

RANCANG BANGUN KENDALI MESIN GILING PADI DENGAN METODE FUZZY



Oleh :

Nama : Stefanus Ardyanto Tandra

NIM : 09.41020.0006

Program : S1 (Strata Satu)

Jurusan : Sistem Komputer

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA DAN KOMPUTER
SURABAYA**

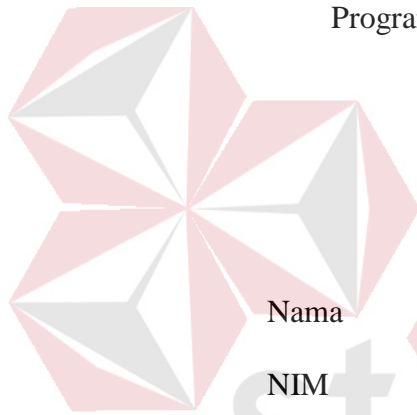
2014

RANCANG BANGUN KENDALI MESIN GILING PADI DENGAN METODE FUZZY

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Oleh :

Nama : Stefanus Ardyanto Tandra

NIM : 09.41020.0006

Program : S1 (Strata Satu)

Jurusan : Sistem Komputer

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA DAN KOMPUTER
SURABAYA**

2014

Tugas Akhir
RANCANG BANGUN KENDALI MESIN GILING PADI DENGAN
METODE FUZZY

Dipersiapkan dan disusun oleh :

Stefanus Ardyanto Tandra

NIM : 09.41020.0006

Telah diperiksa, diuji, dan disetujui oleh Dewan Penguji

Pada : Agustus 2014

Susunan Dewan Penguji

Pembimbing

- I. **Harianto, S.Kom., M.Eng.** _____
- II. **Pauladie Susanto S.Kom.** _____

Penguji

- I. **Helmy Widyantara, S.Kom., M.Eng.** _____
- II. **Susijanto Tri Rasmana, S.Kom., M.T.** _____

Tugas akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Pantjawati Sudarmaningtyas, S.Kom., M.Eng.

Pembantu Ketua Bidang Akademik

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER SURABAYA

PERNYATAAN

Dengan ini menyatakan dengan benar, bahwa Tugas Akhir ini adalah hasil karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam tugas akhir ini adalah semata hanya rujukan yang dicantumkan dalam daftar pustaka saya. Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya tugas akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Surabaya, 22 Agustus 2014



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Stefanus Ardyanto Tandra

NIM : 09.41020.0006

ABSTRAK

Saat ini para petani sudah banyak yang menggunakan mesin pemecah kulit padi yang beredar di pasaran, namun terdapat suatu kekurangan yaitu efisiensi pemakaian. Pada penelitian mesin pemecah kulit padi ini, motor untuk penggerak pemecah padi dikendalikan banyaknya padi yang akan dipecah kulit. Untuk kedepan nya diharapkan penelitian ini dapat digunakan untuk membantu para petani agar dapat mengatur efisiensi penggunaan energi.

Setelah diteliti lebih lanjut penulis menggunakan metode yang digunakan dalam permasalahan ini adalah metode fuzzy, karena merupakan metode yang mampu beradaptasi terhadap perubahan-perubahan yang dipengaruhi oleh lingkungan. Metode fuzzy terbagi menjadi 3 metode yaitu Mamdani, Sugeno, dan Tsukamoto. Dalam hal ini penulis menggunakan metode Sugeno untuk penelitian. Dalam penalaran menggunakan metode Sugeno, untuk mendapatkan output terdapat proses dalam fuzzy yaitu fuzzyfikasi, ruleset, dan defuzzyfikasi. Melalui proses tersebut diharapkan motor penggerak dapat berputar sesuai dengan banyaknya padi dalam tampungan.

Hasil dari proses fuzzy adalah berupa nilai PWM (Pulse Width Modulation), dan akan ditampilkan pada LCD. Dari penelitian ini nilai PWM bernilai 203 jika sensor ultrasound bernilai 2cm dari ketinggian padi, dan nilai PWM 100 jika sensor ultrasound bernilai 26cm dari ketinggian padi. Efisiensi daya jika menggunakan fuzzy adalah 31%.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas berkat rahmat dan lindungan-Nya. Penulis dapat menyelesaikan penyusunan Laporan Tugas Akhir yang merupakan persyaratan dalam menyelesaikan program Sarjana Komputer di Sekolah Tinggi Manajemen Informatika dan Komputer (STIKOM) Surabaya. Laporan Tugas Akhir ini membahas tentang rancang bangun kendali mesin giling padi dengan metode fuzzy.

Dalam menyusun Tugas Akhir ini, Penulis mendapatkan bantuan dari berbagai pihak, baik moril maupun sarana dan fasilitas. Pada kesempatan ini Penulis mengucapkan terima kasih dan penghargaan kepada :

1. Bapak, Ibu, dan Kakak serta Keluarga Besar tercinta yang telah memberikan dukungan moril, fasilitas, materi, nasehat dan semangat.
2. Guru-guru TK, SD, SMP, SMA atas jasanya sehingga penulis dapat melangkah sampai sejauh ini.
3. Seluruh Dosen jurusan Sistem Komputer yang telah memberikan bekal ilmu dalam menempuh kuliah.
4. Bapak Harianto S.kom., M.Eng. dan Pauladie Susanto S.Kom. Selaku Dosen pembimbing yang sudah meluangkan waktunya dalam penyelesaian Tugas Akhir ini.
5. Teresya Yevita Pangandaeng yang telah memberikan support dan semangat kepada penulis.
6. Sahabat yang telah mendukung, membantu, menghibur dalam mengerjakan Tugas Akhir ini hingga selesai khususnya kepada Barry Adam Marela, Ebiet Van Heriyanto, M. Adip, Pradipta Kresna Hadya, Eddy Poerwanto, Yoga, Mustofa, Fahmi Mubarak, serta teman-teman STIKOM Surabaya Prodi Sistem Komputer angkatan 2009 yang belum disebutkan namanya, penulis ucapkan terima kasih.

Penulis menyadari bahwa penyusunan Tugas Akhir ini masih banyak kekurangan. Oleh sebab itu penulis mengharapkan saran dari semua pihak yang ingin memberikan saran baiknya demi perkembangan positif bagi penulis.

Demikian Tugas Akhir ini penulis susun, semoga dapat bermanfaat bagi semua pihak dan penulis sendiri. Akhir kata penulis ucapkan terima kasih.

Surabaya, Agustus 2014

Penulis



DAFTAR ISI

	Halaman
ABSTRAK.....	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Kontribusi	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	
2.1 Padi	5
2.2 Sistem Fuzzy.....	7
2.2.1 Himpunan Fuzzy	9
2.2.2 Fungsi Keanggotaan.....	9
2.2.3 Operator Dasar	10
2.2.4 Metode Sugeno	10
2.3 Sensor Ultrasound.....	11
2.3.1 Prinsip Kerja dan Rangkaian Sensor Ultrasound.....	13
2.3.2 Pemancar Ultrasound.....	14
2.3.3 Penerima Ultrasound.....	15
2.4 Motor DC	17
2.5 Driver motor L298	22

2.6 Mikrokontroler ATmega8535	25
2.6.1 Peta Memory	30
BAB III METODE PENELITIAN	
3.1 Perencanaan Perangkat Keras	32
3.1.1 Rangkaian Mikrokontroler	32
3.1.2 Rangkaian Sensor Parallax PING)))™	39
3.1.3 Konfigurasi motor 21V dengan IC L298	40
3.1.4 Konfigurasi motor 21V dengan IC VNH30SP30	41
3.1.5 Rangkaian LCD	44
3.2 Peletakan Komponen Elektronika	46
3.3 Perancangan Perangkat Lunak	48
3.3.1 Program membaca jarak dan menampilkan LCD	49
3.3.2 Fuzzy	51
BAB IV PENGUJIAN SISTEM	
4.1 Pengujian Minimum Sistem	59
4.1.1 Tujuan	59
4.1.2 Alat yang digunakan	59
4.1.3 Prosedur pengujian	59
4.1.4 Hasil Pengujian	60
2 4.2 Pengujian Minimum Sistem ATmega8535 dengan LCD	61
4.2.1 Tujuan	61
4.2.2 Alat yang digunakan	61
4.2.3 Prosedur pengujian	61
4.2.4 Hasil Pengujian	64
4.3 Pengujian Sensor Ultrasound dengan ATmega8535 dan LCD	65
4.3.1 Tujuan	65
4.3.2 Alat yang digunakan	65

4.3.3 Prosedur pengujian.....	66
4.3.4 Hasil Pengujian	67
4.4 Pengujian Motor Driver L298 dan VNH30SP30 dengan ATmega8535	68
4.4.1 Tujuan	68
4.4.2 Alat yang digunakan	68
4.4.3 Prosedur pengujian.....	68
4.4.4 Hasil Pengujian	69
4.5 Pengujian Metode Fuzzy.....	70
4.5.1 Tujuan	70
4.5.2 Alat yang digunakan	70
4.5.3 Prosedur pengujian.....	70
4.5.4 Hasil Pengujian	70
4.6 Pengujian Keseluruhan Sistem	71
4.6.1 Tujuan	71
4.6.2 Alat yang digunakan	71
4.6.3 Prosedur pengujian.....	72
4.6.4 Hasil Pengujian	72
 BAB V PENUTUP	
5.1 Kesimpulan	74
5.2 Saran	75

DAFTAR TABEL

	Halaman
Tabel 2.1 Mutu Beras.....	6
Tabel 2.2 Detail Motor DC	21
Tabel 2.3 Fungsi khusus PORT B.....	29
Tabel 2.4 Fungsi khusus PORT D	29
Tabel 3.1 Interface Header 1.....	42
Tabel 3.2 Tabel Kebenaran	43
Tabel 3.3 Deskripsi Pin LCD.....	45
Tabel 3.4 Ruleset	56
Tabel 4.1 Konfigurasi minimum system	62
Tabel 4.2 Hasil pembacaan sensor ultrasound.....	67
Tabel 4.3 Pengujian dengan setpoint 28	71
Tabel 4.4 Hasil pengujian keseluruhan sistem.....	72

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Pemetaan Input dan Output.....	8
Gambar 2.2 Sensor PING))) TM	12
Gambar 2.3 Prinsip kerja sensor ultrasound	13
Gambar 2.4 Rangkaian Pemancar Gelombang Ultrasound	14
Gambar 2.5 Rangkaian Penerima Sensor.....	16
Gambar 2.6 Motor DC Sederhana	17
Gambar 2.7 Medan Magnet yang Membawa arus mengelilingi konduktor	18
Gambar 2.8 Konduktor berbentuk U	18
Gambar 2.9 Reaksi Garis Fluks	19
Gambar 2.10 Prinsip Kerja Motor DC.....	20
Gambar 2.11 Pin Connection L298	22
Gambar 2.12 H-Bridge.....	23
Gambar 2.13 H-Bridge mengatur ke kanan	23
Gambar 2.14 H-Bridge mengatur ke kiri	24
Gambar 2.15 Rangkaian L298 Sederhana	25
Gambar 2.16 Arsitektur 8535	27
Gambar 2.17 Konfigurasi Pin	28
Gambar 2.18 Peta Memory program	30
Gambar 3.1 Blok Diagram Keseluruhan Sistem.....	32
Gambar 3.2 Rangkaian Minimum Sistem ATmega8535.....	33
Gambar 3.3 Rangkaian Reset.....	37
Gambar 3.4 Rangkaian Oksilator	38
Gambar 3.5 Rangkaian Sensor Ultrasound.....	39

Gambar 3.6 Duty Cycle	40
Gambar 3.7 Rangkaian Motor Driver dengan L298	41
Gambar 3.8 Rangkaian Motor Driver EMS 30A H-Bridge.....	43
Gambar 3.9 Skema EMS 30A H-Bridge.....	44
Gambar 3.10 Konfigurasi PORTA ke LCD.....	45
Gambar 3.11 Tampilan Keseluruhan	46
Gambar 3.12 Penempatan Sensor Ultrasound	47
Gambar 3.13 Penempatan Motor DC.....	47
Gambar 3.14 Penempatan Minimum System	48
Gambar 3.15 Diagram alir program secara umum pada mikrokontroller.....	49
Gambar 3.16 Diagram alir program pembacaan sensor.....	50
Gambar 3.17 Pemetaan Derajat keanggotaan error	51
Gambar 3.18 Pemetaan Derajat keanggotaan delta error.....	52
Gambar 3.19 Diagram alir fuzzyfikasi.....	55
Gambar 3.20 Diagram alir ruleset.....	56
Gambar 3.21 Diagram alir defuzzyfikasi.....	58
Gambar 4.1 Chip Signature.....	60
Gambar 4.2 Download Program	61
Gambar 4.3 Tampilan LCD	65
Gambar 4.4 Hasil PWM.....	69
Gambar 4.5 Hasil pengujian dengan setpoint 28	71
Gambar 4.6 Tampilan alat sebenarnya.....	73
Gambar 4.7 Hasil padi setelah pecah kulit.....	73

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal sebagai negara agraris dimana sebagian besar penduduknya bermata pencaharian sebagai petani. Banyak diantara petani masih menggunakan pemecah kulit padi dengan menggunakan diesel sebagai penggerak, sistem diesel sendiri tidak akan berhenti berproses sampai bahan bakar habis. Oleh karena itu penulis membuat alat pemecah kulit padi dengan menggunakan metode fuzzy. Bahan dasar alat ini dengan menggunakan energi listrik dari aki untuk menggerakkan motor pemecah kulit padi. Metode yang dipake adalah fuzzy, melalui metode ini kita dapat mengetahui kondisi untuk mendapatkan peforma yang terbaik.

Logika fuzzy mempunyai sifat yang fleksibel, artinya mampu beraptasi dengan perubahan-perubahan, dan ketidakpastian serta memiliki daya penalaran sehingga mudah dimengerti. Sistem kerja alat ini adalah padi yang akan dipecah kulit masuk dalam ruang dalam pecah kulit, dalam ruang ini berisikan robber roll yang berbahan dasar dari karet dan digerakkan dengan motor. Pengaturan motor DC dilakukan dengan proses fuzzyfikasi, rule set, dan defuzifikasi, aturan yang digunakan dalam fuzzy ini adalah dengan metode sugeno, hal ini dikarenakan metode ini memiliki kelebihan secara otomatis mencari nilai-nilai dari parameter baik premis maupun konsekuen dengan algoritma pembelajaran terhadap sekumpulan data.

Berdasarkan nilai eror dan delta eror yang terdeteksi oleh sensor ultrasound, jika nilai eror dan delta eror menuju kearah negatif maka PWM (Pulse Width Modulation) yang diberikan akan semakin besar. Sebaliknya jika nilai eror dan delta eror menuju kearah positif maka PWM (Pulse Width Modulation) yang diberikan akan semakin kecil.

1.2 Rumusan Masalah

Adapun permasalahan yang dihadapi oleh penulis kedepannya dalam proses pengerjaan Tugas Akhir adalah :

1. Bagaimana meningkatkan kinerja mesin pecah kulit menjadi otomatis.
2. Bagaimana membuat sistem untuk mengatur PWM agar motor dapat bergerak cepat saat gabah dalam tampungan banyak dan pelan saat gabah sedikit.
3. Bagaimana membuat *user interface* yang mudah digunakan oleh *user*.

1.3 Batasan Masalah

Dalam perancangan dan pembuatan alat ini, perlu diberikan beberapa batasan permasalahan dengan tujuan agar pembahasan tidak meluas dan menyimpang dari tujuan. Adapun batasan permasalahan dari alat yang akan dibuat ini, antara lain :

1. Tinggi maksimal tempat penampung padi adalah 28cm.
2. Pengaturan PWM berdasarkan proses fuzzy.
3. Motor yang digunakan adalah motor DC.

1.4 Tujuan

Tujuan dari perancangan dan pembuatan perangkat ini adalah Bagaimana meningkatkan kinerja mesin pecah kulit menjadi otomatis. Bagaimana membuat sistem untuk mengatur PWM agar motor dapat bergerak cepat saat gabah dalam tampungan banyak dan pelan saat gabah sedikit. Bagaimana membuat *user interface* yang mudah digunakan oleh *user*. Bagaimana membuat alat yang dapat mengatur efisiensi pemakaian.

1.5 Kontribusi

Penulis mencoba untuk membuat alat dan sistem perontok padi otomatis dengan tujuan meringankan atau memudahkan para petani agar pengeluaran untuk mengupas kulit padi tidak mahal.

Hasil penelitian ini diharapkan dapat membantu para petani, tetapi seperti penelitian lainnya, penelitian ini jauh dari kata sempurna. Untuk itu dibutuhkan penyempurnaan agar menutupi kekurangan-kekurangan yang ada.

1.6 Sistematika Penulisan

Penulisan Tugas akhir ini diatur dan disusun dalam 5 bab yang terdiri dari beberapa sub bab. Secara ringkas uraian materi dari bab pertama hingga bab terakhir adalah sebagai berikut.

BAB I : PENDAHULUAN

Pada bab pendahuluan ini dibahas mengenai latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan, kontribusi, serta sistematika dari penulisan tugas akhir ini.

BAB II : LANDASAN TEORI

Pada bab landasan teori ini dijelaskan tentang Padi, Sistem Fuzzy, Mikrokontroler ATmega8535, Motor DC 24V, Sensor Ultrasound, dan Motor Driver L298.

BAB III : METODE PENELITIAN

Membahas metode penelitian yang digunakan dalam perencanaan sistem dan penulisan tugas akhir ini meliputi perencanaan dalam perancangan perangkat keras dan perangkat lunak.

BAB IV : PENGUJIAN SISTEM

Bab ini menjelaskan mengenai pengujian sistem baik hardware maupun software.

Pengujian hardware meliputi rangkaian minimum sistem ATmega8535, sensor ultrasound, dan driver motor. Sedangkan untuk software meliputi pengujian metode fuzzy dan keseluruhan perangkat.

BAB V : PENUTUP

Merupakan bab penutup yang berisi kesimpulan dan saran.



BAB II

LANDASAN TEORI

Teori-teori yang digunakan dalam perancangan dalam perangkat keras dan perangkat lunak adalah studi kepustakaan berupa data-data literature dari masing-masing komponen, informasi dari internet dan konsep-konsep dari buku penunjang, antara lain :

2.1 Padi

Beras merupakan sumber utama kalori bagi sebagian besar penduduk Indonesia. Pangsa beras pada konsumsi kalori total adalah 54.3% atau dengan kata lain setengah dari intake kalori masyarakat Indonesia bersumber dari beras (Harianto, 2001).

Secara umum mutu beras dapat dikelompokkan ke dalam 4 kategori, yaitu mutu giling, mutu rasa dan mutu tunak, mutu gizi, dan standar spesifik untuk penampakan dan kemurnian biji (misalnya besar, bentuk dan kebeningan beras).

Mutu beras giling dikatakan baik jika hasil proses penggilingan diperoleh beras kepala yang banyak dengan beras patah minimal. Mutu giling ini juga ditentukan dengan banyaknya beras putih atau rendemen yang dihasilkan. Mutu giling ini sangat erat kaitannya dengan nilai ekonomis dari beras. Salah satu kendala dalam produksi beras adalah banyaknya beras pecah sewaktu digiling. Hal ini dapat menyebabkan mutu beras menurun (Allidawati dan Kustianto, 1989).

Saat ini telah dibuat RSNI mengenai mutu beras giling yang dapat dilihat pada tabel 2.1.

