

**MONITORING SUHU DAN TEKANAN PADA
DESTILASI MINYAK ATSIRI BERBASIS
ATMEGA 8535 DENGAN TAMPILAN PC**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh

Gelar Sarjana Teknik Elektro



Disusun oleh :

Nama : Asep Juliyansyah

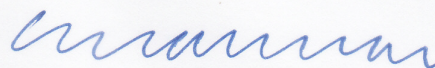
No.Mahasiswa : 04524058

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2012

LEMBAR PENGESAHAN PEMBIMBING

**MONITORING SUHU DAN TEKANAN
PADA DESTILASI MINYAK ATSIRI BERBASIS
ATMEGA 8535 DENGAN TAMPILAN PC**

TUGAS AKHIR**Oleh:****Nama : Asep Juliyansyah****No. Mahasiswa : 04524058****Yogyakarta, Februari 2012****Menyetujui,****Pembimbing I****Pembimbing II****Dwi Ana Ratnawati S.T, M.Eng.****Wahyudi Budi Pramono S.T, M.Eng.**

LEMBAR PENGESAHAN PENGUJI
MONITORING SUHU DAN TEKANAN PADA DESTILASI
MINYAK ATSIRI BERBASIS ATMEGA 8535 DENGAN
TAMPILAN PC

TUGAS AKHIR

Oleh:

Nama : Asep Juliyansyah

No. Mahasiswa : 04524058

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Maret 2012

Tim Penguji,

Wahyudi Budi Pramono, ST., M.Eng.

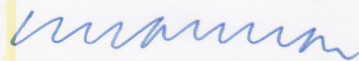
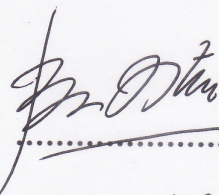
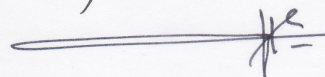
Ketua

Ir. Hj. Budi Astuti, MT.

Anggota I

Medilla Kusriyanto, ST., M.Eng.

Anggota II



Mengetahui,

Ketua Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia



Tito Yuwono, ST., M.Sc.

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Kupersembahkan karya ini untuk:

Orang tua dan keluargaku tercinta

Serta seseorang yang berada nan jauh di sana

yang telah mencurahkan kasih sayang, do'a dan dukungan_nya

yang telah mengajarkan keseimbangan dalam diriku

kesungguhan, ketekunan, kesabaran, welas asih dan cara mencintai makhluk

yang menjadikan ku merasa bersyukur diantara kelebihan dan kekuranganku

yang tetap membuatku ingat untuk selalu menunduk dihadapan – Nya

Semoga Allah SWT mencatatnya sebagai amal kebajikan.

Aamiin.

MOTTO

Jangan pernah mengingkari kalau kita adalah makhluk yang penuh dengan kekurangan, sesungguhnya dengan kekurangan kita akan belajar bagaimana untuk mencari kelebihan.

Jangan pernah menyesali kegagalan, dengan gagal kita akan tahu bagaimana rasanya keberhasilan.

Berusahalah untuk tidak menjadi manusia yang berhasil tapi berusahalah menjadi manusia yang berguna.

Jika kita ingin maju, makaajukanlah orang lain terlebih dahulu.

KATA PENGANTAR



Assalamu'alaikum warahmatullahi wabarakatuh,

Alhamdulillah *rabbi'l'alam*, segala puji syukur saya panjatkan kehadiran Allah SWT yang telah melimpahkan segala karunia-Nya, sehingga saya dapat menyelesaikan Tugas Akhir ini sebagai salah satu persyaratan program S-1 Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Islam Indonesia. Shalawat dan salam semoga tercurah pada junjungan kita Rasulullah Muhammad SAW beserta keluarga, dan pengikutnya yang setia hingga akhir zaman.

Merupakan suatu kelegaan ketika akhirnya saya dapat menyelesaikan bagian akhir dari amanah yang panjang ini dengan begitu banyak kemudahan dan kekuatan yang Allah berikan melalui banyak pihak dengan bantuan, dukungan dan do'a. Untuk itulah, saya sangat ingin menghaturkan terima kasih kepada :

1. Bapak Tito Yuwono, ST, M.Sc. selaku Ketua Jurusan Teknik Elektro Universitas Islam Indonesia.
2. Ibu Dwi Ana Ratna Wati, ST, M.Eng. selaku Pembimbing I Tugas Akhir yang telah meluangkan waktunya untuk memberikan masukan dan bantuan sampai terselesaikannya Tugas Akhir ini.

3. Bapak Wahyudi Budi Pramono, ST, M.Eng. selaku Pembimbing II Tugas Akhir yang telah meluangkan waktunya sampai terselesaikannya Tugas Akhir ini.
4. Seluruh Dosen Jurusan Teknik Elektro, terima kasih atas bimbingan selama saya kuliah di Jurusan Teknik Elektro, Universitas Islam Indonesia.
5. Ayahanda dan Ibunda serta Adinda-adindaku tercinta yang telah memberikan semangat, mendukung dan memberikan doanya tiada henti.
6. Mbak umi, mas heri, mas handri yang telah memberikan bantuan dan informasi
7. Anjar, Aris, Edy, Chandra, Aan, Alvin, Hendy, Hanung, Puguh, Yudi, dan semua teman-teman Elektro 04 yang telah membantu dan mendukung.
8. Mas Agus, mas Wira, mas Asief, Bayu, Dinno, Anggi semua teman-teman satu kosan B12 yang telah mendukung saya.
9. Teman-teman KKN dan juga teman-teman sekolah terima kasih atas dukungannya.
10. Dan banyak pihak lain yang tidak dapat kami sebutkan seluruhnya yang telah membantu dalam penyelesaian Tugas Akhir ini.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Hal ini disebabkan karena keterbatasan kemampuan dan kurangnya pengetahuan yang dimiliki penulis. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun dan mempunyai solusi dari

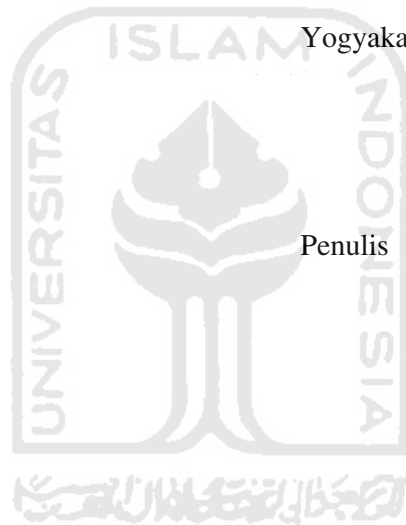
semua pembaca untuk kebaikan dan kesempurnaan tugas akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi siapa saja yang membutuhkan.

Wassalaamu'alaikum warahmatullahi wabarakatuh

Yogyakarta,

Februari 2012

Penulis



ABSTRAK

Minyak atsiri adalah cairan lembut, bersifat aromatik, dan mudah menguap pada suhu kamar yang berasal dari destilasi bahan-bahan seperti sereh, jahe, nilam dan lain-lain jenisnya. Agar mendapatkan hasil maksimal, maka proses destilasi minyak atsiri perlu dimonitoring. Dalam prosesnya, hal yang harus diperhatikan adalah suhu dan tekanan. Pada penelitian ini, komponen yang digunakan dalam merancang monitoring suhu dan tekanan adalah sensor LM35C yang mengubah panas ke bentuk tegangan, sensor MPX2100GP yang mengubah tekanan uap ke bentuk tegangan, ADC (*Analog to Digital Converter*) yang terintegrasi dengan ATmega8535 mengubah tegangan ke dalam bentuk digital, RS232 sebagai komunikasi serial penyangga keluaran ADC untuk dihubungkan ke port serial agar dapat diproses komputer dengan menggunakan *software Microsoft Visual Basic 6.0*. Monitoring suhu dan tekanan pada destilasi minyak atsiri berbasis ATmega8535 dengan tampilan PC dijalankan menggunakan program *Microsoft Visual Basic 6.0* dengan memanfaatkan komunikasi serial sebagai masukan dan keluaran sinyal (dua-arah). *Solenoid valve* yang terhubung dengan *relay* akan aktif (*on*) apabila tekanan mencapai *setpoint* yang telah ditentukan (55 kPa) dan *solenoid valve* akan mati (*off*) jika tekanan pada boiler destilasi dibawah 10 kPa. Pengukuran suhu pada penelitian ini memiliki toleransi *error* 0,823 °C dan pengukuran tekanan memiliki toleransi *error* 0,561 kPa. Hasil monitoring dapat ditampilkan berupa grafik *realtime* dan terekam melalui *database*.

Kata kunci : Destilasi, minyak atsiri, ATmega8535, LM35C, MPX2100GP, *solenoid valve*, *Microsoft Visual Basic 6.0*, *database*

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
HALAMAN PERSEMBAHAN.....	iv
MOTTO.....	v
KATA PENGANTAR.....	vi
ABSTRAK.....	ix
DAFTAR ISI	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	
2.1 Kajian Pustaka.....	6
2.2 Sensor Suhu dan Tekanan.....	8
2.2.1 LM35.....	8
2.2.2 MPX2100.....	10
2.3 Penguatan Operasional.....	12
2.3.1 Pengenalan Penguat Operasional.....	12
2.3.2 Rangkaian Penguat Operasional <i>Non-Inverting</i>	13
2.3.3 LM 358.....	14
2.4 Mikrokontroler.....	15
2.4.1 Mikrokontroler AVR ATMega8535.....	16
2.4.2 Konfigurasi Pin ATMega8535.....	17

2.4.3	Komunikasi Serial Dengan UART.....	18
a.	<i>Register</i> Data I/O UART (UDR).....	19
b.	<i>Register</i> Baud Rate UART (UBRR).....	19
c.	<i>Register</i> Status UART (USR).....	19
d.	<i>Register</i> Kontrol UART (UCR).....	20
2.4.4	<i>Analog to Digital Converter</i> (ADC).....	20
a.	ADCSRA (<i>ADC Control and Status Register A</i>). 22	
b.	SFIO (<i>Special Function Input/Output Register</i>). 23	
c.	ADCH (<i>ADC High-byte Data</i>).....	23
d.	ADCL (<i>ADC Low-byte Data</i>).....	23
2.5	<i>Relay</i>	28
2.5.1	Prinsip Kerja dan Simbol.....	29
2.6	<i>Solenoid Valve</i>	30
2.7	BASCOM-AVR.....	33
2.8	<i>Visual Basic</i> 6.0.....	34
BAB III	PERANCANGAN SISTEM	
3.1	Gambaran Umum Sistem.....	37
3.2	Perancangan Perangkat Keras.....	38
3.2.1	Rangkaian Catu Daya.....	38
3.2.2	Rangkaian Penguat/Op-Amp.....	40
3.2.3	Rangkaian <i>Driver Relay</i>	42
3.2.4	Rangkaian Mikrokontroler ATmega8535.....	43
3.2.5	Rangkaian Komunikasi Serial dengan RS 232.....	44
3.3	Perancangan Perangkat Lunak.....	45
3.3.1	<i>Basic Compiler</i> AVR.....	46
3.3.2	Pemrograman <i>Visual Basic</i>	50
3.3.3	Merancang <i>Database</i>	55
BAB IV	PENGUJIAN, ANALISIS, DAN PEMBAHASAN	
4.1	Metode Pengujian.....	59
4.2	Pengujian Rangkaian Catu Daya.....	60
4.3	Pengujian Rangkaian Sensor Suhu LM35C.....	61

4.4	Pengujian Rangkaian Sensor Tekanan MPX2100GP.....	62
4.5	Pengujian Rangkaian Saklar.....	65
4.6	Pengujian dan Pembahasan Sistem Keseluruhan.....	65

BAB V PENUTUP

5.1	Kesimpulan.....	70
5.2	Saran-saran.....	71

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR GAMBAR

Gambar 2.1 Fisik LM 35.....	8
Gambar 2.2 Kurva LM 35.....	10
Gambar 2.3 MPX2100.....	11
Gambar 2.4 Keluaran Tekanan <i>Differential</i>	11
Gambar 2.5 Simbol Penguat Operasional.....	12
Gambar 2.6 Rangkaian Tidak Membalik (<i>Non-Inverting Amplifier</i>).....	13
Gambar 2.7 Konfigurasi Pin Op-Amp LM 358.....	15
Gambar 2.8 Konfigurasi Pin ATmega8535.....	17
Gambar 2.9 <i>Register</i> Status UART.....	19
Gambar 2.10 <i>Register</i> Kendali UART.....	20
Gambar 2.11 Koneksi dengan <i>Filter</i> LC Pada Suplai ADC.....	21
Gambar 2.12 <i>Register</i> ADMUX.....	22
Gambar 2.13 <i>Register</i> ADCSRA.....	22
Gambar 2.14 <i>Register</i> SFIOR.....	23
Gambar 2.15 <i>Relay</i> yang Tersedia Di pasaran.....	28
Gambar 2.16 Skema <i>Relay</i> Elektromagnetik.....	29
Gambar 2.17 Rangkaian dan Simbol Logika <i>Relay</i>	30
Gambar 2.18 <i>Solenoid Valve</i>	31
Gambar 2.19 Struktur Fungsi <i>Solenoid Valve</i>	32
Gambar 2.20 Tampilan Jendela Program BASCOM-AVR.....	33
Gambar 2.21 Tampilan Simulasi BASCOM-AVR.....	34
Gambar 3.1 Blok Diagram Sistem.....	38
Gambar 3.2 Rangkaian Catu Daya +5V.....	39
Gambar 3.3 Rangkaian Catu Daya +10V.....	39
Gambar 3.4 Rangkaian Catu Daya +24V.....	39
Gambar 3.5 Rangkaian Penguat Untuk LM35C.....	40
Gambar 3.6 Rangkaian Penguat Untuk MPX2100GP.....	41
Gambar 3.7 Rangkaian <i>Driver Relay</i>	42

Gambar 3.8 Sistem Minimum ATmega8535.....	43
Gambar 3.9 Rangkaian RS 232.....	45
Gambar 3.10 <i>Flowchart</i> Program BASCOM.....	46
Gambar 3.11 <i>Compile</i> Program BASCOM.....	49
Gambar 3.12 Skema K-85.....	49
Gambar 3.13 Gambar <i>Software</i> AVRProg.....	50
Gambar 3.14 Layar Pemilihan Jenis Penguat.....	51
Gambar 3.15 Tampilan Standar <i>Project</i>	51
Gambar 3.16 <i>Flowchart</i> Program <i>Visual Basic</i>	53
Gambar 3.17 Program Monitoring Suhu dan Tekanan dengan <i>Visual Basic</i>	55
Gambar 3.18 Jendela VisData.....	55
Gambar 3.19 Membuat <i>Database</i>	56
Gambar 3.20 Membuat Tabel.....	56
Gambar 3.21 Memberi Nama Tabel.....	57
Gambar 3.22 Memasukkan Nama-nama <i>Field</i>	57
Gambar 3.23 Mengaktifkan Tabel Data Monitoring.....	58
Gambar 4.1 Alat Destilasi Minyak Atsiri.....	60
Gambar 4.2 Grafik Perbandingan Termometer dengan Sensor Suhu.....	62
Gambar 4.3 Grafik Perbandingan Pembacaan Tekanan dengan Sensor MPX2100GP.....	64
Gambar 4.4 Tampilan Awal Aplikasi Monitoring Suhu dan Tekanan Pada Destilasi Minyak Atsiri.....	66
Gambar 4.5 Aplikasi Pengaturan.....	67
Gambar 4.6 Tampilan Proses Monitoring Suhu dan Tekanan Pada Destilasi Minyak Atsiri.....	68
Gambar 4.7 Grafik Excel Monitoring Suhu dan Tekanan Pada Destilasi Minyak Atsiri.....	69
Gambar 4.8 <i>Database Microsoft Access</i>	69

DAFTAR TABEL

Tabel 2.1 Pemilihan Tegangan Referensi Pada ADC.....	24
Tabel 2.2 Konfigurasi Bit-bit ADMUX4:0.....	25
Tabel 2.3 Konfigurasi Bit-bit ADTS2:0.....	26
Tabel 2.4 Konfigurasi Bit-bit ADPS.....	27
Tabel 4.1 Hasil Pengukuran Tegangan Catu Daya.....	60
Tabel 4.2 Data Pengukuran Rangkaian LM35C.....	61
Tabel 4.3 Data Pengukuran Rangkaian MPX2100GP.....	63
Tabel 4.4 Data Pengukuran Kondisi Tekanan 0 kPa.....	64
Tabel 4.5 Data Hasil Pengujian <i>Relay</i>	65



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Minyak atsiri merupakan minyak yang berasal dari bahan-bahan antara lain berupa sereh wangi, akar wangi, nilam, cengkeh, pala, melati, dan lain-lain jenisnya. Minyak atsiri tersebut dapat dimanfaatkan sesuai kebutuhannya dan mempunyai harga jual yang tinggi. Pada destilasi minyak atsiri dapat dilakukan dengan alat-alat destilasi yang dirancang sedemikian rupa. Proses destilasi minyak atsiri perlu dimonitoring agar mendapatkan hasil yang maksimal. Dalam proses destilasi, hal yang harus diperhatikan adalah suhu dan tekanannya. Untuk itu dibuatlah alat pemantau tekanan dan suhu selama dalam pemrosesan.

Pada tugas akhir ini dibuat alat untuk memonitoring suhu dan tekanan pada destilasi minyak atsiri berbasis ATMega8535 dengan tampilan PC. Adapun komponen yang digunakan dalam merancang aplikasi tersebut adalah sensor suhu LM35C yang mengubah panas ke bentuk tegangan dan sensor tekanan tipe MPX2100 GP yang mengubah tekanan uap ke bentuk tegangan.

1.2 Rumusan Masalah

Pada pembuatan tugas akhir ini, maka dapat dirumuskan masalahnya adalah bagaimana merancang serta membuat alat dan perangkat lunak atau

software monitoring suhu dan tekanan pada destilasi minyak atsiri berbasis ATmega8535 dengan tampilan PC.

1.3 Batasan Masalah

Agar penulisan tugas akhir ini lebih terarah, maka perlu adanya batasan masalah yang dibahas. Diantaranya sebagai berikut :

1. Sensor suhu LM35C dan sensor tekanan MPX2100GP yang digunakan masing-masing hanya satu.
2. Sensor suhu mempunyai daya kerja dengan suhu maksimal 110°C dan sensor tekanan daya kerja dengan tekanan maksimal 100 kPa.
3. Untuk pengaturan *setpoint* terdapat pada tekanan saja.
4. Penggunaan ATmega8535 sebagai kendali utama.
5. Relay dan *solenoid valve* sebagai driver dan valve elektrik.
6. Sebagai penampil suhu dan tekanan menggunakan tampilan PC.

1.4 Tujuan Penelitian

Tujuan yang akan dicapai dalam penelitian Tugas Akhir ini adalah membangun sebuah alat serta perangkat lunak (*software*) monitoring suhu dan tekanan pada destilasi minyak atsiri berbasis ATmega 8535 dengan tampilan PC agar mendapatkan hasil yang maksimal dalam proses destilasi.

1.5 Metodologi penelitian

Untuk melaksanakan tugas akhir ini dilakukan langkah-langkah sebagai berikut :

1. Studi pustaka penelitian sejenis.
2. Perancangan sistem.
 - a) Perancangan perangkat keras (*hardware*) rangkaian penguat, sistem minimum mikrokontroler beserta penampilnya.
 - b) Perancangan perangkat lunak (*software*).
3. Pengujian dan analisa.
Melakukan pengujian dan pengambilan data terhadap rangkaian perangkat keras, serta melakukan analisa praktis terhadap hasil yang diperoleh.
4. Penulisan laporan.
Penulisan laporan dilakukan sesuai data yang diperoleh dari tugas akhir.

1.6 Sistematika Penulisan

Dari hasil penelitian yang telah dilaksanakan, sistematika penulisan laporannya adalah sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan berisi tentang Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Penulisan, Metode Penelitian, serta Sistematika Penulisan pada Laporan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab studi pustaka membahas tentang penelitian yang telah dilakukan sebelumnya dan penjelasan secara terperinci mengenai sensor suhu LM35C, sensor tekanan MPX2100GP, jenis penguatan operasional yang dipakai, mikrokontroler ATmega8535, komunikasi serial, *relay*, BASCOM-AVR, *Microsoft Visual Basic 6.0* yang digunakan sebagai landasan untuk pemecahan masalah.

BAB III PERANCANGAN SISTEM

Berisi pembahasan tentang perancangan perangkat keras yang meliputi sensor suhu dan tekanan sebagai bahan dasar pengukuran suhu dan tekanan pada proses destilasi minyak atsiri, operasional *amplifier*, *mikrokontroler* sebagai kendali utamanya, relay dan *solenoid valve* sebagai *driver* dan *valve* elektrik dan program untuk monitoring tampilan pada PC.

BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN

Bagian ini menjelaskan tentang pengujian catu daya, sensor suhu LM35C, sensor tekanan MPX2100GP, rangkaian *relay*, serta pengujian sistem keseluruhan dan analisa terhadap hasil pengamatan dari sistem yang telah dibuat.

BAB V PENUTUP

Bab penutup membahas tentang kesimpulan dan saran-saran yang menunjang agar alat tersebut dapat berkerja lebih optimal, sehingga tugas

akhir ini dapat dikembangkan lebih lanjut, dengan harapan dapat digunakan untuk mendukung penerapannya dalam kehidupan masyarakat.



BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian serupa yang telah dilakukan berjudul “*Monitoring Suhu Dan Tekanan Sistem Boiler Dengan Tampilan Pada PC*” (Fahlevy : 2006). Dalam penelitian ini alat yang digunakan dalam perancangan adalah sensor suhu LM35DZ yang mengubah panas ke bentuk tegangan dan sensor MPX 2010 GS yang mengubah tekanan uap ke bentuk tegangan, juga menggunakan ADC (*Analog to Digital Converter*) seri ADC 0801 dengan 8 bit yang mengubah tegangan ke dalam bentuk digital, *buffer* sebagai penyangga keluaran ADC untuk dihubungkan ke port paralel agar dapat diproses oleh komputer dengan menggunakan *software Microsoft Visual Basic 6.0*. Monitoring suhu dan tekanan sistem boiler dengan tampilan PC dijalankan menggunakan program *Microsoft Visual Basic 6.0* dengan memanfaatkan port paralel sebagai masukan dan keluaran *signal* (dwi-arah). Tegangan listrik pada elemen pemanas boiler akan terputus (*off*) apabila suhu atau tekanan mencapai batas yang telah ditentukan (100 °C / 0,8 kPa).

Penelitian serupa yang telah dilakukan berjudul “*Rancang Bangun Alat Ukur Suhu yang Terintegrasi Dengan Tekanan Darah Berbasis Mikrokontroler Atmega8535*”. (Maiyana, Eka : 2011). Dalam penelitian ini alat ukur suhu bekerja seperti termometer biasa yaitu mengukur suhu tubuh pada skala *celcius* dan dapat

mengukur dengan *range* pengukuran antara 33°C sampai 38°C . Suhu tubuh akan terbaca oleh sensor, kemudian tegangan keluaran dari sensor yang berupa data digital akan diproses oleh mikrokontroler ATmega 8535 dan ditampilkan kedalam LCD 2x16. Dari hasil pengambilan data pengukuran suhu tubuh dengan termometer standar dan alat ukur yang dibuat pada tugas akhir ini diperoleh nilai *error* sebesar 0.41, akurasi sebesar 0.988 dan dan presisi dari suatu alat sebesar 99.98 %.

Penelitian serupa yang telah dilakukan berjudul “*Perancangan Sistem Monitoring Suhu dan Tekanan Berbasis Web pada Mini Plant Di Laboratorium Workshop Instrumentasi Teknik Fisika ITS Surabaya*”. (Restu Budianto : 2010). Dalam penelitian ini *miniplant* yang digunakan adalah *miniplant* dengan variabel proses suhu dan workshop instrumentasi dengan mode kontrol *on off*. Jaringan yang digunakan adalah intranet, monitoring *process* pada web dibangun menggunakan PHP dan adobe dreamweaver CS3. Dimana berdasarkan perancangan sistem telah berhasil melakukan sistem monitoring dan tekanan pada ESD sistem dengan menggunakan jaringan LAN, dan web monitoring sistem ini mampu memonitoring dan menyimpan data dengan baik pada setiap perubahan nilai proses variabel dan kondisi *annouciator* tanpa terjadi kesalahan pada kondisi normal maupun kondisi *emergency*.

2.2 Sensor Suhu Dan Tekanan

2.2.1 LM 35

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika elektronika yang diproduksi oleh *National Semiconductor*. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan. Meskipun tegangan sensor ini dapat mencapai 30 volt akan tetapi yang diberikan ke sensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60 μA hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (*self-heating*) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5 $^{\circ}\text{C}$ pada suhu 25 $^{\circ}\text{C}$.



Gambar 2.1 Fisik LM 35

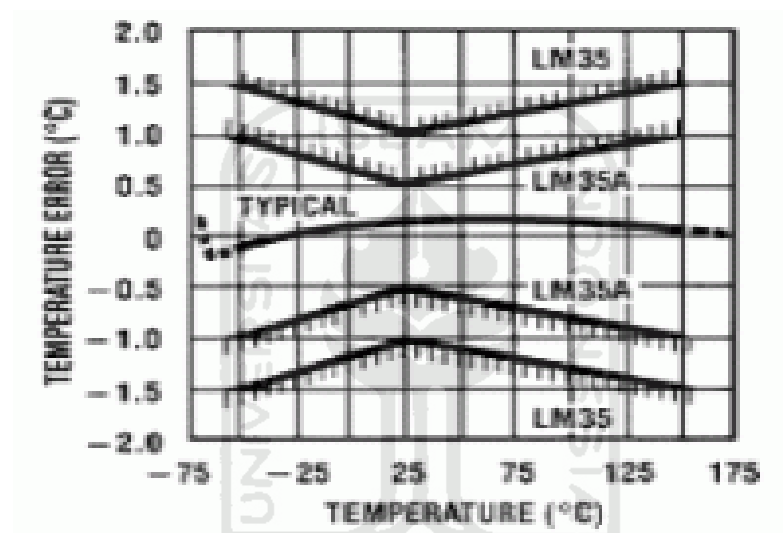
Gambar 2.1 menunjukkan tampak bawah dari LM35. 3 pin LM35 menunjukkan fungsi masing-masing pin diantaranya, pin 1 berfungsi sebagai sumber tegangan atau +Vs dari LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran atau Vout dengan jangkauan kerja dari 0 Volt sampai dengan 1,5 Volt dengan tegangan operasi sensor LM35 yang dapat digunakan antar 4 Volt sampai 30 Volt, dan pin 3 sebagai *Ground*.

Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut :

$$V_{LM35} = \text{Suhu} * 10 \text{ mV} \quad (2.1)$$

Secara prinsip sensor akan melakukan penginderaan pada saat perubahan suhu setiap suhu 1 °C akan menunjukkan tegangan sebesar 10 mV. Berikut ini adalah karakteristik dari sensor LM35 :

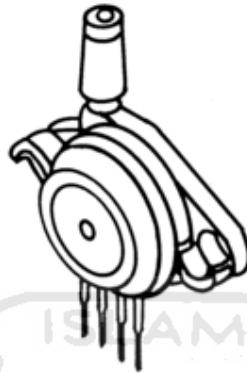
1. Memiliki sensitivitas suhu, dengan faktor skala linier antara tegangan dan suhu 10 mV/°C, sehingga dapat dikalibrasi langsung dalam *celcius*.
2. Memiliki ketepatan kalibrasi yaitu 0,5°C pada suhu 25 °C.
3. Memiliki jangkauan maksimal operasi suhu antara -55 °C sampai +150 °C.
4. Bekerja pada tegangan 4 sampai 30 volt.
5. Memiliki arus rendah yaitu kurang dari 60 µA.



Gambar 2.2 Kurva LM 35

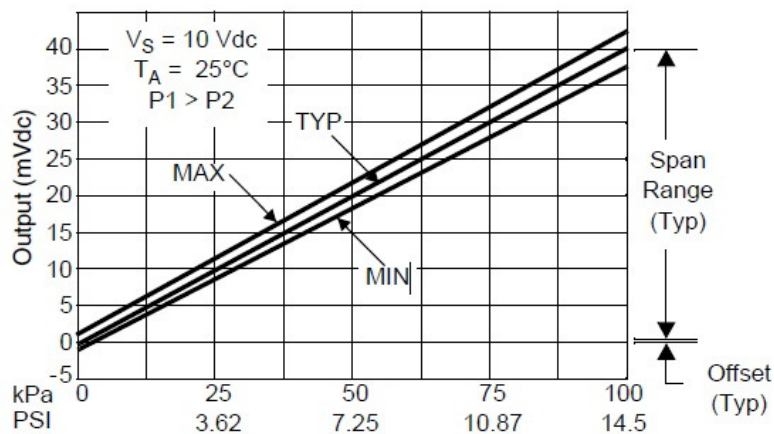
MPX2100 adalah sensor untuk mengukur tekanan udara, sensor ini juga banyak digunakan dalam bidang elektronik misalnya untuk *robotic*, keperluan medis dan masih banyak lagi. MPX2100 ini keluarannya masih sinyal analog, sensor ini mempunyai 4 pin, konfigurasi pinnya dapat dilihat pada gambar 2.3, untuk tegangan masukannya menggunakan 10 sampai 16 Vdc, sensor ini juga

mempunyai satuan kPa dengan maksimal hasil pengukurannya 100 kPa atau 14,5 psi.



Gambar 2.3 MPX 2100

Gambar 2.4 menunjukkan karakteristik keluaran (*output*) dari sensor MPX 2100 *series* pada suhu 25° C. Keluarannya secara proporsional langsung dengan tekanan diferensial dan pada dasarnya adalah sebuah garis lurus.

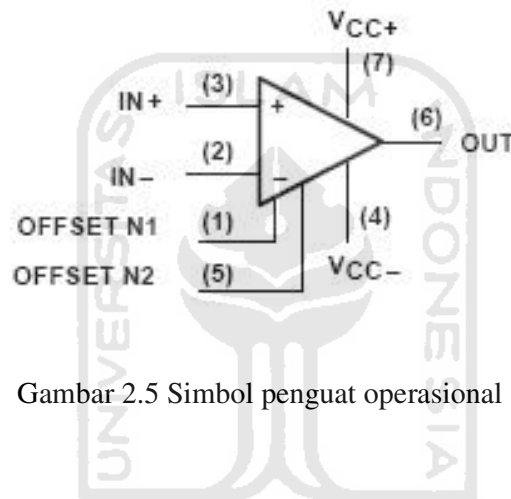


Gambar 2.4 Keluaran tekanan *differential*

2.3 Penguat Operasional

2.3.1 Pengenalan Penguat Operasional

Penguat operasional adalah penguat dengan *gain* tinggi dan terhubung secara langsung. Penguat ini biasanya digunakan untuk menguatkan sinyal berjalan frekuensi lebar dan digunakan bersama jaringan umpan balik eksternal.



Gambar 2.5 Simbol penguat operasional

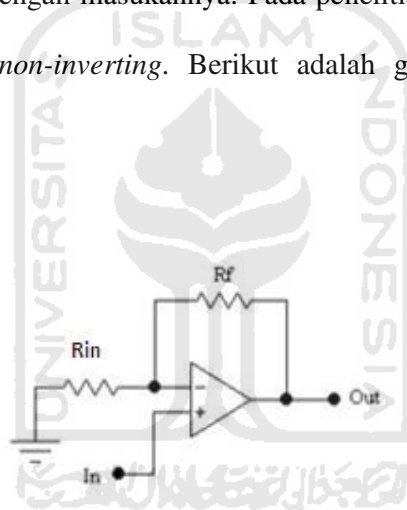
Penguat operasional memiliki lima terminal dasar, yaitu dua untuk mensuplai daya, dua untuk masukan dan satu untuk keluaran. Penguat operasional memiliki beberapa ketentuan ideal. Ketentuan-ketentuan ini tidak terdapat dalam praktek, namun asumsi-asumsi mengenai ini memungkinkan orang untuk melakukan analisa dengan cepat mengenai rangkaian umpan balik pada penguat operasional. Ketentuan-ketentuan ideal penguat operasional ini, antara lain :

1. Penguatan open loop besar ($A \gg$).
2. $V_o = 0$ jika $V_1 = V_2$.
3. Impedansi input tinggi ($Z \gg$).

4. Impedansi output mendekati 0 ($Z_o \ll$).
5. Tidak ada tegangan offset.

2.3.2 Rangkaian Penguat Operasional *Non-Inverting*

Sering kali dibutuhkan penguat yang memberikan keluaran sama besar dan sefasa dengan masukannya serta memenuhi hubungan R_f tertentu dengan R_i . Oleh karena itu digunakannya rangkaian *non inverting amplifier* untuk memperoleh penguatan yang sefasa dengan masukannya. Pada penelitian ini, digunakan jenis penguatan operasional *non-inverting*. Berikut adalah gambar rangkaian *non-inverting amplifier*:



Gambar 2.6 Rangkaian tidak membalik (*non-inverting amplifier*)

Dengan asumsi tegangan antara tegangan terminal *inverting* (-) dan *non-inverting* (+) adalah 0 volt, berarti tegangan di titik A sama dengan V_i . Arus yang mengalir pada R_i sama dengan arus yang mengalir pada R_f , yaitu:

$$I = \frac{V_{in}}{R_{in}} \quad (2.2)$$

$$V_{out} = \left[1 + \frac{R_f}{R_{in}} \right] V_{in} \quad (2.3)$$

atau,

$$\frac{V_{out}}{V_{in}} = \left[1 + \frac{R_f}{R_{in}} \right] \quad (2.4)$$

Dengan :

I = Arus (A)

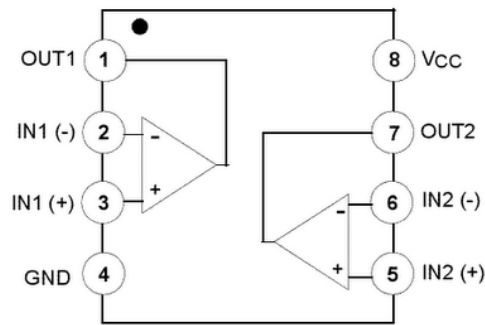
2.3.3 LM 358

LM 358 merupakan sebuah Op-Amp *single supply* 5V. Seperti juga Op-Amp yang lain maka LM 358 memiliki sifat :

- Impedansi input besar
- Impedansi output kecil

LM 358 memiliki 8 pin dalam bentuk DIP (*Dual In Line Package*). Fungsi dari pin-pin LM 358 adalah sebagai berikut :

- VCC : power supply 5V
- GND : ground
- Input + : merupakan pin input non inverting dari OP-Amp
- Input - : merupakan pin input inverting dari Op-Amp
- Output : merupakan output dari Op-Amp



Gambar 2.7 Konfigurasi pin op-amp LM 358

2.4 Mikrokontroler

Mikrokontroler adalah otak dari suatu sistem elektronika seperti halnya mikroprosesor sebagai otak komputer. Namun mikrokontroler memiliki nilai tambah karena didalamnya sudah terdapat memori dan sistem *input/output* dalam suatu kemasan IC. Mikrokontroler AVR (*Alf and Vegard's RISC processor*) standar memiliki arsitektur 8-bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*. Berbeda dengan instruksi MCS-51 yang membutuhkan 12 siklus *clock* karena memiliki arsitektur CISC (seperti komputer).

Teknologi yang digunakan pada mikrokontroler AVR berbeda dengan mikrokontroler seri MCS-51. AVR berteknologi RISC (*Reduced Instruction Set Computer*), sedangkan seri MCS-51 berteknologi CISC (*Complex Instruction Set Computer*). Mikrokontroler AVR dapat dikelompokkan menjadi empat kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, Keluarga ATmega, dan AT89RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, kelengkapan periferal dan fungsi-fungsi tambahan yang dimiliki.

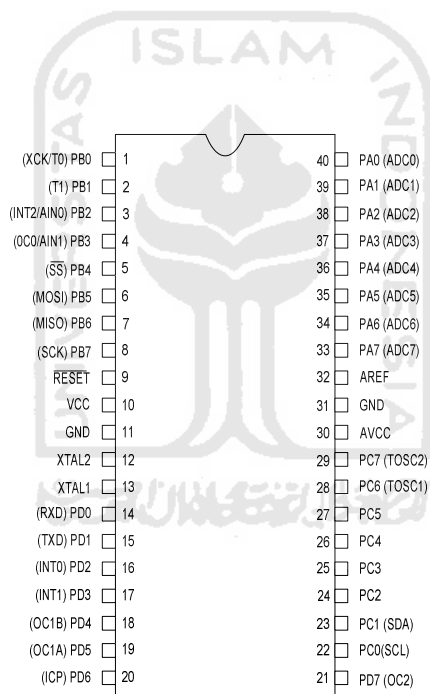
2.4.1 Mikrokontroler AVR ATmega8535

ATmega8535 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC. Kebanyakan instruksi dikerjakan pada satu siklus clock, ATmega8535 mempunyai *throughput* mendekati 1 MIPS per MHz, hal ini membuat ATmega8535 dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah. Mikrokontroler ATmega8535 memiliki beberapa fitur atau spesifikasi yang menjadikannya sebuah solusi pengendali yang efektif untuk berbagai keperluan. Fitur-fitur tersebut antara lain:

1. Saluran I/O sebanyak 32 buah, yang terdiri atas Port A, B, C dan D.
2. ADC (*Analog to Digital Converter*) dengan resolusi 10-bit sebanyak 8 saluran melalui Port A.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 *register*.
5. *Watchdog Timer* dengan osilator internal.
6. SRAM sebesar 512 byte.
7. Memori *Flash* sebesar 8 kb dengan kemampuan *Read While Write*.
8. Unit Interupsi Internal dan Eksternal.
9. Port antarmuka SPI untuk mendownload program ke *flash*.
10. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial.

2.4.2 Konfigurasi Pin ATmega8535

Mikrokontroler ATmega8535 mempunyai jumlah pin sebanyak 40 buah, dimana 32 pin digunakan untuk keperluan port I/O yang dapat menjadi pin *input/output* sesuai konfigurasi. Pada 32 pin tersebut terbagi atas 4 bagian (port), yang masing-masingnya terdiri atas 8 pin. Pin-pin lainnya digunakan untuk keperluan rangkaian osilator, *supply* tegangan, *reset*, serta tegangan referensi untuk ADC. Untuk lebih jelasnya, konfigurasi pin ATmega8535 dapat dilihat pada gambar 2.8.



Gambar 2.8 Konfigurasi Pin ATmega8535

Berikut ini adalah susunan pin-pin dari ATmega8535 :

- VCC merupakan pin yang berfungsi sebagai pin masukan catu daya
- GND merupakan pin *ground*

- Port A (PA0..PA7) merupakan pin I/O 8-bit dua arah dengan resistor *pull-up* internal dan pin masukan ADC (ADC0 s.d. ADC7).
- Port B (PB0..PB7) merupakan pin I/O dua arah resistor *pull-up* internal dan pin fungsi khusus, yaitu *Timer/Counter*, Komparator Analog, dan SPI
- Port C (PC0..PC7) merupakan pin I/O 8-bit dua arah resistor *pull-up* internal dan dua pin yaitu PC6 dan PC7 berfungsi sebagai *oscillator* luar untuk *Timer/Counter2*.
- Port D (PD0..PD7) merupakan pin I/O 8-bit dua arah resistor *pull-up* internal dan pin fungsi khusus, yaitu Komparator Analog, Interupsi Eksternal dan komunikasi serial USART.
- Reset merupakan pin yang digunakan untuk mereset mikrokontroler
- XTAL1 dan XTAL2 merupakan pin masukan clock eksternal (osilator menggunakan kristal, biasanya dengan frekuensi 11,0592 MHz).
- AVCC merupakan catu daya untuk port A dan ADC.
- AREF merupakan tegangan referensi masukan analog untuk ADC.
- AGND merupakan *ground* analog ADC.

2.4.3 Komunikasi Serial Dengan UART

AVR ATmega8535 memiliki 4 buah *register* I/O yang berkaitan dengan komunikasi UART, yaitu UART I/O Data *Register* (UDR), UART *Baud Rate Register* (UBRR), UART Status Register (USR) dan UART *Control Register* (UCR).

a. Register Data I/O UART (UDR)

Proses pengiriman data secara serial dapat dimulai setelah UDR diberi karakter data. Pada sisi penerima, UART memiliki *buffer* sehingga UDR dapat dibaca ketika sebuah data baru sedang digeser masuk.

b. Register Baud Rate UART (UBRR)

UBRR digunakan untuk menentukan *clock* yang dibangkitkan oleh *baud rate generator*. Nilai *baud rate* ditentukan dengan mengisi *register* UBRR sesuai persamaan 2.8

$$UBRR = \frac{f_{clock}}{16 * baud} - 1 \quad (2.5)$$

Misal diinginkan *baud rate* sebesar 9600 bps. Dengan Kristal 11.059200 Hz maka *register* UBRR akan bernilai 71.

c. Register Status UART (USR)

Register USR menyimpan berbagai *flag* status seperti interupsi, *overflow*, dan *framing error*. Susunan bit *register* USR ditunjukkan dalam Gambar 2.9.

bit	7	6	5	4	3	2	1	0
	RXC	TXC	UDRE	FE	OR	-	-	-

Gambar 2.9 Register status UART

Penjelasan bit-bit *register* USR adalah sebagai berikut :

1. RXC (*Receive Complete*) ; bernilai 1 otomatis setelah UART menerima sebuah karakter secara lengkap.
2. TXC (*Transmit Complete*) ; bernilai 1 jika sebuah karakter telah selesai digeser keluar dari *register* geser kirim.
3. UDRE (*UART Data Register Empty*) ; bernilai 1 jika UDR kosong.
4. FE (*Framing Error*) ; bernilai 1 jika *Stop Bit* tidak diterima dengan benar, yaitu jika *Stop Bit* terbaca 0.
5. OR (*Overrun*) ; bernilai 1 jika ada karakter yang dipindahkan dari *register* geser terima ke UDR sebelum karakter yang diterima sebelumnya dibaca.

d. Register Kontrol UART (UCR)

Register UCR mengendalikan berbagai fungsi penerima dan pengirim, serta interupsinya. Susunan bit *register* UCR ditunjukkan dalam Gambar 2.10.

