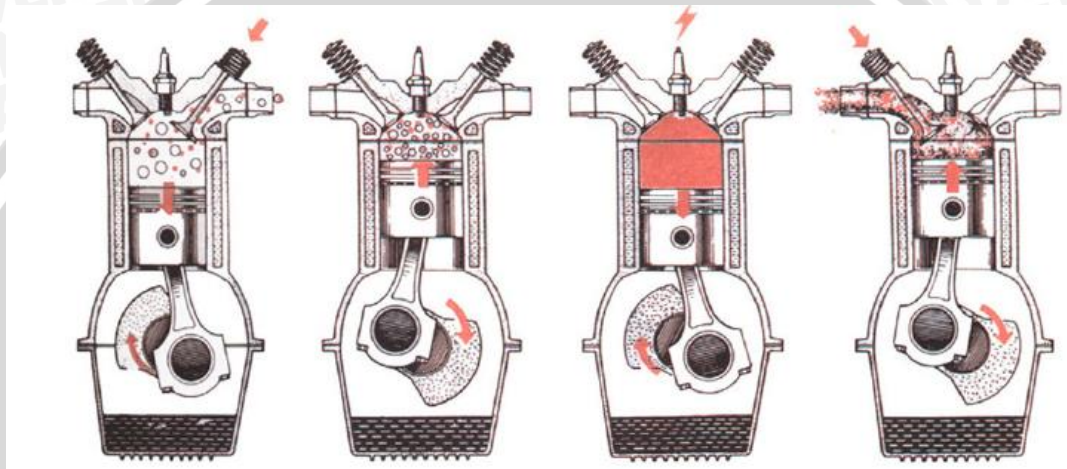


## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Proses Motor Bakar 4 Langkah

Mesin bakar yang dirancang oleh Nikolaus Otto tahun 1882, adalah dengan model mesin 4 tak. Piston menggunakan sebuah katup masukan (intake) dan keluaran (exhaust) untuk mengontrol aliran. Dua putaran crankshaft diperlukan untuk melengkapi sebuah siklus.



**Gambar 2.1.** Siklus pembakaran 4-langkah dasar

Gambar 2.1 menggambarkan siklus tersebut. Langkah mesin 4-langkah terdiri dari

- Langkah hisap (*induction*), piston bergerak turun dari titik mati atas (TMA) menuju titik mati bawah (TMB). Katup masuk membuka dan katup buang menutup, akibat gerakan piston sejumlah gas bensin terhisap masuk ke dalam silinder/ruang bakar melalui saluran katup masuk
- Langkah kompresi (*compression*), piston bergerak naik dari TMB menuju TMA, dimana saat itu baik katup masuk maupun katup keluar dalam kondisi tertutup. Akibat gerakan piston ini gas bensin dimampatkan sehingga suhunya naik.
- Langkah usaha (*combustion*), saat piston hampir mencapai TMA, busi menyala memercikkan bunga api, gas bensin yang sudah dimampatkan terbakar dan meledak menghasilkan daya dorong terhadap piston, sehingga piston bergerak dari TMA ke TMB.
- Langkah buang (*exhaust*), piston bergerak naik lagi dari TMB ke TMA, katup buang membuka. Gerakan piston ini mendorong gas sisa pembakaran keluar.

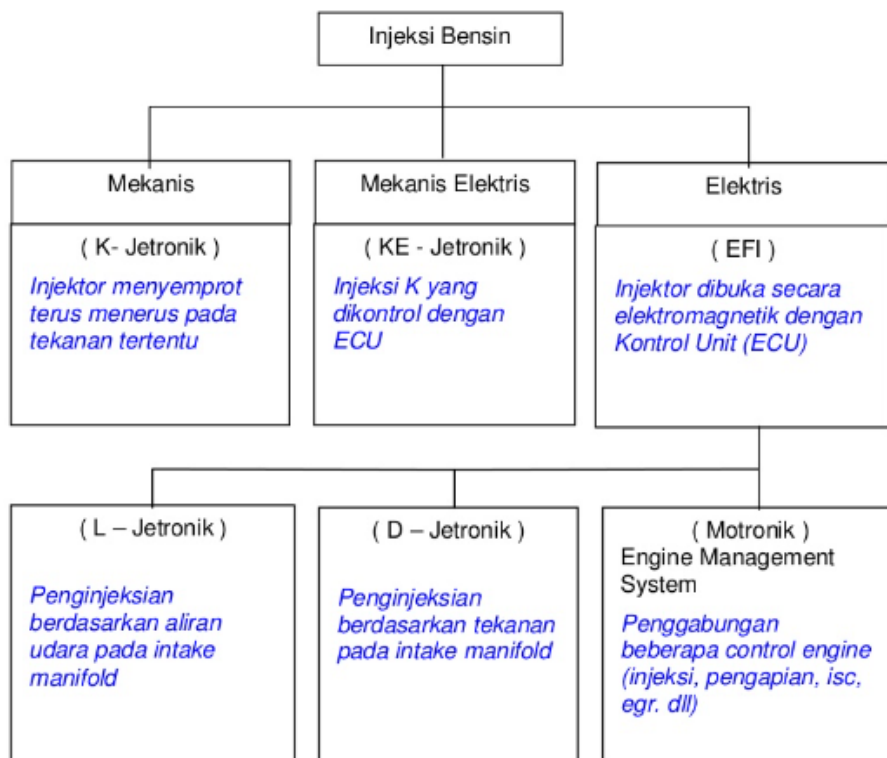
Kemudian siklus dimulai lagi dari langkah hisap, empat langkah piston menurut urutan di atas disebut satu siklus mesin 4-langkah [Arismunandar, 2000].

