PERANCANGAN DAN PEMBUATAN ALAT PENETAS TELUR AYAM OTOMATIS BERBASIS MIKROKONTROLER AT89S51

SKRIPSI
JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

ELLA PUSPA SARI

NIM. 0610633025-63

KEMENTERIAN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2010

repository.ub.ac.id

UNIVERSITAS BRAWIJAYA

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN ALAT PENETAS TELUR AYAM OTOMATIS BERBASIS MIKROKONTROLER AT89S51

SKRIPSI
JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
ELLA PUSPA SARI
NIM. 0610633025-63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Purwanto, MT.	**Ir. Bambang Siswojo, MT.**
NIP. 19540424 198601 1 001	**NIP. 19621211 198802 1 001**

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LEMBAR PENGESAHAN

PERANCANGAN DAN PEMBUATAN ALAT PENETAS TELUR AYAM OTOMATIS BERBASIS MIKROKONTROLER AT89S51

SKRIPSI
JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

ELLA PUSPA SARI
NIM. 0610633025-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 11 November 2010

DOSEN PENGUJI

Ir. Retnowati, MT.
NIP. 19511224 198203 2 001

Ir. Erni Yudaningtyas, MT
NIP. 19650913 199002 2 001

Tri Nurwati, ST., MT.
NIP. 19790615 200812 2 003

Mengetahui
Ketua Jurusan Teknik Elektro

Rudy Yuwono ST., M.Sc.
NIP. 19710615 199802 1 003

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRAKSI

Ella Puspa Sari. 0610633025-63**. Perancangan dan Pembuatan alat Penetas Telur Ayam Otomatisasi Berbasis Mikrokontroller AT89S51.** Skripsi Jurusan Teknik Elektro**,** Fakultas Teknik, Universitas Brawijaya. Pembimbing: **Ir. Purwanto, MT** dan **Ir.Bambang Siswojo,MT.**

Untuk mendapatkan anak ayam dalam jumlah banyak saat yang bersamaan akan menjadi masalah kalau hanya didapatkan secara alami, dimana satu ekor induk ayam hanya bisa mengeram maksimal 10 butir telur, jika menginginkan dalam jumlah yang banyak dan saat bersamaan, akan menjadi kendala. Dengan masa mengeram yang tidak bisa ditentukan secara bersamaan sekian banyak induk ayam, sehingga kalau didaerah pedesaan hal ini akan menimbulkan masalah kalau ada penduduk yang ingin beternak ayam sebagai pekerjaan sambilan.

Dengan mempelajari cara-cara penetasan telur dengan memperhatikan pengaturan suhu ruang penetasan, dengan lama waktu pemanasan yang bisa diatur dan bisa bekerja menyerupai dengan kelakukan seekor induk ayam, maka ada ide pengembangan penetasan telur secara buatan, yang mana semua ini bisa diprogramkan pada mikrokontroller AT89S51 sebagai pusat kontrol dan sensor suhu dan kelembaban HSM 20G sebagai pengatur suhu dan kelembaban agar telur bisa menetas, berapa lama pemanasan, dan banyaknya anak ayam yang bisa didapatkan dapat direncanakan waktu penetasannya.

Dari pernelitian ini bisa didapatkan suatu alat penetas telur yang sederhana, yang dapat bekerja sesuai dengan rencana, dengan kesalahan pengukuran sebesar 2,012% untuk pengujian suhu dan kesalahan rata-rata 2,05% pada saat pengujian kelembaban dengan menggunakan sensor suhu dan kelembaban HSM20G. Sedangkan untuk tingkat keberhasilan penetasan adalah sebesar 70% dari jumlah total telur ayam yg ingin ditetaskan.

Kata kunci: Penetasan telur ayam, Mikrokontroller AT89S51, HSM 20G.

UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Alhamdulillah, segla puji bagi Allah swt, Atas karunia ilmu, nikmat rezeki yang diberikan Nya, sehingga Penulis dapat menyelesaikan skripsi yang berjudul "Perancangan dan Pembuatan Alat Penetas Telur Ayam Otomatis Berbasis Mikrokontroller AT89S51" ini dengan baik.

Tidak lupa juga, Penulis ingin menyampaikan rasa terima kasih kepada :

1. Bapak Rudi Yuwono, ST, M.Sc. dan Bapak M. Azis Muslim, ST.,MT., Ph.D selaku Ketua dan Sekretaris Jurusan Teknik Elektro.
2. Bapak Ir. Purwanto, MT., selaku KKDK Teknik Kontrol dan dosen pembimbing I yang telah memberikan ide, bimbingan serta saran hingga terselesaikannya skripsi ini.
3. Bapak Ir. Bambang Siswojo, MT., selaku pembimbing II yang telah memberikan ide, bimbingan, serta saran hingga terselesaikannya skripsi ini.
4. Ibu Ir. Erni Yudaningtyas, MT selaku pembimbing akademik.
5. Bapak, ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
6. Kedua orang tuaku tercinta, Slamet Susanto dan Hendri Utami yang senantiasa memberi semangat, dorongan, do'a serta kasih sayang yang tiada tara selama ini.
7. Nenekku tercinta, kakakku Martanius Andy Setiadi, Sahabatku Amillia Kusuma Putri , Budi Wijaya, Fabian Riza, Rangga Paksi dan Novi Setyo yang selalu ada dalam suka dan duka, selalu setia mendengarkan curahan hati, dan selalu memberikan do'a,dorongan serta semangat. Makasih sobat.
8. Sahabat- sahabat seperjuanganku di bumi Elektro, Lucky Nindya, Risty Jayanti, Nungky Indra, Nur Annisa, Rendy Lucki, Gayuh Cipta Adi, M. Hifnie, Lunde Ardhenta, M.Denny Wirahadi atas motivasi dan semangatnya selama ini.
9. Teman seperjuanganan Praktek Kerja dan Skripsi, Yusron Fuadi. Makasih buat motivasi dan semangatnya buat cepet menyelesaikan skripsi.

10. Teman-teman Pak-D, G-Force, Cewek-cewek kesebelasan IPA Smansa Tuban'03, teman-teman kost KL-80 dan Mas Rizal yang selalu ngoprak-ngoprak buat cepet lulus.
11. Dan semua pihak yang telah mendukung baik moril, materiil, maupun do'anya yang tidak bisa Penulis sebutkan satu per satu. Terima kasih banyak.

Penulis menyadari dalam penyusunan dan penulisan laporan skripsi ini masih banyak kekurangan dan kekeliruan, untuk itu Penulis mengharapkan masukan kritik dan saran demi kelengkapan dan kesempurnaan laporan skripsi ini.

Akhir kata, Penulis mengharapkan semoga skripsi ini dapat bermanfaat bagi rekan-rekan mahasiswa khususnya dan seluruh pembaca pada umumnya. Amin.

Malang, Oktober 2010

Hormat kami,

Penulis

DAFTAR ISI

ABSTRAK i
KATA PENGANTAR ii
DAFTAR ISI iii
DAFTAR GAMBAR viii
DAFTAR TABEL x

BAB I. PENDAHULUAN

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

BAB II. TINJAUAN PUSTAKA

2.1 Teori Dasar Penetasan 5
2.1.1 Istilah dalam Penetasan 5
2.1.2 Faktor-faktor yang mempengaruhi Daya Tetas 7
2.1.3 Suhu Tetas Telur 9
2.2 Mikrokontroler AT89S51 11
2.2.1 Konfigurasi Pin 13
2.2.2 Struktur dan Operasi Port 14
2.2.3 Organisasi Memori 16
2.2.4 Reset 17
2.3 Sensor Suhu dan Kelembaban HSM20G 18
2.4 Pengkondisi sinyal 21
2.4.1 Penguat Operasional 21
2.4.2 Penguat Tak Membalik 21
2.5 Penambah Kelembaban (*Humidifier*) 22
2.6 LCD (*Liquid Crystal Display*) M1632 23
2.7 ADC (*Analog to Digital Converter*) PCF8591 25

2.8 RTC (*Real Time Clock*) DS1307 26
2.9 Motor Listrik 28
2.9.1 Motor DC 28
2.10 Relay 29
2.11 Transistor 31
2.11.1 Persamaan Transistor 32
2.11.2 Garis Beban DC 32
2.11.3 Transistor sebagai Saklar 33
2.12 Buzzer 34
2.13 Pemanas (*Heater*) 35
2.14 Keypad Matrik 4x4 36
2.15 DC Brushless Fan 36

BAB III. METODOLOGI PENELITIAN

3.1 Studi Literatur 38
3.2 Perancangan Alat 39
3.3 Realisasi Pembuatan Alat 39
3.4 Pengujian Alat dan Analisa Data 40
3.5 Pengambilan Kesimpulan 40

BAB IV. PERANCANGAN DAN PEMBUATAN ALAT

4.1 Tinjauan Umum 41
4.1.1 Blok Diagram Alat 42
4.1.2 Prinsip Kerja Alat 42
4.2. Perancangan Perangkat Keras 43
4.2.1 Rangkaian Sensor Suhu dan Kelembaban HSM20G 43
4.2.2 Rangkaian Penguat Sinyal 45
4.2.3 Rangkaian ADC PCF8591 48
4.2.4 Rangkaian Mikrokontroler AT89S51 49
4.2.5 Rangkaian RTC DS1307 51
4.2.6 Rangkaian Keypad 4x4 52
4.2.7 Rangkaian LCD M1632 54
4.3. Perancangan Rangkaian Driver 55

4.3.1 Driver Motor 55
4.3.1.1 Motor DC 56
4.3.2 Driver Relay 56
4.4 Perancangan Perangkat Lunak 59

BAB V. PENGUJIAN DAN ANALISIS

5.1 Pengujian Sensor Suhu dan Kelembaban HSM20G 62
5.1.1 Tujuan Pengujian 62
5.1.2 Peralatan yang digunakan 62
5.1.3 Prosedur Pengujian 62
5.1.4 Hasil dan Analisis Pengujian 63
5.2 Pengujian ADC PCF8591 67
5.1.1 Tujuan Pengujian 67
5.1.2 Peralatan yang digunakan 67
5.1.3 Prosedur Pengujian 67
5.1.4 Hasil dan Analisis Pengujian 68
5.3 Pengujian LCD M1632 69
5.1.1 Tujuan Pengujian 69
5.1.2 Peralatan yang digunakan 69
5.1.3 Prosedur Pengujian 69
5.1.4 Hasil dan Analisis Pengujian 69
5.4 Pengujian Mikrokontroler AT89S51 70
5.1.1 Tujuan Pengujian 70
5.1.2 Peralatan yang digunakan 70
5.1.3 Prosedur Pengujian 70
5.1.4 Hasil dan Analisis Pengujian 71
5.5 Pengujian Driver Motor 71
5.1.1 Tujuan Pengujian 71
5.1.2 Peralatan yang digunakan 72
5.1.3 Prosedur Pengujian 72
5.1.4 Hasil dan Analisis Pengujian 72
5.6 Pengujian Sistem Keseluruhan 73
5.1.1 Tujuan Pengujian 73

5.1.2 Peralatan yang digunakan 73
5.1.3 Prosedur Pengujian 73
5.1.4 Hasil dan Analisis Pengujian 73

BAB VI. KESIMPULAN DAN SARAN

6.1. Kesimpulan 76
6.2. Saran 76

DAFTAR PUSTAKA 77
LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Blok Diagram AT89S51 12
Gambar 2.2 Konfigurasi Pin AT89S51 13
Gambar 2.3 Struktur Memori MCS-51 17
Gambar 2.4 Rangkaian Power On Reset 17
Gambar 2.5 Sensor Suhu dan Kelembaban HSM20G 18
Gambar 2.6 Kurva Respon HSM20G pada 25°C (*Voltage - Humidity*) 19
Gambar 2.7 Dimensi Sensor Suhu dan Kelembaban HSM20G 20
Gambar 2.8 Kurva Respon HSM20G pada 25°C (*Voltage - Temperature*) 20
Gambar 2.9 Penguat Op-Amp 21
Gambar 2.10 Penguat Tak Membalik 22
Gambar 2.11 Diagram Blok LCD M1632 24
Gambar 2.12 Susunan Pin ADDA PCF8591 25
Gambar 2.13 Susunan Pin DS1307 26
Gambar 2.14 Diagram Blok DS1307 27
Gambar 2.15 Motor DC 29
Gambar 2.16 Rangkaian Relay 30
Gambar 2.17 Rangkaian DPDT 31
Gambar 2.18 Simbol Transistor 31
Gambar 2.19 Bias Basis Transistor dan Garis Beban DC 33
Gambar 2.20 Transistor sebagai Saklar 34
Gambar 2.21 Buzzer 35
Gambar 2.22 Rangkaian Dasar Keypad 4x4 36
Gambar 2.23 DC Brushless Fan 37
Gambar 4.1 Diagram Blok Sistem 42
Gambar 4.2 Rangkaian Perencanaan Sensor Suhu dan Kelembaban HSM20G 43
Gambar 4.3 Rangkaian Penguat Non Inverting 46
Gambar 4.4 Rangkaian Perencanaan ADC PCF8591 49
Gambar 4.5 Rangkaian Perencanaan Mikrokontroller AT89S51 50
Gambar 4.6 Rangkaian Perencanaan RTC (*Real Time Clock*) 51
Gambar 4.7 Flowchart Scanning Keypad dengan Metode Polling 52
Gambar 4.8 Susunan Tombol pada Keypad 4x4 53

Gambar 4.9 Alokasi Pin pada Keypad 4x4 53
Gambar 4.10 Rangkaian Perencanaan Keypad 4x4 54
Gambar 4.11 Rangkaian Perencanaan LCD M1632 55
Gambar 4.12 Rangkaian Perencanaan Driver Motor 56
Gambar 4.13 Rangkaian Perencanaan Driver Relay 59
Gambar 4.14 Flowchart Program Utama 61
Gambar 5.1 Pengujian Rangkaian Sensor Suhu dan Kelembaban dan IcPCF8591 63
Gambar 5.2 Rangkaian Modul HSM20G dari datasheet 63
Gambar 5.3 Gambar Tampilan LCD pada Alat 70
Gambar 5.4 Diagram Blok Pengujian Mikrokontroller AT89S51 71
Gambar 5.5 Rangkaian Pengujian Driver Motor 72

DAFTAR TABEL

Tabel 2.1 Suhu Tetas Telur Unggas 9
Tabel 2.2 Fungsi Alternatif Port 3 16
Tabel 2.3 Spesifikasi HSM20G 18
Tabel 2.4 Pin Sensor Suhu dan Kelembaban HSM20G 20
Tabel 2.5 Fungsi Pin-Pin pada LCD M1632 24
Tabel 2.6 Fungsi Pin ADDA PCF8591 26
Tabel 2.7 Alamat dan Isi Register DS1307 28
Tabel 5.1 Hasil Pengukuran dan Pengujian Suhu 65
Tabel 5.2 Hasil Pengukuran dan Pengujian Kelembaban 66
Tabel 5.3 Hasil Pengujian ADC PCF8591 68
Tabel 5.4 Hasil Pengujian Sistem Mikrokontroller 71
Tabel 5.5 Hasil Pengujian Driver Motor 72
Tabel 5.6 Suhu Pengujian Proses Penetasan Telur 74
Tabel 5.7 Pengujian I Keberhasilan Penetasan Telur 74
Tabel 5.8 Pengujian II Keberhasilan Penetasan Telur 75

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam bidang peternakan khususnya dalam peternakan ayam, masalah yang sering dihadapi oleh para peternak yaitu bagaimana menetaskan telur ayam dalam jumlah banyak dan dalam waktu yang bersamaan. Karena kemampuan induk ayam dalam mengerami telurnya terbatas, yaitu maksimal 10 butir telur tiap induk ayam dan selama mengerami dan mengasuh anak-anaknya kegiatan bertelur terhenti.

Beberapa komoditas daging ayam yang masih impor merupakan permasalahan tersendiri karena rendahnya produktivitas ayam dan masih banyaknya masyarakat yang mengelola penetasan telur secara tradisional. Keberadaaan mesin tetas diharapkan mampu meningkatkan produktivitas telur dan memperbaiki perekonomian rakyat.

Pada prinsipnya untuk menetaskan telur ayam hanya menjaga suhu pada telur tersebut agar stabil sesuai yang dibutuhkan telur agar bisa menetas. Embrio akan berkembang bila suhu udara di sekitar telur minimal 70° F (21,11° C) namun perkembangan ini sangat lambat. Di bawah suhu udara ini praktis embrio tidak mengalami perkembangan., sehingga penyimpanan telur tetas sebaiknya sama atau di bawah suhu tersebut. Suhu yang baik untuk pertumbuhan embrio adalah berkisar diantara 38,5°C-40,5°C dan pada kelembaban 70% (Zainal Abidin, 2003:13).

Mesin penetas telur konvensional mulai dikembangkan untuk meningkatkan produktivitas ayam, yaitu dengan cara mengatur suhu mesin dan kelembaban. Mesin penetas konvensional yang sangat sederhana menggunakan lampu *teplok* dengan menggunakan minyak tanah yang banyak digunakan di desa-desa, tetapi mesin penetas ini banyak mengalami kendala karena kesulitan dalam pengaturan suhu dan pengukuran. Selain itu, ada juga mesin penetas yang menggunakan bimetal sebagai saklar sekaligus kontroler on/off. Mesin penetas jenis ini lebih baik pengaturan suhunya daripada menggunakan lampu minyak tanah. Karena menggunakan saklar paduan dua logam yang mempunyai koefisien muai berbeda. Akan tetapi, saklar bimetal

ini mempunyai kelemahan dalam hal lama pemakaian. Semakin lama pemakaiannya kepekaan bimetal akan berkurang. Selain itu, mesin penetas konvensional yang ada saat ini pun tidak dapat mengatur suhu penetasan yang tiap hari berubah. Serta, mesin penetas telur yang beredar di pasaran saat ini masih manual pada pemutaran rak telurnya.

Dalam penetasan telur secara alami, biasanya induk ayam akan memutar telurnya beberapa kali dalam sehari, hal ini bertujuan untuk meratakan panas telur selama periode penetasan dan menjaga embrio agar tidak lengket pada salah satu kerabang. Telur yang baru keluar dari induk ayam memiliki kuning telur dengan berat jenis normal dan berada dalam albumen encer sehingga jika tidak diputar kuning telur akan terdorong ke salah satu sisi atas dan melekat ke kerabang telur dan embrio akan mati. Maka dari itu, untuk melengkapi kekurangan-kekurangan pada mesin tetas yang ada di pasaran saat ini, dalam perancangan alat ini, pemutaran telur tetas akan dibuat berputar secara otomatis setiap enam jam sekali dengan kemiringan sudut 45° (Zainal Abidin, 2003:7).

Mesin tetas dengan pengaturan suhu telur tetas yang dapat diatur setiap harinya secara otomatis, pemutaran telur tetas otomatis, dan pengaturan sirkulasi udara sangat dibutuhkan untuk peningkatan daya tetas telur dan produktivitas penetasan telur. Pada penelitian ini proses penetasan telur ayam akan memanfaatkan Mikrokontroller AT89S51 sebagai kontroler otomatisnya dikarenakan memiliki fitur dan jumlah I/O yang sesuai dengan kebutuhan sistem.

1.2 Rumusan Masalah

Dari latar belakang masalah di atas, maka didapatkan perumusan masalah sebagai berikut:

1. Bagaimana merancang dan membuat mesin penetas telur ayam otomatis?
2. Bagaimana merancang mikrokontroler AT89S51 untuk mengontrol suhu dan kelembaban pada mesin penetas telur ayam?

1.3 Batasan Masalah

Karena begitu luasnya objek kajian maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah dalam skripsi ini adalah :

1. Mikrokontroler yang digunakan adalah Mikrokontroler AT89S51 yang mempunyai 4 kbyte Flash PEROM (*Flash Programmable and Erasable Read Only Memory*), 128 byte RAM, 32 pin I/O (4 buah port I/O bit), mempunyai dua buah timer/ counter 16 bit, dan mampu beroperasi sampai 24 MHz.
2. Sensor suhu dan kelembaban menggunakan HSM20G.
3. Penentuan posisi dan jumlah sensor suhu disesuaikan dengan pola aliran udara serta dimensi alat pengering.
4. Kapasitas mesin penetas telur dalam penelitian ini adalah 6-18 butir
5. Posisi telur tegak dengan kantung udara di atas, pemutaran ke kanan atau ke kiri dengan sudut 45° dari arah vertikal.
6. Pemanas yang digunakan adalah heater 220 volt dari elemen pemanas solder.

1.4 Tujuan

Tujuan yang ingin dicapai dari tugas akhir ini adalah merancang dan membuat model percobaan (model eksperimental) alat penetas telur ayam otomatis, sehingga diharapkan dapat membantu para peternak ayam untuk mengembangbiakan ayam tanpa harus menghentikan proses perkembangbiakan ayam ketika induk masih mengerami telurnya. Serta meningkatkan daya tetas dan mengurangi persentase kegagalan penetasan telur akibat pengeraman yang tidak merata.

1.5 Sistematika Penulisan

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan serta sistematika penulisan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Metodologi

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan Sistem

Membahas perancangan sistem pengendalian proses penetasan telur ayam dengan mikrokontroler. Setelah itu, bagaimana menerapkan- nya dalam sistem secara keseluruhan.

BAB V Pengujian Alat

Membahas hasil pengujian sistem terhadap alat yang telah direalisasikan.

BAB VI Kesimpulan dan Saran

Membahas kesimpulan dan saran-saran yang diperlukan untuk melakukan pengembangan aplikasi selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1 Teori Dasar Penetasan Telur

Penetasan telur unggas yang dilakukan secara alami yaitu penetasan melalui induk unggas, sangat terbatas jumlahnya. Penetasan telur juga dapat dilakukan dengan mesin penetas yang mengkondisikan telur dalam suhu tertentu selama penetasan, sehingga telur dapat menetas secara alami seperti dierami oleh induknya. Waktu dan suhu penetasan telur untuk tiap jenis unggas berbeda-beda. Pada umumnya, waktu tetas telur unggas sebagai berikut :

- Puyuh : 18 hari
- Merpati : 17 hari
- Ayam : 21 hari
- Itik : 28 hari
- Entok : 35 hari

2.1.1 Istilah Dalam Penetasan

Beberapa istilah dalam penetasan telur unggas adalah :

a. Telur tetas

Telur tetas adalah telur unggas/ burung dan reptil yang telah dibuahi (adanya pertemuan antara sperma dan ovum). Telur yang dihasilkan oleh unggas/burung/reptil tidak semua dibuahi, Karena tanpa ditunasi oleh seekor pejantan sekalipun, maka unggas atau reptil betina tersebut dapat menghasilkan telur. Oleh karena itu, untuk menghasilkan telur tetas yang baik untuk ditetaskan harus ada perkawinan antara unggas jantan dan betina.

b. Fertilitas

Fertilitas merupakan istilah yang menerangkan tentang kesuburan, yaitu setelah adanya pertamuan antara spermatozoa dengan ovum pada bagian infundibulum. Sebagai bukti bahwa telur itu fertil dapat dilihat dengan alat yang disebut *candler*.

Untuk menghitung prosentase fertilitas dari jumlah telur yang dimasukkan dalam mesin tetas adalah :

$$\%\ Fertilitas = \frac{jumlah\ telur\ yang\ fertil}{jumlah\ telur\ yang\ ditetaskan} \text{ x } 100\%$$

c. Daya tetas

Daya tetas merupakan istilah yang menerangkan tentang telur-telur yang pecah menjadi anak unggas artinya banyaknya telur yang menetas dari jumlah telur yang di fertil. Untuk menghitung prosentase daya tetas dapat digunakan rumus sebagai berikut :

$$\%\ Daya\ tetas = \frac{jumlah\ telur\ yang\ menetas}{jumlah\ telur\ yang\ fertil} x100\%$$

d. Bobot tetas

Merupakan istilah yang menerangkan tentang bobot tetas telur yang menetas, artinya diperoleh dari penimbangan bobot anak unggas yang menetas (umur satu hari)

e. Candling

Pemeriksaan telur tetas setelah hari ketiga dengan alat teropong telur yang disebut *candler* disebut *candling*. Tujuan dari proses ini adalah untuk mengetahui telur yang bertunas (fertil) dan embrio yang mati.

f. Dead embrio

Adalah embrio pada saat hari ke-4, ke-7 (telur ayam) mati sebelum menetas, yang ditandai dengan bintik hitam pada saat *candling*.

g. Proses *Pipping*

Proses *Pipping* merupakan proses pemecahan pertama kali pada kulit telur dalam proses penetasan. Pada anak ayam yang baru menetas pada ujung paruhnya terdapat *papilla* yang disebut dengan *dens ovigrafus*, yaitu tonjolan kecil berwarna keputih-putihan yang merupakan zat tanduk. Setelah menetas

berumur satu hari zat tersebut akan menghilang dengan sendirinya.

h. Masa kritis

Masa kritis adalah suatu masa yang paling banyak terjadi kegagalan dalam menetaskan telur tetas (terjadinya kematian pada embrio). Hal ini dapat disebabkan oleh beberapa sebab, diantaranya adalah tegangan listrik sebagai sumber panas yang tidak stabil, sehingga berakibat fluktuasi suhu dan kelembaban mesin tetas yang pada gilirannya akan menyebabkan embrio mati.

Masa kritis dibagi menjadi 2, yaitu :

- Masa kritis pertama

  Masa ini terjadi 3 hari pertama sejak telur dimasukkan ke dalam mesin tetas. Pada periode ini terjadi perkembangan blastoderm yang sangat pesat, adanya perubahan zat kimia dalam telur dan adanya penimbunan asam laktat yang cukup tinggi. Oleh karena itu, suhu dan kelembaban dalam mesin tetas harus senormal mungkin, sehingga embrio akan hidup sempurna pula. Masa ini untuk semua jenis unggas sama waktunya.

