sensor LDR, *keypad*, dan rangkaian LCD.
- Pengujian *software* meliputi pengujian per subrutin dan keseluruhan.

3.4 Prosedur yang Digunakan

- Studi literatur, yaitu pengambilan data untuk mengetahui spesifikasi komponen yang akan digunakan dalam merencanakan pembuatan perangkat keras (hardware)
- Pemilihan komponen yang digunakan berdasarkan pada data literature untuk merealisasikan perencanaan pembuatan PCB (*Printed Circuit Board*). Pemilihan komponen berdasarkan *databook* yang disesuaikan dengan komponen yang ada di pasaran.
- Pembuatan mekanik
- Merakit semua komponen yang digunakan dalam PCB (*Printed Circuit Board*).
- Penyesuaian mekanik yang telah dibuat dengan perangkat keras yang telah ada.
- Perencanaan dan pembuatan perangkat lunak berdasarkan perangkat keras dan diagram alir yang telah dibuat dengan menggunakan bahasa pemrograman.
- Pengujian alat

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Bab ini membahas perancangan dan pembuatan alat pemberi pakan ikan otomatis. Perancangan dan pembuatan alat ini meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*).

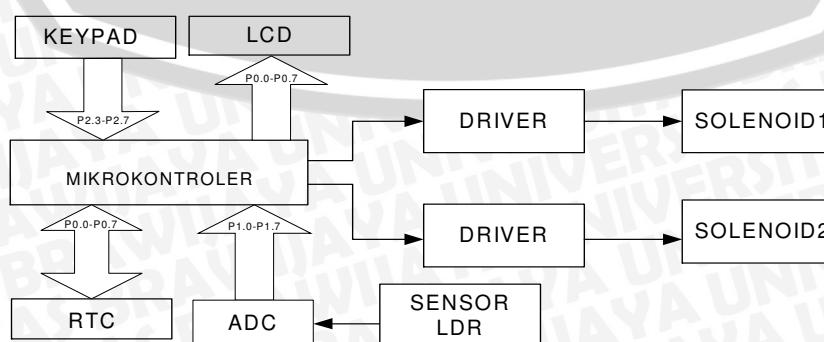
4.1 Gambaran Umum

Pada alat pemberi pakan ikan otomatis dengan menggunakan mikrokontroler AT89S8252 terdapat *keypad* yang berfungsi sebagai masukan, data masukan pada *keypad* akan diproses oleh mikrokontroler AT89S8252. LDR akan dipasang pada alat penakar yang berfungsi sebagai sensor untuk menentukan massa pakan yang ditimbang. Resolusi timbangan dirancang sebesar 1 gram, untuk membaca perubahan massa pada timbangan yang menggunakan sensor LDR yang menghasilkan keluaran analog, maka digunakan ADC0809 agar mikrokontroler bisa membaca perubahan massa dari hasil penimbangan. Mikrokontroler akan membaca data dari ADC dan mengontrol solenoid sebagai penggerak buka atau tutup katup pada tabung penyimpan pakan. Selain itu komponen penunjang seperti RTC (Real Time Clock) yang berfungsi sebagai penunjuk waktu.

Spesifikasi alat pemberi pakan ikan otomatis adalah sebagai berikut :

- Mikrokontroler AT89S8252 sebagai pengendali sistem
- Tampilan menggunakan LCD 2 x 16 karakter tipe M1632
- Menakar per 1 gram takaran

4.2 Blok Diagram Rangkaian



Gambar 4.1. Diagram Blok Perancangan

Keterangan Blok diagram:

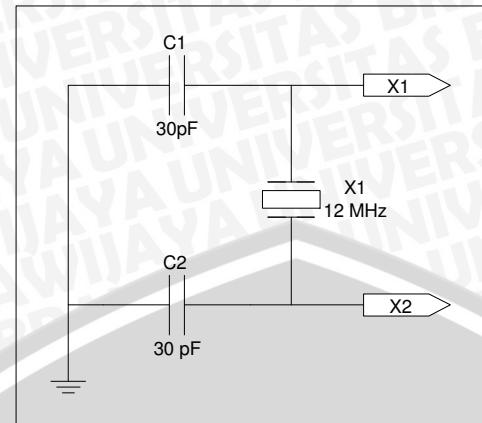
- Keypad berfungsi sebagai masukan pada Mikrokontroler, data masukan berupa waktu pemberian pakan, banyaknya takaran, frekwensi pemberian pakan setiap harinya ataupun untuk penyetelan waktu (tanggal, jam, menit, dan detik).
- LCD berfungsi sebagai alat penampil yang dapat menampilkan waktu real time (tanggallan, dan jam), ataupun tampilan menu yang menampilkan pilihan pada saat pengeditan jadwal pemberian pakan .
- RTC adalah IC yang merupakan rangkaian jam dan kalender, berfungsi sebagai patokan waktu pemberian pakan.
- LDR pada penakar berfungsi sebagai sensor pada penakar yang dapat mengetahui massa pakan yang ditampung pada penakar.
- Switching transistor berfungsi sebagai saklar untuk menggerakkan solenoid, apabila mikrokontroler berlogika high maka solenoid akan dicatuh, sebaliknya bila mikrokontroler berlogika low maka solenoid tidak aktif.
- Solenoid berfungsi sebagai penggerak katup, apabila solenoid dicatuh, maka solenoid akan menarik katup, sehingga katup terbuka dan alat pemberi pakan mengeluarkan pakan. Apabila solenoid tidak dicatuh, ia akan melepaskan katup dan katup tertutup.

4.3 Perangkat Keras

4.3.1 Mikrokontroler AT89S8252

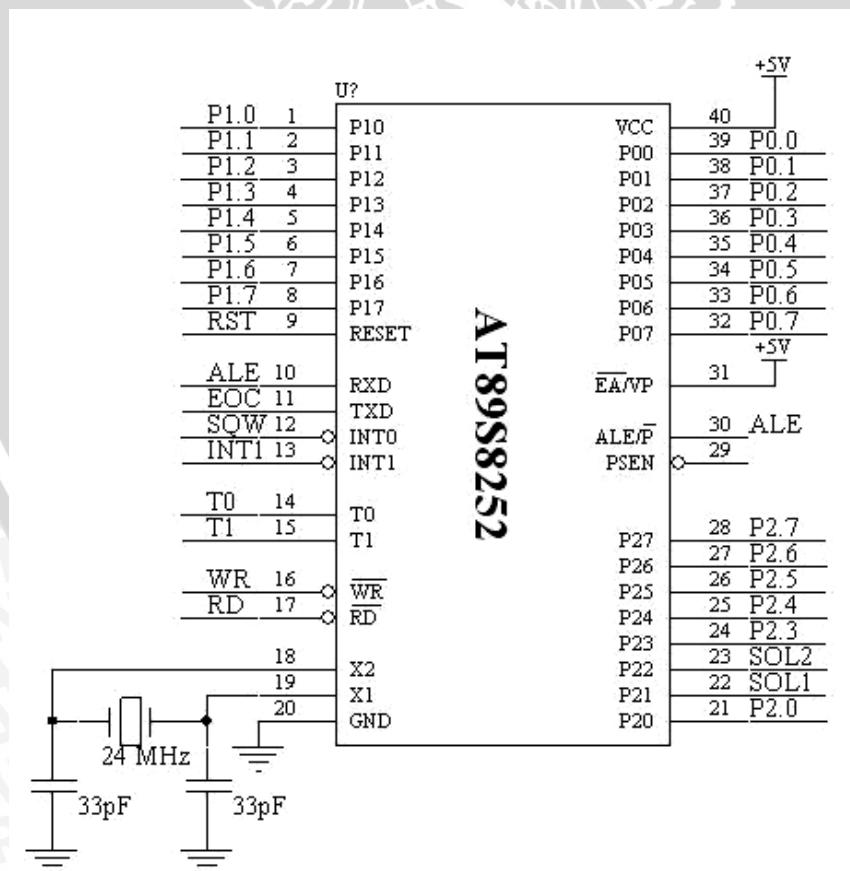
Mikrokontroler AT89S8252 digunakan sebagai pengendali sistem yang berfungsi mengatur gerak katup sesuai dengan masukan dari keypad dan sensor LDR.

Penyemant X1 dan X2 pada AT89S8252 dihubungkan dengan sebuah crystal 24 MHz dan 2 buah kapasitor 33 pF yang akan membangkitkan pulsa clock yang digunakan sebagai penggerak bagi sejumlah operasi internal CPU. Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber clock yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini menggunakan osilator internal yang telah tersedia dalam chip AT89S8252. Untuk menentukan frekuensi osilator cukup dengan menghubungkan kristal dalam pin 19 (X1) dan pin 18 (X2) serta dua buah kapasitor ke ground. Gambar 4.2 menunjukkan rangkaian clock yang direncanakan.



Gambar 4.2. Rangkaian Clock Mikrokontroler

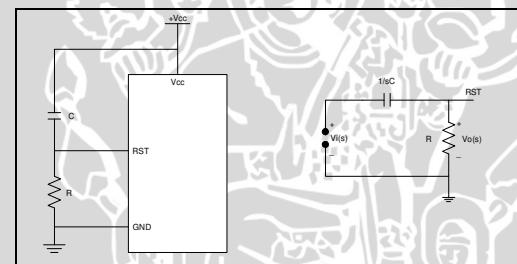
Agar sebuah mikrokontroler dapat bekerja sebagai pengontrol, maka kaki-kaki / port mikrokontroler dihubungkan dalam rangkaian-rangkaian eksternal. Dalam perancangan ini port yang diinginkan adalah sebagai berikut :



Gambar 4.3. Rangkaian Mikrokontroler AT89S8252

4.3.2 Reset mikrokontroller

Rangkaian *power on reset* diperlukan untuk mereset mikrokontroler secara otomatis setiap kali catu daya dinyalakan, hal ini akan me-reset *program counter* sehingga perintah program yang dieksekusi dimulai pada alamat 0. Ketika catu daya diaktifkan, rangkaian *power on reset* menahan logika tinggi pin RST dengan jangka waktu yang ditentukan oleh lamanya pengisian muatan C. Jika pin RST diberi logika *high* selama dua siklus mesin maka mikrokontroler akan direset. Satu siklus mesin ($\frac{12}{f.osc}$) membutuhkan waktu $1,085 \mu\text{s}$, sehingga untuk mereset mikrokontroler dibutuhkan waktu minimal $2,17 \mu\text{s}$. Tegangan RST yang diijinkan sebesar $0,7 \text{ Vcc} - \text{Vcc} + 0,5$. Jika tegangan RST minimal $3,5 \text{ V}$ maka tegangan pada kapasitor maksimal $1,5 \text{ V}$ untuk dapat mereset mikrokontroler. Rangkaian *power on reset* dibentuk dengan rangkaian RC yang dapat dilihat dalam Gambar 4.4



Gambar 4.4. Rangkaian *Power On Reset* dan Setaranya.

Dari rangkaian setara diperoleh:

$$\begin{aligned} V_o &= \frac{R}{R + \frac{1}{sC}} \cdot V_i \\ V_o &= \frac{sCR}{sCR + 1} \cdot V_i \quad \dots \dots \dots \quad (4.1) \end{aligned}$$

dengan tegangan V_i adalah V_{cc} yaitu 5 V , dalam fungsi Laplace adalah $\frac{5}{s}$ sehingga:

$$\begin{aligned} V_o &= \frac{5}{s} \cdot \frac{sCR}{sCR + 1} = 5 \cdot \frac{CR}{sCR + 1} = 5 \cdot \left[\frac{1}{s + \frac{1}{RC}} \right] \\ V_o &= 5 \cdot e^{-\frac{t}{RC}} \end{aligned}$$

$$\frac{5}{V_o} = e^{\frac{t}{RC}}$$
$$\ln \frac{5}{V_o} = \frac{t}{RC}$$
$$t = RC \left[\ln \frac{5}{V_o} \right] \dots\dots\dots(4.2)$$

dengan V_o adalah tegangan logika nominal yang diijinkan pin RST dimana

$$V_o = 0,7 \times V_{cc} = 0,7 \times 5 = 3,5 \text{ volt}$$

sehingga

$$t = RC \left[\ln \frac{5}{3,5} \right]$$

$$t = 0,357 \cdot R \cdot C$$

dengan menggunakan permasalan $R = 10 \text{ k}\Omega$ dan t minimum adalah $2,17 \text{ us}$ maka untuk amannya dimisalkan $= 4 \text{ us}$ maka:

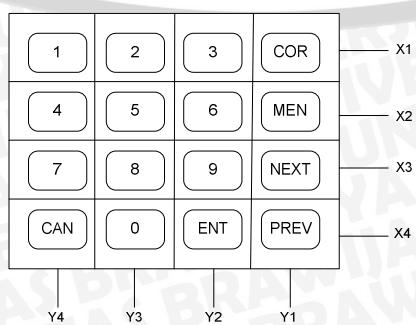
$$4 \cdot 10^{-6} = 0,357 \times 10^3 \times C$$

$$C = 1 \text{ nF}$$

telah memenuhi syarat minimal untuk dapat mereset mikrokontroler AT89S8252. Dalam perancangan menggunakan $10 \mu\text{F}$ dan $56 \text{ k}\Omega$.

4.3.3 Papan Tombol (*Keypad*)

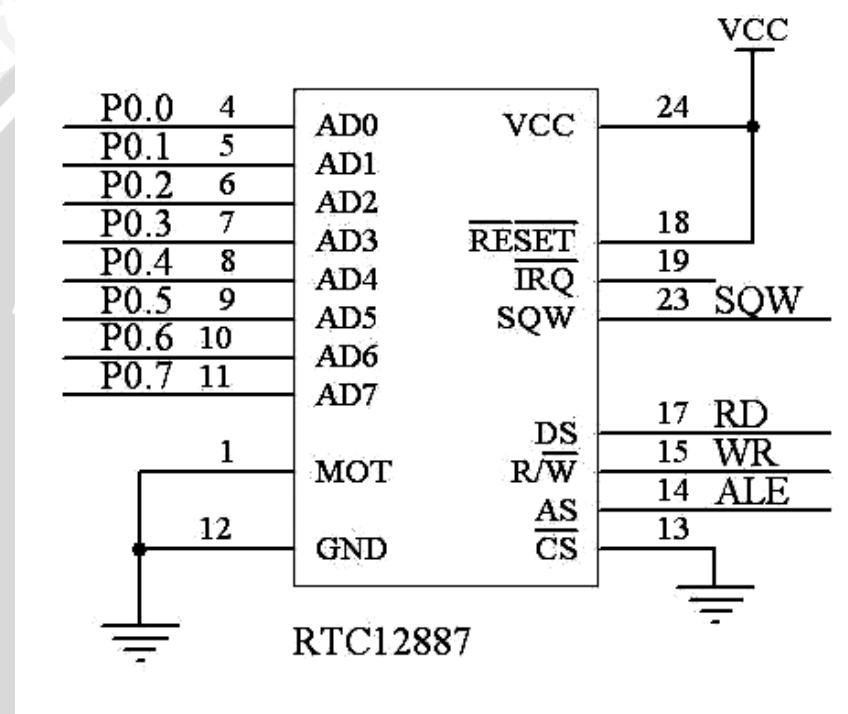
Keypad yang digunakan adalah *keypad* matrik 4×4 . *keypad* matrik ini bekerja dengan menggunakan prinsip scanning pada baris dan kolom. Jika terdeteksi adanya persambungan antara baris dan kolom yang valid, maka software akan mendekodekan baris dan kolom mana yang menyambung menjadi data biner 4 bit. Untuk X1 sampai dengan X3 pada *keypad* dihubungkan ke P1.0 sampai P1.3 pada mikrokontroler dan untuk Y1 sampai Y4 pada *keypad* dihubungkan ke P1.4 sampai P1.7 pada mikrokontroler.



Gambar 4.5. Susunan *Keypad* untuk Mikrokontroler

4.3.4 RTC

Rangkaian RTC yang digunakan adalah RTC DALLAS DS1287. RTC Ini tidak membutuhkan komponen eksternal seperti kristal, kapasitor dan resistor, sehingga pin-pinnya yang tidak terpakai tidak dimunculkan dalam RTC DALLAS ini. Disamping itu, RTC ini dilengkapi dengan catu daya *back-up* yang memungkinkan sistem pemarkutan tetap berjalan walaupun catu daya sistem dimatikan. Antarmuka RTC dengan sistem mikrokontroler AT89S8252 ditunjukkan dalam Gambar 4.6



Gambar 4.6. Rangkaian RTC 12887

Sistem mikrokontroler yang digunakan adalah mikrokontroler buatan Intel oleh karena itu pin MOD dihubungkan dengan GND. Bus data RTC dihubungkan ke bus data sistem mikrokontroler AT89S8252, sedangkan bus kontrol R/W RTC dihubungkan ke pin WR mikrokontroler AT89S8252, pin DS RTC dihubungkan ke pin RD mikrokontroler AT89S8252, dan pin AS RTC dihubungkan ke pin ALE mikrokontroler AT89S8252.