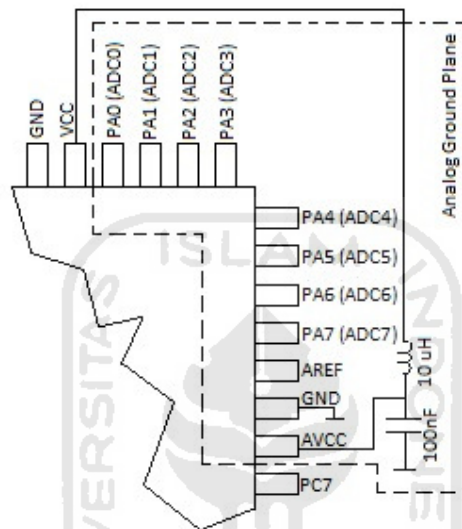
bit	7	6	5	4	3	2	1	0
	RXC	TXC	UDRE	FE	OR	-	-	-

Gambar 2.10 Register kendali UART

2.4.4 Analog To Digital Converter (ADC)

ADC pada ATMega8535 merupakan ADC 10-bit tipe *Successive Approximation*, yang terhubung ke sebuah multiplexer analog yang akan memilih satu dari 8 kanal. Untuk menjaga validitas data, terdapat untai *Sample and Hold*.

Tegangan suplai ADC terpisah dari tegangan suplai mikrokontroler, tetapi selisihnya tidak boleh lebih dari 0,3 V. Untuk mengatasinya, digunakan untai filter LC seperti ditunjukkan pada Gambar 2.11.



Gambar 2.11 Koneksi dengan filter LC pada suplai ADC

Terdapat 8 kanal ADC masing-masing selebar 10 bit. ADC dapat digunakan dengan memberikan masukan tegangan pada port ADC, yaitu port A.0 sampai dengan port A.7.

Ada dua mode ADC yang dapat digunakan, yaitu *single conversion* dan *free running*. Pada mode *single conversion*, pengguna harus mengaktifkan setiap kali ADC akan digunakan. Sedangkan pada mode *free running*, pengguna cukup sekali mengaktifkan, sehingga ADC akan terus mengkonversi tanpa henti.

Dalam kemasan TQFP (*Thin Quad Flat Package*) terdapat fasilitas tambahan, yaitu kanal diferensial dan kanal diferensial dengan penguatan, yang

memungkinkan dua kanal ADC digunakan sekaligus. Kemasan PDIP tidak menyediakan fasilitas ini.

Terdapat beberapa register I/O yang terlibat dalam proses konversi ADC, antara lain : ADMUX (ADC Multiplexer Selection Register). Register ADMUX berisi bit-bit yang mengatur pilihan kanal (MUX4:0), bit pengatur penyajian data (ADLAR), dan bit-bit pemilih tegangan referensi (REFS1:0). Gambar 2.12 menunjukkan isi *register* ADMUX.

bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.12 Register ADMUX

a. ADCSRA (ADC Control and Status Register A)

Register ADCSRA berisi bit-bit yang mengatur nilai *prescaler* (ADPS2:0), bit aktivasi interupsi selesainya ADC (ADIE), bit penanda (*flag*) selesainya konversi (ADIF), bit aktivasi penyulut otomatis ADC (ADATE), bit pemulai konversi (ADSC), dan bit aktivasi ADC (ADEN). Gambar 2.13 menunjukkan isi *register* ADCSRA.

bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.13 Register ADCSRA

b. SFIOR (*Special Function Input/Output Register*)

Register SFIOR berisi beberapa bit yang terlibat dalam ADC, dan beberapa bit yang terlibat dalam fitur lain. Bit-bit yang terlibat dalam ADC adalah bit-bit pemilih sumber penyulut otomatis (ADTS2:0). Gambar 2.14 menunjukkan isi register SFIOR.

bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.14 Register SFIOR

c. ADCH (*ADC High-byte data*)

Register ADCH berisi data hasil konversi. Bentuk penyajiannya ditentukan oleh bit ADLAR. Jika dibiarkan bernilai 0 (*clear*), register ini berisi bit MSB dan MSB-1 dari data hasil konversi.

d. ADCL (*ADC Low-byte data*)

Register ADCL berisi data hasil konversi. Bentuk penyajiannya ditentukan oleh bit ADCL. Jika dibiarkan bernilai 0 (*clear*), register ini berisi bit LSB+7 hingga LSB dari data konversi.

Dalam mengakses ADC, terdapat proses pengaturan *register-register I/O* yang terlibat dalam ADC. Proses pengaturan tersebut meliputi :

1. Menentukan sumber tegangan referensi

Referensi pada ADC merupakan batas rentang representasi nilai digital hasil konversi. Hasil konversi pada mode *single ended conversion* dirumuskan sebagai berikut :

$$ADC = \frac{VIN \cdot 1024}{VREF} \quad (2.6)$$

dengan VIN = tegangan masukan analog pada kanal ADC yang aktif

VREF = tegangan referensi yang dipilih

Ada tiga sumber referensi yang dapat digunakan, dan dipilih dengan mengatur bit REF1:0 (*Reference Selection Bit1:0*) pada *register* ADMUX.

Tabel 2.1 menunjukkan pengaturan bit-bit tersebut dan pilihan referensi yang tersedia

Tabel 2.1 Pemilihan Tegangan Referensi pada ADC

REF1	REF0	Tegangan Referensi yang Dipilih
0	0	Tegangan pada pin AREF
0	1	Tegangan pada pin AVCC, dengan kapasitor eksternal pada pin AREF
1	0	Dilarang
1	1	Tegangan referensi Internal 2,56 V dengan kapasitor eksternal pada pin AREF

2. Menentukan bentuk penyajian data ADC

Bentuk penyajian data pada *register* ADCL dan ADCH ditentukan oleh bit ADLAR (ADC *Left Adjusted Result*) pada *register* ADMUX. Data hasil konversi memiliki panjang 10 bit, sedangkan setiap *register* data dapat menampung 8 bit. Artinya jika sebuah *register* diisi penuh, maka *register* lain tidak penuh berisi. Jika bit ADLAR dibiarkan tetap '0' (*clear*), 8 bit rendah disimpan di ADCL, dan ADCH sisanya. Dan sebaliknya.

3. Memilih kanal yang aktif (satu dari delapan)

Kanal yang aktif ditentukan oleh bit-bit MUX4:0 (*Analog Channel and Gain Selection Bits*) pada *register* ADMUX. Tabel 2.2 menunjukkan konfigurasi bit-bit tersebut.

Tabel 2.2 Konfigurasi bit-bit ADMUX4:0

MUX4	MUX3	MUX2	MUX1	MUX0	Kanal yang Aktif
0	0	0	0	0	ADC0
0	0	0	0	1	ADC1
0	0	0	1	0	ADC2
0	0	0	1	1	ADC3
0	0	1	0	0	ADC4
0	0	1	0	1	ADC5
0	0	1	1	0	ADC6
0	0	1	1	1	ADC7

4. Menentukan mode ADC

Mode *single conversion* atau *free running* ditentukan dengan menulisi bit ADATE (ADC *Auto Trigger Enable*) pada *register* ADCSRA. Jika dibiarkan '0' (*clear*), maka mode *single conversion* yang dijalankan.

Apabila diberi logika ‘1’ (*set*), ADC beraksi dalam mode *free running*. Sebagai catatan, *register* SFIOR dibiarkan apa adanya (*default setting*). Apabila ADC diinginkan disulut oleh sumber lain, maka bit-bit ADTS2:0 (*ADC Auto Trigger Source2:0*) di *register* SFIOR harus diberi nilai. Tabel 2.3 menunjukkan konfigurasi bit-bit pemilih sumber penyulut ADC.

Tabel 2.3 Konfigurasi bit-bit ADTS2:0

ADTS2	ADTS1	ADTS0	Sumber Penyulut
0	0	0	Mode <i>free running</i>
0	0	1	<i>Analog Comparator</i>
0	1	0	Sumber interupsi eksternal INT0
0	1	1	<i>Compare match</i> pada <i>Timer/Counter0</i>
1	0	0	Limpahan (<i>overflow</i>) <i>Timer/Counter0</i>
1	0	1	<i>Compare match B</i> pada <i>Timer/Counter1</i>
1	1	0	Limpahan (<i>overflow</i>) <i>Timer/Counter1</i>
1	1	1	Rekam kejadian (<i>capture event</i>) pada <i>Timer/Counter1</i>

5. Menentukan *prescaler* (*clock* ADC)

Prescaler merupakan faktor pembagi yang diterapkan pada *clock* mikrokontroler agar untai *successive approximation* ADC mampu menerima *clock* yang cukup sehingga data hasil konversi cukup *valid*. Untuk ketelitian ADC 10 bit, rentang frekuensi *clock* yang diperbolehkan adalah 50 kHz hingga 200 Hz. Frekuensi *clock* yang lebih tinggi dapat digunakan jika resolusi lebih rendah dari 10 bit. Nilai *prescaler* ditentukan oleh bit-bit ADPS2:0 (*ADC Prescaler Select Bits2:0*). Tabel 2.4 menunjukkan konfigurasi bit-bit tersebut.

Tabel 2.4 Konfigurasi bit-bit ADPS

ADPS2	ADPS1	ADPS0	Nilai <i>Prescaler</i>
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

6. Inisialisasi ADC

Untuk mengaktifkan ADC, bit ADEN (*ADC Enable*) harus diberi logika '1'. Dan untuk memulai ADC, logika '1' juga harus diberikan pada bit ADSC (*ADC Start Conversion*). Kedua bit ini terletak pada *register* ADCSRA. Waktu yang diperlukan untuk satu konversi adalah 25 siklus *clock* ADC pada konversi pertama, dan 13 siklus *clock* ADC untuk konversi berikutnya.

Idealnya, ADC dapat melakukan konversi secara linier antara GND dan VREF dalam 2^n langkah. Sehingga faktor pengali hasil konversi terendah adalah 0, dan tertinggi $2^n - 1$. Indeks n menunjukkan resolusi ADC (8, 9, atau 10 bit). Tetapi terdapat beberapa hal yang menyebabkan ADC tidak mencapai kondisi ideal, yaitu galat yang terjadi saat konversi, antara lain :

- *Offset error*
- *Gain error*
- *Integral non-linearity*
- *Differential non-linearity*

- *Quantization error*

Dengan adanya galat-galat tersebut, pada akhirnya fasilitas ADC memiliki akurasi absolut sebesar ± 2 LSB.

2.5 Relay

Dalam dunia elektronika, *relay* dikenal sebagai komponen yang dapat mengimplementasikan logika *switching*. Sebelum tahun 70an, *relay* merupakan “otak” dari rangkaian pengendali. *Relay* yang paling sederhana ialah *relay* elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. Secara sederhana *relay* elektromekanis ini didefinisikan sebagai berikut :

- Alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar.
- Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

Di bawah ini contoh *relay* yang beredar di pasaran :



Gambar 2.15 *Relay* yang tersedia di pasaran

Secara umum, *relay* digunakan untuk memenuhi fungsi – fungsi berikut :

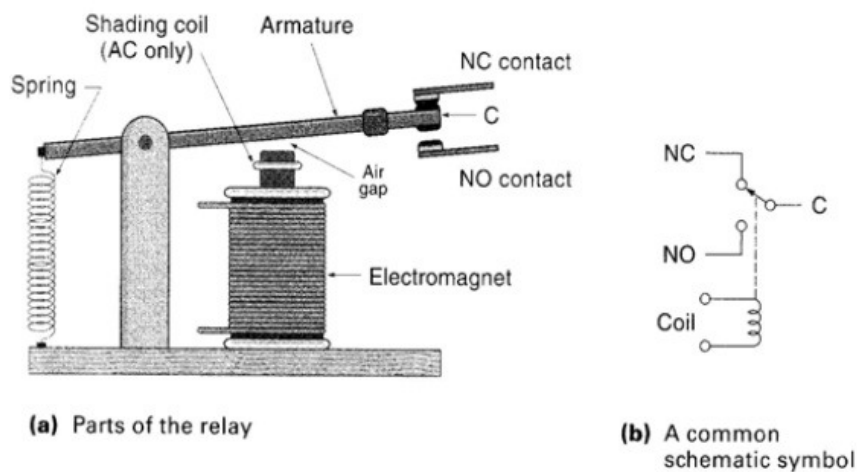
- *Remote control* : dapat menyalakan atau mematikan alat dari jarak jauh

- Penguatan daya : menguatkan arus atau tegangan, Contoh : *starting relay* pada mesin mobil
- Pengatur logika kontrol suatu sistem

2.5.1 Prinsip Kerja dan Simbol

Relay terdiri dari *coil* dan *contact*. Gambar 2.16 menunjukkan skema *relay* elektromagnetik dengan *coil* adalah gulungan kawat yang mendapat arus listrik, sedang *contact* adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*. *Contact* ada 2 jenis : *Normally Open* (kondisi awal sebelum diaktifkan *open*), dan *Normally Closed* (kondisi awal sebelum diaktifkan *close*).

Secara sederhana berikut ini prinsip kerja dari *relay* : ketika *Coil* mendapat energi listrik (*energized*), akan timbul gaya elektromagnet yang akan menarik *armature* yang berpegas, dan *contact* akan menutup.

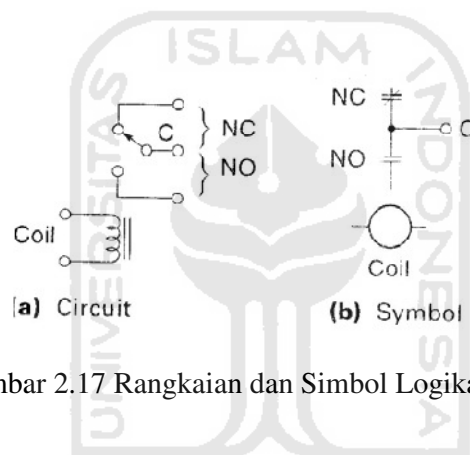


Gambar 2.16 Skema *Relay* Elektromagnetik

Selain berfungsi sebagai komponen elektronik, *relay* juga mempunyai fungsi sebagai pengendali sistem. Sehingga *relay* mempunyai 2 macam simbol yang digunakan pada :

- Rangkaian listrik (*hardware*)
- Program (*software*)

Simbol selalu mewakili kondisi *relay* tidak *dienergized*. Berikut ini simbol yang digunakan :



Gambar 2.17 Rangkaian dan Simbol Logika Relay

2.6 Solenoid Valve

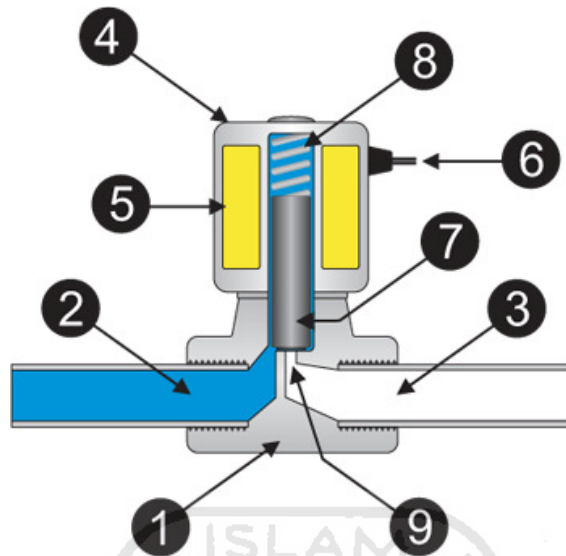
Solenoid valve adalah katup yang digerakan oleh energi listrik, mempunyai kumparan sebagai penggeraknya yang berfungsi untuk menggerakkan piston yang dapat digerakan oleh arus AC maupun DC, *solenoid valve* atau katup (*valve*) solenoida mempunyai lubang keluaran, lubang masukan dan lubang *exhaust*, lubang masukan, berfungsi sebagai terminal / tempat cairan masuk atau *supply*, lalu lubang keluaran, berfungsi sebagai terminal atau tempat cairan keluar yang dihubungkan ke beban, sedangkan lubang *exhaust*, berfungsi sebagai saluran

untuk mengeluarkan cairan yang terjebak saat piston bergerak atau pindah posisi ketika *solenoid valve* bekerja.

Prinsip kerja dari *solenoid valve*/katup (*valve*) solenoida yaitu katup listrik yang mempunyai koil sebagai penggerakanya dimana ketika koil mendapat *supply* tegangan maka koil tersebut akan berubah menjadi medan magnet sehingga menggerakkan piston pada bagian dalamnya ketika piston berpindah posisi maka pada lubang keluaran dari *solenoid valve* akan keluar cairan yang berasal dari *supply*, solenoid valve yang digunakan mempunyai tegangan kerja 24VDC, maksimum tekanan adalah 1 MPa atau 1000 kPa.



Gambar 2.18 *Solenoid Valve*



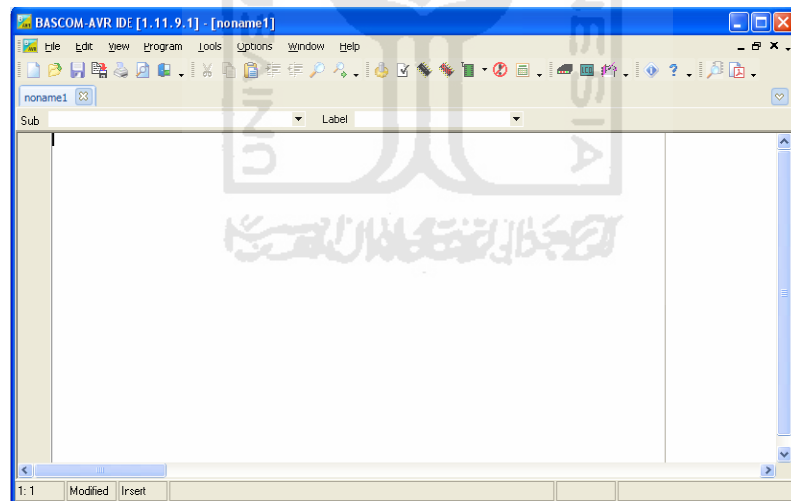
Gambar 2.19 Struktur Fungsi *Solenoid Valve*

Keterangan Gambar :

1. *Valve Body*
2. Terminal masukan (*Inlet Port*)
3. Terminal keluaran (*Outlet Port*)
4. Koil / koil solenoid
5. Kumparan gulungan
6. Kabel suplai tegangan
7. *Plunger*
8. *Spring*
9. Lubang / *exhaust*

2.7 BASCOM – AVR

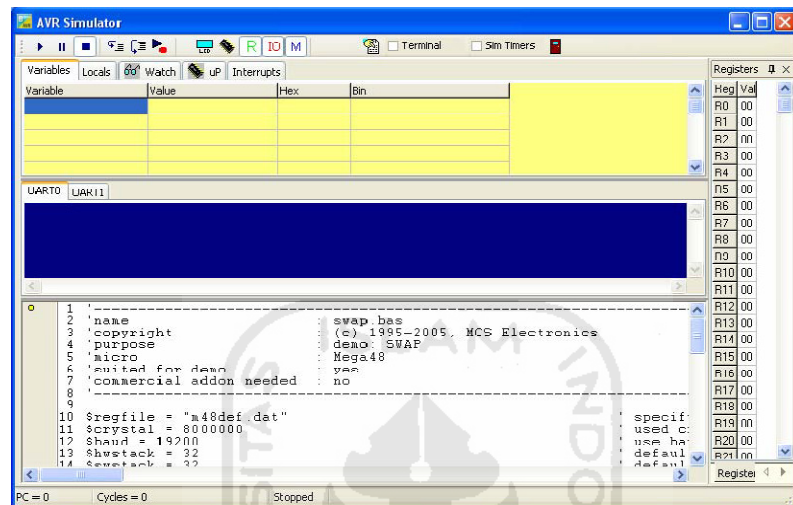
BASCOM-AVR adalah program *basic compiler* berbasis *windows* untuk mikrokontroler keluarga AVR merupakan pemrograman dengan bahasa tingkat tinggi ” *BASIC* ” yang dikembangkan dan dikeluarkan oleh MCS elektronika sehingga dapat dengan mudah dimengerti atau diterjemahkan. Dalam program BASCOM-AVR terdapat beberapa kemudahan, untuk membuat program software ATMEGA 8535, seperti program simulasi yang sangat berguna untuk melihat, simulasi hasil program yang telah dibuat, sebelum program tersebut diunduh ke IC atau ke mikrokontroler. Ketika program BASCOM-AVR dijalankan dengan mengklik icon BASCOM-AVR, maka jendela berikut akan tampil :



Gambar 2.20 Tampilan Jendela Program BASCOM – AVR

BASCOM-AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat

dengan pergerakan LED yang ada pada layar simulasi dan dapat juga langsung dilihat pada LCD, jika dibuat aplikasi yang berhubungan dengan LCD.