## 2.2. Sistem Injeksi

Sistem injeksi mulai ditemukan pada tahun 1922-1927 ketika Robert Bosch menemukan pompa injeksi diesel. Sejak tahun 1960 prinsip injeksi bensin mulai diterapkan pada kendaraan bermotor, pada tahun 1967 pabrik mobil VW sudah menerapkan sistem D-Jetronik, baru tahun 1973 sistem injeksi bensin mulai dipakai secara meluas pada kendaraan bermotor.

Sistem injeksi digunakan untuk menyemprotkan bahan bakar ke dalam *engine* yang akan dicampur dengan udara untuk keperluan pembakaran. Ditinjau dari sistem pengontrol penyemprotan sistem injeksi dapat dibedakan menjadi 3, yaitu sistem injeksi mekanis, sistem injeksi mekanis elektronik, sistem injeksi elektronik.

Prinsip kerja dari semua sistem injeksi sama, ada yang sistem injeksi dan sistem pengapian dibuat dengan ECU terpisah dan ada juga yang satu sistem ECU untuk sistem injeksi dan sistem pengapian.

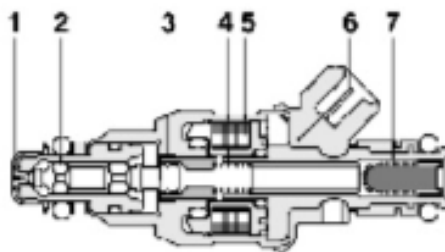


**Gambar 2.2.** Penggolongan Injeksi Berdasarkan Sistem Pengontrol  
(Sumber : Slamet Setiyono, 2008)

Sistem injeksi elektronik dilengkapi dengan sensor-sensor yang dapat dibedakan menjadi sensor utama dan sensor-sensor pengoreksi. Sensor utama digunakan untuk menentukan jumlah penyemprotan injeksi dasar dan sensor-sensor pengoreksi untuk merubah jumlah penyemprotan berdasarkan keadaan-keadaan kerja *engine*. Sensor utama untuk mengetahui jumlah udara yang masuk ke *engine* merupakan kombinasi dari sensor massa udara dan sensor putaran mesin. Kedua sensor tersebut menginformasikan kepada ECU berapa jumlah udara yang masuk ke *engine* pada setiap putaran, lalu ECU memberi sinyal kepada injektor dengan durasi penyemprotan tertentu. Harapan dari pengaturan menginginkan bahan bakar diinjeksikan dengan jumlah yang perbandingannya sesuai dengan hukum *stoichiometric*, 14,7 kg massa udara untuk setiap 1 kg bensin. Karena keadaan kerja *engine* sangat beragam dan kebutuhan perbandingan campuran juga beragam maka dipasangkan sensor-sensor lain.

### 2.2.1. Injektor

Injektor berfungsi menyemprotkan bensin menuju *engine* untuk dicampur dengan udara. Agar bensin mudah bercampur dengan udara maka bensin dikabutkan dengan halus sehingga mudah berubah menjadi uap. Bagian injektor ditunjukkan pada Gambar 2.3



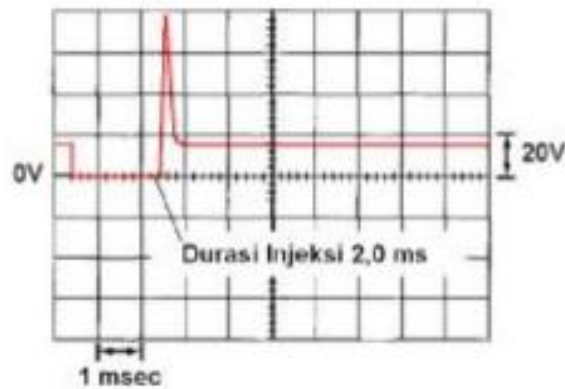
**Gambar 2.3.** Injektor  
(Sumber : Slamet Setiyono, 2008)  
Keterangan :

