

## BAB II

### TINJAUAN PUSTAKA

Dalam bab ini akan dijelaskan mengenai teori dasar komponen yang digunakan. Adapun dasar teori yang akan dijelaskan meliputi dasar teori tentang suhu udara, kelembaban udara, sensor suhu LM 35, sensor kelembaban HS15P, ADC 0809, mikrokontroler AT89S52, DAC 0808, triac, optotriac dan LCD.

#### 2.1 Suhu Udara

Secara fisis suhu udara dapat didefinisikan sebagai tingkat gerakan molekul benda, semakin cepat gerakan molekul, maka semakin tinggi suhunya. Suhu juga dapat didefinisikan sebagai tingkat panas suatu benda. Panas dapat dinyatakan sebagai energi yang ditransfer dari benda yang satu ke benda yang lain dengan proses termal seperti radiasi, konduksi atau konveksi. Energi ini akan pindah dari benda yang panas ke benda yang lebih dingin dan jika sumber dari luar maupun dalam dianggap tidak ada sampai kedua benda tersebut mempunyai temperatur yang sama, dalam arti sudah tidak ada pertukaran panas lagi.

Untuk menyatakan suhu udara dipakai berbagai skala. Dua skala yang sering dipakai dalam pengukuran suhu udara adalah skala *Fahrenheit* yang dipakai Negara Inggris dan skala celcius atau skala perseratusan (*centigrade*) yang dipakai oleh sebagian besar Negara di dunia.

Dalam skripsi ini skala yang digunakan adalah skala *celcius*. Dalam skala perseratusan (skala *celcius*) ditetapkan titik didih air pada 100 derajat dan titik lebur es pada 0 (nol) derajat. Suhu *Fahrenheit* dapat diubah menjadi celcius dengan persamaan berikut:

$$C = \frac{5}{9}(F - 32) \quad (2-1)$$

Fluktuasi suhu harian sebagai akibat adanya neraca antara radiasi matahari yang diterima dan yang dilepaskan oleh bumi. Sejak matahari terbit sampai kira-kira satu atau dua jam setelah tengah hari jumlah energi yang diterima oleh bumi lebih besar dari pada yang hilang. Oleh karena itu, kurva temperatur terus-

menerus naik. Sebaliknya kira-kira jam 13.00 sampai matahari terbit jumlah energi yang dilepaskan oleh bumi lebih besar dari pada yang diterima. Oleh karena itu, kurva suhu harian turun. (Soekardi, 1983:35)

## 2.2 Kelembaban Udara

Kelembaban udara menyatakan banyaknya uap air dalam udara. Jumlah uap air dalam udara ini sebetulnya hanya merupakan sebagian kecil saja dari seluruh atmosfer. Akan tetapi uap air ini merupakan komponen udara yang sangat penting ditinjau dari segi cuaca dan iklim. Sebagian gas yang menyusun atmosfer yang dekat dengan permukaan laut relatif konstan dari tempat satu ke tempat yang lain. Sedangkan uap air merupakan bagian yang tidak konstan.

Jumlah uap air maksimum yang dapat dikandung oleh udara pada suatu temperatur disebut kapasitas udara. Besar kecilnya kapasitas udara itu tergantung temperatur. Semakin tinggi temperatur maka semakin besar kapasitas udara dan sebaliknya. Kapasitas udara itu tidak selalu dapat dicapai. Jika kapasitas udara tersebut dapat dicapai berarti udara itu jenuh dengan uap air. Kandungan uap air dalam atmosfer dinyatakan dalam beberapa cara:

1. Tekanan uap

Bagian dari tekanan atmosfer yang disebabkan oleh uap air. Dinyatakan dalam ukuran yang sama dengan tekanan udara total. Misalnya: atm, milibar atau cm/mm Hg.

2. Kelembaban spesifik

Berat uap air persatuan berat udara (termasuk berat uap airnya). Biasanya dinyatakan dalam gram tiap air per kg udara. Kelembaban spesifik hampir sama dengan tekanan uap.

3. Kelembaban absolut

Berat uap air persatuan volume udara. Misalnya  $\text{gram/m}^3$  udara. Ini kurang digunakan dalam meteorologi oleh karena volume berubah-ubah jika udara naik. Yang berarti kelembaban absolut juga berubah-ubah.

4. Kelembaban relatif

perbandingan antara uap air yang betul-betul ada di udara dengan jumlah uap air dalam udara tersebut jika pada temperatur dan tekanan yang sama udara

tersebut jenuh dengan uap air. Jika kelembaban relatif mencapai harga 100% (= 1) berarti udara itu jenuh dengan uap air.

Misalnya udara pada temperatur 34°C untuk mencapai kejenuhan harus ada 8 gram uap air dan ternyata hanya mengandung 6 gram, berarti kelembaban relatifnya =  $6/8 \times 100\% = 75\%$ . Kelembaban relatif dapat pula dirumuskan sebagai: (Soekardi, 1983:51)

$$\frac{\text{kelembaban absolut}}{\text{kapasitas udara}} \text{ atau } \frac{\text{kelembaban spesifik}}{\text{kapasitas udara}}$$

Dengan syarat satuan yang digunakan harus sama. Kelembaban relatif dapat berubah baik dengan mengubah jumlah uap air atau mengubah kapasitas udara.

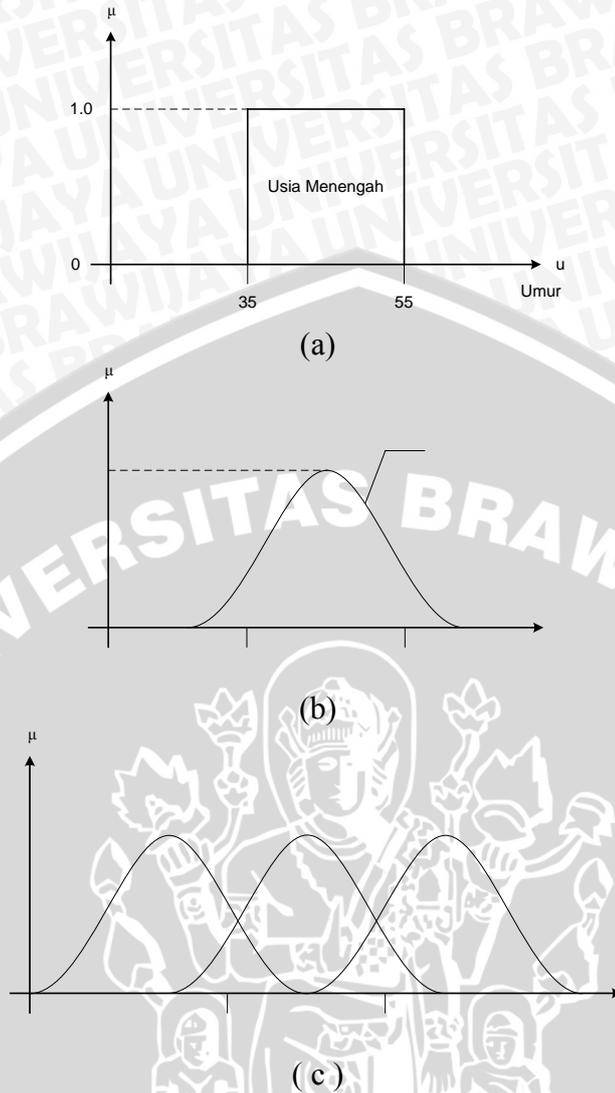
Dalam skripsi ini kelembabannya dinyatakan dalam bentuk kelembaban relatif. Karena suhu dan kelembaban masih merupakan besaran nonelektrik maka dibutuhkan sebuah sensor untuk mengubah dari besaran nonelektrik menjadi besaran elektrik. Untuk memudahkan dalam mengendalikan suhu dan kelembaban maka dibutuhkan suatu mikrokontroler yang digunakan untuk mengolah data dari sensor.