Tabel 2.1. Mutu beras: RSNI 01-6128-200x

No.	Komponen Mutu	Satuan	Mutu				
			I	II	III	IV	V
1	Derajat sosoh (min)	%	100	100	95	95	95
2	Kadar air (max)	%	14	14	14	14	14
3	Butir kepala (min)	%	95	89	78	73	60
4	Butir patah total (max)	%	5	10	20	25	35
5	Butir menir (max)	%	0	1	2	2	5
6	Butir merah (max)	%	0	1	2	3	3
7	Butirkuning/rusak (max)	%	0	1	2	3	5
8	Butir mengapur (max)	%	0	1	2	3	5
9	Benda asing (max)	%	0	0.02	0.02	0.05	0.20
10	Butir gabah (max)	Butir/100g	0	1	1	2	3

Penggilingan beras berfungsi untuk menghilangkan sekam dari bijinya dan lapisan aleuron, sebagian maupun seluruhnya agar menghasilkan beras yang putih serta beras pecah sekecil mungkin. Setelah gabah dikupas kulitnya dengan menggunakan alat pecah kulit, kemudian gabah tersebut dimasukkan ke dalam alat penyosoh untuk membuang lapisan aleuron yang menempel pada beras.

Selama penyosohan terjadi, penekanan terhadap butir beras sehingga terjadi butir patah. Menir merupakan kelanjutan dari butir patah menjadi bentuk yang lebih kecil daripada butir patah (Damardjati, 1988).

Menurut Nugraha et al.(1998), nilai rendemen beras giling dipengaruhi oleh banyak faktor yang terbagi dalam tiga kelompok. Kelompok pertama adalah faktor yang mempengaruhi rendemen melalui pengaruhnya terhadap mutu gabah sebagai bahan baku dalam proses penggilingan yang meliputi varietas, teknik budidaya, cekamaman lingkungan, agroekosistem, dan iklim. Kelompok kedua merupakan faktor penentu rendemen yang terlibat dalam proses konversi gabah menjadi beras, yaitu teknik penggilingan dan alat penggilingan. Kelompok ketiga

menunjukkan kualitas beras terutama derajat sosoh yang diinginkan, karena semakin tinggi derajat sosoh maka rendemen akan semakin rendah.

Susut mutu dari suatu hasil giling dapat diidentifikasi dalam nilai derajat sosoh serta ukuran dan sifat butir padi yang dihasilkan. Umumnya semakin tinggi derajat sosoh, persentase beras patah menjadi semakin meningkat pula. Ukuran butir beras hasil giling dibedakan atas beras kepala, beras patah, dan menir (Anonim, 1983). Berdasarkan persyaratan yang dikeluarkan oleh Bulog, beras kepala merupakan beras yang memiliki ukuran lebih besar dari 6/10 bagian beras utuh. Beras patah memiliki ukuran butiran 2/10 bagian sampai 6/10 bagian beras utuh. Menir memiliki ukuran lebih kecil dari 2/10 bagian beras utuh atau melewati lubang ayakan 2.0 mm (Waries, 2006).

2.2 Sistem Fuzzy

Orang yang belum mengenal logika *fuzzy*, pasti akan mengira bahwa logika *fuzzy* adalah sesuatu yang amat rumit dan tidak menyenangkan. Namun, sesekali seorang mulai mengenalnya, ia pasti akan sangat tertarik dan akan menjadi pendatang baru untuk ikut serta mempelajari logika *fuzzy*. Logika *fuzzy* dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika *fuzzy* modern dan metodis baru ditemukan beberapa tahun yang lalu, padahal sebenarnya konsep tentang logika *fuzzy* itu sendiri sudah ada pada diri kita sejak lama.

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruangan output. Sebagai contoh :

- a. Manager pergudangan mengatakan pada manager produksi beberapa banyak persediaan barang pada akhir minggu ini, kemudian manager produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.

- b. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tidaknya pelayanan yang diberikan.
- c. Anda mengatakan pada saya seberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.
- d. Penumpang taksi berkata pada sopir taksi seberapa cepat laju kendaraan yang diinginkan, sopir taksi akan mengatur pijakan gas taksinya.

Contoh pemetaan suatu input-output dalam bentuk grafis



Gambar 2.1 Pemetaan Input Output (Sri Kusumadewi & Hari Purnomo, 2004)

Ada beberapa alasan mengapa orang memilih logika *fuzzy*, antara lain :

1. Konsep logika *fuzzy* mudah dimengerti, karena logika *fuzzy* menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran *fuzzy* tersebut cukup mudah untuk dimengerti.
2. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan perubahan, dan ketidakpastian yang menyertai permasalahan.

3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat, jika diberikan sekelompok data yang cukup homogeny, dan kemudian ada beberapa yang “eksklusif”, maka logika *fuzzy* memiliki kemampuan untuk menangani data eksklusif tersebut.
4. Logika *fuzzy* mampu memodelkan fungsi fungsi nonlinear yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman para pakar secara langsung tanpa melalui proses pelatihan. Dalam hal ini, sering dikenal dengan nama *fuzzy expert system* menjadi bagian terpenting.
6. Logika *fuzzy* dapat bekerjasama dengan teknik kendali secara konvensional. Hal ini umumnya terjadi pada aplikasi di bidang teknik mesin maupun teknik elektro.
7. Logika *fuzzy* didasarkan pada bahasa alami. Logika *fuzzy* menggunakan bahasa sehari-hari sehingga mudah dimengerti.

2.2.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu:

- satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

2.2.2 Fungsi Keanggotaan

Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara

yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

2.2.3 Operator Dasar

a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y))$$

b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y))$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A^c} = 1 - \mu_A(x)$$

2.2.4 METODE SUGENO

Penalaran dengan metode SUGENO hampir sama dengan penalaran MAMDANI, hanya saja output (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau

persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan Metode TSK. Menurut Cox (1994), Metode TSK terdiri-dari 2 jenis, yaitu:

a. Model *Fuzzy* Sugeno Orde-Nol

Secara umum bentuk model *fuzzy* SUGENO Orde-Nol adalah:

IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z=k$. Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

b. Model *Fuzzy* Sugeno Orde-Satu

Secara umum bentuk model *fuzzy* SUGENO Orde-Satu adalah:

IF $(x_1 \text{ is } A_1) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = p_1 * x_1 + \dots + p_N * x_N + q$. Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

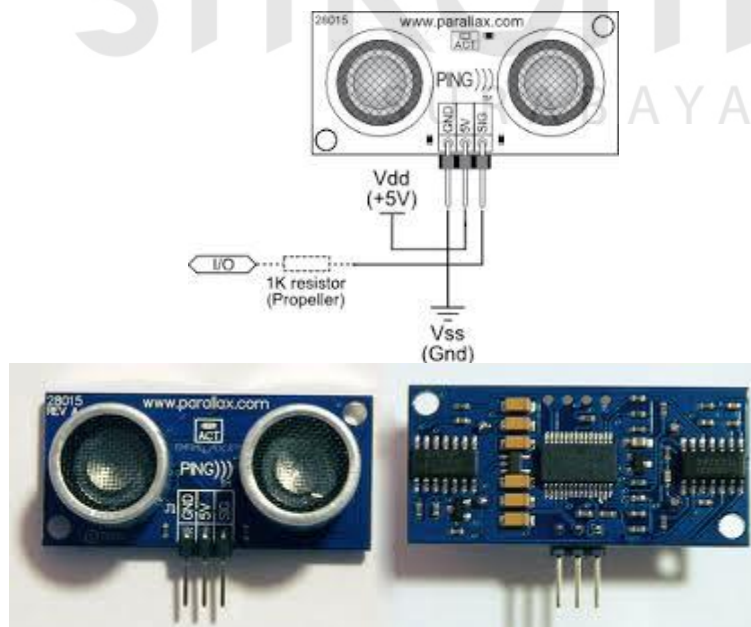
Apabila komposisi aturan menggunakan metode SUGENO, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya. (Sri Kusuma Dewi dan Hari Purnama. 2009)

2.3 Sensor Ultrasound PING)))™

Sensor PING)))™ Dapat mengukur jarak 3cm sampai 300cm. pada dasarnya PING)))™ terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Pada modul PING)))™ terdapat 3 pin yang digunakan untuk jalur power supply (+5V), ground, dan signal. Pin signal

dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan apapun. PING)))™ mendeteksi objek dengan cara mengirimkan suara ultrasonik dan kemudian “mendengarkan” pantulan suara tersebut. PING)))™ hanya akan mengirimkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler (Pulsa high selama $5\mu\text{s}$). suara ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama $200\mu\text{s}$. suara ini dapat merambat di udara dengan kecepatan 344.424m/detik (atau 1cm setiap $29.034\mu\text{s}$), mengenai objek untuk kemudian terpantul kembali ke PING)))™. Selama menunggu pantulan, PING)))™ akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (*low*) ketika suara pantulan terdeteksi oleh PING)))™. Oleh karena itu lebar pulsa tersebut dapat mempresentasikan jarak antara PING)))™ dengan objek. Selanjutnya mikrokontroler cukup mengatur lebar pulsa tersebut dan mengkonversinya dalam bentuk jarak.

Satu hal yang perlu diperhatikan adalah bahwa PING)))™ tidak dapat mengukur objek yang permukaannya dapat menyerap suara seperti busa atau sound damper lainnya. Pengukuran jarak juga akan kacau jika permukaan bergerigi dengan sudut tajam (meruncing).

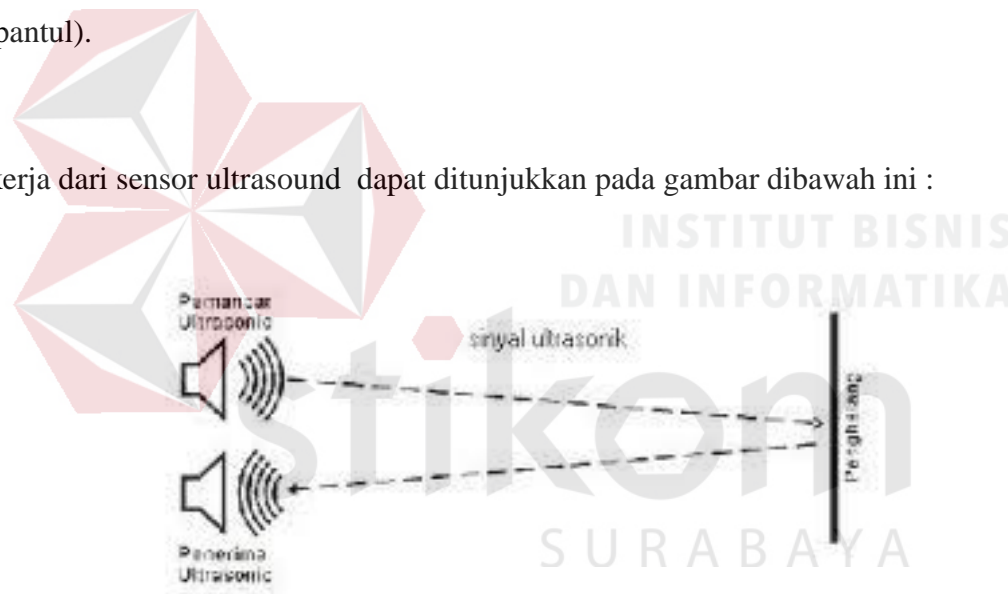


Gambar 2.2 Sensor PING)))™ (Parallax Inc.2005)

2.3.1 Prinsip kerja dan rangkaian sensor ultrasound

Gelombang *ultrasound* adalah gelombang dengan besar frekuensi diatas frekuensi suara yaitu lebih dari 20 KHz. Seperti telah disebutkan bahwa sensor ultrasonik terdiri dari rangkaian pemancar *ultrasound* yang disebut dengan transmitter dan rangkaian penerima ultrasound yang disebut dengan *receiver*. Sinyal *ultrasound* yang dibangkitkan akan dipancarkan dari *transmitter ultrasound*. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver ultrasound*. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depan nya (bidang pantul).

Prinsip kerja dari sensor ultrasound dapat ditunjukkan pada gambar dibawah ini :



Gambar 2.3 Prinsip Kerja Sensor *Ultrasound* (Parallax Inc.2005)

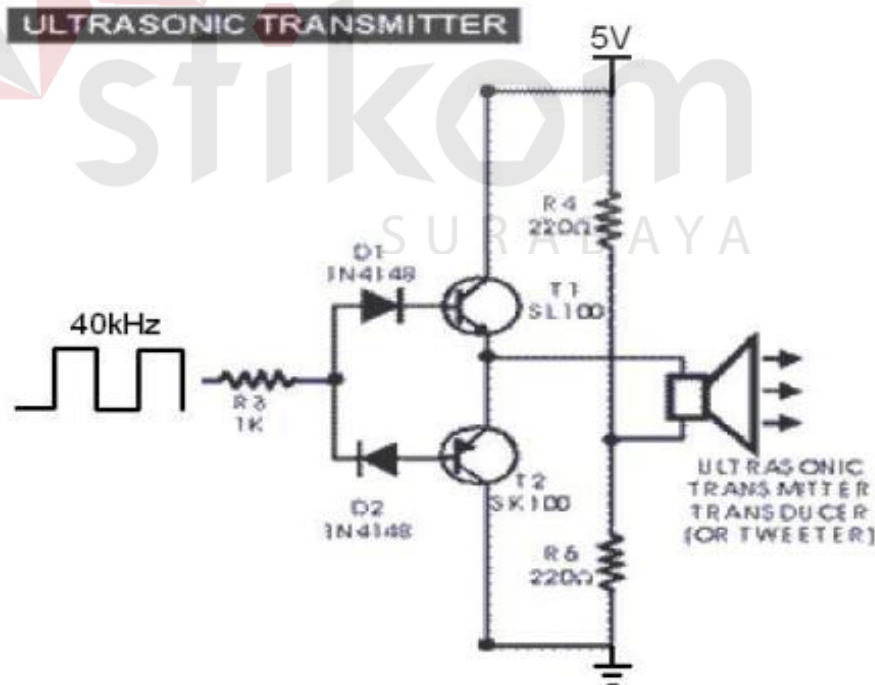
Prinsip kerja sensor ultrasound adalah sebagai berikut :

1. Sinyal dipancarkan oleh pemancar ultrasound. Sinyal tersebut berfrekuensi diatas 20KHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40KHz. Sinyal tersebut dibangkitkan oleh rangkaian pemancar ultrasound.

2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal / gelombang bunyi dengan kecepatan bunyi sekitar 340 m/s. sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima penerima ultrasound
3. Setelah sinyal tersebut sampai di penerima ultrasound, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya, jarak dihitung berdasarkan rumus $S = 340.t/2$ dimana S adalah jarak antara sensor *ultrasound* dengan bidang pantul, dan t adalah selisih waktu antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik.

2.3.2 Pemancar ultrasound (*transmitter*)

Pemancar *ultrasound* ini berupa rangkaian yang memancarkan sinyal sinusoidal berfrekuensi di atas 20 KHz.menggunakan sebuah tranducer transmitter ultrasound.



Gambar 2.4 Rangkaian Pemancar Gelombang Ultrasound (Parallax Inc.2005)

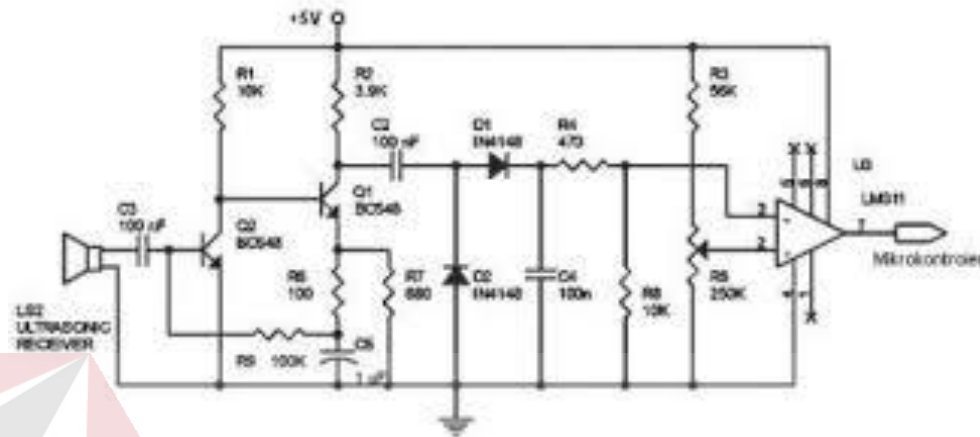
Prinsip kerja dari rangkaian pemancar gelombang ultrasound tersebut adalah sebagai berikut :

1. Sinyal 40 KHz dibangkitkan melalui mikrokontroler.
2. Sinyal tersebut dilewatkan pada sebuah resistor sebesar 3k Ohm untuk pengamanan ketika sinyal itu membias maju rangkaian diode dan transistor.
3. Kemudian sinyal tersebut dimasukkan ke rangkaian penguat arus yang merupakan kombinasi dari 2 buah diode dan 2 buah transistor.
4. Ketika sinyal dari masukan berlogika tinggi (+5V) maka arus akan melewati diode D1 (D1 on), kemudian arus tersebut akan membias transistor T1, sehingga arus yang akan mengalir pada kolektor T1 akan besar sesuai dari penguatan transistor.
5. Ketika sinyal dari masukan berlogika rendah (0V) maka arus akan melawan diode D2 (D2 on), kemudian arus tersebut akan membias transistor T1, sehingga arus yang akan mengalir pada kolektor T2 akan besar sesuai dari penguatan transistor.
6. Resistor R4 dan R6 berfungsi untuk membagi tegangan menjadi 2,5V. sehingga pemancar ultrasound akan menerima tegangan bolak-balik dengan $V_{peak-peak}$ adalah 5V (+2,5V s.d -2,5V).

2.3.3 Penerima ultrasound (*receiver*)

Penerima ultrasound ini akan menerima sinyal ultrasound yang dipancarkan oleh pemancar ultrasound dengan karakteristik frekuensi yang sesuai. Sinyal yang diterima tersebut akan melalui proses filterisasi frekuensi dengan menggunakan rangkaian band pass filter (penyaring pelewat pita), dengan nilai frekuensi yang dilewatkan telah ditentukan. Kemudian sinyal keluarannya akan dikuatkan dan dilewatkan ke rangkaian komparator pembandingan dengan tegangan referensi ditentukan berdasarkan tegangan keluaran penguat pada saat jarak antara sensor dengan sekat/dinding pembatas mencapai jarak minimum. Dapat dianggap keluaran

komparaotr pada kondisi ini adalah high (logika '1') sedangkan jarak yang lebih jauh adalah low (logika '0'). Logika-logika biner ini kemudian diteruskan ke rangkaian pengendali (mikrokontroler).



Gambar 2.5 Rangkaian Penerima Sensor Ultrasound (Parallax Inc.2005)

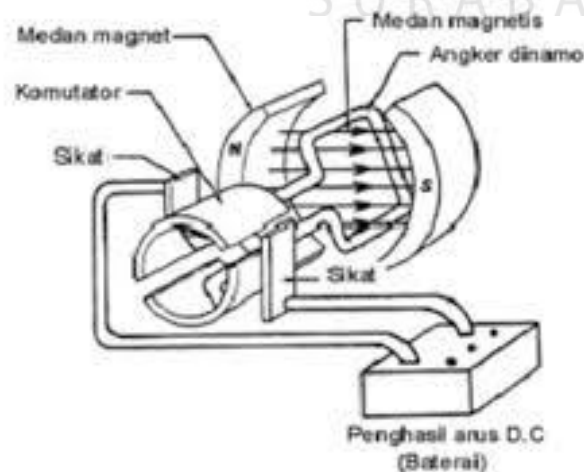
Prinsip kerja dari rangkaian pemancar gelombang ultrasound tersebut adalah sebagai berikut :

1. Pertama-tama sinyal yang diterima akan dikuatkan terlebih dahulu oleh rangkaian transistor penguat Q2.
2. Kemudian sinyal tersebut akan di filter menggunakan high pass filter pada frekuensi > 40kHz oleh rangkaian transistor Q1.
3. Setelah sinyal tersebut dikuatkan dan di filter, kemudian sinyal tersebut akan disearahkan oleh rangkaian diode D1 dan D2.
4. Kemudian sinyal tersebut melalui rangkaian filter low pass filter pada frekuensi < 40kHz melalui rangkaian filter C4 dan R4.
5. Setelah itu sinyal akan melalui komparator op-amp pada U3.
6. Jadi ketika ada sinyal ultrasound yang masuk ke rangkaian, maka komparator akan mengeluarkan logika rendah (0V) yang kemudan akan diproses oleh mikrokontroler untuk menghitung jaraknya.