1. Pintle
2. Katup jarum
3. Jangkar
4. Pegas
5. Kumpan solenoid
6. Terminal/konektor
7. Strainer/saringan

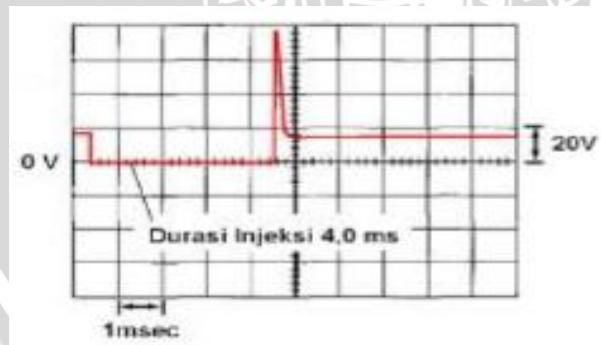
Injektor pada sistem injeksi bekerja secara elektromagnetik. Kerja injector dikontrol oleh ECU dengan sinyal negatif. Lebar pulsa sinyal dari ECU akan menentukan jumlah bahan bakar yang terkabutkan, semakin panjang pulsa semakin banyak bensin terkabutkan.

### 2.2.2. Pengaturan Injeksi

Durasi injeksi pada sistem injeksi elektronik dipengaruhi oleh kondisi-kondisi kerja *engine* yang dapat diketahui dengan memasang sensor-sensor. Grafik durasi injeksi saat idle dan berbeban ditunjukkan pada Gambar 2.4 dan 2.5.



Gambar 2.4. Durasi Injeksi Saat Idle

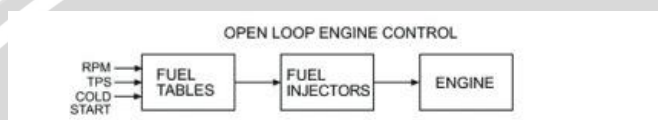


Gambar 2.5. Durasi Saat Berbeban

### 2.3. Engine Control Unit (ECU)

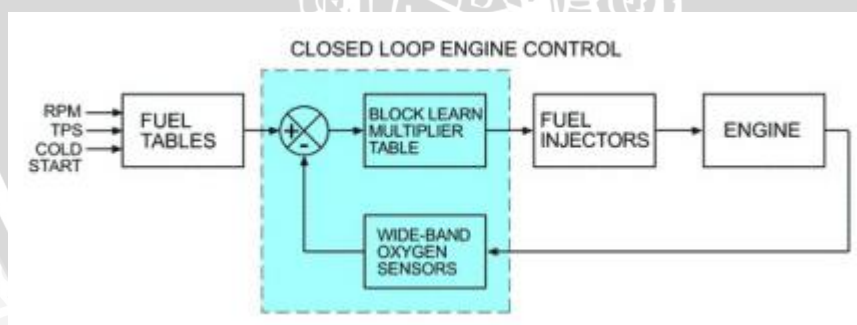
ECU merupakan *embedded system* yang terdiri dari mikrokontroler dan rangkaian elektronis lainnya, yang didalamnya terdapat program dan memori mengenai perlakuan terhadap mesin pada kondisi-kondisi yang berbeda-beda. Misalnya kondisi dihidupkan dalam keadaan mesin dingin, kondisi massa udara yang tinggi, kondisi kecepatan rpm tinggi dan lainnya.

Dalam tugasnya mengontrol injektor, jenis ECU ada dua macam, *open loop* dan *close loop*. Sistem *open-loop* ECU ditunjukkan pada Gambar 2.6



**Gambar 2.6.** Sitem *Open Loop* ECU  
(Sumber : daytona-twintec.com)

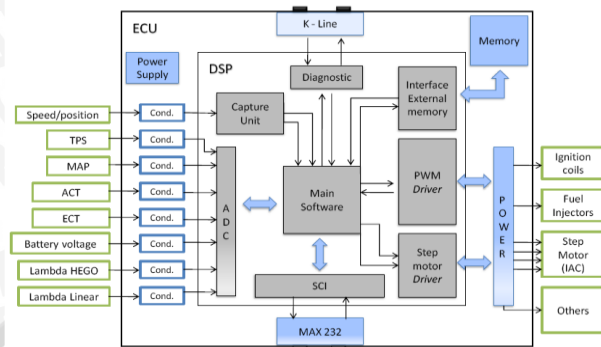
Sistem *open loop*, ECU membaca sinyal dari sensor rpm, *Throttle Position Sensor* (TPS), *air flow*, *manifold pressure*, *intake air pressure*, dll. Kemudian memilihkan suatu nilai dalam tabel di memori, berdasarkan nilai tersebut dibangkitkan sinyal untuk dikirim ke injektor agar mensuplai bensin untuk mencapai *air fuel ratio* yang diinginkan.