Gambar 2.21 Tampilan Simulasi BASCOM – AVR

Intruksi yang dapat digunakan pada editor Bascom-AVR relatif cukup banyak dan tergantung dari tipe dan jenis AVR yang digunakan.

2.8 Visual Basic 6.0

Kata “*Visual*” merujuk kepada metode yang digunakan untuk membuat antar muka yang bersifat grafis *Graphical User Interface* (GUI). Kata “*Basic*” merujuk kepada bahasa *BASIC* (*Beginners All-Purpose Symbolic Instruction Code*), sebuah bahasa yang digunakan oleh banyak *programmer* dibandingkan dengan bahasa lainnya dalam sejarah komputer. *Visual Basic* telah berubah dari bahasa dari bahasa asli *BASIC* dan sekarang memiliki ratusan pernyataan (*statements*), fungsi (*functions*), dan kata kunci (*keywords*), dan kebanyakan di antaranya terkait dengan antar muka grafis di *Windows*.

Visual Basic (atau sering disingkat VB) adalah perangkat lunak untuk menyusun program aplikasi yang bekerja dalam lingkungan sistem operasi *Windows*. Dengan *Visual Basic*, dapat dimanfaatkan kemampuan *Windows* secara optimal. Dengan kecanggihan yang ditawarkan oleh *Visual Basic* dapat dirasakan begitu mudahnya menyusun program aplikasi dengan tampilan grafis yang menawan dalam waktu yang relatif singkat.

Bahasa pemrograman *Visual Basic* tidaklah hanya identik dengan *Visual Basic* saja. Sistem pemrograman *Visual Basic* dalam bentuk edisi aplikasi, telah dimasukkan ke dalam *Microsoft Excel*, *Microsoft Access*, dan banyak aplikasi *Windows* lainnya juga menggunakan bahasa yang sama. *Visual Basic Scripting Edition* (VBScript) adalah sebuah bahasa skrip yang digunakan secara lebih umum dan merupakan bagian dari bahasa *Visual Basic*.

Fitur *Data Access* membolehkan Anda untuk membuat *database*, aplikasi *front-end*, dan komponen di sisi server (*scalable server-side components*) untuk hampir semua *format database* yang terkenal, termasuk *Microsoft SQL Server* dan *database* level perusahaan lainnya. Teknologi ActiveX™ menggunakan fungsionalitas yang disediakan oleh aplikasi-aplikasi lainnya, seperti pengolah kata *Microsoft Word*, *Microsoft Excel spreadsheet*, *Microsoft Access* dan aplikasi *Windows* lainnya. Dalam penggunaannya dapat mengotomatisasikan aplikasi-aplikasi dan objek-objek yang dibuat dengan menggunakan edisi *Professional* atau *Enterprise Visual Basic*.

Kemampuan Internet membuatnya mudah menyediakan pengaksesan ke dokumen-dokumen atau file-file dan aplikasi-aplikasi melalui Internet atau

intranet dari dalam aplikasi Anda, atau untuk membuat aplikasi Internet server. Aplikasi akhir yang telah dibuat nantinya akan berbentuk sebuah file .exe yang menggunakan *Visual Basic Virtual Machine* dan dapat distribusikan secara bebas.



BAB III

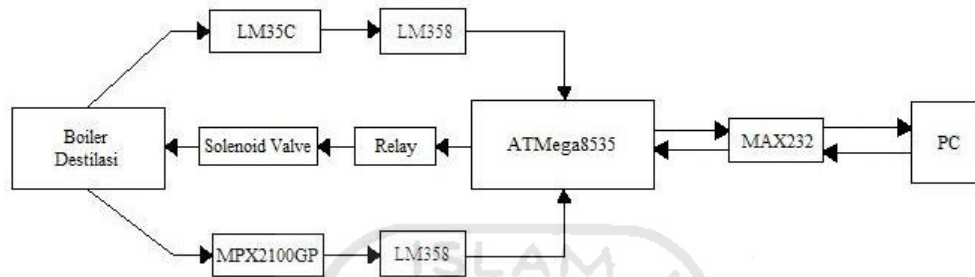
PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Perangkat keras (*hardware*) yang akan dibangun dalam Monitoring Suhu dan Tekanan Pada Destilasi Minyak Atsiri Berbasis ATmega8535 Berbasis PC diantaranya adalah sensor LM35C sebagai sensor suhu yang mengukur panas serta sensor MPX 2100 GP sebagai sensor tekanan yang mengukur tekanan pada boiler destilasi. Kemudian kedua sensor tersebut dikuatkan oleh IC op-amp LM358. Untuk pengolahan tegangan keluaran dari kedua sensor tersebut yang telah dikuatkan oleh Op-Amp digunakan ADC (*Analog to Digital Converter*) dari ATmega8535 yang mana ADC ATmega8535 mempunyai resolusi 10 bit. Dengan menggunakan *Basic-Compiler* (BASCOM) yang mana software berisi perintah-perintah untuk mengaktifkan perangkat yang terdapat pada ATmega8535. Setelah data dikonversi ke digital, data tersebut ditransfer ke komunikasi serial yang menggunakan IC MAX232 sebagai *interface* terhadap komputer (PC).

Data yang telah masuk ke komputer diolah lagi dengan menggunakan *software Microsoft Visual Basic 6.0* yang mana telah didesain untuk monitoring suhu dan tekanan. *Software* tersebut terdiri dari beberapa fitur, yakni : grafik *realtime*, alarm, perekam data masukan kedua sensor, *sampling time* serta set *point*. Setelah diatur *set point* pada nilai tekanan yang diinginkan, maka ketika tekanan menuju *set point* yang telah diatur, akan mengaktifkan *relay* yangmana

relay itu berfungsi untuk saklar guna mengaktifkan *solenoid valve* elektrik. Sebaliknya jika tekanan di bawah *set point* akan mematikan *relay* dan otomatis *solenoid valve* elektrik akan mati. Diagram blok sistemnya terdapat pada Gambar 3.1.



Gambar 3.1 Blok Diagram Sistem

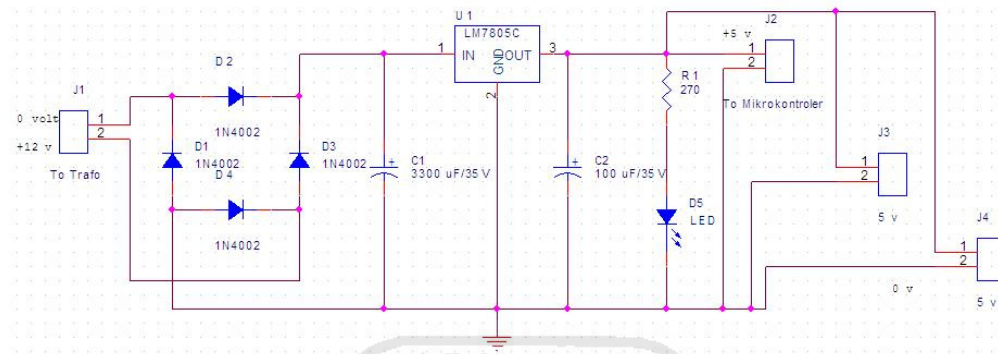
3.2 Perancangan Perangkat Keras

3.2.1 Rangkaian Catu Daya

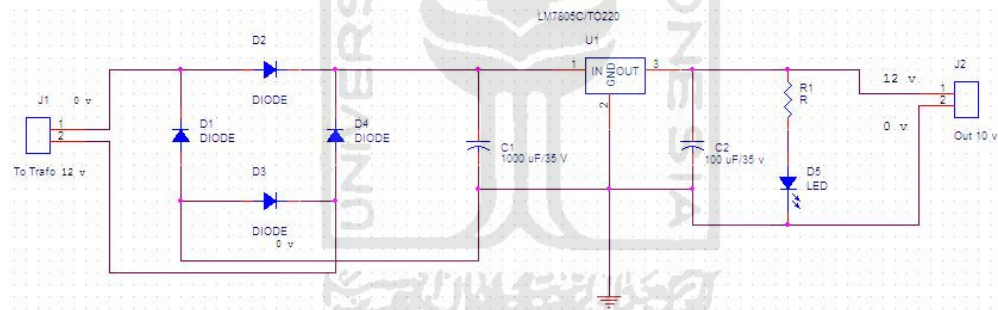
Sumber-sumber tegangan yang digunakan dalam perangkat ini adalah sebagai berikut :

1. Sumber tegangan +5V dengan menggunakan IC Regulator 7805, yang digunakan untuk suplai tegangan pada Sensor LM35C, ATmega8535, IC MAX232, IC op-amp LM358 (op-amp terhubung LM35C), dan Relay.
2. Sumber tegangan +10V dengan menggunakan IC Regulator 7810, yang digunakan untuk suplai tegangan pada sensor MPX2100GP dan IC op-amp LM358 (op-amp terhubung MPX2100GP).

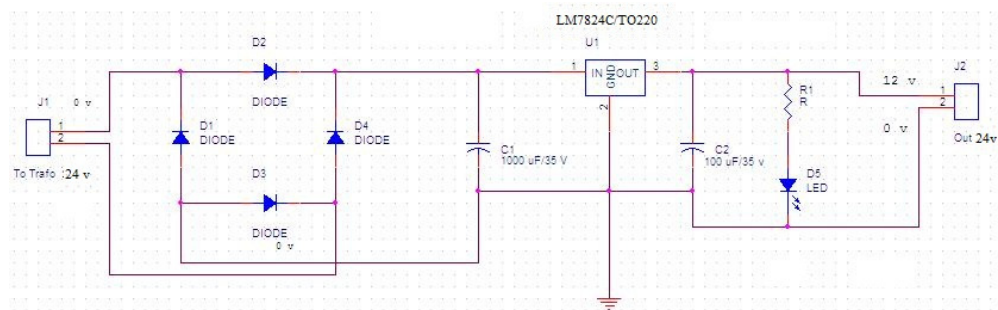
3. Sumber tegangan +24V dengan menggunakan IC Regulator 7824, yang digunakan untuk suplai tegangan pada *Solenoid Valve*.



Gambar 3.2 Rangkaian Catu Daya +5V



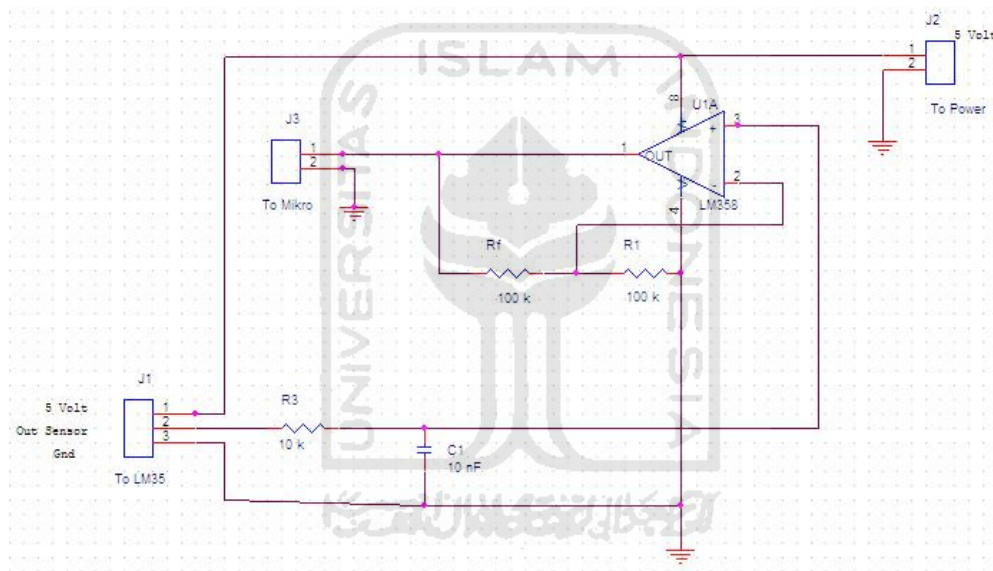
Gambar 3.3 Rangkaian Catu Daya +10V



Gambar 3.4 Rangkaian Catu Daya +24V

3.2.2 Rangkaian Penguat/Op-Amp

Di dalam perancangan sistem ini, digunakan 2 (dua) buah Op-Amp dengan tipe yang sama yaitu LM358. Op-Amp yang pertama dihubungkan dengan sensor suhu LM35C dengan penguatan 2 kali menggunakan penguatan *non-inverting*. Op-Amp yang kedua dihubungkan dengan sensor tekanan MPX2100GP dengan penguatan 101 kali menggunakan penguatan *non-inverting*.

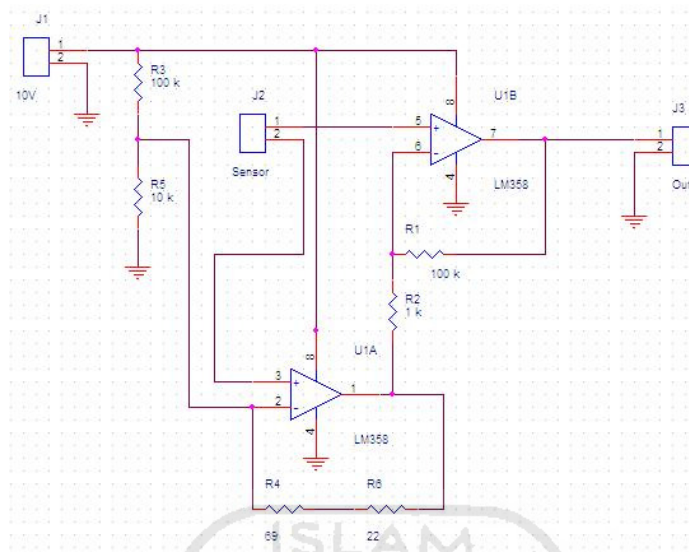


Gambar 3.5 Rangkaian Penguat untuk LM35C

Persamaan penguatan op-amp LM358 untuk sensor suhu LM35C adalah sebagai berikut :

$$A = \frac{R_f}{R_1} + 1 \quad (3.1)$$

Dengan $R_f = 100 \text{ k}$, $R_1 = 100 \text{ k}$, maka nilai A (penguatan) = 2 kali



Gambar 3.6 Rangkaian Penguat untuk MPX2100GP

Persamaan penguatan op-amp LM358 untuk sensor tekanan MPX2100GP adalah sebagai berikut :

$$A = \frac{R1}{R2} + 1 \quad (3.2)$$

Dengan $R1 = 100 \text{ k}$, $R2 = 1 \text{ k}$, maka nilai A (penguatan) = 101 kali. Didalam kalibrasi penguat ini, ada persamaan lagi untuk mencari keluaran pada pin 1, yaitu :

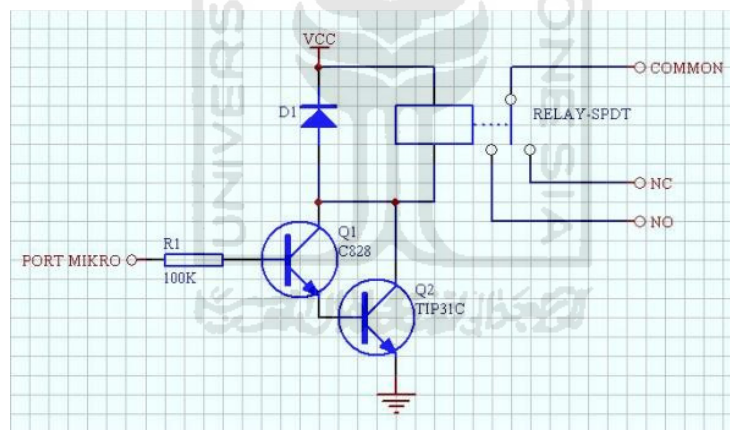
$$\frac{R1}{R2} = \frac{[R3 \cdot R5 / (R3 + R5)]}{R4 + R6} \quad (3.3)$$

Jadi keluaran dari pin 7 dan pin 1 haruslah sama dan seimbang, karena keluaran sensor MPX2100GP terdiri dari *output* (+) dan *output* (-) berbeda dengan sensor suhu LM35 yang terdiri dari *output* (+) dan *ground*.

3.2.3 Rangkaian Driver Relay

Transistor bipolar adalah komponen yang bekerja berdasarkan adanya atau tidaknya arus pemicuan pada kaki Basisnya. Pada aplikasi driver *relay*, transistor bekerja sebagai saklar yang pada saat tidak menerima arus pemicuan, maka transistor akan berada pada posisi *cut-off* dan tidak menghantarkan arus, $I_c=0$. Dan saat kaki basis menerima arus pemicuan, maka transistor akan berubah ke keadaan saturasi dan menghantarkan arus.

Gambar 3.7 adalah rangkaian praktis driver *relay* yang sangat handal untuk digunakan dalam proyek-proyek mikrokontroler.



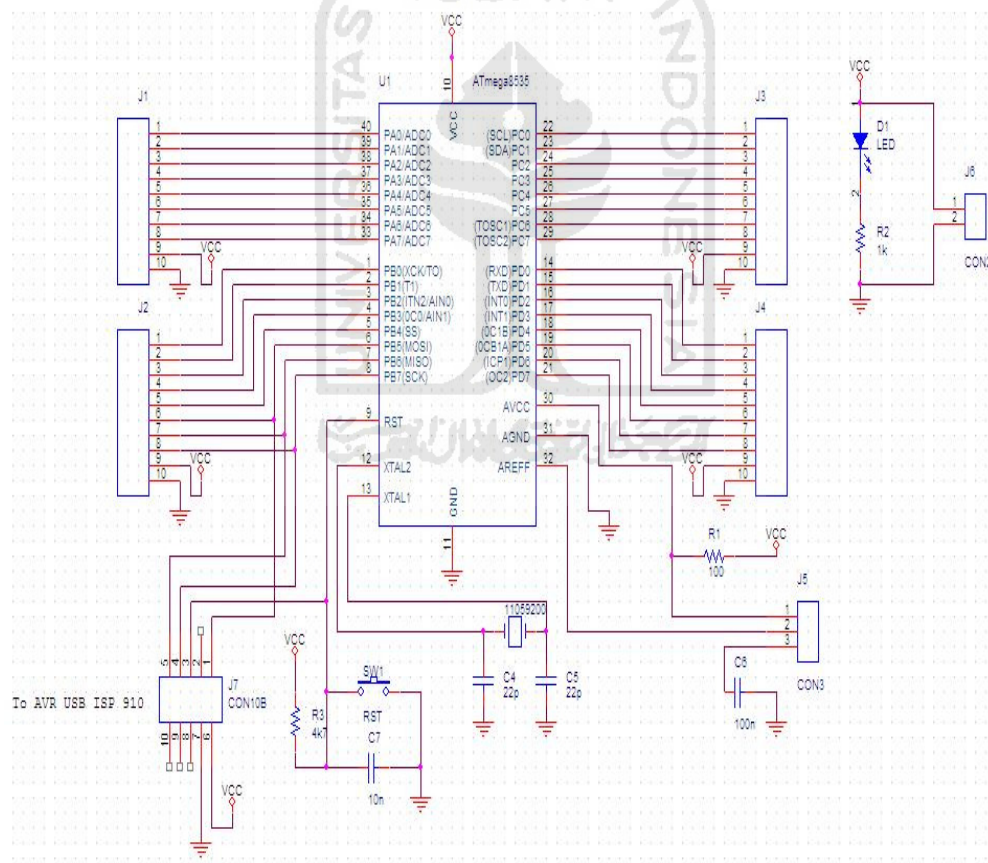
Gambar 3.7 Rangkaian Driver Relay

Komponen aktif rangkaian di atas adalah 2 buah transistor jenis NPN yang disusun secara Darlington. Transistor ini berfungsi sebagai saklar elektronik yang akan mengalirkan arus jika terdapat arus bias pada kaki basisnya, dan akan menyumbat arus jika tidak terdapat arus bias pada kaki basisnya. *Relay* yang dapat digunakan dengan rangkaian ini adalah *relay* dengan tegangan kerja koil

antara 5Vdc hingga 45Vdc. Pemasangan *diode* 1N4002 berfungsi mencegah arus transien yang ditimbulkan oleh kumparan *relay*.

3.2.4 Rangkaian Mikrokontroler ATmega8535

Untuk mengolah keluaran dari penguat yang berupa analog lalu diubah kedalam bentuk digital dengan menggunakan ADC (*Analog to Digital Converter*) dari mikrokontroler ATmega8535. Ada pun rangkaian sistem minimum ATmega8535 pada Gambar 3.8.