2.4 Motor DC

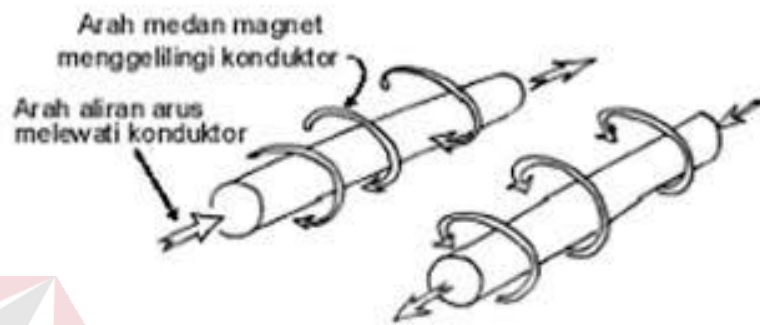
Motor DC merupakan motor arus searah yang berfungsi mengubah tenaga listrik arus searah (listrik DC) menjadi tenaga gerak atau mekanik. Motor dc banyak ditemukan pada peralatan rumah tangga seperti mixer, blender, tape, dll. Selain itu motor DC juga digunakan dalam industry misalkan untuk mengaduk adonan, memutar konveyor, dll.

Motor DC memerlukan supply tegangan yang searah pada kumparan medan untuk diubah menjadi energy mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah searah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik phasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan memiliki arus yang membalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas diantara kutub-kutub magnet seperti gambar di bawah ini :



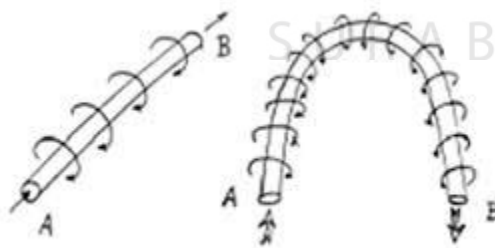
Gambar 2.6 Motor DC Sederhana (Sumanto, 1994)

Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar diatas disebut rotor dinamo. Rotor dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.



Gambar 2.7 Medan magnet yang membawa arus mengelilingi konduktor (Djati, 2010)

Aturan genggam tangan kanan bisa dipakai untuk menentukan arah garis fluks di sekitar konduktor. Genggam konduktor dengan tangan kanan dengan jempol mengarah pada arah aliran arus, maka jari-jari anda akan menunjukkan arah garis fluks. Pada motor listrik konduktor berbentuk U disebut angker dynamo.



Gambar 2.8 Konduktor berbentuk U (Djati, 2010)

Jika konduktor berbentuk U (angker dynamo) diletakkan di antara kutub utara dan kutub selatan yang kuat medan magnet konduktor akan berinteraksi dengan medan magnet kutub.



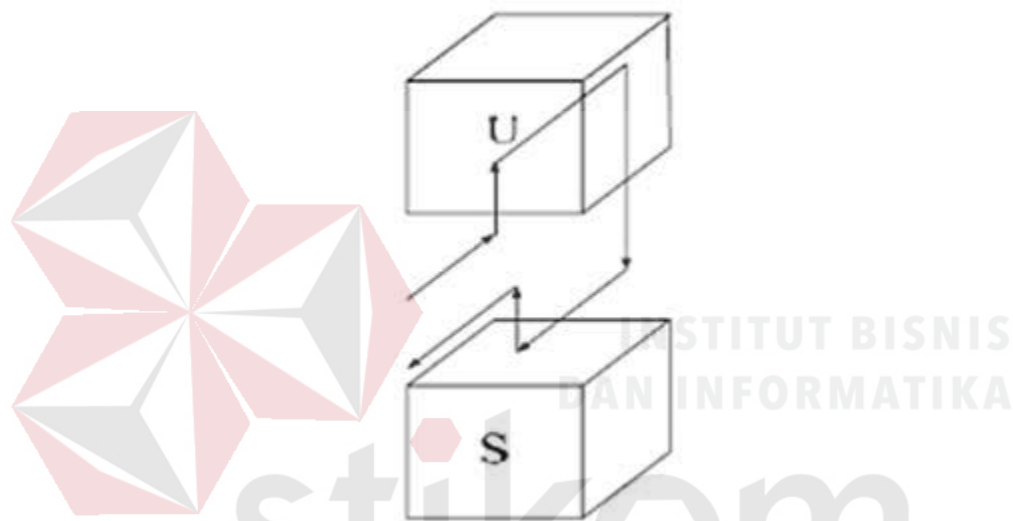
Gambar 2.9 Reaksi garis Fluks (Djati, 2010)

Lingkaran bertanda A dan B pada gambar diatas merupakan ujung konduktor yang dilengkungkan (looped conductor). Arus mengalir masuk melalui ujung A dan keluar melalui ujung B. medan konduktor A yang searah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di bawah konduktor. Konduktor akan berusaha bergerak ke atas untuk keluar dari medan kuat ini. Medan konduktor B yang berlawanan arah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di atas konduktor. Konduktor akan berusaha untuk bergerak turun agar keluar dari medan yang kuat tersebut. Gaya-gaya tersebut akan membuat angker dynamo berputar searah jarum jam.

Mekanisme kerja untuk seluruh jenis motor secara umum :

- a. Arus listrik dalam medan magnet akan memberikan gaya.
- b. Jika kawat yang membawa arus dibangkitkan menjadi lingkaran / loop maka kedua sisi loop, yaitu pada sudut kanan medan magnet akan mendapatkan gaya pada arah yang berlawanan.
- c. Pasangan gaya menghasilkan tenaga putar / torque untuk memutar kumparan.
- d. Motor-motor memiliki beberapa loop pada dinamanya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

Pada motor DC, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melengkapi kumparan jangkar dengan arah tertentu. Konversi dari energy listrik menjadi energy mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energy, sekaligus sebagai tempat berlangsungnya proses perubahan energy. Daerah tersebut dapat dilihat pada gambar dibawah ini.



Gambar 2.10 Prinsip kerja Motor DC (Djati, 2010)

Agar proses perubahan energy mekanik dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar daripada tegangan gerak yang disebabkan reaksi lawan dengan memberikan arus pada kumparan jangkar yang dilindungi oleh medan maka menimbulkan perputaran pada motor.

Dalam memahami sebuah motor, penting mengerti apa yang dimaksud dengan beban motor. Beban dalam hal ini mengacu pada keluaran tenaga putar / torque sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan ke dalam tiga kelompok :

- a. Beban torque konstan adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun torquency tidak bervariasi, contoh beban dengan torque konstan ada konveyors, rotary kilns, dan pompa displacement konstan.
- b. Beban dengan variable torque adalah beban dimana torque yang bervariasi dengan kecepatan operasi. Contoh beban dengan variable torque adalah sentrifugal dan fan (torque bervariasi sebagai kuadrat kecepatan).
- c. Beban dengan energy konstan adalah beban dengan permintaan torque yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya beban adalah peralatan-peralatan mesin.

Untuk menentukan arah putaran motor digunakan kaedah Fleming tangan kiri. Kutub-kutub magnet akan menghasilkan medan magnet dengan arah dari kutub utara ke kutub selatan. Jika medan magnet memotong sebuah kawat penghantar yang dialiri arus searah dengan empat jari, maka akan timbul gerak searah ibu jari. Gaya ini disebut gaya Lorentz, yang besarnya sama dengan F .

Kecepatan putaran dapat diubah dengan cara mengubah besar arus pada kumparan armature. Akan tetapi, karena sumber tegangan tetap biasanya digunakan sebagai input penggunaan sebuah rangkaian elektronik. Rangkaian ini dapat mengontrol nilai rata-rata tegangan, dengan cara mengubah-ubah interval waktu untuk menghasilkan tegangan DC yang bervariasi.

Tabel 2.2 Detail Motor DC

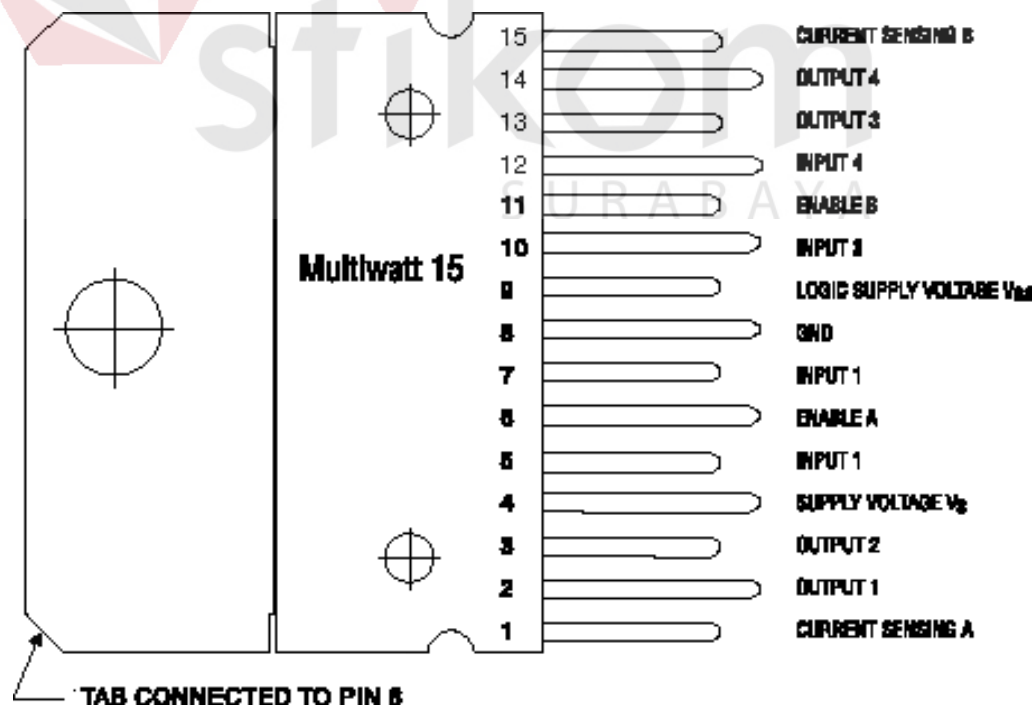
<i>Voltage</i>	24VDC
<i>Power (watts)</i>	24W
<i>Dimensi / ukuran</i>	Square – 120mm L x 120mm H x 38mm
<i>Pelistrikan</i>	2 wire leads
<i>RPM</i>	3898 RPM
<i>Weight</i>	269.89g
<i>Range tegangan</i>	14 ~ 27.6VDC

2.5 Driver motor L298

L298 memiliki dua buah rangkaian H-Bridge di dalamnya, sehingga dapat digunakan untuk men-drive dua buah motor DC. Masing-masing dapat mengantarkan arus hingga 2A. Namun, dalam penggunaannya, L298 dapat digunakan secara paralel, sehingga kemampuan menghantarkan arusnya menjadi 4A.

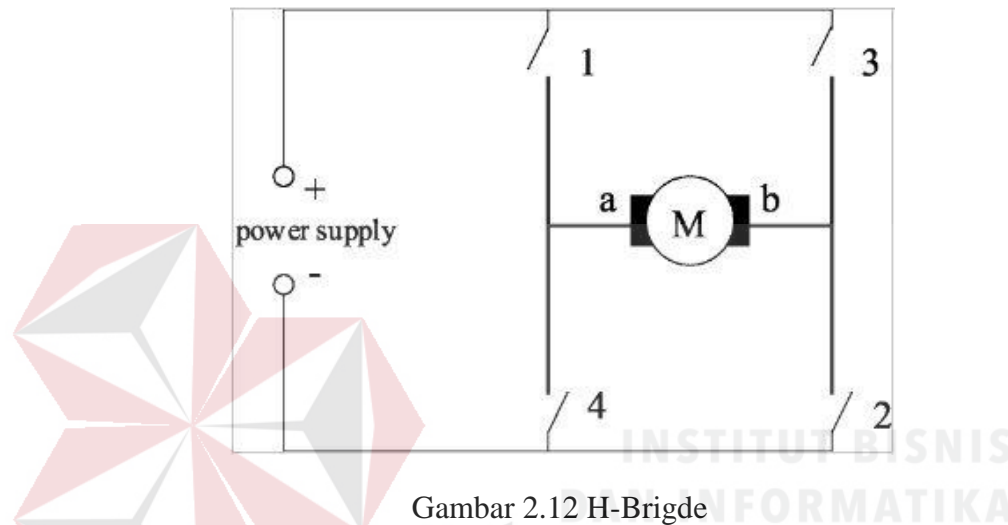
Spesifikasi L298:

1. Tegangan operasi hingga 46 V.
2. Total arus DC hingga 4 A.
3. Over temperature protection.
4. Low saturation volatage.
5. Package: 15-pin Multiwatt Vert.



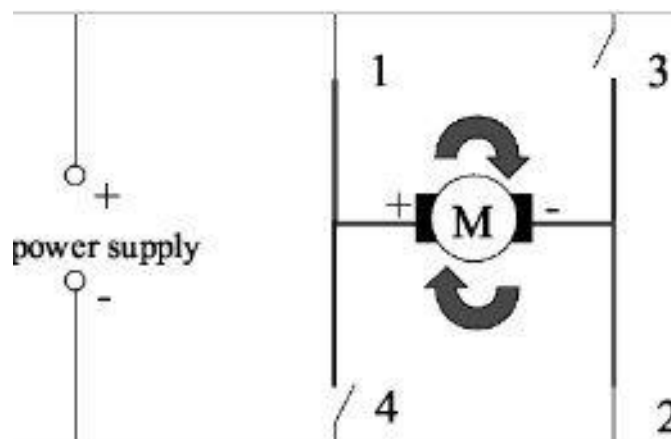
Gambar 2.11 Pin Connection L298 (STMicroelectronics, 2000)

H-Bridge atau yang diterjemahkan secara kasar sebagai “Jembatan H”, adalah sebuah rangkaian dimana motor menjadi titik tengahnya dengan dua jalur yang bisa dibuka tutup untuk melewati arus pada motor tersebut, persis seperti huruf “H” (dengan motor berada pada garis horizontal).



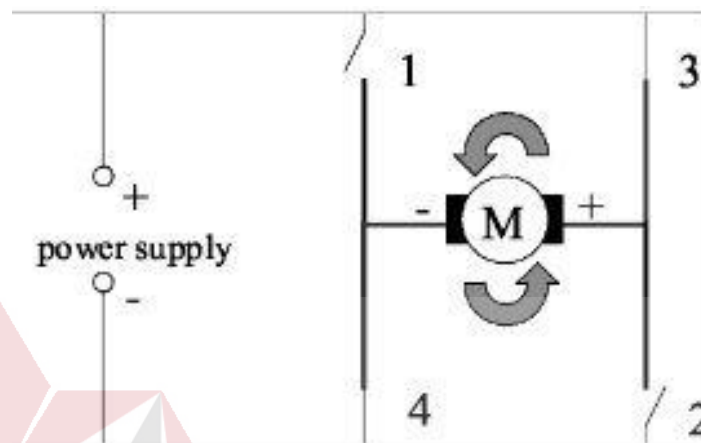
Gambar 2.12 H-Brigde

Dua terminal motor a dan b dikontrol oleh 4 saklar (1 s/d 4). Ketika saklar satu dan dua diaktifkan (saklar 3 dan 4 dalam keadaan off), maka terminal motor a akan mendapatkan tegangan (+) dan terminal b akan terhubung ke ground (-), hal ini menyebabkan motor bergerak maju (atau searah jarum jam)



Gambar 2.13 H-bridge mengatur putaran motor ke kanan

Sedangkan sebaliknya, bila saklar 1 dan 2 dalam keadaan off, saklar 3 dan 4 dalam keadaan aktif, maka terminal a akan terhubung ke ground (-) dan terminal b akan mendapatkan tegangan (+), dan tentunya hal ini dapat menyebabkan motor berubah arah putarnya, menjadi bergerak mundur (atau berlawanan dengan arah jarum jam)



Gambar 2.14 H-bridge mengatur putaran motor ke kiri

Untuk mengimplementasikan H-Bridge ini, tidak bisa langsung dihubungkan ke output yang diambil dari pin I/O mikrokontroler. Sebab output dari mikrokontroler hanya mempunyai daya yang sangat kecil. Sedangkan untuk motor sendiri, kadang-kadang membutuhkan daya yang tidak kecil (misalnya 200 mA, 1 A atau bahkan lebih). Jika kita memaksakan menghubungkan output digital dari mikrokontroler langsung ke motor, bisa jadi merusak mikrokontroler itu sendiri.

Untuk itu kita membutuhkan sebuah rangkaian penguat yang dapat dikontrol dari input digital. Dan tentunya chip (IC) yang mengakomodasi keperluan ini telah ada. *H-Bridge driver*. Salah satu tipenya kita bisa menggunakan **L293D** (untuk meng-handle arus rata-rata 600 mA) atau **LMD18200** (untuk meng-handle arus 3A, tegangan sampai 55V).

Gambar 2.15 Rangkaian L298 Sederhana

Motor Driver VN30SP30

Merupakan H-Bridge yang didesain untuk menghasilkan drive 2 arah dengan arus sampai dengan 30 A pada tegangan 5,5 Volt sampai 16 Volt. Berikut tabel kebenaran Driver VN30SP30.

Tabel 2.3 Tabel Kebenaran



Merupakan H-Bridge yang didesain untuk menghasilkan drive 2 arah dengan arus kontinyu sampai dengan 30 A pada tegangan 5,5 Volt sampai 16 Volt. Berikut tabel kebenaran motor driver VN30SP30.

Status kerja modul H-Bridge	Input dan Status					Output	
	MPWM	MIN1	MIN2	MEN1	MEN2	MOUT1	MOUT2
Forward	H	H	L	H	H	V MOT	MGND
Reverse	H	L	H	H	H	MGND	V MOT
Brake to GND	H	L	L	H	H	MGND	MGND
Brake to VCC	X	H	H	H	H	V MOT	V MOT
Free Running Stop	L	L	L	H	H	OPEN	OPEN
Free Running Stop	L	H	L	H	H	V MOT	OPEN
Free Running Stop	L	L	H	H	H	OPEN	V MOT
Fault di OUT1 dan OUT2	X	X	X	L	L	OPEN	OPEN
Fault di OUT1	H	X	H	L	H	OPEN	V MOT
Fault di OUT1	H	X	L	L	H	OPEN	MGND
Fault di OUT2	H	H	X	H	L	V MOT	OPEN
Fault di OUT2	H	L	X	H	L	MGND	OPEN

MS_30A_HBridge_manual.pdf)

2.7 Mikrokontroler ATmega 8535

Fitur-fitur utama yang dimiliki ATmega8535, sebagai berikut :

1. Saluran I/O sebanyak 32 buah yaitu port A, port B, port C, dan port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga unit timer/counter dengan kemampuan perbandingan.
4. CPU terdiri dari 32 buah register.
5. Watchdog timer dengan osilator internal.
6. SRAM sebesar 512 byte.
7. Memory flash sebesar 8kbytes dengan kemampuan read while write.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI.
10. EEPROM sebesar 512 byte yang dapat deprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial.