**Gambar 2.7.** Sitem *Close Loop* ECU  
(Sumber : daytona-twintec.com)

Sedangkan yang *close loop*, merupakan pengembangan dari sistem *open loop*, yang ditambahkan sensor  $O_2$  pada knalpot untuk mengukur kandungan  $O_2$  pada knalpot, jika kandungan  $O_2$  tinggi maka pembakaran miskin dan sebaliknya. Kemudian sensor  $O_2$  memberikan sinyal ke ECU untuk mengoreksi suplai injektor bbm. ECU jenis *close loop*

jauh lebih maju atau rumit daripada *open loop* karena cara kerjanya memerlukan algoritma program mikrokontroler yang panjang. Diagram blok ECU ditunjukkan pada Gambar 2.8



**Gambar 2.8.** Diagram Blok ECU  
(Sumber : motogokil.com)

#### 2.4. Perbandingan Bahan Bakar dan Udara (AFR)

Pada kondisi beroperasi, *spark-ignition engine* membutuhkan campuran antara bahan bakar dan udara yang direpresentasikan dengan rasio perbandingan jumlah bahan bakar dengan udara yang tercampur (*Air to Fuel Ratio / AFR*). Perbandingan ideal dari udara dan bahan bakar untuk satu proses pembakaran adalah 14,7 : 1 yang berarti bahan bakar sebanyak 1 g berbanding dengan udara sebanyak 14,7 g dan disebut sebagai AFR stokiometrik.

Nilai AFR yang ideal menggambarkan titik keseimbangan terbaik antara daya mesin dengan bahan bakar yang dikeluarkan. Nilai AFR yang lebih besar dari 14,7 : 1 disebut campuran miskin, yang berarti udara lebih banyak daripada kondisi ideal, sehingga daya mesin berkurang dan bahan bakar lebih irit. Nilai AFR yang lebih kecil dari 14,7 : 1 disebut campuran kaya, yang berarti bahan bakar lebih banyak dikeluarkan, namun daya mesin meningkat. AFR kondisi aktual yang terjadi di ruang pembakaran dibagi dengan AFR stokiometrik diperoleh harga AFR relatif yang disebut *Lambda* ( $\lambda$ ).

#### 2.5. Sensor Crankshaft Position (CKP)

Sensor *Crankshaft Position* adalah perangkat elektronik yang digunakan dalam mesin pembakaran internal untuk memantau posisi atau kecepatan rotasi crankshaft. Informasi ini digunakan oleh sistem manajemen mesin untuk mengontrol waktu sistem pengapian dan parameter mesin lainnya. Sebelum sensor engkol elektronik yang tersedia,

distributor harus diatur secara manual untuk tanda waktu pada mesin. Berikut gambar sensor CKP ditunjukkan pada Gambar 2.9



**Gambar 2.9.** *Crankshaft Position Sensor (CKP)*  
(Sumber : Wikipedia.org)

Sensor engkol dapat digunakan dalam kombinasi dengan sensor posisi camshaft mirip dengan memantau hubungan antara piston dan katup di mesin, yang sangat penting dalam mesin dengan variable valve timing. Metode ini juga digunakan untuk "sinkronisasi" mesin empat langkah pada awal, memungkinkan sistem manajemen untuk mengetahui kapan harus menyuntikkan bahan bakar. Hal ini juga sering digunakan sebagai sumber utama untuk pengukuran kecepatan mesin di putaran per menit.

Lokasi pemasangan umum termasuk katrol engkol utama, roda gila, camshaft atau poros engkol itu sendiri. Sensor ini adalah sensor yang paling penting dalam mesin modern. Ketika gagal, ada kemungkinan mesin tidak akan mulai, atau memotong sambil berjalan.

## 2.6. Sensor Suhu LM35

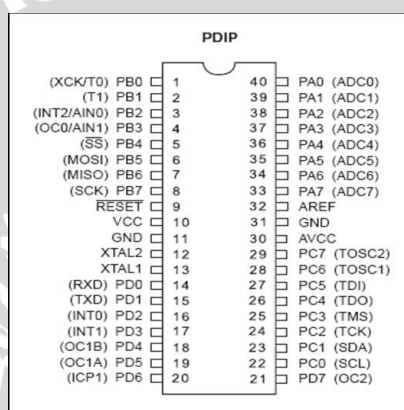
Sensor LM35 adalah sensor suhu presisi yang berbentuk IC, yang output tegangannya linier sebanding dengan temperature (sakala Celcius). Dengan demikian LM35 memiliki keunggulan lebih dari sensor suhu linier yang dikalibrasi dalam °Kelvin. LM35 tidak memerlukan kalibrasi eksternal untuk memberikan akurasi  $\pm \frac{1}{4}^{\circ}\text{C}$  pada suhu ruangan dan  $\pm \frac{3}{4}^{\circ}\text{C}$  untuk rentang suhu -55 sampai 150 °C. LM35 ini dapat digunakan dengan sumber tegangan tunggal, atau dengan sumber tegangan positif dan negative. LM35 tersedia dalam bentuk hermetis TO-46 paket transistor, sedangkan LM35C, LM35CA, dan LM35D juga





telah terbiasa dengan mikrokontroler seri AT89S51/52, dan berencana untuk beralih ke mikrokontroler AVR, maka secara teknis tidak akan banyak kesulitan yang berarti, hal ini dikarenakan selain konsep dan istilah-istilah dasarnya hampir sama, pemrograman level *assembler*-nya pun relative tidak jauh berbeda.