4.3.5 LCD (Liquid Crystal Display)

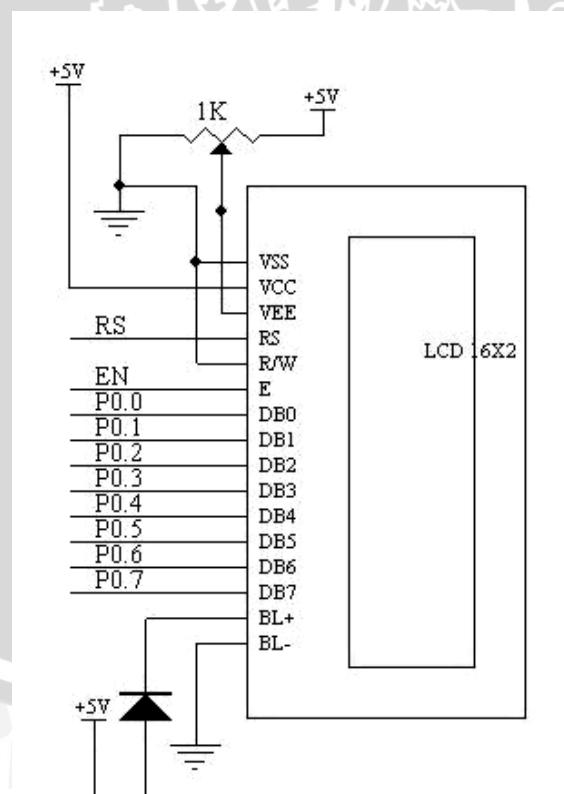
LCD diperlukan pada sistem ini juga menggunakan sistem pengalamatan. Berbeda dengan EEPROM, LCD dikhkususkan untuk melakukan proses tulis saja. Sehingga sebuah LCD hanya dapat digunakan untuk menulis perintah ataupun data.

Untuk mengaktifkan fasilitas baca tulis pada LCD, kita perlu menambahkan beberapa rangkaian logika sebagai decoder status.

Untuk membedakan apakah data yang diberikan berupa data atau instruksi kita menggunakan pin A0. sebuah LCD menganggap data yang diberikan itu sebagai instruksi ataupun data melalui pin RS. Bila pin ini berlogika rendah, otomatis data akan dianggap sebagai instruksi, bila RS berlogika tinggi itu berarti data.

Gambar rangkaian LCD dapat dilihat dalam Gambar 4.7. LCD diakses dengan menggunakan beberapa instruksi yang dapat mengatur letak kursor, tampilan dan lain-lain. Dioda digunakan sebagai pengaman dari tegangan terbalik, mengingat LCD sangat rentan dari tegangan terbalik.

Sebuah LCD memiliki tiga terminal tegangan, yakni : VCC, VSS dan VEE. Pengontrol kontras LCD difungsikan pada terminal VEE, yang mengubah setiap perubahan tegangan antara VSS dan VEE menjadi perubahan kontras pada tampilan layer LCD.



Gambar 4.7 Rangkaian LCD

Tabel 4.1. Susunan Penyemata LCD M1632

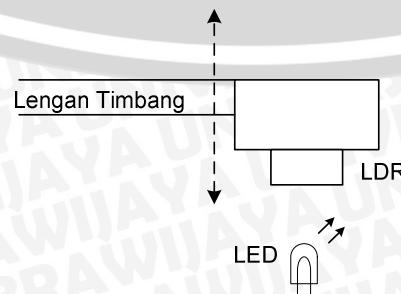
Nama Penyemata	Fungsi
D0 – D7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan pada LCD
Enable (E)	Sinyal operasi awal. Sinyal ini akan mengaktifkan data tulis atau baca.
R/W	Sinyal seleksi tulis dan baca: 0 = tulis 1 = baca
RS	Sinyal pemilih register internal : 0 = instruksi register (tulis) 1 = data register (tulis dan baca)
VEE	Untuk mengendalikan kecerahan LCD dengan mengubah – ubah nilai resistor variabel yang dihubungkan padanya.
VCC	Catu daya +5V
VSS	Terminal ground

Sumber : Seiko Instruments, 1987 : 7

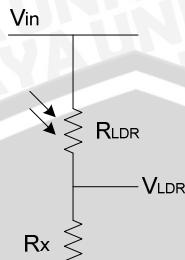
Pada perancangan menggunakan mode 8 bit, 8 jalur data LCD dihubungkan dengan Port 0 Mikrokontroler AT89S8252. LCD disetting agar selalu berada pada kondisi tulis (Write) dengan menghubungkan kaki R/W ke ground.

4.3.6. Sensor LDR dan Pengkondisi Sinyal

LDR pada perancangan alat ini digunakan sebagai sensor pada sistem timbangan yaitu dengan cara meletakkan LDR pada salah satu lengan timbangan berhadapan dengan LED secara vertikal seperti pada Gambar 4.8

**Gambar 4.8.** LDR sebagai Sensor pada Timbangan

Apabila beban pada timbangan bertambah, maka LDR akan bergerak menjauhi LED. Dalam aplikasinya LDR dirangkai dengan rangkaian pembagi tegangan seperti Rangkaian pembagi tegangan seperti Gambar 4.9 berikut:



Gambar 4.9 Rangkaian Pembagi Tegangan LDR

Dari rangkaian pada Gambar 4.9., dengan menggunakan Persamaan 4.3. dapat dihitung nilai V_{LDR} :

$$V_{LDR} = \frac{R_x}{R_x + R_{LDR}} \times V_{in} \quad \dots \dots \dots \quad (4.3)$$

Dalam perancangan dikehendaki nilai $V_{LDR} = 4$ volt pada saat beban pada timbangan nol atau dalam rancangan timbangan nilai nol tersebut dibuat pada saat jarak LED dengan LDR 10 mm, pada kondisi tersebut nilai $R_{LDR} = 25 \text{ k}\Omega$, nilai tegangan input $V_{in} = 5$ volt, dengan menggunakan Persamaan 4.3 dapat dicari nilai R_x :

$$R_x = \frac{R_{LDR} \cdot V_{LDR}}{V_{in} - V_{LDR}} \quad \dots \dots \dots \quad (4.4)$$

Dengan memasukkan nilai R_{LDR} , V_{LDR} , dan V_{in} pada Persamaan 4.4 :

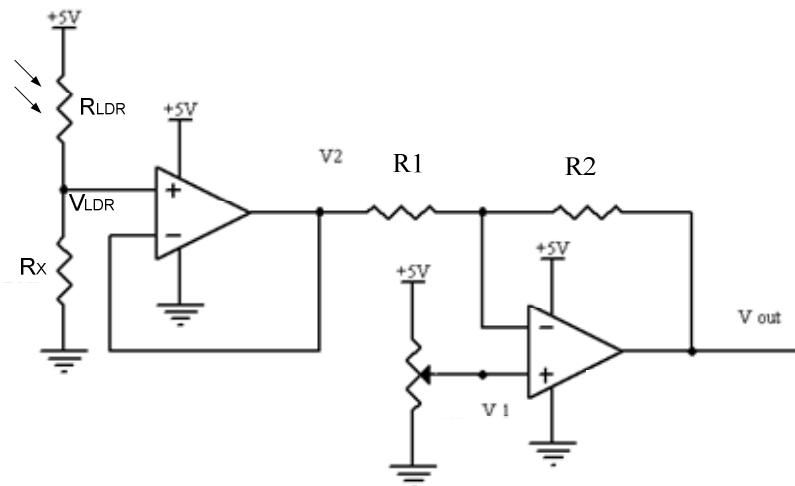
$$R_x = \frac{25 \cdot 4}{5 - 4}$$

Maka didapat nilai $R_x = 100 \text{ k}\Omega$

Berdasarkan perancangan, jarak antara LDR dan LED saat beban maksimum adalah 50 mm. Dengan menggunakan Persamaan 4.4. dapat dihitung bahwa pada saat beban timbangan maksimum (20 gram) $V_{LDR} = 3,22$ volt

Dalam perancangan dikehendaki agar tegangan yang masuk ke ADC semakin meningkat seiring dengan peningkatan massa pada timbangan, tegangan keluaran V_{out} yang dikehendaki harus berada antara 0 volt sampai 5 volt agar sesuai dengan level tegangan masukan ADC. Untuk memenuhi kondisi tersebut ,maka digunakan rangkaian

pengkondisi sinyal. Rangkaian pengkondisi sinyal tersebut terdiri dari sebuah Buffer dan penguat pembalik seperti pada Gambar 4.10:



Gambar 4.10 LDR dan Rangkaian Pengkondisi Sinyal

$$V_2 = V_{LDR}$$

V_{out} didapat dengan metode superposisi

$$V_{out1} = V_1 \left(1 + \frac{R_2}{R_1}\right)$$

$$V_{out2} = -V_2 \frac{R_2}{R_1}$$

$$V_{out} = V_{out1} + V_{out2} = V_1 \left(1 + \frac{R_2}{R_1}\right) - V_2 \frac{R_2}{R_1}$$

karena $V_2 = V_{LDR}$ maka

$$V_{out} = V_1 \left(1 + \frac{R_2}{R_1}\right) - V_{LDR} \frac{R_2}{R_1} \dots \quad (4.5)$$

Langkah selanjutnya adalah menentukan nilai R_1 dan R_2 yang sesuai sehingga dihasilkan tegangan output V_{out} yang dikehendaki, dalam perancangan V_{out} ditetapkan bernilai 1 volt saat beban timbangan nol dan bernilai 4 volt saat beban timbangan maksimum. Perhitungan nilai R_1 dan R_2 adalah sebagai berikut:

Diketahui:

- Saat beban timbangan nol:

$$V_{LDR} = 4 \text{ volt, ditentukan } V_{out} = 1 \text{ volt}$$

- Saat beban timbangan maksimum:

$$V_{LDR} = 3,22 \text{ volt, ditentukan } V_{out} = 4 \text{ volt}$$

Dengan memasukkan nilai-nilai Tegangan baik pada saat beban timbangan nol maupun pada saat beban timbangan Maksimum pada Persamaan 4.5, maka didapat Persamaan-Persamaan berikut:

- Saat beban timbangan nol:

$$1 = V_1 \left(1 + \frac{R_2}{R_1}\right) - 4 \frac{R_2}{R_1} \dots \dots \dots \quad (4.6)$$

- Saat beban timbangan maksimum:

$$4 = V_1 \left(1 + \frac{R_2}{R_1}\right) - 3,22 \frac{R_2}{R_1} \dots \dots \dots \quad (4.7)$$

Dengan cara mengurangi Persamaan 4.6 dengan Persamaan 4.7 sehingga dihasilkan Persamaan 4.8 berikut:

$$\frac{R_2}{R_1} = \frac{50}{13} \dots \dots \dots \quad (4.8)$$

Dengan menetapkan:

$$R_2 = 5 \text{ k}\Omega$$

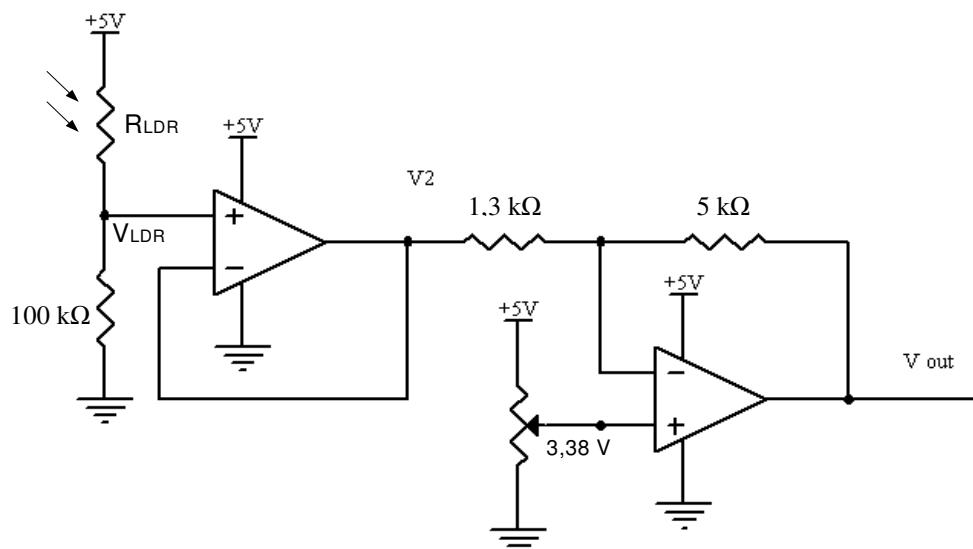
Maka berdasarkan Persamaan 4.8 R₁ dapat diketahui nilainya yaitu:

$$R_1 = \frac{13}{50} R_2 = \frac{13}{50} \cdot 5$$

$$R_1 = 1,3 \text{ k}\Omega$$

Karena nilai $\frac{R_2}{R_1}$ sudah diketahui, maka dengan memasukkan nilai $\frac{R_2}{R_1}$ pada Persamaan 4.6 ataupun pada Persamaan 4.7 dapat dihitung nilai V₁ = 3,38 volt, untuk mendapatkan nilai V₁ sebesar 3,38 volt digunakan resistor variable 10 kΩ yang dihubungkan pada catu daya 5 volt dengan rangkaian seperti terlihat pada Gambar 4.10.

Dari hasil perhitungan tersebut maka menghasilkan Rangkaian pengkondisi sinyal seperti terlihat pada Gambar 4.11 berikut:



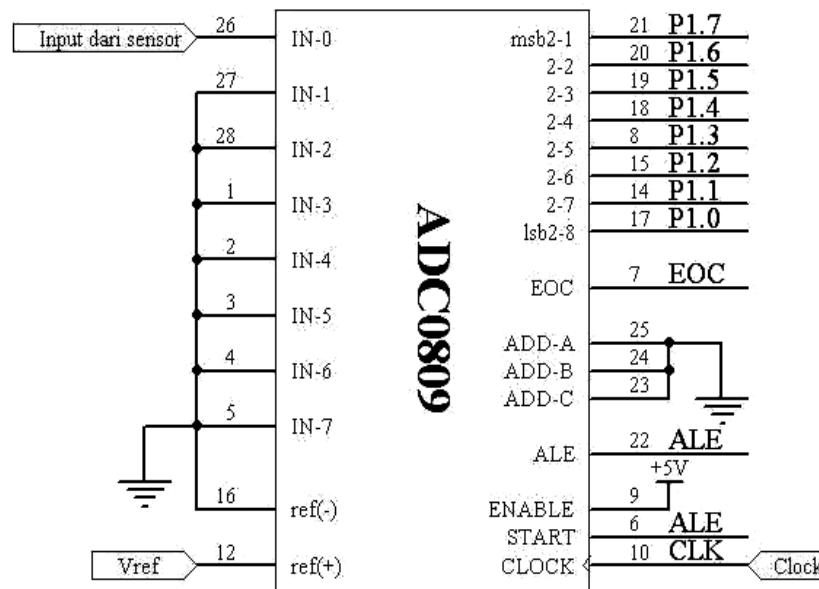
Gambar 4.11 LDR dan Rangkaian Pengkondisi Sinyal serta Nilai-Nilai Komponennya

4.3.7. ADC 0809

ADC0809 adalah komponen akuisisi data CMOS monolit dengan *analog-to-digital converter* 8-bit, multiplexer 8-kanal dan kontrol logika. A/D converter 8-bit ini menggunakan *successive approximation* sebagai metode konversinya. Waktu konversi selama $100 \mu\text{s}$ sangat cepat untuk pengukuran dengan perubahan lambat. Devais ini tidak membutuhkan pengaturan nol dan skala penuh serta memiliki *interface* yang mudah. Resolusi 8-bit mencukupi untuk aplikasi pengukuran yang tidak memerlukan ketelitian tinggi.

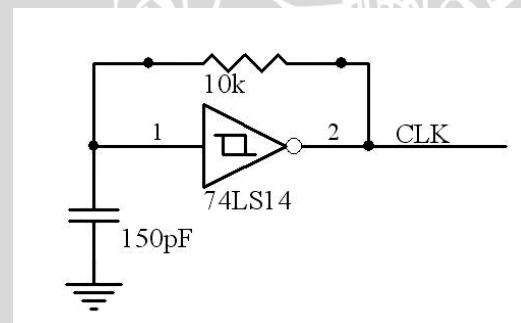
Pada perancangan rangkaian ADC berfungsi untuk mengubah sinyal analog sensor LDR menjadi sinyal digital 8 bit yang mempunyai level logika TTL sehingga dapat diolah oleh Mikrokontroler AT89S8252 yang juga memiliki level logika TTL

Pada perancangan ini dari 8 kanal masukan ADC 0809 yang digunakan untuk mengakuisisi besaran analog hanya satu sedangkan 7 kanal lainnya tidak digunakan atau terhubung ke ground.