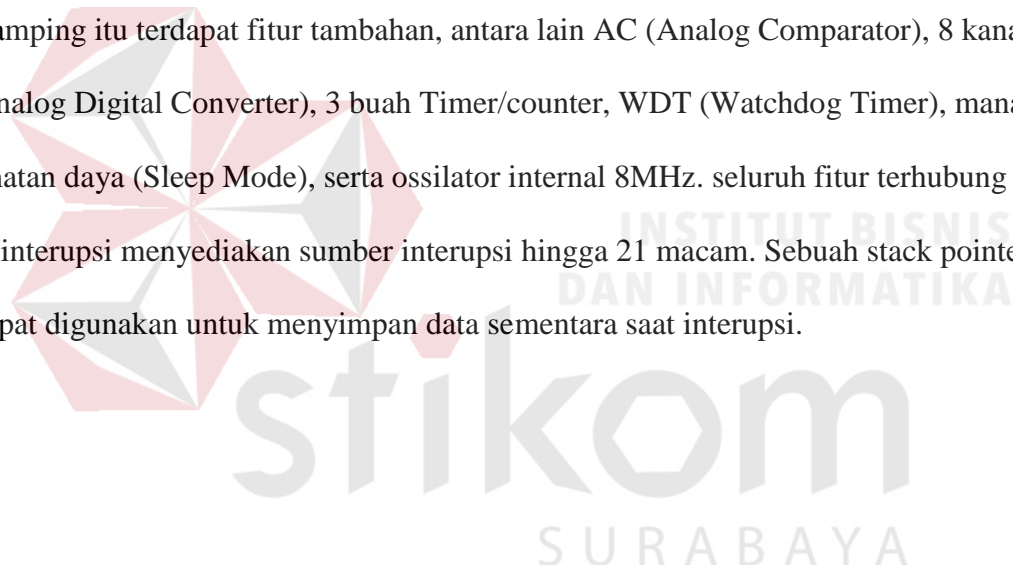
Mikrokontroler AVR ATmega8535 merupakan mikrokontroler produksi atmel dengan 8Kbyte in-system programmable-flash, 512 byte EEPROM, dan 512 bytes internal SRAM. AVR ATmega8535 memiliki seluruh fitur yang dimiliki AT90S8535. Selain itu, konfigurasi pin AVR ATmega8535 juga kompatibel dengan AT90S8535.

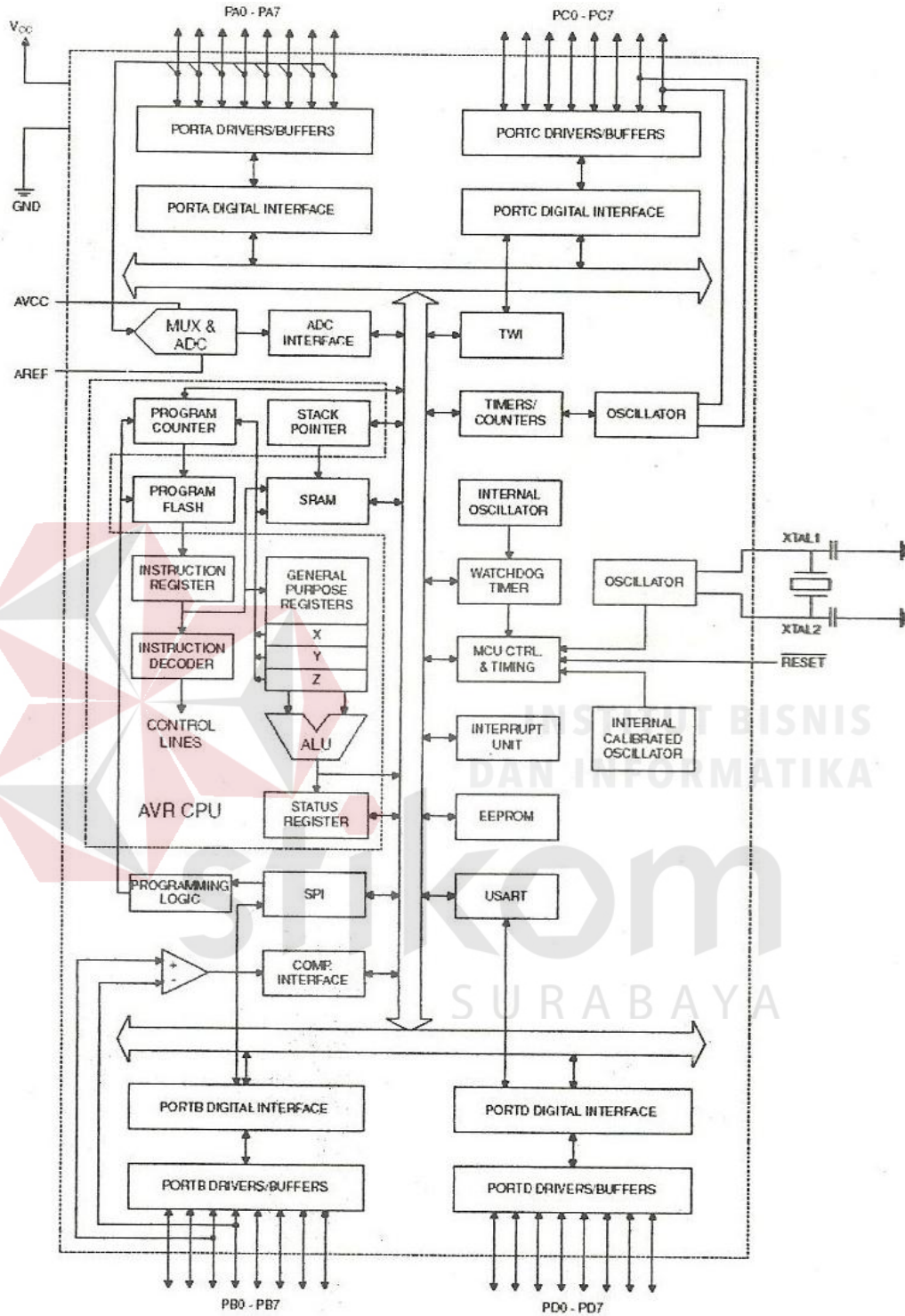
Diagram blok arsitektur ATmega8535 terdapat sebuah inti prosesor yaitu Central Processing Unit, dimana terjadi seluruh pengumpanan intruksi (fetching) dan komputasi data seluruh

register umum sebanyak 32 buah terhubung langsung dengan unit ALU (Arithmetic and Logic Unit). Terdapat empat buah port masing-masing delapan bit dengan diungsikan sebagai masukan maupun keluaran.

Media penyimpanan program berupa flash memory, sedangkan penyimpan data berupa SRAM (Static Random Access Memory) dan EEPROM (Electrical Erasable Programmable Read Only Memory). Untuk komunikasi data tersedia fasilitas SPI (Serial Peripheral Interface), USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter), serta TWI (Two-wire Serial Interface).

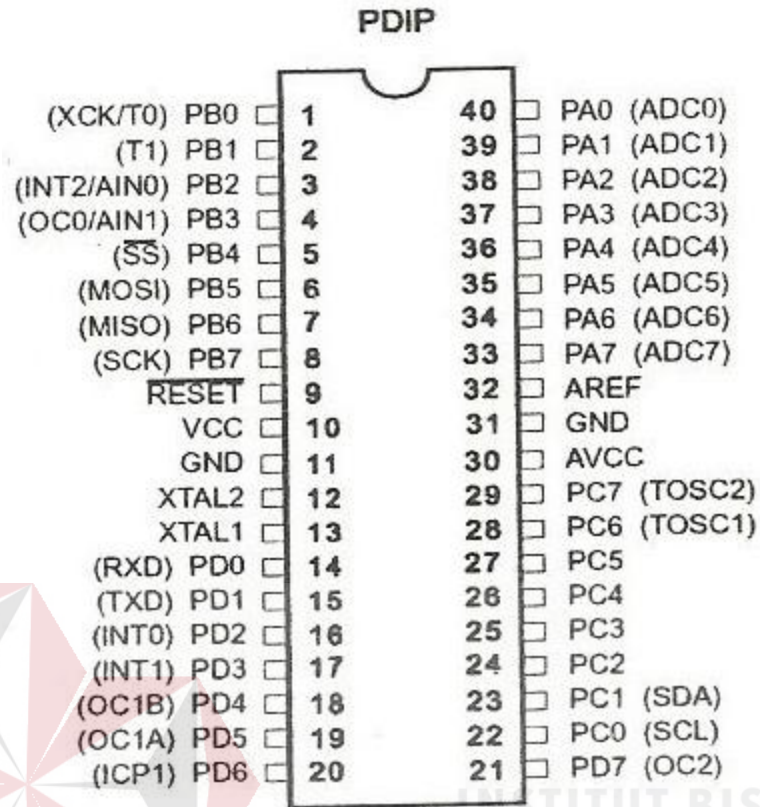
Disamping itu terdapat fitur tambahan, antara lain AC (Analog Comparator), 8 kanal 10-bit ADC (Analog Digital Converter), 3 buah Timer/counter, WDT (Watchdog Timer), manajemen penghematan daya (Sleep Mode), serta oscillator internal 8MHz. seluruh fitur terhubung ke bus 8 bit. Unit interupsi menyediakan sumber interupsi hingga 21 macam. Sebuah stack pointer selebar 16 bit dapat digunakan untuk menyimpan data sementara saat interupsi.





Sumber :Atmel Corporation. 2003.

Gambar 2.16 Arsitektur 8535



Sumber :Atmel Corporation. 2003.

Gambar 2.17 Konfigurasi pin

Konfigurasi pin ATmega8535 dengan kemasan 40 pin DIP (Dual in-line Package). Dari gambar diatas dapat dijelaskan fungsi dari masing-masing pin ATmega8535 sebagai berikut :

1. VCC merupakan pin yang berfungsi sebagai masukan catu daya.
2. GND merupakan pin Ground.
3. Port A (PA0...PA7) merupakan pin input/output dua arah dan pin masukan analog ke ADC (ADC0 s.d. ADC7).
4. Port B (PB0...PB7) merupakan pin input/output dua arah dan pin fungsi khusus.

Tabel 2.4 Fungsi Khusus Port B

Pin	Fungsi Khusus
PB0	T0 (Timer / Counter0 External Counter input)
PB1	T1 (Timer / Counter1 External Counter input)
PB2	AIN0 (Analog Comparator Positive Input)
PB3	AIN1 (Analog Comparator Negative Input)
PB4	SS (SPI Slave Select Input)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

5. Port C (PC0...PC7) merupakan pin input/output dan 2 Pin yaitu PC6 dan PC7 berfungsi sebagai oscillator luar untuk Timer/Counter2.
6. Port D (PD0...PD7) merupakan pin input/output dua arah dan Fungsi khusus.

Tabel 2.5 Fungsi Khusus Port D

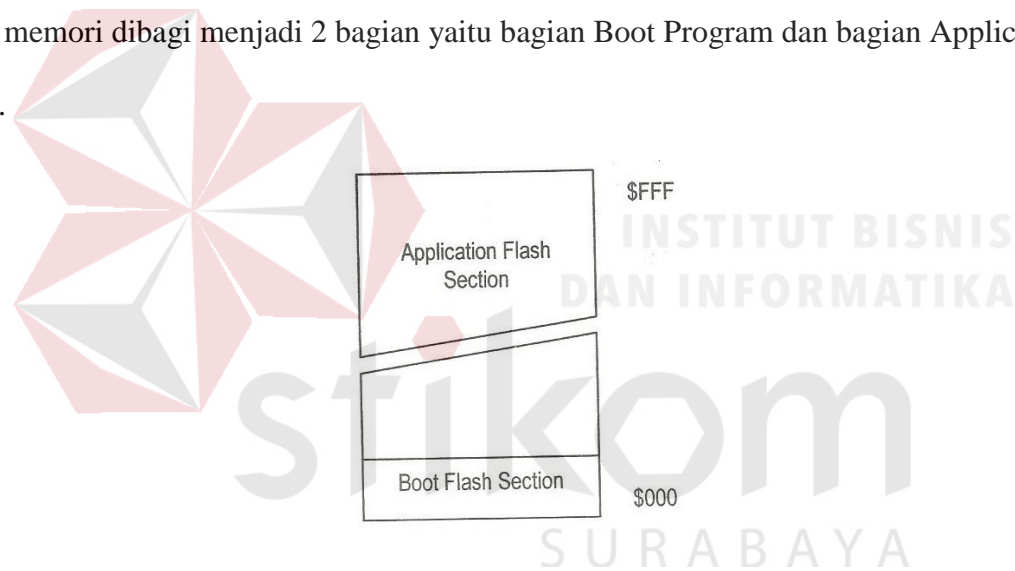
Pin	Fungsi Khusus
PD0	RXD (UART Input Line)
PD1	TXD (UART Output Line)
PD2	INT0 (External Interrupt 0 Input)
PD3	INT1 (External Interrupt 1 Input)
PD4	OC1B (Timer/Counter1 Output CompareB Match Output)
PD5	OC1A (Timer/Counter1 Output CompareA Match Output)
PD6	ICP (Timer/Counter1 Input Capture Pin)
PD7	OC2 (Timer/Counter2 Output Compare Match Output)

7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan clock external.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.

2.6.1 Peta Memori

Arsitektur AVR terdiri atas dua memori utama, yaitu Data memori dan Program memori. Sebagai fitur dari ATmega8535, terdapat EEPROM 512 byte sebagai memori data dan dapat diprogram saat operasi.

ATmega8535 terdiri atas 8 byte On-Chip Reprogrammable Flash Memory untuk penyimpanan program. Karena seluruh intruksi AVR dalam bentuk 16 bit atau 32 bit, maka flash dirancang dengan komposisi 4KX16. Untuk mendukung keamanan software atau program, flash program memori dibagi menjadi 2 bagian yaitu bagian Boot Program dan bagian Application Program.



Sumber :Atmel Corporation. 2003.

Gambar 2.18 Peta memori program

Memori data terbagi menjadi 3 bagian yaitu, 32 buah register umum, 64 buah register I/O, dan 512 byte SRAM internal.

BAB III

METODE PENELITIAN

Perancangan perangkat keras dilakukan dengan metode penelitian yang didasarkan pada studi kepustakaan berupa data-data literature dari masing-masing komponen, informasi dari internet, dan konsep-konsep teoritis dari buku-buku penunjang. Setelah literature terkumpul barulah dilakukan perancangan perangkat keras yang dilanjutkan dengan perancangan dengan pembuatan perangkat lunak yaitu program sederhana untuk melakukan pengujian pada setiap bagian pada perangkat keras yang telah dibuat dan program untuk menjalankan perangkat keras secara utuh. Perancangan perangkat keras dan lunak dilakukan dengan metode penelitian yang didasarkan pada studi kepustakaan berupa data-data literatur dari masing-masing komponen, informasi dari internet, dan konsep-konsep teoritis dari buku-buku penunjang.

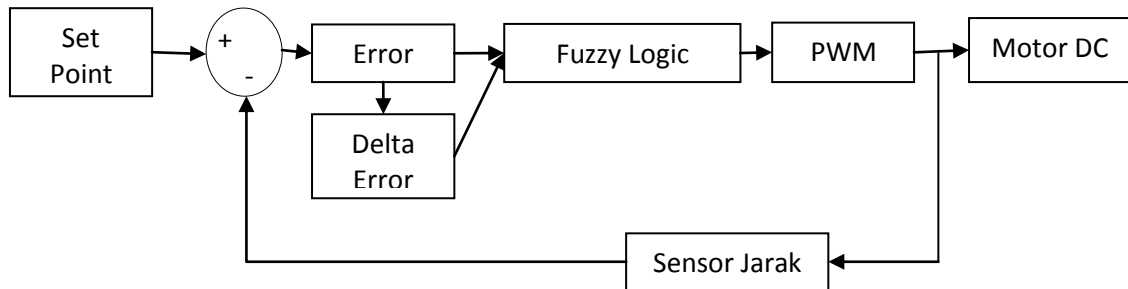
Dari data-data yang diperoleh maka dilakukan perencanaan rangkaian perangkat keras dan lunak. Dalam perangkat keras ini, penulis akan melakukan pengujian perangkat keras dengan program-program yang telah dirancang kemudian melakukan kolaborasi perangkat keras dengan perangkat lunak yang telah selesai dirancang.

Pada bab ini akan dibahas mengenai masalah yang timbul dalam perencanaan dan pembuatan perangkat keras (hardware) maupun perangkat lunak (software). Dari dua bagian tersebut akan dipadukan agar dapat bekerja sama untuk menjalankan sistem dengan baik.

Perencanaan ini perlu dilakukan sebelum proses pembuatan sistem tersebut, perencanaan ini berguna agar pengerjaan tahapan berikutnya berjalan dengan lancar. Tahapan-tahapannya meliputi tahap pembuatan perangkat keras, perangkat lunak dan menghubungkan keduanya.

3.1 Perencanaan Perangkat Keras

Perancangan perangkat keras pada sistem dilakukan berdasarkan blok diagram yang terdapat pada gambar 3.1.



Gambar 3.1 Blok Diagram Keseluruhan Sistem

Dalam sistem diatas mikrokontroler sebagai pemroses akan mendapatkan data input dari suatu modul ultrasound. Modul ultrasound akan memberikan data tentang ketinggian banyaknya padi didepan modul ultrasound, kemudian data tersebut diolah. Dalam hal ini proses fuzzifikasi, penentuan rule set, dan defuzzifikasi sedang berjalan. Hasil proses tersebut adalah mengatur kecepatan motor untuk menggerakkan penggiling padi.

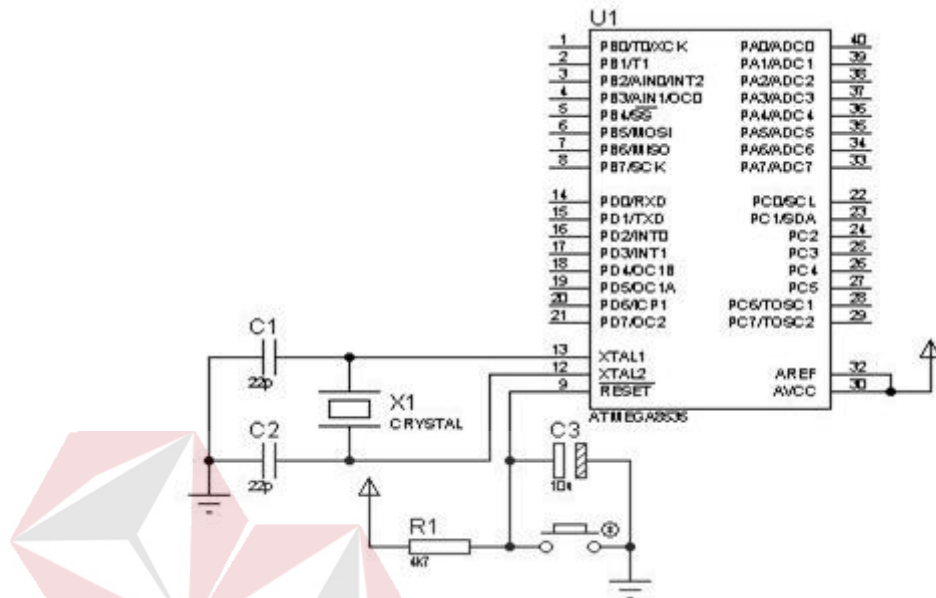
3.1.1 Rangkaian Mikrokontroler

Pada proyek tugas akhir ini dibuat piranti pengendali menggunakan mikrokontroler keluaran AVR, yaitu ATmega8535. Untuk menjalankan mikrokontroler ini diperlukan rangkaian minimum sistem. Rangkaian minimum sistem tersebut terdiri dari rangkaian reset, dan rangkaian oscillator.