Berdasarkan arsitekturnya, AVR merupakan mikrokontroler RISC (*Reduce Instruction Set Computer*) dengan lebar bus data 8 bit. Berbeda dengan sistem AT89S51/52 yang memiliki frekuensi kerja seperduabelas kali frekuensi *oscilator*, frekuensi kerja mikrokontroler ini pada dasarnya sama dengan frekuensi *oscilator*, sehingga hal tersebut menyebabkan kecepatan kerja AVR untuk frekuensi *oscilator* yang sama, akan dua belas kali lebih cepat dibandingkan dengan mikrokontroler keluarga AT89S51/52. Struktur mikrokontroler ATmega32 ditunjukkan pada Gambar 2.11



**Gambar 2.11.** Mikrokontroler ATmega32  
(Sumber : atmel.com)

Dengan instruksi yang sangat variatif (mirip dengan sistem CISC-*Complex Instruction Set Computer*) serta jumlah register serbaguna (*General Purpose Register*) sebanyak 32 buah yang semuanya terhubung secara langsung ke ALU (*Aritmetic Logic Unit*), kecepatan operasi mikrokontroler AVR ini dapat mencapai 16 MIPS (enam belas juta instruksi per detik) sebuah kecepatan yang sangat tinggi untuk ukuran mikrokontroler 8 bit yang ada di pasaran sampai saat ini.

Untuk memenuhi kebutuhan dan aplikasi industri yang sangat beragam, mikrokontroler keluarga AVR ini muncul di pasaran dengan tiga seri utama : tinyAVR, ClassicAVR (AVR), megaAVR. Berikut ini beberapa seri yang dapat anda jumpai di pasaran :

ATtiny13

AT90S2313

ATmega103

ATtiny22

AT90S2323

ATmega128

ATtiny22L	AT90S2333	ATmega16
ATtiny2313	AT90S4414	ATmega162
ATtiny2313V	AT90S4433	ATmega168

Keseluruhan seri AVR ini pada dasarnya memiliki organisasi memori dan set instruksi yang sama (sehingga dengan demikian jika telah mahir menggunakan salah satu seri AVR untuk beralih ke seri yang lain akan relatif mudah). Perbedaan antara tinyAVR, AVR dan megaAVR pada kenyataannya hanya merefleksikan tambahan-tambahan fitur yang ditawarkannya saja (misal adanya tambahan ADC internal pada seri AVR tertentu, jumlah port I/O serta memori yang berbeda, dan sebagainya). Diantara ketiganya, megaAVR umumnya memiliki fitur yang paling lengkap, disusul oleh AVR, dan terakhir tinyAVR.

Untuk memberi gambaran yang lebih jelas, pada tabel di bawah memperlihatkan perbedaan ketiga seri AVR ditinjau dari jumlah memori yang dimilikinya.

**Tabel 2.1.** Seri AVR

Mikrokontroler AVR		Memori (byte)		
Jenis	Paket IC	Flash	EEPROM	SRAM
TinyAVR	8 – 32 pin	1 – 2k	64 – 128	0 – 128
AVR (classic AVR)	20 – 44 pin	1 – 8k	128 – 512	0 – 1k
MegaAVR	32 – 64 pin	8 – 128k	512 – 4k	512 – 4k

Seperti terlihat pada tabel tersebut, semua jenis AVR ini telah dilengkapi dengan memori flash sebagai memori program. Tergantung serinya, kapasitas memori flash yang dimiliki bervariasi dari 1k sampai 128kb. Secara teknis, memori jenis ini dapat diprogram melalui saluran antarmuka yang dikenal dengan nama *Serial Peripheral Interface* (SPI) yang terdapat pada setiap seri AVR tersebut. Dengan menggunakan perangkat lunak *programmer* (*downloader*) yang tepat, pengisian memori *Flash* dengan menggunakan saluran SPI ini dapat dilakukan bahkan ketika chip AVR terpasang pada sistem akhir (*end system*), sehingga dengan demikian pemrogramannya sangat fleksibel dan tidak merepotkan pengguna (secara praktis metode ini dikenal dengan istilah *ISP-In System Programming*-sedangkan perangkat lunaknya dinamakan *In System Programmer*).

Untuk penyimpanan data, mikrokontroler AVR menyediakan dua jenis memori yang berbeda : EEPROM (*Electrically Erasable Programmable Read Only*) dan SRAM (*Static Random Access Memory*). EEPROM umumnya digunakan untuk menyimpan data-data program yang bersifat permanen, sedangkan SRAM digunakan untuk menyimpan data variabel yang dimungkinkan berubah setiap saatnya. Kapasitas simpan data kedua memori ini bervariasi tergantung pada jenis AVR-nya (lihat Tabel 2.2). Untuk seri AVR yang tidak memiliki SRAM, penyimpanan data variabel dapat dilakukan pada register serbaguna yang terdapat pada CPU mikrokontroler tersebut.