Ada beberapa metode yang dapat digunakan dalam mengendalikan suatu sistem. Metode pengendalian yang digunakan dalam skripsi ini yaitu kontrol logika fuzzy, hal ini disebabkan karena penggunaan dan pengendaliannya lebih mudah.

## 2.3 Logika Fuzzy

### 2.3.1 Himpunan *Crisp* dan Himpunan *Fuzzy*

Sifat benar atau salah pada logika tradisional tidak memperhitungkan nilai tengah (antara benar dan salah) yang ada dalam dunia nyata. Sebaliknya logika *fuzzy* memiliki nilai yang banyak, logika *fuzzy* tidak hanya mengenal sesuatu secara utuh (100%) benar atau (100%) salah, tetapi juga mengenal derajat kebenaran atau derajat keanggotaan. Sebagai contoh perbedaan konsep antara himpunan *crisp* dengan himpunan *fuzzy* diperlihatkan dalam Gambar 2.1 berikut:



**Gambar 2.1** (a)Himpunan *Crisp*, (b)Himpunan *Fuzzy*, (c)Himpunan *Fuzzy* Untuk Kelompok Usia  
 Sumber: Yan, 1994:3

Pada himpunan *crisp*, peralihan dari satu fungsi keanggotaan ke fungsi keanggotaan lain terjadi secara mendadak, seperti ditunjukkan pada Gambar 2.1(a), usia 55 dan 56 digolongkan pada kelompok usia yang sangat berbeda. Berbeda dengan himpunan *fuzzy*, peralihan ini tidak terjadi secara mendadak (Gambar 2.1(c)), usia 60 masuk dalam dua kelompok usia yaitu usia ‘menengah’ dan ‘tua’ dengan adanya *overlap* antara dua kelompok usia, saat nilai keanggotaan dalam kelompok ‘menengah’ turun menuju 0, nilai dalam kelompok ‘tua’ meningkat secara bertahap menuju nilai 1.

### 2.3.2 Himpunan Fuzzy

Misalkan  $U$  adalah kumpulan obyek yang ditunjukkan oleh  $\{u\}$ , yang dapat berupa diskrit atau kontinyu.  $U$  disebut himpunan semesta sedangkan  $u$  mewakili elemen umum dari  $U$ . suatu himpunan fuzzy  $F$  dalam himpunan semesta  $U$  dicirikan dengan suatu fungsi keanggotaan  $\mu_F$  yang mempunyai nilai dalam interval  $[0,1]$ .

$$\mu_F : U \rightarrow [0,1] \tag{2-3}$$

sebagaimana diperlihatkan dalam Gambar 2.2 berikut ini.



**Gambar 2.2** Himpunan Fuzzy dan Fungsi Keanggotaannya  
 Sumber: Yan, 1994:16

Suatu himpunan fuzzy  $F$  dalam  $U$  biasanya dituliskan sebagai suatu himpunan pasangan elemen  $u$  dan nilai tingkat keanggotaannya yang berurutan:

$$F = \{(u, \mu_F(u)) | u \in U\} \tag{2-4}$$

Bila  $U$  kontinyu, himpunan fuzzy  $F$  dapat dituliskan secara singkat sebagai:

$$F = \int_u \mu_F(u)/u \tag{2-5}$$

Bila  $U$  diskrit, himpunan fuzzy  $F$  ditunjukkan sebagai: **1.0**

$$F = \sum_{i=1}^n \mu_F(u_i)/u_i \tag{2-6}$$

atau

$$F = \mu_F(u_1)/u_1 + \mu_F(u_2)/u_2 + \dots + \mu_F(u_i)/u_i + \dots + \mu_F(u_N)/u_N \tag{2-7}$$



### 2.3.3 Operasi-operasi Himpunan *Fuzzy*

Operasi himpunan *fuzzy* dilakukan dengan memanipulasi fungsi keanggotaannya. Pada bagian ini beberapa operasi himpunan *fuzzy* dasar akan diringkas. Misalkan A dan B adalah dua himpunan *fuzzy* dalam U dengan fungsi keanggotaannya berturut-turut  $\mu_A$  dan  $\mu_B$  maka operasi himpunan *fuzzy* berikut dapat didefinisikan:

- Kesamaan

$$A = B \rightarrow \mu_A(u) = \mu_B(u) \quad ,u \in U \quad (2-8)$$

- Gabungan

$$A \cup B \rightarrow \mu_{A \cup B}(u) = \max \{ \mu_A(u), \mu_B(u) \} \quad ,u \in U \quad (2-9)$$

- Perpotongan

$$A \cap B \rightarrow \mu_{A \cap B}(u) = \min \{ \mu_A(u), \mu_B(u) \} \quad ,u \in U \quad (2-10)$$

- Komplemen

$$A = \bar{B} \rightarrow \mu_A(u) = \mu_{\bar{B}}(u) = 1 - \mu_B(u) \quad ,u \in U \quad (2-11)$$

- Normalisasi

$$\text{NORM}(A) \rightarrow \mu_{\text{NORM}(A)}(u) = \mu_A(u) / \max(\mu_A(u)) \quad ,u \in U \quad (2-12)$$

- Konsentrasi

$$\text{CON}(A) \rightarrow \mu_{\text{CON}(A)}(u) = (\mu_A(u))^2 \quad ,u \in U \quad (2-13)$$

- Dilasi

$$\text{DIL}(A) \rightarrow \mu_{\text{DIL}(A)}(u) = (\mu_A(u))^{0,5} \quad ,u \in U \quad (2-14)$$

- Intensifikasi

$$\text{INT}(A) \rightarrow \mu_{\text{INT}(A)}(u) = \begin{cases} 2(\mu_A(u))^2, & 0 \leq \mu_A(u) \leq 0,5 \\ 1 - 2(1 - \mu_A(u))^2, & 0,5 \leq \mu_A(u) \leq 1 \end{cases} \quad (2-15)$$

- Perkalian aljabar

$$A \cdot B \rightarrow \mu_{A \cdot B}(u) = \mu_A(u) \cdot \mu_B(u) \quad ,u \in U \quad (2-16)$$

- Penjumlahan terbatas

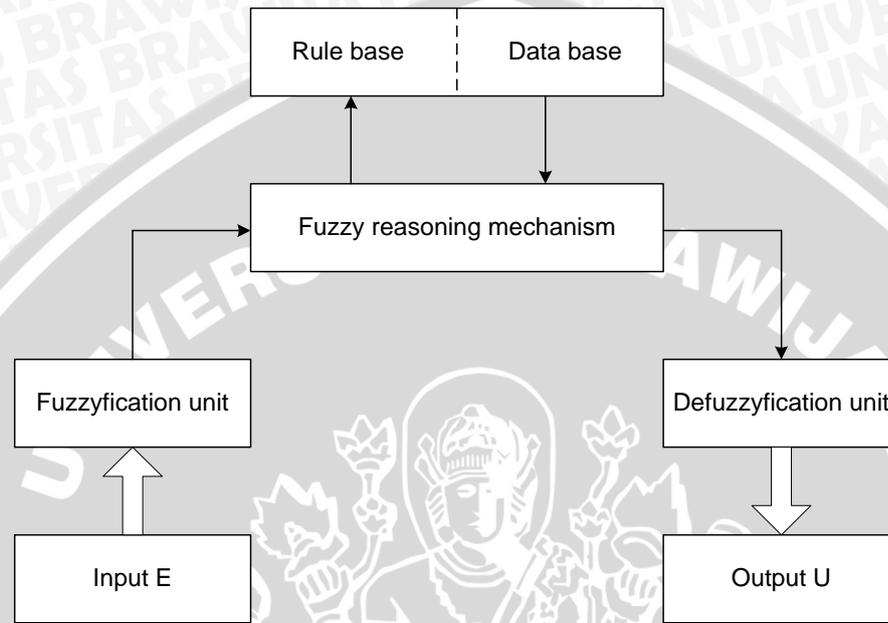
$$A \oplus B \rightarrow \mu_{A \oplus B}(u) = \min \{ 1, \mu_A(u) + \mu_B(u) \} \quad ,u \in U \quad (2-17)$$