- Masa kritis kedua

  Masa ini terajadi 3 hari terakhir dari perkiraan proses penetasan. Pada periode ini untuk setiap jenis unggas berbeda, misalnya untuk ayam 18-21 hari, itik 24-28 hari dan puyuh 15-18 hari. Pada masa ini ditandai oleh perubahan fisiologi embrio yang telah sempurna menjelang proses penetasan.

2.1.2 Faktor – Faktor yang Mempengaruhi Daya Tetas

a. Telur tetas harus fertil

Untuk mendapatkan telur yang fertil harus berasal dari perkawinan antara unggas jantan dan betina.

b. Telur tetas harus dari induk atau pejantan yang sehat

   Pejantan atau induk yang sakit saat perkawinan, maka sperma yang dihasilkan akan terinfeksi oleh penyakit yang diderita oleh unggas tersebut, akibatnya telur yang dihasilkan akan terinfeksi pula oleh penyakit tersebut.

c. Berat telur

   Untuk telur ayam ras, minimal beratnya sekitar 50 gram dan setinggi- tingginya 65 gram. Telur yang terlalu besar, biasanya kuning telurnya ganda dan tidak menetas walaupun dieramkan. Sebaliknya telur yang terlalu kecil, juga kurang menetas dengan baik.

d. Bentuk telur

   Telur-telur yang bentuknya menyimpang dari keadaan normal, umumnya kurang menetas dengan baik. Telur yang bentuknya normal yaitu telur yang mempunyai perbandingan antara panjang dan lebarnya 2 : 3. Bentuk oval.

e. Keutuhan kulit telur

   Telur-telur yag dalam keadaan retak / pecah tetapi isi telur tidak keluar, tidak menetas dengan baik.

f. Kualitas Kulit Telur

   Telur dengan kulit yang tipis, kulit telur lembek, keadaan perkapuran yang kurang merata, umumnya kurang menetas dengan baik. Tebal kulit telur yang normal berkisar antara 0,33 – 0,35 mm.

g. Warna Kulit Telur

   Warna kulit telur sangat berpengaruh terhadap daya tetas telur. Apabila dari sekelompok ayam petelur menghasilkan telur yang kulitnya lebih gelap, maka akan menetas lebih baik dari yang berwarna lebih terang.

h. Kebersihan Kulit Telur

   Telur yang kotor sebaiknya tidak ditetaskan, karena telur yang kotor biasanya daya tetasnya rendah. Bisa dibersihkan dengan menggunakan kertas semen (bila kotorannya ringan)

atau dibersihkan dengan air hangat (temperatur 55°C) kemudian dikeringkan.

2.1.3 Suhu Tetas Telur

Pengaturan suhu dan pemutaran telur harus menyesuaikan dengan kebutuhan telur tetas. Suhu berperan besar dalam keberhasilan penetasan telur, terlalu panas atau dingin mempengaruhi fisiologi perkembangan embrio yang menyebabkan kegagalan penetasan telur.

Pemutaran telur tidak boleh dilakukan pada tahap kritis pertama (3-4 hari awal penetasan) karena pada tahap ini terjadi perkembangan blastoderm yang sangat cepat. Pemutaran telur juga tidak boleh dilakukan pada tahap kritis kedua (4 hari akhir penetasan) karena pada tahap ini terjadi perubahan fisiologi embrio yang telah sempurna. Suhu tetas telur ayam ditunjukan pada tabel 2.1

Tabel 2.1. Suhu tetas telur unggas

Telur puyuh Berat 10-12 gram, waktu tetas 18 hari		
Hari ke-	Suhu Ideal	Keterangan
1-3	37,0 °C	Belum diputar, belum dianginkan
4-14	37,5 °C	Mulai diputar, dianginkan
15-18	38,0 °C	Tidak diputar, dianginkan

Telur merpati Berat 20-22 gram, waktu tetas 17 hari		
Hari ke-	Suhu Ideal	Keterangan
1-3	37,0 °C	Belum diputar, belum dianginkan
4-13	37,5 °C	Mulai diputar, dianginkan
14-17	38,0 °C	Tidak diputar, dianginkan

Telur Ayam Berat 55-60 gram, waktu tetas 21 hari		
Hari ke-	Suhu Ideal	Keterangan
1-4	38,5 °C	Belum diputar, belum dianginkan
5-8	39,0 °C	Mulai diputar, dianginkan
9-14	39,5 °C	
15-17	40,0 °C	
18-21	40,5 °C	Tidak diputar, dianginkan

Telur itik Berat 60-65 gram, waktu tetas 28 hari		
Hari ke-	Suhu Ideal	Keterangan
1-4	38,5 °C	Belum diputar, belum dianginkan
5-9	39,0 °C	Mulai diputar, dianginkan
10-15	39,5 °C	
16-20	40,0 °C	
21-24	40,5 °C	
25-28	41,0 °C	Tidak diputar, dianginkan

Telur entok Berat >80 gram, waktu tetas 35 hari		
Hari ke-	Suhu Ideal	Keterangan
1-4	39,5 °C	Belum diputar, belum dianginkan
5-11	40,0 °C	Mulai diputar, dianginkan
12-21	40,5 °C	
22-30	41,0 °C	
31-35	41,5 °C	Tidak diputar, dianginkan

Sumber : Nurhayati dkk, 2003:22-26

2.2 Mikrokontroler AT89S51

Secara umum, mikrokontroler berfungsi sama dengan komputer. Bedanya adalah mikrokontroler memiliki desain dalam sebuah *single chip* (IC). Mikrokontroler terdapat di hampir semua peralatan elektronik, didalam tape, TV, radio, telepon genggam *(Hand Pho*ne) dan laian-lain. Mikrokontroler memiliki kemampuan yang diperlukan untuk membuat keputusan berdasarkan sinyal dari luar dengan kata lain mikrokontroler merupakan otak dari sebuah perangkat elektronik.

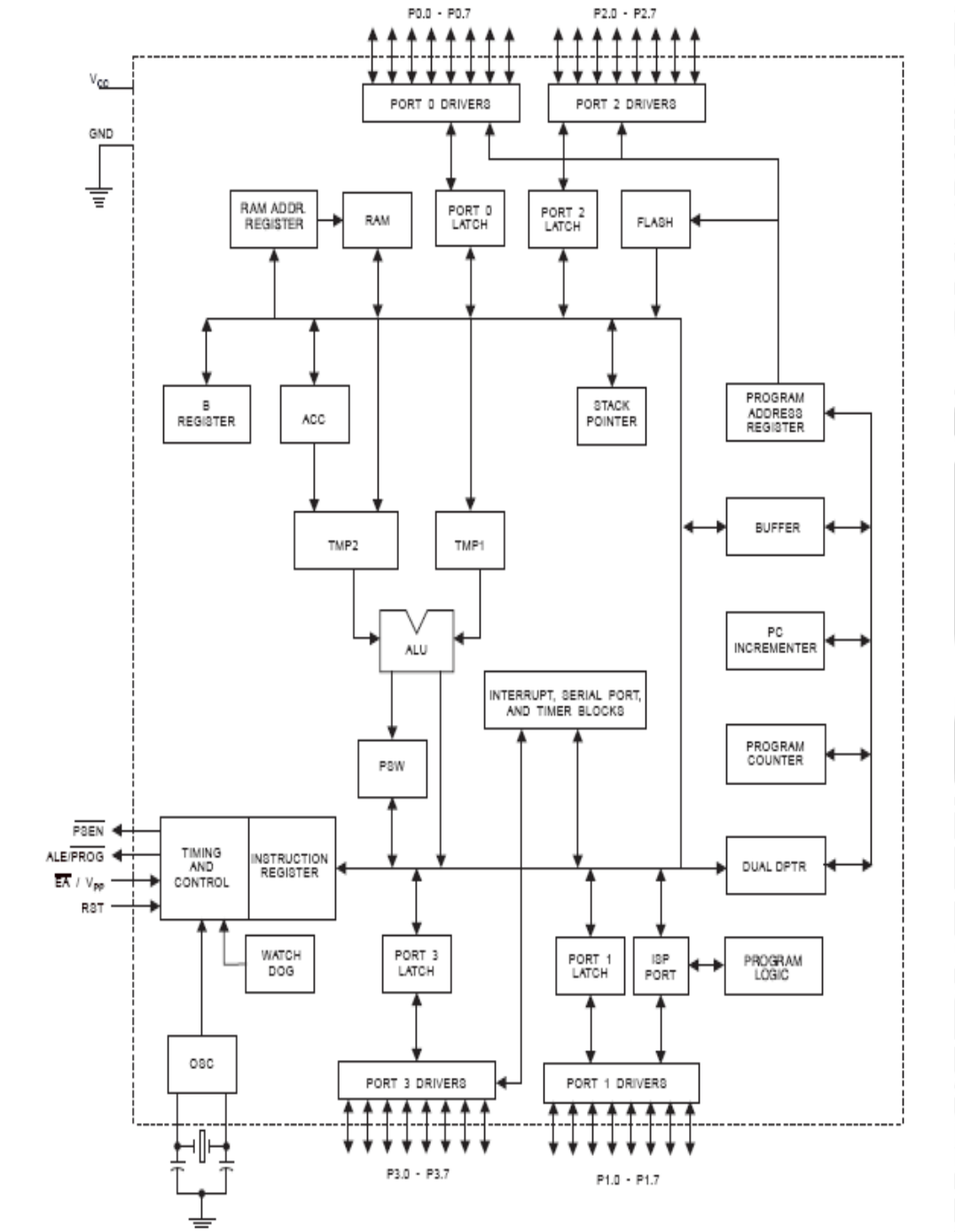
AT89S51 adalah mikrokontroler keluaran Atmel dengan 4K *byte* Flash PEROM *(Programmable and Erasable Read Only Memory)*, AT89S51 merupakan memori dengan teknologi *nonvolatile memory*, isi memori tersebut dapat diisi ulang ataupun dihapus berkali-kali.

Memori ini biasa digunakan untuk menyimpan instruksi (perintah) berstandar MCS-51 code sehingga memungkinkan mikrokontroler ini untuk bekerja dalam mode *single chip operation* (mode operasi keping tunggal) yang tidak memerlukan *external memory* (memori luar) untuk menyimpan source code tersebut.

Sebagai suatu sistem kontrol, mikrokontroler AT89S51 bila dibandingkan dengan mikroprosesor memiliki kemampuan dan segi ekonomis yang bisa diandalkan karena dalam mikrokontroler sudah terdapat RAM dan ROM sedangkan mikroprosesor didalamnya tidak terdapat keduanya. Secara umum konfigurasi yang dimiliki mikrokontroler AT89S51 adalah sebagai berikut :

- Sebuah CPU 8 bit dengan menggunakan teknologi dari Atmel.
- Memiliki memori baca-tulis (RAM) sebesar 128 *byte*.
- Empat buah *programmable port* I/O, masing-masing terdiri atas 8 buah jalur I/O.
- Sebuah *port* serial dengan kontrol *full duplex*.
- *Reprogrammable Flash memory* yang besarnya 4 kbyte untuk memori program.
- Mampu beroperasi sampai 33 MHz.

AT89S51 adalah mikrokontroler mempunyai kompabalitas instruksi dan konfigurasi pin dengan mikrokontroler MCS-51. Blok diagram MCS-51 ditunjukan dalam Gambar 2.1

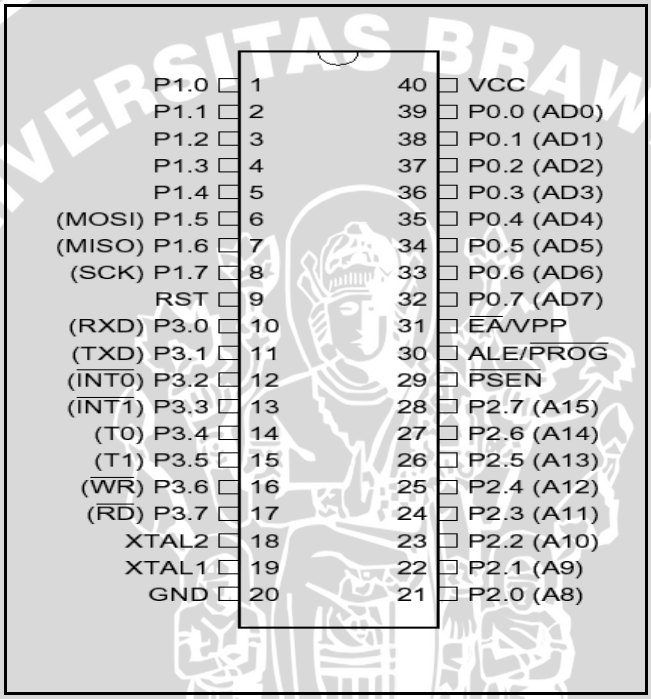


Gambar 2.1. Blok Diagram AT89S51

Sumber : Atmel, 1997:3

2.2.1 Konfigurasi Pin

Masing-masing kaki dalam mikrokontroler AT89S51 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler AT89S51, perancangan aplikasi mikrokontroler AT89S51 akan lebih mudah merencanakan dan membuat sistem yang dirancang. AT89S51 mempunyai 40 pin, susunan masing-masing pin dapat dilihat dalam Gambar 2.2



Gambar 2.2 Konfigurasi Pin AT89S51

Sumber : Atmel, 1997:2

Fungsi kaki-kaki AT89S51 adalah:

1. *Port* 1 (Pin 1..8), merupakan saluran masukan/keluaran dua arah.
2. Pin 9 RST, merupakan saluran dua masukan untuk mereset mikrokontroler dengan cara memberi masukan logika tinggi.
3. *Port* 3 (Pin 10..17), merupakan saluran masukan/keluaran dua arah dan mempunyai fungsi

khusus. Fungsi khusus meliputi TXD (*Transmit Data*), RXD (*Receive Data*), $\overline{INT0}$ (*Interupt 0*), $\overline{INT1}$ (*Interupt 1*), T0 (*Timer 0*), T1 (*Timer 1*), $\overline{WR}$ (*Write*), $\overline{RD}$ (*Read*).

4. Pin 18 dan 19 ($XTAL_1$ dan $XTAL_2$), merupakan saluran untuk mengatur pewaktuan sistem. Untuk pewaktuan dapat menggunakan pewaktuan internal maupun eksternal.
5. Pin 20 V_{SS}, merupakan hubungan ke *ground* dari rangkaian.
6. *Port* 2 (Pin S21..28), merupakan saluran masukan/keluaran dua arah.
7. Pin 29 $\overline{PSEN}$ (*Program Store Enable*), merupakan sinyal baca untuk mengaktifkan memori program eksternal.
8. Pin 30 *ALE* / $\overline{PROG}$ (*Address Latch Enable*), merupakan pulsa yang berfungsi untuk menahan alamat rendah (A0-A7) dalam *port* 0, selama proses baca/tulis memori eksternal. Frekuensi ALE adalah $1/6$ kali frekuensi osilator, dan dapat digunakan sebagai pewaktu. Pin ini juga berfungsi sebagai saluran program selama dilakukan pemrograman jika menggunakan memori program internal.
9. Pin 31 $\overline{EA}$ /VPP (*External Access Enable*), untuk mengatur penggunaan memori program eksternal dan internal. Pin ini harus dihubungkan dengan *ground* bila menggunakan memori program eksternal dan dihubungkan dengan VPP sebesar 5 volt jika menggunakan memori program internal.
10. *Port* 0 (Pin 32..39), merupakan saluran masukan/keluaran *open drain*.
11. Pin 40 V_{CC}, merupakan saluran masukan untuk catu daya positif sebesar 5 volt DC dengan toleransi lebih kurang 10%.

2.2.2 Struktur dan Operasi Port

Mikrokontroler AT89S51 memiliki 4 buah port. Setiap port memiliki 8 buah jalur I/O yang bersifat *bidirectional.* Beberapa karakteristik port mikrokontroler AT89S51 dijelaskan secara singkat sebagai berikut:

1. Port 0 merupakan port I/O 8-bit yang tidak mempunyai *pull-up* internal. Sebagai sebuah keluaran, maka setiap pin juga dapat mengendalikan 8 beban TTL. Port 0 juga dapat digunakan untuk memultipleks address bus rendah dan data memori dengan menggunakan *pull-up* internal. Selain itu, port 0 juga menerima kode mesin (dalam byte) selama pemrograman EPROM dan mengeluarkan kode mesin selama program verifikasi dari EPROM. Selama program verifikasi dibutuhkan *pull-up* eksternal. Pada port ini berlaku ketetapan yang berbeda dengan port-port lain, yaitu bila digunakan sebagai keluaran harus diberikan tambahan resistor *pull-u*p.
2. Port 1 merupakan sebuah port I/O *bidirectional* yang mempunyai *pull-up* internal. *Buffer* keluaran dari port 1 dapat mengendalikan 4 beban TTL. Pin-pin dari port 1 dapat juga digunakan sebagai masukan jika di *pull-up* tinggi oleh *pull-up* internal dan jika *pull-up low* internal. Port 1 juga menerima address bus rendah (dalam byte) selama pemrograman EPROM dan selama program verifikasi dari EPROM.
3. Port 2 dapat dipergunakan sebagai masukan atau keluaran seperti pada port 1. Alternatif lain dari port 2 dapat dipergunakan sebagai *address bus high byte* pada saat mengakses memori eksternal.
4. Port 3 merupakan sebuah port I/O 8-bit bidirectional yang mempunyai *pull-up* internal. *Buffer* keluaran dari port 3 dapat mengendalikan dan menghasilkan arus IIL karena

adanya *pull-up* internal. Port 3 juga mempunyai fungsi yang lain seperti yang tertera pada Tabel 2.2.

Tabel 2.2 Fungsi Alternatif Port 3

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Sumber: Atmel, 1997:4

2.2.3 Organisasi Memori

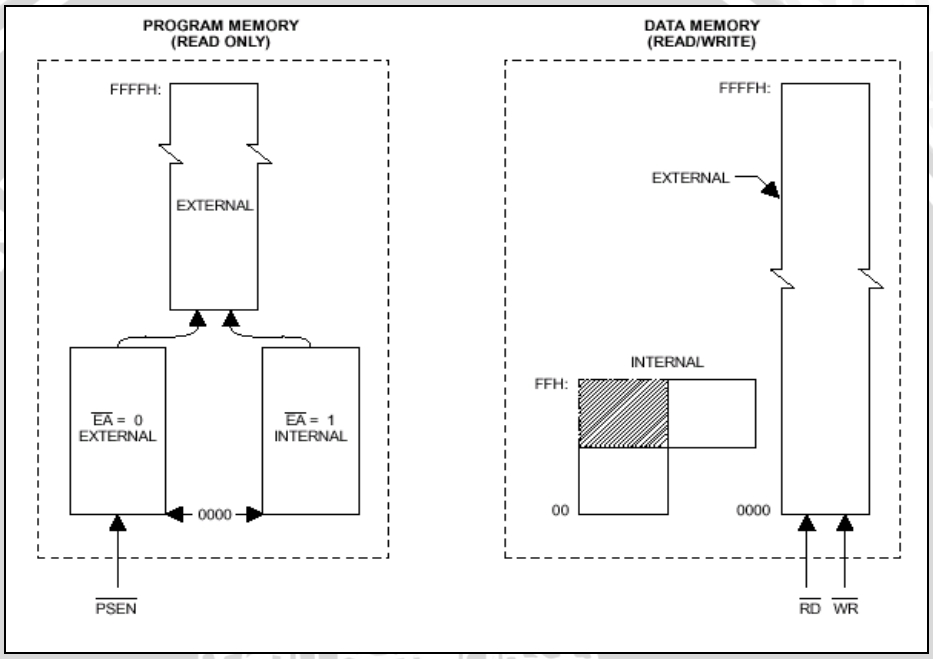
Mikrokontroler MCS-51 memiliki pembagian ruang alamat untuk program dan data. Memori data diakses oleh alamat 8 bit, tetapi alamat data 16 bit juga dapat dihasilkan mikrokontroler melalui register DPTR (*Data Pointer Register*). Alamat data dan program yang bisa dialamati oleh mikrokontroler adalah sebesar 64 *kilobyte* yaitu dari alamat 0000_H-$FFFF_H$.

$\overline{PSEN}$ adalah sinyal yang digunakan untuk pembacaan memori program eksternal. Mikrokontroler MCS-51 mempunyai dua buah alternatif untuk pembacaan memori program yaitu internal dan eksternal. Pembacaan memori program eksternal dengan mengatur pin $\overline{EA}$ pada logika 0 dan pembacaan memori program internal pin $\overline{EA}$ diset pada logika 1.

Mikrokontroler AT89S51 memiliki *RAM* internal 128 *byte* (00_H-$7F_H$) yang dapat digunakan untuk menampung data-data

yang diperlukan dalam pemrograman. *RAM* internal tersebut dapat diklasifikasikan sebagai berikut: 80 *byte general purpose* (30_H-$7F_H$), 32 *byte* (00_H – $1F_H$) sebagai *register bank* yang dapat dimanfaatkan seperti *RAM* biasa, dan 16 *byte* (20_H-$2F_H$) *bit addresable*.

Gambar 2.3 memperlihatkan struktur memori MCS-51. *ROM* internal mikrokontroler AT89S51 jenis *flash EEPROM* sebesar 4 *kilobyte* dapat diprogram ulang sebanyak 1000 kali.

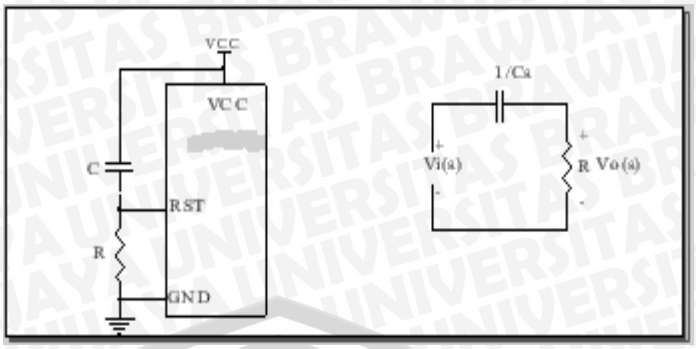


Gambar 2.3 Struktur Memori MCS-51

Sumber : AT89C51 Architectural Overview, 1997:2-4

2.2.4 Reset

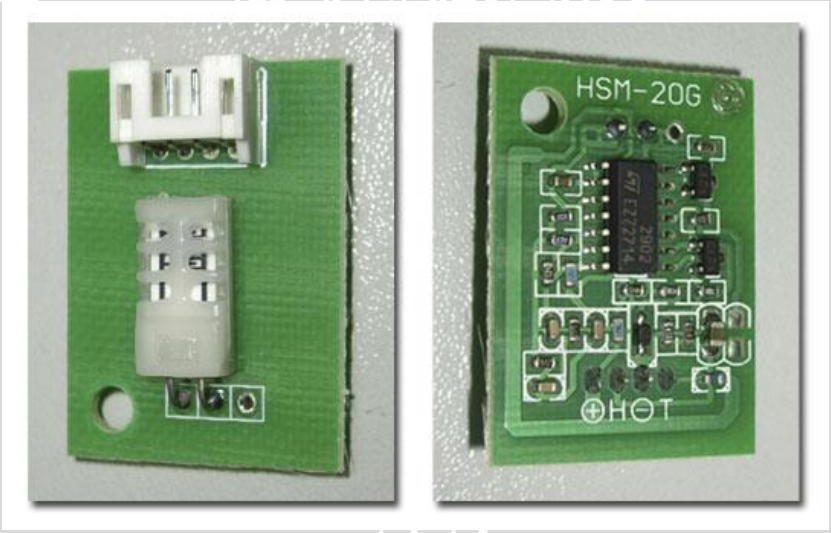
Rangkaian *reset* dibutuhkan untuk me-*reset* mikrokontroler pada saat *power on*. Tegangan berlogika tinggi selama 2 siklus mesin dibutuhkan untuk me-*reset* MCU pada saat dihidupkan. Rangkaian *reset* terdiri dari resistor dan kapasitor yang dihubungkan seperti dalam Gambar 2.4(a).



Gambar 2.4 (a) Rangkaian Power On Reset
(b) Rangkaian Ekivalen Power On Reset
Sumber : Atmel, 1997: 2-63

2.3 Sensor Suhu dan Kelembaban HSM20G

Sensor HSM- 20G adalah sensor pengukur kelembaban dan temperatur. Dimana wujud dari sensor HSM20G tersebut seperti Gambar 2.5



Gambar 2.5. Sensor Suhu dan Kelembaban HSM20G
Sumber :
http://www.imariscal.es/user_upload/Image/Arduino/hsm20g.jpg

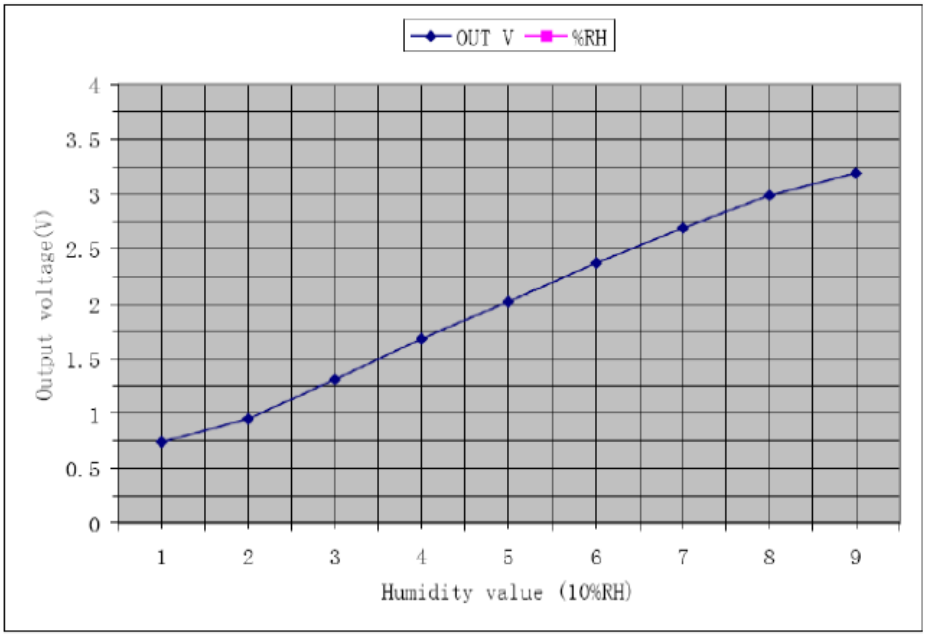
Dengan sensor suhu dan kelembaban HSM20G, kelembaban relatif bisa dikonversi ke tegangan yang standart. Macam-macam dari jenis aplikasi yang dapat digunakan oleh sensor ini adalah lembab, dan sangat lembab, untuk AC, data *loggers* kelembaban, *automotive climate control*, dll.