Selain seri-seri diatas yang sifatnya lebih umum, perusahaan ATMEL juga memproduksi beberapa jenis mikrokontroler AVR untuk tujuan yang lebih khusus dan terbatas, seperti seri AT86RF401 yang khusus digunakan untuk aplikasi *wireless remote control* dengan menggunakan gelombang radio (RF), seri AT90SC yang khusus digunakan untuk peralatan sistem-sistem keamanan kartu SIM GSM, pembayaran via internet, dan lain sebagainya.

## 2.8. Arsitektur Atmega32

Mikrokontroler AVR mempunyai arsitektur *harvard*, dimana memori untuk data dan untuk program terpisah, bus untuk data dan bus untuk program pun terpisah. Dalam arsitektur AVR, seluruh GPR (*General Purpose Register*) terhubung langsung ke ALU (*Aritmetic and Logic Unit*) processor. Sehingga eksekusi instruksi lebih cepat. Dalam satu siklusclock terdapat dua register yang dapat diakses oleh satu instruksi. Teknik yang digunakan adalah memegang sambil mengerjakan. Hal ini berarti, dua operan dibaca dari satu register, dilakukan eksekusi operasi, dan hasilnya disimpan kembali dalam satu register, semuanya dilakukan hanya dalam satu siklus clock.

Dari 32 register terdapat enam buah register yang dapat digunakan untuk pengalamatan tidak langsung 16-bit sebagai pointer (penunjuk). Register tersebut memiliki nama khusus, yaitu X, Y, Z. Masing-masing terdiri dari sepasang register, seperti : X (R27:26), Y (R29:28), Z (R31:30).

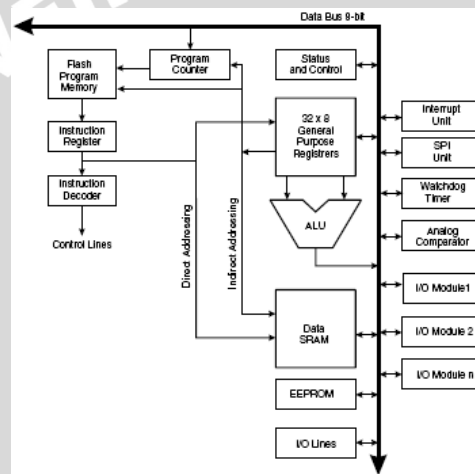
ALU mendukung operasi bit, fungsi aritmatika dan logika antara register dengan register atau antara register dengan nilai konstan, atau hanya operasi satu register.

Untuk kontrol aliran program disediakan instruksi lompatan bersyarat dan tak bersyarat, instruksi CALL (panggil), dapat ditempatkan di seluruh ruangan program.

Kebanyakan instruksi AVR mempunyai format 16-bit *word*. Setiap alamat memori program mengandung instruksi 16 atau 32-bit.

Selama interupsi dan pemanggilan subrutin, alat *program counter* (PC) disimpan kedalam *stack*. *Stack* akan efektif jika diletakkan di SRAM, dan konsekuensinya ukuran stack dibatasi oleh ukuran SRAM. Setiap pengguna harus menginisialisasi SP dalam rutin RESET sebelum subrutin atau interupsi dieksekusi. *Stack pointer* SP dapat ditulis dan dibaca dalam ruangan I/O. SRAM dapat diakses melalui lima mode pengalamatan dalam arsitektur AVR.

AVR ATmega32 mempunyai 8 *channel* ADC dan 4 port I/O, yaitu port A (8 bit), port B (8 bit), port C (8 bit), dan port D (8 bit).



Gambar 2.12. Arsitektur ATmega32  
(Sumber : atmel.com)

## 2.9. Kontrol Logika Fuzzy (KLF)

Fuzzy secara harfiah berarti samar, sedangkan kebalikannya dalam hal ini adalah Crisp yang secara harfiah berarti tegas. Dalam kehidupan sehari-hari nilai samar lebih akrab daripada nilai tegas. Temperatur/suhu tertentu biasa dinyatakan sebagai panas, agak panas, atau sangat dingin daripada dinyatakan dalam nilai terukur tertentu.

Tahun 1965 L.A. Zadeh memodifikasi teori himpunan yang disebut himpunan kabur (Fuzzy Set). Himpunan fuzzy didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sehingga fungsi tersebut akan mencakup bilangan real pada interval  $[0, 1]$ . Nilai keanggotaannya menunjukkan bahwa suatu nilai dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga nilai yang terletak diantaranya. Dengan kata lain nilai

kebenaran suatu hal tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar dan masih ada nilai-nilai yang terletak diantaranya.

Sejak tahun 1985 pengaturan berbasis logika fuzzy mengalami perkembangan pesat, terutama dalam hubungannya dengan penyelesaian masalah kendali yang bersifat tak linier, sulit dimodelkan, berubah karakteristiknya terhadap waktu (time varying) dan kompleks.