Gambar 3.8 Sistem Minimum ATmega8535

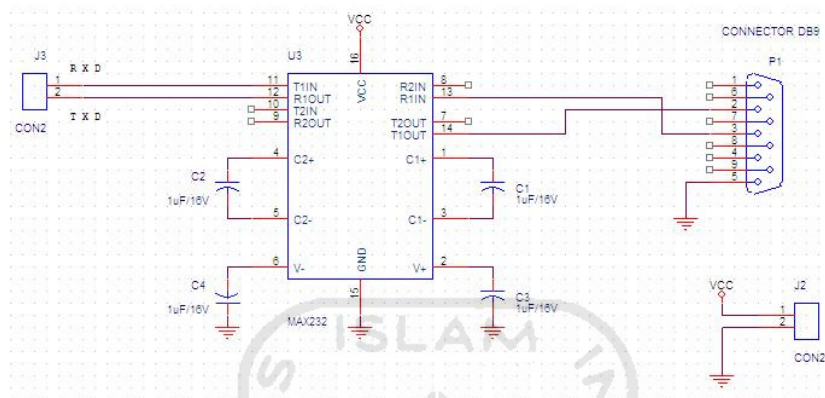
Resolusi ADC ATmega 8535 adalah 10 bit, sehingga mampu membaca tegangan input minimal 5 mV. Pada rangkaian osilator digunakan kristal 11,059200 MHz. Menurut *datasheet* crystal yang bisa digunakan untuk mikrokontroler ATMEGA8535 adalah 0 – 16 MHz dan dua kapasitor 22 pF. Waktu yang dibutuhkan untuk melakukan eksekusi sebuah instruksi dinamakan waktu siklus instruksi (*instruction cycles time*) yang nilainya tergantung pada kristal yang digunakan.

3.2.5 Rangkaian Komunikasi Serial dengan RS 232

Istilah RS232 sudah begitu populer sehingga sering dianggap RS232 mengatur semua hal tentang komunikasi data seri, meskipun sesungguhnya RS232 tidak mengatur sejauh itu. RS 232 (*Recommended Standard Number 232*) merupakan standart yang paling umum digunakan bagi antar muka untuk serial komputer. RS 232 mempunyai fungsi sebagai hubungan antar muka dan mengirim data dengan jarak yang relatif jauh. IC yang digunakan adalah IC dengan tipe MAX 232.

Max 232 berfungsi sebagai konverter dari level tegangan TTL ke level tegangan komputer. Hampir semua piranti digital menggunakan tingkatan logika TTL atau CMOS. MAX 232 sangat berperan dalam melakukan perubahan level tegangan timbal balik antar TTL RS-232 pada komunikasi *serial port*, IC memiliki 2 buah *line driver* dan 2 buah *line receiver*. IC ini juga dilengkapi dengan pengganda tegangan DC atau *charge pump* yang dapat menghasilkan tegangan -10 Volt sampai +10 Volt dari catu daya tunggal +5 Volt, sehingga

meskipun catu daya untuk IC MAX 232 hanya +5 Volt, IC ini mampu melayani tegangan RS 232 antara -10 Volt sampai +10 Volt.



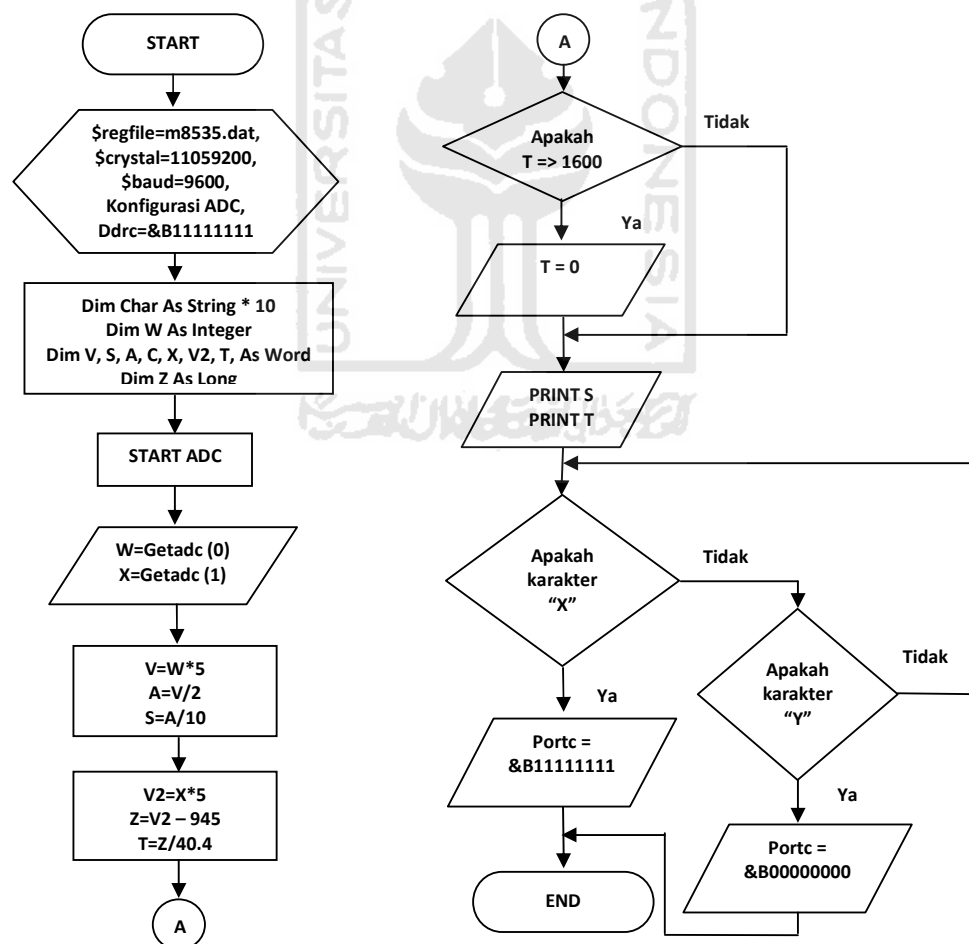
Gambar 3.9 Rangkaian RS 232

3.3 Perancangan Perangkat Lunak

Agar sistem bekerja dengan sebagaimana mestinya, maka diperlukan perangkat lunak yang mengatur kerja dari keseluruhan rangkaian. Pertama-tama yang dibuat adalah diagram alir (*flowchart*) dan kemudian dilakukan pembuatan program. Pembuatan program ditulis dengan bahasa Basic menggunakan tool BASCOM, dan program tersebut disimpan dalam memori flash mikrokontroler ATmega8535. Pada mikrokontroler ATmega8535 terdapat memori program sebesar 8 kbyte flash, EEPROM 512 byte dan memori data 512 byte RAM. Untuk pembacaan di komputer (PC) menggunakan software Microsoft Visual Basic 6.0, pembuatan program ditulis dengan bahasa Basic.

3.3.1 Basic Compiler AVR

Bascom AVR atau yang disebut *Basic Compiler* adalah suatu piranti lunak yang termasuk bahasa tingkat tinggi yang sangat mudah untuk dipelajari. Sebagai *compiler*, yaitu pengubah intruksi dari bahasa *basic* ke *file* yang berbentuk *hexa* dengan tujuan dimengerti oleh mesin atau mikrokontroler, sehingga mikrokontroler mampu menterjemahkan instruksi-instruksi yang telah dibuat dengan benar dan tepat. Piranti lunak inilah yang nantinya akan digunakan sebagai media perancangan dalam pembuatan perangkat lunak.



Gambar 3.10 Flowchart Program BASCOM

Pada gambar 3.10 merupakan diagram alir atau *Flowchart* Program BASCOM. Sebelum membuat program BASCOM, maka harus dibuat diagram alir agar mempermudah dalam pembuatan program.

Pada *flowchart* program BASCOM dimulai dengan awalan/start, pemberian nilai awalan berupa : jenis mikrokontroler, kristal, *baudrate*, konfigurasi ADC dan Portc sebagai output. Dilanjutkan proses pengenalan variabel, memulai ADC. Kemudian mempresentasikan input data dari pin ADC yakni pin 0 dan pin 1. Melakukan proses perhitungan untuk mencari nilai suhu dan tekanan. ADC pada mikrokontroler ATMEGA8535 mempunyai resolusi 10 bit, atau $2^{10} = 1024$. Perhitungan untuk mencari suhu :

$$\begin{aligned} \text{Nilai tegangan tiap 1 bit} &= \text{Tegangan Referensi} / 1024 \\ &= 5V / 1024 = 4,882 \text{ mV} \approx 5 \text{ mV} \end{aligned} \quad (3.4)$$

$$V = W * 5$$

$$A = V / 2$$

$$S = A / 10$$

Penjelasannya adalah nilai V yakni tegangan yang didapat dari W (nilai keluaran ADC pada pin 0) dikalikan 5 (nilai tegangan tiap 1 bit). Nilai A didapat dari nilai V dibagi 2 (penguatan 2 kali). Nilai S adalah nilai A dibagi 10 (tiap 1^0 C itu kenaikannya 10mV). Perhitungan untuk mencari tekanan :

$$\begin{aligned} \text{Nilai tegangan tiap 1 bit} &= \text{Tegangan Referensi} / 1024 \\ &= 5V / 1024 = 4,882 \text{ mV} \approx 5 \text{ mV} \end{aligned}$$

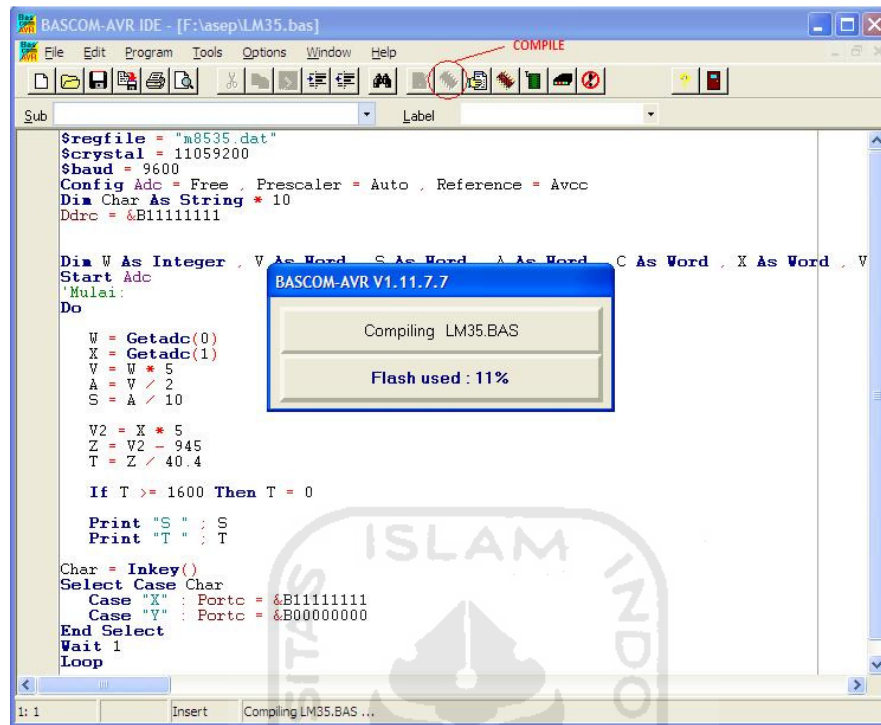
$$V2 = X * 5$$

$$Z = V2 - 945$$

$$T = Z / 40,4$$

Penjelasannya adalah nilai V2 yakni tegangan yang didapat dari X (nilai keluaran ADC pada pin 1) dikalikan 5 (nilai tegangan tiap 1 bit). Nilai Z didapat dari nilai V2 dikurangi 945 mV (kondisi tekanan 0 kPa = 945 mV). Nilai T didapat dari nilai Z dibagi 40,4 (setiap kenaikan 1 kPa = 0,4 mV karena dikuatkan 101 kali menjadi 40,4 mV). Selanjutnya keputusan dalam program, apakah $T \geq 1600$ jika “Ya” maka $T = 0$ sebaliknya jika “Tidak” maka dilanjutkan ke output data yaitu perintah print / cetak nilai “S”, “T”. Kemudian keputusan dalam program, apakah karakter “X” jika “Ya” maka `portc=&B11111111` (logika 1 untuk mengaktifkan *solenoid valve*), jika “Tidak” maka bertemu dengan keputusan program lain, apakah karakter “Y” jika “Ya” maka `portc=&B00000000` (logika 0 untuk mematikan *solenoid valve*), jika “Tidak” maka akan kembali kepada keputusan apakah karakter “X”. Akhir program diakhiri dengan *Finish*.

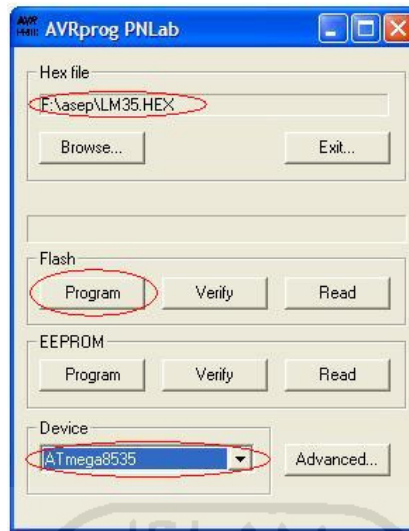
Untuk program BASCOM terdapat dilampiran. Hasil program yang telah dibuat di-*compile* menjadi file .hex. Gambar 3.11 adalah cara *compile* program yang telah dibuat. Setelah di-*compile*, program .hex ditranfer ke ATMega8535 dengan K-85 menggunakan AVRProg. K-85 harus terpasang dengan komputer dahulu kemudian jalankan AVRProg. Jika mengalami *error*, matikan dan tutup *software* kemudian pasang ulang K-85 dan jalankan AVRProg. File .hex tadi ketika ditekan program pada AVRProg maka langsung tersimpan pada memori ATMega8535.



Gambar 3.11 Compile Program BASCOM



Gambar 3.12 Skema K-85



Gambar 3.13 Gambar Software AVRProg

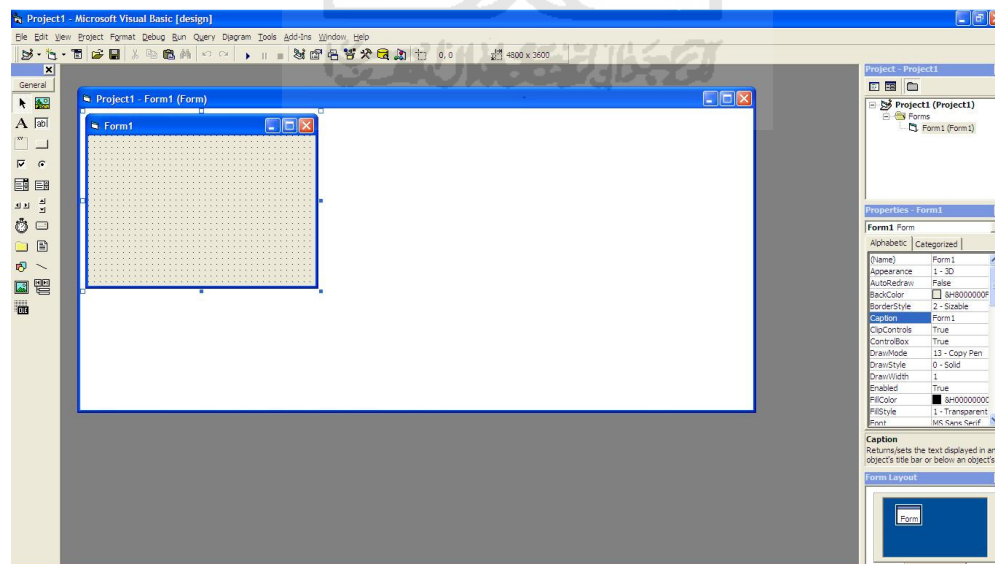
3.3.2 Pemrograman *Visual Basic*

Pembangkitan grafik atau sinyal dari input tegangan digital dari mikrokontroller memanfaatkan pemrograman *Visual Basic* 6.0. Pembuatan program aplikasi menggunakan *Visual Basic* dilakukan dengan membuat tampilan aplikasi pada *form*, kemudian diberi *script* program di dalam komponen-komponen yang diperlukan. *Form* disusun oleh komponen-komponen yang berada di *toolbox*, dan setiap komponen yang dipakai harus diatur propertinya lewat jendela *property*. Untuk memulai pembuatan program aplikasi di dalam *Visual Basic*, yang dilakukan adalah membuat *project* baru. *Project* adalah sekumpulan *form*, modul, fungsi, data dan laporan yang digunakan dalam suatu aplikasi. Membuat *project* baru dapat dilakukan dengan memilih menu [*File*] >> [*New Project*] atau dengan menekan ikon [*new project*] pada *Toolbar* yang terletak di pojok kiri atas.



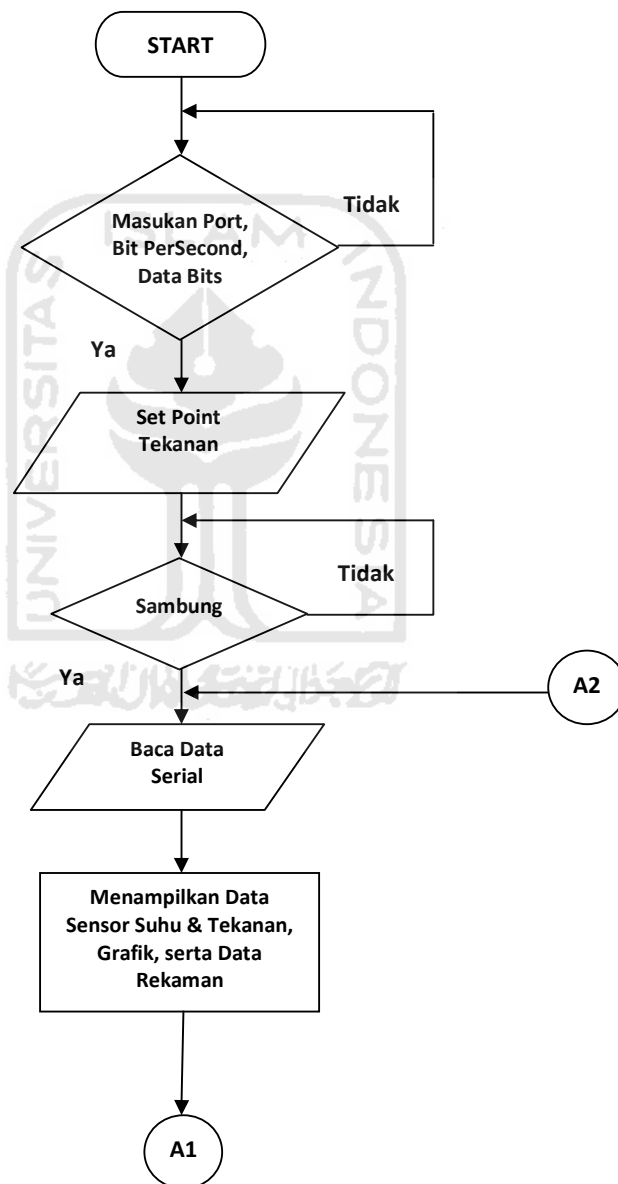
Gambar 3.14 Layar Pemilihan Jenis *Project*

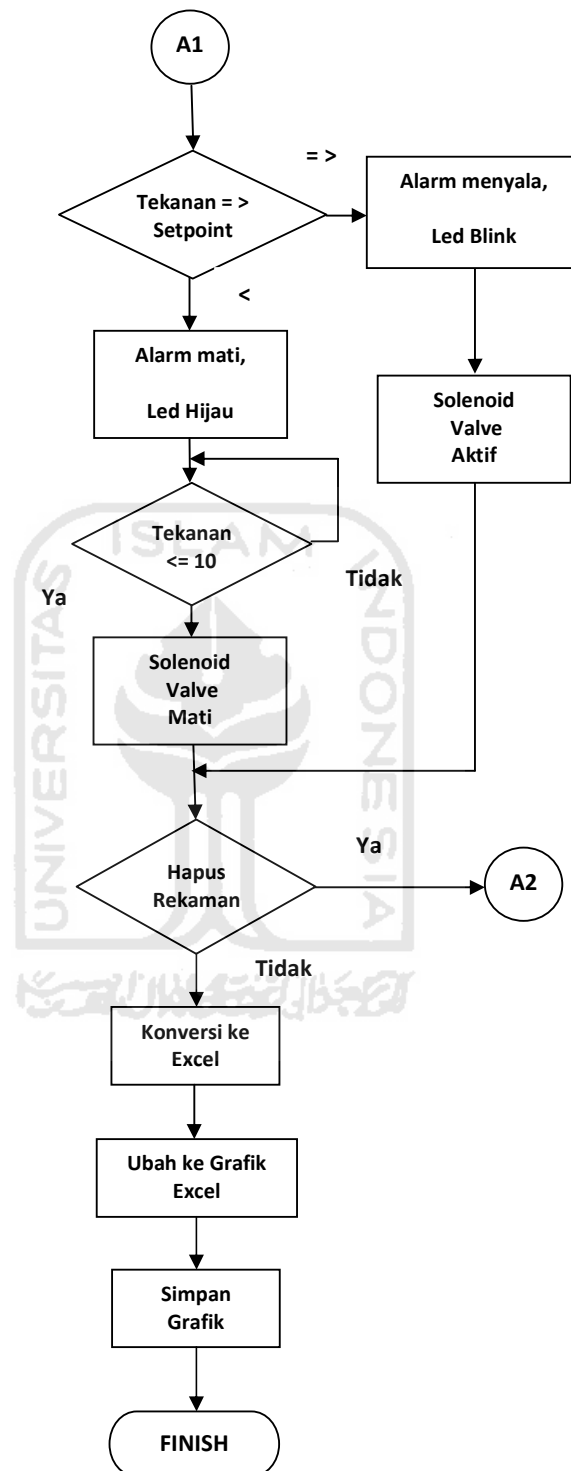
Selanjutnya pilih *Standard EXE* dan tekan [Ok]. Lalu muncul tampilan dari *Standard Exe* seperti pada gambar 3.15 Dengan demikian *project* sudah siap dibuat.