A. Rangkaian Minimum Sistem

Untuk menjalankan mikrokontroler dibutuhkan sebuah rangkaian agar mikrokontroler tersebut dapat bekerja dengan baik. Rangkaian mikrokontroler terdiri dari rangkaian reset dan oscillator.

Pada pin VCC diberikan masukan tegangan operasi berkisar antara 4,5 Volt sampai dengan 5,5 Volt. Pin RST mendapat input dari manual reset. Rangkaian minimum sistem dapat dilihat pada gambar 3.2.



Gambar 3.2 Rangkaian Minimum Sistem ATmega8535

Berdasarkan gambar 3.2 pin VCC diberi tegangan operasi yang besarnya maksimal 5V jika tidak menggunakan IC regulator, 35V jika menggunakan IC regulator 7805. Port A digunakan sebagai LCD, Port B digunakan untuk sebagai inputan sensor ultrasound. Port D digunakan sebagai port yang terhubung dengan motor driver L298 dan VNH3SP30. Dalam gambar 3.2 di atas, pin XTAL1 dan XTAL2 dihubungkan dengan komponen XTAL sebesar 11.000000 MHz. pemilihan frekuensi osilasi dari XTAL tersebut berdasarkan penggunaan mikrokontroler agar setiap clock mikrokontroler berlangsung setiap 1 μ s. berikut adalah perhitungan besar clock cycle yang dipakai :

$$1 \text{ siklus mesin} = 12 \text{ clock}$$

$$T(s) = 1 / F$$

$$= \frac{1}{12 \times \text{Frekuensi Kristal}}$$

$$\begin{aligned}
&= \frac{1}{\frac{1}{12} \times 11 \text{ MHz}} \\
&= 1,09 \times 10^{-6} \\
&= 1 \mu\text{s}
\end{aligned}$$

Untuk melakukan proses downloading program dari komputer ke dalam memori program internal mikrokontroler, digunakan kabel downloader dengan interface DB25 yang dihubungkan pada port LPT1 pada komputer dengan konfigurasi seperti pada gambar 3.3. pin 5 dihubungkan dengan resistor sebesar 4,7 k dan tegangan sebesar 5 volt secara seri. Pin 25 dihubungkan dengan ground. Pin 6, 7, 8, 9, dan 10 berturut-turut dihubungkan dengan pin SCK, MOSI, RESET, dan MISO pada mikrokontroler. Sedangkan pin 2 dan 12 saling terhubung.

Program bantu yang terintegrasi untuk menulis sekaligus debug aplikasi mikrokontroler AVR adalah CodeVision AVR Version 2.05.3 Standart yang support dengan windows 9x/Me/NT/2000/XP/7. CodeVision C Compiler (CVAVR) merupakan compiler dengan bahasa C untuk AVR. Compiler ini cukup memadai untuk belajar AVR, karena selain mudah penggunaannya juga didukung berbagai fitur yang sangat membantu dalam pembuatan software untuk keperluan pemrograman AVR. Tampilan AVR dapat dilihat pada.

a. Setting software CodeVision AVR

Sebelum menggunakan software CodeVision AVR sebagai downloader, pertama-tama harus melakukan penyetelan pada software ini. Berikut adalah langkah-langkahnya:

1. Pada tampilan awal software terdapat menu bar pada bagian atas.
2. Klik file > new, selanjutnya akan muncul dialog create new file, pilih project > OK.

Dialog create new file dapat dilihat pada gambar 3.4, lalu pilih “yes” or “no” karena untuk memastikan bahwa project benar-benar akan digunakan dalam tugas akhir ini.

3. Kemudian tampak dialog CodeWizardAVR, pilih AVR Chip Type AT90, ATtiny, ATmega, FPSLIC. Setelah itu pilih “Ok” untuk melanjutkan “Cancel” untuk membatalkan Dialog CodeWizardAVR dapat dilihat pada.
4. Kemudian tampak dialog CodeWizardAVR – untitled.cwp. Ubah bagian tab Chip, pilih seri mikrokontroler yang sesuai dengan yang digunakan, ATmega8535. Pilih nilai clock sesuai dengan komponen kristal external yang digunakan 11.000000 Mhz.
5. Setting Port D4 dan D5 untuk keluaran output PWM melalui OC1A dan OC1B.
6. Setting Timer/Counter1 yang akan digunakan untuk PWM Motor DC. Pilih Clock Value 10.742 KHz dengan mode Fast PWM top=0x00FF, Output A dan B Non-inv.
7. Setelah itu untuk settingan terakhir yaitu setting LCD, klik enable Alphanumeric LCD support. lalu pilih character LCD 16, setelah itu pilih port A untuk keluaran LCD.
8. Setelah semuanya telah disetting pilih menu bar klik program > generate, save and quit. Lalu simpan ketiga-tiganya dengan nama yang sama.
9. Selanjutnya tampak kode program pada software CodeVision AVR. Berikut potongan program inisialisasi port yang akan digunakan sebagai input dan output atau inisialisasi lainnya,

```

/*****
This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

```

```

Project :
Version :
Date    : 6/24/2014
Author  : tyery08
Company : embeeminded.blogspot.com
Comments:

```

```

Chip type           : ATmega8535
Program type        : Application

```

```

AVR Core Clock frequency: 11.000000 MHz
Memory model           : Small
External RAM size      : 0
Data Stack size       : 128
*****/

#include <mega8535.h>

// Alphanumeric LCD functions
#include <alcd.h>

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=Out Func4=Out Func3=In Func2=In Func1=In
// Func0=In
// State7=T State6=T State5=0 State4=0 State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x30;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 10.742 kHz
// Mode: Fast PWM top=0x00FF
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA1;
TCCR1B=0x0D;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;

```

```

OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

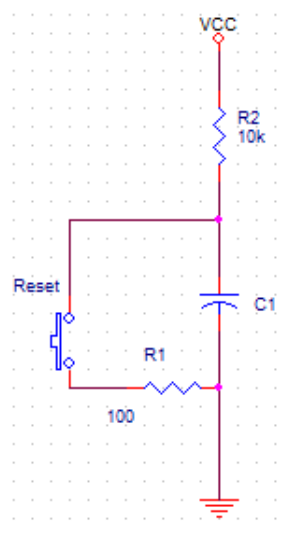
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTA Bit 0
// RD - PORTA Bit 1
// EN - PORTA Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 16
lcd_init(16);

while (1)
{
    // Place your code here
}

```

B. Rangkaian Reset

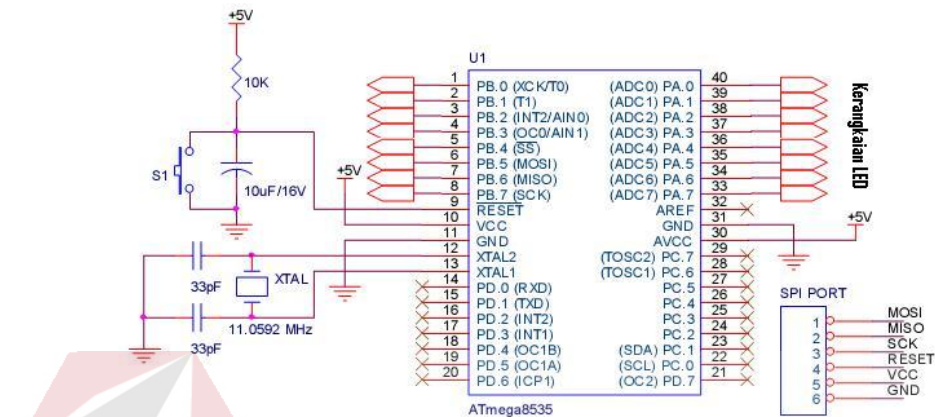
Reset pada mikrokontroler ATmega8535 terjadi dengan adanya logika high “1” selama dua cycle pada kaki RST mikrokontroler. Setelah kondisi pin kembali low, maka mikrokontroler ATmega8535 akan menjalankan program dari alamat 0000H. dalam hal ini reset digunakan adalah manual reset. Rangkaian reset dapat dilihat pada gambar 3.14.



Gambar 3.3 Rangkaian Reset

C. Rangkaian Oscillator

Pin XTAL1 dan XTAL2 merupakan pin oscillator bagi mikrokontroler ATmega8535. Pin XTAL1 berfungsi sebagai input dan XTAL2 sebagai output oscillator. Oscillator ini bisa berasal dari kristal. Rangkaian oscillator dapat dilihat pada gambar 3.15.



Gambar 3.4 Rangkaian Osillator

D. Interface I/O

Rangkaian *I/O* dari mikrokontroler mempunyai kontrol direksi yang tiap bitnya dapat dikonfigurasi secara individual, maka dalam perancangan *I/O* yang digunakan ada yang berupa operasi port ada pula yang dikonfigurasi tiap bit *I/O*. berikut ini akan diberikan konfigurasi dari *I/O* mikrokontroler tiap bit yang ada pada masing-masing port yang terdapat pada mikrokontroler.

1. Port A

Port A digunakan untuk LCD.

2. Port B

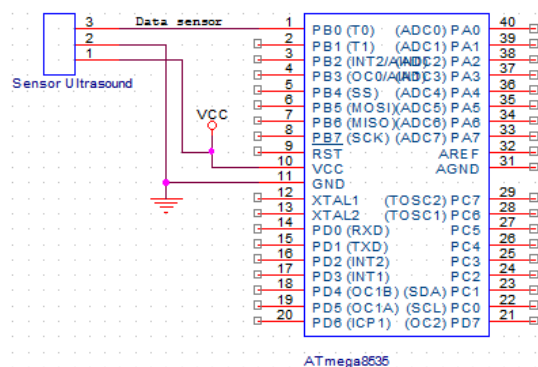
Port B digunakan untuk input data sensor *ultrasound*.

3. Port D

Port D digunakan untuk sensor motor DC.

3.1.2 Rangkaian sensor Parallax PING)))™

Sensor Parallax PING)))™ digunakan untuk membaca jarak, Parallax PING)))™ merupakan sensor jarak berbasis gelombang suara, yang akan membaca jarak secara terus menerus dan memberikan output berupa tegangan *digital*. Pada modul Parallax PING)))™ terdapat 3 pin yang digunakan untuk jalur power supply (+5V), ground dan signal. Pin signal dapat langsung dihubungkan dengan mikrokontroler port B.0 tanpa tambahan komponen apapun. Parallax PING)))™ mendeteksi objek dengan cara mengirimkan suara ultrasonik kemudian menerima pantulan suara tersebut. Parallax PING)))™ hanya akan mengirimkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler (pulsa high selama 10µs). suara ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama 200µs. suara ini akan merambat di udara dengan kecepatan 344.42m/detik (atau 1cm tiap 29.034µs). mengenai objek untuk kemudian terpantul kembali ke Parallax PING)))™. Selama menunggu pantulan Parallax PING)))™ akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (low) ketika suara pantulan terdeteksi oleh Parallax PING)))™. Oleh karena itulah lebar pulsa tersebut dapat mempresentasikan jarak antara Parallax PING)))™ dengan objek. Selanjutnya mikrokontroler cukup mengukur lebar pulsa tersebut dengan mengkonversinya dalam bentuk jarak. Pin 1 pada sensor ultrasound berfungsi untuk VCC (+5V), pin 2 berfungsi sebagai GND, dan pin 3 berfungsi untuk data sensor.

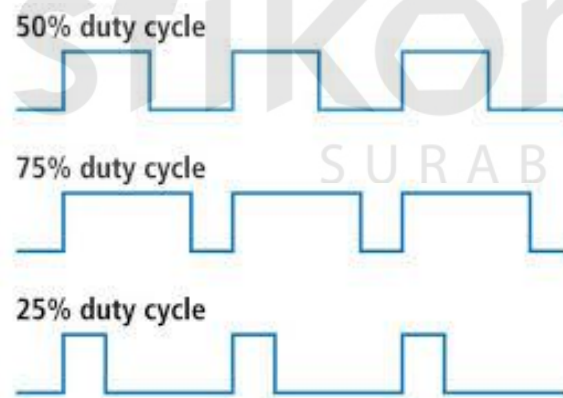


Gambar 3.5 Rangkaian sensor *ultrasound*

3.1.3 Konfigurasi Motor 21V dengan IC L298

Motor DC memiliki dua kabel terhubung. Kabel pertama untuk ground, kabel kedua untuk power supply dengan besar tegangan sampai 21 Volt. Kecepatan putar motor dapat dikendalikan dengan mengatur tegangan yang di masukkan, atau dapat juga dengan menggunakan teknik PWM (*Pulse Width Modulation*).

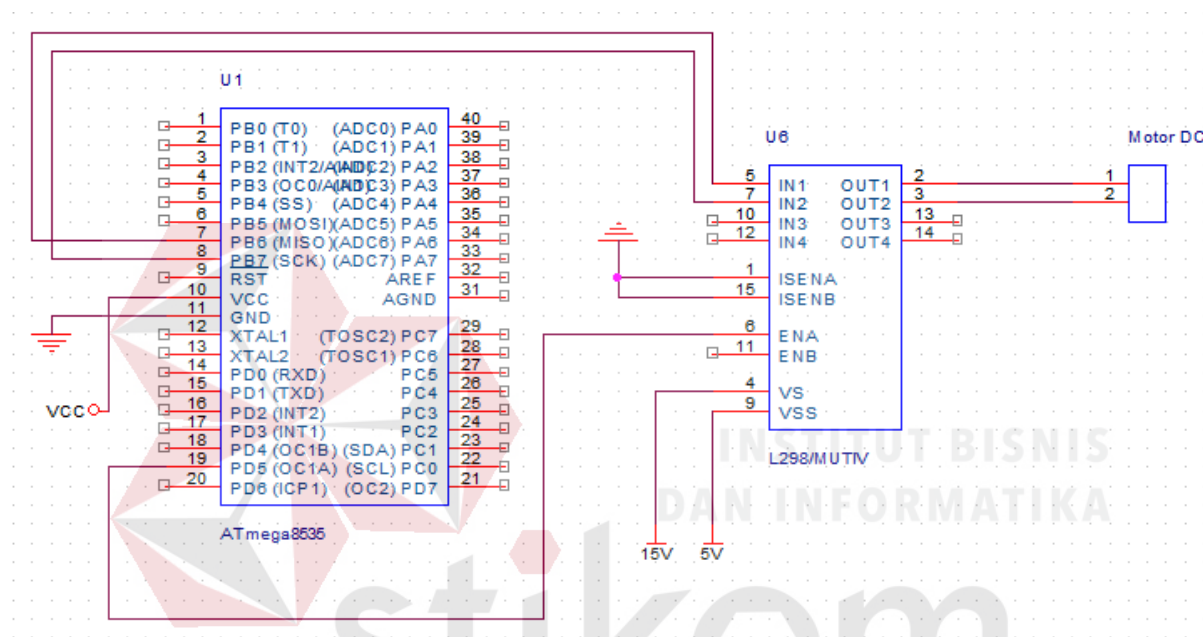
Dengan menggunakan PWM kita dapat mengatur kecepatan yang diinginkan dengan mudah, teknik PWM untuk pengaturan kecepatan motor adalah dengan cara merubah-ubah duty cycle pulsa. Pulsa yang berubah-ubah duty cycle-nya inilah yang menentukan kecepatan motor. Penulis menggunakan mode fast PWM, dalam mode fast PWM sifat cacahan naik (Counting-up) hingga mencapai top (nilai maksimal yang ditentukan sesuai resolusi, misal resolusi 10 bit maka nilai top = 0x01ff) kemudian mulai dari bottom lagi dan begitu seterusnya atau dinamakan single slope (Ardi Winoto, 2008). Berikut adalah contoh gambar pulsa dengan duty cycle 25%, 50%, dan 75%.



Gambar 3. 6 Duty cycle

Besarnya amplitudo dan frekuensi pulsa adalah tetap, sedangkan besarnya duty cycle berubah-ubah sesuai dengan yang diinginkan. Semakin besar duty cycle maka semakin cepat pula kecepatan motor. Sebagai contoh bentuk pulsa yang dikirimkan adalah 50%. Semakin besar duty cycle pulsa kotak, maka semakin lama pula posisi logika high. Jika motor diatur

agar berjalan ketika diberi logika high, maka jika member pulsa dengan duty cycle 50% maka motor akan berada dalam kondisi “nyala-mati-nyala-mati” sesuai dengan bentuk pulsa tersebut. Semakin lama motor berada pada kondisi “nyala” maka semakin cepat pula kecepatan motor tersebut. Motor akan berputar dengan kecepatan maksimum jika mendapat pulsa dengan duty cycle 100%. Berikut adalah gambar rangkaian motor driver dengan IC L298.



Gambar 3.7 Rangkaian Motor Driver dengan IC L298

3.1.4 Konfigurasi Motor 24V dengan IC VNH30SP30

Motor Driver adalah rangkaian yang digunakan sebagai switching sehingga nantinya motor dapat berputar searah jarum jam (*Clockwise*) dan berlawanan dengan jarum jam (*Counterclockwise*). Dan motor driver yang dipakai penulis adalah *Embedded Module Series* (EMS) 30A H-Bridge merupakan driver H-Bridge yang didesign untuk menghasilkan drive 1 arah maupun 2 arah dengan arus kontinyu sampai dengan 30A tegangan maksimal 36 Volt.

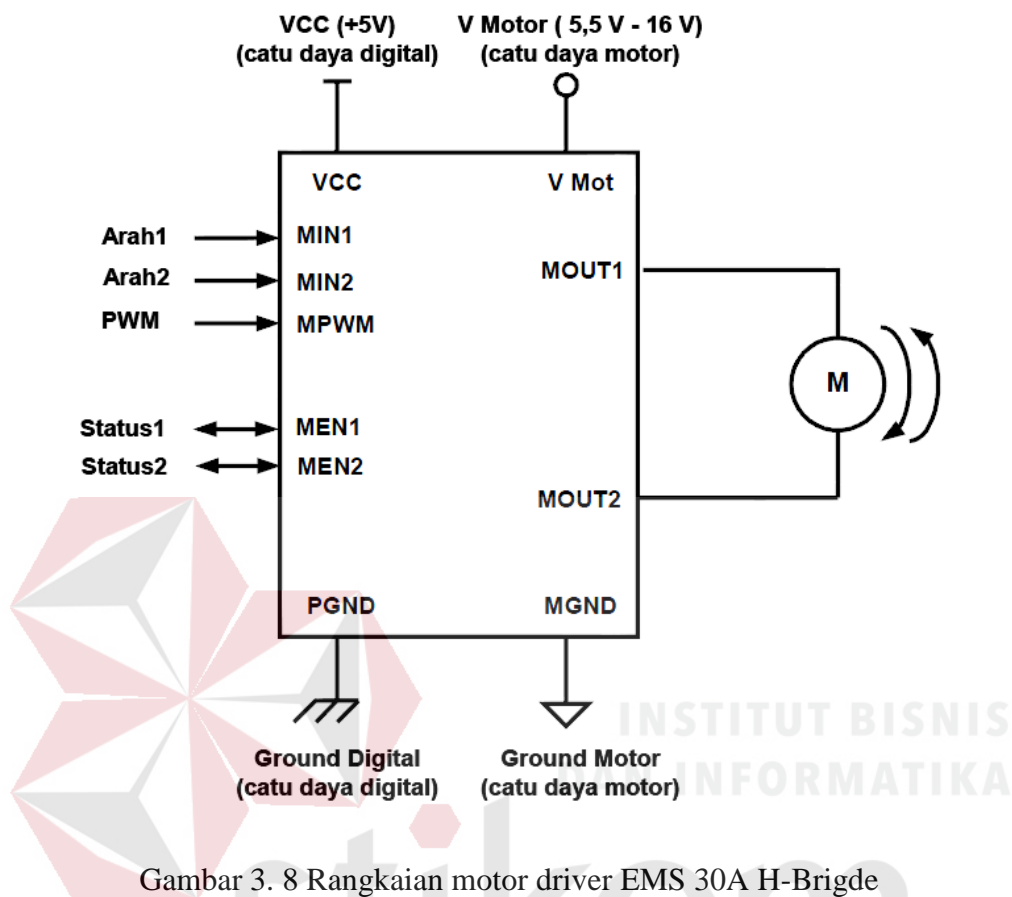
Modul ini mampu men-drive beban-beban induksi seperti relay, solenoid, motor DC, motor stapper, dan berbagai macam beban lainnya. Berikut deskripsi dari masing masing pin pada *Interface header* 1 dapat dilihat pada table 3.1.