- Perkalian drastis

$$A \otimes B \rightarrow \mu_{A \otimes B}(u) = \begin{cases} \mu_A(u), \mu_B(u) = 1 \\ \mu_B(u), \mu_A(u) = 1 \\ 0, \mu_A(u), \mu_B(u) < 1 \end{cases} \quad ,u \in U \quad (2-18)$$

### 2.3.4 Kontrol Logika *Fuzzy*

Kontrol logika *fuzzy* adalah suatu metodologi kontrol digital yang mensimulasikan jalan pikiran manusia terhadap sistem fisik dan strategi kontrol yang diperlukan secara alami dan logis (Yan, 1994:6). Struktur dasar KLF diperlihatkan dalam Gambar 2.3 berikut ini:



**Gambar 2.3** Struktur Dasar KLF  
Sumber: Yan, 1994:47

Variabel sistem ada dua jenis utama, variabel masukan (E) yang diukur dari proses yang dikontrol dan variabel keluaran (U) yang digunakan oleh KLF untuk mengontrol proses. Untuk masing-masing variabel sistem yang digunakan dalam mengekspresikan aturan-aturan, nilai-nilai yang diijinkan harus dinyatakan sebagai himpunan *fuzzy* dalam himpunan semesta yang sesuai. Penentuan nilai-nilai dalam himpunan *fuzzy* ini merupakan salah satu langkah paling kritis dalam proses perancangan dan sangat mempengaruhi performansi sistem.

Dalam Gambar 2.3 diperlihatkan bahwa elemen utama KLF adalah sebuah unit fuzzyfikasi, unit penyimpulan *fuzzy*, *basis* pengetahuan dan unit defuzzyfikasi. Nilai actual yang diperoleh dari atau dikirim ke proses yang dikontrol berupa nilai *crisp*. Operasi fuzzyfikasi dan defuzzyfikasi dibutuhkan untuk memetakan nilai tersebut ke dan dari nilai *fuzzy* yang digunakan secara internal oleh KLF.

Basis pengetahuan terdiri dari dua jenis informasi utama, yaitu:

- a. Basis data yang mendefinisikan fungsi keanggotaan himpunan *fuzzy* yang digunakan sebagai nilai untuk masing-masing variabel sistem.
- b. Basis aturan, yang memetakan nilai-nilai *fuzzy* masukan ke nilai-nilai *fuzzy* keluaran.

Modul penyimpulan *fuzzy* menggunakan logika *fuzzy* mirip dengan beberapa aspek pembuatan keputusan manusia. Modul ini melakukan penyimpulan *fuzzy* untuk tiba pada aksi kontrol *fuzzy* dengan mengevaluasi basis pengetahuan untuk masukan terfuzzifikasi.

Untuk keluaran, proses defuzzifikasi lebih rumit dan dilakukan sebagai bagian tingkat akhir penyimpulan *fuzzy*. Proses ini meliputi pengkombinasian sejumlah himpunan *fuzzy* (yang dihasilkan dari proses penyimpulan *fuzzy*) dan *weighting* dalam suatu perhitungan yang memberikan suatu nilai *crisp* tunggal untuk masing-masing variabel keluaran.

#### 2.3.4.1 Fuzzyfikasi

Fuzzyfikasi adalah proses pemetaan masukan-masukan yang diamati ke himpunan *fuzzy* dalam himpunan semesta masukan yang bervariasi. Dalam pengendalian proses, data yang diamati berupa *crisp* dan fuzzyfikasi diperlukan untuk memetakan range masukan *crisp* teramati ke nilai *fuzzy* yang sesuai untuk variabel masukan sistem. Data yang terpetakan selanjutnya dikonversi menjadi istilah linguistik yang sesuai sebagai label dari himpunan *fuzzy* yang didefinisikan untuk variabel masukan sistem. Proses ini bisa diekspresikan oleh

$$x = \text{fuzzifier}(x_0) \quad (2-19)$$

Dimana  $x_0$  adalah nilai *crisp* variabel masukan dari proses,  $x$  adalah himpunan *fuzzy* yang didefinisikan untuk variabel tersebut dan fuzzifier adalah operator fuzzyfikasi yang mempunyai efek mentransformasikan data *crisp* ke himpunan *fuzzy*.

Secara ringkas strategi fuzzyfikasi dasar meliputi:

1. Mendapatkan nilai *crisp* dari variabel masukan.
2. Memetakan nilai *crisp* dari variabel masukan ke dalam himpunan semesta yang sesuai.

3. Mengkonversi data yang terpetakan menjadi *fuzzy singleton* atau mengkonversinya menjadi istilah linguistic yang sesuai sebagai label dari himpunan *fuzzy* yang didefinisikan untuk variabel itu.

#### 2.3.4.2 Basis Pengetahuan

Basis pengetahuan terdiri dari sebuah basis data dan sebuah basis aturan. Basis data menyediakan definisi yang perlu dari parameter-parameter *fuzzy* sebagai himpunan *fuzzy* dengan fungsi keanggotaan yang didefinisikan pada himpunan semesta untuk masing-masing variabel. Penyusunan *basis* data melibatkan pendefinisian himpunan semesta untuk, masing-masing variabel, penentuan jumlah himpunan *fuzzy* serta perancangan fungsi keanggotaannya.

Basis aturan terdiri dari aturan-aturan kontrol *fuzzy* yang menyatakan hubungan pengontrolan, yang ditunjukkan untuk mencapai sasaran pengontrolaan. Aturan-aturan kontrol *fuzzy* disusun berdasarkan tujuan dan pengalaman pengontrolan yang diberikan oleh orang yang ahli pada bidang tersebut. Tidak ada struktur standar yang formal tentang aturan-aturan *fuzzy* ini. Aturan-aturan *fuzzy* ini biasa dinyatakan dalam bentuk ‘IF...THEN...’ misalnya untuk suatu sistem *multi input multi output* (MIMO), basis aturan dapat ditunjukkan oleh:

Aturan 1 IF  $x_1 = A_{11}$  OR...AND  $x_m = A_{1m}$  THEN  $y_1 = B_{11}$  ALSO... $y_n = B_{1n}$

Aturan 2 IF  $x_1 = A_{21}$  OR...AND  $x_m = A_{2m}$  THEN  $y_1 = B_{21}$  ALSO... $y_n = B_{2n}$

⋮

Aturan N IF  $x_1 = A_{N1}$  OR...AND  $x_m = A_{Nm}$  THEN  $y_1 = B_{N1}$  ALSO... $y_n = B_{Nn}$

IF  $x_1 = A_{N1}$  OR...AND  $x_m = A_{Nm}$  disebut *antecedent*

$y_1 = B_{N1}$  ALSO... $y_n = B_{Nn}$  disebut *consequent*

Dimana  $x$  adalah variabel masukan sistem,  $A$  adalah subhimpunan *fuzzy* untuk  $x$ ,  $y$  adalah variabel keluaran sistem,  $B$  adalah subhimpunan *fuzzy* untuk  $y$ ,  $N$  adalah jumlah aturan,  $m$  adalah jumlah variabel masukan,  $n$  adalah variabel keluaran. Ada beberapa kata sambung yaitu *AND*, *OR* dan *ALSO* yang digunakan. *AND* adalah *operator* perpotongan, *OR* adalah *operator* gabungan dan *ALSO* menunjukkan adanya keluaran yang lebih dari satu dalam aturan *fuzzy*.