### 2.9.1. Struktur Dasar Logika Fuzzy

Dalam sistem pengaturan dengan logika fuzzy dilibatkan suatu blok pengendali yang menerima satu atau lebih masukan dan mengumpangkan satu atau lebih keluaran ke plant atau blok lain. Komponen utama penyusun Fuzzy Logic Controller adalah unit fuzzifikasi, fuzzy inference, basis pengetahuan dan unit defuzzifikasi.

Basis pengetahuan terdiri dari dua jenis (Yan,1994):

#### 1. Basis data

Mendefinisikan parameter fuzzy sebagai bagian dari himpunan fuzzy dengan menentukan batas-batas fungsi keanggotaan pada semesta pembicaraan untuk tiap-tiap variabel.

#### 2. Basis aturan

Memetakan nilai masukan fuzzy menjadi nilai keluaran fuzzy.

### 2.9.2. Fungsi Keanggotaan

Fungsi keanggotaan menotasikan nilai kebenaran anggota-anggota himpunan fuzzy. Interval nilai yang digunakan untuk menentukan fungsi keanggotaan, yaitu nol dan satu. Tiap fungsi keanggotaan memetakan elemen himpunan crisp ke semesta himpunan fuzzy.

Suatu himpunan fuzzy  $A$  dalam semesta pembicaraan  $U$  dinyatakan dengan fungsi keanggotaan,  $\mu_A$  yang harganya berada dalam interval  $[0,1]$  (Kuswadi, 2000:27). Secara matematika hal ini dinyatakan dalam Persamaan (2-1) berikut:

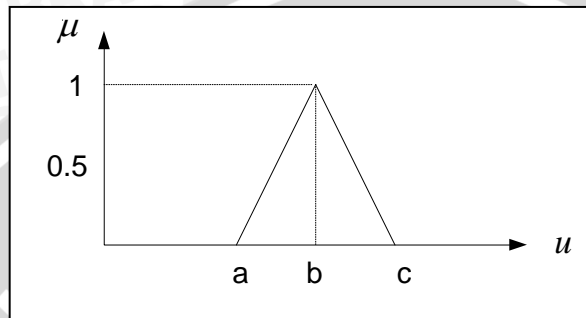
$$\mu_A : U \rightarrow [0,1] \quad (2-1)$$

- Fungsi keanggotaan bentuk Triangular

Definisi fungsi triangular ditunjukkan dalam Persamaan (2-2) berikut:

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (2-2)$$

Fungsi keanggotaan bentuk triangular ditunjukkan dalam Gambar 2.18.



**Gambar 2.13.** Fungsi Keanggotaan Bentuk Triangular

(Sumber: Yan, 1994)

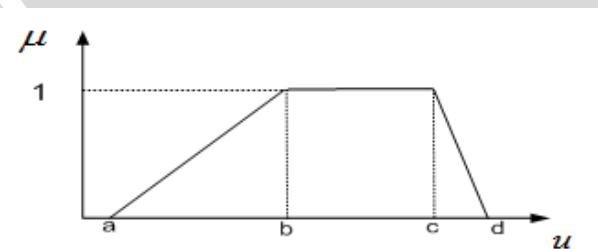
Fungsi keanggotaan bentuk triangular ini digunakan bila diinginkan himpunan fuzzy mempunyai nilai proporsional terhadap nol maupun satu.

- Fungsi keanggotaan bentuk Trapesium

Definisi fungsi trapesium ditunjukkan dalam Persamaan (2-3) berikut:

$$T(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ 1 & b \leq u \leq c \\ \frac{d-u}{d-c} & c \leq u \leq d \\ 0 & d \leq u \end{cases} \quad (2-3)$$

Fungsi keanggotaan bentuk trapesium ditunjukkan dalam Gambar 2.14.



**Gambar 2.14.** Fungsi Keanggotaan Bentuk Trapesium

(Sumber: Yan, 1994)

### 2.9.3. Kontroler Logika Fuzzy

Kontrol Logika Fuzzy (KLF) adalah sistem berbasis aturan (*rule based system*) yang didalamnya terdapat himpunan aturan fuzzy yang mempresentasikan mekanisme pengambilan keputusan. Aturan yang dibuat digunakan untuk memetakan variabel input ke variabel output dengan pernyataan If – Then.

Kontroler ini akan menggunakan data tertentu (crisp) dari sejumlah sensor kemudian mengubahnya menjadi bentuk linguistik atau fungsi keanggotaan melalui proses fuzzifikasi. Lalu dengan aturan fuzzy, inference engine yang akan menentukan hasil keluaran fuzzy. Setelah itu hasil ini akan diubah kembali menjadi bentuk numerik melalui proses defuzzifikasi.