Tabel 3.1 *Interface header* 1

Alokasi Pin Interface Header J1			
Pin	Nama	I/O	Fungsi
1	MIN1	I	Pin input untuk menentukan output MOUT1
2	MIN2	I	Pin input untuk menentukan output MOUT2
3	MEN1	I/O	Pin enable untuk output MOUT1 Diberi logika High untuk mengaktifkan <i>half</i> H-Bridge 1, diberi logika Low secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 1 Jika terjadi kondisi <i>Fault</i> (<i>thermal shutdown, undervoltage, overvoltage, dsb.</i>), maka pin ini akan ditarik Low secara internal oleh modul H-Bridge untuk melaporkan adanya kondisi <i>Fault</i>
4	MEN2	I/O	Pin enable untuk output MOUT2 Diberi logika High untuk mengaktifkan <i>half</i> H-Bridge 2, diberi logika Low secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 2 Jika terjadi kondisi <i>Fault</i> (<i>thermal shutdown, undervoltage, overvoltage, dsb.</i>), maka pin ini akan ditarik Low secara internal oleh modul H-Bridge untuk melaporkan adanya kondisi <i>Fault</i>
5	MCS	O	Output tegangan analog yang berbanding lurus dengan arus beban (<i>Range</i> output 0 - 5 V)
6	MPWM	I	Pin input untuk mengatur kerja modul H-Bridge secara PWM
7,9	VCC	-	Terhubung ke catu daya untuk input (5 Volt)
8,10	PGND	-	Titik referensi untuk catu daya input

(Sumber: http://www.innovativeelectronics.com/innovative_electronics/download_files/manual/EMS_30A_HBridge_manual.pdf)

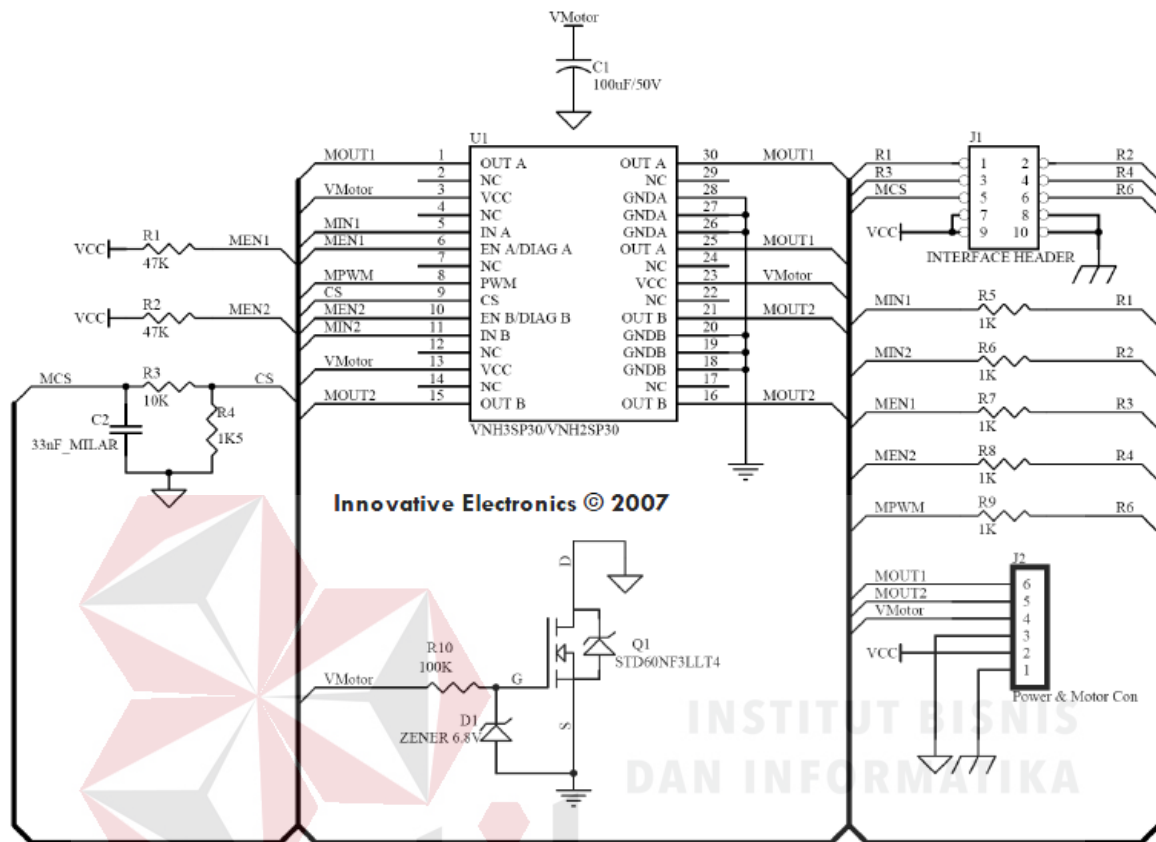
Sebuah modul H-Bridge 30A dapat digunakan untuk mengatur kerja 1 buah motor DC secara 2 arah, koneksinya dapat dilihat pada Gambar 3.18.



Gambar 3. 8 Rangkaian motor driver EMS 30A H-Bridge

(Sumber: http://www.innovativeelectronics.com/innovative_electronics/download_files/manual/EMS_30A_HBridge_manual.pdf)

Dan untuk gambar skema rangkaian motor driver (EMS) 30A H-Bridge dapat dilihat pada gambar 3.19.

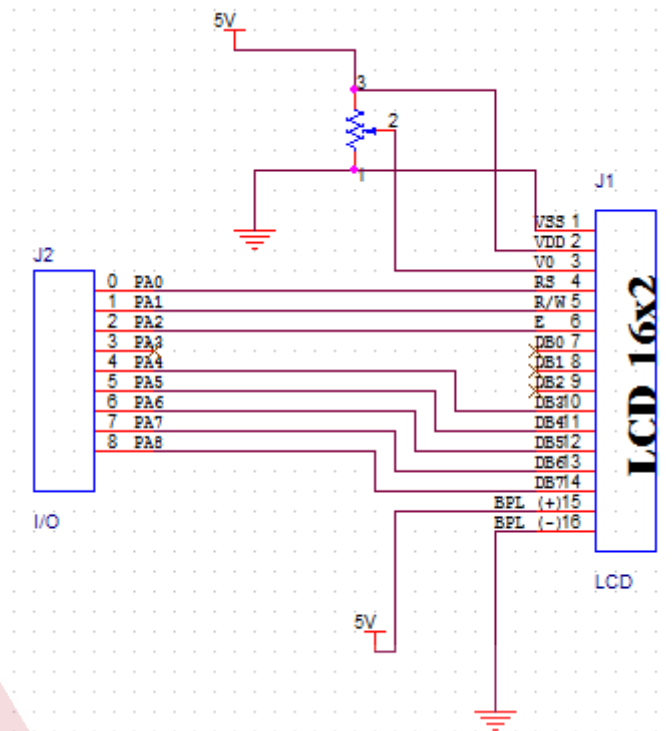


Gambar 3. 9 Skema EMS 30A H-bridge

(Sumber: http://www.innovativeelectronics.com/innovative_electronics/download_files/manual/EMS_30A_HBridge_manual.pdf)

3.1.5 Rangkaian LCD (*Liquid Crystal Display*) 16x2

Dengan sistem ini menggunakan LCD tipe 16x2 dengan lebar display 2 baris 16 kolom dan mempunyai 16 pin konektor. Dengan menggunakan CodeWizardAVR dapat dengan mudah mendefinisikan port yang terhubung dengan LCD. Untuk mendefinisikan port yang terhubung ke LCD dapat dilihat pada gambar 3.20.



Gambar 3.10 Konfigurasi PORTA ke LCD

Berikut deskripsi dari masing-masing pin LCD :

Tabel 3.3 deskripsi pin LCD

Pin	Name	Descriptions
1	VSS	0V Power supply, Ground
2	VDD	Positive Power Supply
3	V0	LCD Contrast Reference Supply
4	RS	Register Select
5	R/W	Read/Wrire Control Bus
6	E	Data Enable
7	DB0	Bi-Directional Tri State data Bus
:	:	
14	DB7	
15	BLA	LED Backlight Positive
16	BLK	LED Backlight Negative

Untuk jalur data dan alamat LCD dihubungkan ke microcontroller dengan lebar jalur data sebesar 8 bit. Pada LCD juga terdapat pin pengatur contrast dengan menggunakan variable resistor 10k. variable resistor dapat mengatur atau mengubah besarnya tegangan untuk mengatur contrast. Untuk pemasangan power LCD dan backlight tidak boleh terbalik

karena dapat mengakibatkan LCD menjadi rusak. Pada software CodeWizardAVR sudah ada program inisialisasi untuk mengakses LCD sehingga programmer dapat langsung memprogram sesuai dengan perintah-perintah untuk mengakses LCD, seperti pada potongan program dibawah ini :

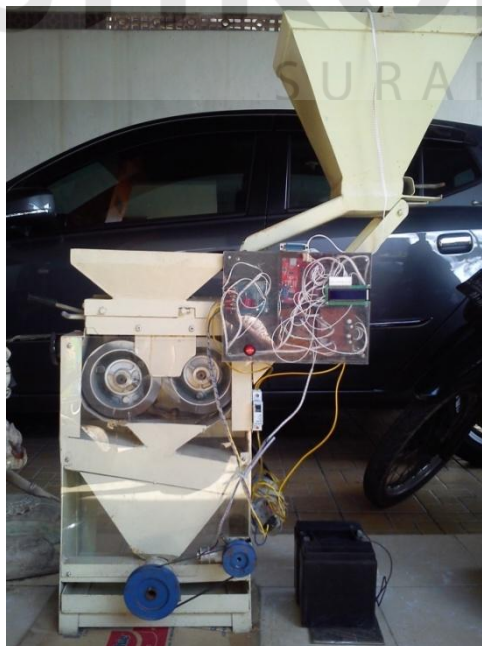
```
lcd_init(16);

/*****Tulisan yang akan tampil pada LCD*****/
lcd_gotoxy(0,0);
lcd_puts(" W E L C O M E ");
lcd_gotoxy(1,0);
lcd_puts("by : Stefanus A.T");

while (1)
{
    // Place your code here
}
}
```

3.2 Peletakan Komponen Elektronika

Rangkaian elektronika dari rangkaian *microcontroller*, *driver* motor terletak pada bagian depan, sedangkan sensor *ultrasound* terletak di bagian atas penampung padi. Sedangkan Motor DC berada bagian bawah. Berikut adalah gambar keseluruhan :



Gambar 3.11 Tampilan keseluruhan

Sensor ultrasound terletak pada bagian atas penampung padi.



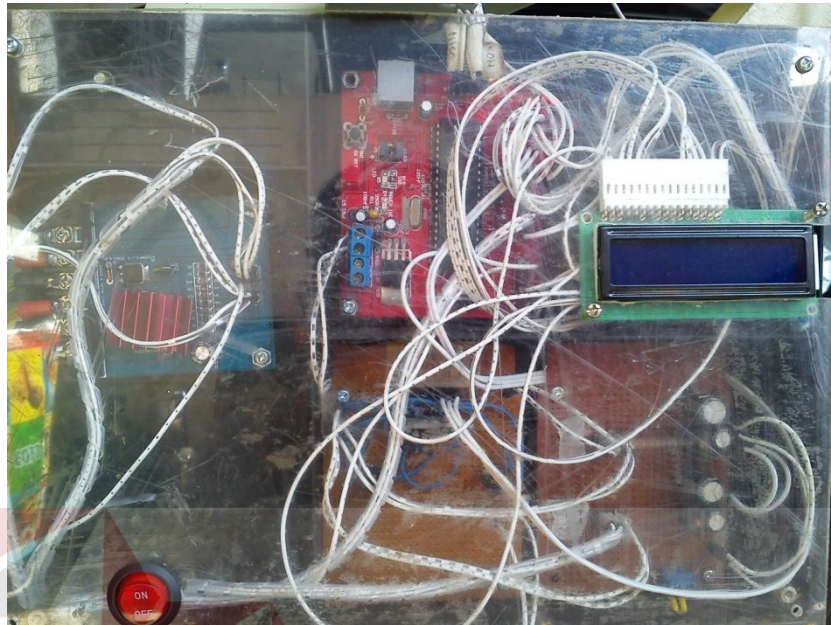
Gambar 3.12 Penempatan Sensor *Ultrasound*

Sensor ultrasound diletakkan pada bagian atas penampung padi dengan posisi menghadap kebawah. Hal ini bertujuan untuk memantau ketinggian padi.



Gambar 3.13 Penempatan Motor DC

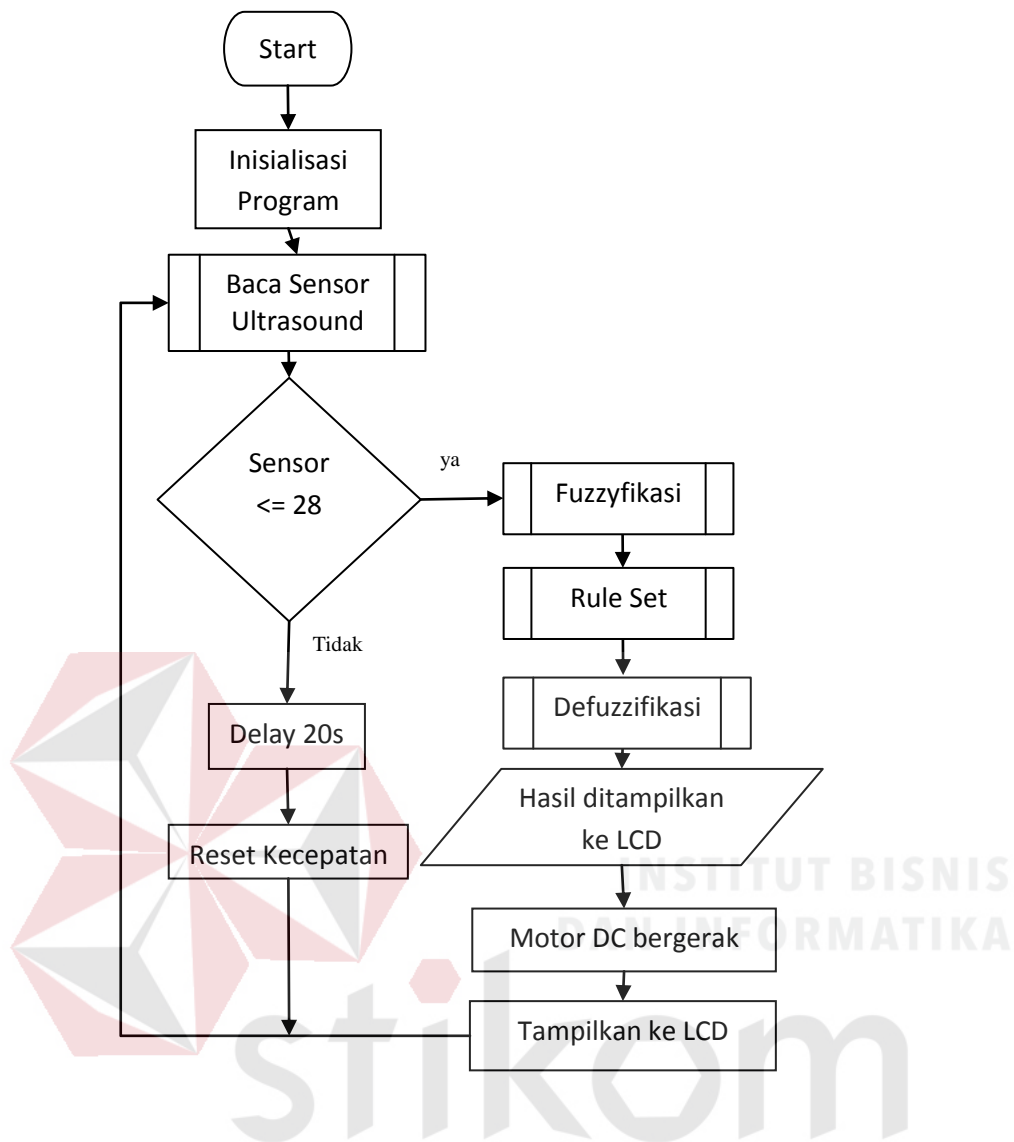
Motor DC berada di bagian bawah dan menggunakan *fan belt* untuk menyambungkan antara motor DC dengan penggiling padi.



Gambar 3.14 Penempatan *minimum system*

3.3 Perancangan Perangkat Lunak

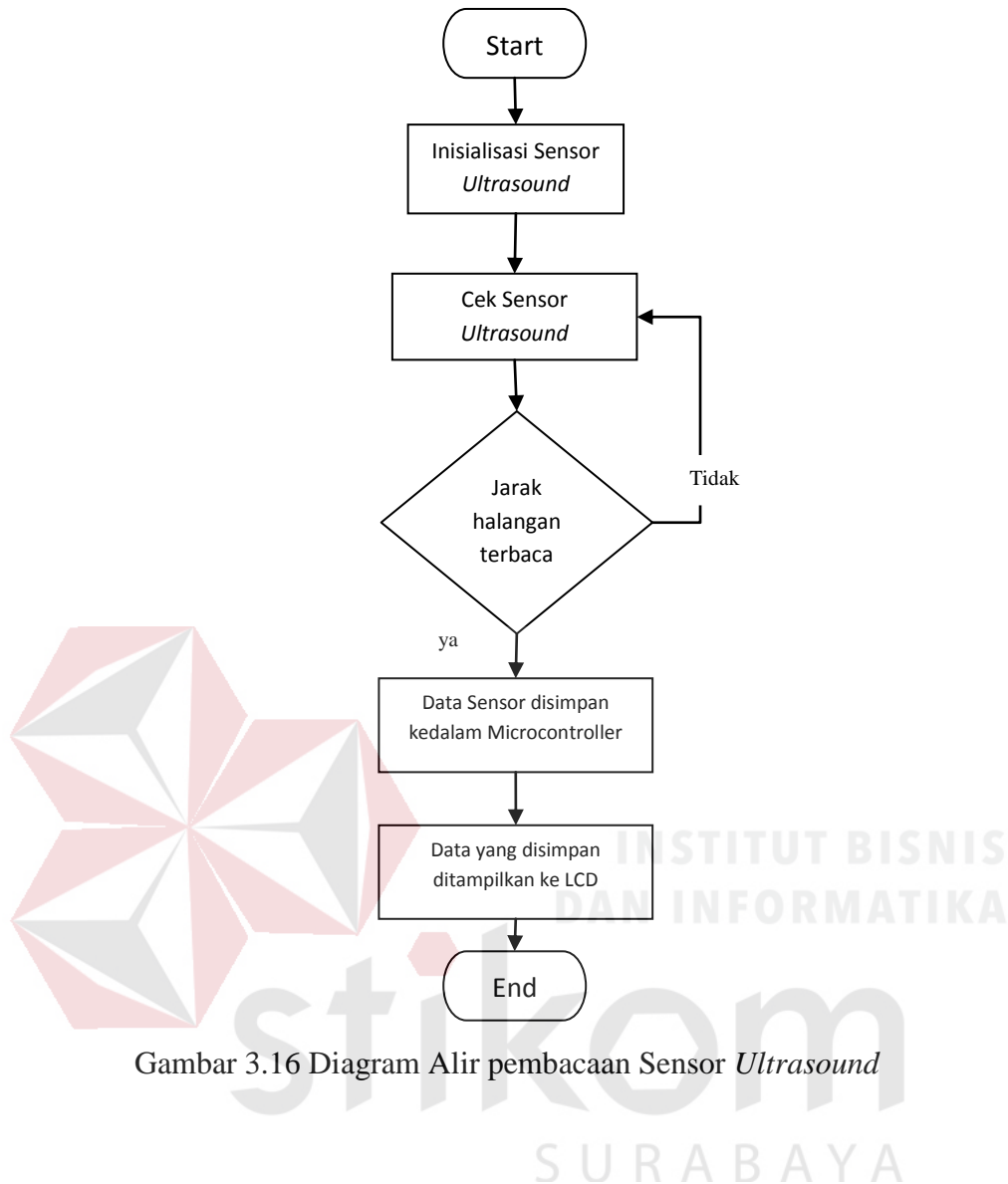
Perancangan perangkat lunak bertujuan untuk mengetahui informasi data yang sedang diproses. Minimum system memperoleh data dari sensor ultrasound yang berupa data jarak halangan antara sensor dengan banyaknya padi, serta mengatur kecepatan putaran motor DC melalui PWM yang dikirimkan oleh driver motor. Perancangan perangkat lunak terbagi dalam beberapa program antara lain : program motor DC, program membaca sensor, program algoritma fuzzy. Diagram alir perangkat lunak secara umum dapat dilihat pada gambar 3.25.