Gambar 4.12. Rangkaian ADC0809

Dalam menjalankan proses konversi dihubungkan sebuah masukan clock yang diambil dari rangkaian pembangkit, seperti Gambar 4.13.



Gambar 4.13. Rangkaian Clock ADC0809

Berdasarkan datasheet, ADC 0809 memiliki frekuensi clock minimal 10 kHz, tipikal 640 kHz dan maksimum sekitar 1280 kHz. Frekuensi clock yang diharapkan dalam perancangan ini adalah sekitar 640 kHz. Persamaannya adalah :

$$f_{clock} = \frac{1}{1,1RC}$$

Malvino (1999:346) menyatakan “ jangkauan nilai hambatan yang lazim adalah 10 sampai 50 kΩ ”, sehingga nilai R yang digunakan 10 kΩ dan nilai kapasitor dapat ditentukan

$$\begin{aligned} C &= \frac{1}{1,1 \times 10^4 \times 640 \cdot 10^3} \\ &= 142,045 \text{ pF} \end{aligned}$$

Karena nilai kapasitor tersebut tidak ada di pasaran maka digunakan kapasitor sebesar 150 pF, sehingga frekuensi yang dihasilkan adalah

$$\begin{aligned} f_{\text{clock}} &= \frac{1}{1,1 \times 10^4 \times 150 \cdot 10^{-12}} \\ &= 606 \text{ kHz} \end{aligned}$$

Tingkat resolusi yang dimiliki oleh ADC ini adalah :

$$\begin{aligned} \text{Resolusi} &= \frac{V_{\text{ref}}}{2^8 - 1} \\ &= \frac{2,4}{255} \\ &= 0,00941 \text{ V/bit} \end{aligned}$$

Jadi setiap 1 bit mewakili kenaikan sebesar 9,41 mV.

Waktu konversi (waktu untuk pengubahan 1 cuplikan) untuk ADC jenis SAC ditentukan berdasarkan jumlah bit keluarannya.

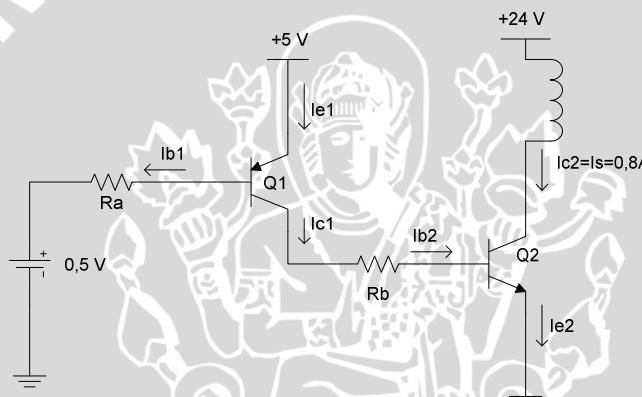
$$\begin{aligned} \text{Waktu konversi} &= 8 \text{bit} \times \frac{1}{f_{\text{clock}}} \\ &= 8 \text{bit} \times \frac{1}{606 \cdot 10^3} \\ &= 0,0132 \text{ ms} \end{aligned}$$

Add A, B dan C adalah jalur pengalamanan saluran masukan analog ADC. Dalam perancangan ini jumlah input hanya satu sehingga add A, B, dan C dihubungkan ke ground. ALE (*Address Latch Enable*) berfungsi sebagai penahan alamat. Jika jalur ALE ADC 0809 memperoleh logika tinggi, alamat yang diberikan pada jalur alamat add A, B dan C akan ditahan. Proses pengkonversian akan mulai berlangsung saat start berlogika tinggi. Selama proses pengkonversian, EOC akan berlogika rendah. Untuk menandakan bahwa proses konversi selesai dan data valid telah tersedia, EOC berubah dari logika 0 ke logika 1. OE berfungsi untuk mengatur apakah data dapat dikeluarkan dalam bentuk keluaran digital ADC. Dalam perancangan ini, OE selalu dibuat dalam kondisi high (logika 1).

4.3.8. Driver Solenoid

Solenoid berfungsi sebagai penggerak katup, baik katup untuk bak penampung maupun katup yang digunakan untuk timbangan. Prinsip kerja solenoid ini berdasarkan prinsip elektromagnetik, solenoid akan menggerakkan katup apabila dialiri arus listrik, untuk mengerakan solenoid diperlukan suatu rangkaian penggerak. Gambar rangkaian penggerak solenoid ditunjukkan dalam Gambar 4.14.

Dioda dipasang paralel dengan kumparan solenoid untuk menghubung singkatkan arus yang tersimpan pada kumparan solenoid saat transistor dalam keadaan *cut off*. Selain itu untuk melindungi arus transien yang diakibatkan oleh kumparan solenoid.



Gambar 4.14. Rangkaian Driver Solenoid

Tegangan pada solenoid sebesar 24 V, dan arus masukan solenoid sebesar 0,8 A. Dalam perancangan, untuk mengaktifkan solenoid digunakan logika *low* dari Port 2 mikrokontroler, dimana pada kondisi logika *low* tersebut mikrokontroler mengalirkan arus sebesar 1,6 mA ($I_{OL}=1,6$ mA) dan tegangan pada port tersebut maksimum sebesar 0,5 V. Kondisi tersebut mengharuskan suatu rangkaian penguat arus agar mikrokontroler dapat menggerakkan solenoid, dalam perancangan ini digunakan Rangkaian Darlington.

Perhitungan tegangan, arus, dan tahanan adalah sebagai berikut:

Diketahui :

$$I_s = 0,8 \text{ A}$$

Sedangkan $I_s = I_c$, sehingga diperlukan suatu transistor yang memiliki I_c lebih besar dari 0,8 A, dalam perancangan dipilih transistor tipe TIP31 yang diketahui dari *datasheet* :

$$I_{b2} = \frac{I_c}{h_{fe}} = 25$$

Sehingga dapat diketahui bahwa :

$$I_{b2} = \frac{I_c}{h_{fe}} = \frac{0,8}{25} = 32 \text{ mA}$$

$$I_c = I_s = 0,8 \text{ A}$$

$$h_{fe} = 25$$

sehingga dapat ditentukan bahwa :

$$I_{b2} = \frac{0,8}{25} = 32 \text{ mA}$$

$$I_{c1} = I_{b2} = 32 \text{ mA}$$

Karena $I_{b1} = I_{OL} = 1,6 \text{ mA}$ maka dapat diketahui h_{fe} transistor 1 minimal sebesar :

$$h_{fe} = \frac{I_{c1}}{I_{b1}} = \frac{32}{1,6} = 20$$

dalam perancangan transistor yang digunakan BD 139 yang mempunyai h_{fe} minimum sebesar 25 dan maksimum sebesar 250 dan I_c maksimum sebesar 1,5 A.

R_a dapat dihitung nilainya dengan:

$$R_a = \frac{V_{cc} - V_{eb1} - V_{bb}}{I_{b1}} = \frac{5 - 0,6 - 0,5}{1,6} = 2,437 \text{ k}\Omega$$

Dalam perancangan resistor yang digunakan adalah 2,5 k Ω

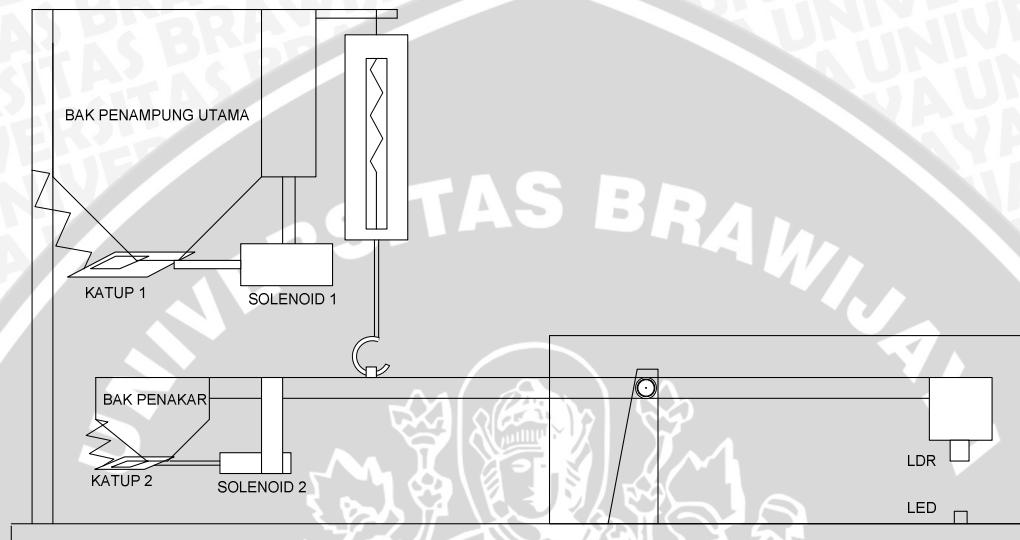
R_b dapat dihitung nilainya dengan:

$$R_b = \frac{V_{cc} - V_{ec1} - V_{be2}}{I_{b2}} = \frac{5 - 0,05 - 1,8}{32} = 98 \text{ }\Omega$$

Dalam perancangan resistor yang digunakan adalah 100 Ω

4.3.9. Rangkaian Mekanik Pemberi Pakan

Untuk desain kontruksi pemberi pakan ikan menggunakan penakar dengan sensor LDR dan untuk katup digerakan oleh solenoid, kontruksi secara detail dapat dilihat pada Gambar 4.15.



Gambar 4.15. Kontruksi Pemberi Pakan Ikan

4.4. Perancangan Perangkat Lunak

4.4.1. Format Penyimpanan Data dalam Memory EEPROM

Format dari paket penyimpanan secara rinci penyimpanan datanya adalah sebagai berikut :

Alamat memori yang digunakan adalah :

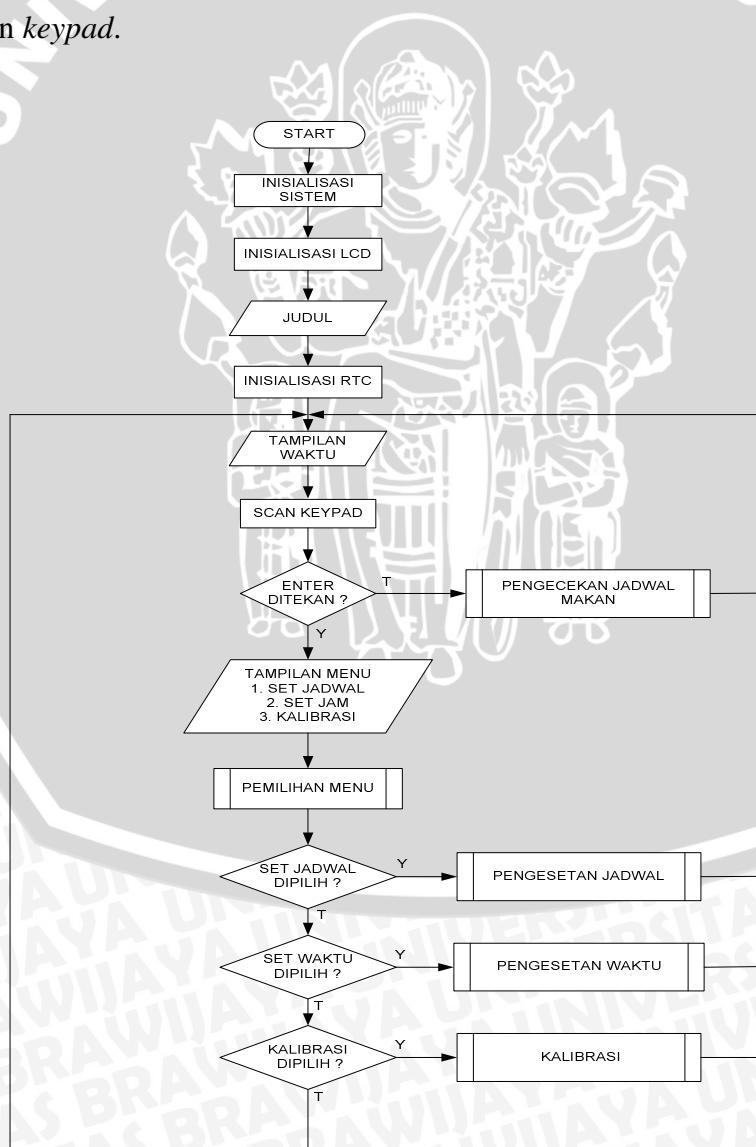
- Setting jam pemberian pakan : 000H, 004H, 008H, 0A2H, 0A6H
- Setting menit pemberian pakan : 001H, 005H, 009H, 0A3H, 0A7H
- Setting banyaknya pakan : 002H, 006H, 0A0H, 0A4H, 0A8H
- Setting pengaktifan jadwal : 003H, 007H, 0A1H, 0A5H, 0A9H

4.4.2 Perancangan Diagram Alir Program

Untuk mendukung perangkat keras yang dirancang diperlukan perangkat lunak. Perangkat lunak menggunakan bahasa pemrograman assembler mikrokontroler MCS-51. Perangkat lunak yang dirancang terdiri atas beberapa prosedur yaitu diagram alir program utama mikrokontroler dan diagram alir subrutin program. Algoritma dari perancangan perangkat lunak dapat dilihat dalam diagram alir yang terdapat pada tiap-tiap perancangan perangkat lunak.

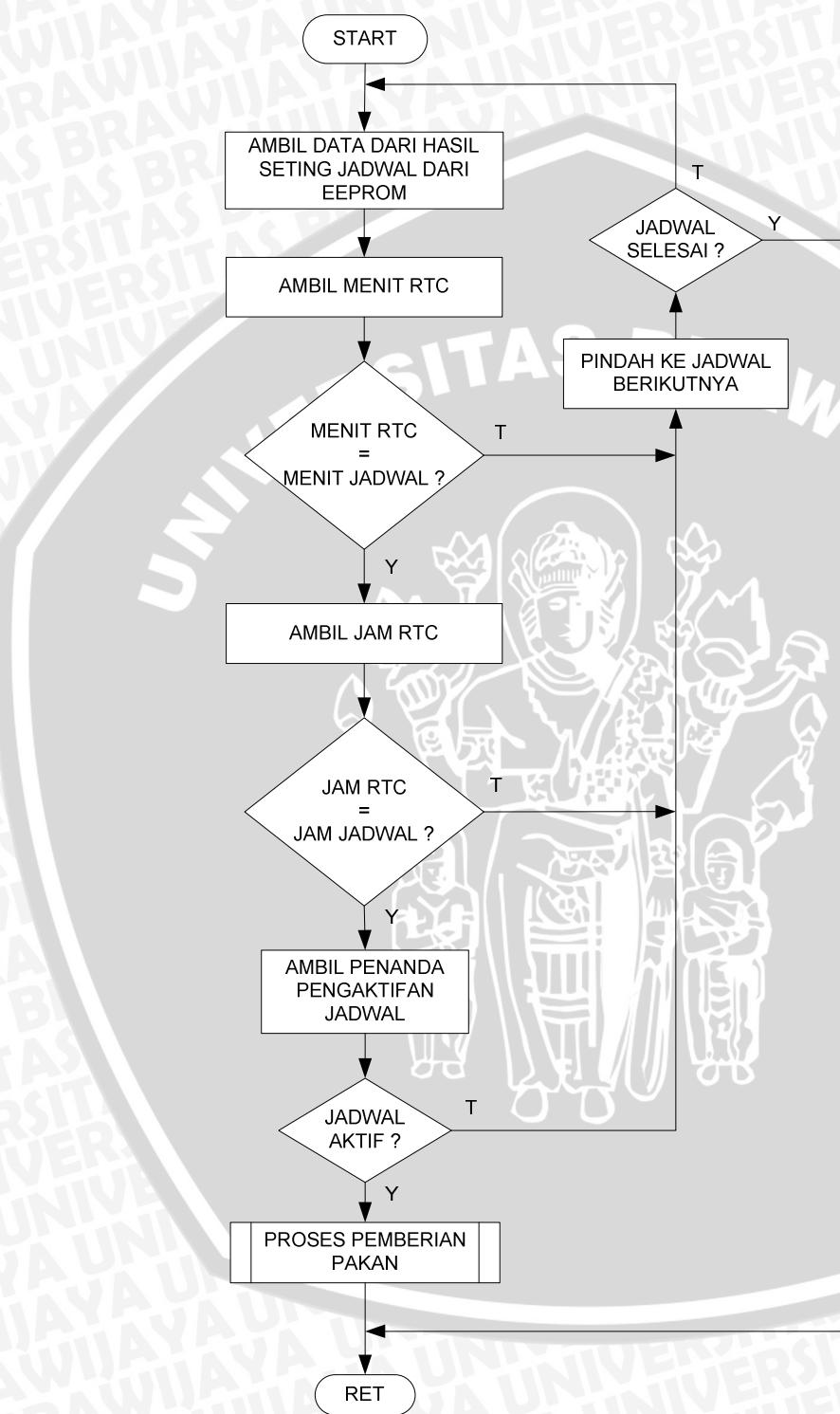
4.4.2.1 Diagram Alir Program Utama

Diagram alir program utama seperti yang ditunjukkan pada Gambar 4.16 merupakan rangkaian prosedur kerja perangkat lunak yang juga merupakan Gambaran proses dari cara kerja alat. Dalam perancangan pemakai dapat mengeset alat dengan menekan tombol enter pada *keypad* sehingga LCD akan menampilkan pilihan menu yang terdiri atas tiga pilihan yaitu : 1. set jadwal 2. set waktu 3. kalibrasi, apabila salah satu dari ketiga menu itu dapat dipilih dengan melakukan penekan tombol enter., selanjutnya setelah masuk ke salah satu menu, pemakai dapat melakukan proses pengesetan jadwal, waktu, maupun melakukan kalibrasi tergantung menu yang dipilih. Proses pengecekan jadwal makan dilakukan oleh mikrokontroler apabila tombol tidak ada penekanan *keypad*.