Gambar 3.15 Tampilan Standar *Project*

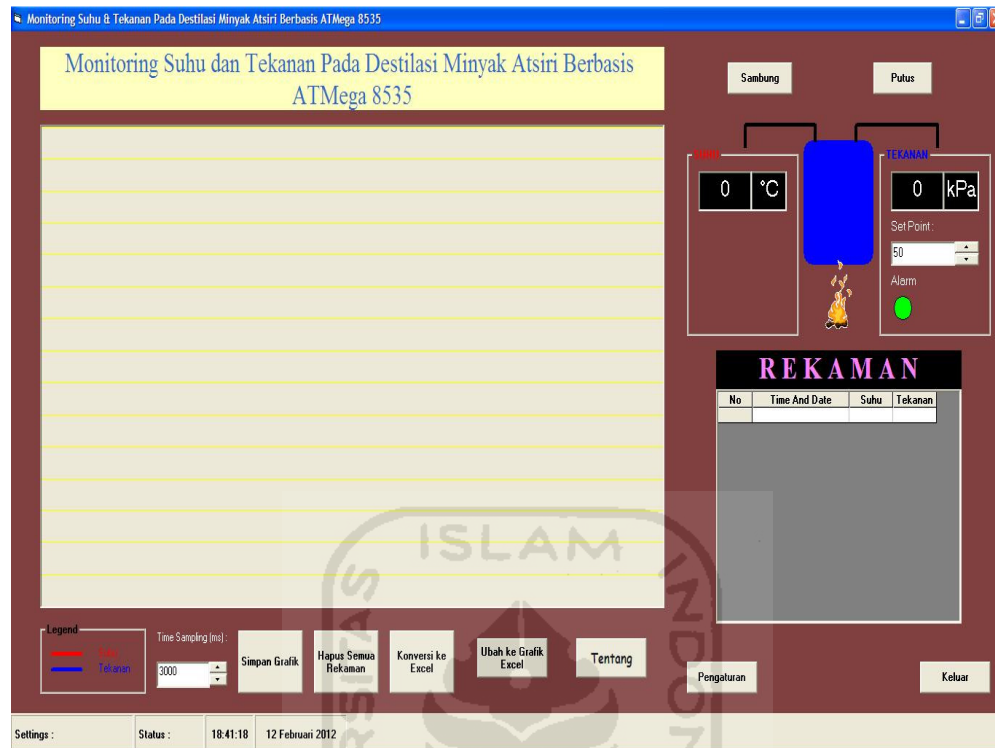
Sebelum membuat program Monitoring Suhu dan Tekanan dengan menggunakan *software Microsoft Visual Basic 6.0*, maka harus dibuat *flowchart* program untuk mempermudah mendesain program Monitoring Suhu dan Tekanan. Berikut *flowchart* programnya :





Gambar 3.16 Flowchart Program Visual Basic

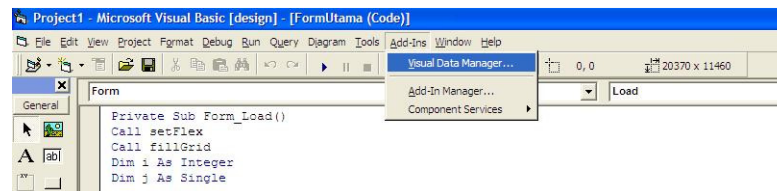
Pada *flowchart* program *visual basic* dimulai dengan awalan/*start*, selanjutnya keputusan program, apakah masukan port, bit per *second* dan data *bits* sesuai, jika “Ya” maka dilanjutkan ke step program berikutnya, jika “Tidak” kembali ke awalan. Step berikutnya, masukan input setpoint untuk suhu dan tekanan. Selanjutnya sambung atau mengkoneksi komputer dengan mikrokontroler, jika “Ya” maka diteruskan ke step selanjutnya, jika “Tidak” akan kembali ke perintah apakah siap koneksi kembali. Baca data serial merupakan input / masukan data yang akan diproses. Kemudian melakukan proses menampilkan data grafik, rekaman, sensor suhu dan tekanan yang telah di ubah ke digital oleh ADC. Proses selanjutnya adalah melakukan keputusan setpoint tekanan. Dimana setpoint tekanan jika tekanan “ $\geq$ ” setpoint maka alarm menyala dan Led berkedip/*blink* serta *solenoid valve* aktif/buka, sebaliknya jika tekanan “ $<$ ” setpoint maka alarm mati dan Led hijau, jika tekanan ≤ 10 maka *solenoid valve* mati/tutup. Step berikutnya adalah keputusan program hapus rekaman jika “Ya” kembali ke step baca data serial, jika “Tidak” maka dilanjutkan ke proses konversi ke Excel, kemudian proses ubah ke grafik excel dilanjutkan proses simpan grafik. Program pun ditutup dengan akhiran / *finish*. Untuk program monitoring suhu dan tekanan dengan *visual basic* terlampir.



Gambar 3.17 Program Monitoring Suhu dan Tekanan dengan Visual Basic

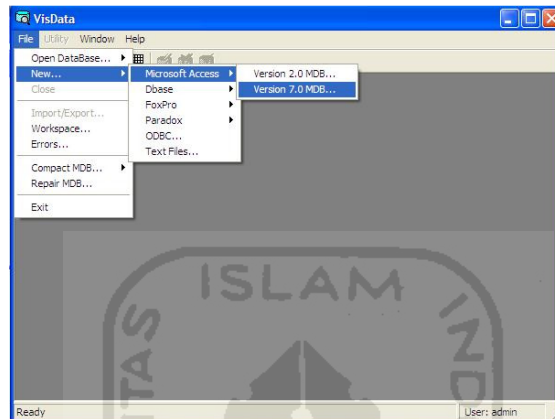
3.3.3 Merancang Database

Database adalah tempat menyimpan data suatu informasi. *Database* yang digunakan dalam monitoring ini adalah *database Access*. Setelah selesai membuat tampilan *form* pada *visual basic*, sekarang klik menu [**Add-Ins**], [**Visual Data Manager**].



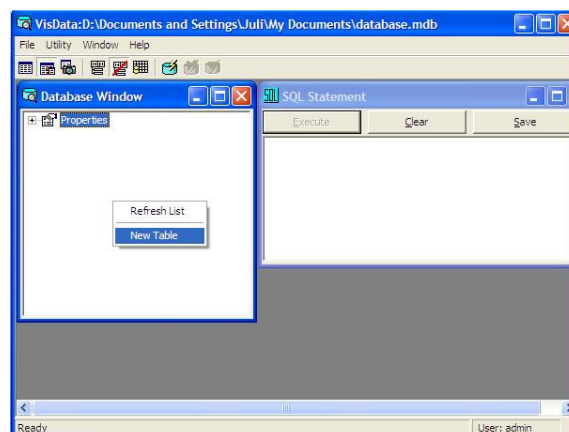
Gambar 3.18 Jendela VisData

Pada jendela VisData, klik [**File**][**New**][**Microsoft Access**][**Version 7.0 MDB**]. Buatlah *database* dengan nama “**database**” dan simpanlah ke dalam folder yang diinginkan.

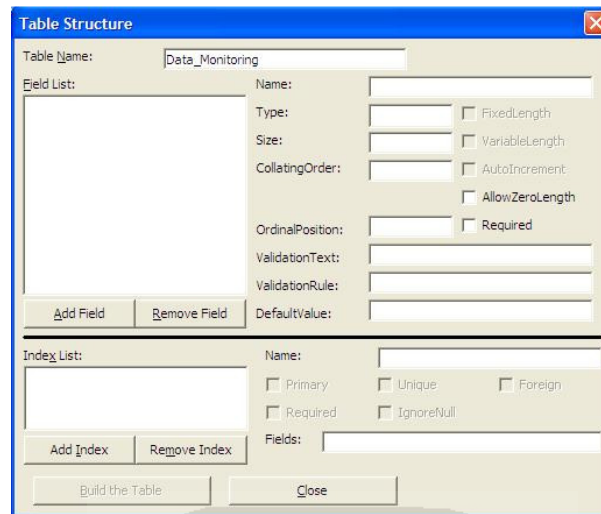


Gambar 3.19 Membuat *Database*

Selanjutnya pembuatan tabel, di dalam jendela VisData, klik kanan *properties* pada bagian *Database Windows*, kemudian pilih *New Table*. Buat nama tabel dengan “**Data_Monitoring**”, kemudian klik tombol [**Add Field**] untuk memasukkan *field*.

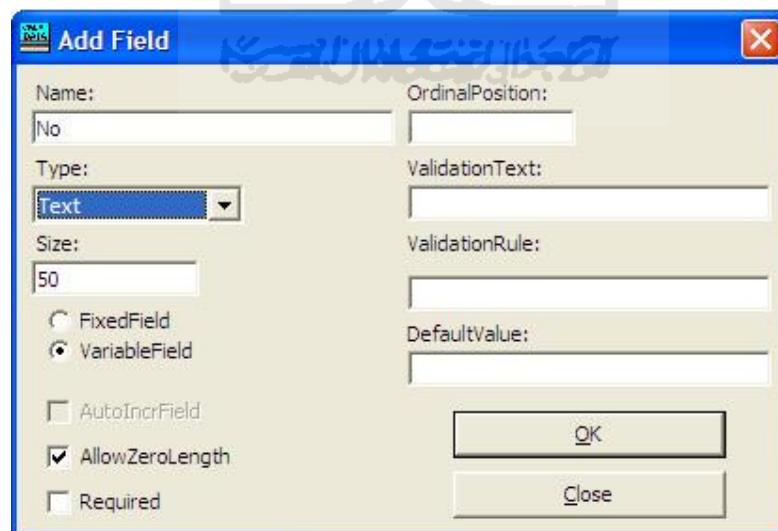


Gambar 3.20 Membuat Tabel



Gambar 3.21 Memberi Nama Tabel

Pada kotak dialog *Add Field*, ketik “No” pada bagian *Name*. Pilih tipe *Text* pada bagian *Type*, kemudian ubahlah ukuran *Size* menjadi 4. Klik [OK]. Lakukan hal yang sama untuk memasukkan *field TimeAndDate*, *Suhu*, *Tekanan*.



Gambar 3.22 Memasukkan Nama-nama *Field*

Langkah terakhir, klik tombol [**Build the Table**] di dalam kotak dialog *Table Structure*. Lalu keluarlah dari jendela VisData dengan klik menu [**File**][**Exit**].



Gambar 3.23 Mengaktifkan Tabel Data_Monitoring

Sebagai alat koneksi ke *database*, digunakan komponen ADODC, pada *Project Form*, klik kanan pada *Toolbox*, kemudian pilih *Components*, beri tanda cek pada pilihan *Microsoft ADO Data Control 6.0 (OLEDB)*, kemudian [**OK**]. Selanjutnya masukkan sebuah komponen Adodc ke dalam *Form*. Untuk mengatur *properties* dari Adodc1 yang telah diubah nama menjadi Data1 ubah *default type-*nya menjadi *UseJet*. Untuk menghubungkan file “*database*” dengan *form* maka pada *properties* dari Data1, lalu pilih *DatabaseName*, cari file “*database*” yang tersimpan tadi.

BAB IV

PENGUJIAN, ANALISIS DAN PEMBAHASAN

4.1 Metode Pengujian

Berdasarkan spesifikasi sistem yang telah dijelaskan sebelumnya, selanjutnya dilakukan pengujian terhadap sistem menggunakan beberapa metode pengujian. Tujuan pengujian ini untuk membuktikan apakah sistem yang diimplementasikan telah memenuhi spesifikasi yang telah direncanakan sebelumnya. Hasil pengujian akan dimanfaatkan untuk menyempurnakan kinerja sistem dan sekaligus digunakan dalam pengembangan lebih lanjut.

Metode pengujian dipilih berdasarkan fungsi operasional dan beberapa parameter yang ingin diketahui dari sistem tersebut. Data yang diperoleh dari metode pengujian yang dipilih tersebut dapat memberikan informasi yang cukup untuk keperluan penyempurnaan sistem.

Dalam penelitian ini dipilih dua macam metode pengujian, yaitu pengujian fungsional dan pengujian kinerja sistem. Pengujian fungsional digunakan untuk membuktikan apakah sistem yang diimplementasikan dapat memenuhi persyaratan fungsi operasional seperti yang direncanakan.

Pengujian kinerja sistem dimaksudkan untuk memperoleh beberapa parameter yang dapat menunjukkan kemampuan dan kehandalan sistem dalam menjalankan fungsi operasionalnya.



Gambar 4.1 Alat Destilasi Minyak Atsiri

4.2 Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya dilakukan dengan mengukur tegangan keluaran dari rangkaian catu daya itu sendiri dengan menggunakan multimeter. Hasil pengukuran dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Pengukuran Tegangan Catu Daya

No.	Sumber Catu Daya (Volt)	Tegangan Terbaca (Volt)
1	5	4,9
2	10	10,1
3	24	24,2

Selisih tegangan catu daya antara sumber catu daya dengan tegangan terbaca oleh multimeter selisihnya tidak begitu jauh sehingga masih dapat memberikan catu daya ke perangkat sistem.

4.3 Pengujian Rangkaian Sensor Suhu LM35C

Dengan melakukan pengukuran LM35C dan karakteristiknya sebagai sensor suhu, didapatkan data seperti pada tabel di bawah :

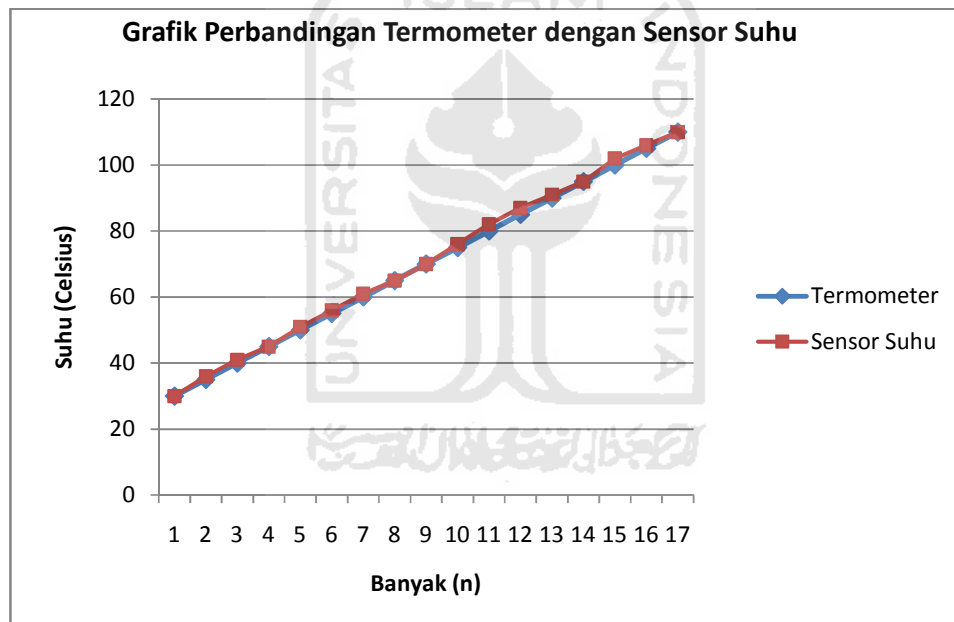
Tabel 4.2 Data Pengukuran Rangkaian LM35C

No.	Pembacaan Termometer (°C)	Pembacaan Sensor Suhu (°C)	Error
1	30	30	0
2	35	36	1
3	40	41	1
4	45	45	0
5	50	51	1
6	55	56	1
7	60	61	1
8	65	65	0
9	70	70	0
10	75	76	1
11	80	82	2
12	85	87	2
13	90	91	1
14	95	95	0
15	100	102	2
16	105	106	1
17	110	110	0

Toleransi error dari pengukuran rangkaian LM35C di atas adalah :

$$\text{Toleransi Error} = \frac{\sum |Error|}{n} \quad (4.1)$$

Data di atas diambil berdasarkan perbandingan langsung antara termometer suhu manual (yang menggunakan air raksa) dengan sensor suhu LM35. Dari data di atas, menggunakan persamaan rumus 4.1 dengan jumlah *error* sama dengan 14 dan banyaknya (*n*) sama dengan 17, maka didapat hasil toleransi *error* sama dengan 0,823 °C. Nilai toleransi kesalahan hasil pengukuran ini dapat disebabkan banyak faktor, salah satunya adalah sensitifitas antara termometer manual dan sensor suhu berbeda.



Gambar 4.2 Grafik Perbandingan Termometer dengan Sensor Suhu

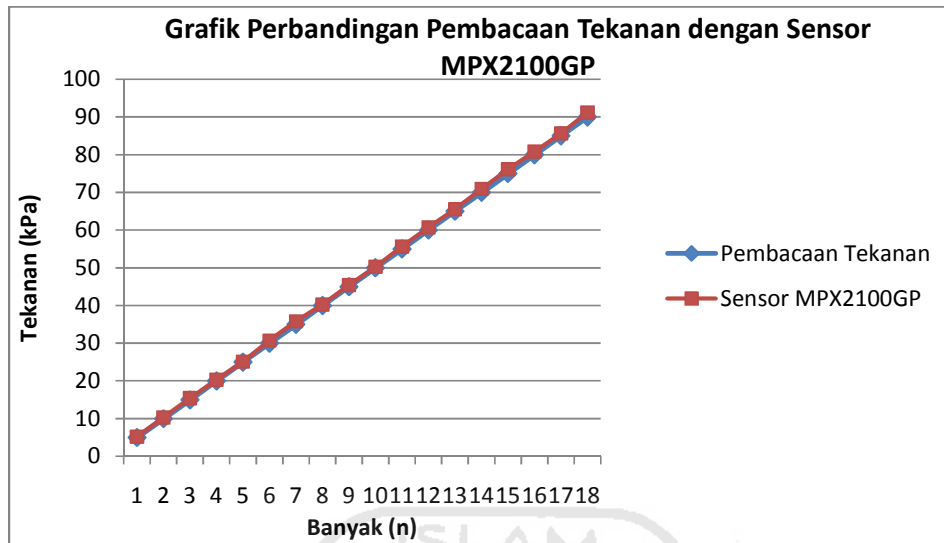
4.4 Pengujian Rangkaian Sensor Tekanan MPX2100GP

Dengan melakukan pengukuran MPX2100GP dan karakteristiknya sebagai sensor tekanan, didapatkan data seperti pada tabel di bawah :

Tabel 4.3 Data Pengukuran Rangkaian MPX2100GP

No.	Pembacaan Sensor MPX2100GP (kPa)	Pembacaan Tekanan (kPa)	Error
1	5	5,2	0,2
2	10	10,3	0,3
3	15	15,4	0,4
4	20	20,3	0,3
5	25	25,1	0,1
6	30	30,7	0,7
7	35	35,8	0,8
8	40	40,2	0,2
9	45	45,4	0,4
10	50	50,3	0,3
11	55	55,6	0,6
12	60	60,7	0,7
13	65	65,5	0,5
14	70	70,9	0,9
15	75	76,1	1,1
16	80	80,8	0,8
17	85	85,6	0,6
18	90	91,2	1,2

Data di atas diambil berdasarkan perbandingan langsung antara manometer digital dengan sensor tekanan MPX2100GP. Dari data di atas, menggunakan persamaan rumus 4.1 dengan jumlah *error* sama dengan 10,1 dan banyaknya (*n*) sama dengan 18, maka didapat hasil toleransi *error* sama dengan 0,561 kPa. Nilai toleransi kesalahan dari hasil pengukuran ini dapat disebabkan banyak faktor, salah satunya adalah sensitifitas antara manometer digital dan sensor tekanan berbeda.