### 2.3.4.3 Penyimpulan Fuzzy

Penyimpulan *fuzzy* adalah proses pengevaluasian basis pengetahuan untuk masukan-masukan terfuzzyfikasi yang akan menghasilkan aksi kontrol *fuzzy*. Ada beberapa cara dimana nilai masukan yang akan diamati dapat digunakan untuk mendapatkan aturan-aturan yang seharusnya dipakai dan aksi kontrol *fuzzy* yang tepat. Diantara bermacam-macam metode penyimpulan *fuzzy* yang paling umum digunakan di industri sekarang adalah metode penyimpulan *fuzzy* *MAX-MIN* yang akan diperlihatkan sebagai berikut:

Diasumsikan suatu basis aturan kontrol *fuzzy* dengan hanya dua aturan:

Aturan 1       $IF\ x = A_1\ AND\ y = B_1\ THEN\ z = C_1$

Aturan 2       $IF\ x = A_2\ AND\ y = B_2\ THEN\ z = C_2$

Dari aturan-aturan tersebut diperoleh nilai derajat kebenaran (*fire strength*). Saat *antecedent* dihubungkan melalui operator “AND”, *fire strength* mengasumsikan nilai *strength* terkecil dari *antecedent*. Nilai minimum ini yang akan menjadi nilai kebenaran bagi aturan tersebut.

Misalkan *fire strength* dari aturan ke-1 ditunjukkan  $\alpha_1$ . untuk masukan  $x_0$  dan  $y_0$ , *fire strength*  $\alpha_1$  dan  $\alpha_2$  dari basis aturan dapat ditunjukkan oleh:

$$\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \quad (2-20)$$

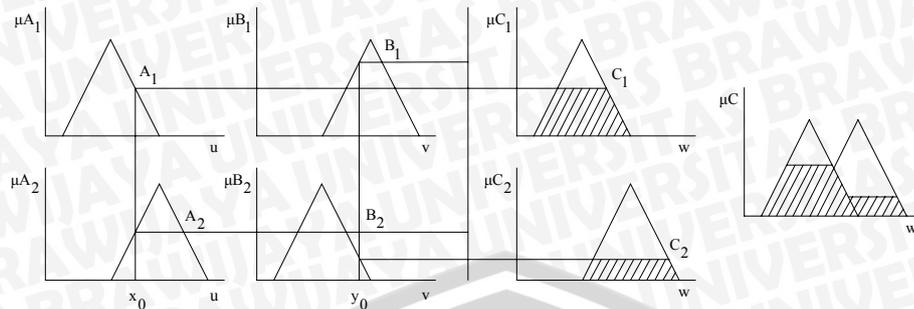
$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0) \quad (2-21)$$

Kontribusi kontrol untuk masing-masing aturan kontrol dihitung berdasarkan pada *fire strength*  $\alpha$  dan bagian *consequent* dari aturan tersebut.

Dalam penyimpulan *fuzzy* *MAX-MIN*, keputusan kontrol yang ditimbulkan oleh aturan ke-1 dapat diekspresikan oleh  $\alpha_1 \wedge \mu_{C_1}(W)$  ( $\wedge$  menyatakan *min* dan sebaliknya). Sehingga keanggotaan dari *consequent* tersimpul C diberikan oleh

$$\mu_C(W) = (\alpha_1 \wedge \mu_{C_1}(W)) \vee (\alpha_2 \wedge \mu_{C_2}(W)) \quad (2-22)$$

Proses penyimpulan *MAX-MIN* untuk nilai masukan *crisp*  $x_0$  dan  $y_0$  yang telah dianggap sebagai *fuzzy singleton* diperlihatkan dalam Gambar 2.4 berikut:



**Gambar 2.4** Penyimpulan *Fuzzy* MAX-MIN dengan Masukan *Crisp*  
 Sumber: Yan, 1994:55

Secara ringkas penyimpulan *fuzzy* merupakan inti dari suatu KLF yang mempunyai kemampuan untuk mensimulasikan pembuatan keputusan manusia yang didasarkan pada konsep *fuzzy* dan menyimpulkan aksi kontrol *fuzzy* menggunakan aturan-aturan penyimpulan dalam logika *fuzzy*.

### 2.3.4.4 Defuzzyfikasi

Defuzzyfikasi adalah proses pemetaan dari suatu aksi kontrol *fuzzy* tersimpul ke suatu aksi kontrol *non fuzzy* (*crisp*). Strategi defuzzyfikasi bertujuan untuk menghasilkan aksi kontrol *non fuzzy* yang paling baik mewakili distribusi kemungkinan dari aksi kontrol *fuzzy* tersimpul. Ini dapat dinyatakan oleh:

$$y_0 = \text{defuzzyfier}(y) \quad (2-23)$$

$y$  adalah aksi kontrol *fuzzy*,  $y_0$  adalah aksi kontrol *crisp* dan defuzzyfier adalah operator defuzzyfikasi.

Dalam implementasi *real time*, strategi defuzzyfikasi yang paling umum digunakan adalah metode *center of area* (COA). Strategi COA adalah dengan mendapatkan pusat gravitasi dari distribusi kemungkinan aksi kontrol. Ini secara luas digunakan dalam implementasi kontrol logika *fuzzy* saat ini. Untuk sederhananya, misalkan suatu sistem *fuzzy* MISO (*Multi Input Single Output*). Jumlah aturan ditunjukkan oleh  $n$ , momen (sekeliling sumbu nol sepanjang himpunan semesta keluaran) fungsi keanggotaan dari himpunan *fuzzy* yang didefinisikan untuk kontrol keluaran aturan ke- $i$  ditunjukkan oleh  $M_i$  dan luasnya ditunjukkan oleh  $A_i$ , serta *fire strength* dari aturan ke- $i$  ditunjukkan oleh  $\alpha_i$ . maka nilai kontrol *crisp*  $W$  yang didefuzzyfikasi menggunakan metode COA diberikan oleh:

$$W = \frac{\sum_{i=1}^n \alpha_i M_i}{\sum_{i=1}^n \alpha_i A_i} \quad (2-24)$$

Secara ringkas antar muka defuzzyfikasi meliputi:

1. Pemetaan nilai variabel keluaran kedalam himpunan semesta yang sesuai.
2. Menghasilkan suatu aksi kontrol *non fuzzy* dari suatu aksi kontrol *fuzzy* yang tersimpulkan.

Untuk mengolah data dari sensor suhu dan kelembaban maka dibutuhkan suatu rangkaian yang programabel. Sebagai pengolah data dalam skripsi ini menggunakan mikrokontroler, hal ini disebabkan karena penggunaannya lebih mudah dan harganya relatif murah. Selain itu juga bentuknya kecil dan banyak tersedia dipasaran. Mikrokontroler yang digunakan dalam alat ini yaitu AT89S52.

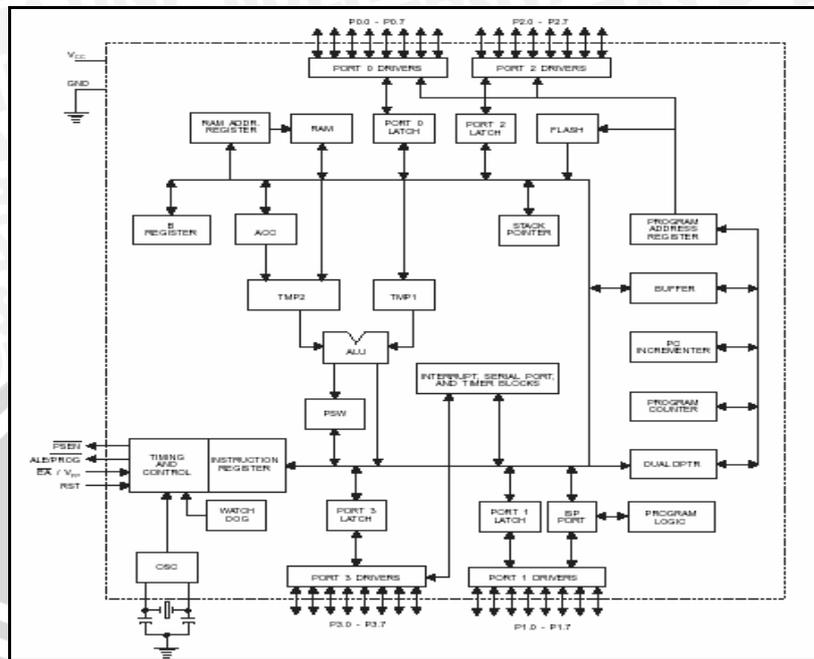
#### 2.4 Mikrokontroler AT89S52

Mikrokontroler di dalam alat ini merupakan komponen utama, karena komponen inilah yang akan mengontrol keseluruhan sistem agar dapat bekerja dengan baik dan optimal. Arsitektur dari MCU AT89S52 dapat dilihat dalam Gambar 2.5.