Spesifikasi dari sensor suhu dan kelembaban HSM20G dapat dilihat pada Tabel 2.3.

Tabel 2.3 Spesifikasi HSM20G

Characteristics	HSM20G
Input voltage range	DC 5.0±0.2 V
Output voltage range	DC 1.0-3.0 V
Measurement Accuracy	±5% RH (*Relative Humidity*)
Operating Current (Maximum)	2 mA
Storage RH range	0 to 99% RH(*Relative Humidity*)
Operating RH range	20 to 95% (100% RH intermittent)
Transient Condensation	<3% RH(*Relative Humidity*)
Storage temperatur range	-20^0C to 70^0C
Operating temperatur range	0^0C to 50^0C
Hysterisis (RH @25^0C)	MAX 2% RH(*Relative Humidity*)
Long term stability (typical drift per year)	±1.5%
Linearity	Linearity
Time response (63% step change)	1 min
Dimensions(L*W)	34mm*22mm

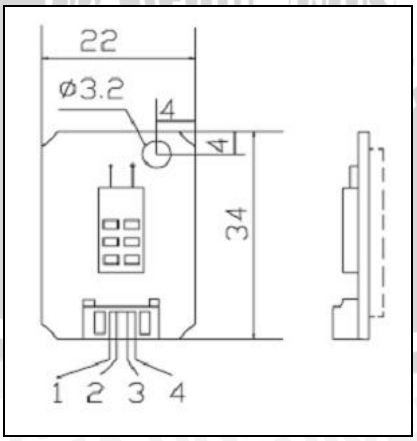
Sumber : http://www.szgeha.com/uploadfile/product/2008119154715655.pdf, 2008



Gambar 2.6 Grafik Kurva Respon HSM -20G pada 25^0C (3)

Sumber : Datasheet HSM-20G

Pada Gambar 2.6 di atas dapat dilihat jelas bagaimana hubungan antara nilai kelembaban dan tegangan keluaran yang membentuk garis linier. Hal ini dikarenakan kelembaban berbanding lurus dengan tegangan keluaran.Pada tabel 2.3 di atas dapat dilihat range atau batas untuk nilai kelembaban pada sensor ini sebagaimana terlihat bahwa nilai tegangan berbanding lurus dengan presentase kelembaban. Nilai yang tertera diatas menunjukkan bahwa nilai batas kelembaban maksimum 90%RH dan batas minimum 10%RH dengan tegangan 0, 74 volt dan maksimal 3,19 volt



Gambar 2.7 Dimensi Sensor Suhu dan Kelembaban HSM20G

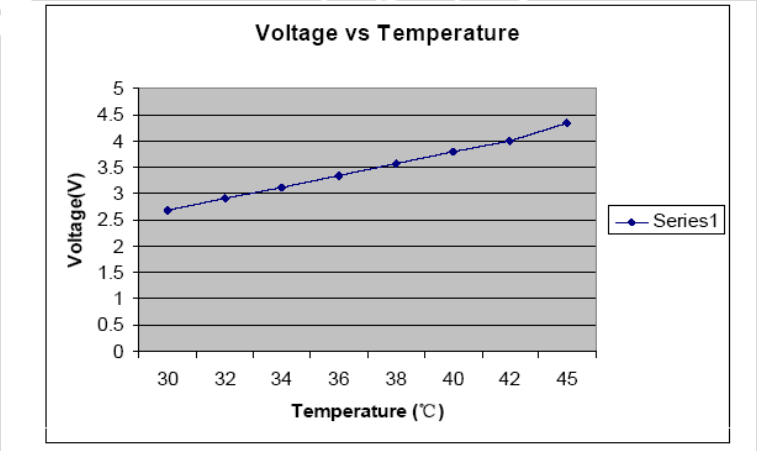
Sumber : Datasheet HSM-20G

Tabel 2.4 Pin Sensor Suhu dan Kelembaban HSM20G

Pin	Fungsi
1	Output temperature
2	GND
3	Output Kelembaban
4	Vcc (+ 5. 0 V)

Sumber : Datasheet HSM-20G

Dari Gambar 2.7 dan Tabel 2.4 dapat dilihat konfigurasi dari sensor suhu dan kelembaban HSM20G, dimana pada setiap kaki pada sensor suhu dan kelembaban HSM20G mempunyai fungsi yang berbeda-beda untuk dihubungkan antara satu dengan yang lainnya. Pada sensor ini, ada 4 kaki yaitu untuk kelembaban , temperature, ground, dan juga Vcc.



Gambar 2.8 Grafik Kurva Respon HSM -20G pada 25^0C (3)

Sumber : Datasheet HSM-20G

2.4 Pengkondisi sinyal

Rangkaian pengkondisi sinyal dibentuk berdasarkan hubungan yang linier antara tegangan keluaran dan tegangan masukan rangkaian. Hubungan ini dapat dinyatakan dalam bentuk persamaan garis lurus, yaitu :

$$V_{OUT} = m \cdot V_{IN} + V_O \qquad (2\text{-}1)$$

Dengan :

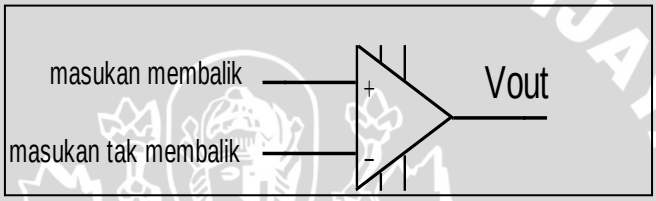
m = kemiringan garis yang menyatakan penguatan

V_O = tegangan *offset* keluaran

2.4.1 Penguat Operasional

Keluaran suatu rangkaian sebelum masuk ke rangkaian berikutnya jika sinyalnya masih kecil membutuhkan suatu penguat. Peralatan elektronika yang sering dimanfaatkan sebagai penguat adalah penguat operasional. Karakteristik op-amp yang terpenting adalah :

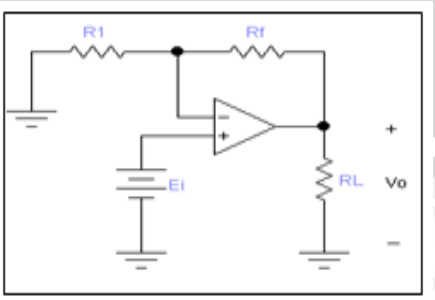
- Impedansi masukan amat tinggi, sehingga arus masukan dapat diabaikan.
- Penguatan tinggi.
- Impedansi keluaran rendah, sehingga keluaran penguat tidak terpengaruh oleh pembebanan.



Gambar 2.9 Penguat Op-Amp
Sumber : *Coughlin, 1982*

2.4.2 Penguat Tak Membalik (*Non Inverting Amplifier*)

Penguat umpan balik tegangan tak membalik adalah penguat tegangan yang mendekati ideal karena impedansi masukan tinggi, impedansi keluarannya rendah dan bati tegangannya yang mantap. Pada penguat jenis ini sinyal masuk menggerakkan masukan tak membalik dari penguat, sebagian dari tegangan keluar kemudian dicuplik dan diumpankan kembali ke masukan membalik.



Gambar 2.10 Penguat Tak Membalik
Sumber : *Coughlin, 1982*

Tegangan keluaran Vo mempunyai polaritas yang sama dengan tegangan masukan Ei. Tahanan masukan dari masukan pembalik adalah Ri tetapi tahanan masukan dari masukan tak membalik adalah besar, biasanya melebihi 100 MΩ. karena tegangan antara masukan (+) dan masukan (-) secara praktis adalah 0 maka kedua masukan itu berada pada potensial yang sama yaitu Ei. Karena Ei melintasi R1 dan Ei menyebabkan arus I mengalir seperti diberikan oleh

$$I = \frac{Ei}{R1} \tag{2-2}$$

Arah I tergantung pada polaritas Ei, arus yang mengalir lewat masukan (-) diabaikan karenanya I mengalir melalui Rf dan penurunan tegangan melintasi Rf dinyatakan oleh VRi dan dinyatakan sebagai

$$VRf = I.(Rf) = \frac{Rf}{R1} xEi \tag{2-3}$$

Maka tegangan keluaran Vo adalah

$$Vo = Ei + \frac{Rf}{R1}.Ei \tag{2-4}$$

$$Vo = \left(1 + \frac{Rf}{R1}\right).Ei \tag{2-5}$$

dengan begitu gain tegangannya yaitu

$$A_{CL} = \frac{Vo}{Ei} = 1 + \frac{Rf}{R1} \tag{2-6}$$

2.5 Penambah Kelembaban (*Humidifier*)

Penambah kelembaban berfungsi untuk memasukkan uap air dari wadah bejana ke dalam mesin penetasan agar kandungan uap air dapat ditambah sesuai yang diinginkan. Proses penetasan telur ayam dengan pemanas bersuhu tinggi membuat kandungan uap air menjadi kering sehingga diperlukan sebuah wadah untuk menghasilkan uap air dan kipas untuk mengalirkan uap air yang dihasilkan ke dalam mesin penetas.

Arah putaran kipas pada mesin penetas ayam ini dirancang sedemikian rupa agar dapat mengalirkan udara keluar mesin penetas. Sehingga udara

jenuh di dalam ruang penetas dapat dialirkan keluar. Dengan membuang udara jenuh tersebut dan memasukkan uap air diharapkan tingkat kelembaban udara di dalam mesin penetas tetap terjaga sehingga proses penetasan telur ayam dapat berlangsung baik. Aktifnya kipas ini juga diikuti dengan membukanya jendela pada mesin penetas.

2.6 LCD

Untuk menampilkan menu dari suatu peralatan elektronika seperti catu daya *switching* misalnya tentang berapa tegangan keluaran yang diinginkan, maka diperlukan suatu tampilan. Yang lebih sering digunakan adalah LCD.

Liquid crystal display (LCD) merupakan komponen elektronika yang digunakan untuk menampilkan suatu karakter baik itu angka, huruf atau karakter tertentu, sehingga tampilan tersebut dapat dilihat secara *visual*. Pemakaian LCD sebagai indikator tampilan banyak digunakan disebabkan daya yang dibutuhkan LCD relatif kecil (orde mikrowatt), di samping itu dapat juga menampilkan angka, huruf atau simbol dan karakter tertentu. Meskipun pada komponen ini dibatasi oleh sumber cahaya eksternal/internal, suhu, dan *lifetime*.

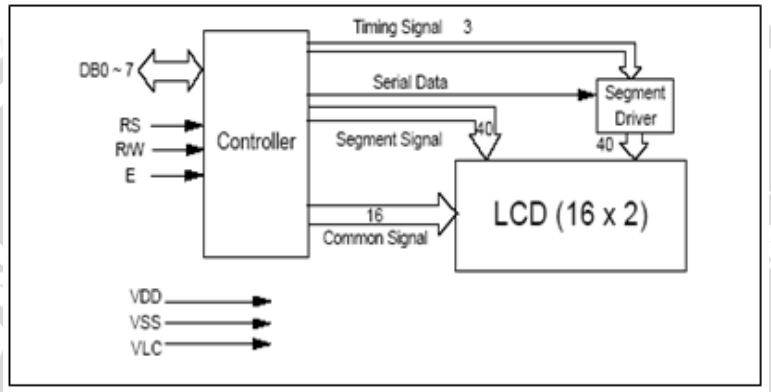
LCD terdiri atas tumpukan tipis atau sel dari dua lembar kaca dengan pinggiran tertutup rapat. Antara dua lembar kaca tersebut diberi bahan kristal cair (*liquid crystal*) yang tembus cahaya. Permukaan luar masing-masing keping kaca mempunyai lapisan tembus cahaya seperti oksida timah (*tin oxide*) atau oksida indium (*indium oxide*). Sel mempunyai ketebalan 1x10-5 meter dan diisi dengan kristal cair.

LCD yang digunakan disini untuk menampilkan perintah-perintah dalam tahapan pengisian bahan bakar secara mandiri. LCD yang akan digunakan bertipe M1632 produksi SEIKO instrument inc. corporation. Spesifikasi dari LCD ini adalah sebagai berikut:

- Menampilkan 16 karakter pada tiap baris TN LCD dengan 5 x 7 dot matrik
- Pembangkit karakter ROM untuk 192 jenis karakter
- Pembangkit karakter RAM untuk 8 jenis karakter
- 80 x 8 bit data RAM

- Tegangan catu 5 volt dan temperatur operasi 0–50°C
- Otomatis reset pada saat dihidupkan

Masukan yang diperlukan untuk mengendalikan modul LCD ini berupa bus data yang masih termultipleks dengan bus alamat serta 3 bit sinyal kontrol, yaitu RS, R/W dan E. Sementara pengendali dot matrix LCD dilakukan secara internal oleh kontroler yang sudah terpasang pada modul LCD. Gambar 2.11 adalah diagram blok dari LCD M1632.



Gambar 2.11 Diagram Blok LCD M1632.

Sumber : *Anonymous, 2001:3*

Fungsi pin dari LCD M1632 ditunjukkan dalam Tabel 2.5.

Tabel 2.5 Fungsi Pin-Pin pada LCD M1632

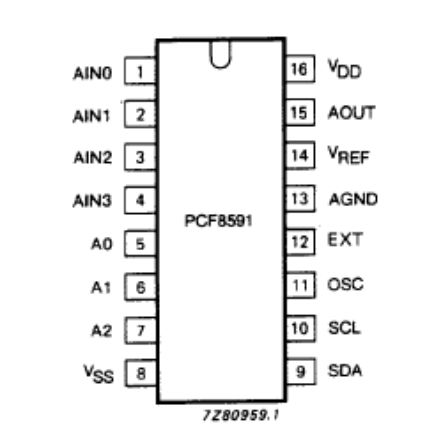
No Pin	Nama Pin	Fungsi
16	V – BL	Sebagai *ground* dari *backlight*
15	V + BL	Sebagai kutub positif dari *backlight*
7 – 14	DB0– DB7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan.
6	E	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca
5	R/W	Sinyal seleksi tulis atau baca 0 = tulis 1 = baca
4	RS	Sinyal pemilih register 0 = register instruksi (tulis)

		1 = register data (tulis dan baca)
3	Vlc	Untuk mengendalikan kecerahan LCD dengan mengubah Vlc
2	Vcc	Tegangan catu + 5 volt
1	Vss	Terminal *Ground*

Sumber : Datasheet LCD M1632

2.7 ADC PCF 8591 (Analog to Digital Converter)

Analog to Digital Converter atau lebih dikenal dengan ADC adalah sebuah piranti yang dirancang untuk mengubah sinyal-sinyal analog menjadi sinyal-sinyal digital. ADDA merupakan *Analog Input Output add-on board* untuk mikrokontroler AT89S51 *Development Tools* DT51 yang menggunakan I2C-bus. DT51 I2C ADDA digunakan untuk mengubah sinyal analog menjadi digital atau sebaliknya. ADDA PCF8591 terdiri atas 16 pin yang ditunjukkan dalam Gambar 2.12 dan fungsi masing-masing pin ditunjukkan dalam Tabel 2.6.



Gambar 2.12 Susuna Pin ADDA PCF 8591

Sumber : Philips Semiconductor, 1998: 5

Spesifikasi yang dimiliki oleh DT51 I2C ADDA adalah:

1. Kompatibel penuh dengan mikrokontroler AT89S51 *development Tools* DT51.
2. Hanya perlu dua jalur kabel untuk antarmuka dengan mikrokontroler.

3. Analog input 4 chanel 8 bit.
4. Analog output 1 chanel 8 bit.
5. Input range tegangan 0 V – 2,5 V.
6. Spesifikasi ADDA PCF8591 adalah 10mV/bit

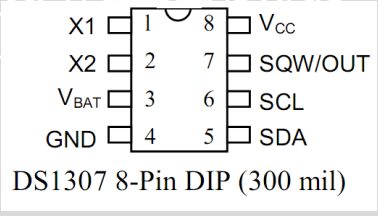
Tabel 2.6 Fungsi Pin ADDA PCF8591

SYMBOL	PIN	DESCRIPTION
AINO	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware address
A1	6	
A2	7	
V_{SS}	8	negative supply voltage
SDA	9	I²C-bus data input/output
SCL	10	I²C-bus clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground
V_{REF}	14	voltage reference input
AOUT	15	analog output (D/A converter)
V_{DD}	16	positive supply voltage

Sumber : Philips Semiconductor, 1998: 5

2.8 RTC

DS1307 merupakan sebuah IC RTC (*Real Time Clock*) yang dapat merekam dan memberikan informasi waktu lengkap mulai informasi detik, menit, jam, tanggal, bulan hingga tahun. Susunan pin IC RTC DS1307 ditunjukkan dalam Gambar 2.13.



Gambar 2.13 Susunan pin DS1307

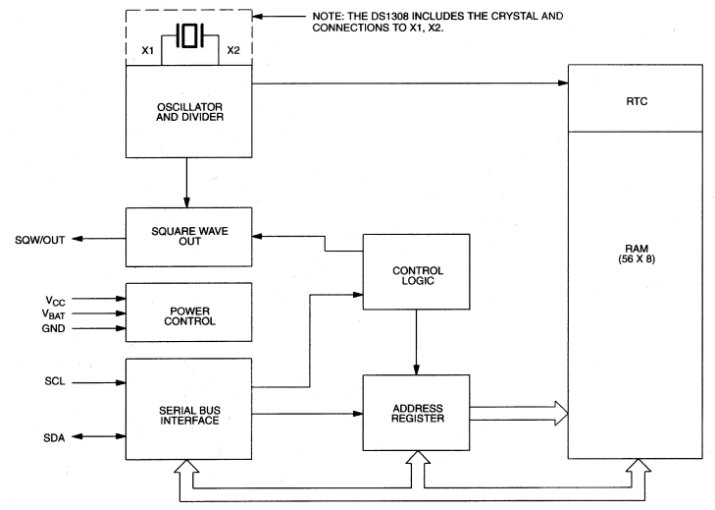
Sumber : Dallas, 2000: 1

Berbagai fitur yang disediakan chip ini antara lain:

- Perhitungan *real time clock* mulai dari detik menit, jam, tanggal, bulan hingga tahun
- RAM untuk penyimpanan data sebesar 56 byte

- *Interface* I2C yang dikembangkan oleh Philips
- Dapat memberikan *output* berupa sinyal pulsa yang dapat diprogram
- Mengkonsumsi arus kurang dari 500 nA dalam mode baterai dengan *oscillator* yang tetap berjalan.

Diagram blok DS1307 ditunjukkan dalam Gambar 2.14.



Gambar 2.14 Diagram blok DS1307

Sumber : Dallas, 2000: 2

X_1 dan X_2 dihubungkan dengan X_{tal} berukuran 32,768 kHz. V_{bat} dihubungkan dengan catu daya baterai sebesar 3 V. Jenis komunikasi yang digunakan oleh DS1307 adalah komunikasi serial I2C yang dikembangkan oleh Philips. Komunikasi serial I2C menggunakan dua jalur transmisi, yaitu SCL yang berfungsi sebagai jalur *clock* dan SDA yang berfungsi sebagai jalur data.

Informasi waktu dapat ditulis dan dibaca menggunakan mikrokontroler. Informasi waktu diformat dalam bentuk 2 byte bilangan BCD yang disimpan dalam register-register berbeda. Tabel 2.7 menunjukkan alamat dan isi register dalam DS1307.

Tabel 2.7 Alamat dan isi register DS1307

	BIT7							BIT0	
00H	CH	10 SECONDS			SECONDS				00–59
	X	10 MINUTES			MINUTES				00–59
	X	12/24	10 HR A/P	10 HR	HOURS				01–12 00–23
	X	X	X	X	X	DAY			1–7
	X	X	10 DATE		DATE				01–28/29 01–30 01–31
	X	X	X	10 MONTH	MONTH				01–12
	10 YEAR				YEAR				00–99
07H	OUT	X	X	SQWE	X	X	RS1	RS0	

Sumber : Dallas, 2000: 4

2.9 Motor Listrik

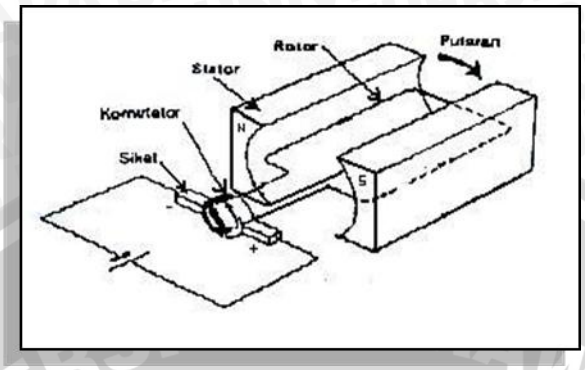
Motor listrik merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor listrik biasanya digunakan untuk, menggerakkan kompresor, kipas angin, mengangkat beban dan untuk keperluan lain-lain. Motor listrik terbagi menjadi menjadi 2, yaitu motor arus bolak-balik (AC) dan motor listrik arus searah (DC).

2.9.1 Motor DC

Pengertian motor arus searah adalah suatu mesin yang berfungsi mengubah tenaga listrik arus searah menjadi tenaga gerak atau tenaga mekanik, di mana gerak tersebut berupa putaran dari pada rotor.

Motor DC hampir sama konstruksinya dengan motor AC, perbedaannya terletak pada sikat dan cincin belah (komutator). Saat siklus pertama, arus mengalir dari kutub positif ke negatif. Aliran arus yang melewati bagian kabel yang berada didekat kutub N magnet akan menimbulkan gaya Lorentz ke bawah. Sementara itu aliran arus yang melewati kabel yang berada di dekat kutub S magnet akan menyebabkan gaya Lorentz ke atas. Kedua perpaduan gaya Lorentz tersebut akan menyebabkan kawat berputar. Pada siklus berikutnya terjadi hal yang serupa seperti pada siklus sebelumnya. Apabila arus terus-menerus dialirkan, maka kawat akan berputar secara terus menerus pula.

Pada aplikasi sesungguhnya, kawat adalah sebuah rotor yang akan dikopel dengan sebuah as dan akan memutar as tersebut terus menerus seiring perputaran motor. Motor DC ditunjukkan dalam Gambar 2.15.



Gambar 2.15 Motor DC.

Sumber : *Kismet F [1994:98]*

2.10 Relay

Relay adalah komponen elektronika yang terdiri dari sebuah lilitan kawat (kumparan/koil) yang terlilit pada suatu inti besi lunak. Jika kumparan dialiri oleh arus listrik, maka inti besi akan menjadi magnet. *Relay* merupakan alat untuk membuka dan menutup kontak secara elektrik dengan tujuan menghubungkan fungsi dari rangkaian satu dengan rangkaian yang lainnya.

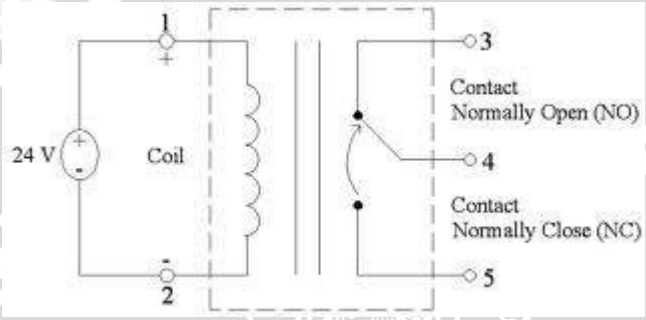
Kontak-kontak yang ada pada relay ada dua macam, yaitu *Normally Open* (relay yang kontaknya terbuka pada saat belum ada arus yang melalui kumparan dan tertutup pada saat ada arus) dan *Normally Closed* (relay yang kontaknya tertutup pada saat belum ada arus yang melalui kumparan dan terbuka pada saat ada arus).

Keuntungan memakai *relay* umumnya terletak pada pengaturan *switching*-nya, sehingga terjadi isolasi antara rangkaian catu daya rendah dengan catu daya beban yang tinggi yang akan diputus dan disambung. Kerugian relay umumnya terjadi tanggapan waktu yang *relative* lambat saat ON/OFF. Cara kerja relay adalah sebagai berikut ini :

Jika ada arus yang masuk melalui kumparan, maka pada kumparan tersebut akan menghasilkan induksi magnet. Induksi magnet tersebut akan

menarik pegas kontak, yang akan merubah posisi awalnya menjadi terhubung ke bagian yang lain. Setelah arus terhenti maka tidak ada induksi sehingga kontak kembali pada kondisi semula.

Relay pada Gambar 2.16 adalah komponen elektronika yang terdiri dari sebuah lilitan kawat (kumparan/koil) yang terlilit pada sebuah inti besi lunak. Jika kumparan dialiri arus listrik, maka inti besi akan menjadi magnet dan menarik kontak pegas sehingga kontak AB terhubung dan BC terputus, begitu juga sebaliknya.