#### 2.9.3.1. Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi variabel fuzzy (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dikuantisasi sebelum diolah oleh pengendali logika fuzzy harus diubah terlebih dahulu ke dalam variabel fuzzy. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai-nilai masukan tersebut menjadi informasi fuzzy yang berguna nantinya untuk proses pengolahan secara fuzzy pula. Proses ini disebut fuzzifikasi (Yan,1994). Proses fuzzifikasi diekspresikan sebagai berikut:

$X$  = fuzzifier ( $x_0$ )

dengan:

$x_0$  = nilai crisp variabel masukan

$x$  = himpunan fuzzy variabel yang terdefinisi

fuzzifier = operator fuzzifikasi yang memetakan himpunan crisp ke himpunan fuzzy

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi, menurut Jun Yan, menggunakan:

1. Himpunan fuzzy dengan distribusi simetris.
2. Gunakan himpunan fuzzy dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (rules).

3. Mengatur himpunan fuzzy agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk segitiga atau trapesium.

### 2.9.3.2. Kaidah Aturan Fuzzy (Fuzzy Rule)

Fuzzy rule adalah bagian yang menggambarkan dinamika suatu sistem terhadap masukan yang dikarakteristikan oleh sekumpulan variabel-variabel linguistik dan berbasis pengetahuan seorang operator ahli. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat.

Dalam pengendali berbasis fuzzy, aturan pengendalian fuzzy berbentuk aturan “IF – THEN”. Untuk sebuah sistem Multi Input Single Output (MISO) basis aturan pengendalian fuzzy berbentuk seperti berikut ini:

Rule 1 IF X is A1 AND Y is B1 THEN Z is C1

Rule 2 IF X is A2 AND Y is B2 THEN Z is C2

·  
·  
·

Rule n IF X is An AND Y is Bn THEN Z is Cn

Dengan X, Y, Z merupakan variabel linguistik, dimana X dan Y merupakan variabel masukan, dan Z merupakan variabel keluaran sistem. An, Bn, dan Cn merupakan nilai linguistik dari X, Y, dan Z (Lee, 1990).

### 2.9.3.3. Metode Inferensi MAX-MIN

Metode inferensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk fuzzy yaitu derajat keanggotaan keluaran.

Pada metode Max–Min aturan operasi minimum Mamdani digunakan untuk implikasi fuzzy yang ditunjukkan dalam Persamaan (2-4) dan Persamaan (2-5). Persamaan aturan minimum adalah:

$$\mu_{C_i} = \bigcup_1^n \alpha_i \wedge \mu_{C_i} \quad (2-4)$$

$$\text{dengan } \alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0) \quad (2-5)$$

Sebagai contoh, terdapat dua baris kaidah atur fuzzy, yaitu:



R1 : Jika x adalah A1 dan y adalah B1 maka z adalah C1

R2 : Jika x adalah A2 dan y adalah B2 maka z adalah C2

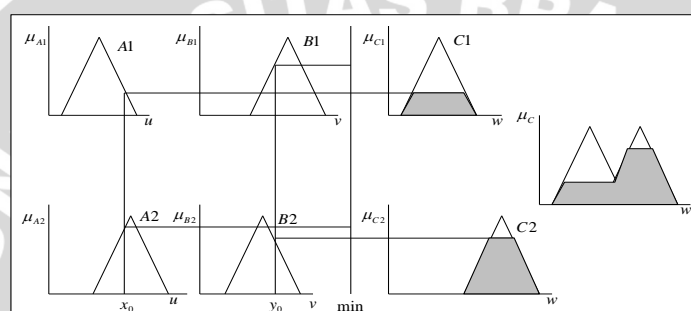
Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen dinyatakan dalam Persamaan (2-6), Persamaan (2-7), dan Persamaan (2-8) berikut:

$$\mu_{C_1}(W) = \mu_{C_1} \vee \mu_{C_2} = [\alpha_1 \wedge \mu_{C_1}(w)] \vee [\alpha_2 \wedge \mu_{C_2}(w)] \quad (2-6)$$

$$\text{dimana } \alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \quad (2-7)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0) \quad (2-8)$$

Lebih jelas metode ini dideskripsikan dalam Gambar 2.15.



**Gambar 2.15.** Inferensi Fuzzy dengan Metode MAX-MIN

Sumber: Yan, 1994

#### 2.9.3.4. Metode Defuzzifikasi Weighted Average (WA)

Defuzzifikasi adalah proses untuk mendapatkan nilai numerik dari data fuzzy yang dihasilkan dari proses inferensi (Yan, 1994). Proses defuzzifikasi dinyatakan dalam Persamaan (2-9) sebagai berikut:

$$y_0 = \text{defuzzifier}(y) \quad (2-9)$$

dengan:

$y$  : aksi kontrol *fuzzy*

$y_0$  : aksi kontrol *crisp*

*defuzzifier* : operator defuzzifikasi

Metode Weighted Average (WA) ini didefinisikan dalam Persamaan (2-10) sebagai berikut:

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \quad (2-10)$$

dengan:

$U$  = Keluaran

$w_i$  = Bobot nilai benar  $w_i$

$u_i$  = Nilai linguistik pada fungsi keanggotaan keluaran

$n$  = Banyak derajat keanggotaan

UNIVERSITAS BRAWIJAYA