Gambar 3.15 Diagram alir program secara umum pada *microcontroller*

3.3.1 Program Membaca Jarak dan menampilkan ke LCD

Diagram alir untuk mengetahui jarak halangan berdasarkan pembacaan sensor ultrasound dan ditampilkan ke komputer :



Gambar 3.16 Diagram Alir pembacaan Sensor *Ultrasound*

Pada gambar 3.16, pembacaan jarak halangan dimulai dengan inisialisasi sensor. Jika jarak halangan berhasil dideteksi oleh sensor ultrasound maka data akan disimpan ke microcontroller dan ditampilkan ke LCD. Berikut potongan program pembacaan jarak halangan serta menampilkan ke LCD :

```

unsigned int ultrasonic()
{
    unsigned int sensor;
    unsigned int count=0;

    ARAH=OUT;//mengatur PIN I/O sebagai output
    PULSE=1;//memberikan tanda ke PING untuk memancarkan ultrasonic
    burst
    delay_us(5);//waktu tunggu sebelum pengukuran min. 2us biasanya 5us
    PULSE=0;//menberikan sinyal low ke PING
  
```



```

    ARAH=INP;//arah PIN I/O diatur sebagai input
    PULSE=1;//mengatur PIN I/O sebagai pull-up

while (ECHO==0) {};//menunggu sinyal ECHO high

while (ECHO==1)
{
    count++; //menghitung lebar sinyal ECHO high
}

sensor=(unsigned int)((float)count/115);
sprintf(kata,"Jarak = %3dcm", ultrasonic());//menyusun karakter ke dalam
satu baris dengan menggunakan fungsi sprintf
lcd_clear();
lcd_gotoxy(1,0);
lcd_puts(kata);

return(sensor)
}

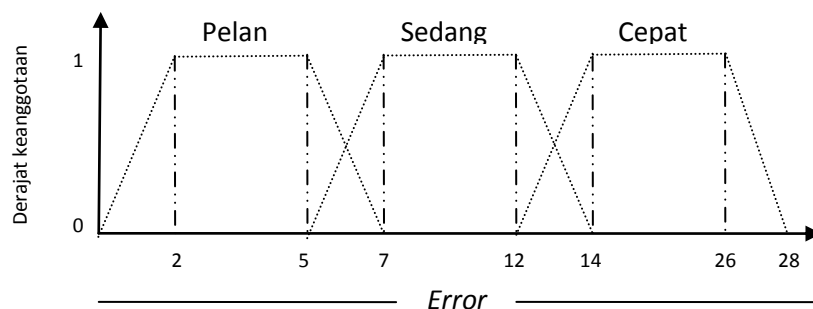
```

3.3.2 Fuzzy

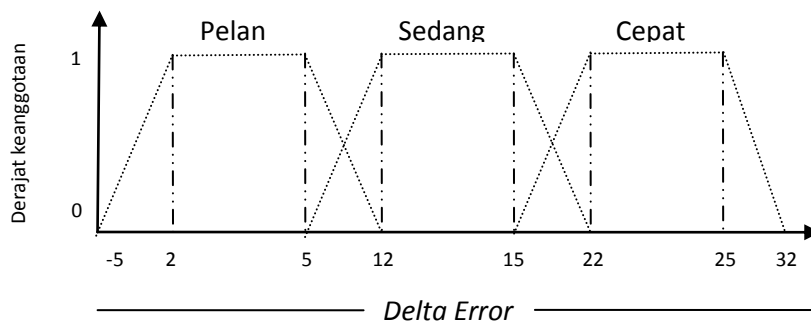
Untuk menentukan kecepatan motor dengan metode *fuzzy* terdapat 3 proses yaitu :

1. Fuzzifikasi

Pada proses fuzzifikasi terjadi pengambilan keputusan dengan cara menentukan error dan delta error ke dalam data fuzzy dengan tipe dan bentuk fungsi keanggotaan. Penentuan dari fungsi keanggotaan error dan delta error dapat dilihat pada gambar 3.17 dan 3.18.



Gambar 3.17 Pemetaan Derajat Keanggotaan *Error*



Gambar 3.18 Pemetaan Derajat Keanggotaan Delta Error

Pada saat proses fuzzifikasi, pendeteksian jarak halangan melalui sensor ultrasound dilakukan terlebih dahulu, selanjutnya data hasil pendeteksian tersebut akan dipetakan sesuai dengan fungsi keanggotaan fuzzy yang ada. Rumus yang ditentukan adalah sebagai berikut :

$Error = Set\ point - Present\ Value$

$Delta\ Error = Error\ baru - Error\ lama$

Realisasi proses memperoleh nilai derajat keanggotaan dari masing masing fungsi keanggotaan adalah sebagai berikut :

```

erorlama=0;
setpoint=28;
error = setpoint - sensor ;
delta = error - erorlama;

//-----Pelan Error-----//
dke[0] = anggota(0, 2, 5, 7, error);
//-----Sedang Error-----//
dke[1] = anggota(5, 7, 12, 14, error);
//-----Cepat Error-----//
dke[2] = anggota(12, 14, 26, 28, error);
//-----Pelan Delta Error-----//
dkd[0] = anggota2(-5, 2, 5, 12, delta);
//-----Sedang Delta Error-----//
dkd[1] = anggota2(5, 12, 15, 22, delta);
//-----Cepat Delta Error-----//
dkd[2] = anggota2(15, 22, 25, 32, delta);

float anggota(float a, float b, float c, float d, float error)
{
    float u1;

```

```

    if ((a==0) && (d==7))
    {
        if ((error>=0) && (error <=2))
        {
            u1= (error-a) / (b-a);
        }

        if ((error >=5) && (error <=7))
        {
            u1= (d-error)/(d-c);
        }

        if ((error>=2) && (error <=5))
        {
            u1=1;
        }

    }

    else if ((a==5) && (d==14))
    {
        if ((error>=a) && (error <=b))
        {
            u1= (error-a) / (b-a);
        }
        else {u1=0;}

        if ((error >=c) && (error <=d))
        {
            u1= (d-error)/(d-c);
        }

        if ((error>=b) && (error <= c))
        {
            u1=1;
        }

    }

    else if ((a==12) && (d==28))
    {
        if ((error>=a) && (error <=b))
        {
            u1= (error-a) / (b-a);
        }
        else {u1=0;}

        if ((error >=c) && (error <=d))
        {
            u1= (d-error)/(d-c);
        }

        if ((error>=b) && (error <= c))
        {
            u1=1;
        }

    }
}

```

```

    return u1;
}
float anggota2(float a2, float b2, float c2, float d2, float delta)
{
    float u2;

    if ((a2==5) && (d2==12))
    {
        if ((delta >= a2) && (delta <= b2))
        {
            u2= (delta - a2) / (b2 - a2);
        }

        if ((delta >= c2) && (delta <= d2))
        {
            u2= (d2 - delta)/(d2 - c2);
        }

        if ((delta >= b2) && (delta <= c2))
        {
            u2=1;
        }
    }
    else if ((a2==5) && (d2==22))
    {
        if ((delta >= a2) && (delta <= b2))
        {
            u2= (delta - a2) / (b2 - a2);
        }
        else {u2=0;}

        if ((delta >= c2) && (delta <= d2))
        {
            u2= (d2 - delta)/(d2 - c2);
        }

        if ((delta >= b2) && (delta <= c2))
        {
            u2=1;
        }
    }

    else if ((a2==15) && (d2==32))
    {
        if ((delta >= a2) && (delta <= b2))
        {
            u2= (delta - a2) / (b2 - a2);
        }
        else {u2=0;}

        if ((delta >= c2) && (delta <= d2))
        {
            u2= (d2 - delta)/(d2 - c2);
        }
    }
}

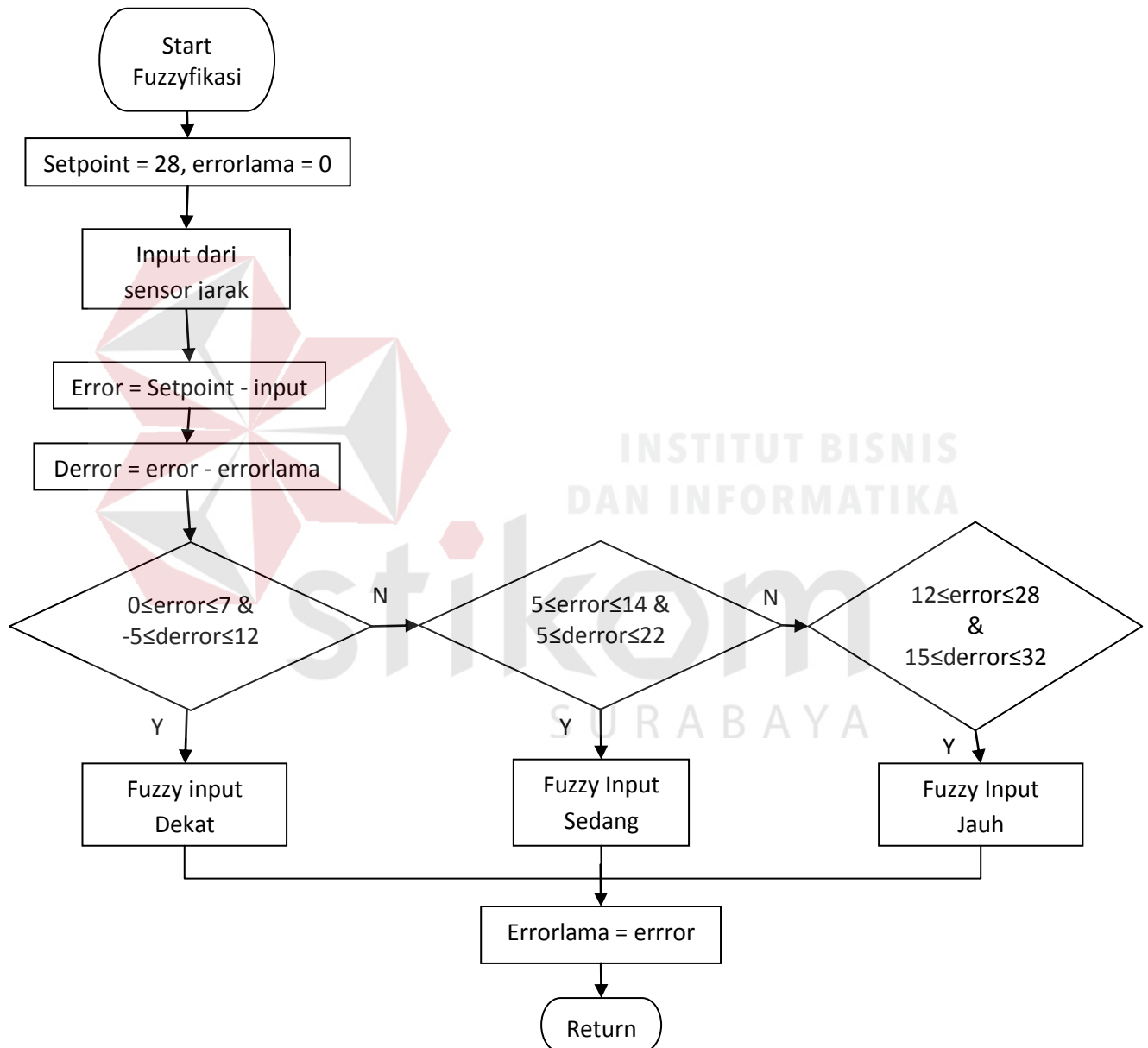
```

```

        if ((delta >= b2) && (delta <= c2))
        {
            u2=1;
        }
    }

    return u2;
}

```



Gambar 3.19 Diagram alir Fuzzifikasi

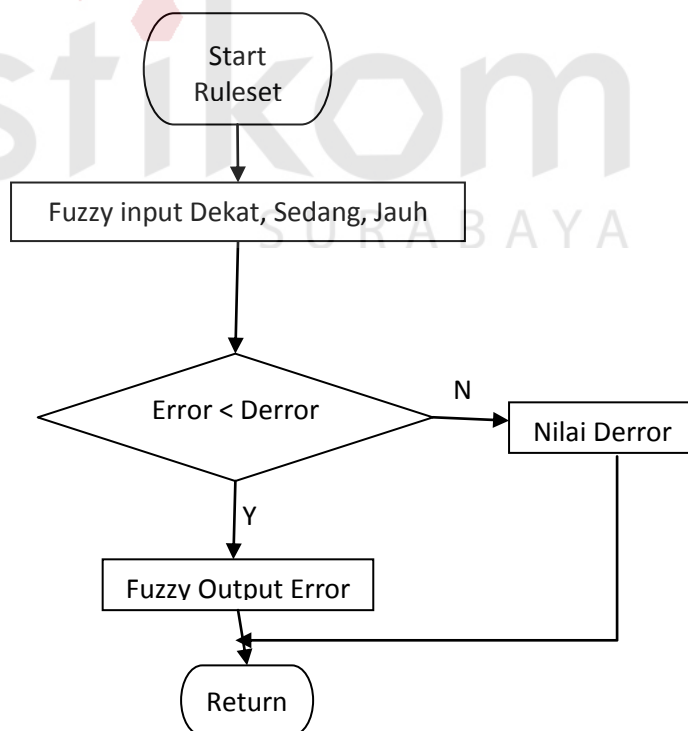
2. Rule Set

Rule Set / Evaluasi aturan adalah proses mengevaluasi derajat keanggotaan tiap-tiap fungsi keanggotaan himpunan *fuzzy* masukkan ke dalam basis aturan yang

telah ditetapkan. Tujuan dari evaluasi aturan ini adalah menentukan derajat keanggotaan dari keluaran *fuzzy*. Sebelum melakukan evaluasi aturan terlebih dahulu ditetapkan basis aturan. Basis aturan merupakan keseluruhan aturan dari kombinasi dua masukan yang mungkin. Secara lengkap, jumlah kombinasi yang mungkin dari dua himpunan *fuzzy* masukan dengan masing-masing 3 fungsi keanggotaan sehingga jumlah aturan nya adalah Sembilan aturan. Basis aturan yang dibuat berdasarkan *error* dan *delta error*. Pada bentuk yang lebih sederhana, Sembilan aturan kendali *fuzzy* dapat dilihat pada table **MacVicarWhelan** berikut

Tabel 3.4 Penentuan Ruleset

<i>Delta Error</i>	<i>Error</i>			
		<i>Dekat</i>	<i>Sedang</i>	<i>Jauh</i>
	<i>Dekat</i>	Pelan	Sedang	Cepat
	<i>Sedang</i>	Pelan	Sedang	Sedang
	<i>Jauh</i>	Sedang	Sedang	Cepat



Gambar 3.20 Diagram alir Rule Set

Berikut potongan program dari rule set :

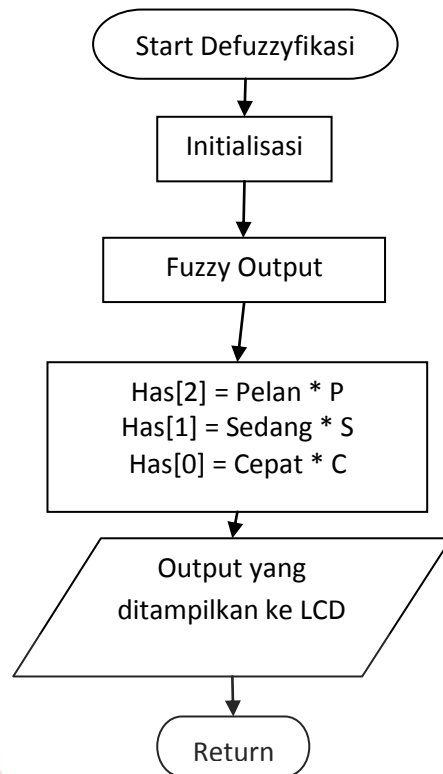
```
k=0;
for(i=0 ; i < 3; i++)
{
for(j=0; j < 3; j++)
{
u[k] = ruleset(dke[i],dkd[j]);
k++;
};
};
```

```
total_u=0;
for (i=0; i<9; i++)
{
total_u = total_u + u[i];
};
float ruleset(float a, float b)
{ float nilai;
if (a<b)
{
nilai=a;
}
else
{
nilai=b;
}
return nilai;
```

3. Defuzzifikasi

Defuzzifikasi adalah kebalikan dari proses fuzzyfikasi yaitu mengubah himpunan fuzzy keluaran menjadi keluaran tegas (Crisp). Pengubahan ini diperlukan karena konstanta pengendali fuzzy hanya mengenal nilai tegas sebagai variable sinyal kontrol. Hasil keluaran crisp output akan dikirim ke LCD . realisasi proses pengambilan keputusan metode sugeno menjadi bentuk crisp output dalam bentuk rumus sebagai berikut :

$$Crisp Out = \frac{\sum Fuzzy Output \times (posisi singleton X axis)}{\sum Fuzzy Output}$$



Gambar 3.21 Diagram alir Defuzzifikasi

Berikut potongan program dari defuzzifikasi :

```

C=255;
S=160;
P=100;

maxc = op(u[2],u[8]);
has[0] = maxc * C;
maxs[0]= op(u[1],u[4]);
maxs[1]= op(u[5],u[6]);
maxs[2]= op(maxs[0],maxs[1]);
maxs[3]= op(maxs[2],u[7]);
// maxs[4]= op(maxs[3],u[7]);
has[1]= maxs[3] * S;
maxp = op(u[0],u[3]);
has[2]= maxp * P;
pembilang=0;
for(j=0; j<3; j++)
{
  pembilang = pembilang + has[j];
};

hasil = pembilang / total_u;
  
```


BAB IV

PENGUJIAN SISTEM

Pada bab pengujian sistem ini dijelaskan hasil dan analisis pengujian yang telah dilakukan. Pengujian tersebut berupa pengujian terhadap perangkat keras dan pengujian terhadap perangkat lunak yang telah dibuat.