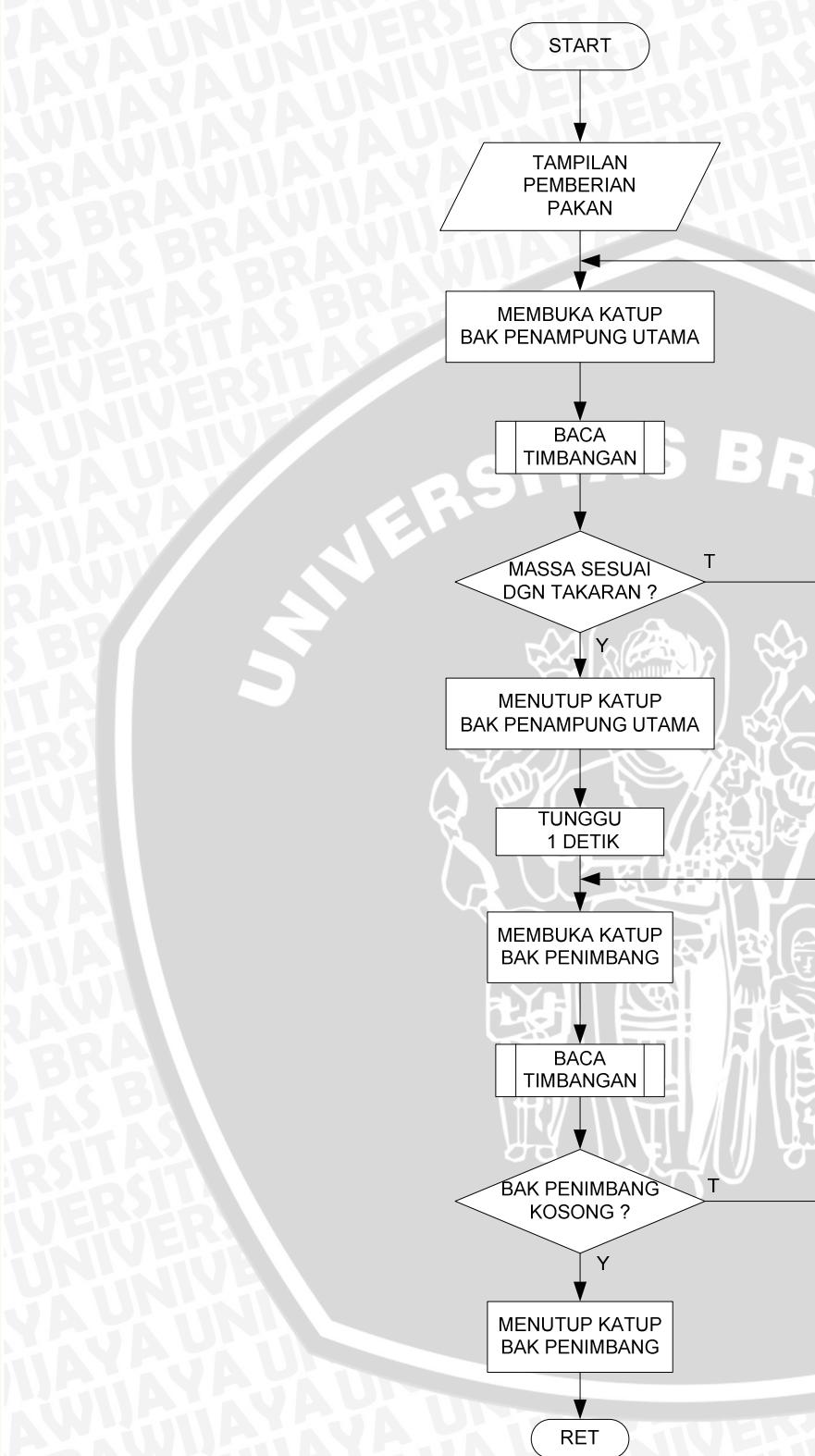


Gambar 4.16. Diagram Alir Program Utama

4.4.2.2 Diagram Alir Subrutin Pengecekan Jadwal Makan



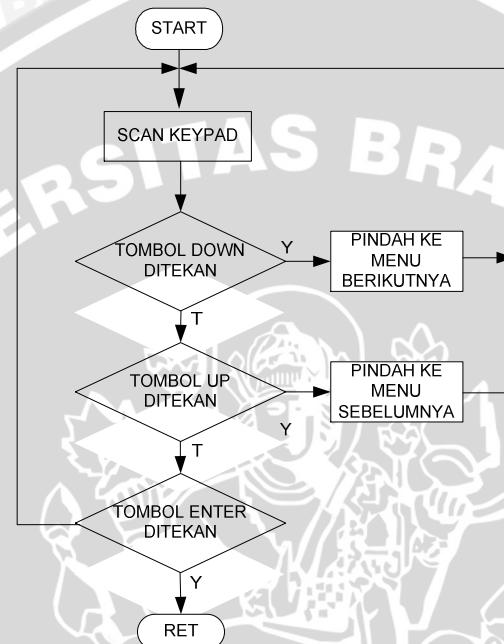
Gambar 4.17. Diagram Alir Subrutin Pengecekan Jadwal Makan



Gambar 4.18. Diagram Alir Subrutin Proses Pemberian Pakan

4.4.2.3 Diagram Alir Subrutin Pemilihan Menu

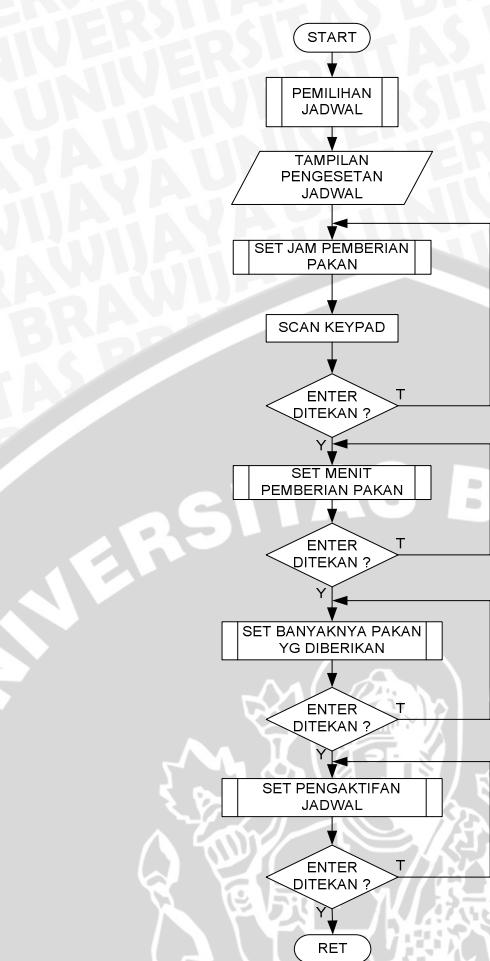
Menu ditampilkan di LCD dan dapat dipilih dengan menekan tombol up ataupun down untuk memindah kursor yang menunjuk ke menu yang akan dipilih. Untuk memilih menu cukup dengan menekan tombol enter. Diagram alir ditunjukkan pada Gambar 4.19 di bawah



Gambar 4.19. Diagram Alir Subrutin Pemilihan Menu

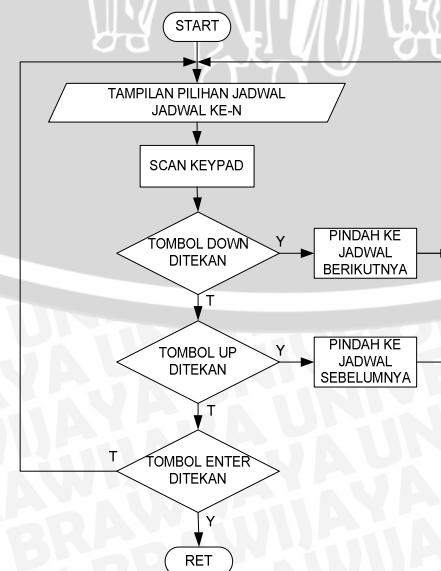
4.4.2.4 Diagram Alir Pengesetan Jadwal

Pengesetan jadwal diawali dengan memilih jadwal yang akan diseting, kemudian akan muncul tampilan yang memungkinkan pemakai mengeset jam, menit, jumlah pakan yang akan diberikan dan pengaktifan menu. Diagram alir ditunjukkan pada Gambar 4.20 di bawah, untuk proses pemilihan jadwal, set jam pemberian pakan, set menit pemberian pakan, set banyaknya pakan yang diberikan, dan set pengaktifan menu akan dijelaskan melalui diagram alir subrutin masing-masing.



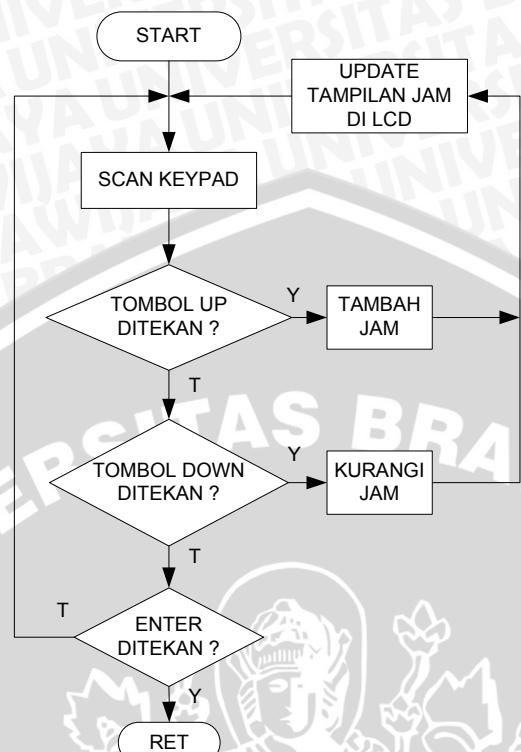
Gambar 4.20. Diagram Alir Subrutin Pengesetan Jadwal

- Diagram alir subrutin pemilihan jadwal



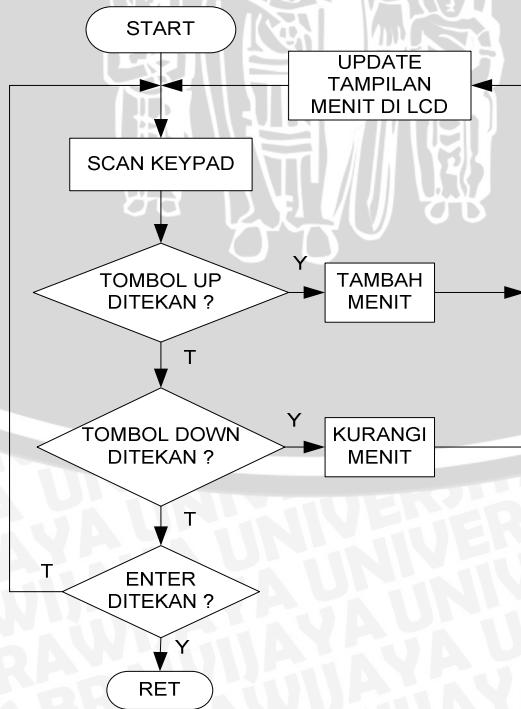
Gambar 4.21. Diagram Alir Subrutin Pemilihan Jadwal

- Diagram alir subrutin set jam pemberian pakan



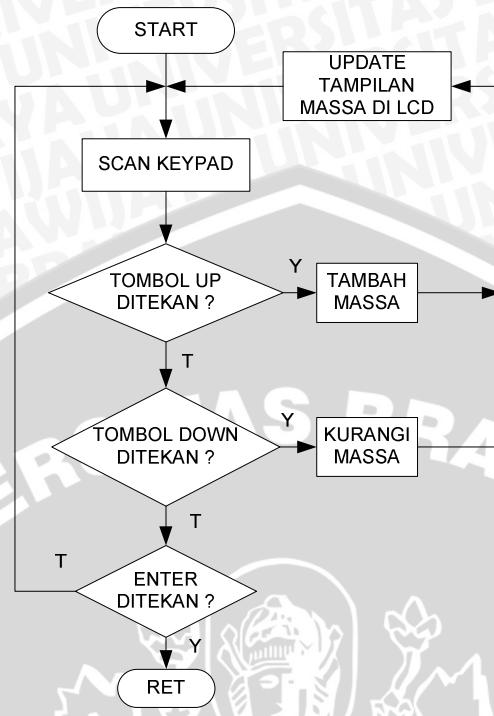
Gambar 4.22. Diagram Alir Subrutin Set Jam Pemberian Pakan

- Diagram alir subrutin set menit pemberian pakan



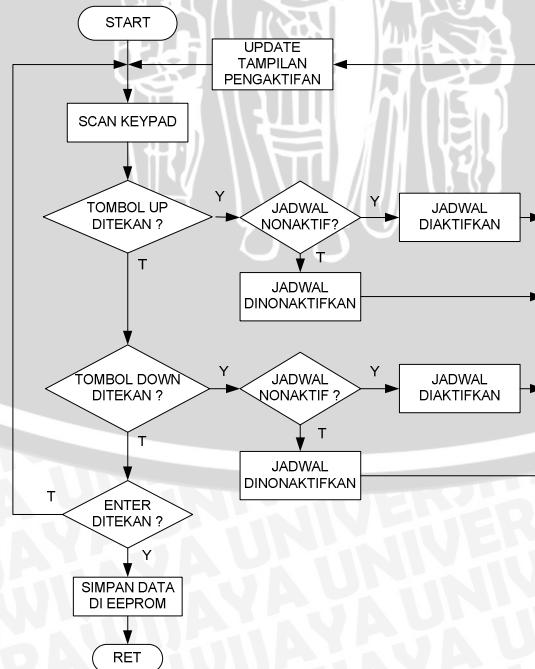
Gambar 4.23. Diagram Alir Subrutin Set Menit Pemberian Pakan

- Diagram alir subrutin banyaknya pakan yang diberikan



Gambar 4.24. Diagram Alir Subrutin Banyaknya Pakan yang Diberikan

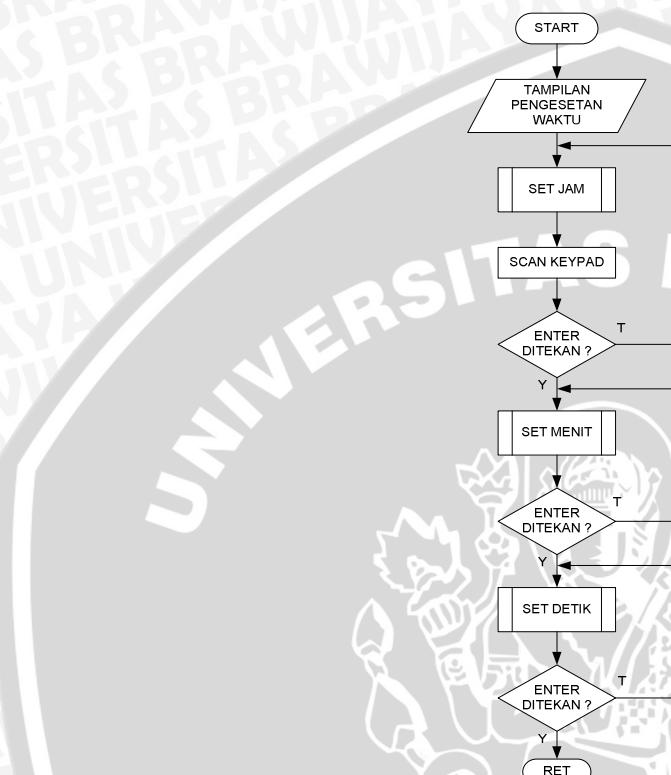
- Diagram alir subrutin set pengaktifan jadwal



Gambar 4.25. Diagram Alir Subrutin Set Pengaktifan Jadwal

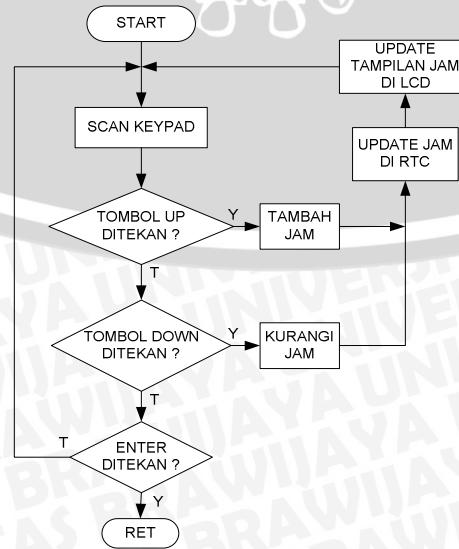
4.4.2.5 Diagram Alir Pengesetan Waktu

Proses pengesetan waktu dapat dilihat dari diagram alir yang ditunjukkan pada Gambar 4.26, Pengesetan jam, menit, detik ditunjukkan pada subrutin-subrutin tersendiri.