Gambar 2.16 Rangkaian Relay

Sumber : http://kpelektro05.wordpress.com/

Hal-hal yang perlu diketahui dari relay :

1. Tahanan kumparan : besarnya tahanan kumparan ditentukan oleh tebal kawat yang dipakai dan banyaknya lilitan
2. Kuat arus yang diperlukan untuk menggerakan relay dengan perlawanan kecil menggunakan arus yang besar, dan relay dengan perlawanan besar akan memerlukan arus yang kecil
3. Tegangan untuk menggerakan relay
4. Daya yang dipakai oleh relay
5. Banyaknya kontak pada relay

Relay merupakan suatu alat untuk menghubungkan atau memutuskan kontak antara komponen yang satu dengan komponen yang lain. Dalam memutus atau menghubungkan kontak digerakkan oleh fluksi yang timbul akibat adanya magnet listrik. Jadi fluksi inilah yang menghubungkan dan memutuskan kontak dan antara kumparan dengan bagian saklar tidak ada hubungan listrik.

Ada beberapa jenis susunan kontak relay pada Gambar 2.17 dimana semuanya terisolasi terhadap arus listrik yang ada didalam kumparan.

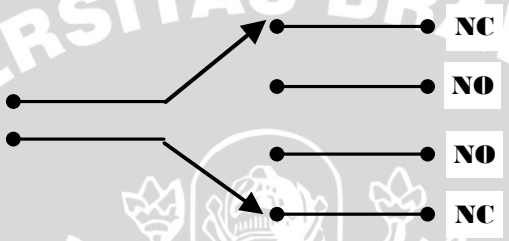
Jenis susunan kontak sebagai berikut :

- *Normally Open* (Normal terbuka)

  Yaitu kontak-kontak tertutup pada saat kumparan relay dialiri arus listrik.

- *Normally Close* (Normal tertutup)

  Yaitu kontak-kontak terbuka pada saat kumparan relay dialiri arus listrik



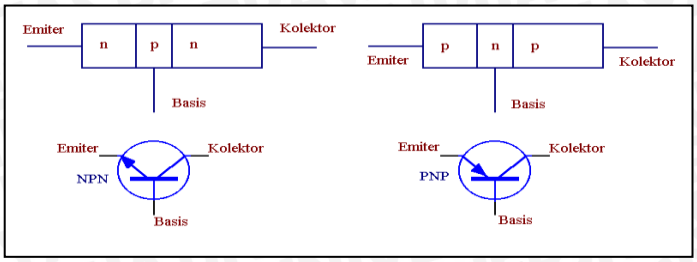
Gambar 2.17 Relay DPDT

Sumber : http://kpelektro05.wordpress.com/

2.11 Transistor

Transistor merupakan salah satu komponen elektronika dengan terminal yang terdiri dari kolektor (C), Basis (B) dan Emiter (E). Setelah bahan semikonduktor dasar diolah, terbentuklah bahan semikonduktor jenis P dan N. walaupun proses pembuatannya banyak, pada dasarnya transistor merupakan tiga lapis gabungan kedua jenis bahan tadi, yaitu NPN dan PNP.

Simbol sirkuit kedua jenis bahan tersebut hampir sama, perbedaannya hanya terletak pada arah panah di ujung emiter. Arah panah ini menunjukkan arah aliran arus konvensional yang berlawanan arah dalam kedua jenis tadi. Adapun lambang komponen transistor NPN maupun PNP dapat dilihat pada Gambar 2.18.



Gambar 2.18 Simbol Transistor

Sumber : *Perancangan*

2.11.1 Persamaan Transistor

Sesuai dengan hukum arus Kirchoff, jumlah arus yang masuk pada titik atau sambungan sama dengan jumlah arus yang keluar dari titik atau sambungan itu. Jika diterapkan pada transistor maka,

$$I_e = I_b + I_c \qquad (2\text{-}7)$$

Persamaan tersebut mengatakan bahwa arus emiter (I_e) merupakan jumlah arus kolektor (I_c) dan arus basis (I_b). Karena arus basis sangat kecil, arus emiter kira-kira sama dengan arus kolektor. βdc didefinisikan sebagai rasio arus kolektor DC dengan arus basis DC.

$$\beta dc = \frac{Ic}{Ib} \qquad (2\text{-}8)$$

βdc juga dikenal sebagai gain arus. Karena arus basis yang kecil dapat menghasilkan arus kolektor yang lebih besar. Penguatan arus adalah keuntungan utama sebuah transistor dan telah dipakai pada banyak aplikasi. Untuk transistor daya rendah (dibawah 1 W), gain arus biasanya 100 – 300. Transistor daya tinggi (diatas 1 W) biasanya memiliki gain arus 20 -100

2.11.2 Garis Beban DC

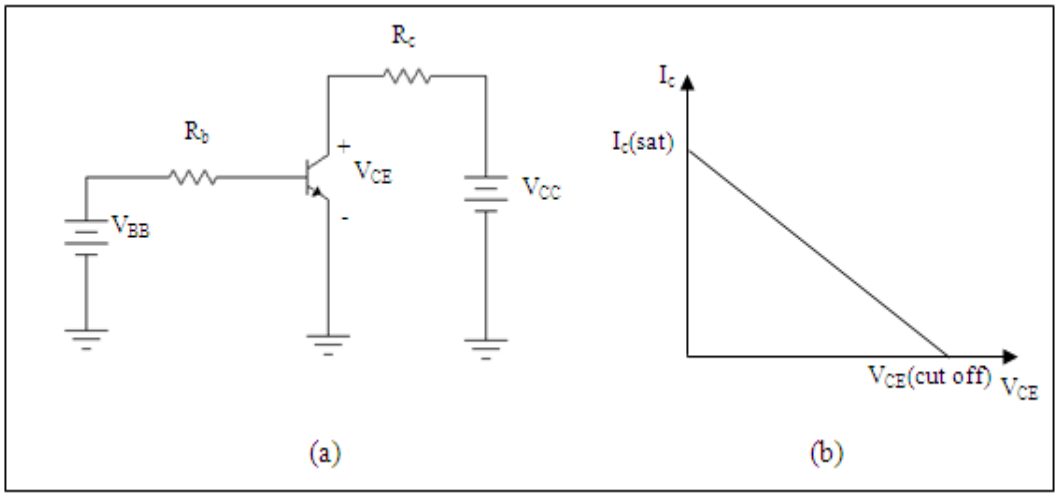
Rangkaian dalam Gambar 2.19a. merupakan contoh bias basis, yang berarti mengatur nilai tetap arus basis. Dari rangkaian dalam Gambar 2.19, besar tegangan kolektor-emiter (V_{CE}) ditentukan dengan:

$$Vce = Vcc - IcRc \quad (2\text{-}9)$$

Penyelesaian Ic akan menghasilkan :

$$Ic = \frac{Vcc - Vce}{Rc} \quad (2\text{-}10)$$

Jika menggambarkan Persamaan 2-10 (Ic terhadap VCE), akan mendapatkan sebuah garis lurus. Garis ini dinamakan garis beban karena menggambarkan pengaruh beban pada Ic dan VCE. Persamaan 2-10 merupakan persamaan linear, sehingga grafiknya merupakan garis lurus. Gambar garis beban ditunjukkan dalam Gambar 2.19.



Gambar 2.19 (a) Bias basis transistor, (b) Garis beban DC

Sumber : Malvino, 1999 : 235

Perpotongan garis beban dengan Ic disebut penjenuhan. Pada titik ini, arus kolektor adalah maksimum. Titik ini disebut juga titik jenuh (*saturation*). Titik jenuh menyatakan arus maksimum kolektor yang mungkin pada rangkaian. Arus kolektor penjenuhan adalah:

$$Ic(sat) = \frac{Vcc}{Rc} \quad (2\text{-}11)$$

Titik *cut off* adalah titik dimana garis beban berpotongan dengan $V_{CE.}$ Titik *cut off* menyatakan tegangan kolektor-emiter maksimal yang mungkin dalam rangkaian. Besar V_{CE}(*cut off*) adalah sama dengan V_{CC}.

2.11.3 Transistor Sebagai Saklar

Suatu transistor dalam sistem kerjanya mengalami keadaan saturasi dan titik sumbat. Dengan demikian jika transistor digunakan sebagai saklar maka hubungan saklar ini terjadi antara kolektor dan emitor dari transistor tersebut. Sedangkan buka dan tutup saklar itu ditentukan oleh suatu tegangan yang diberikan pada basisnya .

- Saklar terbuka

  Jika V_B = 0V, Ib = 0V, maka transistor dalam keadaan titik sumbat sehingga tidak ada arus yang mengalir dan saklar dalam keadaan terbuka.

- Saklar tertutup

  Untuk mengembalikan saklar dalam keadaan tertutup maka harus diberikan Vinput pada V_B dan arus pada basis, sehingga keadaan transistor menjadi saturasi, dimana perpotongan dari garis beban dan kurva $I_B = I_{B(sat)}$. Keadaan ini seperti sebuah switch yang tertutup.

  Jika transistor dikerjakan pada keadaan saturasi (Vce =0,7v), maka antara kolektor dan emiter akan terhubung dengan cepat atau dengan kata lain saklar dalam keadaan tertutup. Adapun besarnya arus basis yang diperlukan untuk rangkaian ini dapat dicari melalui persamaan :

$$Ib = \frac{Ic}{\beta}$$

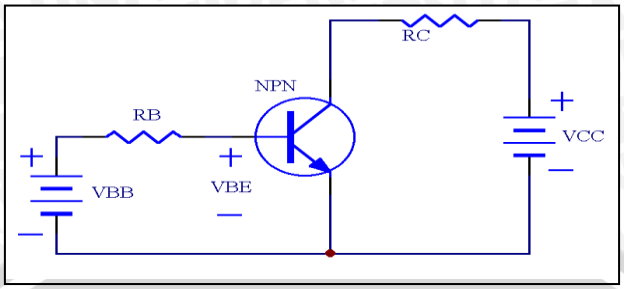
  Maka nilai Rb didapatkan sebagai berikut :

$$Rb = \frac{Vbb - Vbe}{Ib}$$

  Jika arus kolektor yang mengalir terlalu banyak, transistor akan menuju ke kondisi jenuh. Secara ideal terjadi suatu hubung singkat antara terminal kolektor dan emitor, dengan arus kolektor jenuh sebesar (Malvino, 1991).

$$I_{C(Sat)} \cong \frac{V_{CC}}{R_L}$$

Rangkaian transistor switch dapat dilihat pada Gambar 2.20.



Gambar 2.20 Transistor sebagai saklar

Sumber : *Perancangan*

2.12 Buzzer

Buzzer yang berukuran kecil ini menghasilkan suara satu nada yang berfrekuensi sangat tinggi. *Buzzer* yang diperlihatkan dalam Gambar 2.21 beroperasi dengan tegangan 3-16 V dan hanya membutuhkan arus sebesar 5-7 mA.

Gambar 2.21 Buzzer

Sumber : http://www.mutr.co.uk/

Buzzer ini digunakan sebagai tanda peringatan atau sebagai alarm. Intensitas suara yang dihasilkan berkisar 100 dB hingga 110 dB. Untuk mendapatkan tingkat kekerasan yang maksimum, harus dipasang kokoh didalam sebuah badan pembungkus atau pada sebuah papan rangkaian.

Sebuah *sounder* terdiri dari sebuah piringan logam yang dilapisi oleh oleh bahan piezo-elektris. *Buzzer* dapat digunakan untuk menghasilkan berbagai jenis bunyi-bunyian pemberi tanda peringatan. Untuk menghasilkan efek suara maksimum sounder harus dipasang secara kokoh

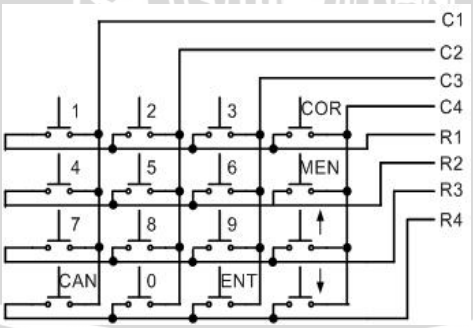
didalam sebuah badan pembungkus atau pada sebuah papan rangkaian, dan digerakan oleh sinyal dengan frekuensi yang mendekati resonansinya.

2.13 Heater (Pemanas)

Pemanas yang digunakan dalam perancangan alat penetas ini menggunakan elemen pemanas seperti yang digunakan pada solder. Panas yang dihasilkan oleh elemen pemanas cukup untuk memanaskan udara dalam mesin penetas yang hanya membutuhkan suhu antara 38,5-40,5°C. Elemen pemanas ini terbuat dari kawat nikrom yang dililit memutar dimana setiap panjang tertentu menentukan daya listrik yang diperlukan. Kawat nikrom merupakan campuran dari dua logam nikel dan kromium. Kelebihan dari kawat nikrom antara lain :

1. Kawat nikrom adalah konduktor listrik yang lemah apabila dibandingkan dengan kawat tembaga. Hasilnya, kawat ini memiliki resistansi yang cukup untuk menghasilkan panas dari arus listrik yang melewatinya.
2. Kawat nikrom tidak mengoksidasi saat dipanaskan, sehingga tidak mudah mengarat walau digunakan pada suhu tinggi, sementara logam lainnya seperti besi mudah berkarat pada suhu panas.

2.14 Keypad Matrix 4x4



Gambar 2.22 Rangkaian Dasar keypad 4x4

Sumber: http://www.mytutorialcafe.com/mikrokontroller%20bab7%20Keypad.htm, Juni 2010

Keypad sering digunakan sebagi suatu *input* pada beberapa peralatan yang berbasis mikroprosessor atau mikrokontroler. Keypad sesungguhnya terdiri dari sejumlah saklar, yang terhubung sebagai baris dan kolom dengan susuan seperti yang ditunjukkan dalam Gambar 2.22. Agar mikrokontroler dapat melakukan *scan keypad*, maka port mengeluarkan salah satu bit dari 4 bit yang terhubung pada kolom dengan logika low "0" dan selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut. Sebagai konsekuensi, selama tidak ada tombol yang ditekan, maka mikrokontroler akan melihat sebagai logika high "1" pada setiap pin yang terhubung ke baris.

2.15 DC brushless Fan

DC *brushless fan* biasanya tersedia pada tiga tegangan nominal: 12V, 24V dan 48V. Jika sistem telah diatur listrik di salah satu nilai tegangan, maka kipas DC *brushless* dapat dipilih dan akan memberikan kinerja yang tepat yang dibutuhkan. Karena kecepatan dan aliran udara dari kipas DC khas adalah proporsional terhadap tegangan yang disediakan, satu produk dapat digunakan untuk memenuhi berbagai aplikasi dengan memberi tegangan suplai untuk memberikan aliran udara yang diinginkan. Gambar 2.23 menggambarkan hasil variasi tegangan DC yang diberikan oleh DC *brushless fan*. Jika, misalnya, kipas angin 110 CFM pasokan udara pada pengiriman bebas, 28V dapat menghasilkan 127 CFM, yang harus dibutuhkan. Di sisi lain, operasi dapat menyediakan arus 24V terlalu banyak; tegangan suplai mungkin dikurangi ke tingkat yang memberikan aliran udara yang diinginkan. Rentang tegangan yang dapat diterapkan pada kipas angin untuk memastikan operasi yang sesuai tergantung pada desain kipas individu dan mungkin sekecil 10-14V untuk 12V unit dan sampai 12-56V untuk 48V unit.

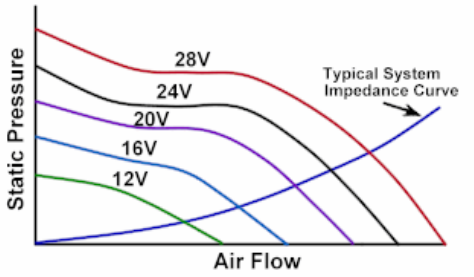


Figure 1: Results of Varying DC Fan Voltage

Gambar 2.23 DC Brushless Fan

Sumber : www.accesscontrolsales.com/

BAB III
METODOLOGI

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif yang diwujudkan dalam bentuk prototipe, yaitu perencanaan dan perealisasian alat agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat secara umum adalah sebagai berikut:

- Studi literatur
- Perancangan alat
- Realisasi pembuatan alat
- Pengujian alat dan analisa data
- Pengambilan kesimpulan

3.1 Studi Literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem pengendalian mesin penetas telur ayam otomatis menggunakan mikrokontroller AT89S51, karakteristik dari komponen yang digunakan serta studi tentang pengendalinya yaitu mikrokontroler AT89S51.

Adapun hal-hal yang berhubungan dengan hal tersebut adalah :

a. Studi tentang teori dasar proses penetasan telur ayam
   - Faktor-faktor yang mempengaruhi daya tetas.
   - Suhu tetas telur.
   - Kelembaban yang dibutuhkan pada proses penetasan telur.

b. Studi tentang rangkaian sensor
   - Karakteristik spesifikasi sensor suhu dan kelembaban HSM20G.
   - Penerapan sensor suhu dan kelembaban sesuai sistem.
   - Karakteristik komponen elektronik yang digunakan diantaranya meliputi pengkondisi sinyal, relay, LCD M1632, RTC DS1307, Mikrokontroler AT89S51, ADC PCF8591,transistor sebagai saklar dan motor DC.

- Penerapan *fan*, jendela dan pemanas sebagai aktuator sesuai sistem.

c. Mikrokontroller AT89S51

- Karakteristik umum mikrokontroler AT89S51.
- Cara memprogram mikrokontroler AT89S51 sebagai pusat pengontrol suhu dan kelembaban mesin selama proses penetasan.

3.2 Perancangan Alat

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengendalian pada mesin penetas telur tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

- Penentuan dimensi dan bahan yang digunakan dalam pembuatan mesin penetas telur.
- Penentuan rangkaian elektronik yang digunakan.
- Perancangan program pada mikrokontroler AT89S51 sebagai pusat pengontrol.
- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
    a. Penentuan deskripsi kerja sistem secara keseluruhan.
    b. Mikrokontroler AT89S51 dan rangkaian elektronik pendukung

3.2 Realisasi Pembuatan Alat

3.2.1 Perancangan Perangkat Keras dan Realisasi Tiap Blok

a. Pembuatan blok diagram lengkap sistem

b. Penentuan dan perhitungan komponen yang akan digunakan

c. Merakit perangkat keras masing-masing blok

3.2.2 Perancangan dan Penyusunan Perangkat Lunak

Setelah mengetahui perangkat keras yang dirancang, maka dibutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Parameter yang diperoleh dari hasil perhitungan kemudian diterapkan kedalam mikrokontroler AT89S51 dengan menggunakan bahasa assembly.

3.3 Pengujian Alat dan Analisis

Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka dilakukan pengujian alat meliputi perangkat keras *(hardware)* yang dilakukan baik per blok rangkaian maupun keseluruhan sistem.

3.3.1 Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan dengan tujuan untuk menyesuaikan nilai tegangan dan arus yang diijinkan bekerja dalam komponen berdasarkan data sekunder komponen yang diambil dari buku data komponen elektronika maupun dari datasheet.

3.3.2 Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

3.4 Pengambilan Kesimpulan.

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya.

BAB IV
PERANCANGAN SISTEM

4.1 Tinjauan Umum

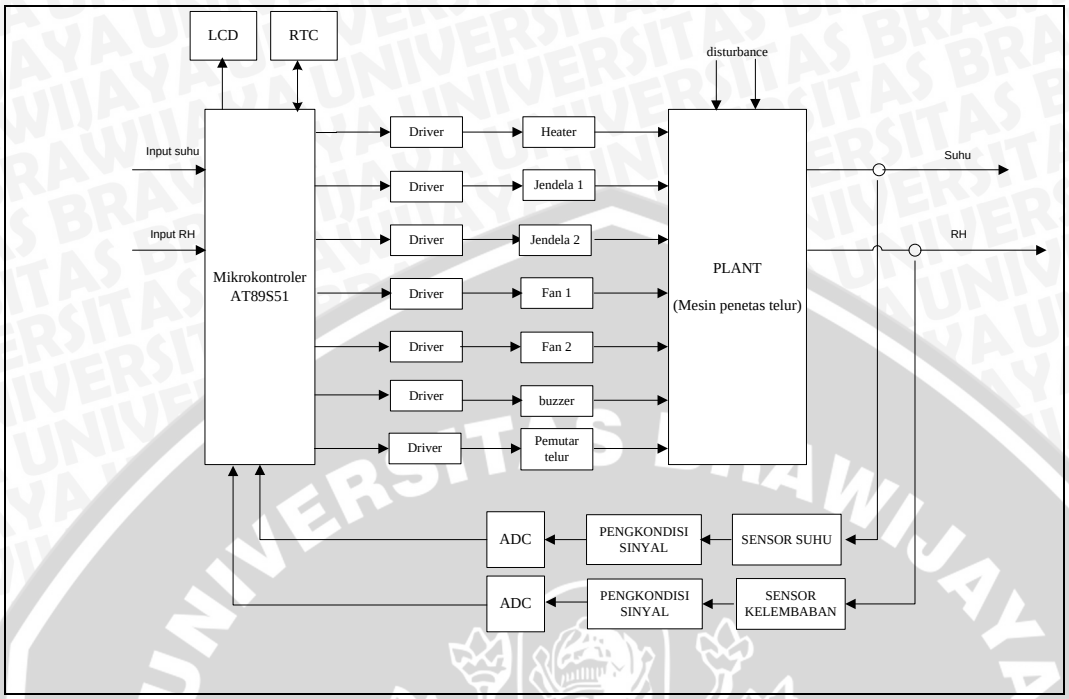
Bab ini menjelaskan mengenai spesifikasi alat, perancangan perangkat keras dari mesin penetas telur ayam, yang meliputi diagram blok sistem, cara kerja sistem, *flowchart* atau diagram alir kerja sistem, dan gambar rancangan alat. Selain perancangan perangkat keras akan dijelaskan juga mengenai perancangan perangkat lunak.

Perancangan pengontrolan suhu dan kelembaban udara dalam mesin penetas telur ayam ini dirancang dengan spesifikasi sebagai berikut :

1. Mengkondisikan suhu mesin penetas pada kisaran 38,5^{o} - 40,5^{o}C.
2. Mengkondisikan kelembaban ruang pada set point 70%.
3. Menggunakan Mikrokontroler AT89S51 sebagai pengontrol utama.
4. Menggunakan LCD tipe M1632 (16 kolom x 2 baris) sebagai penampil informasi yang dikontrol oleh Mikrokontroler.
5. Menggunakan ADC PCF8591 sebagai konverter masukan sinyal analog dari pengkondisi sinyal menjadi sinyal digital sehingga dapat dibaca oleh mikrokontroler.
6. Menggunakan sensor suhu dan kelembaban HSM20G.
7. Menggunakan keypad 4x4 untuk memasukkan set point suhu dan kelembaban serta memasukkan tanggal dimulainya proses pengeraman telur pada mesin tetas.
8. Menggunakan rangkaian driver sebagai penggerak dari motor DC, yang berfungsi sebagai penggerak rak telur.
9. Menggunakan RTC DS1307 sebagai penghitung hari lamanya penetasan.
10. Aktuator berupa *fan* 12V DC, *heater* dan jendela yang digerakkan menggunakan motor 12V DC sehingga dapat terbuka dan tertutup.

4.1.1 Blok Diagram Alat

Perancangan diagram blok sistem ditunjukkan dalam Gambar 4.1



Gambar 4.1 Diagram Blok Sistem

Sumber : *Perancangan*

4.1.2 Prinsip Kerja Alat

Untuk lebih memudahkan dalam perencanaan, maka perlu dijabarkan deskripsi kerja sistem secara keseluruhan. Mula- mula suhu pada alat penetas telur diatur melalui *keypad*. Kemudian sensor suhu dan kelembaban akan membaca kondisi suhu dan kelembaban pada alat penetas telur dan akan memberikan sinyal yang akan diproses oleh rangkaian pengkondisi sinyal sebagai masukan ADC. Setelah itu, oleh ADC, sinyal dari sensor suhu dan kelembaban yang masih berupa sinyal analog tersebut diubah menjadi sinyal digital sebagai masukan mikrokontroler.

Setelah sinyal digital masuk ke dalam mikrokontroler maka akan langsung diproses oleh mikrokontroler. Salah satu keluaran dari mikrokontroler adalah penggerak motor DC yang akan mengatur pemutaran telur setiap 6 jam sekali.

Selain itu, mikrokontroler juga akan mengatur setiap perubahan suhu pada alat penetas telur yang setiap harinya berubah. Set point suhu untuk proses penetasan telur ayam yaitu antara 38,5-40,5 ^{o}C

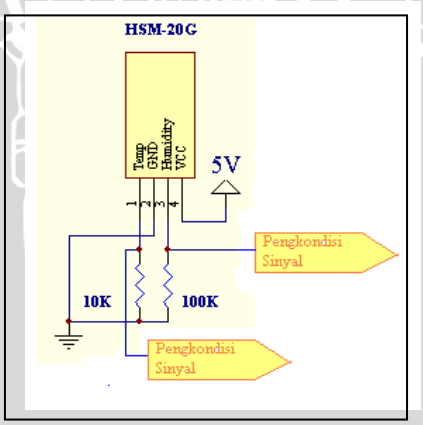
dengan memanfaatkan *heater* dan *fan* sebagai aktuatornya. *Heater* dikondisikan untuk mengeluarkan udara panas dan *fan* dikondisikan untuk mengurangi suhu yang berlebihan di dalam alat penetas telur tersebut.

Sebagai pemberi informasi berakhirnya proses penetasan telur diberikan *buzzer* yang akan aktif ketika telah mencapai hari ke -21 dan kemudian alat penetas telur akan mati/OFF secara otomatis.