Secara umum arsitektur yang dimiliki oleh mikrokontroler AT89S52 adalah sebagai berikut:

- 8 bit CPU dengan *register A (accumulator)* dan *match register (register B)*.
- 16 bit *Program Counter (PC)* dan *Data Pointer (DPTR) register*.
- 8 bit *Program Status Word (PSW) register*, 8 bit *Stack Pointer*.
- Internal ROM dan EPROM dengan kapasitas 8 *kbyte*.
- Internal RAM dengan kapasitas 128 *byte* yang digunakan untuk 4 buah *register bank*, yang masing-masing terdiri dari 8 *register*, 16 *byte*, yang mana dapat dieksekusi pada masing-masing bit secara *independent (Bit Addesable)* dan sebagai memori *variable* 8 bit.
- 32 input/output yang disusun pada 4 port (port 0 – port 3).
- 2 buah 16 bit *timer/counter*: T0 dan T1.
- *Full Duplex Serial Data Communication*: SBUF.
- *Control Register* :TCON, TMOD, PCON, IP dan IE.

- 2 eksternal interrupt dan 3 internal *interrupt*.
- Oscilator dan *Clock Circuit*.



**Gambar 2.5** Blok Diagram AT89S52

Sumber: Anonymous, 2004: 3

### 2.4.1 Konfigurasi Pin

Masing-masing kaki atau pin dalam mikrokontroler AT89S52 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler AT89S52, akan lebih mudah merencanakan dan membuat sistem aplikasi mikrokontroler AT89S52. AT89S52 mempunyai 40 pin, susunan masing-masing pin dapat dilihat dalam Gambar 2.6.

Fungsi kaki-kaki AT89S52 adalah:

- *Port 1* (Pin 1..8), berfungsi sebagai *port I/O* biasa.
- Pin 9 (RST), pulsa transisi dari rendah ke tinggi yang diumpankan ke pin RST akan mereset AT89S52. Pin ini dihubungkan dengan rangkaian *power on reset*.
- *Port 3* (Pin 10..17), *port* paralel 8 bit dua arah yang memiliki fungsi pengganti. Fungsi pengganti meliputi TXD (*Transmit Data*), RXD (*Receive Data*), INT0 (*Interrupt 0*), INT1 (*Interrupt 1*), T0 (*Timer 0*), T1 (*Timer 1*), WR (*Write*), RD (*Read*). Apabila fungsi pengganti tidak digunakan, pin-pin ini dapat digunakan sebagai *port I/O* biasa.

- Pin 18 (XTAL1), merupakan pin masukan ke rangkaian osilator internal. Osilator kristal dan sumber osilator luar dapat digunakan.
- Pin 19 (XTAL2), merupakan pin masukan ke rangkaian osilator internal. Pin ini dipakai bila menggunakan osilator kristal.
- Pin 20 (*Ground*), dihubungkan ke  $V_{SS}$  atau *ground*.
- *Port* 2 (Pin 21..28), *port* paralel 8 bit dua arah, dapat digunakan sebagai *port* I/O 8 bit biasa dan digunakan untuk mengirim *byte* alamat bila digunakan untuk mengakses memori eksternal.
- Pin 29 (*PSEN/Program Store Enable*), merupakan pengontrol yang digunakan untuk mengakses program memori eksternal masuk ke dalam bus selama proses pemberian/pengambilan instruksi.
- Pin 30 (ALE), digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
- Pin 31 (EA), jika diberi logika tinggi, maka mikrokontroler akan melaksanakan instruksi dari ROM/EPROM. Bila diberikan logika rendah, mikrokontroler akan melaksanakan instruksi dari memori program luar.
- *Port* 0 (Pin 32..39), merupakan *port* paralel 8 bit open drain dua arah. *Port* 0 dapat digunakan sebagai *port* I/O biasa dan dapat juga digunakan untuk memultiplek alamat dengan data pada waktu mengakses memori eksternal.
- Pin 40 ( $V_{CC}$ ), dihubungkan ke  $V_{CC}$  (+5 volt).

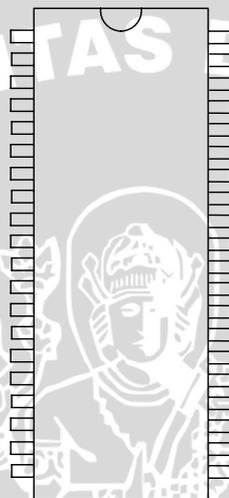
#### 2.4.2 Struktur dan Operasi *Port*

Mikrokontroler AT89S52 memiliki 3 buah *port*. Setiap *port* memiliki 8 buah jalur I/O yang bersifat *bidirectional*. Beberapa karakteristik *port* mikrokontroler AT89S52 dijelaskan secara singkat berikut ini:

- *Port* 0 dapat digunakan sebagai *port* I/O biasa, pada fungsi sebagai *port* I/O biasa *port* ini dapat memberikan *output sink* ke delapan buah TTL *input* atau dapat diubah sebagai *input* dengan memberikan logika 1 pada *port* tersebut.
- *Port* 1 merupakan *port* I/O biasa. *Port* ini mempunyai internal *pull-up* dan berfungsi sebagai *output* dengan memberikan logika 1. Sebagai *output port* ini dapat memberikan *output sink* ke empat buah *input* TTL. *Port* 1

memiliki kemampuan untuk menyangga beban sampai dengan 20 mA dan dapat mengemudikan LED secara langsung.

- Port 2 merupakan port I/O biasa. Port ini mempunyai internal pull-up dan berfungsi sebagai output dengan memberikan logika 1. Sebagai output port ini dapat memberikan output sink ke empat buah input TTL.
- Port 3, sebagai I/O biasa port 3 mempunyai sifat yang sama dengan port 1 maupun port 2. Sedangkan sebagai fungsi khusus port-port ini mempunyai penjelasan seperti yang terlihat dalam Tabel 2.1.



**Gambar 2.6** Susunan Pin AT89S52  
 Sumber: Anonymous, 2004: 2

**Tabel 2.1** Fungsi Khusus Port 3  
 Sumber: Anonymous, 2004: 4

Jalur	Fungsi Alternatif		
		(T2) P1.0	1
		(T2 EX) P1.1	2
P3.0	Input data serial RXD	P1.2	3
P3.1	Output data serial TXD	P1.3	4
P3.2	Input Interupsi Eksternal INT0	P1.4	5
P3.3	Input Interupsi Eksternal INT1	(MOSI) P1.5	6
P3.4	Input Counter T0	(MISO) P1.6	7
P3.5	Input Counter T1	(SCK) P1.7	8
P3.6	External Data Memory Write Strobe	RST	9
P3.7	External Data Memory Read Strobe	(RXD) P3.0	10
		(TXD) P3.1	11
		(INT0) P3.2	12
		(INT1) P3.3	13
		(T0) P3.4	14
		(T1) P3.5	15
		(WR) P3.6	16
		(RD) P3.7	17
		XTAL2	18

### 2.4.3 Register Fungsi Khusus

*Register* fungsi khusus (*Special Function Register*) terletak pada 128 *byte* bagian atas memori data internal. Wilayah SFR ini terletak pada alamat  $80_H$  sampai  $FF_H$ . *Register-register* ini hanya dapat diakses dengan pengalamatan langsung, baik per bit maupun per *byte*.

SFR berisi *register-register* dengan fungsi tertentu. Masing-masing *register* seperti ditunjukkan dalam Tabel 2.1 yang meliputi simbol, nama, dan alamatnya serta keadaannya dalam nilai biner pada saat terjadi *power-on reset*.