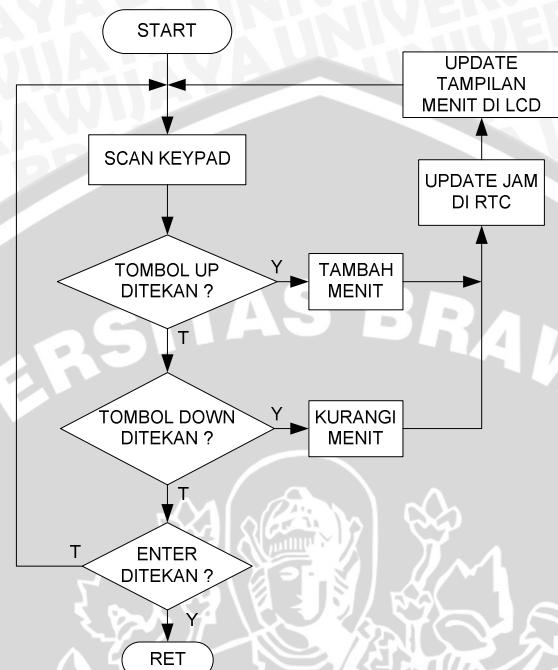
Gambar 4.26. Diagram Alir Subrutin Pengesetan Waktu

- Diagram alir subrutin set jam



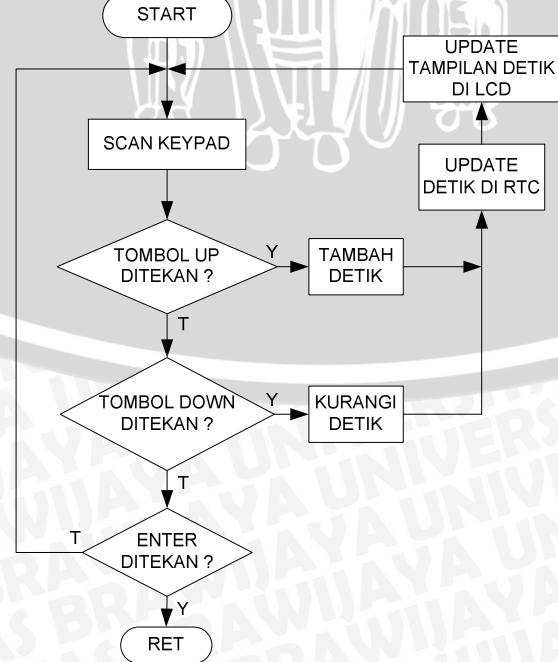
Gambar 4.27. Diagram Alir Subrutin Set Jam

- Diagram alir subrutin set menit



Gambar 4.28. Diagram Alir Subrutin Set Menit

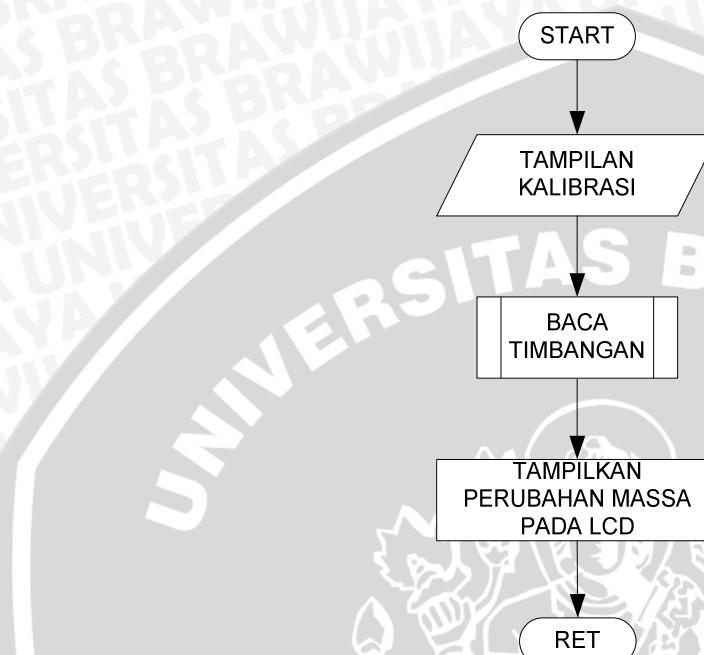
- Diagram alir subrutin set detik



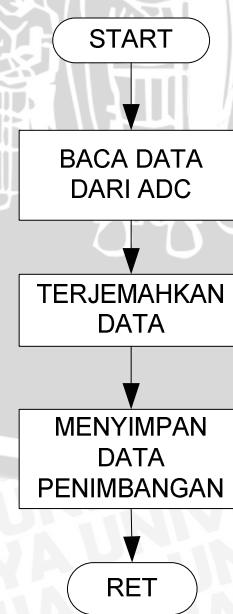
Gambar 4.29. Diagram Alir Subrutin Set Detik

4.4.2.6 Diagram Alir Proses Kalibrasi

Diagram alir proses kalibrasi dapat dilihat pada Gambar 4.30



Gambar 4.30. Diagram Alir Subrutin Kalibrasi



Gambar 4.31. Diagram Alir Subrutin Baca Timbangan

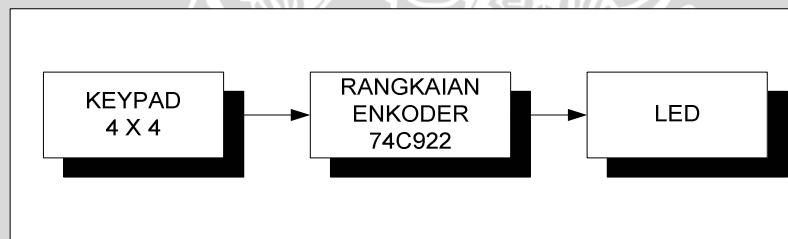
BAB V

PENGUJIAN DAN ANALISA

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 Unit Masukan Keypad

Tujuan pengujian keypad dilakukan untuk membaca tombol - tombol yang disusun secara matriks. Pengujian dilakukan dengan cara menyusun rangkaian seperti Gambar 5.1, kemudian menjalankan rangkaian dengan menghubungkan rangkaian dengan catu daya 5 volt, hasil pengujian diketahui dengan cara mengamati dan mencatat hasil keluaran data output dan *data available* (DA) yang diwakili oleh nyala LED



Gambar 5.1. Rangkaian Pengujian Unit Masukan Keypad

Hasil pengujian rangkaian keypad dapat dilihat pada Tabel 5.1. Ketika ada penekanan tombol maka lampu LED dari pin DA akan menyala dan lampu LED data output A,B,C,D akan menyala sesuai dengan kode biner data output.

Tabel 5.1. Hasil Pengujian Rangkaian Keypad

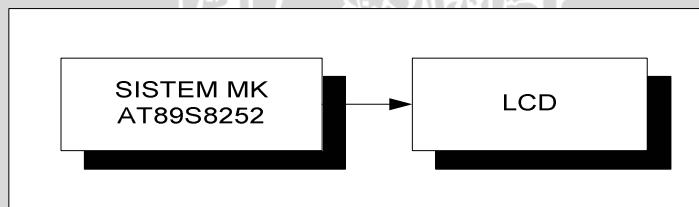
Tombol		0 Y1,X1	1 Y1,X2	2 Y1,X3	3 Y1,X4	4 Y2,X1	5 Y2,X2	6 Y2,X3	7 Y2,X4
Data Output	A	0	1	0	1	0	1	0	1
	B	0	0	1	1	0	0	1	1
	C	0	0	0	0	1	1	1	1
	D	0	0	0	0	0	0	0	0
Tombol		8	9	A	B	C	D	E	F

		Y3,X1	Y3,X2	Y3,X3	Y3,X4	Y4,X1	Y4,X2	Y4,X3	Y4,X4
Data	A	0	1	0	1	0	1	0	1
Output	B	0	0	1	1	0	0	1	1
	C	0	0	0	0	1	1	1	1
	D	1	1	1	1	1	1	1	1

Dari hasil pengujian diketahui bahwa ketika ada penekanan tombol maka lampu LED dari pin DA akan menyala dan lampu LED data dari output A, B, C, D akan menyala sesuai dengan kode biner data output. Hal ini menandakan bahwa rangkaian *keypad* dapat bekerja sesuai dengan yang diharapkan.

5.2 Pengujian Tampilan LCD

Tujuan pengujian rangkaian tampilan LCD adalah untuk mengetahui apakah rangkaian tampilan LCD dapat menampilkan data yang sesuai dengan perencanaan. Pengujian dilakukan dengan cara menyusun rangkaian pengujian seperti yang terlihat pada Gambar 5.2., kemudian menjalankan program untuk menampilkan karakter pada LCD yang telah dituliskan pada sistem mikrokontroler.



Gambar 5.2. Rangkaian Pengujian LCD

Hasil pengujian tampilan LCD dapat dilihat pada Gambar 5.3.

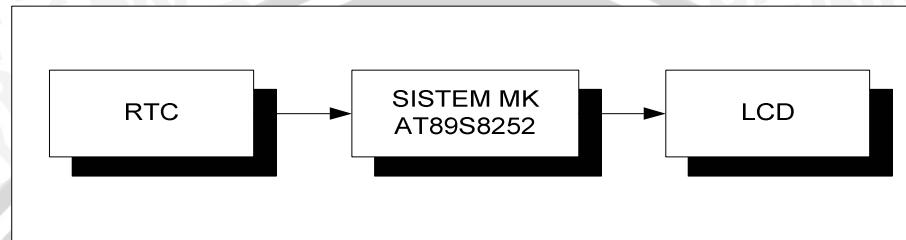


Gambar 5.3. Hasil Pengujian LCD

Dari pengujian didapatkan hasil bahwa rangkaian tampilan LCD bekerja dengan baik, yaitu dapat menampilkan karakter sesuai dengan yang direncanakan.

5.3 Pengujian RTC

Pengujian ini bertujuan untuk mengetahui keluaran rangkaian *Real Time Clock*, dimana setelah dimasukan ke sistem mikrokontroler keluarannya ditampilkan oleh LCD, kemudian keluaran tersebut dibandingkan dengan stopwatch. Rangkaian pengujian seperti yang terlihat pada Gambar 5.4.



Gambar 5.4. Rangkaian Pengujian RTC

Hasil catatan detik-detik yang bergerak dari 00:00:00 sampai dengan 00:00:60. Ketika sistem dinyalakan maka tampilan di LCD menunjukkan perubahan setiap detiknya dan dicocokkan dengan *stopwatch*.

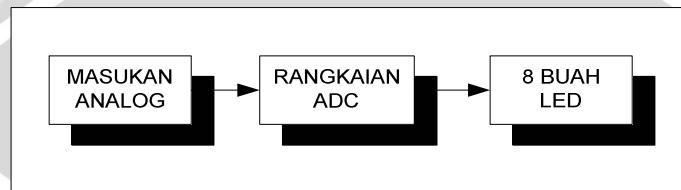
Tabel 5.2 Hasil Pengujian Sistem RTC

No.	Tampilan LCD
1	00:00:00
2	00:00:01
3	00:00:02
4	00:00:03
5	00:00:04
6	00:00:05
7	00:00:06
8	00:00:07
9	00:00:08
10	00:00:09

Dari hasil pengujian diketahui bahwa RTC bekerja dengan baik dengan dihasilkanya perubahan tiap detiknya dan hasilnya relatif sesuai dengan perubahan detik yang dihasilkan *stopwatch*. Hal tersebut menunjukkan bahwa RTC dengan baik.

5.4 Pengujian ADC 0809

Pengujian ADC bertujuan untuk mengetahui kelinieran ADC dalam mengkonversi masukan analog menjadi keluaran digital. Pengujian dilakukan dengan cara menyusun rangkaian sesuai blok diagram dalam Gambar 5.5. Keluaran ADC 8 bit dihubungkan dengan LED peraga untuk mengetahui konversi masukan analog ke bentuk keluaran digital yang berupa data biner, rangkaian ADC diberi masukan secara bertahap dari tegangan 0 V-5 V. Hasil pengujian dapat diketahui dengan mengamati nyala masing-masing LED peraga untuk tiap-tiap masukan analog yang berbeda.

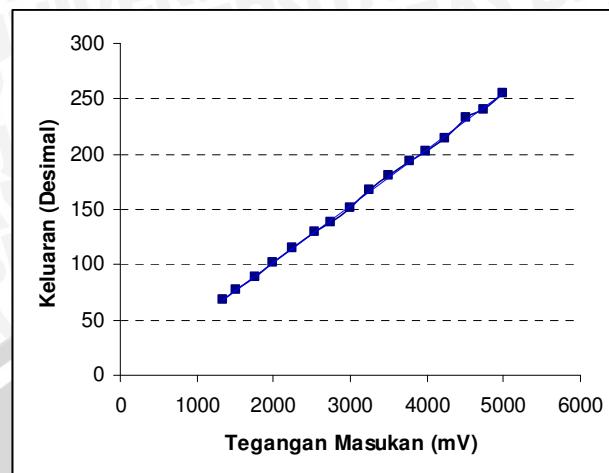


Gambar 5.5. Rangkaian Pengujian ADC

Hasil pengujian ADC terlihat pada Tabel 5.3 berikut :

Tabel 5.3 Data Pengujian ADC 0809

No.	Tegangan Masukan (mV)	Keluaran	
		Biner	Desimal
1.	1345	01000100	68
2.	1505	01001101	77
3.	1759	01011001	89
4.	1994	01100110	102
5.	2250	01110011	115
6.	2540	10000001	129
7.	2750	10001011	139
8.	3000	10011000	152
9.	3260	10100111	167
10.	3510	10110101	181
11.	3780	11000001	193
12.	3990	11001011	203
13.	4230	11010110	214
14.	4520	11101001	233
15.	4740	11110001	241
16.	4999	11111111	255

Gambar 5.6. Grafik Hasil Pengujian ADC

Hasil pengujian memperlihatkan hubungan antara tegangan masukan analog menjadi keluaran biner. Grafik pada Gambar 5.6. menunjukkan kelinearan ADC. Sehingga dapat disimpulkan bahwa ADC dapat bekerja dengan baik.

5.5 Pengujian Sensor

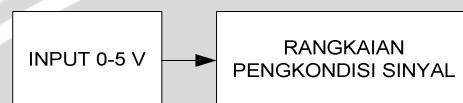
Tujuan Pengujian ini adalah untuk mengetahui hubungan keluaran dari sensor LDR dengan semakin jauhnya jarak sumber cahaya dalam hal ini berupa LED .