4.2 Perancangan Perangkat Keras

4.2.1 Rangkaian Sensor Suhu dan Kelembaban HSM20G

Pada perencanaan alat ini, menggunakan sensor suhu dan kelembaban HSM20G. dimana sensor ini memiliki keluaran tegangan linier. Karena tegangan keluaran HSM-20G cukup kecil maka diperlukan pengkondisi sinyal atau penguat tegangan, sehingga bila terjadi perubahan temperatur atau suhu maka perubahan tegangan keluaran pada sensor masih dapat diamati dalam skala yang cukup besar. Adapun rangkaian perencanaan sensor suhu dan kelembaban HSM20G seperti terlihat dalam Gambar 4.2.



Gambar 4.2 Rangkaian Perencanaan Sensor Suhu dan Kelembaban HSM20G

Sumber : Perancangan

Untuk menghitung tegangan *output* dari sensor ini pada saat membaca temperatur adalah:

$$Vout = \frac{R1}{Rs + Rl} \times V$$

Dimana:

Vout = Tegangan *output* dari sensor HSM20G (volt)

R_1 = 10KΩ (R_1 = hambatan pada modul HSM20G untuk pengukuran suhu)

Rs = 47,00kΩ data sheet unutk 25°C

Rs = 37,43kΩ data sheet untuk 30°C

Rs = 24,19kΩ data sheet untuk 40°C

Rs = 16,01kΩ data sheet untuk 50°C

Sebagai contoh:

Suhu 25°C maka tegangan dari sensor adalah:

$$Vout = \frac{10\text{k}\Omega}{47{,}00k\Omega + 10\text{k}\Omega} \times 5\text{V}$$

$$Vout = 0{,}877V$$

Suhu 30°C maka tegangan dari sensor adalah:

$$Vout = \frac{10\text{k}\Omega}{37{,}43k\Omega + 10\text{k}\Omega} \times 5\text{V}$$

$$Vout = 1{,}054V$$

Suhu 40°C maka tegangan dari sensor adalah:

$$Vout = \frac{10\text{k}\Omega}{24{,}19k\Omega + 10\text{k}\Omega} \times 5\text{V}$$

$$Vout = 1{,}462V$$

Suhu 50°C maka tegangan dari sensor adalah:

$$Vout = \frac{10\text{k}\Omega}{16{,}01k\Omega + 10\text{k}\Omega} \times 5\text{V}$$

$$Vout = 1{,}922V$$

Untuk tegangan output dari sensor ini pada saat membaca kelembaban adalah:

Vout 50% = 2,02 V (Tegangan dari datasheet HSM 20G)

Vout 60% = 2,37 V (Tegangan dari datasheet HSM 20G)

Vout 70% = 2,69 V (Tegangan dari datasheet HSM 20G)

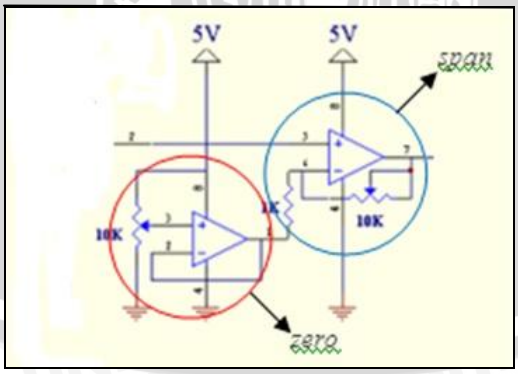
HSM 20G dihubungkan dengan pengkondisi sinyal lalu di teruskan ke ADC PCF 8591. Untuk sensor kelembaban berada pada pin

1 sebagai *output* temperatur yang kemudian dihubungkan ke AIN 1 yang terletak pada PCF8591. sedangkan untuk suhu kelembaban dihubungkan ke AIN 0 yang juga terletak pada ADC PCF8591.

4.2.2 Rangkaian Penguat Sinyal

Untuk menaikkan resolusi pembacaan data suhu dan kelembaban, maka *output* dari sensor diumpankan ke suatu rangkaian penguat *non inverting* dengan menggunakan *Op-Amp*, hal ini dibutuhkan karena *output* yang dihasilkan oleh sensor masih sangat kecil, untuk itu maka dibutuhkan suatu rangkaian untuk memperkuat *output* dari sensor tersebut.

Rangkaian pengkondisi sinyal untuk sensor suhu dan kelembaban HSM20G yang dirancang pada sistem ini terbagi menjadi dua macam rangkaian op-amp, yang kemudian digunakan istilah sebagai rangkaian *zer*o dan *span*. Rangkaian *zero* berfungsi sebagai *virtual input* bagi rangkaian *span*, dimana nilai tegangan keluaran pada rangkaian *zero* tersebut diatur sehingga sama dengan nilai tegangan untuk menyatakan batas bawah kondisi suhu pada range kerja untuk pengendalian suhu. Rangkaian *span* merupakan rangkaian penguat selisih tegangan (penguat *differensial*) yang berfungsi untuk menguatkan sinyal-sinyal tegangan yang nilainya kecil. Untuk lebih jelasnya dapat dilihat pada rangkaian dalam Gambar 4.3.



Gambar 4.3 Rangkaian Penguat Non-Inverting

Sumber : *Perancangan*

Keluaran tegangan dari sensor ini tidak memenuhi untuk dapat mengubah resolusi satu bit pada ADC. Resolusi ADC PCF8951 dengan Vref sama dengan 5 Volt adalah:

$$\frac{Vref}{2^8} = \frac{5}{2^8} = \frac{5}{255} = 0{,}0196Volt$$

Maka untuk itu diperlukan suatu penguat. Berikut ini perancangan untuk mendapatkan gain yang tepat:

- Range yang akan diukur ditetapkan yakni 25°C – 50°C
- Untuk suhu 25°C maka output sensor adalah 877 mV
- Untuk suhu 50°C maka output sensor adalah 1922 mV
- Perubahan bit dari 25°C sampai 50°C adalah sebanyak 250, sehingga pada saat suhu 25°C maka diharapkan output data pada ADC adalah 0 dan ketika suhu 50°C maka data *output* dari ADC adalah 250.

$$Tampilan_Layar = \frac{data_ADC}{10} + 25$$

Vout Op-Amp = data_ADC x 0,0196

ketika suhu 25°C maka:

$$Tampilan_Layar = \frac{0}{10} + 25 = 25$$

Vout Op-Amp = 0 x 0,0196

= 0 mV

ketika suhu 50°C maka:

$$Tampilan_Layar = \frac{250}{10} + 25 = 50$$

Vout Op-Amp = 250 x 0,0196

= 4900 mV

Dari hasil perhitungan di atas maka dapat dicari gain yang tepat yakni:

$$Gain = \frac{V_{out-OP_Amp}}{V_{in-OP_Amp}}$$

$$Gain = \frac{4900 - 0}{1922 - 877} = 4{,}688$$

Jadi untuk menentukan besarnya nilai hambatan R yang diperlukan untuk mendapatkan gain sebesar 4,688 adalah:

$$Gain = \frac{R2}{R1} + 1 = 4{,}688$$

Karena suhu pada perancangan ini diset pada range 25,0°C - 40,5°C, maka perlu dibuat agar pada suhu 25,0°C dalam keadaan 0 mV. Pada perancangan ini hambatan untuk temperatur dengan nilai 10KΩ dan *humidity* dengan nilai 100KΩ dihubungkan dengan kaki nomor satu dan tiga pada HSM 20G yang berfungsi sebagai zero, yaitu membuat tegangan pada keluaran kaki nomor empat HSM 20G menjadi 0V ketika inputnya 877mV.

Dengan rumus:

Vin $_{\text{Op-Amp}}$ = Vin-Vreff (Zero)

Vout $_{\text{Op-Amp}}$ = (Vin $_{\text{Op-Amp}}$. Gain) + Vreff (Zero)

$$Gain = \frac{R2}{R1} + 1$$

Diasumsikan Vout $_{\text{Op-Amp}}$ pada suhu 25,0°C adalah 0 mV maka

Vout Op-Amp = (Vin Op-Amp . Gain) + Vreff (Zero)

$$Vout_{Op-Amp} = ((Vin - Zero)x\left(\frac{R2}{R1} + 1\right) + Zero$$

$$Zero = \left(\frac{Vin\left(\frac{R2}{R1} + 1\right)}{\frac{R2}{R1}}\right)$$

$$Zero = \left(\frac{0{,}877V \times 4{,}688}{3{,}688}\right)$$

$$Zero = 1{,}114V$$

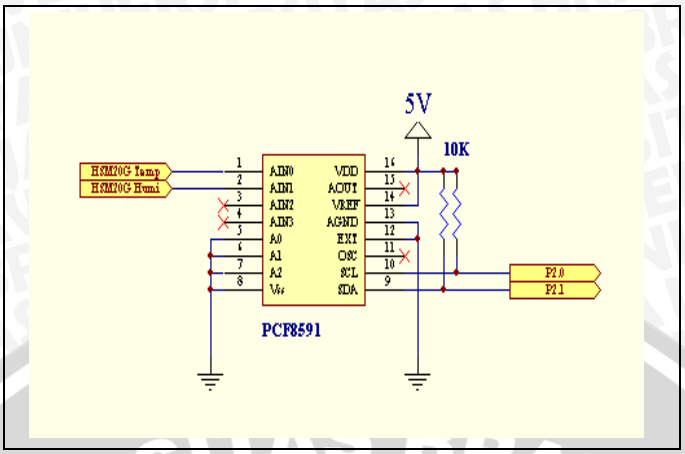
Zero = 1,114V

Pada perancangan ini digunakan penguatan sebesar 4,688kali dan zero sebesar 1,114V.

4.2.4 Rangkaian ADC PCF 8591

Tegangan keluaran dari rangkaian pengkondisi sinyal berupa tegangan analog. Agar nilai tersebut dapat dibaca oleh mikrokontroler AT89S51, maka nilai tegangan tersebut harus diubah menjadi data digital 8 bit menggunakan konverter analog ke digital. Dalam perancangan ini digunakan konverter analog ke digital berupa ADDA PCF8591 berfungsi sebagai konverter analog ke digital dengan 4

masukan analog yang multipleks menjadi data digital 8 bit. Rangkaian ADC PCF8591 seperti yang ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Rangkaian Perencanaan ADC (PCF8591)

Sumber : *Perancangan*

Keluaran dari ADC dapat dihitung sebagai berikut, diketahui Vin (analog) = 2,5 V, Vref (analog) = 5 V, resolusi digital = 2^8 =256.

Out (digital)= $\frac{Vin}{Vref}$ x resolusi (digital)

Out (digital)= $\frac{2.5}{5}$ x 256

Out (digital) = 128 (dalam bentuk desimal)

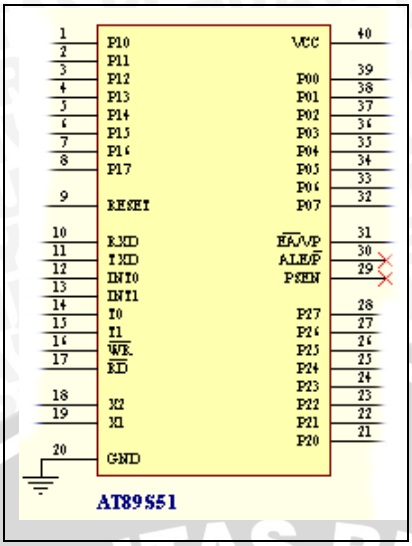
= 2^7 (dalam bentuk heksadesimal)

= (1000 0000) (dalam biner)

Jadi keluaran dari ADC-nya adalah 128 (bentuk desimal) atau 2^7 (heksadesimal) = 1000 0000 (bentuk biner).

4.2.5 Rangkaian Mikrokontroller AT89S51

Pada perancangan ini, menggunakan mikrokontroler AT89S51 sebagai pengatur aliran data dari ADC menuju LCD dan konverter. Dalam Gambar 4.8 ditunjukkan pin-pin mikrokontroler AT89S51 yang terhubung ke Serial ADC PCF8591, LCD M1632, RTC DS1307, keypad, driver motor DC, driver *heater*, driver *fan*, dan driver *buzzer*.



Gambar 4.5 Rangkaian Perencanaan Mikrokontroller AT89S51

Sumber : *Perancangan*

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

1. **Port 0**

   Port 0 merupakan port yang terletak pada pin 32-39, dalam perancangannya P0.0-0.7 digunakan sebagai inputan LCD M1632.

2. **Port 1**

   Port 1 disediakan sebagai port I/O dan menempati pin 1-8, dalam perancangan ini P1.0-1.7 digunakan sebagai saluran keypad.

3. **Port 2**

   Port 2 disediakan sebagai port I/O dan menempati pin 21-28, dalam perancangannya memiliki fungsi sebagai berikut :

   - P2.0-P2.1 sebagai inputan ke RTC DS1307 dan ADC PCF8591
   - P2.2-P2.5 sebagai inputan driver motor
   - 2.6 digunakan sebagai control RS dan E

4. **Port 3**

   Port 3 menempati pin 10-17, dalam perancangannya memiliki fungsi sebagai output untuk driver relay, driver motor dan jendela.

5. **XTAL1 dan XTAL2**

Digunakan sebagai input dari rangkaian osilator kristal. Rangkaian osilator kristal terdiri dari kristal osilator 11.0592 MHz, kapasitor C1 dan C2 yang masing-masing bernilai 33 pF, akan membangkitkan pulsa *clock* yang menjadi penggerak bagi seluruh operasi internal mikrokontroler.

6. **VCC**

VCC dihubungkan dengan tegangan sebesar +5V sesuai dengan tegangan operasi *chip* tunggal yang diijinkan dalam *data sheet*.

7. **GND**

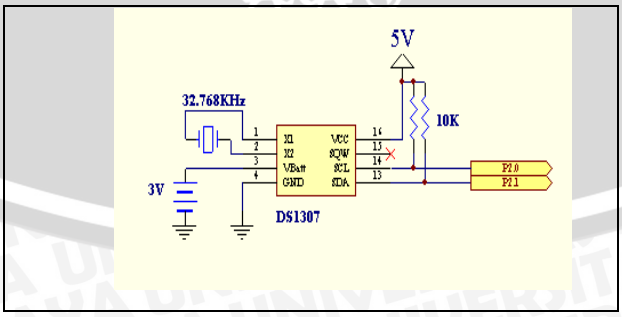
GND dihubungkan ke *ground* catu daya.

8. **Reset**

Digunakan untuk mereset program kontrol mikrokontroler, maka pin reset diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset, kapasitor dihubungkan dengan VCC dan sebuah resistor yang dihubungkan ke *ground.*

4.2.6 Rangkaian RTC DS1307

RTC (*Real Time Clock*) merupakan suatu IC yang digunakan sebagai pencatat waktu dari keseluruhan waktu. Dan IC ini sudah dilengkapi dengan baterai sendiri, maka bila terjadi pemutusan *power supply*, data dari waktu yang telah catat tidak akan hilang dan IC akan tetap mencatat penambahan waktu.



Gambar 4.6 Rangkaian Perencanaan RTC (*Real Time Clock*)

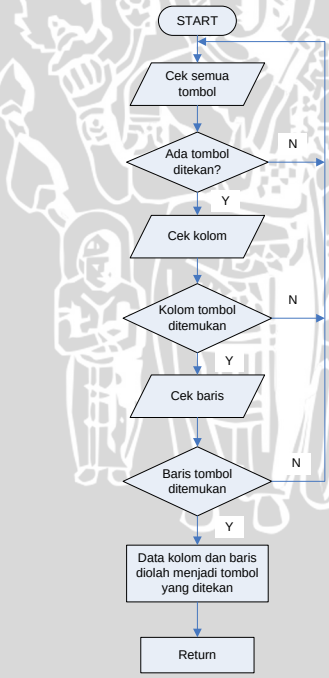
Sumber : *Perancangan*

Pin X1 dan X2 dihubungkan dengan sebuah kristal yang besar nilainya 32,768 KHz. Pin Vbat dihubungkan dengan baterai cadangan

sebesar 3V. Pin GND dihubungkan pada ground, sedangkan pin Vcc dihubungkan dengan power supply sebesar +5V. Pin SCL dan SDA dihubungkan dengan mikrokontroler AT89S51 (port 2 P2.0-P2.1).

4.2.7 Rangkaian Keypad 4x4

Pembuatan *keypad*, disini dimaksudkan untuk memberikan masukan dari konsumen. *Scanning keypad* yang dilakukan oleh Mikrokontroler AT89S51 harus mampu menentukan posisi dari tombol yang ditekan. Setelah posisi *keypad* yang aktif dapat ditemukan maka data tersebut diolah menjadi data data tombol yang ditekan. Dalam aplikasi ini digunakan satu macam *scanning* yang digunakan untuk mendeteksi *keypad* 4x4. Gambar 4.7 menunjukkan *flowchart scanning keypad* matriks yang digunakan dalam program. *Scanning keypad* ini menggunakan metode *polling*. Yaitu *scan* terus menerus hingga ada tombol yang ditekan.



Gambar 4.7 Flowchart scanning keypad dengan metode polling

Sumber : http://www.delta-electronic.com/

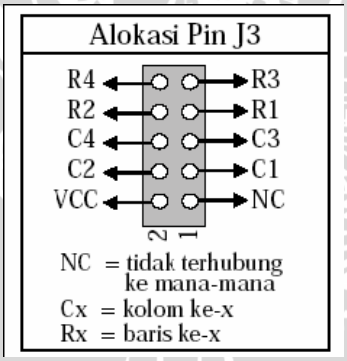
Keypad yang digunakan dalam aplikasi ini adalah *keypad* matriks 4x4 (4 baris dan 4 kolom). Berisikan angka 0 sampai 9. dan 6 tombol tyang tersisa dimanfaatkan untuk CAN,COR, ENT, MEN, UP, dan DOWN sedangkan 1 tombol lagi tidak dipakai. Untuk mengenali

bagian kolom dan baris yang aktif maka keypad ini dihubungkan dengan minimum sistem AT89S51. kemudian dibuat program yang dapat mengenali tombol yang sedang ditekan. Program yang dibuat harus mampu mengenali setiap tombol yang ditekan sesuai perencanaan. Dan berikut merupakan bagian bagian dari keypad tersebut.

Gambar 4.8 Susunan tombol pada keypad 4x4

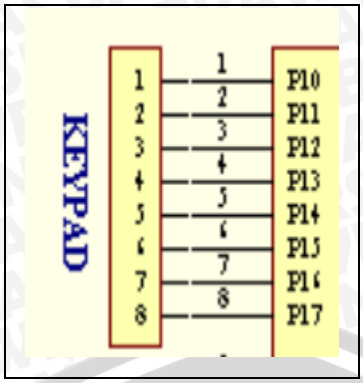
Sumber : http://www.innovativeelectronics.com/

Gambar 4.9 merupakan alokasi pin *output* pada modul *keypad* 4x4 beserta skematiknya tersebut.



Gambar 4.9 Alokasi Pin-Pin Pada Keypad 4x4

Sumber : 4x4 Keypad Module



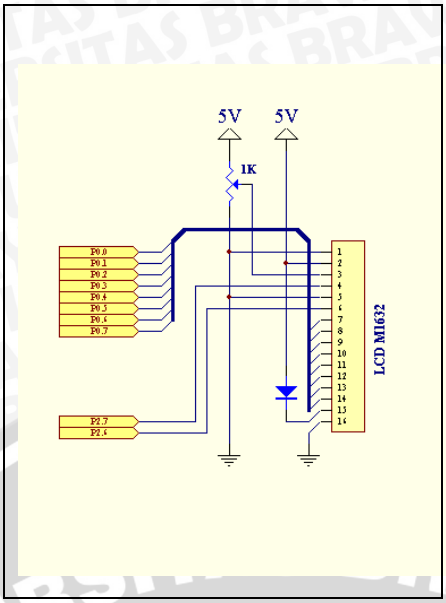
Gambar 4.10 Rangkaian Perencanaan Keypad 4x4

Sumber : *Perancangan*

Untuk menjalankan menu program pertama kali dapat dilihat dari tampilan LCD, yang mana untuk menu pengecekan kondisi suhu dapat diakses dengan menekan tombol 1 (satu) dan untuk menu Kelembaban dapat diakses dengan menekan tombol 2 (dua) pada keypad. Untuk intensitas cahaya yang diinginkan kita tinggal menekan tombol angka yang telah tersedia pada *keypad*. Apabila terjadi penekanan tombol *keypad* maka akan men-*drive* pin port pada (P1.0 – P1.7) dan rangkaian minimum sistem akan memilih dan menampilkan informasi yang berasal dari *keypad* ke LCD .

4.2.8 Rangkaian LCD M1632

LCD yang digunakan adalah tipe M1632 (16 kolom x 2 baris). *Bus* data LCD (D0-D7) terhubung dengan *port* 0 mikrokontroler (P0.0-P0.7). Karena LCD dioperasikan hanya menerima data, maka pin R/W dihubungkan dengan *ground*. RS dihubungkan dengan pin 2.6 dari mikrokontroler. Sedangkan untuk mengaktifkan E (*Enable*) LCD dibutuhkan keluaran dari pin 2.7. Untuk mengatur tingkat kecerahan LCD digunakan resistor variabel 1 kΩ. Gambar LCD dan konfigurasi pinnya dapat dilihat dalam Gambar 4.11.



Gambar 4.11 Rangkaian Perencanaan LCD M1632

Sumber : Perancangan

4.3 Perancangan Rangkaian Driver (Driver Relay dan Driver Motor)

4.3.1 Driver Motor

Driver motor DC ini digunakan untuk menggerakan motor jendela dan motor rak telur sehingga dapat digerakkan secara otomatis. Rangkaian *driver* ini dirancang sesuai program mikrokontroler, dimana kombinasinya adalah :

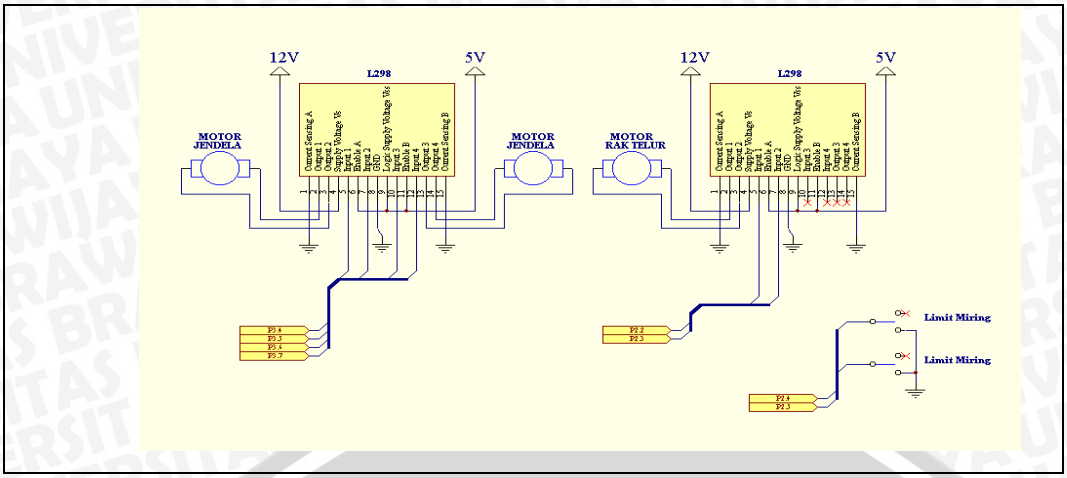
00 : Berhenti / *Stop (Free)*

01 : Putar ke kiri

10 : Putar ke kanan

11 : Berhrnti / *Stop (Break)*

Gambar 4.12 adalah gambar perancangan untuk driver motor yang menggerakkan jendela dan memutar rak telur,



Gambar 4.12 Rangkaian Perencanaan Driver Motor

Sumber : *Perancangan*

4.3.1.1 Motor DC

Penggunaan motor DC lebih menguntungkan apabila dibandingkan dengan motor jenis lain, karena motor DC lebih mudah diatur kecepatannya dalam rentang yang lebar dan karakteristik kopel putaran yang baik.

Pemutaran secara otomatis dengan bantuan motor DC untuk memindahkan posisi tray didalam mesin penetas telur agar terjadi sudut 45^{o} untuk tiap-tiap waktu yang ditetapkan secara bergantian. Pemutaran telur dalam perencanaan alat ini dibuat 4 kali perputaran motor dalam 24 jam. Hal ini dilakukan untuk mencegah embrio telur melekat pada selaput membran bagian dalam telur. Motor ini akan menggunakan rak, sehingga rak ini akan berubah posisi sejauh 45^{o}.

4.3.2 Driver Relay

Driver relay pada perancangan alat ini digunakan sebagai penyambung arus dari mikrokontroler ke rangkaian lainnya, di dalam *driver* relay ini terdapat beberapa komponen diantaranya *heater, fan* dan *buzzer*. Rangkaian *driver* ini dirancang sesuai program mikrokontroler, dimana terdapat dua sinyal kontrol dari mikrokontroler.

Sinyal kontrol pertama digunakan mikrokontroler untuk menghentikan motor putar, Bila sinyal ini berlogika *high* (5 volt)

motor DC berhenti (*OFF*) dan bila sinyal berlogika *Low* (0 volt) motor bisa berputar *(ON)*.

Sinyal kontrol kedua berlogika tinggi bila mikrokontroler hendak memutar motor searah jarum jam, dan berlogika rendah bila mikrokontroler hendak memutar motor berlawanan dengan arah jarum jam.

Gambar rangkaian *driver relay* diperlihatkan dalam gambar 4.13 Pensaklaran *supply* motor dilakukan oleh relay yang dikendalikan oleh transistor. Transistor-transistor yang digunakan dari jenis 9012 PNP dan 9013 NPN, dengan β sebesar 60.