Gambar 4.3 Grafik Perbandingan Pembacaan Tekanan dengan Sensor

MPX2100GP

Dalam pengujian nilai tegangan pada kondisi tekanan awal sama dengan 0 (nol) terdapat pada tabel di bawah ini :

Tabel 4.4 Data Pengukuran Kondisi Tekanan 0 kPa

No.	Kondisi Tekanan 0 kPa	Tegangan (mV)
1	0	945
2	0	945
3	0	945
4	0	945
5	0	945

Dari tabel di atas, setelah dilakukan pengukuran ulang sebanyak lima kali didapat data pengukuran tekanan awalan sama dengan 0 (nol) yaitu dengan nilai tegangan 945 mV. Nilai tersebut digunakan sebagai nilai acuan dalam pengkalibrasian tekanan terhadap tegangan.

4.5 Pengujian Rangkaian Saklar

Pengukuran ini bertujuan untuk melihat apakah *relay* berfungsi dengan baik. Dengan memberikan logika 1 (*high*) pada PortC, maka arus akan mengalir menuju basis dan menginduksi magnet *coil* dari *relay*, sehingga *relay* yang pada mulanya *Normally Open* atau logika 0 (*low*) menjadi tertutup dan menyalakan *solenoid valve*.

Tabel 4.5 Data Hasil Pengujian *Relay*

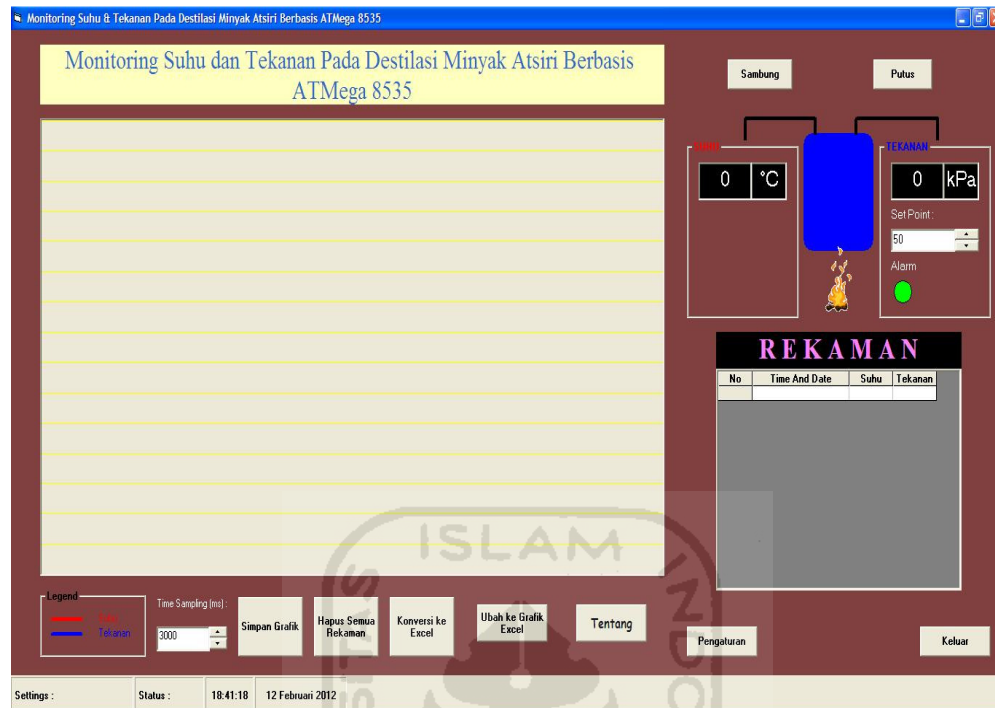
Input dari Port C	Kondisi <i>Relay</i>	Keterangan
Logika 1 (<i>high</i>)	<i>Close</i>	<i>Solenoid valve</i> aktif
Logika 0 (<i>low</i>)	<i>Open</i>	<i>Solenoid valve</i> mati

4.6 Pengujian dan Pembahasan Sistem Keseluruhan

Pengujian alat secara keseluruhan dilakukan dengan proses sebagai berikut :

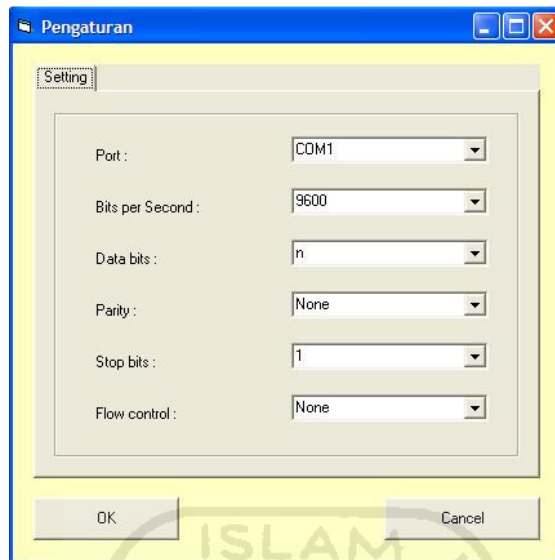
1. Menghubungkan rangkaian seperti pada blok diagram.
2. Menghubungkan perangkat keras dengan perangkat lunak dan menjalankan aplikasi monitoring suhu dan tekanan dengan program *Visual Basic 6.0*
3. Mengamati kinerja sistem.

Tampilan awal aplikasi monitoring suhu dan tekanan pada destilasi minyak atsiri berbasis ATmega8535 dengan tampilan PC dapat dilihat pada Gambar 4.4



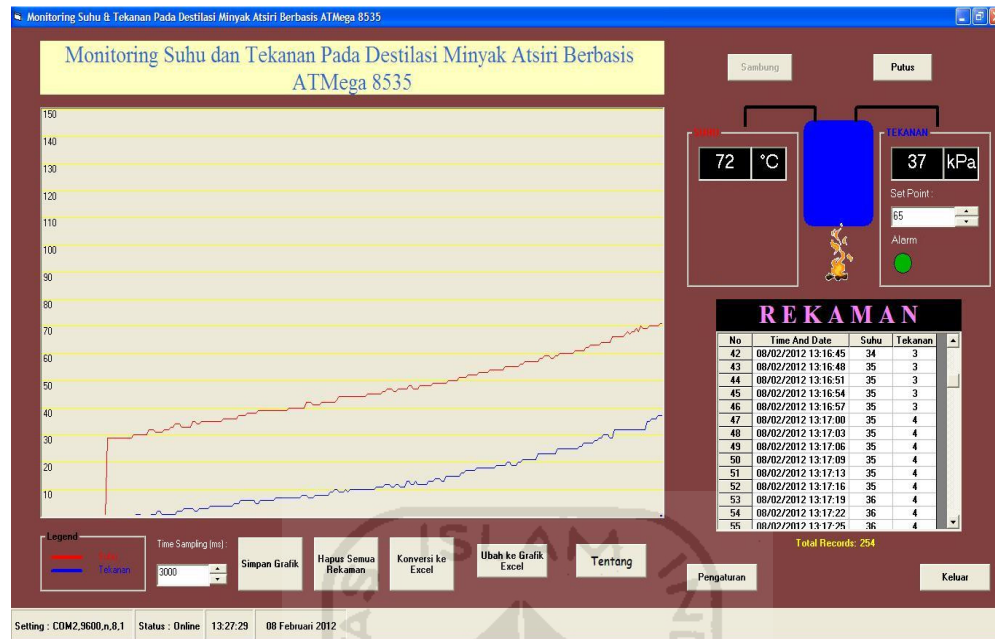
Gambar 4.4 Tampilan Awal Aplikasi Monitoring Suhu dan Tekanan Pada Destilasi Minyak Atsiri

Sebelum ditekan tombol sambung, maka terlebih dahulu masuk ke pengaturan dengan menekan tombol pengaturan. Di bagian pengaturan akan dipilih *port*, *bit per second*, *data bits*, *parity*, *stop bits* dan *flow control* sesuai dengan kebutuhan. Gambar 4.5 adalah pengaturan pada aplikasi program monitoring suhu dan tekanan pada destilasi minyak atsiri.



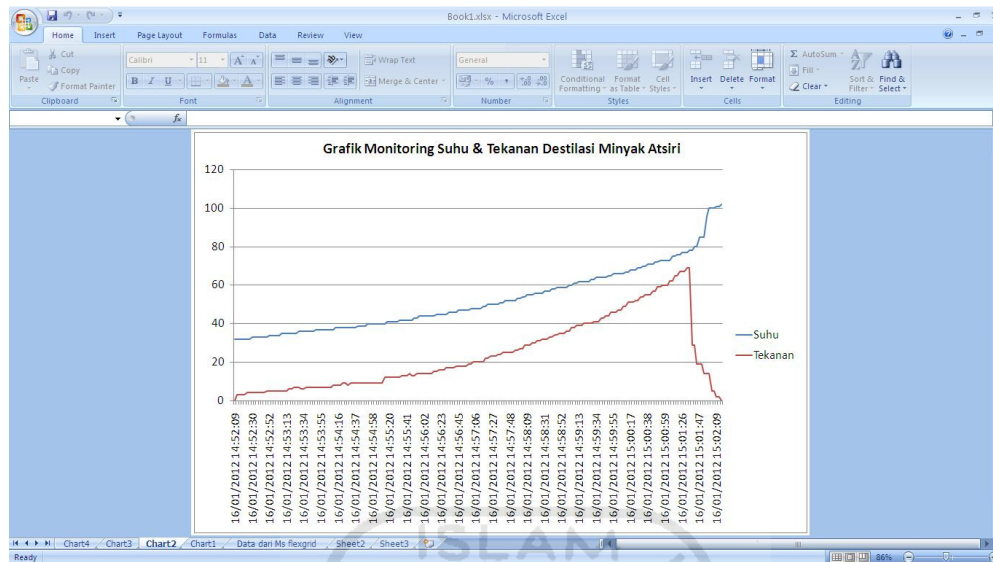
Gambar 4.5 Aplikasi Pengaturan

Setelah diatur, tekan “OK”, lalu tekan “Sambung” rangkaian mikrokontroler dengan sensor suhu dan tekanan yang tersambung akan mengirimkan data serial ke komputer dengan komunikasi serial. Untuk pembacaan tekanan, jika pembacaan tekanan $\geq$ (lebih besar sama dengan) setpoint tekanan maka gambar led akan berkedip merah, alarm akan menyala, akan komputer mengirimkan karakter “X” ke mikrokontroler untuk mengaktifkan *solenoid valve* yang terhubung *relay* dengan mikrokontroler sebaliknya jika pembacaan suhu $\leq$ (lebih kecil sama dengan) setpoint tekanan maka gambar led akan berhenti berkedip menjadi hijau, alarm akan mati. Jika pembacaan tekanan $\leq$ (lebih kecil sama dengan) nilai 10 yangmana nilai tekanan dalam kPa (kilopascal) maka komputer akan mengirimkan karakter “Y” ke mikrokontroler untuk mematikan *solenoid valve* yang terhubung *relay* dengan mikrokontroler.



Gambar 4.6 Tampilan Proses Monitoring Suhu dan Tekanan Pada Destilasi Minyak Atsiri

Data rekaman monitoring suhu dan tekanan terekam pada database Microsoft Access, penghubung Microsoft Access dengan Form Visual Basic adalah komponen Microsoft ADO Data Control 6.0 (OLEDB). Gambar grafik yang ada di program visual basic dapat *dicapture* atau disimpan ke dalam bentuk JPEG atau dalam bentuk Excel.



Gambar 4.7 Grafik Excel Monitoring Suhu dan Tekanan Pada Destilasi Minyak
Atsiri

The screenshot shows the Microsoft Access Datasheet View for a table named 'Data_Monitoring'. The table contains 12 rows of data, each representing a time-based measurement of temperature and pressure.

No	TimeAndDate	Suhu	Tekanan
1	08/02/2012 13:16:45	34	3
2	08/02/2012 13:16:48	35	3
3	08/02/2012 13:16:51	35	3
4	08/02/2012 13:16:54	35	3
5	08/02/2012 13:16:57	35	3
6	08/02/2012 13:17:00	35	4
7	08/02/2012 13:16:03	35	4
8	08/02/2012 13:16:06	35	4
9	08/02/2012 13:16:09	35	4
10	08/02/2012 13:16:13	35	4
11	08/02/2012 13:16:16	35	4
12	08/02/2012 13:16:19	36	4

Gambar 4.8 Database Microsoft Access

BAB V

PENUTUP

Setelah dilakukan pengujian alat, maka diperoleh beberapa kesimpulan dan saran yang diharapkan berguna untuk perbendaharaan ilmu dan teknologi serta bagi kelanjutan dalam penyempurnaan proyek akhir ini.

5.1 Kesimpulan

Berdasarkan hasil studi dan analisa terhadap pembuatan perangkat keras (*Hardware*), maka dapat diambil beberapa kesimpulan, antara lain :

1. Rangkaian sensor suhu dan sensor tekanan dapat bekerja dengan baik, dengan penyimpangan yang kecil. Dimana toleransi *error* suhu adalah 0,823 °C dan toleransi *error* tekanan adalah 0,561 kPa. Nilai toleransi *error* dari kedua sensor didapat karena perbedaan sensitifitas antara sensor dan alat pembanding.
2. Sistem monitoring suhu dan tekanan pada destilasi minyak atsiri dapat menampilkan grafik suhu dan tekanan secara *realtime*, grafik pun dapat disimpan kedalam format JPEG atau Excel.
3. Sistem pada aplikasi ini juga dapat menyimpan data dari sensor suhu dan tekanan ke *database Microsoft Acces* juga dapat dikonversikan ke dalam bentuk *Microsoft Excel*.

5.2 Saran - saran

Beberapa saran yang dapat dipertimbangkan untuk pengembangan di masa mendatang, antara lain :

1. Pada sistem monitoring suhu dan tekanan pada destilasi minyak atsiri ini sensor suhu yang dipakai masih maksimal 110 °C dan sensor tekanan maksimal 100 kPa. Untuk itu diharapkan kedepannya dengan range maksimal kedua sensor melebihi range sebelumnya.
2. Pada pembuatan proyek akhir ini tidak ada pengkondisian/pemberitahuan keadaan *solenoid valve on/off* saat *user* ingin mengontrol/memonitor suhu dan tekanan. Untuk pengembangannya diharapkan supaya dapat memberitahukan keadaan/status *solenoid valve* sebelum *user* memilih menu untuk mengontrol.
3. Pada sistem monitoring suhu dan tekanan belum ada pemanas atau *heater* listrik, sehingga pengembangan ke depannya dapat direalisasikan.

DAFTAR PUSTAKA

- Atmel Corporation, 2011, "*ATMEGA8535(L) Datasheet*", available at www.atmel.com.
- Budianto Restu, 2010. "*Perancangan Sistem Monitoring Suhu dan Tekanan Berbasis Web pada Mini Plant Di Laboratorium Workshop Instrumentasi Teknik Fisika ITS Surabaya*". Skripsi, tidak diterbitkan. Surabaya: ITS.
- Fahlevy, 2006. "*Monitoring Suhu Dan Tekanan Sistem Boiler Dengan Tampilan Pada PC*". Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri, Teknik Elektro, Universitas Islam Indonesia.
- Firdaus, 2006. "*7 Jam Belajar Interaktif Visual Basic 6.0 untuk Orang Awam*". Palembang: Maxikom.
- Freescale Semiconductor, 2002. "*100 kPa On-Chip Temperature Compensated and Calibrated Silicon Pressure Sensors*". Available at www.freescale.com.
- Info Kit, Trubus, Vol. 07, Juni 2009. "*Minyak Atsiri*". Depok: PT. Trubus Swadaya.
- Iswanto, 2008. "*Design dan Implementasi Sistem Embedded Mikrokontroller ATmega8535 dengan Bahasa Basic*". Yogyakarta: Penerbit Gava Media.
- Maiyana, Eka, 2011, "*Rancang Bangun Alat Ukur Suhu yang Terintegrasi Dengan Tekanan Darah Berbasis Mikrokontroler Atmega8535*". Skripsi, tidak diterbitkan. Surabaya: ITS.
- Malvino, Albert Paul, 2003. "*Prinsip-prinsip Elektronika*" Terjemahan Alb Joko Sutoso, Jakarta: Salemba Teknik.
- Maxim, 1996. "*±15kV ESD-Protected, +5V RS-232 Transceivers*". Available at www.maxim-ic.com.

- National Semiconductor, 1994. "*LM35/LM35A/LM35C/LM35CA/LM35D Precision Centigrade Temperature Sensors*". Available at www.datasheetcatalog.com.
- Nuryoto, Jayanudin, & Rudi Hartono, 2011. "*Karakterisasi Minyak Atsiri dari Limbah Daun Cengkeh*". Makalah disampaikan pada Prosiding Seminar Nasional Teknik Kimia "Kejuangan - Pengembangan Teknologi Kimia untuk Pengolahan Sumber Daya Alam Indonesia". Yogyakarta. 22 Februari 2011.
- Prasetia Ratna, & Catur Edi Widodo, 2004. "*Visual Basic 6.0*". Yogyakarta: Penerbit Andi.
- Schultz, Warren, 1997. "*A Simple Sensor Interface Amplifier*". Available at www.freescale.com.
- STMicroelectronics, 2002. "*LM158A-LM258A-LM358A, Low Power Dual Operational Amplifiers*". Available at www.st.com or www.datasheetcatalog.com.



LAMPIRAN-LAMPIRAN



Lampiran 1

```
'=====
=====
'Program BASCOM
'Oleh Asep Juliyansyah
'NIM : 04524058
'Teknik Elektro, Universitas Islam Indonesia
'=====
=====
```

```
$regfile = "m8535.dat"
$crystal = 11059200
$baud = 9600
Config Adc = Free , Prescaler = Auto , Reference = Avcc
Dim Char As String * 10
Ddrc = &B11111111
```

```
Dim W As Integer , V As Word , S As Word , A As Word , C As Word , X As Word , V2 As
Word , Z As Long , T As Word
Start Adc
'Mulai:
Do
```

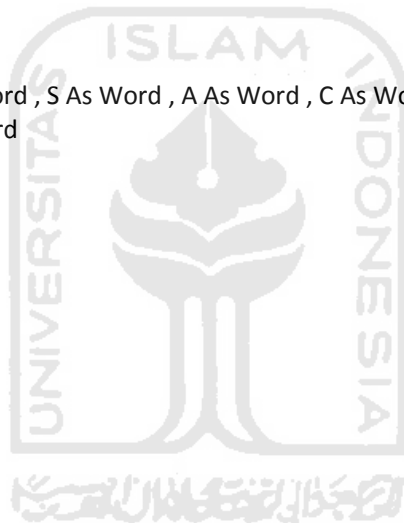
```
W = Getadc(0)
X = Getadc(1)
V = W * 5
A = V / 2
S = A / 10
```

```
V2 = X * 5
Z = V2 - 945
T = Z / 40.4
```

```
If T >= 1600 Then T = 0
```

```
Print "S " ; S
Print "T " ; T
```

```
Char = Inkey()
Select Case Char
Case "X" : Portc = &B11111111
Case "Y" : Portc = &B00000000
End Select
Wait 1
Loop
```



```
'=====
=====
=====
'Monitoring Suhu dan Tekanan Pada Destilasi Minyak Atsiri Berbasis ATmega 8535
dengan Tampilan PC
'Desain Program oleh : Asep Juliyansyah
'Universitas Islam Indonesia, Teknik Elektro 2004
'=====
=====
=====
Option Explicit
Private Declare Function PlaySound Lib "winmm.dll" Alias "PlaySoundA" (ByVal
lpzName As String, ByVal hModule As Long, ByVal dwFlags As Long) As Long
' Pixels we move horizontally at each clock tick.
Private Const m_Dx As Single = 3
Private Const m_Ex As Single = 3
Dim c
Dim sensor
Dim db As Database
Dim rs As Recordset
Private n_OldValue As Integer
Private m_OldValue As Integer
'Untuk database
Public Sub setFlex()
Dim T As Single
Dim i As Single

Set db = OpenDatabase(App.Path & "\database.mdb")
Set rs = db.OpenRecordset("Data_Monitoring", dbOpenDynaset)

fg.TextMatrix(0, 0) = "No"
fg.TextMatrix(0, 1) = "Time And Date"
fg.TextMatrix(0, 2) = "Suhu"
fg.TextMatrix(0, 3) = "Tekanan"

fg.ColWidth(0) = 700
fg.ColWidth(1) = 2000
fg.ColWidth(2) = 900
fg.ColWidth(3) = 900

For i = 0 To 3
fg.Row = 0
fg.Col = i
fg.CellFontBold = True
fg.ColAlignment(i) = 3
Next i
End Sub
```

Lampiran 2

```
Sub fillGrid()
Do While Not rs.EOF
Dim T As Single

T = T + 1
fg.TextMatrix(T, 0) = rs.Fields("No")
fg.TextMatrix(T, 1) = rs.Fields("TimeAndDate")
fg.TextMatrix(T, 2) = rs.Fields("Suhu")
fg.TextMatrix(T, 3) = rs.Fields("Tekanan")

totalrec.Caption = "Total Records: " + CStr(fg.Rows - 1)
If Not rs.EOF Then rs.MoveNext
If fg.TextMatrix(fg.Rows - 1, 2) <> "" Then
fg.Rows = fg.Rows + 1
End If
Loop
End Sub
'Bagian untuk menghapus rekaman
Sub delRec()
db.Execute ("Delete * from Data_Monitoring")
fg.Clear
fg.Rows = 2
Call setFlex
totalrec.Caption = "Total Records: 0"
End Sub
'Menyimpan gambar
Private Sub cmdSave_Click()
'SavePicture picGraph.Image, "Grafik.bmp"

Dim X As String
cdlg.Filter = "File Gambar(*.jpg) | *.jpg|" & "File Gambar(*.bmp)|*.bmp|"
cdlg.ShowSave
X = cdlg.FileName

If Trim(X) <> "" Then
SavePicture picGraph.Image, X
End If
End Sub
'Tombol sambung
Private Sub Connect_Click()
On Error GoTo HandleErrors
MSComm1.PortOpen = True
MSComm1.RThreshold = 7
MSComm1.InputMode = comInputModeText
MSComm1.Handshaking = comNone
MSComm1.InputLen = 0
Connect.Enabled = False
```

Lampiran 2

```
tmrGraph.Enabled = True
StatusBar1.Panels("Status").Text = "Status : Online"
Exit Sub
```

```
HandleErrors:
Dim strMessage As String
strMessage = "Silahkan hubungkan perangkat" & vbCrLf & vbCrLf _
& Err.Description
MsgBox strMessage, vbExclamation, "Error"
```

```
End Sub
'Tombol Putus
Private Sub Disconnect_Click()
On Error GoTo HandleErrors
MSComm1.PortOpen = False
Connect.Enabled = True
tmrGraph.Enabled = False
tmrA.Enabled = False
tmrB.Enabled = False
tmrAlarmA.Enabled = False
tmrAlarmB.Enabled = False
```

```
StatusBar1.Panels("Status").Text = "Status : Offline"
Exit Sub
```

```
HandleErrors:
Dim strMessage As String
strMessage = "Port telah terputus" & vbCrLf & vbCrLf & Err.Description
MsgBox strMessage, vbExclamation, "Error"
On Error GoTo 0
End Sub
```

```
'Untuk konfirmasi keluar program
Private Sub exit_Click()
Dim Msg, Style, Title, Response, MyString
Msg = "Apakah Anda ingin keluar ?"
Style = vbYesNo + vbCritical + vbDefaultButton2
Title = "Konfirmasi"
Response = MsgBox(Msg, Style, Title)
If Response = vbYes Then
End
End If
End Sub
```

```
Private Sub Form_Load()
Call setFlex
```

Lampiran 2

Call fillGrid

Dim i As Integer

Dim j As Single

VScroll3.Value = 50

Text7.Text = "50"

VScroll2.Value = 3000

Text2.Text = "3000"

' Menggambar vertikal pada skala 1-150 dan

' skala piksel horizontal.

picGraph.AutoRedraw = True

picGraph.ScaleMode = vbPixels

picGraph.ScaleTop = 150

picGraph.ScaleHeight = -150

' Gambar Garis pandu.