Tabel 5.4. Data Pengujian Sensor

No.	Jarak dengan LED (mm)	Resistansi LDR / R_{LDR} (kΩ)	Tegangan Keluaran LDR / V_{LDR} (V)
1	0	17,7	4,24
2	5	22,1	4,11
3	10	25,0	4,00
4	15	28,6	3,89
5	20	32,6	3,77
6	25	36,1	3,67
7	30	40,1	3,57
8	35	43,6	3,48
9	40	47,8	3,38
10	45	51,2	3,32
11	50	55,2	3,22

5.6 Pengujian Rangkaian Pengkondisi Sinyal

Tujuan dari pengujian rangkaian pengkondisi sinyal ini adalah untuk mengetahui apakah keluaran rangkaian pengkondisi sinyal sesuai dengan hasil perhitungan. Pengujian dilakukan dengan cara menyusun rangkaian seperti Gambar 5.7. kemudian memberikan masukan tegangan pada rangkaian pengkondisi sinyal dari 0 volt sampai 2 volt. Hasil pengujian didapat dengan cara mengukur dan mencatat tegangan keluaran rangkaian setiap perubahan masukan sebesar 0,1 volt



Gambar 5.7. Pengujian Rangkaian Pengkondisi Sinyal

Hasil pengujian dapat dilihat pada Tabel 5.5 berikut :

Tabel 5.5 Data Pengujian Rangkaian Pengkondisi Sinyal

No.	Tegangan Masukan V _{LDR} (V)	V _{out} pengukuran (V)	V _{out} perhitungan (V)	Error	% Error
1	4,24	0,06	0,07	0,01	13,8%
2	4,11	0,56	0,57	0,01	1,7%
3	4	0,98	1,00	0,02	2,0%
4	3,89	1,41	1,42	0,01	0,7%
5	3,77	1,86	1,88	0,02	1,1%
6	3,67	2,25	2,26	0,01	0,4%
7	3,57	2,63	2,65	0,02	0,8%
8	3,48	2,99	3,00	0,01	0,3%
9	3,38	3,36	3,38	0,02	0,6%
10	3,32	3,59	3,61	0,02	0,6%
11	3,22	3,99	4,00	0,01	0,3%

$$\text{Error rata-rata} = \frac{0,16}{11} = 0,014 \text{ volt}$$

$$\% \text{ error rata-rata} = \frac{22,3\%}{11} = 2,03\%$$

Dari hasil pengujian terlihat bahwa rangkaian pengkondisi sinyal dapat bekerja dengan baik dengan persentasi error yang relatif kecil.

5.7 Pengujian Error Timbangan

Tujuan dari pengujian ini adalah untuk mengetahui besarnya kesalahan timbangan.

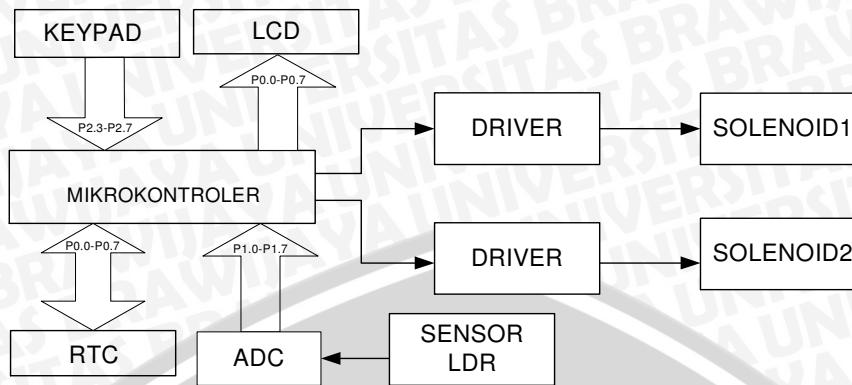
Tabel 5.6 Data Pengujian Timbangan

No.	Berat yang diuji (gram)	Berat terbaca (gram)	Error (gram)	% Error
1	1	1	0	0,0%
2	2	2	0	0,0%
3	3	3	0	0,0%
4	4	4	0	0,0%
5	5	5	0	0,0%
6	6	6	0	0,0%
7	7	6	1	14,3%
8	8	7	1	12,5%
9	9	8	1	11,1%
10	10	9	1	10,0%
11	11	10	1	9,1%
12	12	11	1	8,3%
13	13	12	1	7,7%
14	14	13	1	7,1%
15	15	14	1	6,7%
16	16	15	1	6,3%
17	17	16	1	5,9%
18	18	17	1	5,6%
19	19	17	2	10,5%
20	20	18	2	10,0%

Tabel 5.6 menunjukkan bahwa ada sedikit kesalahan pada timbangan, akan tetapi untuk aplikasi penimbangan pakan ikan yang tidak perlu tingkat ketepatan yang tinggi, maka disimpulkan bahwa timbangan dapat digunakan untuk penimbangan pakan.

5.8 Pengujian Keseluruhan Sistem

Tujuan dari pengujian ini adalah untuk mengetahui kinerja sistem secara keseluruhan , apakah telah berhasil sesuai perencanaan. Pengujian dilakukan dengan cara menjalankan sistem secara keseluruhan ditunjukkan dalam Gambar 5.8.

**Gambar 5.8.** Pengujian Sistem Keseluruhan

Hasil pengujian sistem secara keseluruhan dapat dilihat pada Tabel 5.7

Tabel 5.7. Pengujian Sistem Secara Keseluruhan

No.	Pengujian	Hasil
1	Pengesetan Menu	Pengesetan menu meliputi : set jam dan menit pemberian pakan, set massa pakan yang diberikan serta pengaktifan jadwal dapat berjalan dengan baik
2	Pengesetan Jam	Pengesetan jam alat meliputi set jam, set menit, dan set detik dapat dilakukan dengan baik
3	Kalibrasi	Sistem timbangan dapat bekerja dengan baik yaitu timbangan dapat menimbang setiap 1 gram massa yang ditimbang
4	Jam Sistem	Jam sistem dapat menunjukkan waktu yang tepat
5	Proses pemberian pakan	Pada waktu pemberian pakan, solenoid 1 membuka katup1 sehingga pakan jatuh ke bak penakar, pada saat massa yang pakan yang ada di bak penakar sesuai dengan setting awal, maka katup 1 menutup, 1 detik kemudian solenoid 2 membuka katup 2 sehingga pakan yang ada pada bak penakar dikeluarkan, pada saat pakan pada bak penakar habis, katup 2 menutup dan sistem siap untuk melakukan proses pemberian pakan selanjutnya.

Dari hasil pengujian yang ditunjukkan pada Tabel 5.7. dapat disimpulkan bahwa

keseluruhan sistem dapat bekerja dengan baik.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari perancangan dan pembuatan sistem serta dilanjutkan dengan proses pengujian, maka diperoleh kesimpulan sebagai berikut :

1. Secara keseluruhan, sistem bekerja dengan baik sesuai dengan yang direncanakan
2. Tampilan dari *display LCD* bekerja dengan baik.
3. Timbangan pakan mampu menimbang setiap 1 gram pakan.
4. Jumlah pemberian pakan maksimum setiap pemberian pakan sebesar 20 gram.
5. Hubungan antara *keypad*, AT89S8252, dan LCD telah bekerja dengan baik.
6. Hubungan Mikrokontroler, Sensor LDR, ADC0809 bekerja dengan baik.
7. Hubungan Mikrokontroler, Solenoid, dan bagian mekanis telah bekerja dengan baik.

6.2. Saran

Saran-saran untuk penyempurnaan kinerja alat dan pengembangan lebih lanjut adalah sebagai berikut :

1. Agar dikembangkan sistem pemberian pakan yang bisa mengetahui kondisi ikan yang ada di akuarium dan kondisi air di akuarium misalnya apabila ikan dalam kondisi kenyang atau akuarium dalam kondisi kotor karena pakan yang berlebih maka pemberian pakan dihentikan atau dikurangi, dan apabila ikan dalam kondisi lapar maka pemberian pakan dilakukan .
2. Agar dikembangkan sistem pemberi pakan yang tidak hanya bisa memberikan satu jenis pakan tertentu, akan tetapi bisa memberikan berbagai jenis pakan secara bervariasi sesuai kebutuhan ikan.
3. Agar dikembangkan mekanik pemberi pakan yang tidak hanya bisa memberikan pakan berbentuk butiran kecil, tapi juga bisa memberikan pakan dalam bentuk *gel* ataupun bentuk lainnya.

DAFTAR PUSTAKA

- ATMEL. 1997. *8-Bit Microcontroller with 8K bytes Flash AT89S8252*. ATMEL.
<http://www.atmel.com>.
- Dallas Semiconductor. 2002. *DS12887 Real-Time Clock*. Dallas Semiconductor.
<http://www.maxim-ic.com>.
- Effendy, Hersanto.1990. *Memelihara Maskoki dalam Akuarium*. Yogyakarta: Kanisius.
- Fairchild Semiconductor. 2001. *MM74C922 16-Key General Encoder*. Fairchild Semiconductor. <http://www.fairchildsemi.com>.
- Malvino, Albert Paul. 1999. *Prinsip-Prinsip Elektronika*. Jakarta: Erlangga.
- Nalwan, Paulus Andi. 2004. *Panduan Praktis Penggunaan dan Antarmuka Modul LCD M1632*. Jakarta: Elex Media Komputindo.
- National Semiconductor.1995. *National Data Acquisition Databook*. California, USA: National Semiconductor.
- RS Components. 1997. *Light Dependent Resistors*. RS Components
<http://www.rswww.com.hk>.
- Seiko Instruments. 1987. *Liquid Crystal Display Module M1632 User Manual*. Seiko Instrument. <http://www.sii.co.jp>



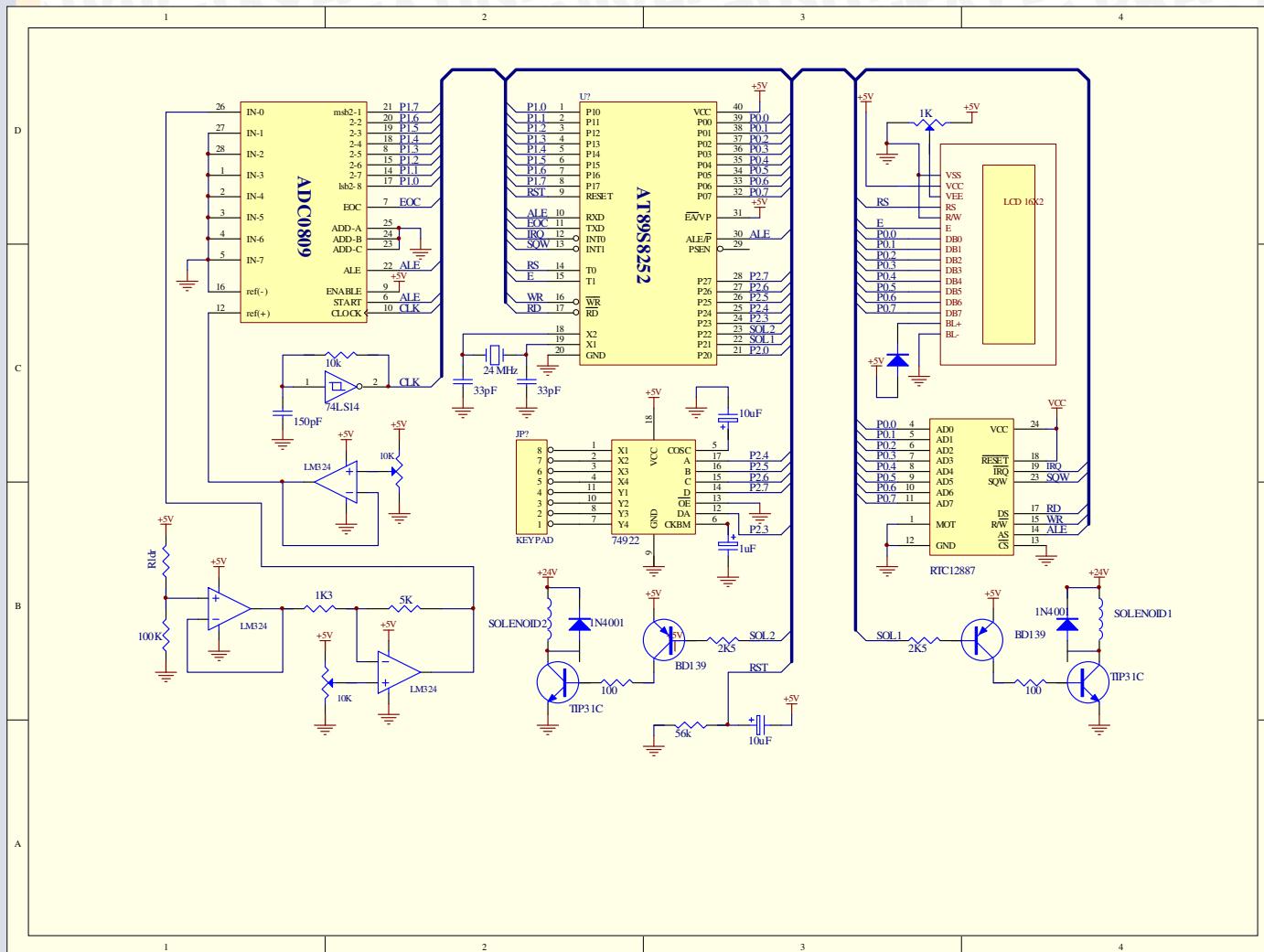
LAMPIRAN

- FOTO ALAT
- SKEMATIK RANGKAIAN
- LISTING PROGRAM
- DATASHEET

FOTO ALAT



SKEMATIK RANGKAIAN ALAT



; LISTING PROGRAM
; ALAT PEMBERI PAKAN UNTUK IKAN SECARA OTOMATIS
; BERBASISKAN MIKROKONTROLER ATS8252

```
; ****
; DEFINISI KONSTANTA
; ****
RTC_JAM EQU 04H
RTC_MENIT EQU 02H
RTC_DETIK EQU 00H
RTC_ALARM_JAM EQU 05H
RTC_ALARM_MENIT EQU 03H
RTC_ALARM_DETIK EQU 01H

; ****
; DEFINISI VARIABEL
; ****
SATUAN EQU 21H
PULUHAN EQU 22H
RATUSAN EQU 23H
ADC_DAT EQU 24H
JAM EQU 25H
MENIT EQU 26H
DETIK EQU 27H
BERAT EQU 28H
ENABLE EQU 29H
PILIH EQU 2AH
TOMBOL EQU 2BH
POINTER EQU 2CH
BERAT_TIMBANG EQU 2DH

; ****
; DEFINISI BIT
; ****
DONE BIT 20H.0
ON_SET BIT 20H.1
ON_CEK BIT 20H.2
ON_SET_JAM BIT 20H.3
ON_INTERUPSI BIT 20H.4

; ****
; DEFINISI PORT
; ****
PINTU_TIMBANG BIT P2.2
PINTU_TANDON BIT P2.1
STROBE BIT P2.3
ALES BIT P3.0
EOC BIT P3.1
RS BIT P3.4
E BIT P3.5

ORG 00H
SJMP BEGIN

ORG 13H
AJMP H_SQWE
```

BEGIN:

```
MOV SP, #40H
ACALL INIT
SETB ON_SET
ACALL INIT_LCD
MOV DPTR, #JUDUL ;Cetak judul ke LCD
MOV P0, #80H
ACALL PRINT
INC DPTR
MOV P0, #0C0H
ACALL PRINT
ACALL ONE_SEC_DELAY
INC DPTR
MOV P0, #80H
ACALL PRINT
INC DPTR
MOV P0, #0C0H
ACALL PRINT
ACALL ONE_SEC_DELAY
MOV DPTR, #CLEAR
MOV P0, #80H
ACALL PRINT
MOV DPTR, #CLEAR
MOV P0, #0C0H
ACALL PRINT
ACALL INIT_RTC
CLR ON_SET
```

REPEAT:

```
JNB STROBE, CEK_EAT ;Cek penekanan tombol
JB STROBE, $
ACALL BACA_KEYPAD
MOV A, TOMBOL
CJNE A, #0BH, REPEAT
ACALL PILIH_MENU
MOV A, PILIH
CJNE A, #1, CEK_DUA
ACALL SET_JADWAL
```

CEK_EAT:

```
CLR ON_SET
CLR ON_SET_JAM
JNB ON_CEK, REPEAT
CLR ON_CEK
ACALL PERIKSA_JADWAL
SJMP CEK_EAT
```

CEK_DUA:

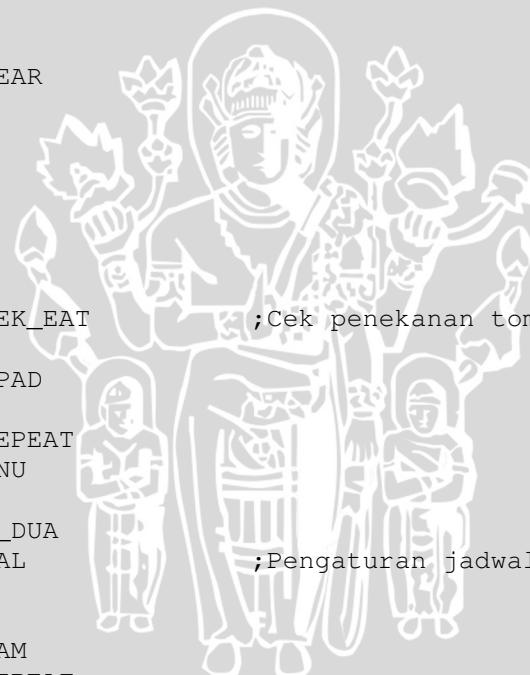
```
CJNE A, #2, KALIBRASI
ACALL SET_JAM
SJMP CEK_EAT
```