Relay dihubungkan pada mikrokontroler yang dihubungkan pada port 2.0 pada pin 21 hingga port 2.4 dan pin 25. Dimana rumusnya sebagai berikut:

Untuk transistor (9012 PNP)

Arus dari mikrokontroler yang merupakan arus Ib bagi transistor kedua ketika kondisi *high* adalah 0,5 mA. Bila hfe dari transistor yang digunakan ditentukan sebesar 60 maka besar Rb untuk transistor 9012.

$$Rb = \frac{V - V_{BE}(sat)}{Ib}$$

$$Rb = \frac{5 - 0{,}7}{0{,}5} = 8{,}6K\Omega$$

Nilai hambatan Rb adalah 8,6KΩ ≈ 10 KΩ. dengan menggunakan hambatan dengan nilai 10KΩ maka arus Ib dari transistor adalah

Rb_1 = 10 kΩ

Rb_2 = 1KΩ

Tr_1 PNP 9012 (Hfe -60)

Tr_2 NPN 9013 (Hfe -60)

V_{be} (saturasi) = 60 (data sheet)

$Ib_1 = \frac{V - Vbe\ (saturasi)}{Rb1}$

$= \frac{5 - 0{,}7}{10\ k\Omega}$

= 0,43mA

Sedangkan besarnya Ic adalah

Ic_1 = Ib_1 . Hfe

= 0,43 . 60 (data sheet)

= 25,8 mA

Ib_2 = $\frac{V-Vbe\ (saturasi)}{Rb2}$

= $\frac{5-0,7}{1\ k\Omega}$

= 4,3mA

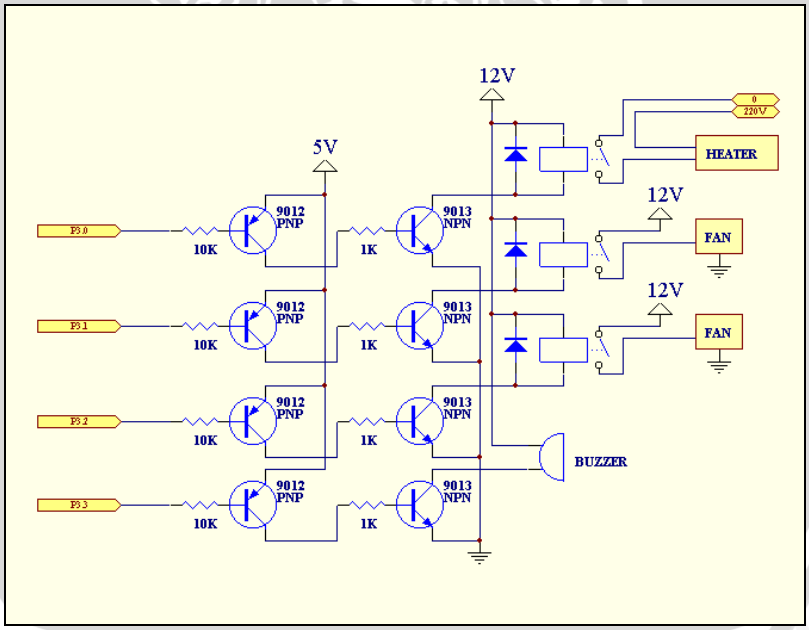
Sedangkan besarnya Ic adalah

Ic_2 = Ib_2 . Hfe

= 4,3 . 60(data sheet)

= 258mA

Dengan Ic sebesar 25,8 mA dan hfe sebesar 60 maka pada pembuatan digunakan transistor tipe 9012. Transistor ini mempunyai hfe sebesar 60 dan arus maksimal pada Ic mencapai 800 mA sehingga dengan arus Ic = 25,8 mA, transistor 9012 dianggap memenuhi syarat untuk menguatkan arus dari mikrokontroler.



Gambar 4.13 Rangkaian perencanaan driver relay

Sumber : *Perancangan*

Pada kaki-kaki belitan relay yang dialiri arus kolektor dipasang dioda. Bila arus dari kolektor diputus maka arus balik dari belitan relay akan dihubung singkat dan tidak merusak transistor. Digunakan dioda 1N4002 yang mampu melewatkan arus maksimum 1 A.

4.4 Perencanaan Perangkat Lunak

Untuk mendukung agar perangkat keras berfungsi sesuai dengan perencanaan, maka diperlukan perangkat lunak sebagai penunjangnya. Untuk mengatur dan mengendalikan keseluruhan sistem perangkat keras yang telah dibuat, harus dibantu dengan perangkat lunak. Sistem aplikasi Mikrokontroler AT89S51 ini dapat mengatur dan mengendalikan keseluruhan sistem apabila ada urutan instruksi yang mendefenisikan secara jelas urutan tugas yang harus dikerjakannya.

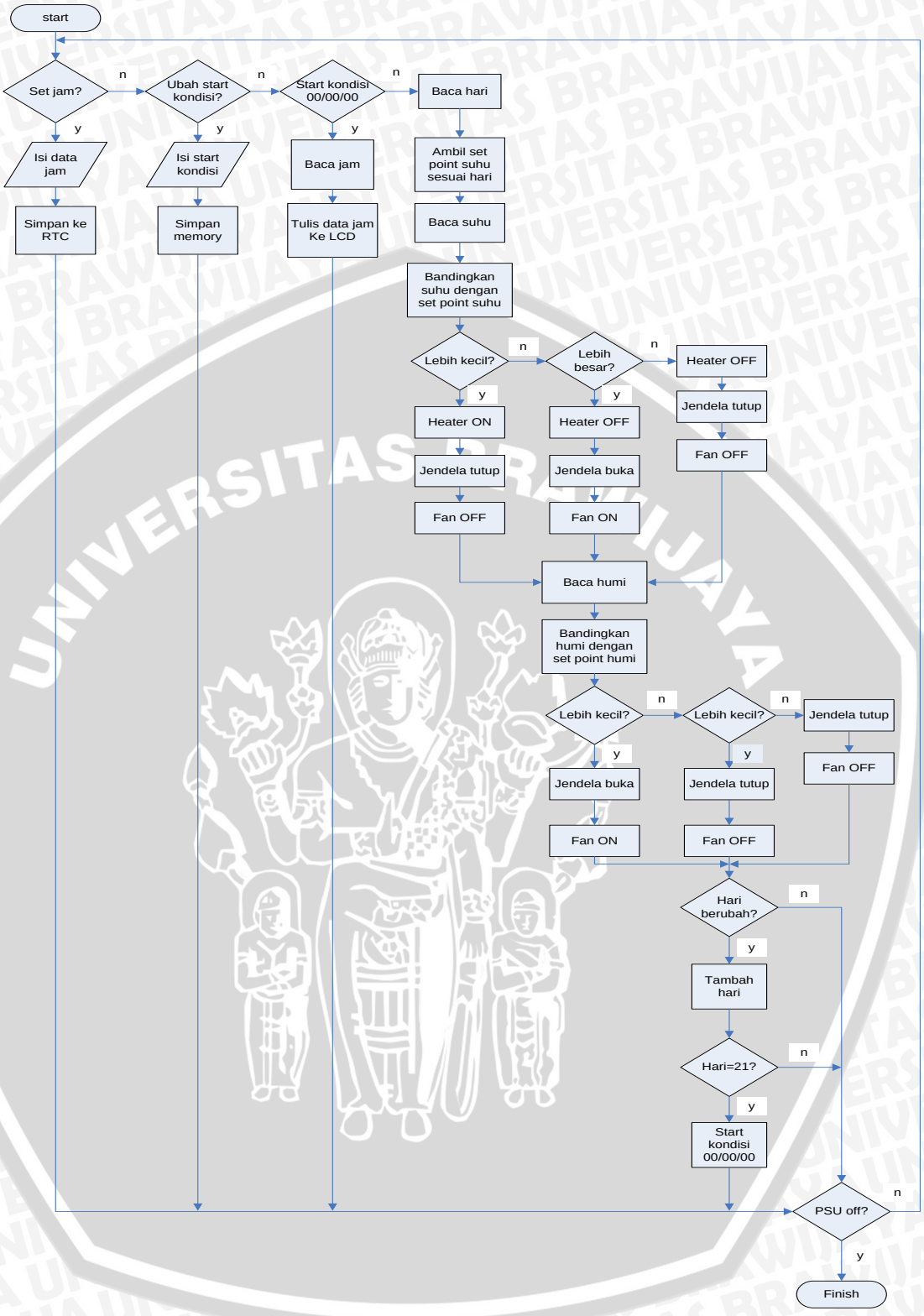
Urutan instruksi ini sangat penting untuk didefenisikan, karena mikrokontroler bekerja secara pasti berdasarkan urutan insruksi ini. Susunan logika perancangan yang salah tidak dapat diketahui oleh mikrokontroler. Selama instruksi yang diterima sesuai dengan aturannya, mikrokontroler tetap mengerjakan instruksi tersebut. Kesalahan seperti ini baru diketahui ketika kerja sistem aplikasi tidak sesuai dengan spesifikasi awal. Oleh karena itu, perancangan perangkat keras sangat menentukan dalam keberhasilan pembuatan perangkat lunak, sama pentingnya dengan perancangan perangkat keras. Sebuah mikrokontroler tidak akan bekerja bila tidak diberikan program. Program tersebut memberitahukan apa yang harus dilakukan oleh mikrokontroler.

Pembuatan perangkat lunak sistem pengontrol peralatan penetas telur berbasis mikrokontroler AT89S51 ini didasarkan pada semua kemungkinan kejadian yang harus dikerjakan oleh perangkat keras. Pembuatan perangkat lunak ini berdasarkan pada pengendali utamanya yaitu mikrokontroler AT89S51. Perangkat lunak terdiri atas program utama dan beberapa sub program. Tahap pembuatan perangkat lunak sistem pengontrol peralatan penetas telur berbasis mikrokontroler AT89S51 meliputi :

a. Penulisan kode *mnemonic* bahasa *assembler* dengan menggunakan editor teks menjadi *file* berekstensi H51.
b. Mengkompilasi *file* dengan ekstensi H51 dengan program ASM51 (*cross assembler* keluarga MCS-51) menjadi *file* LST dan OBJ.
c. Pengujian *file* OH dengan program simulasi ONJ menjadi HEX.
d. Mengubah format file HEX menjadi file BIN dengan program HB.
e. Mengisikan kode biner pada *file* BIN ke EPROM dengan bantuan

EPROM *writer*.

Dalam gambar 4.14 ditunjukkan diagram alir program utama system pengontrolan peralatan penetas telur menggunakan mikrokontroler AT89S51.



Gambar 4.14 Flowchart Program Utama
Sumber : Perancangan

BAB V
PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian pada sistem ini dibagi menjadi enam bagian, yaitu :

- Pengujian rangkaian sensor suhu dan kelembaban HSM20G
- Pengujian rangkaian ADC
- Pengujian LCD
- Pengujian Mikrokontroller AT89S51
- Pengujian driver motor
- Pengujian keseluruhan sistem

5.1 Pengujian Rangkaian Sensor Suhu dan Kelembaban HSM20G

5.1.1 Tujuan Pengujian :

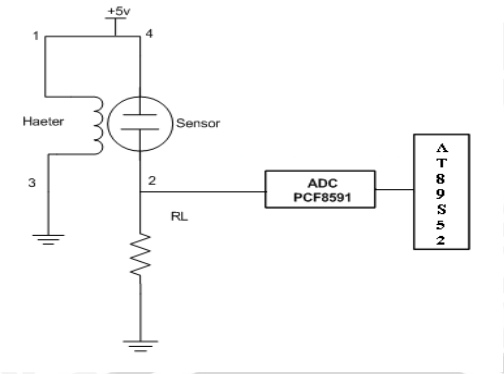
Untuk mengetahui besarnya tegangan output dari sensor suhu dan kelembaban HSM-20G berdasarkan perubahan kondisi suhu dan kelembaban pada ruangan saat mendeteksi suhu dan kelembaban dan diterima oleh ADC kemudian di lanjutkan oleh fan dan heater.

5.1.2 Peralatan yang digunakan :

1. Multimeter Digital Range RE830D.
2. Rangkaian yang akan diuji.
3. Catu Daya 5 Volt DC.

5.1.3 Prosedur Pengujian :

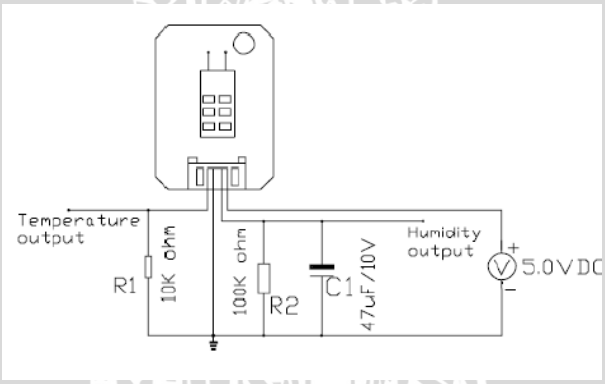
1. Merangkai rangkaian seperti pada Gambar 5.1.



Gambar 5.1 Pengujian Rangkaian Sensor Suhu dan Kelembaban dan Ic PCF8591

sebagai ADC (Analog to Digital Converter)

2. Menghubungkan IC PCF8591 dengan tegangan DC 5 volt pada kaki untuk Vref.
3. Menghubungkan kaki lainnya pada *ground*.
4. Analisa pengujian.



Gambar 5.2 Rangkain modul HSM 20G dari data sheet.

5.1.4 Hasil dan Analisis Pengujian :

Dalam pengujian rangkaian sensor ini dilakukan untuk mengetahui berapa tegangan *output* dari sensor suhu pada setiap perubahan suhu. Pengujian dan pengukuran rangkaian sensor dilakukan berdasarkan blok diagram diatas. Untuk kalibrasi sensor HSM-20G dengan menggunakan termometer yang sudah ada yaitu *hygrometer*. Untuk mencari kemungkinan kesalahan relatif dapat dicari dengan rumus sebagai berikut:

Δt=| Vout_Op-Amp (pengukuran) - Vout_Op-Amp (perhitungan) |

$$\Delta\% = \left| \frac{\text{Vout_Op - Amp (pengukuran) - Vout_Op - Amp (perhitungan)}}{\text{Vout_Op - Amp (perhitungan)}} \right| x100\%$$

$$Kesalahan_rata-rata = \left| \frac{\sum \Delta\%}{\sum \text{Pengukuran}} \right|$$

Diambil suatu sampel, suhu ruangan adalah 30°C. Untuk mengetahui berapa tegangan output yang dihasilkan maka kita dapat mengukurnya pada pin nomor 1 dari HSM 20G.

$$Vout = \frac{\text{Rl}}{\text{Rs} + \text{Rl}} \times v$$

Dimana:

Vout = Tegangan output dari sensor HSM 20G (volt)

Rl = 10KΩ

Rs = 37,43KΩ data sheet untuk 30°C

Sebagai contoh suhu adalah 30°C maka tegangan dari sensor adalah:

$$Vout = \frac{\text{Rl}}{\text{Rs} + \text{Rl}} \times v$$

$$Vout = \frac{10}{37{,}43 + 10} \times 5$$

Vout= 1,054 V

Δt = |Vout_Op-Amp(pengukuran)-Vout_Op-Amp(perhitungan)|

Δt = |1,078 – 1,054| = 0,024

$$\Delta\% = \left| \frac{\Delta t}{\text{Vout_Op - Amp (perhitungan)}} \right| x100\%$$

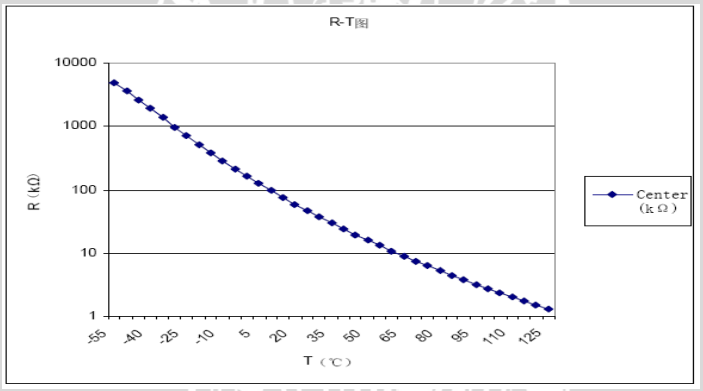
$$\Delta\% = \left| \frac{0{,}024}{1{,}054} \right| x100\%$$

Δ% = 2,27%

Dengan cara yang sama, nilai kesalahan relative dapat dilihat pada Tabel 5-1.

Tabel 5.1 Hasil Pengukuran dan Pengujian Suhu

NO.	Temperatur (°C)	Vout Sensor Suhu (Volt)		Δt	%
		Perhitungan	Pengukuran		
1	30	1,054	1,078	0,024	2,277
2	32	1,149	1,145	0,004	0,348
3	34	1,185	1,215	0,030	2,531
4	36	1,357	1,302	0,055	4,05
5	38	1,408	1,384	0,024	1,704
6	40	1,462	1,479	0,012	1,162
Kesalahan rata-rata				2,012 %	



Grafik.5.1 Hubungan antara suhu T (^{0}C) dan hambatan R (k Ω)

Diambil suatu sampel, Kelembaban adalah 70 %RH. Untuk mengetahui berapa tegangan *output* yang dihasilkan maka kita dapat mengukurnya pada pin nomor 3 dari HSM 20G.

$$Vout = \frac{Rl}{Rs + Rl} \times v$$

Dimana:

Vout = 2,69 V data sheet untuk 70%

R1 = 100KΩ

V = 5 V

Sebagai contoh suhu adalah 70%RH maka tegangan dari sensor adalah:

$$Vout = \frac{\mathrm{Rl}}{\mathrm{Rs} + \mathrm{Rl}} \times v$$

$$Rs = \left(\frac{\mathrm{Rl}}{\mathrm{Voutl}} \times v \right) - Rl$$

$$Rs = \left(\frac{100}{2{,}69} \times 5 \right) - 100$$

$$Rs = (37{,}17 \times 5) - 100$$

$$Rs = 85{,}87\ K\Omega$$

Δt = |Vout_Op-Amp(pengukuran)-Vout_Op-Amp(perhitungan)|

Δt = |2,73 – 2,69| = 0,04

$$\Delta\% = \left| \frac{\Delta \mathrm{t}}{\mathrm{Vout_Op\text{-}Amp(perhitungan)}} \right| x100\%$$

$$\Delta\% = \left| \frac{0{,}04}{2{,}69} \right| x100\%$$

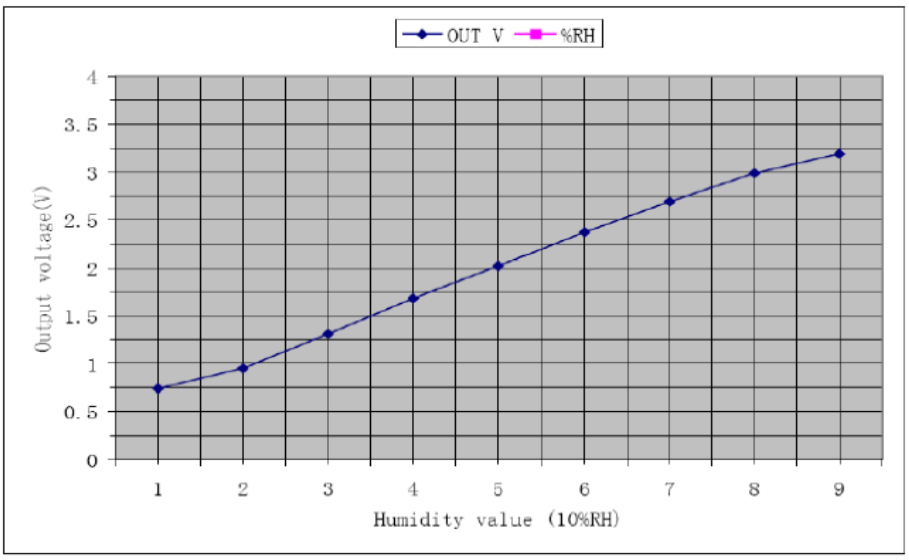
$$\Delta\% = 1{,}48\%$$

Dengan cara yang sama, nilai kesalahan relative dapat dilihat pada Tabel 5.2.

Tabel 5.2 Hasil Pengukuran dan Pengujian kelembaban

No	Humidity % (RH)	Vout Sensor kelembaban (Volt)		Δt	Δ%
		Perhitungan	Pengukuran		
1	70	2,69	2,73	0,04	1,48
2	65	2,53	2,59	0,06	2,37
3	60	2,37	2,40	0,03	1,26
4	55	2,20	2,27	0,07	3,18
5	50	2,02	2,06	0,04	1,98
6	45	1,94	1,98	0,04	2,06
Kesalahan rata-rata				2,05 %	

Untuk pengukuran kelembabapan dapat dilihat dari Grafik 5.2



Grafik 5.2 Hubungan antara Vout dengan kelembaban (%RH)

5.2 Pengujian ADC PCF8591

5.2.1 Tujuan Pengujian :

Untuk mengetahui cara kerja dari rangkaian ADC PCF8591, apakah sesuai dengan yang direncanakan atau belum maka dilakukan pengujian.

5.2.2 Peralatan yang digunakan :

1. Rangkaian ADC PCF 8591
2. Sumber tegangan 5 volt
3. Resistor 330 Ω
4. LED

5.2.3 Prosedur Pengujian :

1. Menghubungkan pin 9 (SDA) dan pin 10 (SCL) ADC PCF 8591 dengan rangkaian saklar
2. Menghubungkan rangkaian pengujian ADC PCF 8591 dengan resistor 330 Ω, led, dan sumber tegangan 5 V
3. Ketika ADC itu diberikan clock pada kaki SCL dengan SDA diberi perintah untuk menerjemahkan data konversi analog ke digital makal LED akan akan menyala dan mati
4. Mengamati LED yang menyala dan mati kemudian mencatatat hasilnya pada Tabel 5.3.

5.2.4 Hasil dan Analisis Pengujian :

Untuk mengetahui perubahan tegangan tiap bit (Vperstep) ADC adalah dengan menggunakan perhitungan sebagai berikut:

Diketahui : V = 5 Volt Step ADC = 2^8 (255), maka

$$Vperstep = \frac{V\,max}{255}$$

$$= \frac{5}{255} = 0{,}0196\ V$$

Tabel 5.3 Hasil Pengujian ADC PCF8591

Input Tegangan (V)	SDA	SCL	Data Led	Keterangan
-	90h	Clock	01011010	Kirim base address ke PCF
-	00h	Clock	0000 0000	Pilih ADC ch0
-	91h	Clock	01011011	Tampilan data
0.00	00	Clock	00000000	Tampilan data
0.25	13	Clock	00001101	Tampilan data
0.50	26	Clock	00011010	Tampilan data
0.75	38	Clock	00100110	Tampilan data
1.01	52	Clock	00110100	Tampilan data
1.26	64	Clock	01000000	Tampilan data
1.50	77	Clock	01001101	Tampilan data
1.75	89	Clock	01011001	Tampilan data
2.02	103	Clock	01100111	Tampilan data
2.25	115	Clock	01110011	Tampilan data
2.50	128	Clock	10000000	Tampilan data
2.75	140	Clock	10001100	Tampilan data
3.00	153	Clock	10011001	Tampilan data
3.25	166	Clock	10100110	Tampilan data
3.50	179	Clock	10110011	Tampilan data
3.75	191	Clock	10111111	Tampilan data
4.00	204	Clock	11001100	Tampilan data
4.25	217	Clock	11011001	Tampilan data

4.50	230	Clock	11100110	Tampilan data
4.75	242	Clock	11110010	Tampilan data
4.99	255	Clock	11111111	Tampilan data

Sedangkan untuk mengetahui data tegangan keluaran dari ADC ditentukan dengan perhitungan sebagai berikut:

Diket V analog = 0.25 Volt, maka

$$\text{Vout} = \frac{Vana\log}{Vperstep}$$

$$= \frac{3}{0.0196} = 153.03 = 153$$

$$= 10011001_{(2)}$$

$$= 99_{(16)}$$

5.3 Pengujian LCD

5.3.1 Tujuan Pengujian :

Pengujian LCD bertujuan untuk mengetahui LCD masih dapat berfungsi dengan baik sesuai dengan program yang diberikan oleh mikrokontroler.

5.3.2 Peralatan yang digunakan :

1. Mikrokontroler
2. LCD
3. Catu Daya 5 V

5.3.3 Prosedur Pengujian :

1. Menginisialisasi LCD
2. Member data karakter “Ella Puspasari” pada baris pertama dan “0610633025” pada baris kedua.

5.3.4 Hasil dan Analisis Pengujian :

Gambar 5.3 Menunjukkan tampilan NAMA pada baris pertama dan NIM pada baris kedua. Pada tampilan LCD terbukti bahwa data karakter yang diprogram oleh mikrokontroler dapat ditampilkan dengan baik.

Gambar 5.3 Gambar Tampilan LCD Pada Alat

.

5.4 Pengujian Mikrokontroler AT89S51

5.4.1 Tujuan Pengujian :

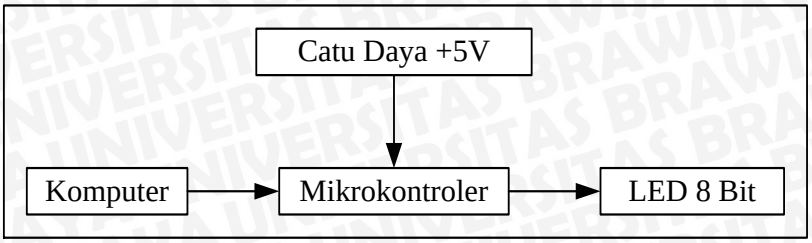
Pengujian mikrokontroler bertujuan untuk mengetahui kondisi awal dari sistem mikrokontroler agar sesuai dengan yang diharapkan

5.4.2 Peralatan yang digunakan :

1. Komputer dengan kabel penghubungnya
2. Minimum sistem mikrokontroler AT89S51
3. Lampu LED mewakili keluaran 8 bit
4. Catu daya +5 volt

5.4.3 Prosedur Pengujian :

1. Merangkai peralatan seperti dalam Gambar 5.4 keluaran terhubung dengan lampu LED yang mewakili keluaran 8 bit.
2. Mengisi mikrokontroler dengan program sederhana yang meletakkan 0FH dan F0H pada *Accumulator* secara bergantian kemudian memindahkannya ke port 1. kemudian *download* pada mikrokontroler AT89S51.
3. mengaktifkan catu daya.
4. mereset program untuk mengeluarkan data mikrokontroler.
5. mencatat data keluaran yang diwakili oleh lampu LED 8 bit ke dalam bentuk biner.