For i = 10 To 150 Step 10

picGraph.Line (0, i)-(picGraph.ScaleWidth, i), vbYellow

Next i

picGraph.Picture = picGraph.Image

End Sub

'Menampilkan statusbar

Public Sub Get_Connect()

StatusBar1.Panels("Setting").Text = "Setting : COM" & MSComm1.CommPort & "," &

MSComm1.Settings

StatusBar1.Panels("Waktu").Text = Format\$(Now, "Long Time")

Connect.Enabled = True

Exit Sub

End Sub

Private Sub graphExcel_Click()

Form3.Show vbModal

End Sub

'tombol hapus rekaman

Private Sub hapus_Click()

Dim Msg, Style, Title, Response, MyString

Msg = "Anda akan kehilangan semua data yang terekam, Apakah ingin lanjut ?"

Style = vbYesNo + vbCritical + vbDefaultButton2

Lampiran 2

```
Title = "Konfirmasi"
Response = MsgBox(Msg, Style, Title)
If Response = vbYes Then
Call delRec
End If
End Sub

'Tombol konversi ke Excel
Private Sub konversi_Click()
FlexGrid_To_Excel fg, fg.Rows, fg.Cols, 1, "Data dari Ms flexgrid"
End Sub

'Komunikasi serial
Private Sub MSComm1_OnComm()
Dim sdata As String
Dim tdata As String
Dim rdata As String

tmrGraph.Interval = Val(Text2)

If (MSComm1.CommEvent = comEvReceive) Then
rdata = MSComm1.Input
If (Left(rdata, 1) = "S") Then
sdata = CInt(Trim(Mid(rdata, 2, 4)))
Text3.Text = sdata
sdata = ""

ElseIf (Left(rdata, 1) = "T") Then
tdata = CInt(Trim(Mid(rdata, 2, 4)))
Text5.Text = tdata
tdata = ""

End If
End If

'Menseting Alarm 1 untuk suhu
If ((Val(Text3.Text)) >= (VScroll1.Value)) Then
a1(0).BackColor = RGB(255, 0, 0)
boiler.BackColor = vbRed

tmrAlarmA.Enabled = True
tmrAlarmA.Interval = 300
ElseIf ((Val(Text3.Text)) <= (VScroll1.Value)) Then
a1(0).BackColor = RGB(0, 180, 0)
```

Lampiran 2

```
tmrAlarmA.Enabled = False
boiler.BackColor = vbBlue

tmrA.Enabled = True
tmrA.Interval = 300
End If

'Menseting Alarm 2 untuk tekanan
If ((Val(Text5.Text)) >= (VScroll3.Value)) Then
    a1(1).BackColor = RGB(255, 0, 0)
    boiler.BackColor = RGB(255, 0, 0)
    MSComm1.Output = "X" & Chr$(13)
    'MSComm1_OnComm
    tmrAlarmB.Enabled = True
    tmrAlarmB.Interval = 300
ElseIf ((Val(Text5.Text)) <= (VScroll3.Value)) Then
    a1(1).BackColor = RGB(0, 180, 0)
    tmrAlarmB.Enabled = False
    boiler.BackColor = RGB(0, 0, 255)
    tmrB.Enabled = True
    tmrB.Interval = 300
If Val(Text5.Text) <= 10 Then
    MSComm1.Output = "Y" & Chr$(13)
    'MSComm1_OnComm

End If
End If
End Sub

'Tombol penampil pengaturan
Private Sub Pengaturan_Click()
Form2.Show
End Sub

'untuk profil pembuat
Private Sub tentang_Click()
Form1.Show vbModal
End Sub

'mematikan kedip Alarm 2
Private Sub tmrB_Timer()
a1(1).Visible = True
boiler.Visible = True
End Sub

'Menseting timer kedip Alarm 2
Private Sub tmrAlarmB_Timer()
If a1(1).Visible = True Then
a1(1).Visible = False
```

Lampiran 2

```
Else
a1(1).Visible = True
End If

If boiler.Visible = True Then
boiler.Visible = False
Else
boiler.Visible = True
End If

'Set suara alarm 2
Static suara_alarm
If ((Val(Text5.Text)) >= (VScroll3.Value)) Then
If suara_alarm > 0 Then suara_alarm = suara_alarm - 1
If suara_alarm = 0 Then
' bunyikan suara beep
'If Dir("beep.wav") <> "" Then
If App.Path & "\\beep.wav" <> "" Then
PlaySound "beep.wav", 0, 1
End If
' alarm bunyi tiap 1500ms/interval_timer
suara_alarm = (1500 / tmrAlarmA.Interval)
End If
End If
End Sub

'Menampilkan waktu dan tanggal
Private Sub tmrClock_Timer()

StatusBar1.Panels("Waktu").Text = Format$(Now, "Long Time")
StatusBar1.Panels("Tanggal").Text = Format$(Now, "Long Date")
End Sub

'Menampilkan Grapik
Private Sub tmrGraph_Timer()
Dim new_value As Integer
Dim new_value2 As Integer
Dim i As Integer
Dim a As Integer
Dim b As Integer

' Memindahkan data lama.
picGraph.PaintPicture picGraph.Picture, _
0, 1, picGraph.ScaleWidth - m_Dx, 151, _
m_Dx, 1, picGraph.ScaleWidth - m_Dx, 151

' Menghapus tepi kanan dan menggambar garis panduan
picGraph.Line (picGraph.ScaleWidth - m_Dx, 1)-(picGraph.ScaleWidth - 1, 151),
picGraph.BackColor, BF
```

Lampiran 2

```
For i = 10 To 150 Step 10
    picGraph.Line (picGraph.ScaleWidth - m_Dx, i)-(picGraph.ScaleWidth, i), vbYellow
Next i

' Plot nilai baru.
a = Val(Text3)

new_value = a
picGraph.Line (picGraph.ScaleWidth - 1 - m_Dx, m_OldValue)-(picGraph.ScaleWidth - 1, new_value), vbRed
m_OldValue = a
picGraph.Picture = picGraph.Image

b = Val(Text5)

new_value2 = b
picGraph.Line (picGraph.ScaleWidth - 1 - m_Ex, n_OldValue)-(picGraph.ScaleWidth - 1, new_value2), vbBlue
n_OldValue = b
picGraph.Picture = picGraph.Image

picGraph.CurrentY = 10
picGraph.CurrentX = 2
picGraph.Print "10"

picGraph.CurrentY = 20
picGraph.CurrentX = 2
picGraph.Print "20"

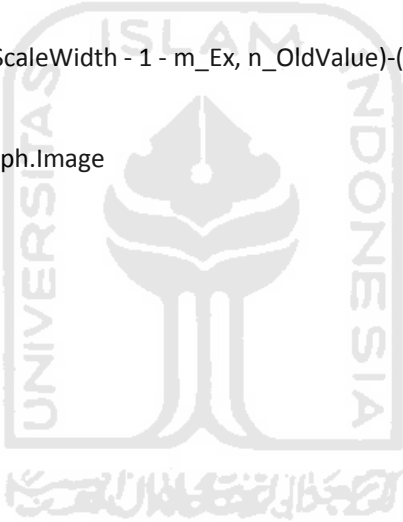
picGraph.CurrentY = 30
picGraph.CurrentX = 2
picGraph.Print "30"

picGraph.CurrentY = 40
picGraph.CurrentX = 2
picGraph.Print "40"

picGraph.CurrentY = 50
picGraph.CurrentX = 2
picGraph.Print "50"

picGraph.CurrentY = 60
picGraph.CurrentX = 2
picGraph.Print "60"

picGraph.CurrentY = 70
picGraph.CurrentX = 2
```

The image contains a large, light gray watermark logo of Universitas Islam Indonesia. The logo is circular with a stylized open book and a quill pen in the center. The words "UNIVERSITAS ISLAM INDONESIA" are written around the perimeter of the circle, and there is Arabic calligraphy at the bottom.

Lampiran 2

```
picGraph.Print "70"
```

```
picGraph.CurrentY = 80  
picGraph.CurrentX = 2  
picGraph.Print "80"
```

```
picGraph.CurrentY = 90  
picGraph.CurrentX = 2  
picGraph.Print "90"
```

```
picGraph.CurrentY = 100  
picGraph.CurrentX = 2  
picGraph.Print "100"
```

```
picGraph.CurrentY = 110  
picGraph.CurrentX = 2  
picGraph.Print "110"
```

```
picGraph.CurrentY = 120  
picGraph.CurrentX = 2  
picGraph.Print "120"
```

```
picGraph.CurrentY = 130  
picGraph.CurrentX = 2  
picGraph.Print "130"
```

```
picGraph.CurrentY = 140  
picGraph.CurrentX = 2  
picGraph.Print "140"
```

```
picGraph.CurrentY = 150  
picGraph.CurrentX = 2  
picGraph.Print "150"
```

```
fg.TextMatrix(fg.Rows - 1, 0) = CStr(fg.Rows - 1)  
fg.TextMatrix(fg.Rows - 1, 1) = CStr(Now)  
fg.TextMatrix(fg.Rows - 1, 2) = Text3.Text  
fg.TextMatrix(fg.Rows - 1, 3) = Text5.Text
```

```
totalrec.Caption = "Total Records: " + CStr(fg.Rows - 1)  
rs.AddNew  
rs.Fields("No").Value = CStr(fg.Rows - 1)  
rs.Fields("TimeAndDate").Value = Now  
rs.Fields("Suhu").Value = Val(fg.TextMatrix(fg.Rows - 1, 2))  
rs.Fields("Tekanan").Value = Val(fg.TextMatrix(fg.Rows - 1, 3))  
rs.Update  
fg.Rows = fg.Rows + 1
```



Lampiran 2

```
End Sub
'Scroll setpoin Suhu
Private Sub VScroll1_Change()
Text1.Text = CStr(VScroll1.Value)
End Sub
'Scroll Time sampling
Private Sub VScroll2_Change()
Text2.Text = CStr(VScroll2.Value)
End Sub
'Scroll setpoin Tekanan
Private Sub VScroll3_Change()
Text7.Text = CStr(VScroll3.Value)
End Sub
```

```
'=====
'Form pengaturan
Private Sub Form_Load()
Dim i As Integer, Setting As String, Offset As Integer

'Membuka Port Setting
For i = 1 To 20
    cbPort.AddItem "COM" & Trim$(Str$(i))
Next i

'Membuka Speed Setting
cbBitsPerSecond.AddItem "600"
cbBitsPerSecond.AddItem "1200"
cbBitsPerSecond.AddItem "2400"
cbBitsPerSecond.AddItem "4800"
cbBitsPerSecond.AddItem "9600"
cbBitsPerSecond.AddItem "14400"
cbBitsPerSecond.AddItem "19200"
cbBitsPerSecond.AddItem "28800"
cbBitsPerSecond.AddItem "38400"
cbBitsPerSecond.AddItem "57600"
cbBitsPerSecond.AddItem "115200"
cbBitsPerSecond.AddItem "460800"
cbBitsPerSecond.AddItem "921600"

'Membuka Data Bit Setting
cbDataBits.AddItem "5"
cbDataBits.AddItem "6"
cbDataBits.AddItem "7"
cbDataBits.AddItem "8"

'Membuka Parity Setting
```

Lampiran 2

```
cbParity.AddItem "Even"
cbParity.AddItem "Mark"
cbParity.AddItem "None"
cbParity.AddItem "Odd"
cbParity.AddItem "Space"

'Membuka Stop Bit Setting
cbStopBits.AddItem "1"
cbStopBits.AddItem "1.5"
cbStopBits.AddItem "2"

'Membuka Flow Control Setting
cbFlowControl.AddItem "None"
cbFlowControl.AddItem "Xon / Xoff"
cbFlowControl.AddItem "RST"
cbFlowControl.AddItem "Xon / RST"

'Set Default Setting
Settings = FormUtama.MSComm1.Settings

If InStr(Setting, ".") > 0 Then
    Offset = 2
Else
    Offset = 0
End If

cbBitsPerSecond.Text = Left$(Settings, Len(Settings) - 6 - Offset)
Select Case Mid$(Settings, Len(Settings) - 4 - Offset, 1)
Case "e"
    cbParity.ListIndex = 0
Case "m"
    cbParity.ListIndex = 1
Case "n"
    cbParity.ListIndex = 2
Case "o"
    cbParity.ListIndex = 3
Case "s"
    cbParity.ListIndex = 4
End Select

cbDataBits.Text = Mid$(Settings, Len(Settings) - 4 - Offset, 1)
cbStopBits.Text = Right$(Settings, 1 + Offset)
cbPort.ListIndex = FormUtama.MSComm1.CommPort - 1
cbFlowControl.ListIndex = Mid$(FormUtama.MSComm1.Handshaking, 1, 1)
End Sub

Private Sub OK_Click()
```

Lampiran 2

```
On Error GoTo HandleErrors
If FormUtama.MSComm1.PortOpen = True Then
    FormUtama.MSComm1.PortOpen = False
End If
FormUtama.MSComm1.CommPort = cbPort.ListIndex + 1
FormUtama.MSComm1.Settings = Trim$(cbBitsPerSecond.Text) & "," &
Left(cbParity.Text, 1) & "," & Trim$(cbDataBits.Text) & "," & Trim$(cbStopBits.Text)
FormUtama.MSComm1.Handshaking = cbFlowControl.ListIndex
FormUtama.Get_Connect
Unload Me
Exit Sub
```

HandleErrors:

```
Dim strMessage As String
strMessage = "Setting Properti Salah" & vbCrLf & vbCrLf _
& Err.Description
MsgBox strMessage, vbExclamation, "Error"
End Sub
```

```
'=====
' form konversi ke grafik excel
Dim xlApp As Excel.Application
Dim xlWkb As Excel.Workbook
Dim xlSht As Excel.Worksheet
Dim xlChart As Excel.Chart
Private Sub open_Click()
    CommonDialog1.CancelError = True
    On Error GoTo ErrHandler
```

```
Set xlApp = New Excel.Application
CommonDialog1.Filter = "File Excel (*.xlsx)|*.xlsx|" & "File Excel (*.xls)|*.xls|"
CommonDialog1.ShowOpen
Set xlWkb = xlApp.Workbooks.open(CommonDialog1.FileName)
Set xlSht = xlWkb.Worksheets(1)
'jika pengguna menekan tombol cancel
ErrHandler:
Exit Sub
```

End Sub

```
Private Sub tampil_Click()
    On Error Resume Next
    Set xlChart = xlWkb.Charts.Add
    xlChart.ChartType = xlLine
    xlChart.SetSourceData xlSht.Range(Text1), xlColumns
    xlChart.Visible = xlSheetVisible
    xlChart.HasLegend = True
```

Lampiran 2

```
xlChart.HasTitle = True
xlChart.ChartTitle.Caption = "Grafik Monitoring Suhu & Tekanan Destilasi Minyak Atsiri"
xlChart.ChartTitle.Font.Bold = True
xlChart.ChartTitle.Font.Size = 14
xlChart.Legend.Position = xlLegendPositionRight
xlChart.ChartArea.Font.Size = 15
xlChart.ChartArea.Font.Color = vbBlack
xlChart.ChartArea.Select
xlChart.ChartArea.Copy
Picture1.Picture = Clipboard.GetData(vbCFBitmap)

ActiveWorkbook.Save
ActiveWorkbook.Close
End Sub
Private Sub exit_Click()
Unload Me
End Sub
Private Sub Form_Unload(Cancel As Integer)
Exit Sub
End Sub
'=====
'form Tentang
Private Sub Form_Load()

End Sub

Private Sub Timer1_Timer()
If Image1.Visible = True Then
Image1.Visible = False
Image2.Visible = True
ElseIf Image2.Visible = True Then
Image2.Visible = False
Image3.Visible = True
ElseIf Image3.Visible = True Then
Image3.Visible = False
Image4.Visible = True
ElseIf Image4.Visible = True Then
Image4.Visible = False
Image5.Visible = True
ElseIf Image5.Visible = True Then
Image5.Visible = False
Image6.Visible = True
ElseIf Image6.Visible = True Then
Image6.Visible = False
Image7.Visible = True
ElseIf Image7.Visible = True Then
Image7.Visible = False
```

Lampiran 2

```
Image8.Visible = True
ElseIf Image8.Visible = True Then
Image8.Visible = False
Image1.Visible = True
End If
End Sub

'=====
'module untuk konversi ke excel
Public Sub FlexGrid_To_Excel(TheFlexgrid As MSFlexGrid, _
    TheRows As Integer, TheCols As Integer, _
    Optional GridStyle As Integer = 1, Optional WorkSheetName _
    As String)

Dim objXL As New Excel.Application
Dim wbXL As New Excel.Workbook
Dim wsXL As New Excel.Worksheet
Dim intRow As Integer ' counter
Dim intCol As Integer ' counter

If Not IsObject(objXL) Then
    MsgBox "You need Microsoft Excel to use this function", _
        vbExclamation, "Print to Excel"
    Exit Sub
End If

'On Error Resume Next is necessary because
'someone may pass more rows
'or columns than the flexgrid has

'you can instead check for this,
'or rewrite the function so that
'it exports all non-fixed cells
'to Excel

On Error Resume Next

' open Excel
objXL.Visible = True
Set wbXL = objXL.Workbooks.Add
Set wsXL = objXL.ActiveSheet

' Menamai worksheet
With wsXL
    If Not WorkSheetName = "" Then
        .Name = WorkSheetName
    End If
End With
```

Lampiran 2

```
' Mengisi worksheet
For intRow = 1 To TheRows
  For intCol = 1 To TheCols
    With TheFlexgrid
      wsXL.Cells(intRow, intCol).Value = _
        .TextMatrix(intRow - 1, intCol - 1) & " "
    End With
  Next
Next

' format tampilan
For intCol = 1 To TheCols
  wsXL.Columns(intCol).AutoFit
  'wsXL.Columns(intCol).AutoFormat (1)
  wsXL.Range("a1", Right(wsXL.Columns(TheCols).AddressLocal, _
    1) & TheRows).AutoFormat GridStyle
Next

End Sub
```