KALIBRASI:

```
MOV DPTR, #CALIBRATE
MOV P0, #80H
ACALL PRINT
SETB ON_SET_JAM
```

REP_KALIBRASI:

```
ACALL BACA_TIMBANGAN
MOV A, BERAT_TIMBANG
MOV B, #10
DIV AB
ADD A, #30H
```



```
MOV    P0, #8BH
ACALL CONTROL
MOV    P0, A
ACALL WRITE
MOV    A, B
ADD    A, #30H
MOV    P0, #8CH
ACALL CONTROL
MOV    P0, A
ACALL WRITE
JNB    STROBE, REP_KALIBRASI
JB     STROBE, $
SJMP   CEK_EAT
```

H_SQWE :

```
PUSH   ACC
PUSH   PSW
PUSH   00
PUSH   01
SETB   ON_INTERUPSI
JB     ON_SET, SKIP_JAM
JB     ON_SET_JAM, NO_CLEAR
ACALL TAMPIL_JAM
SJMP   SKIP_JAM
```

NO_CLEAR :

```
ACALL TAMPIL_SET_JAM
```

SKIP_JAM :

```
MOV    R0, #RTC_DETIK
MOVX  A, @R0
CJNE  A, #00H, BELUM_GANTI_MENIT
JB    DONE, METU
SETB  DONE
SETB  ON_CEK
SJMP  METU
```

BELUM_GANTI_MENIT :

```
CLR   DONE
```

METU :

```
POP   01
POP   00
POP   PSW
POP   ACC
RETI
```

```
; ****
; SUBRUTIN PENGECEKAN JADWAL MAKAN
; ****
```

PERIKSA_JADWAL :

```
SETB  ON_SET_JAM
MOV   POINTER, #0
MOV   R7, #5
```

RECEK:

```
ACALL GET_Seting
MOV   R0, #RTC_MENIT
MOVX A, @R0
CJNE A, MENIT, TIDAK_SAMA
MOV   R0, #RTC_JAM
MOVX A, @R0
CJNE A, JAM, TIDAK_SAMA
```

```
MOV A,ENABLE
CJNE A,#0,JADWAL_AKTIF
RET
JADWAL_AKTIF:
ACALL BERI_MAKAN
RET
TIDAK_SAMA:
INC POINTER
DJNZ R7,RECEK
RET

;*****
; SUBRUTIN MENIMBANG PAKAN IKAN
;*****
BERI_MAKAN:
SETB ON_SET
MOV DPTR,#JUMLAH
MOV P0,#80H
ACALL PRINT
CLR PINTU_TANDON ;Buka pintu stok pakan
REP_MAKAN:
ACALL BACA_TIMBANGAN
MOV A,BERAT_TIMBANG
MOV P0,#8AH
ACALL CONTROL
ACALL PUL_SAT
MOV A,BERAT_TIMBANG
CJNE A,BERAT,REP_MAKAN ;Tunggu sampai berat = seting
SETB PINTU_TANDON ;Tutup pintu stok pakan
ACALL ONE_SEC_DELAY
CLR PINTU_TIMBANG ;Buka pintu timbangan
BELUM_HABIS:
ACALL BACA_TIMBANGAN
MOV A,BERAT_TIMBANG
CJNE A,#0,BELUM_HABIS
SETB PINTU_TIMBANG
CLR ON_SET
RET

;*****
; SUBRUTIN PILIH MENU
;*****
PILIH_MENU:
SETB ON_SET
MOV PILIH,#1 ;Pilih menu 1
REF_PILIH:
ACALL SHOW_PILIH
JNB STROBE,$
JB STROBE,$
ACALL BACA_KEYPAD
MOV A,TOMBOL
CJNE A,#ODH,CEK_PANAH_DOWN
DEC PILIH
MOV A,PILIH
CJNE A,#0,REF_PILIH
MOV PILIH,#1
SJMP REF_PILIH
CEK_PANAH_DOWN:
CJNE A,#OCH,CEK_PIL_ENTER
```

```
INC    PILIH
MOV    A, PILIH
CJNE   A, #4, REF_PILIH
MOV    PILIH, #3
SJMP   REF_PILIH

CEK_PIL_ENTER:
CJNE   A, #0BH, REF_PILIH
CLR    ON_SET
RET
```

```
; ****
; SUBRUTIN TAMPIL PILIHAN
; ****
```

```
SHOW_PILIH:
MOV    DPTR, #PIL_MENU
CLR    C
MOV    A, PILIH
DEC    A
MOV    B, #17
MUL    AB
ADD    A, DPL
MOV    DPL, A
CLR    A
ADDC   A, DPH
MOV    DPH, A
MOV    P0, #80H
ACALL  PRINT
INC    DPTR
MOV    P0, #0C0H
ACALL  PRINT
MOV    P0, #80H
ACALL  CONTROL
MOV    P0, #3
ACALL  WRITE
RET
```

```
; ****
; SUBRUTIN SET JAM ALAT
; ****
```

```
SET_JAM:
SETB   ON_SET
SETB   ON_SET_JAM
MOV    DPTR, #CLEAR
MOV    P0, #80H
ACALL  PRINT
MOV    P0, #87H
ACALL  CONTROL
MOV    P0, #2
ACALL  WRITE
CLR    ON_SET
```

```
ULANG_SET_JAM:
JNB    STROBE, $
JB     STROBE, $
ACALL  BACA_KEYPAD
CJNE   A, #0DH, CEK_SET_JAM_DOWN
MOV    R0, #RTC_JAM
MOVX   A, @R0
MOV    JAM, A
INC    JAM
```

; cek tombol up ditekan



```

MOV    A, JAM
CJNE  A, #24, OKE_SET_JAM
MOV    JAM, #0
SJMP   OKE_SET_JAM

CEK_SET_JAM_DOWN:
CJNE  A, #0CH, CEK_SET_JAM_ENTER           ;cek tombol down ditekan
MOV    R0, #RTC_JAM
MOVX  A, @R0
MOV    JAM, A
DEC   JAM
MOV    A, JAM
CJNE  A, #255, OKE_SET_JAM
MOV    JAM, #23

OKE_SET_JAM:
MOV    P0, #0C7H
ACALL CONTROL
MOV    R0, #RTC_JAM
MOV    A, JAM
MOVX  @R0, A
ACALL PUL_SAT
SJMP   ULANG_SET_JAM

CEK_SET_JAM_ENTER:
CJNE  A, #0BH, ULANG_SET_JAM               ;cek tombol enter ditekan

```

```

;*****
; RUTIN SET MENIT
;*****
MOV    P0, #87H
ACALL CONTROL
MOV    P0, #' '
ACALL WRITE
MOV    P0, #8AH
ACALL CONTROL
MOV    P0, #2
ACALL WRITE

```

```

ULANG_SET_MENIT:
JNB   STROBE, $
JB    STROBE, $
ACALL BACA_KEYPAD
CJNE  A, #0DH, CEK_SET_MENIT_DOWN         ;cek tombol up ditekan
MOV    R0, #RTC_MENIT
MOVX  A, @R0
MOV    MENIT, A
INC   MENIT
MOV    A, MENIT
CJNE  A, #60, OKE_SET_MENIT
MOV    MENIT, #0
SJMP   OKE_SET_MENIT

```

```

CEK_SET_MENIT_DOWN:
CJNE  A, #0CH, CEK_SET_MENIT_ENTER        ;cek tombol down ditekan
MOV    R0, #RTC_MENIT
MOVX  A, @R0
DEC   MENIT
MOV    A, MENIT
CJNE  A, #255, OKE_SET_MENIT
MOV    MENIT, #59

```

```

OKE_SET_MENIT:
MOV    P0, #0CAH
ACALL CONTROL
MOV    R0, #RTC_MENIT

```



```
MOV    A,MENIT
MOVX   @R0,A
ACALL  PUL_SAT
SJMP   ULANG_SET_MENIT
CEK_SET_MENIT_ENTER:
        CJNE  A,#0BH,ULANG_SET_MENIT

;*****
; RUTIN SET DETIK
;*****
MOV    P0,#8AH
ACALL CONTROL
MOV    P0,' '
ACALL WRITE
MOV    P0,#8DH
ACALL CONTROL
MOV    P0,#2
ACALL WRITE
ULANG_SET_DETIK:
JNB    STROBE,$
JB     STROBE,$
ACALL BACA_KEYPAD
CJNE  A,#0DH,CEK_SET_DETIK_DOWN
MOV    R0,#RTC_DETIK
MOVX   A,@R0
MOV    DETIK,A
INC    DETIK
MOV    A,DETIK
CJNE  A,#60,OKE_SET_DETIK
MOV    DETIK,#0
SJMP   OKE_SET_DETIK
CEK_SET_DETIK_DOWN:
CJNE  A,#0CH,CEK_SET_DETIK_ENTER
MOV    R0,#RTC_DETIK
MOVX   A,@R0
DEC    DETIK
MOV    A,DETIK
CJNE  A,#255,OKE_SET_DETIK
MOV    DETIK,#59
OKE_SET_DETIK:
MOV    P0,#0CDH
ACALL CONTROL
MOV    R0,#RTC_DETIK
MOV    A,DETIK
MOVX   @R0,A
ACALL PUL_SAT
SJMP   ULANG_SET_DETIK
CEK_SET_DETIK_ENTER:
CJNE  A,#0BH,ULANG_SET_DETIK
CLR    ON_SET_JAM
RET

;*****
; SUBRUTIN SET JADWAL
;*****
SET_JADWAL:
SETB   ON_SET
MOV    DPTR,#PIL_JADWAL
MOV    P0,#80H
```

;cek tombol up ditekan

;cek tombol down ditekan

```
ACALL PRINT
INC DPTR
MOV P0, #0C0H
ACALL PRINT
MOV POINTER, #0
REP_SET:
    ACALL TAMPIL_SET_JADWAL
    JNB STROBE, $
    JB STROBE, $
    ACALL BACA_KEYPAD
    MOV A, TOMBOL
    CJNE A, #0DH, CEK_DOWN ; cek tombol up ditekan
    DEC POINTER
    MOV A, POINTER
    CJNE A, #255, REP_SET
    MOV POINTER, #0
    SJMP REP_SET
CEK_DOWN:
    CJNE A, #0CH, CEK_ENTER ; cek tombol down ditekan
    INC POINTER
    MOV A, POINTER
    CJNE A, #5, REP_SET
    MOV POINTER, #4
    SJMP REP_SET
CEK_ENTER:
    CJNE A, #0BH, REP_SET
    ACALL GET_SETING
    MOV DPTR, #SET_JADWAL_TEXT
    MOV P0, #80H
    ACALL PRINT
    INC DPTR
    MOV P0, #0C0H
    ACALL PRINT
    ACALL SHOW_JADWAL

; *****
; RUTIN SET JAM
; *****
    MOV P0, #80H
    ACALL CONTROL
    MOV P0, #2
    ACALL WRITE
ULANG_JAM:
    JNB STROBE, $
    JB STROBE, $
    ACALL BACA_KEYPAD
    CJNE A, #0DH, CEK_JAM_DOWN ; cek tombol up ditekan
    INC JAM
    MOV A, JAM
    CJNE A, #24, OKE_JAM
    MOV JAM, #0
    SJMP OKE_JAM
CEK_JAM_DOWN:
    CJNE A, #0CH, CEK_JAM_ENTER ; cek tombol down ditekan
    DEC JAM
    MOV A, JAM
    CJNE A, #255, OKE_JAM
    MOV JAM, #23
OKE_JAM:
    MOV P0, #0C0H
    ACALL CONTROL
```

```
MOV A, JAM
ACALL PUL_SAT
SJMP ULANG_JAM
CEK_JAM_ENTER:
CJNE A, #0BH, ULANG_JAM

;*****
; RUTIN SET MENIT
;*****
MOV P0, #80H
ACALL CONTROL
MOV P0, #' '
ACALL WRITE
MOV P0, #83H
ACALL CONTROL
MOV P0, #2
ACALL WRITE
ULANG_MENIT:
JNB STROBE, $
JB STROBE, $
ACALL BACA_KEYPAD
CJNE A, #0DH, CEK_MENIT_DOWN ;cek tombol up ditekan
INC MENIT
MOV A, MENIT
CJNE A, #60, OKE_MENIT
MOV MENIT, #0
SJMP OKE_MENIT
CEK_MENIT_DOWN:
CJNE A, #0CH, CEK_MENIT_ENTER ;cek tombol down ditekan
DEC MENIT
MOV A, MENIT
CJNE A, #255, OKE_MENIT
MOV MENIT, #59
OKE_MENIT:
MOV P0, #0C3H
ACALL CONTROL
MOV A, MENIT
ACALL PUL_SAT
SJMP ULANG_MENIT
CEK_MENIT_ENTER:
CJNE A, #0BH, ULANG_MENIT

;*****
; RUTIN SET BERAT
;*****
MOV P0, #83H
ACALL CONTROL
MOV P0, #' '
ACALL WRITE
MOV P0, #88H
ACALL CONTROL
MOV P0, #2
ACALL WRITE
ULANG_BERAT:
JNB STROBE, $
JB STROBE, $
ACALL BACA_KEYPAD
CJNE A, #0DH, CEK_BERAT_DOWN ;cek tombol up ditekan
INC BERAT
MOV A, BERAT
CJNE A, #21, OKE_BERAT
```

```
MOV    BERAT, #1
SJMP   OKE_BERAT
CEK_BERAT_DOWN:
    CJNE  A, #0CH, CEK_BERAT_ENTER ; cek tombol down ditekan
    DEC   BERAT
    MOV   A, BERAT
    CJNE  A, #0, OKE_BERAT
    MOV   BERAT, #20
OKE_BERAT:
    MOV   P0, #0C8H
    ACALL CONTROL
    MOV   A, BERAT
    ACALL PUL_SAT
    SJMP   ULANG_BERAT
CEK_BERAT_ENTER:
    CJNE  A, #0BH, ULANG_BERAT
;*****
; RUTIN SET PENGAKTIFAN
;*****
MOV   P0, #88H
ACALL CONTROL
MOV   P0, #' '
ACALL WRITE
MOV   P0, #8EH
ACALL CONTROL
MOV   P0, #2
ACALL WRITE
ULANG_AKTIF:
    JNB   STROBE, $
    JB    STROBE, $
    ACALL BACA_KEYPAD
    CJNE  A, #0DH, CEK_AKTIF_DOWN ; cek tombol up ditekan
    MOV   A, ENABLE
    CJNE  A, #0, AKTIF_UP
    MOV   ENABLE, #0FFH
    SJMP   OKE_AKTIF
AKTIF_UP:
    MOV   ENABLE, #00H
    SJMP   OKE_AKTIF
CEK_AKTIF_DOWN:
    CJNE  A, #0CH, CEK_AKTIF_ENTER ; cek tombol down ditekan
    MOV   A, ENABLE
    CJNE  A, #0, AKTIF_DOWN
    MOV   ENABLE, #0FFH
    SJMP   OKE_AKTIF
AKTIF_DOWN:
    MOV   ENABLE, #00H
    SJMP   OKE_AKTIF
OKE_AKTIF:
    MOV   A, ENABLE
    ACALL BUTTON
    SJMP   ULANG_AKTIF
CEK_AKTIF_ENTER:
    CJNE  A, #0BH, ULANG_AKTIF
    ACALL ONE_SEC_DELAY
    ACALL SAVE_SETING
    RET
```

```
;*****  
; SUBRUTIN SIMPAN SETING KE EEPROM  
;*****  
SAVE_SETING:  
    CLR C  
    MOV DPTR, #0  
    MOV A, POINTER  
    MOV B, #4  
    MUL AB  
    ADD A, DPL  
    MOV DPL, A  
    CLR A  
    ADDC A, DPH  
    MOV DPH, A  
    MOV A, JAM  
    ACALL TULIS_EEPROM  
    INC DPTR  
    MOV A, MENIT  
    ACALL TULIS_EEPROM  
    INC DPTR  
    MOV A, BERAT  
    ACALL TULIS_EEPROM  
    INC DPTR  
    MOV A, ENABLE  
    ACALL TULIS_EEPROM  
    RET  
  
;*****  
; SUBRUTIN BACA SETING DARI EEPROM  
;*****  
GET_SETING:  
    CLR C  
    MOV DPTR, #0  
    MOV A, POINTER  
    MOV B, #4  
    MUL AB  
    ADD A, DPL  
    MOV DPL, A  
    CLR A  
    ADDC A, DPH  
    MOV DPH, A  
    ACALL BACA_EEPROM  
    MOV JAM, A  
    INC DPTR  
    ACALL BACA_EEPROM  
    MOV MENIT, A  
    INC DPTR  
    ACALL BACA_EEPROM  
    MOV BERAT, A  
    INC DPTR  
    ACALL BACA_EEPROM  
    MOV ENABLE, A  
    RET  
  
;*****  
; SUBRUTIN DISPLAY JADWAL MAKAN  
;*****  
SHOW_JADWAL:
```

```
MOV P0, #0C0H
ACALL CONTROL
MOV A, JAM
ACALL PUL_SAT
MOV P0, #0C3H
ACALL CONTROL
MOV A, MENIT
ACALL PUL_SAT
MOV P0, #0C8H
ACALL CONTROL
MOV A, BERAT
ACALL PUL_SAT
MOV A, ENABLE
ACALL BUTTON
RET

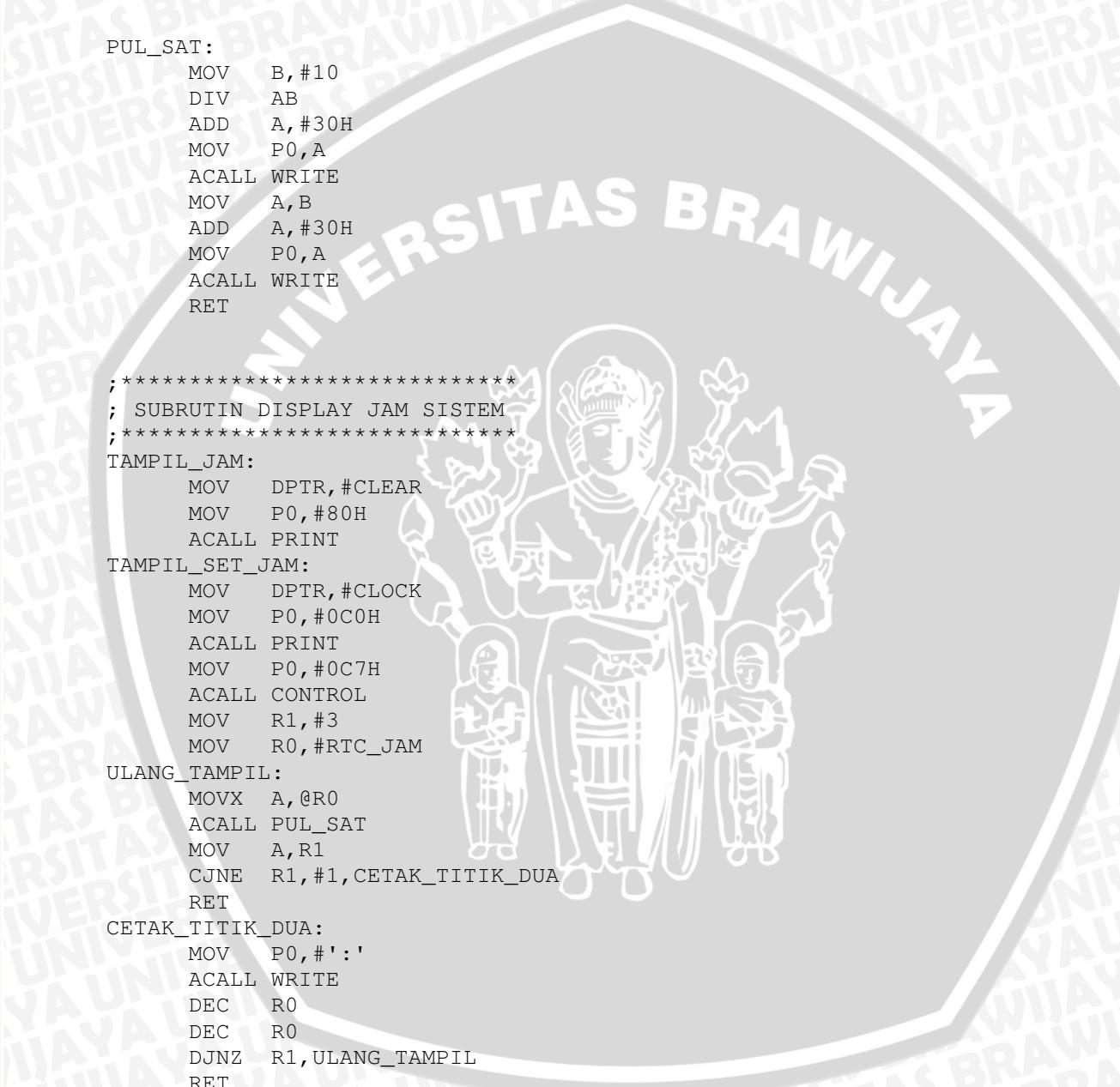
;*****
; SUBRUTIN TAMPILAN TIAP JADWAL
;*****

TAMPIL_SET_JADWAL:
    MOV P0, #8CH
    ACALL CONTROL
    MOV A, POINTER
    ADD A, #31H
    MOV P0, A
    ACALL WRITE
    MOV DPTR, #0
    CLR C
    MOV A, POINTER
    MOV B, #4
    MUL AB
    ADD A, DPL
    MOV DPL, A
    CLR A
    ADDC A, DPH
    MOV DPH, A
    MOV P0, #0C0H
    ACALL CONTROL
    ACALL BACA_EEPROM
    ACALL PUL_SAT
    INC DPTR
    MOV P0, #0C3H
    ACALL CONTROL
    ACALL BACA_EEPROM
    ACALL PUL_SAT
    INC DPTR
    MOV P0, #0C8H
    ACALL CONTROL
    ACALL BACA_EEPROM
    ACALL PUL_SAT
    INC DPTR
    ACALL BACA_EEPROM
    ACALL BUTTON
    RET

BUTTON:
    MOV P0, #0CEH
    ACALL CONTROL
    CJNE A, #0, AKTIF
```