Gambar 5.4 Diagram Blok Pengujian Mikrokontroler

5.4.4 Hasil dan Analisis Pengujian :

Tabel 5.4. Hasil Pengujian Sistem Mikrokontroler

Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
I	1	1	1	1	0	0	0	0
II	0	0	0	0	1	1	1	1

Dari tabel 5.4 terlihat bahwa port 1 memberikan logika 0FH dan F0H secara bergantian sesuai dengan isi program.

5.5 Pengujian Driver Motor

5.5.1 Tujuan Pengujian :

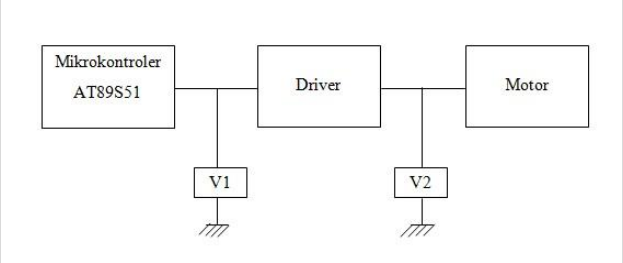
Pengujian ini bertujuan untuk mengetahui apakah *driver* motor rak telur dan jendela dapat berfungsi dengan baik sebagai penggerak dan pembalik arah putaran dan untuk mengetahui kecepatan respon dari *driver* motor tersebut terhadap kecepatan perubahan arah putaran. Dalam pengujian ini hal yang perlu dilakukan adalah, pertama memberikan tegangan 12Volt pada konektor rangkaian penggerak, pasangkan motor pada rangkaian *driver* dan beri tegangan 5 Volt pada kaki yang akan dikoneksikan ke mikrokontroler sebagai pembalik arah putaran maupun sebagai PWM. Kemudian hubungkan ground sebagai logika ‘1’ Bila logika ‘0’ diberikan pada kaki PWM maka motor akan begerak dan bila logika ‘0’ diberikan pada kaki pengubah arah putaran maka putaran motor akan berbalik.

5.5.2 Peralatan yang digunakan:

1. Motor DC
2. Mikrokontroler AT89S51
3. Multimeter digital
4. Power Supply driver Motor DC
5. Power Supply Motor DC

5.5.3 Prosedur Pengujian :

1. Alat dirangkai seperti dalam Gambar 5.5 dan memberikan logika rendah pada rangkaian *ON/OFF driver* Motor.
2. Mengukur nilai tegangan keluaran pada Motor.
3. Memberikan logika tinggi pada rangkaian ON/OFF driver Motor.
4. Mengukur kembali nilai tegangan keluaran pada Motor.



Gambar 5.5 Rangkaian Pengujian Driver Motor

Sumber : Perancangan

5.5.4 Hasil dan Analisis Pengujian :

Hasil pengujian dari rangkaian motor rak telur/jendela ditunjukkan dalam tabel di bawah ini. Dari Tabel 5.5 data tersebut didapatkan perubahan motor rak telur/jendela yang menyala dan tidak.

Tabel 5.5 Hasil Pengujian *driver* Motor

Logic (mikrokontroller)		Vout Logic Motor Perhitungan (V)		Vout Logic Motor Pengukuran (V)		Vout Driver Motor Perhitungan (V)		Vout Driver Motor Pengukuran (V)		Keterangan Arah Putaran
A	B	A	B	A	B	A	B	A	B	
0	0	0	0	0	0	0	0	0	0	Stop
0	1	0	5	0,00	4,24	0	12	0,00	12,91	Putar kiri
1	0	5	0	4,24	0,00	12	0	12,91	0,00	Putar kanan
1	1	5	5	4,24	4,24	12	12	12,91	12,91	Stop (*Break*)

5.6 Pengujian Sistem Keseluruhan

5.6.1 Tujuan Pengujian :

Pengujian alat secara keseluruhan bertujuan untuk mengetahui apakah perangkat keras dan perangkat lunak berjalan dengan baik. Hal ini dimaksudkan untuk melihat apakah sistem pengendalian berjalan sesuai dengan perencanaan dan sebagai indikasi penelitian ini berhasil.

5.6.2 Peralatan yang Digunakan :

1. Mesin penetas telur yang telah dibuat dan diprogram.
2. Catu daya 220 V AC
3. Telur ayam
4. Air sebagai penghasil uap air.

5.6.3 Prosedur Pengujian :

1. Masukkan telur ke dalam mesin penetas lalu hidupkan catu daya.
2. Hidupkan saklar pada posisi ON.
3. Masukkan tanggal dan waktu akan dimulainya proses penetasan.
4. Tekan tombol START jika proses penetasan telah siap dilaksanakan.
5. mencatat data.

5.6.4 Hasil dan Analisis Pengujian :

Berdasarkan pengujian yang dilakukan, untuk mencapai kondisi stabil pertama kali dipengaruhi oleh suhu lingkungan dan cuaca. Secara umum waktu yang diperlukan untuk mencapai kestabilan suhu pertama kali adalah 10-15 menit.

Salah satu contoh pengujian suhu ruang tetas sampai penetasan yang dilakukan terhadap telur ayam ditunjukan dalam tabel 5.6.

Tabel 5.6 Suhu Pengujian Proses Penetasan Ayam

Hari ke -	Suhu acuan	Suhu ruang tetas
1-3	38,5^{o}C	38,25-38,75 ^{o}C
4-7	39,0^{o}C	38,75-39,25 ^{o}C
8-12	39,5^{o}C	39,25-39,75 ^{o}C
13-17	40,0^{o}C	39,75-40,25 ^{o}C
18-22	40,5^{o}C	40,25-4075 ^{o}C

Suhu ruang tetas berfluktuasi ± 0,25 ^{o}C dari suhu acuan, perubahan suhu ini terjadi juga dengan telur lainnya, hal tersebut disebabkan perubahan 1LSB keluaran ADC dan pengaruh suhu luar sistem.

Dan keberhasilan penetasan telur dengan menggunakan mesin penetas dapat dilihat dari tabel 5.7 dan 5.8

Pengujian I

Jumlah telur yang ditetaskan sebanyak 12 butir telur.

Tabel 5.7 Pengujian I Keberhasilan Penetasan Telur

Hari ke -	Jumlah telur yang menetas
18	-
19	2 telur menetas
20	1 telur menetas
21	5 telur menetas
22	1 telur menetas

Pengujian II

Jumlah telur yang ditetaskan sebanyak 18 butir telur.

Tabel 5.8 Pengujian II Keberhasilan Penetasan Telur

Hari ke-	Jumlah telur yang menetas
18	1 telur menetas
19	-
20	-
21	10 telur menetas
22	2 telur manetas

Dari hasil pengujian penetasan telur diatas. Pada pengujian pertama, dari 12 butir telur yang ditetaskan, ada 9 butir telur yang berhasil menetas, dan 3 butir telur yang gagal menetas. Dan pada pengujian kedua, dari 18 butir telur yang ditetaskan ada 13 butir telur ayam yang berhasil menetas dan 5 butir telur yang gagal menetas. Kegagalan penetasan ini dikarenakan telur tetas yang tidak sesuai dengan standar penetasan, bisa karena telur yang tidak fertil, berat telur yang tidak ideal, atau karena kulit telur yang terlalu tipis, dll.

Dari hasil pengujian diperoleh bahwa mesin tetas dapat berfungsi dengan baik.

BAB VI
KESIMPULAN DAN SARAN

6.1 KESIMPULAN

Kesimpulan yang dapat diambil setelah melakukan penelitian, perancangan, pembuatan alat dan pengujian alat adalah:

1. Sensor suhu dan kelembaban pada mesin penetas telur digunakan untuk mendeteksi kondisi suhu dan kelembaban di dalam mesin penetas telur ayam dengan rata-rata kesalahan pengukuran sebesar 2,012% untuk pengujian suhu dan kesalahan rata-rata 2,05% pada saat pengujian kelembaban.
2. a. Waktu yang dibutuhkan untuk mencapai suhu stabil (suhu acuan untuk hari pertama) dari keadaan awal pada suhu mesin penetas adalah 10-15 menit.
   b. Rak telur akan berputar sebanyak 6 kali sehari, setiap 4 jam sekali ke arah kiri ataupun ke kanan dengan sudut 45^{o}.
   c. Keberhasilan mesin penetas ini adalah 70%

6.2 SARAN

Ada beberapa hal yang perlu diperhatikan untuk pengembangan dan kesempurnaan alat ini:

1. Pada skripsi ini hanya dapat mengontrol telur sebanyak 6-18 butir saja, maka dalam perkembangannya dapat diaplikasikan untuk keperluan yang lebih besar.
2. Perlu adanya kajian yang lebih lengkap untuk hubungan elemen pemanas denan volume atau ruang udara tertentu serta besarnya ruang ventilasi.

DAFTAR PUSTAKA

Atmel. 2000, 8-bit Microcontroller with 4K Bytes Flash AT89S51. HTTP://www.Atmel.com/AT89S51.pdf.

Atmel. 2000, 8-bit Microcontroller with 8K Bytes In-System Programmable Flash AT89S51. HTTP://www.Atmel.com/AT89S51.pdf.

Eko, A.P. 2002. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*. Yogyakarta : Gava Media.

Ogata, K. 1991. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga.

Malvino, Albert Paul. 1987. *Prinsip-prinsip Elektronika, Jilid 2, Cetakan ketiga*, terjemahan Prof. M. Barmawi, Ph.D, M.O Tjia, Ph.D. Jakarta : Erlangga.

Edwin Soedjarwo. 1997. *Membuat Mesin Tetas Sederhana*.Jakarta : PT.Penebar Swadaya.

Farry B. Paimin. 1995. *Membuat dan Mengelola Mesin Tetas*. Jakarta : PT. Penebar Swadaya .