Beberapa kegunaan *register* fungsi khusus yang penting dijelaskan sebagai berikut:

- *Accumulator* (ACC); merupakan *register* untuk penambahan dan pengurangan. Akumulator juga disebut sebagai *register A*.
- *Register B*; merupakan *register* yang berfungsi untuk melayani operasi perkalian dan pembagian.
- PSW; terdiri dari beberapa bit status yang menggambarkan kejadian di akumulator sebelumnya, yang pengkondisian keadaan akumulator tersebut melalui *flag register* yang terdiri dari *carry flag*, *auxiliary carry flag*, *parity flag*, *overflow flag*, dua bit pemilih *bank*, dan dua *flag* yang dapat didefinisikan sendiri oleh pemakai.
- *Control Register*; terdiri atas *register* yang mempunyai fungsi kontrol. Terdapat dua *register* khusus untuk mengontrol sistem interupsi, yaitu *register IP* (*Interrupt Priority*) dan *register IE* (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/counter* terdapat *register* khusus, yaitu *register TMOD* (*Timer/counter Mode Control*) dan *register TCON* (*Timer/counter Control*).
- *Stack Pointer* (SP); merupakan *register* 8 bit yang diletakkan di alamat manapun pada RAM internal. Isi *register* ini ditambah sebelum data disimpan, selama instruksi *PUSH* dan *CALL*. Pada saat *reset*, *register* SP diinisialisasi pada alamat  $07_H$ , sehingga *stack* akan dimulai pada lokasi  $08_H$ .

- *Data Pointer (DPTR)*; terdiri dari dua *register*, yaitu untuk *byte* tinggi (*Data Pointer High*) dan untuk *byte* rendah (*Data Pointer Low*). Fungsinya untuk menahan alamat 16-bit. DPTR dapat dimanipulasi sebagai *register* 16-bit atau sebagai dua buah *register* 8-bit.
- *Port 0* sampai *Port 3* merupakan *register* yang berfungsi untuk membaca dan mengeluarkan data pada *port* 0, 1, 2, dan 3. Masing-masing *register* ini dapat dialamati per-bit maupun per-*byte*.
- *Serial Data Buffer* seharusnya merupakan dua buah *register* yang terdiri dari *transmitt buffer register* dan *receive buffer register*. Pada saat data dipindahkan ke SBUF, *register* tersebut akan menjadi *transmit buffer register* sedangkan pada saat data dipindahkan dari SBUF maka *register* tersebut akan berubah menjadi *receive buffer register*.
- *Timer Register*; terdiri dari pasangan *register* TH0, TL0 dan TH1, TL1 merupakan *register* 16 bit yang berfungsi sebagai *register counter* untuk *timer/counter* 0 dan *timer/counter* 1.

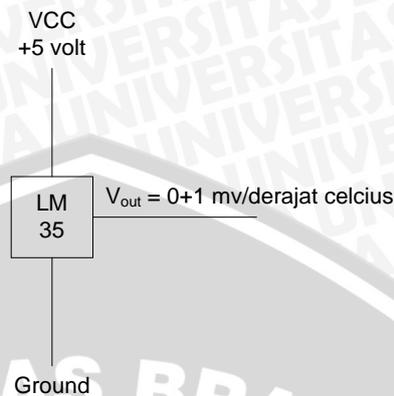
Dalam skripsi ini kelembabannya dinyatakan dalam bentuk kelembaban relatif. Karena suhu dan kelembaban masih merupakan besaran nonelektrik maka dibutuhkan sebuah sensor untuk mengubah dari besaran nonelektrik menjadi besaran elektrik. Sensor yang digunakan dalam skripsi ini adalah LM 35 dan HS 15 P. Kedua sensor ini merupakan masukan untuk mikrokontroler.

## 2.5 Sensor Suhu LM35

Sensor suhu IC LM35 merupakan komponen yang sangat peka atau sangat mudah mengalami perubahan tegangan, apabila IC LM35 tersebut dikenai perubahan suhu. Hal disebabkan sifat dari bahan konduktor yang peka terhadap perubahan suhu. Besarnya tegangan keluaran pada IC LM35 adalah berbanding lurus dengan suhu mutlak dan perubahan sebesar 10mv tiap derajat celcius. Hal ini memungkinkan pembacaan tegangan sebagai akibat dari perubahan suhu dengan alat ukur yang mempunyai sensitivitas milivolt.

Karakteristik umum IC LM35 tidak memerlukan kalibrasi eksternal, memiliki tingkat ketelitian 0,5°C pada suhu dibawah 25°C, beroperasi dalam jangkauan tegangan antara -550 mv sampai 1500 mv dan mempunyai jangkauan

penginderaan antara  $-55^{\circ}\text{C}$  sampai  $150^{\circ}\text{C}$ . Gambar IC LM35 adalah seperti Gambar 2.7 dibawah ini:

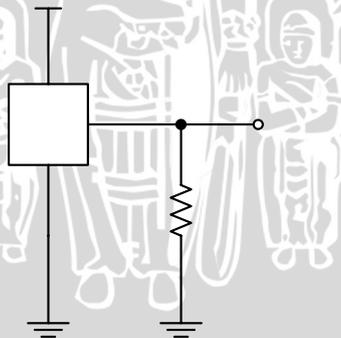


**Gambar 2.7** IC LM35

Sumber : National Semiconductor, 1995

Untuk mempermudah jangkauan penginderaan (*full range*) maka harus ditambah suatu resistor ( R ) yang dipasang parallel dengan *output* IC LM35 dan eksternal supply (*ground*) seperti yang terlihat pada Gambar 2.8 dibawah ini. Untuk menentukan besarnya harga resistor dan *supply* yang harus dipasang sesuai dengan spesifikasi perhitungan dari IC LM35 yaitu:

$$V_{\text{out}} = 0 \text{ mv} + 10,0 \text{ mv}/^{\circ}\text{C}. \quad (2-2)$$



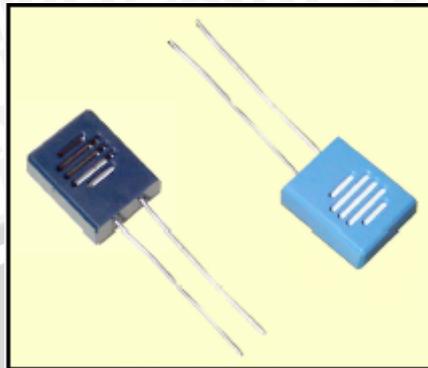
**Gambar 2.8** IC LM35 dengan R Parallel

Sumber : National Semiconductor, 1995

## 2.6 Sensor Kelembaban HS15P

Sensor kelembaban yang umum digunakan adalah HS15P. Sensor ini tahan terhadap air (*water resistant*), bekerja pada temperatur  $0^{\circ}\text{C}$  -  $50^{\circ}\text{C}$  dan kelembaban 20%-100% RH. Memiliki impedansi  $60 \text{ k}\Omega$  pada suhu  $25^{\circ}\text{C}$  dan

kelembaban 50% RH. Tegangan operasinya 1Vrms AC dengan frekuensi 50 Hz – 1 kHz dan menyerap daya sebesar 0,3 mW.



**Gambar 2.9** Sensor Kelembaban HS12P dan HS15P

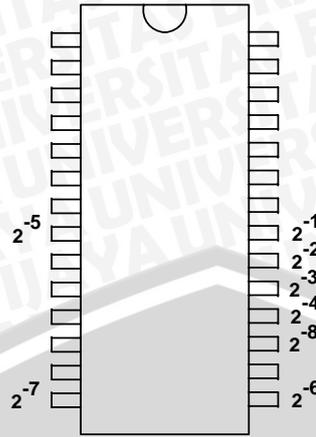
Sumber : Data Sheet Crown Industrial Estate

Karena keluaran dari sensor suhu dan kelembaban masih berupa sinyal *analog* maka dibutuhkan sebuah ADC yang digunakan untuk mengubah sinyal *analog* menjadi sinyal *digital*. Sehingga data dari *sensor* dapat diolah oleh mikrokontroler. ADC yang digunakan dalam skripsi ini adalah ADC 0809.