; Pointer ke data setting di EEPROM



```
MOV    P0, #1
ACALL WRITE
RET
AKTIF:
MOV    P0, #0
ACALL WRITE
RET

PUL_SAT:
MOV    B, #10
DIV    AB
ADD    A, #30H
MOV    P0, A
ACALL WRITE
MOV    A, B
ADD    A, #30H
MOV    P0, A
ACALL WRITE
RET

; *****
; SUBRUTIN DISPLAY JAM SISTEM
; *****

TAMPIL_JAM:
MOV    DPTR, #CLEAR
MOV    P0, #80H
ACALL PRINT

TAMPIL_SET_JAM:
MOV    DPTR, #CLOCK
MOV    P0, #0C0H
ACALL PRINT
MOV    P0, #0C7H
ACALL CONTROL
MOV    R1, #3
MOV    R0, #RTC_JAM

ULANG_TAMPIL:
MOVX   A, @R0
ACALL PUL_SAT
MOV    A, R1
CJNE   R1, #1, CETAK_TITIK_DUA
RET

CETAK_TITIK_DUA:
MOV    P0, #':'
ACALL WRITE
DEC    R0
DEC    R0
DJNZ   R1, ULANG_TAMPIL
RET

; *****
; SUBRUTIN MEMBACA TAKARAN PAKAN IKAN
; *****

BACA_TIMBANGAN:
ACALL BACA_ADC
MOV    A, ADC_DAT
MOV    DPTR, #TERJEMAHAN
MOVC   A, @A+DPTR
MOV    BERAT_TIMBANG, A
```

RET

TERJEMAHAN:

```
;          0 1 2 3 4 5 6 7 8 9
DB 00,00,00,00,00,00,00,00,00,00 ;0
DB 00,00,00,00,00,00,00,00,00,00 ;1
DB 00,00,00,00,00,01,01,01,01,01 ;2
DB 01,01,01,01,01,01,01,01,01,02 ;3
DB 02,02,02,02,02,02,02,02,02,02 ;4
DB 02,03,03,03,03,03,03,03,03,03 ;5
DB 03,03,03,03,04,04,04,04,04,04 ;6
DB 04,04,04,04,04,04,04,05,05,05 ;7
DB 05,05,05,05,05,05,05,05,05,06 ;8
DB 06,06,06,06,06,06,06,06,06,06 ;9
DB 06,07,07,07,07,07,07,07,07,07 ;10
DB 07,07,07,08,08,08,08,08,08,08 ;11
DB 08,08,08,08,09,09,09,09,09,09 ;12
DB 09,09,09,09,09,10,10,10,10,10 ;13
DB 10,10,10,10,11,11,11,11,11,11 ;14
DB 11,11,11,12,12,12,12,12,12,12 ;15
DB 12,12,12,13,13,13,13,13,13,13 ;16
DB 13,13,13,14,14,14,14,14,14,14 ;17
DB 14,14,15,15,15,15,15,15,15,15 ;18
DB 15,16,16,16,16,16,16,16,16,17 ;19
DB 17,17,17,17,17,17,17,18,18,18 ;20
DB 18,18,18,18,18,19,19,19,19,19 ;21
DB 19,19,19,19,20,20,20,20,20,20 ;22
DB 20,20,20,20,20,20,20,20,20,20 ;23
DB 20,20,20,20,20,20,20,20,20,20 ;24
DB 20,20,20,20,20,20,20,20,20,20 ;25
DB 20,20,20,20,20
```

```
;*****
; SUBRUTIN BACA DATA TIMBANGAN
;*****
```

BACA_ADC:

```
SETB ALES
CLR ALES
JNB EOC,$
MOV A,P1
MOV C,ACC.4
MOV ADC_DAT.0,C
MOV C,ACC.7
MOV ADC_DAT.1,C
MOV C,ACC.5
MOV ADC_DAT.2,C
MOV C,ACC.6
MOV ADC_DAT.3,C
MOV C,ACC.3
MOV ADC_DAT.4,C
MOV C,ACC.2
MOV ADC_DAT.5,C
MOV C,ACC.1
MOV ADC_DAT.6,C
MOV C,ACC.0
MOV ADC_DAT.7,C
RET
```

```
;*****
```

```
; SUBRUTIN MEMBACA PENEKANAN TOMBOL
;*****
```

BACA_KEYPAD:

```
    MOV    A, P2
    SWAP   A
    ANL    A, #0FH
    MOV    DPTR, #TERJEMAH
    MOVC   A, @A+DPTR
    MOV    TOMBOL, A
    RET
```

TERJEMAH:

```
    DB    0AH    ;0
    DB    0BH    ;1_ENTER
    DB    00H    ;2
    DB    0CH    ;3_DOWN
    DB    04H    ;4
    DB    06H    ;5
    DB    05H    ;6
    DB    0EH    ;7
    DB    07H    ;8
    DB    09H    ;9
    DB    08H    ;A
    DB    0DH    ;B_UP
    DB    01H    ;C
    DB    03H    ;D
    DB    02H    ;E
    DB    0FH    ;F
```

```
;*****
; SUBRUTIN MENCETAK STRING KE LCD
;*****
```

PRINT:

```
    ACALL CONTROL
```

LCD:

```
    CLR    A
    MOVC   A, @A+DPTR
    CJNE   A, #00, LCD1
    RET
```

LCD1:

```
    MOV    P0, A
    ACALL WRITE
    INC    DPTR
    SJMP   LCD
```

```
;*****
; SUBRUTIN INISIALISASI SISTEM
;*****
```

INIT:

```
    CLR    ALES
    CLR    DONE
    CLR    ON_SET
    CLR    ON_CEK
    CLR    ON_SET_JAM
    CLR    ON_INTERUPSI
    SETB   PINTU_TANDON
    SETB   PINTU_TIMBANG
    MOV    TCON, #04H
    MOV    IE, #84H
```

```
MOV    96H, #0AH
RET

;*****
; SUBRUTIN TULIS EEPROM INTERNAL
;*****
TULIS_EEPROM:
    MOV    96H, #18H
    MOVX   @DPTR, A
TUNGGU_EEPROM:
    MOV    A, 96H
    JNB    ACC.1, TUNGGU_EEPROM
    MOV    96H, #0AH
    RET

;*****
; SUBRUTIN MEMBACA EEPROM INTERNAL
;*****
BACA_EEPROM:
    MOV    96H, #08H
    MOVX   A, @DPTR
    MOV    96H, #0AH
    RET

;*****
; SUBRUTIN INISIALISASI RTC
;*****
INIT_RTC:
    MOV    R0, #0AH
    MOV    A, #2FH
    MOVX   @R0, A
    MOV    R0, #0BH
    MOV    A, #2EH
    MOVX   @R0, A
    MOV    R0, #0CH
    MOV    A, #00H
    MOVX   @R0, A
    RET

;Aktifkan clock, set divider SQWE 500mS

;Aktifkan interupsi alarm dan SQWE

;*****
; SUBRUTIN INISIALISASI LCD
;*****
INIT_LCD:
    MOV    P0, #38H      ;FUNCTION SET 8 BIT
    ACALL CONTROL
    MOV    P0, #0CH      ;DISPLAY ON, CURSOR OFF, BLINK OFF
    ACALL CONTROL
    MOV    P0, #06H      ;INCREMENT, NO DISPLAY SHIFT
    ACALL CONTROL
    MOV    P0, #01H      ;CLEAR DISPLAY
    ACALL CONTROL
    MOV    P0, #02H      ;CURSORHOME
    ACALL CONTROL
    ACALL BIGDELAY
    MOV    P0, #01000000B
    MOV    DPTR, #BLOK1
    ACALL MY_PATTERN
```

```
MOV    P0, #01001000B
MOV    DPTR, #BLOK2
ACALL MY_PATTERN
MOV    P0, #01010000B
MOV    DPTR, #BLOK3
ACALL MY_PATTERN
MOV    P0, #01011000B
MOV    DPTR, #BLOK4
ACALL MY_PATTERN
RET
```

```
; ****
; SUBRUTIN PENULISAN MATRIK KARAKTER
; ****
```

```
MY_PATTERN:
```

```
    ACALL CONTROL
    MOV    R0, #8
```

```
ULANG_PAT:
```

```
    CLR    A
    MOVC  A, @A+DPTR
    MOV    P0, A
    ACALL WRITE
    INC    DPTR
    DJNZ  R0, ULANG_PAT
    RET
```

```
BLOK1:
```

```
    DB    00000B
    DB    01110B
    DB    10101B
    DB    11111B
    DB    10101B
    DB    01110B
    DB    00000B
    DB    00000B
```

```
BLOK2:
```

```
    DB    00000B
    DB    01110B
    DB    10001B
    DB    10001B
    DB    10001B
    DB    01110B
    DB    00000B
    DB    00000B
```

```
BLOK3:
```

```
    DB    00000B
    DB    00100B
    DB    00100B
    DB    00100B
    DB    11111B
    DB    01110B
    DB    00100B
    DB    00000B
```

```
BLOK4:
```

```
    DB    01000B
    DB    01100B
    DB    01110B
```



DB 01111B
DB 01110B
DB 01100B
DB 01000B
DB 00000B

CONTROL:

CLR RS
CLR E
SETB E
ACALL DELAY
RET

WRITE:

SETB RS
CLR E
SETB E
ACALL DELAY
RET

BIGDELAY:

MOV R7, #10

BIG:

MOV R6, #164
DJNZ R6, \$
DJNZ R7, BIG
RET

DELAY:

MOV R7, #40
DJNZ R7, \$
RET

ONE_SEC_DELAY:

MOV R0, #50

ONE_SEC1:

MOV R1, #100

ONE_SEC2:

MOV R2, #100
DJNZ R2, \$
DJNZ R1, ONE_SEC2
DJNZ R0, ONE_SEC1
RET

VAR_DELAY:

MOV R1, #250
DJNZ R1, \$
DJNZ R0, VAR_DELAY
RET

;*****
; DEFINISI KONSTANTA STRING LCD
;*****

JUDUL:



```
;          0123456789ABCDEF
DB      'PENGATUR MAKANAN',0
DB      ' IKAN AQUARIUM ',0
DB      ' BERBASIS MCU ',0
DB      'ATMEL AT89S8252 ',0

CLOCK:
DB      ' PUKUL JJ:MM:DD ',0

PIL_MENU:
DB      ' 1.SET JADWAL ',0
DB      ' 2.SET WAKTU ',0
DB      ' 3.KALIBRASI ',0

CLEAR:
SET_JADWAL_TEXT:
DB      ' ',0
DB      'JJ:MM - BBgr E ',0

PIL_JADWAL:
DB      ' JADWAL KE-N ',0
DB      'JJ:MM - BBgr E ',0

JUMLAH:
DB      'MAKANAN : gr ',0

CALIBRATE:
DB      'KALIBRASI: gr ',0

END
```