4.1 Pengujian *Minimum system*

4.1.1 Tujuan

Pengujian *minimum system* bertujuan untuk mengetahui apakah *minimum system* dapat melakukan proses *signature* dan *download* program ke *microcontroller* dengan baik.

4.1.2 Alat yang digunakan

Peralatan yang penulis butuhkan untuk pengujian ini adalah sebagai berikut :

1. Rangkaian *minimum system* ATmega8535
2. Downloader
3. PC
4. Program *CodeVisionAVR*
5. Power supply 10000mA – 24V
6. Kabel *downloader port parallel*

4.1.3 Prosedur pengujian

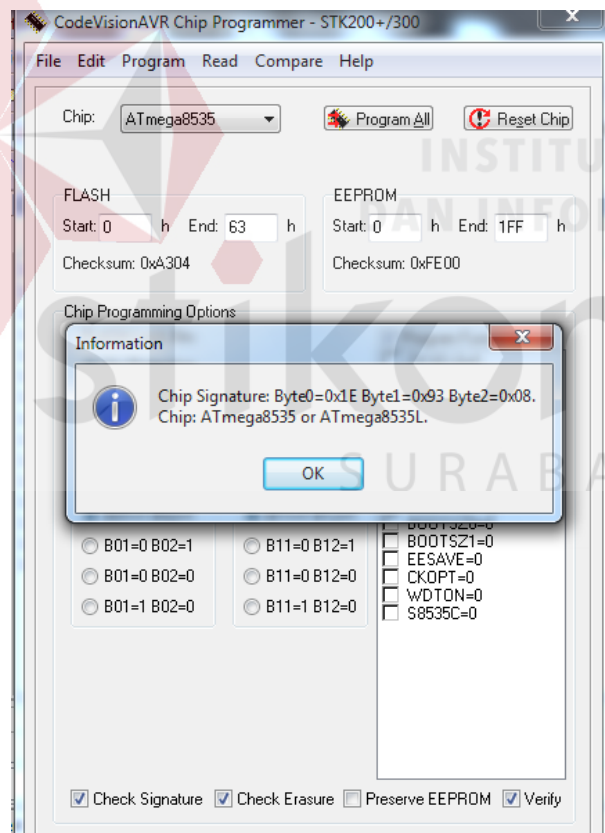
Langkah-langkah untuk melakukan pengujian *minimum system* adalah sebagai berikut :

1. Hubungkan *power supply* dengan regulator serta *minimum system* ATmega8535.
2. Sambungkan *minimum system* dengan kabel downloader pada port parallel.

3. Jalankan program *CodeVisionAVR*.
4. Untuk mengetahui *minimum system* sudah terbaca dengan baik oleh program *CodeVisionAVR* adalah dengan cara menjalankan *Chip Signature programmer*.
5. Setelah *minimum system* terbaca dengan baik selanjutnya *compile project* dengan menekan F9 pada *keyboard* kemudian proses *download* program.

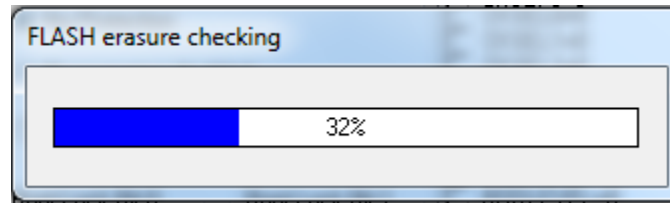
4.1.4 Hasil pengujian

Dari percobaan diatas apabila menu chip signature programmer download program berhasil maka minimum system dapat dikatakan bekerja dengan baik. Berikut hasil percobaan chip signature dapat dilihat pada gambar 4.1.



Gambar 4.1 Chip Signature

Pada gambar 4.2 menunjukan bahwa minimum system sedang dalam proses men-download program ke *microcontroller*.



Gambar 4.2 Download program

4.2 Pengujian minimum system ATmega8535 dengan LCD

4.2.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah *microcontroller* ATmega8535 dapat memberikan output ke LCD.

4.2.2 Alat yang digunakan

Peralatan yang penulis butuhkan untuk pengujian ini adalah sebagai berikut :

1. Rangkaian *minimum system ATmega8535*
2. LCD 16x2
3. *Power Supply* 10000mA- 24V
4. Regulator +5V

4.2.3 Prosedur pengujian

Langkah-langkah untuk melakukan pengujian *minimum system* adalah sebagai berikut :

1. Hubungkan LCD dengan *minimum system*.
2. Hubungkan power supply dengan regulator serta *minimum system ATmega8535*.
3. Hubungkan terminal +5V pada *minimum system ATmega8535* pada (+) LCD.
4. Hubungkan terminal 0V pada *minimum system ATmega8535* pada (-) LCD.

5. Hubungkan PORTA pada *minimum system ATmega8535* dengan pin LCD seperti table dibawah ini :

Tabel 4.1 Konfigurasi *minimum system ATmega8535* dengan LCD 16x2

Minimum System ATmega8535	LCD 16x2
PORTA.0	RS
PORTA.1	R/W
PORTA.2	E
PORTA.4	DB4
PORTA.5	DB5
PORTA.6	DB6
PORTA.7	DB7

6. Masukkan instruksi program kedalam *minimum system ATmega8535* untuk menampilkan tulisan ke LCD 16x2.

```
/*  
This program was produced by the  
CodeWizardAVR V2.05.3 Standard  
Automatic Program Generator  
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project :  
Version :  
Date : 6/24/2014  
Author : tyery08  
Company : embeeminded.blogspot.com  
Comments:
```

```
Chip type : ATmega8535  
Program type : Application  
AVR Core Clock frequency: 11.000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 128  
*****/
```

```
#include <mega8535.h>
```

```
// Alphanumeric LCD functions  
#include <alcd.h>
```

```
// Declare your global variables here
```

```
void main(void)
```

```

{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=Out Func4=Out Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=0 State4=0 State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x30;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 10.742 kHz
// Mode: Fast PWM top=0x00FF
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA1;
TCCR1B=0x0D;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;

```

```

OCR1BL=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTA Bit 0
// RD - PORTA Bit 1
// EN - PORTA Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 16
lcd_init(16);

/*****Tulisan yang akan tampil pada
LCD*****/
lcd_gotoxy(0,0);
lcd_puts(" W E L C O M E ");
lcd_gotoxy(1,0);
lcd_puts("by : Stefanus A.T");

while (1)
{
    // Place your code here
}

```

4.2.4 Hasil Pengujian

Pada pengujian ini diperoleh output pada LCD yang sesuai dengan instruksi program yang telah dibuat. Untuk mengetahui apakah program berjalan sesuai program dapat berjalan sesuai dengan cuplikan program diatas dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan LCD

4.3 Pengujian Sensor ultrasound dengan minimum system ATmega8535 dan LCD

4.3.1 Tujuan

Pengujian sensor ultrasound dilakukan untuk mengetahui informasi jarak antar sensor dengan benda di depannya. Dalam hal ini penulis menggunakan banyak paku yang menumpuk. Data yang dihasilkan selanjutnya akan diolah oleh microcontroller dan selanjutnya akan ditampilkan hasilnya ke LCD.

4.3.2 Alat yang digunakan

Peralatan yang penulis butuhkan untuk pengujian ini adalah sebagai berikut :

1. Minimum system ATmega8535
2. LCD 16x2
3. Sensor Ultrasound
4. Power Supply 10000mA – 24V
5. Regulator +5V

4.3.3 Prosedur pengujian

Langkah-langkah untuk melakukan pengujian *minimum system* adalah sebagai berikut :

1. Hubungkan LCD dan Sensor Ultrasound dengan Minimum System ATmega8535.
2. Hubungkan power supply dengan regulator serta *minimum system ATmega8535*.
3. Hubungkan terminal +5V pada *minimum system ATmega8535* pada (+) LCD dan (+) sensor Ultrasound.
4. Hubungkan terminal 0V pada *minimum system ATmega8535* pada (-) LCD dan (-) sensor Ultrasound.
5. Masukkan instruksi program kedalam *minimum system ATmega8535* untuk mengolah data sensor ultrasound dan menampilkan ke LCD.

```
unsigned int ultrasonic()
{
    unsigned int sensor;
    unsigned int count=0;

    ARAH=OUT;//mengatur PIN I/O sebagai output
    PULSE=1;//memberikan tanda ke PING untuk memancarkan ultrasonic
    burst
    delay_us(5);//waktu tunggu sebelum pengukuran min. 2us biasanya
    5us
    PULSE=0;//menberikan sinyal low ke PING
    ARAH=INP;//arah PIN I/O diatur sebagai input
    PULSE=1;//mengatur PIN I/O sebagai pull-up

    while (ECHO==0) {};//menunggu sinyal ECHO high

    while (ECHO==1)
    {
        count++; //menghitung lebar sinyal ECHO high
    }

    sensor=(unsigned int)(((float)count)/115);//nilai pembagi dikalibrasi
    sampai sesuai dengan satuan yang diinginkan

    return(sensor);//mengembalikan jarak ke fungsi ultrasonic dengan tipe
    data unsigned int
}

while (1)
{
```



```
// Place your code here
unsigned char kata[16];
sprintf(kata,"Jarak = %3dcm", ultrasonic()); //menyusun karakter
ke dalam satu baris dengan menggunakan fungsi sprintf
lcd_clear();
lcd_gotoxy(1,0);
lcd_puts(kata);
delay_ms(1000);
}
```

4.3.4 Hasil pengujian

Pada pengujian ini diperoleh output jarak sensor ultrasound dalam satuan centimeter (cm). berikut tabel hasil pengujian sensor ultrasound.

Tabel 4.2 Hasil pembacaan sensor ultrasound

No	Jarak Sebenarnya (cm)	Jarak Terdeteksi (cm)
1	2	2.1
2	3	3
3	4	4
4	5	5.1
5	6	6.2
6	7	7.1
7	8	8
8	9	9.2
9	10	10
10	11	11
11	12	12.2
12	13	13.1
13	14	14.1
14	15	15
15	16	16.2
16	17	17.1
17	18	18.1
18	19	19
19	20	20
20	21	21.1
21	22	22
22	23	23.2
23	24	24
24	25	25.2
25	26	26.1
26	27	27.1
27	28	28

4.4 Pengujian Motor driver IC L298 dan VNH30SP30 dengan minimum system ATmega8535

4.4.1 Tujuan

Pengujian Motor driver dilakukan untuk mengetahui kerja IC motor driver dapat mengeluarkan PWM secara baik. Dalam hal ini penulis merubah rubah nilai PWM dengan cara mencounter nilai PWM dari 0-255.

4.4.2 Alat yang digunakan

Peralatan yang penulis butuhkan untuk pengujian ini adalah sebagai berikut :

1. Minimum System ATmega8535
2. Motor driver IC L298
3. Motor driver VNH30SP30
4. Motor DC
5. Power Supply 10000mA – 24V

4.4.3 Prosedur pengujian

Langkah-langkah untuk melakukan pengujian *minimum system* adalah sebagai berikut :

1. Hubungkan power supply dengan regulator serta *minimum system ATmega8535*.
2. Hubungkan terminal +5V pada *minimum system ATmega8535* pada (+) IC L298 (+) IC VNH30SP30.
3. Hubungkan terminal 0V pada *minimum system ATmega8535* pada (-) IC L298 dan (-) IC VNH30SP30.
4. Hubungkan input 1, 2 pada PORTB.6 DAN PORTB.7.

5. Hubungkan MIN1, MIN2, MEN1, MEN2 pada PORTC.2, PORTC.3, PORTC.4, PORTC.5.
6. Hubungkan EN1 L298 pada PORTD.4.
7. Hubungkan MPWM VNH30SP30 pada PORTD.5.
8. Masukkan intruksi program kedalam minimum system ATmega8535

```
int i;

PORTC.2 = 1;
PORTC.3 = 0;
PORTC.4 = 1;
PORTC.5 = 1;
PORTB.1 = 1;
PORTB.2 = 0;

i=0;
for(i<256; i=0;i+5)
{
    OCR1A = i;
    OCR1B = i;
}
```

4.4.4 Hasil pengujian

Pada pengujian ini didapatkan hasil bahwa motor driver L298 dan VNH30SP30 dapat mengatur kecepatan motor DC dengan PWM.



Gambar 4.4 Hasil PWM

4.5 Pengujian metode Fuzzy

4.5.1 Tujuan

Pengujian ini dilakukan untuk menguji apakah program fuzzy yang dibuat mampu berjalan sesuai dengan keinginan dan kebutuhan tugas akhir ini.

4.5.2 Alat yang digunakan

Peralatan yang penulis butuhkan untuk pengujian ini adalah sebagai berikut :

1. Minimum System ATmega8535
2. Power supply 10000mA – 24V
3. Regulator +5V
4. Motor DC 24V
5. Sensor Ultrasound

4.5.3 Prosedur pengujian

Langkah-langkah untuk melakukan pengujian *minimum system* adalah sebagai berikut :

1. Hubungkan sensor ultrasound, motor DC, LCD dengan minimum system.
2. Hubungkan power supply dengan regulator serta *minimum system ATmega8535*.
3. Amati data yang tampil pada LCD.
4. Lakukan percobaan beberapa kali untuk mengetahui hasil dari metode fuzzy.

4.5.4 Hasil pengujian

Hasil pengujian dilakukan secara bertahap dengan melihat hasil yang tampil pada LCD.

Berikut adalah hasil pengujian fuzzy dengan percobaan ketinggian padi sebanyak 10 kali :



Gambar 4.5 Hasil Pengujian dengan *setpoint* 28

Tabel 4.3 Pengujian dengan *setpoint* 28

No	Setpoint	Sensor	Hasil	PWM	Error (Hasil – PWM)
1	28	2	208	207	1
2	28	3	208	207	1
3	28	7	171	171	0
4	28	10	135	134	1
5	28	20	117	117	0
6	28	25	100	100	0
7	28	23	151	151	0
8	28	17	100	100	0
9	28	14	80	80	0
10	28	12	98	98	0
Rata-Rata		13.3	136.8	136.5	0.3
				%Error	0.21%

4.6 Pengujian Keseluruhan sistem

4.6.1 Tujuan

Pengujian keseluruhan sistem bertujuan untuk mengetahui berapa besar daya yang dibutuhkan saat proses berlangsung.

4.6.2 Alat yang digunakan

Peralatan yang penulis butuhkan untuk pengujian ini adalah sebagai berikut :

1. Rangkaian minimum system ATmega8535
2. Power supply 10000mA – 24V
3. Regulator +5V
4. Motor DC
5. Sensor ultrasound
6. LCD

4.6.3 Prosedur pengujian

Langkah – langkah untuk melakukan pengujian metode fuzzy adalah sebagai berikut :

1. Hubungkan sensor ultrasound, driver motor DC, LCD dengan minimum system
2. Hubungkan power supply dengan regulator serta *minimum system ATmega8535*.
3. Amati jalannya proses eksekusi program pada dan yang tampil pada LCD.

4.6.4 Hasil pengujian

Pengujian keseluruhan sistem dilakukan dengan melakukan pengamatan dan mengamati setiap perubahan yang tampil pada LCD.

Tabel 4.4 Hasil pengujian keseluruhan sistem

No	Sensor	PWM	I (Ampere/hour)	V (Volt)	P (Watt/hour)	P . 3/60
1	2	203	9.1	24	218.4	10.92
2	3	207	8.4	24	201.6	10.08
3	6	189	8.3	24	199.2	9.96
4	9	152	6.5	24	156	7.8
5	11	116	5.1	24	122.4	6.12
6	13	80	3.5	24	84	4.2
7	16	160	6.8	24	163.2	8.16
8	17	151	6.2	24	148.8	7.44
9	23	100	4.5	24	108	5.4
10	26	100	4.2	24	100.8	5.04
Total						75.12
% Efisiensi						31%

Dari hasil diatas jika menggunakan fuzzy daya yang dibutuhkan adalah 75.12 watt, tetapi jika tidak menggunakan fuzzy daya yang dibutuhkan adalah 109.2 watt. Efisiensi daya nya adalah 31%.



Gambar 4.6 Tampilan alat sebenarnya



Gambar 4.7 Hasil padi setelah pecah kulit

BAB V

PENUTUP

Berdasarkan pengujian pada perangkat keras dan perangkat lunak yang dipergunakan dalam tugas akhir ini, maka dapat diambil kesimpulan dan saran-saran dari hasil yang diperoleh.

5.1 Kesimpulan

Setelah melakukan penelitian dan pengujian, penulis mengambil kesimpulan sebagai berikut :

1. Mesin sudah dapat berjalan otomatis, saat gabah dalam tampungan banyak motor bergerak secara cepat tetapi pada saat gabah dalam tampungan sedikit kecepatan motor akan berkurang.
2. Program fuzzy yang dibuat mampu mengatur PWM untuk mengatur kecepatan motor.
3. Dalam tugas akhir ini tombol yang digunakan adalah on dan off saja.
4. Efisiensi daya jika menggunakan fuzzy adalah 31%.

5.2 Saran

Sebagai pengembangan dari penelitian yang dilakukan, penulis memberikan saran sebagai berikut :

1. Penambahan jumlah derajat keanggotaan fuzzy sehingga pengaturan kecepatan motor dapat lebih stabil.
2. Memberikan gearbox agar kinerja motor tidak terlalu berat.
3. Penggantian pisau polesher agar padi dapat terasah lebih putih.



DAFTAR PUSTAKA

Allidawati dan B.Kustianto. 1989. Metode uji mutu beras dalam program pemuliaan padi.

Dalam: Ismunadji M., M. Syam dan Yuswadi. Padi Buku 2. Badan Penelitian dan Pengembangan Pertanian. Pusat Penelitian dan Pengembangan Tanaman Pangan. Bogor. Hal: 363-375.

Anonim. 1983. Studi Konservasi dan Susut Gabah ke Beras Tingkat Nasional. Biro Pusat Statistik, Departemen Pertanian, Badan Urusan Logistik, Badan Perencanaan Pembangunan Nasional, Fakultas Teknologi Pertanian, IPB. Bogor.

ATMEL Corporation. 2003, ATmega8535, (Online). (<http://www.atmel.com>, diakses 20 Juni 2014)

Damardjati, D.S. 1988. Struktur kandungan gizi beras. *Dalam:* Ismunadji, M., S.Partohardjono, M.Syam, A.Widjono. Padi-Buku 1. Balai Penelitian dan Pengembangan Pertanian, Pusat Penelitian dan Pengembangan Pertanian, Pusat Penelitian dan Pengembangan Tanaman Pangan. Bogor. Hal: 103-159.

Djati, Dmin. 2010. Motor DC. (Online). (<http://d-kecil.blogspot.com>). Diakses tanggal 20 Juni 2014.

PARALLAX INC. 2006, PING)))™ Ultrasonic Distance Sensor, diakses 20 Juni 2014. URL : (www.parallax.com.)

Sri Kusumaderi, Hari Purnomo, 2010. “Aplikasi Logika Fuzzy Untuk Pendukung Keputusan, edisi 2”. Yogyakarta, Graha Ilmu.

ST, Iswanto. 2008. “Design dan Implementasi Sistem Embedded Mikrokontroller ATmega8535 dengan Bahasa Basic”. Yogyakarta, Gava Media.

Waries, A. 2006. Teknologi Penggilingan Padi. PT Gramedia Pustaka Utama. Jakarta.

Winoto, Ardi. 2008. "Mikrokontroller AVR ATMEGA8/16/32/8535 dan Pemrogramannya dengan bahasa C pada WinAVR". Informatika. Bandung.