### 2.7 ADC 0809

ADC 0809 adalah komponen akuisisi data CMOS monolit dengan *analog-to-digital converter* 8-bit, multiplekser 8-kanal dan kontrol logika yang cocok dengan mikroprosesor. *A/D converter* 8-bit ini menggunakan *successive approximation* sebagai metode konversinya. Waktu konversi selama 100  $\mu$ s cukup cepat untuk aplikasi pengukuran dengan perubahan besaran yang lambat. *Devais* ini tidak membutuhkan pengaturan nol dan skala penuh serta memiliki *interface* yang mudah dengan mikroprosesor. Resolusi 8-bit mencukupi untuk aplikasi pengukuran yang tidak memerlukan ketelitian tinggi.

Untuk mengolah data dari sensor maka dibutuhkan suatu alat yang dapat mengolah data yang disebut mikrokontroler. Dalam skripsi ini mikrokontroler yang digunakan adalah AT89S52. Akan tetapi mikrokontroler tidak bisa langsung mengolah data dari sensor melainkan membutuhkan perangkat lain sehingga data dari sensor bisa diolah oleh mikrokontroler, perangkat lain ini disebut dengan ADC. ADC ini berfungsi untuk mengubah data analog menjadi data digital. Dalam skripsi ini ADC yang digunakan yaitu ADC 0809.



IN3	1
IN4	2
IN5	3
IN6	4
IN7	5
START	6
EOC	7
OE	8
CLOCK	9
VCC	10
Vref (+)	11
GND	12
	13
	14

**Gambar 2.10** Susunan Pin ADC 0809

Sumber : National Semiconductor, 1995

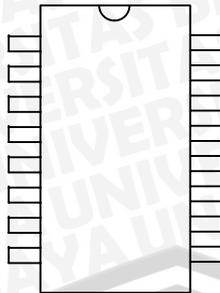
Karena data keluaran dari ADC berupa sinyal digital, maka dapat dihubungkan dengan mikrokontroler. Adapun mikrokontroler yang digunakan dalam skripsi ini adalah MCU AT89S52. Selain membutuhkan masukan, mikrokontroler juga mengeluarkan *output*. Keluaran dari mikrokontroler ini dihubungkan dengan DAC.

**2.8 DAC (Digital to Analog Converter) 0808**

DAC 0808 merupakan sebuah konverter yang mengambil harga yang dinyatakan dalam kode digital dan mengubahnya menjadi tegangan atau arus yang sebanding dengan nilai digitalnya tersebut. Sistem yang dipakai adalah sistem R-2R 8-Bit yang dilengkapi dengan sumber arus acuan dan delapan buah transistor saklar. Gambar IC DAC ditunjukkan dalam Gambar 2.12, pin 5 sampai pin 12 merupakan masukan data yang akan dikonversi kedalam bentuk analog. Pin 1 tidak digunakan dalam artian tidak ada sambungan. Pin 2 digunakan untuk *ground* atau pentanahan dengan tegangan catu positif pada pin 13 dan tegangan negatif pada pin3.

Tegangan referensi dihubungkan dengan pin 14 dan 15 untuk negatifnya. Referensi yang digunakan sebenarnya referensi arus, sehingga diperlukan resistor eksternal untuk mengatur arus referensi. Keluaran DAC ini terdapat pada pin 4 yang berupa arus.





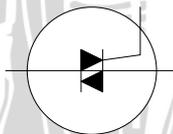
**Gambar 2.11** Susunan Pin DAC 0808  
 Sumber : National Semiconductor, 1995

- NC 1
- GND 2
- VEE 3
- IO 4
- MSB A1 5
- A2 6
- A3 7
- A4 8

**2.9 Triac**

Triac adalah komponen 3 elektroda dari keluarga *thyristor*. Triac merupakan 2 buah SCR yang dihubungkan paralel berkebalikan dengan terminal *gate* bersama. Berbeda dengan SCR yang hanya melibatkan tegangan dengan polaritas positif saja, triac dapat dipicu dengan tegangan polaritas positif dan negatif serta dapat dihidupkan dengan menggunakan tegangan bolak-balik yang ada pada *gate*. Triac banyak digunakan pada rangkaian pengendali dan pensaklaran.

Triac akan menghantar jika pada terminal *gate* diberi pemicuan yang berupa arus dengan tegangan positif dan negatif. Triac akan tetap on bila arus *thermis* yang mengalir pada triac lebih besar dari arus penahanan ( $I_T > I_H$ ).



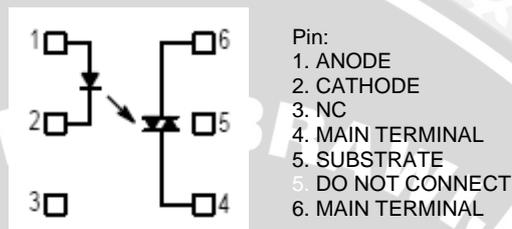
**Gambar 2.12** Simbol Triac  
 Sumber: Malvino, 1987: 271

**2.10 Optotriac MOC3021**

Optotriac MOC3021 mempunyai dioda LED *gallium arsenida* dan sebuah *phototriac*. Saat cahaya menimpa *phototriac* maka *phototriac* akan berkonduksi. MOC3021 didesain untuk pemakaian *triac* sebagai antarmuka sistem logika ke jaringan 220 VAC. *Phototriac* dipakai sebagai pengendali *triac*, oleh karena itu, *phototriac* dalam *optotriac* ini dinamakan *triac driver*.



LED GaAs mempunyai tegangan jatuh 1,5V nominal pada 10 mA dan suatu tegangan *breakdown* balik lebih besar dari 3V. Arus maksimal yang dapat dilewatkan melalui LED adalah 60 mA. MOC mempunyai tegangan penghalang minimum sebesar 400 Vdc pada kedua arah dengan kurang dari 3 V tegangan jatuhnya. Sekali terpicu dalam keadaan on (menghantar), *phototriac* akan tetap on demikian sampai arusnya turun di bawah arus penahanan sebesar 100  $\mu$ A, pada saat di mana *phototriac* akan kembali ke keadaan off.



**Gambar 2.13** Rangkaian Aplikasi Standar MOC3021

Sumber: Nasional Semikonduktor

## 2.11 LCD (Liquid Crystal Display)

*Liquid Crystal Display* atau LCD adalah salah satu jenis penampil yang digunakan untuk menampilkan angka, karakter, atau bahkan angka dan karakter. LCD terdiri atas tumpukan tipis atau sel dari dua lembar kaca dengan pinggiran tertutup rapat. Antara dua lembar kaca tersebut diberi bahan kristal cair (*liquid crystal*), yang tembus cahaya. Permukaan luar dari masing-masing keping kaca mempunyai lapisan penghantar tembus cahaya seperti oksida timah (*tin oxide*) atau oksida indium (*indium oxide*). Sel mempunyai ketebalan sekitar  $1 \times 10^{-5}$  meter dan diisi dengan kristal cair (*liquid crystal*). Blok diagram LCD ditunjukkan dalam Gambar 3.8.

LCD merupakan modul tampilan yang mempunyai konsumsi daya yang relatif rendah. LCD memiliki keistimewaan dibandingkan tampilan yang lain seperti *sevent segment* yaitu kemampuan untuk menampilkan karakter dan berbagai macam simbol. Salah satu jenis LCD diantaranya adalah LCD tipe *dot matriks* 5x7, tersusun sebanyak dua baris dan masing-masing baris terdiri atas 20 karakter. Setiap karakter dibentuk oleh 5x7 buah titik, sehingga jenis huruf yang

mampu ditampilkan akan lebih banyak dan lebih baik dibandingkan dengan penampil 7 *segment* atau 16 *segment*.

Dalam gambar terlihat bahwa dengan adanya controler pada LCD maka LCD dapat dianggap sebagai suatu lokasi memori dari suatu unit prosesor, sehingga instruksi penampilan karakternya dapat digabungkan menjadi satu dengan unit prosesor.

Untuk dapat menggunakan LCD maka hal-hal yang perlu diperhatikan adalah sinyal kontrol dari LCD yaitu : RS, R/W dan EN. Instruksi operasi meliputi operasi dasar *register*, *busy flag*, *address counter*, *display data RAM*, *character generator ROM*.

- *Register*

Kontroler LCD mempunyai dua buah *register* 8 bit, yaitu *register* interupsi (IR) dan *register* data (DR). Kedua *register* ini dipilih melalui *register select* (RS)

- *Busy Flag*

*Busy flag* menunjukkan bahwa modul LCD siap untuk menerima instruksi selanjutnya. Sebagaimana terlihat pada tabel *register* seleksi, sinyal akan melalui DB<sub>7</sub> jika RS = 0 dan R/W = 1.

- *Address Counter (AC)*

*Address counter* menunjukkan suatu lokasi memori dalam LCD dimana suatu data diletakkan dalam modul LCD. Pemilihan lokasi alamat diberikan melalui *register* instruksi (IR). Ketika data berada pada AC, maka secara otomatis AC akan menaikkan atau menurunkan alamat dari *entry mode set*.

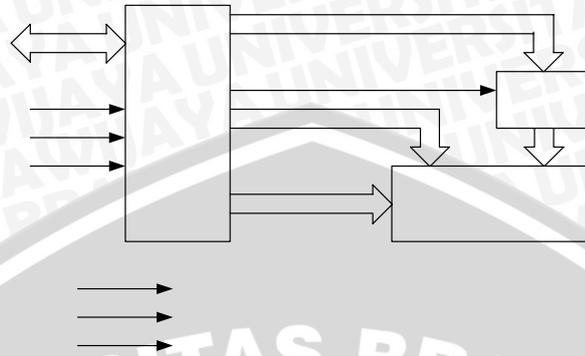
- *Display Data RAM (DD RAM)*

Pada LCD masing-masing line mempunyai *range* alamat tersendiri. Alamat-alamat itu diekspresikan melalui bilangan hexadesimal. Untuk line 1 menggunakan kisaran alamat 00<sub>H</sub> - 0F<sub>H</sub>, sedangkan pada line 2 menggunakan kisaran alamat 40<sub>H</sub> - 4F<sub>H</sub>.

- *Character Generator ROM (CG ROM)*

CG ROM membangkitkan 192 buah tipe 5 X 7 karakter dot matriks. Pada LCD telah tersedia ROM sebagai pembangkit karakter dalam kode ASCII. CG

RAM digunakan apabila diinginkan untuk pembuatan karakter tersendiri melalui program, dengan maksimal 8 buah karakter.



**Gambar 2.14** Diagram Blok LCD Seiko Instrument M1632  
 Sumber: *Seiko Manual*, 1998:13 DB0-7

**2.11.1 Konfigurasi Pin-pin LCD**

Konfigurasi dan fungsi-fungsi umum pin-pin LCD ditunjukkan dalam Tabel 2.2 berikut ini:

**Tabel 2.2** Konfigurasi pin-pin LCD  
 Sumber: EL-TECH Electronics, 1987:2

No	SIMBOL	LEVEL	FUNGSI	
1	Vss	-	Power Supply	0 Volt ( <i>Ground</i> )
2	Vcc	-		5 Volt +/- 10 %
3	Vee	-		For LCD Drive
4	RS	H/L	H = <i>Input Data</i> L = <i>Input Instruksi</i>	VDD VSS
5	R/W	H/L	H = <i>Read</i> L = <i>Write</i>	VLC
6	E	H	Data Bus	
7	DB0	H/L		
8	DB1	H/L		
9	DB2	H/L		
10	DB3	H/L		
11	DB4	H/L		
12	DB5	H/L		
13	DB6	H/L		
14	DB7	H/L		
15	V+ BL	-	Back Light Supply	4 – 4,2 Volt, 50 – 200 mA
16	V- BL	-		0 Volt ( <i>Ground</i> )

Controller



### 2.11.2 Fungsi-fungsi Terminal

Fungsi-fungsi terminal ditunjukkan dalam Tabel 2.3 berikut ini:

**Tabel 2.3** Fungsi-fungsi Terminal

Sumber: EL-TECH Electronics, 1987:7

NAMA SINYAL	NO. TERM.	I/O	TUJUAN	FUNGSI
DB <sub>0</sub> – DB <sub>3</sub>	4	I/O	MPU	Lalu lintas data dan instruksi, lower <i>byte</i>
DB <sub>4</sub> – DB <sub>7</sub>	4	I/O	MPU	Lalu lintas data dan instruksi, upper <i>byte</i>
E	1	INPUT	MPU	Sinyal Start (Read/Write)
R/W	1	INPUT	MPU	Sinyal seleksi instruksi
RS	1	INPUT	MPU	Sinyal seleksi <i>register</i>
V <sub>LC</sub>	1	-	Power Supply	Driver LCD
V <sub>DD</sub>	1	-	Power Supply	5 Volt
V <sub>SS</sub>	1	-	Power Supply	0 Volt ( <i>Ground</i> )

LCD tipe M1632 ini memiliki spesifikasi sebagai berikut:

- LCD ini terdiri dari 32 karakter dengan 2 baris masing-masing 16 karakter dengan *display* dot matrik 5x7
- Karakter generator ROM dengan 192 tipe karakter
- Karakter generator RAM dengan 8 bit karakter
- 80x8 bit display data RAM
- Dapat diinterfacekan ke MCU 8 atau 4
- Dilengkapi fungsi tambahan antara lain *display clear*, *cursor home*, *display on/off*, *cursor on/off*, *display character blink*, *cursor shift*, *display shift*
- *Internal data*
- *Internal* otomatis, reset pada saat *power on*
- Tegangan +5 volt PSU tunggal

*Liquid Crystal Display* (LCD) ini mempunyai konsumsi daya yang relatif rendah dan terdapat sebuah kontroler CMOS di dalamnya. Kontroler tersebut sebagai pembangkit dari karakter ROM/RAM dan *display* data RAM. Semua

fungsi tampilan dikontrol oleh suatu instruksi dan modul LCD dapat dengan mudah untuk diinterfacekan dengan mikrokontroler. Masukan yang diperlukan untuk mengendalikan modul ini berupa bus data yang masih termultiflex dengan bus alamat serta 3 bit sinyal kontrol. Sementara pengendalian dot matrik LCD dilakukan secara internal oleh kontroler yang sudah ada pada modul LCD.

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

- Register

Kontroler dari LCD mempunyai 2 buah register 8 bit yaitu register instruksi (IR) dan register data (DR). IR menyimpan instruksi seperti *display clear*, *cursor shift* dan *display data* (DD RAM) serta *character generator* (CG RAM). DR menyimpan data untuk ditulis di DD RAM atau CG RAM ataupun membaca data dari DD RAM atau CG RAM. Ketika data ditulis ke DD RAM atau CG RAM maka DR secara otomatis menulis data ke DD RAM atau CG RAM. Ketika data pada DD RAM atau CG RAM akan dibaca maka alamat data ditulis pada IR sedangkan data akan dimasukkan melalui DR dan mikrokontroler membaca data dari DR.

- *Busy Flag*

*Busy flag* menunjukkan bahwa *module* siap untuk menerima instruksi selanjutnya. Sebagaimana yang terlihat dalam Tabel 3.3, register seleksi sinyal akan melalui  $DB_7$  jika  $RS=0$  dan  $R/W=1$ . Jika bernilai 1 maka modul LCD sedang melakukan kerja internal dan instruksi tidak akan diterima. Oleh karena itu status dari *flag* harus diperiksa sebelum melaksanakan instruksi selanjutnya.

**Tabel 2.4** Register seleksi

Sumber: EL-TECH Electronics, 1987:48

RS	R/W	OPERASI
0	0	Seleksi IR, IR Write <i>Display Clear</i>
0	1	<i>Busy Flag</i> ( $DB_7$ ) @ <i>counter</i> ( $DB_0$ - $DB_7$ ) read
1	0	Seleksi DR, DR write
1	1	Seleksi DR, DR write