Zainal, Abidin. 2003. *Membuat dan Mengelola Mesin Tetas Semimodern*. Jakarta : AgroMedia Pustaka.

```
        org     00h
;
        ISCL    Bit P2.0        ; I2C data
        ISDA    Bit P2.1        ; I2C clock
        Lmkr    Bit P2.2        ; limit miring kiri
        Lmkn    Bit P2.3        ; limit miring kanan
        Mrkr    Bit P2.4        ; motor miring kiri
        Mrkn    Bit P2.5        ; motor miring kanan
        Rest    Bit P2.6        ; RS LCD
        Enbl    Bit P2.7        ; E  LCD

        Fan0    Bit P3.0        ; fan 0
        Fan1    Bit P3.1        ; fan 1
        Buzr    Bit P3.2        ; buzzer
        Heat    Bit P3.3        ; heater
        Jbk0    Bit P3.4        ; jendela buka 0
        Jtt0    Bit P3.5        ; jendela tutup 0
        Jtt1    Bit P3.6        ; jendela tutup 1
        Jbk1    Bit P3.7        ; jendela buka 1

        Sjd0    Bit 20h.0       ; status jendela 0
        Sjd1    Bit 20h.1       ; status jendela 1
        Smrg    Bit 20h.2       ; status miring

        Djam    Equ 30h                 ; data jam
        Dmnt    Equ 31h                 ; data menit
        Ddtk    Equ 32h                 ; data detik
        Dhri    Equ 33h                 ; data hari
        Dtgl    Equ 34h                 ; data tanggal
        Dbln    Equ 35h                 ; data bulan
        Dcen    Equ 36h                 ; data century
        Dthn    Equ 37h                 ; data tahun
        Prdt    Equ 38h                 ; data perdetik
        Tmo0    Equ 39h                 ; data time out 0
        Tmo1    Equ 3Ah                 ; data time out 1
        Adc0    Equ 3Bh                 ; ADC sensor 0
        Adc1    Equ 3Ch                 ; ADC sensor 1

        Hari    Equ 40h                 ; hari penetasan
        Ghri    Equ 41h                 ; ganti hari
        Sttg    Equ 42h                 ; start kondisi
tanggal
        Stbl    Equ 43h                 ; start kondisi bulan
        Stth    Equ 44h                 ; start kondisi tahun
        Stpn    Equ 45h                 ; seting point
        Stsh    Equ 46h                 ; seting point suhu
        Sthm    Equ 47h                 ; seting point humi

        Char    Equ 50h
        Cntr    Equ 51h
```

```
Cnt0    Equ 52h
Cnt1    Equ 53h
Bufr    Equ 54h
Buf0    Equ 55h
Buf1    Equ 56h
Dly0    Equ 57h
Dly1    Equ 58h
Dly2    Equ 59h
Dly3    Equ 5Ah
;
init:  lcall   lcd_in
       lcall   rd_mem
       lcall   bc_jam
       mov     Ghri,Dhri
       mov     Sthm,#140              ; 70'/,RH
       setb    Sjd0
       setb    Sjd1
       clr     Smrg
       lcall   ttjnd0
       lcall   ttjnd1
;       mov     Hari,#04             ; baypass test
;
       mov     DPTR,#tpnama
       lcall   line1
       mov     Char,#16
       lcall   tulis
       mov     DPTR,#tpnims
       lcall   line2
       mov     Char,#16
       lcall   tulis
       lcall   delay2
       mov     DPTR,#tpjurs
       lcall   line1
       mov     Char,#16
       lcall   tulis
       mov     DPTR,#tpuniv
       lcall   line2
       mov     Char,#16
       lcall   tulis
       lcall   delay2
;
mulai:  lcall   lcdclr                 ; start sequential
       mov     Cntr,#10
loop0:  lcall   bc_jam                 ;\
       mov     DPTR,#angka             ; |
       mov     P0,#084h          ; |
       lcall   w_ins                  ; |
       mov     A,Djam                 ; |
       lcall   nil               ; |
       mov     P0,#':'                ; |
```

```
        lcall   w_chr                   ; |
        mov     A,Dmnt                  ; |
        lcall   nil             ; |
        mov     P0,#':'                 ; |
        lcall   w_chr                   ; |
        mov     A,Ddtk                  ; |
        lcall   nil             ; |
        mov     P0,#0C3h                ; |
        lcall   w_ins                   ; | tulis jam
        mov     A,Dtgl                  ; |
        lcall   nil             ; |
        mov     P0,#'/'                 ; |
        lcall   w_chr                   ; |
        mov     A,Dbln                  ; |
        lcall   nil             ; |
        mov     P0,#'/'                 ; |
        lcall   w_chr                   ; |
        mov     A,#20                   ; |
        lcall   nil             ; |
        mov     A,Dthn                  ; |
        lcall   nil             ; |
        mov     P0,#0D0h                ; |
        lcall   w_ins                   ; |
        lcall   delay1                  ; |
        djnz    Cntr,loop0              ;/
        mov     DPTR,#tpkdsk            ;\
        lcall   line1                   ; |
        mov     Char,#16        ; |
        lcall   tulis                   ; |
        mov     DPTR,#tphrke            ; |
        lcall   line2                   ; |
        mov     Char,#16        ; |
        lcall   tulis                   ; | tulis kondisi
sekarang
        mov     DPTR,#angka             ; | hari ke
        mov     P0,#0CBh                ; |
        lcall   w_ins                   ; |
        mov     A,Hari                  ; |
        lcall   nil             ; |
        mov     P0,#0D0h                ; |
        lcall   w_ins                   ; |
        lcall   delay2                  ;/
        lcall   ckhrpn                  ; cek hari=0 /
setpoint thd hari
        lcall   lcdclr                  ;\
        mov     DPTR,#tpshst            ; |
        lcall   line1                   ; |
        mov     Char,#16        ; |
        lcall   tulis                   ; |
        mov     DPTR,#angka             ; |
```

```
        mov     P0,#0C0h                ; |
        lcall   w_ins                   ; |
        mov     A,Adc0                  ; |
        mov     B,#10                   ; | display sensor
suhu
        div     AB                  ; |
        mov     Bufr,B                  ; |
        mov     B,#25                   ; |
        add     A,B                     ; |
        lcall   nil                 ; |
        mov     P0,#'.'                 ; |
        lcall   w_chr                   ; |
        mov     A,Bufr                  ; |
        lcall   wr_chr                  ;/
        mov     P0,#0C6h                ;\
        lcall   w_ins                   ; |
        mov     A,Stpn                  ; |
        jnz     loop1                   ; |
        mov     P0,#' '                 ; |
        lcall   w_chr                   ; |
        mov     P0,#'-'                 ; |
        lcall   w_chr                   ; |
        mov     P0,#'-'                 ; |
        lcall   w_chr                   ; |
        ljmp    loop2                   ; |
loop1:  mov     B,#10                   ; | display seting
point suhu
        div     AB                  ; |
        mov     Bufr,B                  ; |
        mov     B,#25                   ; |
        add     A,B                     ; |
        lcall   nil                 ; |
        mov     P0,#'.'                 ; |
        lcall   w_chr                   ; |
        mov     A,Bufr                  ; |
        lcall   wr_chr                  ;/
loop2:  mov     P0,#0CDh                ;\
        lcall   w_ins                   ; |
        jb      Fan1,loop3          ; |
        mov     P0,#'1'                 ; |
        lcall   w_chr                   ; |
        ljmp    loop4                   ; |
loop3:  mov     P0,#'0'                 ; |
        lcall   w_chr                   ; |
loop4:  mov     P0,#0CEh                ; |
        lcall   w_ins                   ; |
        jb      Heat,loop5          ; |
        mov     P0,#'1'                 ; |
        lcall   w_chr                   ; | display status
        ljmp    loop6                   ; | actuator on/off
```

```
loop5:  mov     P0,#'0'                 ; |
        lcall   w_chr                   ; |
loop6:  mov     P0,#0CFh                ; |
        lcall   w_ins                   ; |
        jb      Fan0,loop7          ; |
        mov     P0,#'1'                 ; |
        lcall   w_chr                   ; |
        ljmp    loop8                   ; |
loop7:  mov     P0,#'0'                 ; |
        lcall   w_chr                   ; |
loop8:  mov     P0,#0D0h                ; |
        lcall   w_ins                   ; |
        lcall   delay2                  ;/
        lcall   lcdclr                  ;\
        mov     DPTR,#tphmst            ; |
        lcall   line1                   ; |
        mov     Char,#16            ; |
        lcall   tulis                   ; |
        mov     DPTR,#angka             ; |
        mov     P0,#0C1h                ; | display sensor
humi
        lcall   w_ins                   ; |
        mov     A,Adc1                  ; |
        mov     B,#2                    ; |
        div     AB                  ; |
        lcall   nil                 ;/
        mov     P0,#0C7h                ;\
        lcall   w_ins                   ; |
        mov     A,Stpn                  ; |
        jnz     loop9                   ; |
        mov     P0,#'-'                 ; |
        lcall   w_chr                   ; |
        mov     P0,#'-'                 ; |
        lcall   w_chr                   ; |
        ljmp    loopA                   ; |
loop9:  mov     A,Sthm                          ; | display
seting
        mov     B,#2                    ; | point humi
        div     AB                  ; |
        lcall   nil                 ;/
loopA:  mov     P0,#0CDh                ;\
        lcall   w_ins                   ; |
        jb      Fan1,loopB          ; |
        mov     P0,#'1'                 ; |
        lcall   w_chr                   ; |
        ljmp    loopC                   ; |
loopB:  mov     P0,#'0'                 ; |
        lcall   w_chr                   ; |
loopC:  mov     P0,#0CEh                ; |
        lcall   w_ins                   ; |
```

```
        jb      Heat,loopD          ; |
        mov     P0,#'1'                     ; |
        lcall   w_chr                       ; | display status
        ljmp    loopE                       ; | actuator on/off
loopD:  mov     P0,#'0'                     ; |
        lcall   w_chr                       ; |
loopE:  mov     P0,#0CFh                    ; |
        lcall   w_ins                       ; |
        jb      Fan0,loopF          ; |
        mov     P0,#'1'                     ; |
        lcall   w_chr                       ; |
        ljmp    loopG                       ; |
loopF:  mov     P0,#'0'                     ; |
        lcall   w_chr                       ; |
loopG:  mov     P0,#0D0h                    ; |
        lcall   w_ins                       ; |
        lcall   delay2                      ;/
;
        lcall   ckgnhr                      ; cek ganti hari
        lcall   rakmrg                      ; cek jam rak miring
        ljmp    mulai
;
setime: lcall   lcdclr
        mov     DPTR,#tpstjm
        lcall   line1
        mov     Char,#16
        lcall   tulis
stime0: lcall   bc_jam
        mov     DPTR,#angka
        mov     P0,#0C1h
        lcall   w_ins
        mov     A,Djam
        lcall   nil
        mov     P0,#':'
        lcall   w_chr
        mov     A,Dmnt
        lcall   nil
        mov     P0,#' '
        lcall   w_chr
        mov     A,Dtgl
        lcall   nil
        mov     P0,#'/'
        lcall   w_chr
        mov     A,Dbln
        lcall   nil
        mov     P0,#'/'
        lcall   w_chr
        mov     A,Dthn
        lcall   nil
        mov     P0,#0D0h
```

```
        lcall   w_ins
        lcall   tg_lps
stime1: lcall   scnkpd
        cjne    R0,#10,stime2
        ljmp    stime1
stime2: cjne    R0,#11,stime3
        mov     SP,#07h
        ljmp    mulai
stime3: cjne    R0,#12,stime4
        ljmp    stime1
stime4: cjne    R0,#13,stime5
        ljmp    stime1
stime5: cjne    R0,#14,stime6
        ljmp    stime1
stime6: cjne    R0,#15,stime7
        ljmp    stime1
stime7: cjne    R0,#16,stime8
        mov     R1,#1
        ljmp    stpnsh
stime8: mov     P0,#0C1h
        lcall   w_ins
        mov     Buf0,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        lcall   tg_tkn
        mov     Buf1,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        mov     A,Buf0
        RL      A
        RL      A
        RL      A
        RL      A
        orl     A,Buf1
        mov     Djam,A
        mov     P0,#':'
        lcall   w_chr
        lcall   tg_tkn
        mov     Buf0,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        lcall   tg_tkn
        mov     Buf1,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        mov     A,Buf0
```

```
RL      A
RL      A
RL      A
RL      A
orl     A,Buf1
mov     Dmnt,A
mov     P0,#' '
lcall   w_chr
lcall   tg_tkn
mov     Buf0,R0
mov     A,R0
lcall   wr_chr
lcall   tg_lps
lcall   tg_tkn
mov     Buf1,R0
mov     A,R0
lcall   wr_chr
lcall   tg_lps
mov     A,Buf0
RL      A
RL      A
RL      A
RL      A
orl     A,Buf1
mov     Dtgl,A
mov     P0,#'/'
lcall   w_chr
lcall   tg_tkn
mov     Buf0,R0
mov     A,R0
lcall   wr_chr
lcall   tg_lps
lcall   tg_tkn
mov     Buf1,R0
mov     A,R0
lcall   wr_chr
lcall   tg_lps
mov     A,Buf0
RL      A
RL      A
RL      A
RL      A
orl     A,Buf1
mov     Dbln,A
mov     P0,#'/'
lcall   w_chr
lcall   tg_tkn
mov     Buf0,R0
mov     A,R0
lcall   wr_chr
```

```
        lcall   tg_lps
        lcall   tg_tkn
        mov     Buf1,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        mov     A,Buf0
        RL      A
        RL      A
        RL      A
        RL      A
        orl     A,Buf1
        mov     Dthn,A
        mov     P0,#0D0h
        lcall   w_ins
;
        mov     A,#0D0h                 ; address DS1207
write address
        lcall   adrtx                   ; send
        mov     A,#000h                 ; register second
        lcall   dtatx                   ; send
        mov     A,#00                   ; data second (time)
        lcall   dtatx                   ; send
        mov     A,Dmnt                  ; data minute (time)
        lcall   dtatx                   ; send
        mov     A,Djam                  ; data hour (time)
        lcall   dtatx                   ; send
        mov     A,Dhri                  ; data day (time)
        lcall   dtatx                   ; send
        mov     A,Dtgl                  ; data date (time)
        lcall   dtatx                   ; send
        mov     A,Dbln                  ; data month (time)
        lcall   dtatx                   ; send
        mov     A,Dthn                  ; data year (time)
        lcall   dtatx                   ; send
        lcall   i2cstp                  ; i2c stop
        ljmp    stime0
;
stpnsh: lcall   lcdclr                  ; seting point suhu
        mov     DPTR,#tpstps
        lcall   line1
        mov     Char,#16
        lcall   tulis
stpsh0: cjne    R1,#1,stpsh1
        mov     DPTR,#tpmgk0
        mov     Stsh,#135               ; 38.5
stpsh1: cjne    R1,#2,stpsh2
        mov     DPTR,#tpmgk1
        mov     Stsh,#140               ; 39.0
stpsh2: cjne    R1,#3,stpsh3
```

```
        mov     DPTR,#tpmgk2
        mov     Stsh,#145               ; 39.5
stpsh3: cjne    R1,#4,stpsh4
        mov     DPTR,#tpmgk3
        mov     Stsh,#150               ; 40.0
stpsh4: cjne    R1,#5,stpsh5
        mov     DPTR,#tpmgk4
        mov     Stsh,#155               ; 40.5
stpsh5: lcall   line2
        mov     Char,#16
        lcall   tulis
        mov     DPTR,#angka
        mov     P0,#0CAh
        lcall   w_ins
        mov     A,Stsh
        mov     B,#10
        div     AB
        mov     Bufr,B
        mov     B,#25
        add     A,B
;
        mov     B,#10
        div     AB
        lcall   wr_chr
        mov     A,B
        lcall   wr_chr
        mov     P0,#'.'
        lcall   w_chr
        mov     A,Bufr
        lcall   wr_chr
        mov     P0,#0DFh
        lcall   w_chr
        mov     P0,#'C'
        lcall   w_chr
        lcall   tg_lps
stpsh6: lcall   scnkpd
        cjne    R0,#11,stpsh7
        mov     SP,#07h
        ljmp    mulai
stpsh7: cjne    R0,#15,stpsh9
        dec     R1
        cjne    R1,#0,stpsh8
        mov     R1,#1
        ljmp    setime
stpsh8: ljmp    stpsh0
stpsh9: cjne    R0,#16,stpsh6
        inc     R1
        cjne    R1,#6,stpshA
        ljmp    stpnhm
stpshA: ljmp    stpsh0
```

```
;
stpnhm: lcall   lcdclr
        mov     DPTR,#tpstph
        lcall   line1
        mov     Char,#16
        lcall   tulis
        mov     DPTR,#angka
        mov     P0,#0C5h
        lcall   w_ins
        mov     A,Sthm
        mov     B,#2
        div     AB
        lcall   nil
        mov     P0,#' '
        lcall   w_chr
        mov     P0,#025h
        lcall   w_chr
        mov     P0,#'R'
        lcall   w_chr
        mov     P0,#'H'
        lcall   w_chr
        mov     P0,#0D0h
        lcall   w_ins
        lcall   tg_lps
stphm1: lcall   scnkpd
        cjne    R0,#11,stphm2
        mov     SP,#07h
        ljmp    mulai
stphm2: cjne    R0,#15,stphm3
        mov     R1,#5
        ljmp    stpnsh
stphm3: cjne    R0,#16,stphm1
;
stknds: lcall   lcdclr
        mov     DPTR,#tpstkd
        lcall   line1
        mov     Char,#16
        lcall   tulis
        mov     DPTR,#tpstmg
        lcall   line2
        mov     Char,#16
        lcall   tulis
        lcall   rd_mem
stknd0: mov     DPTR,#angka
        mov     P0,#0C7h
        lcall   w_ins
        mov     A,Sttg
        lcall   nil
        mov     P0,#'/'
        lcall   w_chr
```

```
        mov     A,Stbl
        lcall   nil
        mov     P0,#'/'
        lcall   w_chr
        mov     A,Stth
        lcall   nil
        mov     P0,#0D0h
        lcall   w_ins
        lcall   tg_lps
stknd1: lcall   scnkpd
        cjne    R0,#10,stknd2
        ljmp    stknd1
stknd2: cjne    R0,#11,stknd5
        mov     A,Sttg
        jnz     stknd3
        mov     A,Stbl
        jnz     stknd3
        mov     A,Stth
        jnz     stknd3
        mov     Hari,#0
        ljmp    stknd4
stknd3: mov     A,Hari
        jnz     stknd4
        inc     Hari
stknd4: lcall   wr_mem
        mov     SP,#07h
        ljmp    mulai
stknd5: cjne    R0,#12,stknd6
        ljmp    stknd1
stknd6: cjne    R0,#13,stknd7
        ljmp    stknd1
stknd7: cjne    R0,#14,stknd8
        ljmp    stknd1
stknd8: cjne    R0,#15,stknd9
        ljmp    stpnhm
stknd9: cjne    R0,#16,stkndA
        ljmp    stknd1
stkndA: mov     P0,#0C7h
        lcall   w_ins
        lcall   tg_tkn
        mov     Buf0,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        lcall   tg_tkn
        mov     Buf1,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        mov     A,Buf0
```

```
        mov     B,#10
        mul     AB
        mov     B,Buf1
        add     A,B
        mov     Sttg,A
        mov     P0,#'/'
        lcall   w_chr
        lcall   tg_tkn
        mov     Buf0,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        lcall   tg_tkn
        mov     Buf1,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        mov     A,Buf0
        mov     B,#10
        mul     AB
        mov     B,Buf1
        add     A,B
        mov     Stbl,A
        mov     P0,#'/'
        lcall   w_chr
        lcall   tg_tkn
        mov     Buf0,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        lcall   tg_tkn
        mov     Buf1,R0
        mov     A,R0
        lcall   wr_chr
        lcall   tg_lps
        mov     A,Buf0
        mov     B,#10
        mul     AB
        mov     B,Buf1
        add     A,B
        mov     Stth,A
        ljmp    stknd0
;
kalbrs: lcall   lcdclr
        mov     DPTR,#tpadsh
        lcall   line1
        mov     Char,#16
        lcall   tulis
        lcall   tg_lps
        mov     DPTR,#angka
```

```
kalbr0: lcall   bcsns0
        mov     P0,#0C1h
        lcall   w_ins
        mov     A,Adc0
        lcall   nilai
        mov     P0,#0C9h
        lcall   w_ins
        mov     A,Adc0
        mov     B,#10
        div     AB
        mov     Bufr,B
        mov     B,#25
        add     A,B
        lcall   nil
        mov     P0,#'.'
        lcall   w_chr
        mov     A,Bufr
        lcall   wr_chr
        mov     P0,#0DFh
        lcall   w_chr
        mov     P0,#'C'
        lcall   w_chr
        mov     P0,#0D0h
        lcall   w_ins
        mov     Dly3,#1
        lcall   delay3
kalbr1: lcall   scnkpd
        cjne    R0,#11,kalbr2
        mov     SP,#07h
        ljmp    mulai
kalbr2: cjne    R0,#15,kalbr3
        ljmp    kalbr1
kalbr3: cjne    R0,#16,kalbr0
;
kalbr4: lcall   lcdclr
        mov     DPTR,#tpadhm
        lcall   line1
        mov     Char,#16
        lcall   tulis
        lcall   tg_lps
        mov     DPTR,#angka
kalbr5: lcall   bcsns1
        mov     P0,#0C1h
        lcall   w_ins
        mov     A,Adc1
        lcall   nilai
        mov     P0,#0C9h
        lcall   w_ins
        mov     A,Adc1
        mov     B,#2
```

```
        div     AB
        lcall   nil
        mov     P0,#' '
        lcall   w_chr
        mov     P0,#025h
        lcall   w_chr
        mov     P0,#'R'
        lcall   w_chr
        mov     P0,#'H'
        lcall   w_chr
        mov     P0,#0D0h
        lcall   w_ins
        mov     Dly3,#1
        lcall   delay3
kalbr6: lcall   scnkpd
        cjne    R0,#11,kalbr7
        mov     SP,#07h
        ljmp    mulai
kalbr7: cjne    R0,#15,kalbr8
        ljmp    kalbrs
kalbr8: cjne    R0,#16,kalbr5
;
posrak: lcall   lcdclr
        mov     DPTR,#tppsrk
        lcall   line1
        mov     Char,#16
        lcall   tulis
        mov     DPTR,#tpkrkn
        lcall   line2
        mov     Char,#16
        lcall   tulis
        lcall   tg_lps
psrak0: lcall   scnkpd
        cjne    R0,#11,psrak1
        mov     SP,#07h
        ljmp    mulai
psrak1: cjne    R0,#01,psrak5
        clr     Mrkr
        setb    Mrkn
psrak2: lcall   delay0
        jb      Lmkr,psrak3
        ljmp    psrak4
psrak3: lcall   scnkpd
        cjne    R0,#10,psrak2
psrak4: setb    Mrkr
        setb    Mrkn
;
psrak5: cjne    R0,#02,psrak9
        setb    Mrkr
        clr     Mrkn
```

```
psrak6: lcall   delay0
        jb      Lmkn,psrak7
        ljmp    psrak8
psrak7: lcall   scnkpd
        cjne    R0,#10,psrak6
psrak8: setb    Mrkr
        setb    Mrkn
;
psrak9: cjne    R0,#15,psrakA
        ljmp    kalbr4
psrakA: cjne    R0,#16,psrak0
        ljmp    psrak0
;
ckgnhr: mov     A,Dhri                          ; cek ganti
hari
        mov     B,Ghri
        clr     C
        subb    A,B
        jz      ckghr1
        mov     A,Sttg
        jnz     ckghr0
        mov     A,Stbl
        jnz     ckghr0
        mov     A,Stth
        jnz     ckghr0
        setb    Smrg                    ; if 00/00/00 gk
boleh miring
        ljmp    ckghr1
ckghr0: mov     Ghri,Dhri
        inc     Hari
        lcall   wr_mem
ckghr1: ret
;
ckhrpn: mov     A,Hari                          ; cek max
hari
        cjne    A,#23,ckhpn0
        mov     Sttg,#00        ;\
        mov     Stbl,#00        ; |
        mov     Stth,#00        ; | reset setelah semua
kondisi
        mov     Hari,#00        ; |
        lcall   wr_mem                  ;/
        ljmp    ckhpn1
ckhpn0: mov     DPTR,#lkpstp                    ;\
        movc    A,@A+DPTR               ; | seting point suhu
thd Hari
        mov     Stpn,A                  ;/
ckhpn1: ret
;
```

```
stblsh: lcall   bcsns0                  ; stabil suhu / jaga
suhu
        mov     A,Stpn
        jz      rstact
        mov     A,Adc0
        mov     B,Stpn
        div     AB
        jnz     stbsh0
        clr     Heat
        lcall   ttjnd0
        setb    Fan0
        ljmp    stblhm
stbsh0: mov     A,Adc0
        mov     B,Stpn
        clr     C
        subb    A,B
        jnz     stbsh1
        setb    Heat
        lcall   ttjnd0
        setb    Fan0
        ljmp    stblhm
stbsh1: setb    Heat
        lcall   bkjnd0
        clr     Fan0
;
stblhm: lcall   bcsns1                  ; stabil humu / jaga
humi
        mov     A,Adc1
        mov     B,Sthm
        div     AB
        jnz     stbhm0
        lcall   bkjnd1
        clr     Fan1
        ljmp    rsact
stbhm0: lcall   ttjnd1
        setb    Fan1
        ljmp    rsact
;
rstact: setb    Heat                    ;\
        setb    Fan0                    ; | matikan heater
        setb    Fan1                    ; | semua fan
        lcall   ttjnd0                  ; | tutup semua
jendela
        lcall   ttjnd1                  ;/
rsact:  ret
;
rakmrg: jb      Smrg,rkmrg4
        mov     A,Djam
        cjne    A,#00,rkmrg0
        mov     A,Dmnt
```

```
        cjne    A,#00,rkmrg0
        ljmp    rkmrg3
rkmrg0: mov     A,Djam
        cjne    A,#06,rkmrg1
        mov     A,Dmnt
        cjne    A,#00,rkmrg1
        ljmp    rkmrg3
rkmrg1: mov     A,Djam
        cjne    A,#12,rkmrg2
        mov     A,Dmnt
        cjne    A,#00,rkmrg2
        ljmp    rkmrg3
rkmrg2: mov     A,Djam
        cjne    A,#18,rkmrg4
        mov     A,Dmnt
        cjne    A,#00,rkmrg4
        ljmp    rkmrg3
rkmrg3: lcall   pndpss
        setb    Smrg
rkmrg4: ret
;
bcsns0: mov     A,#90h                  ; address
PCF8591 write address
        lcall   adrtx                   ; kirim
        mov     A,#00h                  ; adc ch-0
        lcall   dtatx                   ; kirim
        mov     A,#91h                  ; address PCF8591
read address
        lcall   adrtx                   ; kirim
        lcall   dtarx                   ; baca data
        lcall   givack                  ; beri ack
        lcall   dtarx                   ; baca data
        lcall   i2cstp                  ; i2c stop
        mov     Adc0,A                  ; simpan data
sensor 0
        ret
;
bcsns1: mov     A,#90h                  ; address
PCF8591 write address
        lcall   adrtx                   ; kirim
        mov     A,#01h                  ; adc ch-1
        lcall   dtatx                   ; kirim
        mov     A,#91h                  ; address PCF8591
read address
        lcall   adrtx                   ; kirim
        lcall   dtarx                   ; baca data
        lcall   givack                  ; beri ack
        lcall   dtarx                   ; baca data
        lcall   i2cstp                  ; i2c stop
```

```
        mov     Adc1,A                  ; simpan data
sensor 1
        ret
;
bc_jam: mov     A,#0D0h                 ; address
DS1207 write address
        lcall   adrtx                   ; send
        mov     A,#000h                 ; register second
        lcall   dtatx                   ; send
        mov     A,#0D1h                 ; address DS1207
read address
        lcall   adrtx                   ; send
        lcall   dtarx                   ; read data
        lcall   bcddec                  ; convert bcd to dec
        mov     Ddtk,A                  ; save second (time)
        lcall   givack                  ; send ack
        lcall   dtarx                   ; read data
        lcall   bcddec                  ; convert bcd to dec
        mov     Dmnt,A                  ; save minute (time)
        lcall   givack                  ; send ack
        lcall   dtarx                   ; read data
        lcall   bcddec                  ; convert bcd to dec
        mov     Djam,A                  ; save hour (time)
        lcall   givack                  ; send ack
        lcall   dtarx                   ; read data
        lcall   bcddec                  ; convert bcd to dec
        mov     Dhri,A                  ; save hour (time)
        lcall   givack                  ; send ack
        lcall   dtarx                   ; read data
        lcall   bcddec                  ; convert bcd to dec
        mov     Dtgl,A                  ; save date (time)
        lcall   givack                  ; send ack
        lcall   dtarx                   ; read data
        lcall   bcddec                  ; convert bcd to dec
        mov     Dbln,A                  ; save month (time)
        lcall   givack                  ; send ack
        lcall   dtarx                   ; read data
        lcall   bcddec                  ; convert bcd to dec
        mov     Dthn,A                  ; save year (time)
        lcall   i2cstp                  ; i2c stop
        lcall   delay0
ckmnt0: mov     A,Dmnt
        cjne    A,#00,ckmnt1
        ljmp    ckmnt2
ckmnt1: clr     Smrg
ckmnt2: ret
;
bcddec: mov     R1,A
        anl     A,#11110000b
        RR      A
```

```
        RR      A
        RR      A
        RR      A
bcddc0: cjne    A,#0,bcddc1
        mov     A,R1
        ljmp    bcddc6
bcddc1: cjne    A,#1,bcddc2
        mov     A,R1
        mov     B,#6
        clr     C
        subb    A,B
        ljmp    bcddc6
bcddc2: cjne    A,#2,bcddc3
        mov     A,R1
        mov     B,#12
        clr     C
        subb    A,B
        ljmp    bcddc6
bcddc3: cjne    A,#3,bcddc4
        mov     A,R1
        mov     B,#18
        clr     C
        subb    A,B
        ljmp    bcddc6
bcddc4: cjne    A,#4,bcddc5
        mov     A,R1
        mov     B,#24
        clr     C
        subb    A,B
        ljmp    bcddc6
bcddc5: cjne    A,#5,bcddc6
        mov     A,R1
        mov     B,#30
        clr     C
        subb    A,B
bcddc6: ret
;
wr_mem: mov     A,#0D0h                 ; AT24C16
write address
        lcall   adrtx
        mov     A,#008h                 ; address memory
        lcall   dtatx
        mov     A,Hari                  ; data memory
        lcall   dtatx
        mov     A,Sttg                  ; data memory
        lcall   dtatx
        mov     A,Stbl                  ; data memory
        lcall   dtatx
        mov     A,Stth                  ; data memory
        lcall   dtatx
```

```
        lcall   i2cstp                  ; i2c stop
        lcall   delay0
        ret
;
rd_mem: mov     A,#0D0h                 ; AT24C16
write address
        lcall   adrtx
        mov     A,#008h                 ; address memory
        lcall   dtatx
        mov     A,#0D1h                 ; AT24C16 read
address
        lcall   adrtx
        lcall   dtarx                   ; terima data
        mov     Hari,A
        lcall   givack                  ; beri ack
        lcall   dtarx                   ; terima data
        mov     Sttg,A
        lcall   givack                  ; beri ack
        lcall   dtarx                   ; terima data
        mov     Stbl,A
        lcall   givack                  ; beri ack
        lcall   dtarx                   ; terima data
        mov     Stth,A
        lcall   i2cstp                  ; i2c stop
        lcall   delay0
        ret
;
adrtx:  lcall   i2cstr                  ; kirim address
        lcall   putbit                  ; kirim data
        ret                             ; back
;
dtatx:  lcall   putbit                  ; kirim data
        ret                             ; back
;
dtarx:  lcall   getbit                  ; terima data
        ret                             ; back
;
putbit: mov     R7,#8                   ;\
putbt:  RLC     A                       ; |
        mov     ISDA,C                  ; |
        setb    ISCL                    ; |
        nop                             ; |
        clr     ISCL                    ; | kirim bit
        djnz    R7,putbt        ; |
        setb    ISDA                    ; |
        lcall   getack                  ; |
        ret                             ;/
;
getbit: mov     R7,#8                   ;\
getbt:  setb    ISCL                    ; |
```

```
        nop                             ; |
        mov     C,ISDA                  ; |
        RLC     A                   ; | terima bit
        clr     ISCL                    ; |
        djnz    R7,getbt            ; |
        setb    ISDA                    ; |
        ret                             ;/
;
getack: setb    ISDA                    ;\
        setb    ISCL                    ; |
ackbit: mov     C,ISDA                      ; | tunggu
ack
        jc      ackbit                  ; | D=1, C=1D=0,
C=0
        clr     ISCL                    ; |
        ret                             ;/
;
givack: clr     ISDA                    ;\
        setb    ISCL                    ; |
        clr     ISCL                    ; | kirim ack ->
D=0, C=1, C=0, D=1
        setb    ISDA                    ; |
        ret                             ;/
;
i2cstr: setb    ISCL                    ;\
        setb    ISDA                    ; |
        clr     ISDA                    ; | i2c start -> C=1,
D=1, D=0, C=0
        clr     ISCL                    ; |
        ret                             ;/
;
i2cstp: clr     ISDA                    ;\
        setb    ISCL                    ; |
        setb    ISDA                    ; | i2c stop -> D=0,
C=1, D=1, C=0
        clr     ISCL                    ; |
        ret                             ;/
;
nilai:  mov     B,#100
        div     AB
        lcall   wr_chr
        mov     A,B
nil:    mov     B,#10
        div     AB
        lcall   wr_chr
        mov     A,B
        lcall   wr_chr
        ret
;
bkjnd0: jb      Sjd0,bkjd0              ; buka jendela 0
```

```
        setb    Sjd0
        clr     Jbk0
        setb    Jtt0
        mov     Dly3,#3
        lcall   delay3
        setb    Jbk0
        setb    Jtt0
bkjd0:  ret
;
ttjnd0: jnb     Sjd0,ttjd0
        clr     Sjd0
        setb    Jbk0
        clr     Jtt0
        mov     Dly3,#4
        lcall   delay3
        setb    Jbk0
        setb    Jtt0
ttjd0:  ret
;
bkjnd1: jb      Sjd1,bkjd1
        setb    Sjd1
        clr     Jbk1
        setb    Jtt1
        mov     Dly3,#3
        lcall   delay3
        setb    Jbk1
        setb    Jtt1
bkjd1:  ret
;
ttjnd1: jnb     Sjd1,ttjd1
        clr     Sjd1
        setb    Jbk1
        clr     Jtt1
        mov     Dly3,#4
        lcall   delay3
        setb    Jbk1
        setb    Jtt1
ttjd1:  ret
;
pndpss: mov     Cnt0,#100
        clr     Mrkr
        setb    Mrkn
pdpss0: lcall   delay0
        jb      Lmkr,pdpss1
        ljmp    pdpss2
pdpss1: mov     A,Cnt0
        jz      pdpss0
        djnz    Cnt0,pdpss0
        ljmp    pdpss0
pdpss2: setb    Mrkr
```

```
        setb    Mrkn
        mov     A,Cnt0
        jz      pdpss5
pdpss3: setb    Mrkr
        clr     Mrkn
pdpss4: lcall   delay0
        jb      Lmkn,pdpss4
        setb    Mrkr
        setb    Mrkn
pdpss5: ret
;
line1:  mov     P0,#080h
        lcall   w_ins
        ret
;
line2:  mov     P0,#0C0h
        lcall   w_ins
        ret
;
tulis:  clr     A
        lcall   wr_chr
        inc     DPTR
        djnz    Char,tulis
        ret
;
wr_chr: movc    A,@A+DPTR
        mov     P0,A
        lcall   w_chr
        ret
;
w_ins:  clr     Enbl
        clr     Rest
        setb    Enbl
        clr     Enbl
        lcall   delay0
        ret
;
w_chr:  clr     Enbl
        setb    Rest
        setb    Enbl
        clr     Enbl
        lcall   delay0
        ret
;
lcd_in: mov     Dly3,#1
        lcall   delay3
        mov     P0,#01h                 ; Display Clear
        lcall   w_ins
        mov     P0,#38h                 ; Function Set
        lcall   w_ins
```

```
        mov     P0,#0Dh                 ; Display On,
Cursor, Blink
        lcall   w_ins
        mov     P0,#06h                 ; Entry Mode
        lcall   w_ins
        mov     P0,#02h                 ; Cursor Home
        lcall   w_ins
        ret
;
lcdclr: mov     P0,#01h                         ; Display
Clear
        lcall   w_ins
        lcall   delay0
        lcall   delay0
        lcall   delay0
        ret
;
scnkpd: mov     R0,#10
        lcall   delay0
col1:   mov     P1,#11111110b
        mov     A,P1
c1b1:   cjne    A,#11101110b,c1b2
        mov     R0,#1
c1b2:   cjne    A,#11011110b,c1b3
        mov     R0,#2
c1b3:   cjne    A,#10111110b,c1b4
        mov     R0,#3
c1b4:   cjne    A,#01111110b,col2
        mov     R0,#13
;
col2:   mov     P1,#11111101b
        mov     A,P1
c2b1:   cjne    A,#11101101b,c2b2
        mov     R0,#4
c2b2:   cjne    A,#11011101b,c2b3
        mov     R0,#5
c2b3:   cjne    A,#10111101b,c2b4
        mov     R0,#6
c2b4:   cjne    A,#01111101b,col3
        mov     R0,#14
;
col3:   mov     P1,#11111011b
        mov     A,P1
c3b1:   cjne    A,#11101011b,c3b2
        mov     R0,#7
c3b2:   cjne    A,#11011011b,c3b3
        mov     R0,#8
c3b3:   cjne    A,#10111011b,c3b4
        mov     R0,#9
c3b4:   cjne    A,#01111011b,col4
```

```
        mov     R0,#15
;
col4:   mov     P1,#11110111b
        mov     A,P1
c4b1:   cjne    A,#11100111b,c4b2
        mov     R0,#11
c4b2:   cjne    A,#11010111b,c4b3
        mov     R0,#0
c4b3:   cjne    A,#10110111b,c4b4
        mov     R0,#12
c4b4:   cjne    A,#01110111b,back
        mov     R0,#16
back:   ret
;
tg_tkn: lcall   scnkpd
        lcall   delay0
tg_tk0: cjne    R0,#16,tg_tk1
        ljmp    tg_tkn
tg_tk1: cjne    R0,#15,tg_tk2
        ljmp    tg_tkn
tg_tk2: cjne    R0,#14,tg_tk3
        ljmp    tg_tkn
tg_tk3: cjne    R0,#13,tg_tk4
        ljmp    tg_tkn
tg_tk4: cjne    R0,#12,tg_tk5
        ljmp    tg_tkn
tg_tk5: cjne    R0,#11,tg_tk6
        ljmp    tg_tkn
tg_tk6: cjne    R0,#10,tg_tk7
        ljmp    tg_tkn
tg_tk7: ret
;
tg_lps: lcall   scnkpd
        lcall   delay0
        cjne    R0,#10,tg_lps
        ret
;
delay0: djnz    Dly0,delay0
        ret
;
delay1: lcall   scnkpd
        cjne    R0,#13,dely10
        ljmp    setime
dely10: cjne    R0,#14,dely11
        ljmp    kalbrs
dely11: djnz    Dly1,delay1
        lcall   stblsh
        ret
;
delay2: mov     Dly2,#15
```

```
dely2:  lcall   delay1
        djnz    Dly2,dely2
        ret
;
delay3: lcall   delay0
        djnz    Dly1,delay3
        djnz    Dly3,delay3
        ret
;
tpnama: DB      ' Ella Puspasari '
tpnims: DB      '   0610633025   '
tpjurs: DB      ' Teknik Elektro '
tpuniv: DB      'Brawijaya Malang'

tpkdsk: DB      'Kondisi Sekarang'
tphrke: DB      '   Hari ke      '
tpshst: DB      'Suhu  Stpn   FHF'
tphmst: DB      'Humi  Stpn   FHF'

tpstjm: DB      '   Seting Jam   '
tpstps: DB      ' Set Point Suhu '
tpmgk0: DB      'Hr 01-04:       '
tpmgk1: DB      'Hr 05-08:       '
tpmgk2: DB      'Hr 09-14:       '
tpmgk3: DB      'Hr 15-17:       '
tpmgk4: DB      'Hr 18-21:       '
tpstph: DB      ' Set Point Humi '
tpstkd: DB      ' Start  Kondisi '
tpstmg: DB      ' Hr 1    /  /   '
tpadsh: DB      ' ADC     Suhu   '
tpadhm: DB      ' ADC     Humi   '
tppsrk: DB      'Posisi Rak Telur'
tpkrkn: DB      '1:kiri   2:kanan'
angka:  DB      '0123456789      '
;
lkpstp: DB      000,135,135,135,135,140,140,140,140,145
                ; 000-009
        DB      145,145,145,145,150,150,150,150,155,155
                ; 010-019
        DB      155,155,000,000,000,000,000,000,000,000
                ; 020-029
;
        end
```

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA