

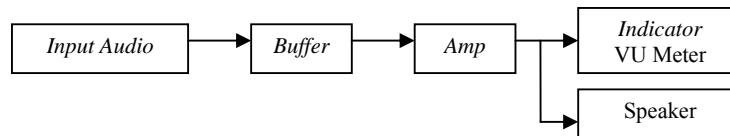
2. DASAR TEORI

2.1. Teori VU Meter

VU meter (*Volume – Unit* meter) merupakan salah satu pelengkap dari perangkat *audio* dan saat ini banyak dikembangkan misalnya pada *tape compo*, *power amplifier* dan aplikasi – aplikasi *audio* pada komputer. Pada dasarnya VU meter adalah sebuah tampilan dalam bentuk grafik yang bergerak mengikuti irama atau perubahan tekanan suara pada frekuensi – frekuensi tertentu. Gerakan pada *level VU* meter mewakili besarnya *gain* (dB) yang yang dihasilkan dari suara *audio*. Selain itu VU meter juga dapat menambah kesan lebih menarik dan lebih hidup bagi pendengar. Pada umumnya VU meter dibagi menjadi 2 bentuk yaitu VU meter *analog* dan VU meter *digital*, untuk pembahasan lebih lanjut akan dibahas pada sub bab berikut :

2.1.1. VU Meter Analog

Biasanya VU meter ini memiliki *indicator* berupa gerakan jarum yang sesuai dengan perubahan irama atau musik secara *analog*. Biasanya digunakan pada *power amplifier* sebagai *indicator* dari *input* atau *output*. Untuk penggunaannya dari VU meter ini tidak banyak biasanya *Power amplifier stereo* hanya menggunakan 2 buah *indicator* yaitu untuk menampilkan tekanan suara pada sisi kiri dan sisi kanan tanpa pembagian frekuensi.



Gambar 2.1. Sistem VU Meter Analog

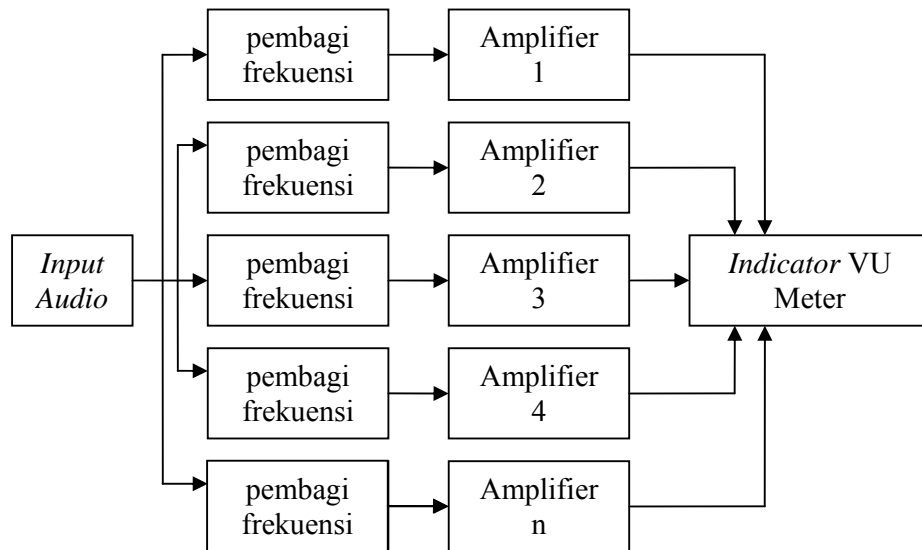
Adapun bentuk dari VU meter *analog* adalah sebagai berikut :



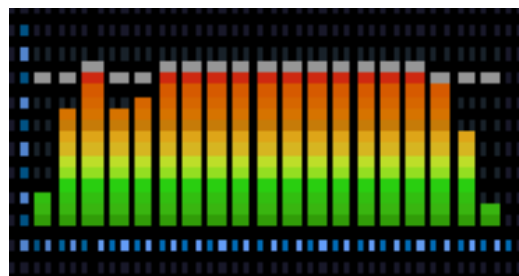
Gambar 2.2. Bentuk VU Meter Analog

2.1.2 VU Meter Digital

VU meter *Digital* merupakan *indicator* dari irama musik yang dibuat dengan menggunakan tampilan lampu atau grafik. Untuk media dapat berupa LED atau LCD. Biasanya pada VU meter *digital* suara dibagi menjadi beberapa bagian menurut *range* frekuensinya, dan selanjutnya untuk tiap bagian ditampilkan pada media.



Gambar 2.3. Sistem VU Meter *Digital* n Channel



Gambar 2.4. Bentuk VU Meter *Digital*

2.1.3. Pemilihan Frekuensi

Rentang frekuensi atau *bandwidth* musik sangat lebar. Dan untuk mengatur frekuensi tertentu yang diinginkan atau tidak diinginkan, perlu adanya pembagian rentang frekuensi tersebut menjadi *band – band* yang lebih kecil, (mulai dari *band – band* untuk frekuensi rendah atau *bass*, frekuensi tengah atau *middle*, dan frekuensi tinggi atau *treble*).

Untuk merencanakan suatu pembagian frekuensi pada *VU* meter, harus ditentukan terlebih dahulu rentang frekuensi atau *bandwidth*-nya dengan menetapkan besar *oktaf*, (satu *oktaf* adalah kenaikan frekuensi sebesar dua kali frekuensi mula-mula). Misalnya pembagian frekuensi 1 *oktaf*, maka frekuensi standart dikali dan dibagi 2, selanjutnya muncul nilai baru nilai tersebut dikali dan dibagi kembali. Dari hasil perkalian dan pembagian tersebut diperoleh hasil sebagai berikut :

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 31 | 63 | 125 | 250 | 500 | 1k0 | 2k0 | 4k0 | 8k0 | 16k | | | | | | | | | | | | | | | | | | | | |
| 31 | 44 | 63 | 87 | 125 | 175 | 250 | 350 | 500 | 700 | 1k0 | 1k4 | 2k0 | 2k8 | 4k0 | 5k8 | 8k0 | 11k | 16k | 20k | | | | | | | | | | |
| 25 | 31 | 40 | 50 | 63 | 80 | 100 | 125 | 160 | 200 | 250 | 315 | 400 | 500 | 630 | 800 | 1k0 | 1k2 | 1k6 | 2k0 | 2k5 | 3k2 | 4k0 | 5k0 | 6k3 | 8k0 | 10k | 12k | 16k | 20k |

Gambar 2.5. Pembagian Frekuensi 1 *Oktaf*, 1/2 *Oktaf*, 1/3 *Oktaf*

Nilai-nilai pembagian frekuensi diatas merupakan hasil pembulatan. Untuk pembagian frekuensi dibatasi oleh *range* frekuensi *audio* (0-20kHz), sehingga nilai frekuensi diatas 20 kHz dihilangkan. Untuk memperoleh frekuensi *center* dapat menggunakan rumus berikut :

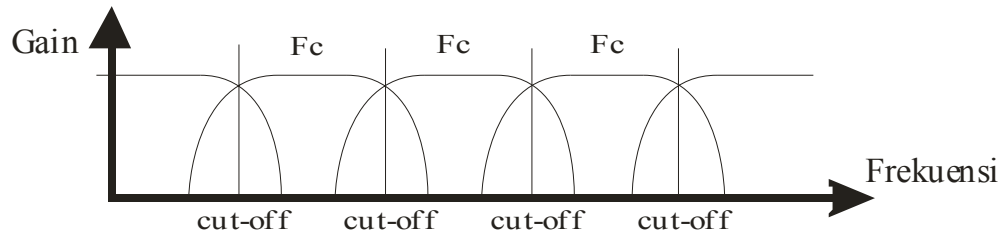
$$Center\ band = \sqrt{F_{max} \cdot F_{min}} \quad (2.1)$$

Kemudian tentukan faktor pengali dan pembagi untuk *center band* tersebut dengan menggunakan rumus :

$$Faktor\ pengali = (selisih\ n\ oktaf)^{\frac{n}{d}} \quad (2.2)$$

Dimana, $\frac{n}{d}$ adalah nilai pembagi *oktaf*.

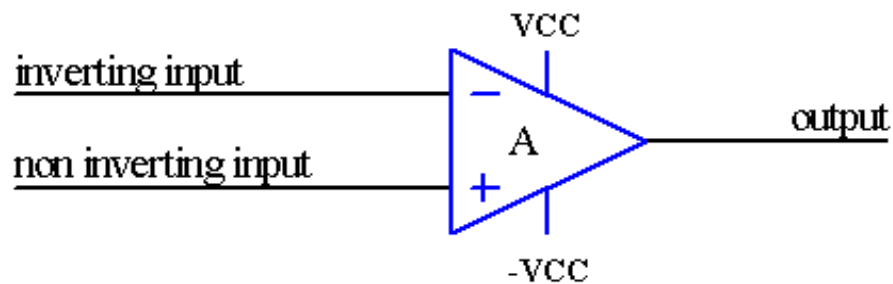
Untuk pembagian *range* frekuensi tiap *filter* diharapkan mencakup keseluruhan frekuensi *audio* (tanpa menghilangkan frekuensi tertentu). Sehingga menghasilkan respon frekuensi yang saling berpotongan pada frekuensi *cut-off*.



Gambar 2.6. Pembagian Respon Frekuensi

2.2. Operational Amplifier (*Op-Amp*)

Op-Amp atau (*Operational Amplifier*) merupakan rangkaian dengan elemen tahanan, resistor, kapasitor dan transistor yang dibuat secara *integrated circuit* (IC). *Op-Amp* mempunyai lima terminal dasar yaitu, dua terminal untuk mensuplai daya, dua terminal untuk masukan (masukan pembalik / *inverting input* dan masukan tak membalik/ *non-inverting input*), dan satu terminal untuk keluaran (*output*).



Gambar 2.7. *Operational Amplifier (Op-Amp)*

Op-Amp mempunyai beberapa fungsi, diantaranya sebagai *amplifier* (*inverting amplifier* dan *non-inverting amplifier*) dan sebagai *buffer*.

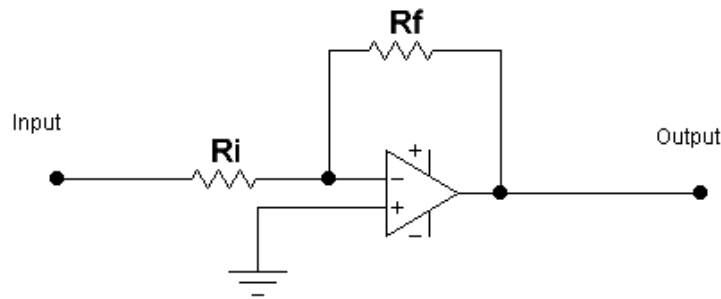
2.2.1. Amplifier

Karakteristik-karakteristik yang dimiliki oleh *Op-Amp* sebagai *amplifier* ideal, yaitu:

- Impedansi *input* yang tinggi (Z_i)
- Impedansi *output* yang rendah (Z_o)
- Mempunyai penguatan tegangan yang tinggi
- Tegangan *output* = 0, jika *input* = 0

2.2.2. Inverting Amplifier

Salah satu penggunaan *Op-amp* yang terpenting adalah sebagai *amplifier*. *Amplifier* adalah suatu rangkaian yang menerima sinyal dengan tegangan kecil dan menghasilkan sinyal dengan tegangan yang lebih besar pada *output*-nya. Selain itu *inverting amplifier* memiliki sifat membalik *signal input* (jika *input* negative maka *output* positif). Rangkaian *inverting amplifier* dapat dilihat pada gambar 2.8 berikut ini :



Gambar 2.8. *Inverting Amplifier*

Pada gambar terlihat bahwa sebuah resistor *feedback* R_f dihubungkan antara terminal *output* dan terminal *input inverting*. Konfigurasi seperti ini dinamakan *negative feedback*. Keuntungan dari penggunaan *negative feedback* ini adalah bahwa penguatan *amplifier* tidak lagi tergantung pada *open loop gain* A_{OL} dari *Op-amp* tetapi bergantung pada *closed loop gain* A_{CL} . Penguatan dari rangkaian ini hanya ditentukan oleh resistor-resistor *eksternal*, yaitu resistor *feedback* R_f dan resistor *input* R_i , dan dinamakan *closed loop gain* A_{CL} .

Antara terminal *input inverting* (-) dan terminal *input non inverting* (+) pada *Op-Amp* tidak terdapat beda tegangan ($E_d = 0$), sehingga tidak ada arus yang masuk melalui terminal *input Op-Amp*. Arus akan mengalir melalui R_i , R_f kemudian terhubung dengan terminal *output* dari *Op-Amp*. Karena arus mengalir dari tegangan yang lebih positif menuju negatif, tegangan pada terminal *output* dari *Op-amp* besarnya negatif.

Tegangan *output* yang dihasilkan oleh rangkaian yang terdapat pada gambar 2.8 dapat ditentukan dengan menggunakan persamaan berikut di bawah ini :

$$V_A = 0$$

$$; V_A = \text{tegangan pada terminal } \textit{input inverting} (-) \quad (2.3)$$

$$I = \frac{V_i}{R_i} \quad ; I = \text{ arus yang mengalir pada } R_i \quad (2.4)$$

$$V_{R_f} = I \times R_f \quad ; V_{R_f} = \text{ tegangan pada } R_f \quad (2.5)$$

$$V_{R_f} = \frac{V_i}{R_i} \times R_f$$

$$V_{R_f} = \frac{R_f}{R_i} \times V_i$$

$$V_o = -V_{R_f} \quad ; V_o = \text{ tegangan } \textit{output Op-amp} \quad (2.6)$$

$$V_o = -V_i \frac{R_f}{R_i}$$

Tanda negatif pada persamaan 2.5 menunjukkan bahwa polaritas dari tegangan *output Op-amp* terbalik dengan polaritas tegangan *input Op-amp*, untuk alasan inilah rangkaian pada gambar 2.8. disebut rangkaian *inverting amplifier*.

Penguatan dalam rangkaian *inverting amplifier* ini dapat dihitung dengan menggunakan rumus :

$$A_{CL} = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_i} \quad (2.7)$$

Dimana :

A_{CL} = Penguatan *loop* tertutup

R_f = Resistansi *feedback* (ohm atau Ω)

R_i = Resistansi *input* (ohm atau Ω)

Arus beban (I_L) pada beban (R_L) hanya ditentukan oleh R_L dan V_o . Sehingga besarnya arus beban (I_L) dapat dihitung sebagai berikut :

$$I_L = -\frac{V_o}{R_L} \quad (2.8)$$

Arus *Op-amp* merupakan penjumlahan dari arus yang masuk dari sumber tegangan *Op-amp* (V_i) dan arus beban (I_L).

Pada *inverting amplifier*, impedansi *input* (Z_{in}) sama dengan R_i karena kaki pada resistor selain terhubung dengan tegangan *input* juga terhubung dengan *virtual ground* ($E_d = 0$), dan juga menurut teorema umpan balik negatif (PP = paralel - paralel). PP (paralel – paralel) berarti *input* dari *amplifier* terhubung paralel dengan *feedback* sedangkan *outputnya* juga terhubung secara paralel, berikut ini adalah penurunan rumus dari impedansi *input* (Z_{in}) dari rangkaian *inverting amplifier* :

$$Z_{in(PP)} = \frac{R}{A} \parallel Z_{in} \quad (2.9)$$

Karena nilai R/A jauh lebih kecil dari Z_{in} , maka:

$$Z_{in(PP)} \cong \frac{R}{A} \quad (2.10)$$

Dimana :

$Z_{in(PP)}$: impedansi *input* paralel -paralel

A : penguatan dalam *Op-Amp*.

R : penguatan tegangan umpan balik.

Z_{in} : impedansi dalam dari *op - amp*

Sedangkan untuk perhitungan Impedansi *output* (Z_{out}) dari rangkaian *inverting amplifier* dapat diturunkan sebagai berikut :

$$Z_{out(PP)} = \frac{Z_{out}}{1 + AB} \quad (2.11)$$

Dimana :

$Z_{out(PP)}$: impedansi *output* paralel-paralel.

Z_{out} : impedansi *output* dalam *Op-Amp*

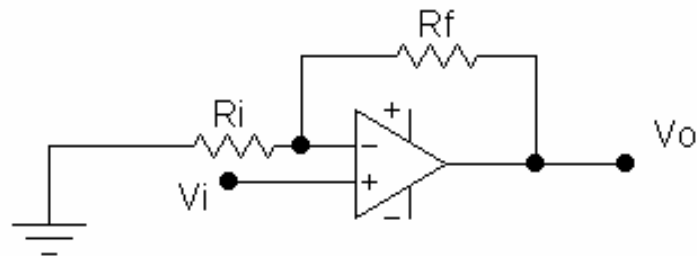
A : penguatan *open loop* dari *Op-Amp*.

B : penguatan *negative feedback* dari *op - amp*

Pada persamaan (2.9) dan persamaan (2.10), nilai 1 (satu) sangat kecil dari A_{OL} dari *Op-Amp* yang sangat besar, maka nilai 1 (satu) dapat diabaikan. Nilai Z_{in} dan Z_{out} dari hasil persamaan (2.9) didapatkan $Z_{in} = R_i$, dan nilai Z_{out} didapatkan $Z_{out} = R_L$. Biasanya, nilai dari R_L lebih kecil dibandingkan dengan nilai R_i .

2.2.3. Non-Inverting Amplifier

Tegangan *output Op-amp* (V_o) dari rangkaian ini mempunyai *polaritas* yang sama dengan tegangan *input Op-amp* (V_i). Antara terminal *input inverting* (-) dan terminal *input non inverting* (+) pada *Op-amp* tidak terdapat beda tegangan ($E_d = 0$), sehingga tidak ada arus yang masuk melalui terminal *input Op-amp*. Arus akan mengalir melalui terminal *output* dari *Op-amp*, R_f , R_i kemudian terhubung dengan terminal *input* dari *Op-amp*. Karena arus mengalir dari tegangan yang lebih positif menuju negatif, tegangan pada terminal *output* dari *Op-amp* besarnya positif.



Gambar 2.9. Non Inverting Amplifier

Tegangan *output* yang dihasilkan oleh rangkaian yang terdapat pada gambar 2.9 dapat ditentukan dengan menggunakan persamaan berikut di bawah ini :

$$V_A = 0$$

; V_A = tegangan pada terminal *input inverting* (-) (2.11)

$$I = \frac{V_i}{R_i} \quad ; I = \text{ arus yang mengalir pada } R_i \quad (2.13)$$

$$V_{R_f} = I \times R_f \quad ; V_{R_f} = \text{ tegangan pada } R_f \quad (2.14)$$

$$V_{R_f} = \frac{V_i}{R_i} \times R_f$$

$$V_{R_f} = \frac{R_f}{R_i} \times V_i$$

Tegangan terminal *output Op-amp* merupakan penjumlahan dari tegangan pada R_i yaitu V_i dengan tegangan di R_f yaitu V_{R_f} sehingga :

$$V_o = V_i + \frac{R_f}{R_i} \times V_i \quad ; V_o = \text{tegangan output Op-amp} \quad (2.15)$$

$$V_o = \left(1 + \frac{R_f}{R_i} \right) \times V_i$$

Penguatan rangkaian *non inverting amplifier* dapat dihitung dengan menggunakan rumus sebagai berikut :

$$A_{CL} = \frac{R_f}{R_i} + 1 \quad (2.16)$$

Dimana:

A_{CL} = Penguatan *loop* tertutup

R_f = Resistansi *feedback* (ohm atau Ω)

R_i = Resistansi *input* (ohm atau Ω)

Karena sinyal masukan langsung masuk ke *input* positif dari *Op-Amp*, maka dengan menggunakan rangkaian ini akan didapatkan impedansi *input* yang sangat besar (dalam hal ini impedansinya sama dengan impedansi *input Op-Amp* itu sendiri). Impedansi *input* dari rangkaian *non inverting amplifier* lebih baik apabila dibandingkan dengan Impedansi *input* dari rangkaian *inverting amplifier*.

Pada *non-inverting amplifier*, impedansi *input* tidak sama dengan pada *inverting amplifier* (karena tegangan yang *input* adalah tegangan *input* yang dihubungkan dengan *pin input non-inverting* dari *Op-Amp*). Dengan teorema umpan balik *negative SP* (seri – paralel). *SP* berarti *input* dari *amplifier* terhubung seri dengan *feedback* sedangkan *output*-nya terhubung paralel, berikut ini adalah rumus dari impedansi *input* (Z_{in}) dari rangkaian *non-inverting amplifier* :

$$Z_{in(SP)} = (1 + A_{OL} B) \times Z_{IN} \quad (2.17)$$

Dimana :

A_{OL} : penguatan dalam *Op-Amp*.

Z_{IN} : impedansi dalam *Op-Amp*.

$Z_{in(SP)}$: impedansi *input* seri-paralel.

B : penguatan tegangan umpan balik.

Karena penguatan dalam (A_{OL}) sangat besar, maka impedansi *input* adalah lebih besar dari impedansi *input Op-Amp*.

Sedangkan untuk perhitungan Impedansi *output* (Z_{out}) dari rangkaian *non-inverting amplifier* dapat diturunkan sebagai berikut :

$$Z_{out(SP)} = \frac{Z_{out}}{1 + A_{OL}B} \quad (2.18)$$

Karena faktor $(1 + A_{OL}B)$ sangat besar maka, impedansi *output* dengan umpan balik adalah mendekati nol.

$$Z_{out(SP)} \cong 0 \quad (2.19)$$

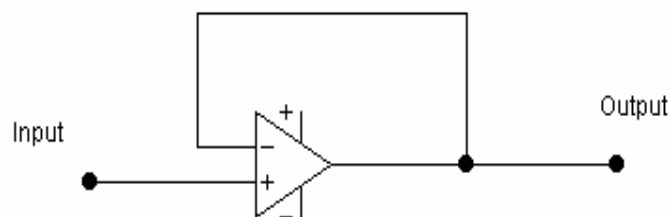
dimana,

Z_{out} : impedansi *output Op-Amp*.

$Z_{out(SP)}$: impedansi *output seri-paralel*

2.2.4. Buffer

Buffer disebut sebagai pengikut tegangan karena tegangan yang dihasilkan sama dengan tegangan *input*-nya. *Buffer* berfungsi untuk mengurangi pembebanan pada *input* sebuah *Op-amp*. Sehingga setelah tegangan melewati *buffer*, tidak akan mengganggu kerja komponen sebelumnya. Berikut gambar rangkaian dari *buffer* :



Gambar 2.10. Rangkaian *Buffer*

Pada rangkaian *buffer*, *output* dihubungkan langsung dengan *input* masukan negatif. Sebuah *level* tegangan diberikan pada *input* masukan positif, akan mengakibatkan tegangan *output* (V_{out}) adalah sama dengan *input* positif (V_{in}). Penguatannya dapat dihitung dengan rumus :

$$A_{CL} = \frac{V_{out}}{V_{in}} \quad (2.20)$$

Karena nilai V_{in} selalu sama dengan V_{out} , maka penguatannya adalah satu. Rangkaian ini mempunyai kelebihan yaitu nilai impedansi *input*nya jauh lebih besar dari impedansi *output*-nya. Dengan teorema umpan balik negatif, diperoleh impedansi *input* dan impedansi *output* seperti persamaan berikut :

- Impedansi *input* (Z_i)

$$\begin{aligned} Z_{in(SP)} &= (1 + A_{OL} B) \times Z_{in} \\ Z_{in(SP)} &= (1 + A_{OL}) \times Z_{in} \end{aligned} \quad (2.21)$$

Karena penguatan dalam *Op-Amp* (A_{OL}) sangat besar, maka nilai 1 (satu) dapat diabaikan, dan impedansi *input* yang diperoleh sangat besar sekali.

$$Z_{in(SP)} \cong A_{OL} \times Z_{IN} \quad (2.22)$$

- Impedansi *output* (Z_{out})

$$\begin{aligned} Z_{out(SP)} &= \frac{Z_{out}}{1 + A_{OL} B} \\ Z_{out(SP)} &\cong \frac{Z_{out}}{A_{OL}} \end{aligned} \quad (2.23)$$

Impedansi *output* (Z_{out}) dibagi dengan penguatan dalam dari *Op-Amp* yang sangat besar maka, impedansi *output* ($Z_{out(SP)}$) mendekati nol.

2.3. Filter

Filter adalah suatu rangkaian yang dirancang agar dapat melewatkan suatu sinyal *input* pada *range* frekuensi tertentu, dan melemahkan sinyal *input* yang mempunyai frekuensi di luar *range* frekuensi yang telah ditentukan. *Filter* terdapat dua macam yaitu *filter* aktif atau *filter* pasif. Pada *filter* pasif, rangkaian hanya terdiri atas komponen – komponen pasif (berupa tahanan, *inductor* dan kapasitor saja). *Filter* aktif dapat dirangkai dengan menggunakan transistor ataupun *Op-amp*, tetapi secara umum yang banyak dipakai adalah *Op-amp*. Ada empat jenis buah *filter* yaitu *low pass filter*, *high pass filter*, *band pass filter*, dan *band elimination* atau *band reject*.

Sebelum dibahas lebih lanjut tentang *active filter* ini akan dibahas terlebih dahulu tentang *oktaf*, *decade* dan *decibel*(dB).

- *Oktaf*

Yang disebut dengan satu *oktaf* adalah kenaikan frekuensi sebesar dua kali frekuensi mula-mula. Misalnya 50 Hz sampai 100 Hz disebut satu *oktaf*.

- *Decade*

Yang disebut dengan satu *decade* adalah kenaikan frekuensi sebesar sepuluh kali frekuensi mula-mula. Misalnya 50 Hz sampai 500 Hz disebut satu *decade*.

- *Decibel (dB)*

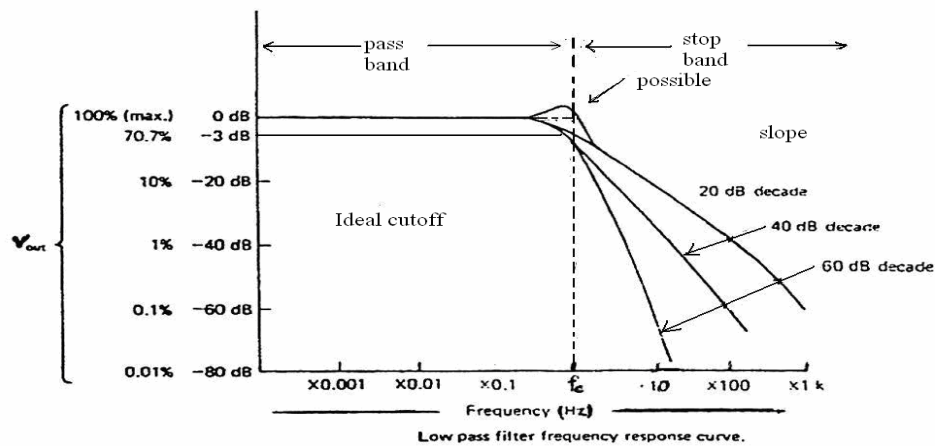
Decibel (dB) pada umumnya digunakan untuk menyatakan derajat penguatan (*gain*) atau pelemahan (*attenuation*). Bisa juga digunakan untuk menyatakan tegangan, arus, daya *output* dan banyak lainnya, misalnya:

- $\text{dB} = 20 \log \left(\frac{V_o}{V_i} \right)$, persamaan ini digunakan untuk menyatakan derajat penguatan atau pelemahan dari tegangan *output*.

- $\text{dB} = 10 \log \left(\frac{P_o}{P_i} \right)$, persamaan ini digunakan untuk menyatakan derajat penguatan atau pelemahan dari daya *output*.

2.3.1 *Low Pass Filter*

Filter low pass adalah sebuah rangkaian yang berfungsi meneruskan tegangan pada frekuensi rendah (kurang dari f_c) dan menghambat tegangan pada frekuensi tinggi (lebih dari f_c). Ketika tegangan *input* dari frekuensi rendah ke tinggi telah melewati batas frekuensi *cut-off* (f_c) maka amplitudo dari tegangan *output* akan melemah secara berkala. Berikut ini adalah gambar grafik tegangan *output* dari sebuah *low pass filter* terhadap frekuensi :

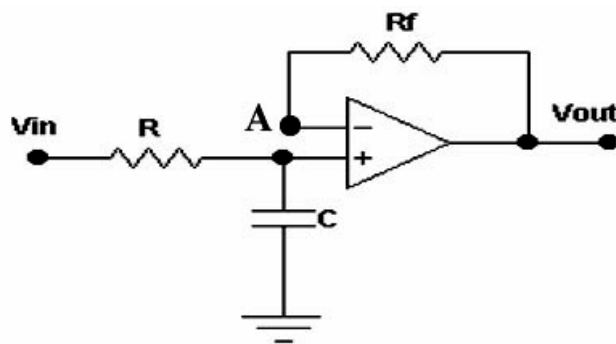


Gambar 2.11. Kurva Tanggapan Frekuensi *Low Pass Filter*

Sumber: Coughlin, Robert F. *Amplifiers and Linear Integrated Circuits*. 2th ed. New Jersey: Prentice Hall International, Inc, 1996, p.252

2.3.1.1. *Active Low Pass Filter Orde Pertama*

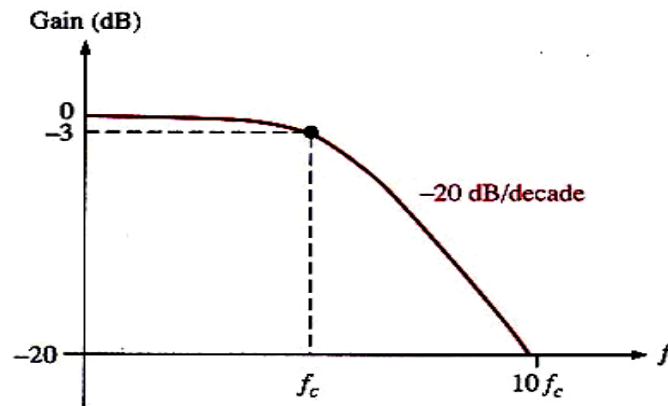
Active low pass filter orde pertama ini menggunakan *Op-Amp* yang mempunyai konfigurasi *non-inverting* (tidak membalik sinyal *input* yang masuk) dengan *input* nya yang berupa satu buah *resistor* yang dipasang secara seri dan sebuah *kapasitor* yang dipasang secara paralel dengan *ground*. Gambar rangkaian *active low pass filter orde pertama* ini terlihat pada gambar 2.12.



Gambar 2.12. *Active Low Pass Filter Orde Pertama*

Sumber: Floyd, Thomas L. *Electronic Devices*. 4th ed. New Jersey: Prentice Hall International, Inc, 1996, p.845

Pada *orde* pertama ini terjadi penurunan *gain* frekuensi di atas frekuensi *cut-off* sebesar -20 dB/decade. Gambar respon frekuensi dari *orde* pertama *active low pass filter* dapat dilihat pada gambar 2.13 :



Gambar 2.13. Respon Frekuensi *Orde* Pertama *Active Low Pass Filter*

Sumber: Floyd, Thomas L. *Electronic Devices*. 4th ed. New Jersey: Prentice Hall International, Inc, 1996, p.846

Untuk memperoleh nilai C dan R, maka dapat digunakan menggunakan penurunan rumus sebagai berikut :

$$V_A = \frac{1/j\omega C}{R + 1/j\omega C} \times V_i \quad ; V_A = \text{tegangan pada titik A (2.24)}$$

$$= \frac{1}{1 + 1/j\omega C R} \times V_i$$

Karena $E_d = 0$ maka $V_A = V_o$

Jadi

$$V_o = \frac{1/j\omega C}{R + 1/j\omega C} \times V_i \quad ; V_o = \text{tegangan output (2.24)}$$

$$A_{CL} = \frac{V_{out}}{V_{in}} = \frac{1}{1 + j\omega RC}$$

Sebagai syarat nilai dari \bar{A}_{CL} adalah 0,707 dengan sudut fasa -45° dan $\omega = \omega_c$, sehingga menjadi

$$\bar{A}_{CL} = \frac{1}{1 + R^2 \omega^2 C^2} - j \frac{\omega RC}{1 + R^2 \omega^2 C^2}$$

$$\text{Arc tan } \frac{\text{Im}}{\text{Re}} = \text{sudut fasa } -45^\circ$$

$$\text{Arc tan } -\omega_c RC = \text{sudut fasa } -45^\circ$$

$$\omega_c RC = \tan 45^\circ$$

$$\omega_c = \frac{1}{RC}$$

dan menjadi

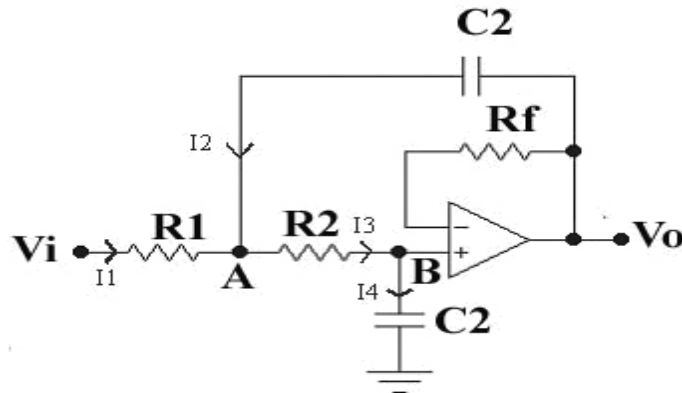
$$f_c = \frac{1}{2\pi RC} \quad (2.25)$$

Untuk menghasilkan sebuah *low pass filter orde satu* yang baik, maka prosedur desain yang perlu diikuti adalah :

1. Menentukan frekuensi *cut-off* yang hendak digunakan.
2. Memilih nilai resistor (R,Rf) untuk perancangan di atas dengan mempertimbangkan hal praktis di antara 10 KΩ sampai dengan 100 KΩ.
3. Menghitung nilai kapasitor dari *filter* tersebut dengan menggunakan persamaan (2.25).
4. Mengambil nilai terdekat untuk kapasitor dan hitung kembali nilai resistor nya.

2.3.1.2. Active Low Pass Filter Orde Kedua

Pada *active low pass filter orde kedua* ini menghasilkan penurunan *gain* frekuensi lebih besar dua kali daripada yang terjadi pada *orde pertama*. Untuk gambar dari rangkaian *active low pass filter orde kedua* ini dapat dilihat pada gambar 2.14.



Gambar 2.14. Active Low Pass Filter Orde Kedua

Sumber: Floyd, Thomas L. *Electronic Devices*. 4th ed. New Jersey: Prentice Hall International, Inc, 1996, p.847

Persamaan untuk mencari frekuensi *cut-off* dapat didapatkan dengan menggunakan persamaan berikut:

$$I_1 = \frac{V_i - V_A}{R_1} ; \quad I_2 = \frac{V_o - V_A}{\frac{1}{j\omega C_2}} ; \quad (2.26)$$

$$I_3 = \frac{V_A - V_B}{R_2} ; \quad I_4 = \frac{V_B}{\frac{1}{j\omega C_1}} ; \quad (2.27)$$

$E_d = 0$, sehingga $V_B = V_o$, maka :

$$I_3 = \frac{V_A - V_o}{R_2} ; \quad I_4 = j\omega C_1 V_o ; \quad I_3 = I_4 ;$$

$$\frac{V_A - V_o}{R_2} = j\omega C_1 V_o \quad (2.28)$$

$$V_A = V_o + j\omega C_1 R_2 V_o$$

$$I_1 + I_2 = I_3 \quad (2.29)$$

$$\frac{V_i - V_A}{R_1} + (V_o - V_A) j\omega C_2 = \frac{V_A - V_B}{R_2}$$

sehingga didapatkan persamaan sebagai berikut :

$$\frac{V_o}{V_i} = \frac{1}{(1 - \omega^2 C_1 C_2 R_1 R_2) + j(\omega C_1 R_1 + \omega C_1 R_2)} = A_{CL} \quad (2.30)$$

Jika $R_1 = R_2 = R$ dan $C_2 = 2C_1$ maka didapatkan rumus :

$$A_{CL} = \frac{1}{(1 - 2\omega^2 \cdot C1^2 R^2) + (2\omega \cdot C1 \cdot R)} \quad (2.31)$$

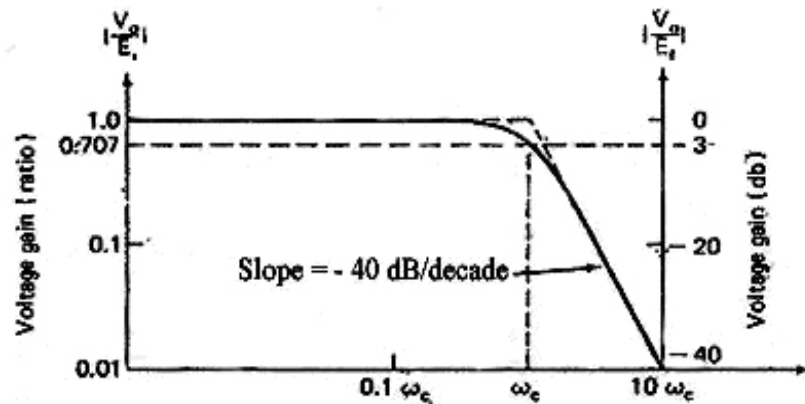
Syarat frekuensi *cut-off* adalah $\omega = \omega_c$ dan $A_{cl} = 0.707$ sehingga:

$$1 - 2\omega_c^2 C1^2 R^2 = 0;$$

$$\omega_c = \frac{0.707}{C1 \cdot R} \quad (2.32)$$

$$f_c = \frac{0.707}{6,28 \cdot C1 \cdot R} \quad (2.33)$$

Pada *orde* kedua ini terjadi penurunan *gain* frekuensi di atas frekuensi *cut-off* sebesar -40 dB/decade. Gambar respon frekuensi dari *orde* kedua *active low pass filter* dapat dilihat pada gambar 2.15 :



Gambar 2.15. Respon Frekuensi *Orde* Kedua *Active Low Pass Filter*

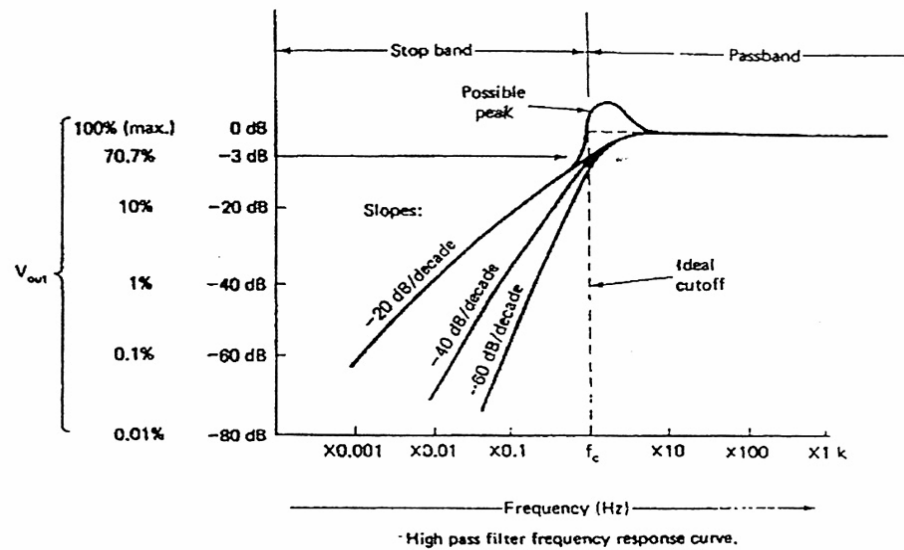
Sumber: Coughlin, Robert F. *Amplifiers and Linear Integrated Circuits*. 2th ed. New Jersey: Prentice Hall International, Inc, 1996, p.254

Untuk menghasilkan sebuah *low pass filter orde* dua yang baik, maka prosedur desain yang perlu diikuti adalah :

1. Menentukan frekuensi *cut-off* yang hendak digunakan.
2. Memilih nilai resistor (R, R_f) untuk perancangan di atas dengan mempertimbangkan hal praktis di antara $10 \text{ K}\Omega$ sampai dengan $100 \text{ K}\Omega$.
3. Menghitung nilai kapasitor dari *filter* tersebut dengan menggunakan persamaan (2.33).
4. Mengambil nilai terdekat untuk kapasitor dan hitung kembali nilai resistor nya.

2.3.2. High Pass Filter

Filter high pass adalah sebuah rangkaian yang berfungsi meneruskan tegangan pada frekuensi tinggi (lebih dari f_c) dan menghambat tegangan pada frekuensi rendah (kurang dari f_c). Ketika tegangan *input* dari frekuensi tinggi ke rendah telah melewati batas frekuensi *cut-off* (f_c) maka amplitudo dari tegangan *output* akan melemah secara berkala. Berikut ini adalah gambar grafik tegangan *output* dari sebuah *High pass filter* terhadap frekuensi :

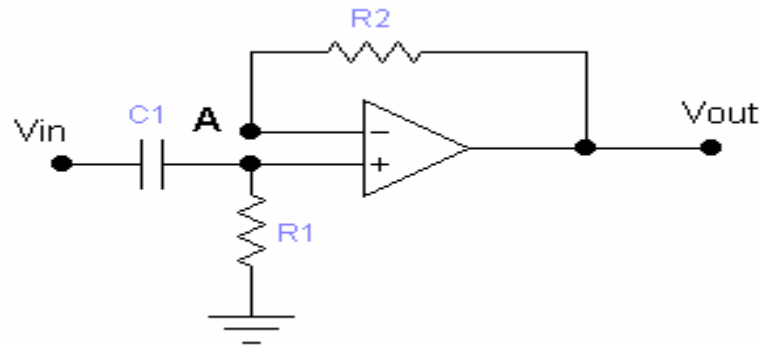


Gambar 2.16. Kurva Tanggapan Frekuensi *High Pass Filter*

Sumber: Coughlin, Robert F. *Amplifiers and Linear Integrated Circuits*. 2th ed. New Jersey: Prentice Hall International, Inc, 1982, p.253

2.3.2.1. Active High Pass Filter Orde Pertama

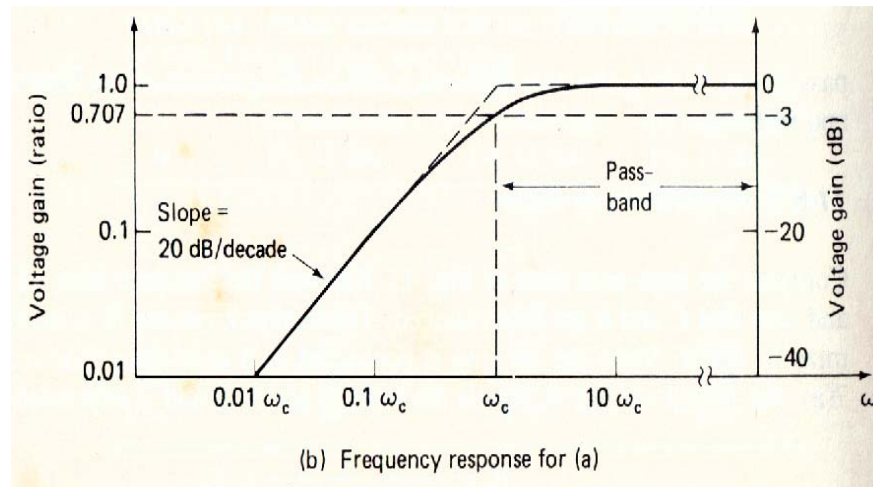
Active high pass filter orde pertama ini menggunakan *Op-Amp* yang mempunyai konfigurasi *non-inverting* (tidak membalik sinyal *input* yang masuk) dengan *input*-nya yang berupa sebuah kapasitor yang dipasang secara seri dan satu buah resistor yang dipasang secara paralel dengan *ground*. Gambar dari rangkaian *active high pass filter orde pertama* ini terlihat pada gambar 2.17.



Gambar 2.17. *Active High Pass Filter Orde Pertama*

Sumber: Floyd, Thomas L. *Electronic Devices*. 4th ed. New Jersey: Prentice Hall International, Inc, 1996, p.851

Pada *orde* pertama ini terjadi kenaikan *gain* frekuensi di bawah frekuensi *cut-off* sebesar 20 dB/decade. Gambar respon frekuensi dari *orde* pertama *active high pass filter* dapat dilihat pada gambar 2.18 :



Gambar 2.18. Respon Frekuensi *Orde Pertama Active High Pass Filter*

Sumber: Coughlin, Robert F. *Amplifiers and Linear Integrated Circuits*. 2th ed. New Jersey: Prentice Hall International, Inc, 1982, p. 286

Untuk memperoleh nilai C dan R maka dapat digunakan menggunakan penurunan rumus sebagai berikut :

$$V_A = \frac{R}{R + \frac{1}{j\omega C}} \times V_i \quad ; V_A = \text{tegangan pada titik A} \quad (2.34)$$

$$= \frac{1}{1 + \frac{1}{j\omega cR}} x V_i$$

Karena $E_d = 0$ maka $V_A = V_o$

Jadi

$$V_o = \frac{1}{1 - j \frac{1}{\omega cR}} x V_i \quad ; V_o = \text{tegangan output} \quad (2.35)$$

$$A_{CL} = \frac{V_{out}}{V_{in}} = \frac{1}{1 - j \frac{1}{\omega cR}}$$

Sebagai syarat nilai dari \bar{A}_{CL} adalah 0,707 dengan sudut fasa -45° dan $\omega = \omega_c$, sehingga menjadi

$$\bar{A}_{CL} = \frac{1}{1 + R^2 \omega^2 C^2} - j \frac{\omega RC}{1 + R^2 \omega^2 C^2}$$

$$\text{Arc tgn} \frac{\text{Im}}{\text{Re}} = \text{sudut fasa } -45^\circ$$

$$\text{Arc tgn } -\omega_c RC = \text{sudut fasa } -45^\circ$$

$$\omega_c RC = \text{tgn } 45^\circ$$

$$\omega_c = \frac{1}{RC}$$

dan menjadi

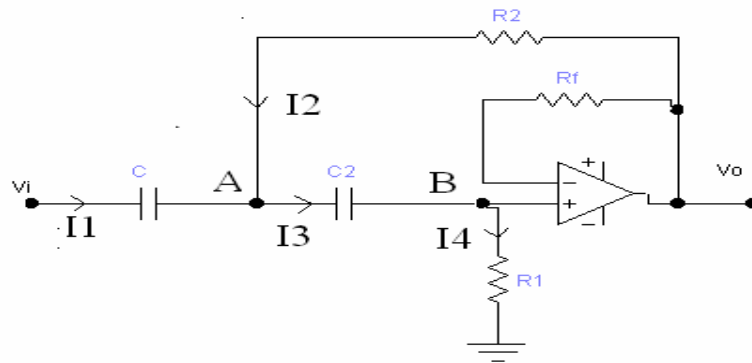
$$f_c = \frac{1}{2\pi RC} \quad (2.36)$$

Untuk menghasilkan sebuah *high pass filter orde satu* yang baik, maka prosedur desain yang perlu diikuti adalah :

1. Menentukan frekuensi *cut-off* yang hendak digunakan.
2. Memilih nilai resistor (R, R_f) untuk perancangan diatas dengan mempertimbangkan hal praktis diantara $10 \text{ K}\Omega$ sampai dengan $100 \text{ K}\Omega$.
3. Menghitung nilai kapasitor dari *filter* tersebut dengan menggunakan persamaan (2.36).
4. Mengambil nilai terdekat untuk kapasitor dan hitung kembali nilai resistor nya.

2.3.2.2. Active High Pass Filter Orde Kedua

Pada *active high pass filter orde kedua* ini menghasilkan penurunan *gain* frekuensi lebih besar dua kali daripada yang terjadi pada *orde pertama*. Untuk gambar dari rangkaian *active high pass filter orde kedua* ini dapat dilihat pada gambar 2.20.



Gambar 2.19. Active High Pass Filter Orde Kedua

Persamaan untuk mencari frekuensi *cut-off* dapat didapatkan dengan menggunakan persamaan berikut:

$$I1 = \frac{Vi - VA}{\frac{1}{j\omega C}} ; \quad I2 = \frac{Vo - VA}{R} ; \quad (2.37)$$

$$I3 = \frac{VA - VB}{\frac{1}{j\omega C2}} ; \quad I4 = \frac{VB}{R1} ; \quad (2.38)$$

$E_d = 0$, sehingga $VB = Vo$ maka :

$$I3 = \frac{VA - Vo}{\frac{1}{j\omega C2}} ; \quad I4 = \frac{Vo}{R1} ; \quad I3 = I4;$$

$$\frac{VA - Vo}{\frac{1}{j\omega C2}} = \frac{Vo}{R1} \quad (2.39)$$

$$(VA - Vo) R1 = Vo \frac{1}{j\omega C2}$$

$$I1 + I2 = I3 \quad (2.40)$$

sehingga didapatkan persamaan sebagai berikut :

$$\frac{V_o}{V_i} = \frac{1}{\left(1 - \frac{1}{\omega^2 \cdot C_1 \cdot C_2 \cdot R_1 \cdot R_2}\right) + j \frac{1}{\omega R_1} \left(\frac{1}{C_1} + \frac{1}{C_2}\right)} = A_{CL} \quad (2.41)$$

Jika $R_2 = 0.5 R$ dan $C_2 = C_1 = C$ maka didapatkan rumus :

$$A_{CL} = \frac{1}{\left(1 - \frac{2}{\omega^2 \cdot C^2 R_1}\right) + j \frac{2}{\omega \cdot C \cdot R_1}} \quad (2.42)$$

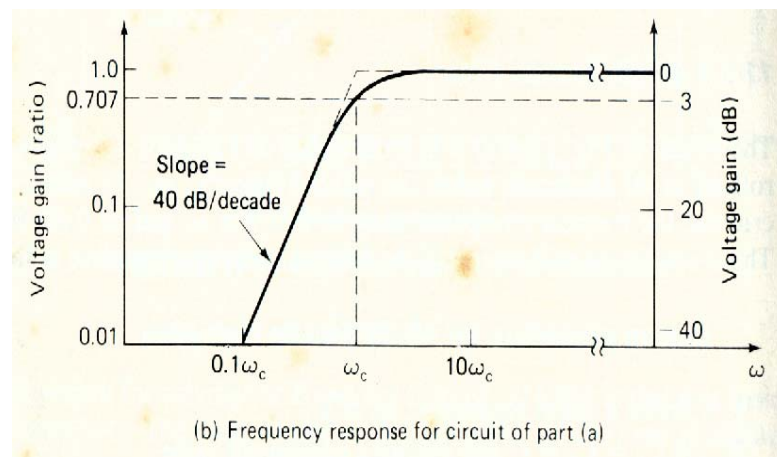
Syarat frekuensi *cut-off* adalah $\omega = \omega_c$ dan $A_{cl} = 0.707$ sehingga:

$$1 - \frac{2}{\omega_c^2 C^2 R_1} = 0;$$

$$\omega_c = \frac{1.414}{C \cdot R_1} \quad (2.43)$$

$$f_c = \frac{1.414}{6,28 \cdot C \cdot R_1} \quad (2.44)$$

Pada *orde* kedua ini terjadi kenaikan *gain* frekuensi di sebelum frekuensi *cut-off* sebesar 40 dB/decade. Gambar respon frekuensi dari *orde* kedua *active high pass filter* dapat dilihat pada gambar 2.20 :



Gambar 2.20. Respon Frekuensi *Orde* Kedua *Active High Pass Filter*

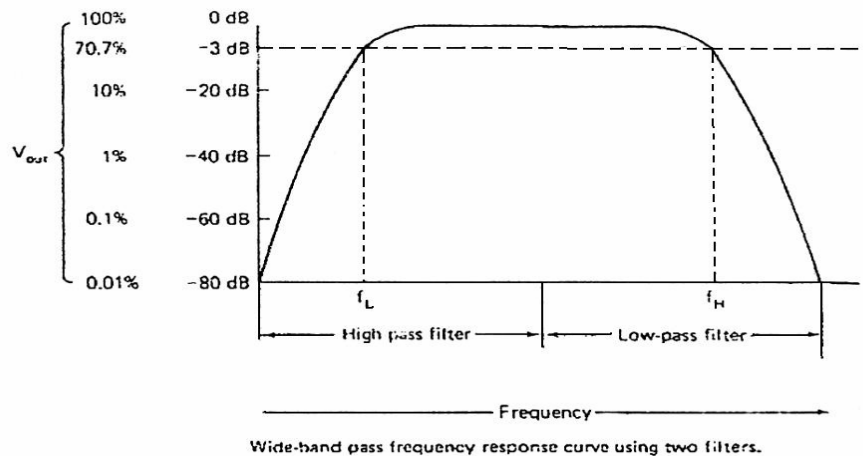
Sumber: Coughlin, Robert F. *Amplifiers and Linear Integrated Circuits*. 2th ed. New Jersey: Prentice Hall International, Inc, 1982, p.288

Untuk menghasilkan sebuah *high pass filter orde dua* yang baik, maka prosedur desain yang perlu diikuti adalah :

1. Menentukan frekuensi *cut-off* yang hendak digunakan.
2. Memilih nilai resistor (R, R_f) untuk perancangan di atas dengan mempertimbangkan hal praktis diantara $10\text{ K}\Omega$ sampai dengan $100\text{ K}\Omega$.
3. Menghitung nilai kapasitor dari *filter* tersebut dengan menggunakan persamaan (2.44).
4. Mengambil nilai terdekat untuk kapasitor dan hitung kembali nilai resistornya.

2.3.3. Band Pass Filter

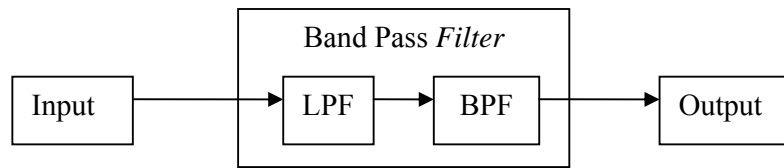
Sebuah *band pass filter* akan melewati suatu grup frekuensi tertentu dan menolak frekuensi yang lainnya. Pada dasarnya rangkaian *band pass* merupakan penggabungan dari rangkaian *low pass filter* dengan rangkaian *high pass filter*.



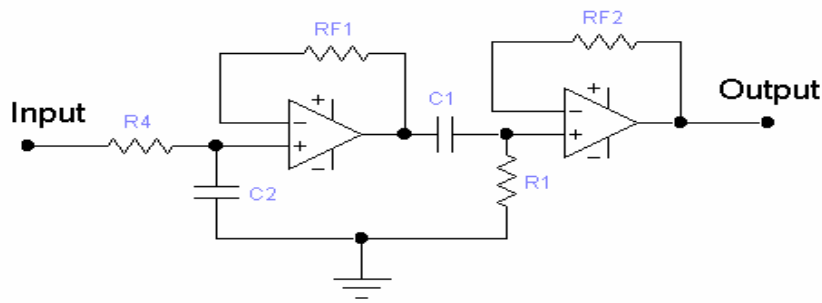
Gambar 2.22. Kurva Frekuensi Saat Menggabungkan Dua *Filter*

Sumber : Fredrick W. Hughes. *Op-amp Handbook*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1981. p.75

Untuk membuat rangkaian *band pass filter* dengan cara menghubungkan rangkaian *low pass filter* dengan rangkaian *high pass filter* secara seri seperti pada gambar 2.23.



Gambar 2.22. Blok Diagram *Band Pass Filter* Orde Pertama

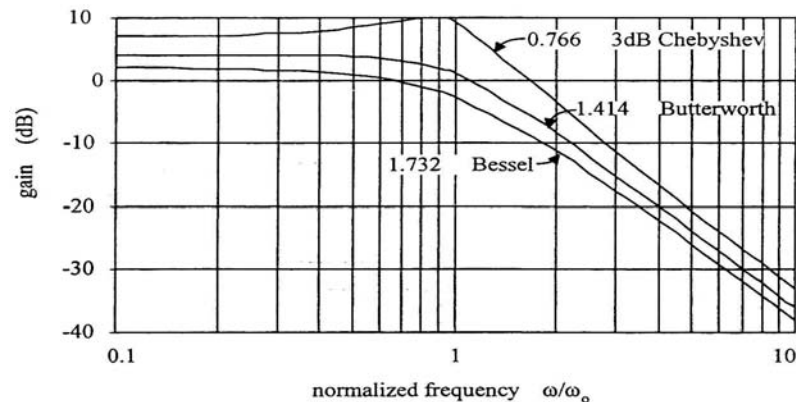


Gambar 2.23. Rangkaian *Band Pass Filter* Orde Pertama

Untuk perhitungan *low pass filter* dan *high pass filter* dapat dilihat pada sub bab 2.3.1 dan 2.3.2.

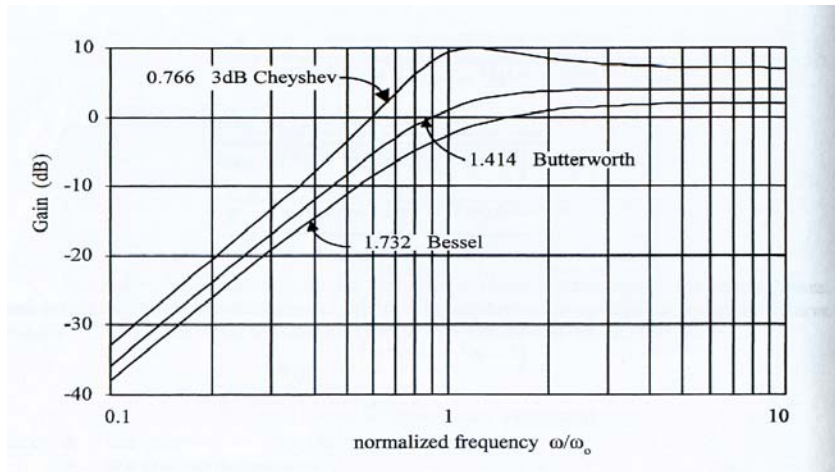
2.3.4. Karakteristik Dari *Active Filter*

Active filter ini mempunyai 3 macam karakteristik yaitu *Butterworth*, *Chebyshev* dan *Bessel*. Tiap-tiap karakteristik *active filter* ini mempunyai respon frekuensi yang berbeda antara yang satu dengan yang lain. Gambar dari respon frekuensi grafik karakteristik dapat dilihat pada gambar 2.24.



Gambar 2.24. Grafik Respon Frekuensi Karakteristik *Active Low Pass Filter*

Sumber: Jacob, Michael J. *Analog Integrated Circuit Applications*. New Jersey: Prentice-Hall, Inc. 2000, p.418



Gambar 2.25. Grafik Respon Frekuensi Karakteristik *Active High Pass Filter*

Sumber: Jacob, Michael J. *Analog Integrated Circuit Applications*. New Jersey: Prentice-Hall, Inc. 2000, p.419

Active filter yang menggunakan karakteristik *Butterworth*, *Chebyshev* dan *Bessel* mempunyai suatu parameter yaitu *Damping Factor* (DF). *Damping Factor* dari suatu rangkaian *active filter* menentukan karakteristik respon mana yang ditunjukkan oleh *filter*. *Damping Factor* ini akan mempengaruhi respon *filter* melalui reaksi *negative feedback*.

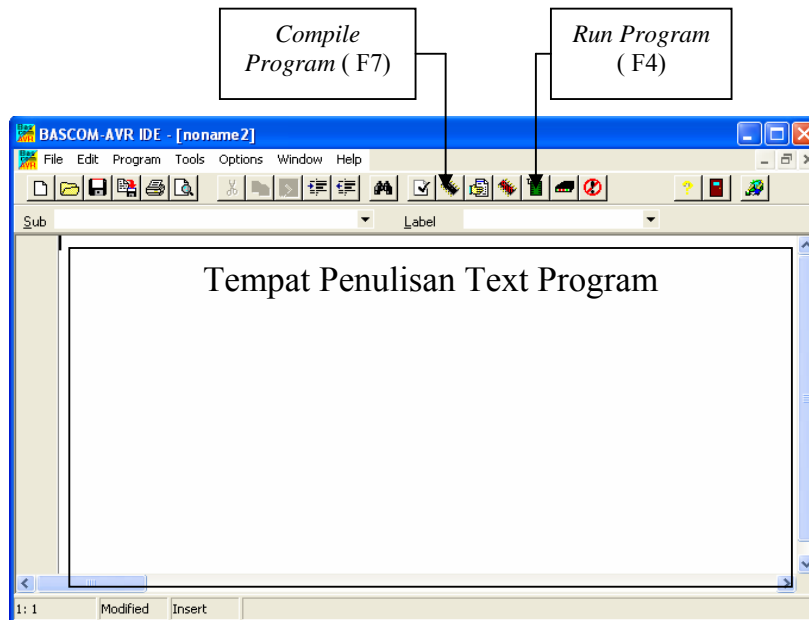
2.4. BASCOM AVR

BASCOM AVR merupakan singkatan dari *Basic Compiler AVR*. BASCOM AVR termasuk dalam program *microcontroller* buatan MCS *Electronics* yang mengadaptasi bahasa tingkat tinggi yang sering digunakan oleh awam (Bahasa *Basic*). Dengan menggunakan bahasa pemrograman tingkat tinggi, maka pemrogram mendapatkan banyak kemudahan dalam mengatur sistem kerja dari *microcontroller* (terutama apabila tipe *microcontroller* yang digunakan masih baru dan tidak terlalu dikenal).

BASCOM AVR memiliki program sendiri untuk memasukkan program yang telah dikompilasi ke dalam AVR. Berikut *setting* awal sebelum digunakan untuk mengisi atau melakukan *compile*

2.4.1. Pengaturan *Programmer*

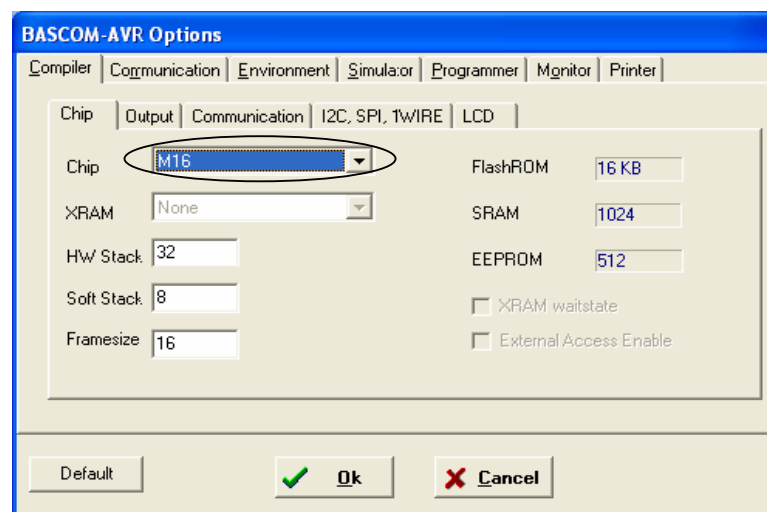
- a) Masuk program Bascom AVR, maka akan muncul tampilan pada layar seperti gambar berikut :



Gambar 2.26. Tampilan Model BASCOM AVR

- b) Mengatur sistem *microcontroller compiler* :

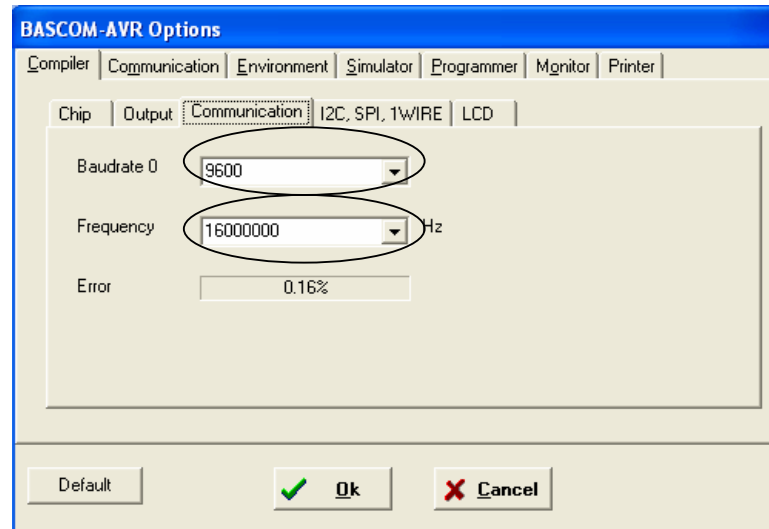
- pengaturan tipe dari *microcontroller*
 untuk pengaturan tipe *microcontroller* klik *option* → *compiler* → *chip*.
 Setelah itu tampilan layar seperti pada gambar 2.27. Untuk tipe *microcontroller* ATMEGA 16 menggunakan *chip* M16.



Gambar 2.27. Pengaturan Jenis *Microcontroller* yang Digunakan

- pengaturan *baudrate* dan frekuensi

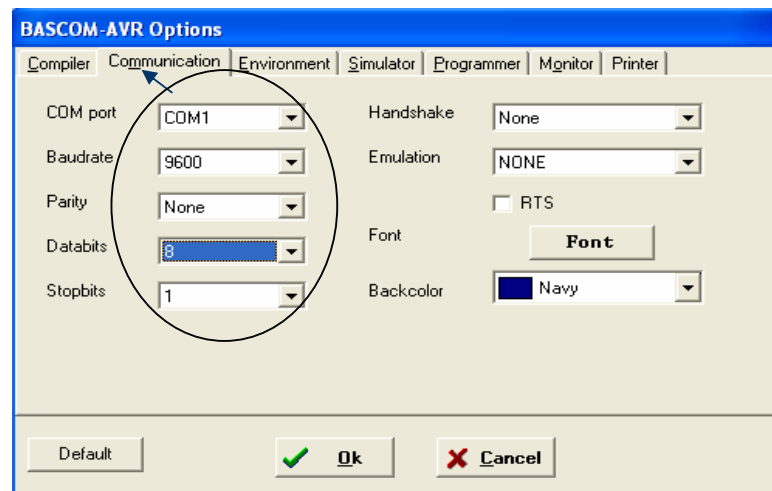
Klik “Communication” pada bagian *compiler*, maka tampilan layar akan seperti gambar 2.28. Pada gambar dapat dilihat kecepatan kirim sebesar 9600 dengan frekuensi 16 MHz. Besarnya frekuensi ini disesuaikan dengan besarnya *crystal* pada *hardwere microcontroller*.



Gambar 2.28. Pengaturan *Baudrate* dan Frekuensi

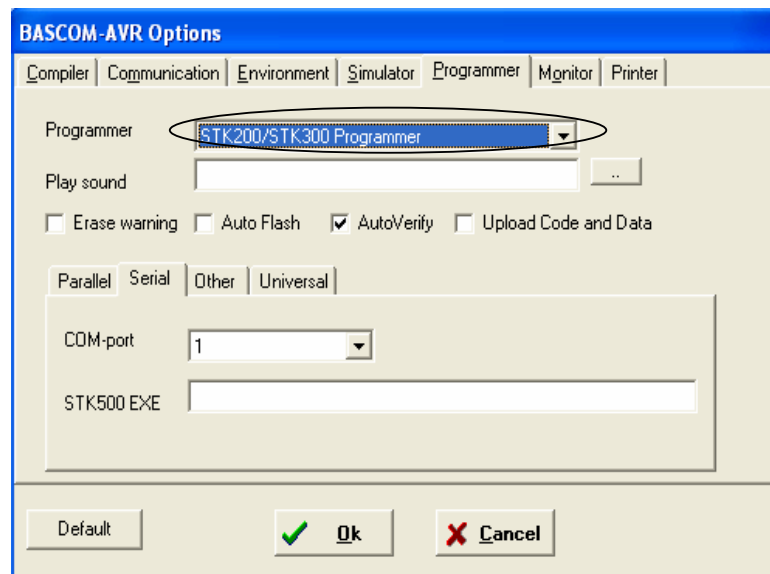
- c) Pengaturan komunikasi data

Klik “Comunikation” seperti pada gambar 2.29. Karena kabel *port* menggunakan menggunakan soket DB 25, maka COM yang digunakan adalah COM1 dan selanjutnya *baudrate*, *parity*, *databits*, *stopbit* di-*set* seperti pada gambar :



Gambar 2.29. Pengaturan *Port* dari Alat Pengisi

- d) Klik “*Programmer*”, untuk ATMEGA 16 yang menggunakan kabel ISP, konfigurasi dari *programmer* harus diubah menjadi STK300/STK200.

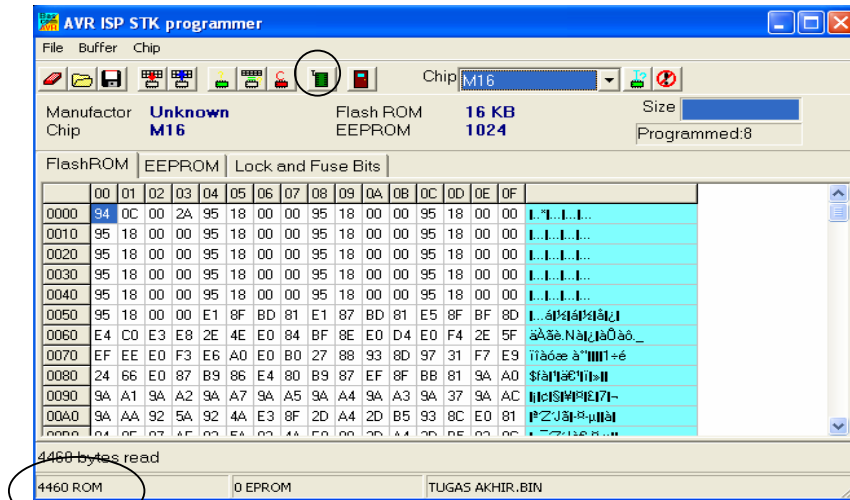


Gambar 2.30. Pengaturan Konfigurasi Dari *Programmer*

2.4.2. Langkah Pengisian *Microcontroller*

Langkah pengisian program ke dalam *microcontroller* sebagai berikut :

- a) Pastikan penghubung antara *hardwere* pengisi dengan *port* komputer telah terhubung dan nyalakan *supply* tegangan pengisi
- b) Pastikan posisi *pin* IC *microcontroller* sesuai pada tempat pengisinya.
- c) Selanjutnya setelah penulisan *text* program selesai, maka program di-*compile* (F7). Selanjutnya akan terlihat apakah penulisan prosedur program sudah benar atau ada kesalahan. Jika masih ada kesalahan maka pada sisi bawah layar akan muncul keterangan dari kesalahan program.
- d) Setelah program sudah tidak ada kesalahan, maka program mulai dijalankan (F4).
- e) Setelah program dijalankan, maka akan muncul tampilan layar seperti gambar 2.31 :

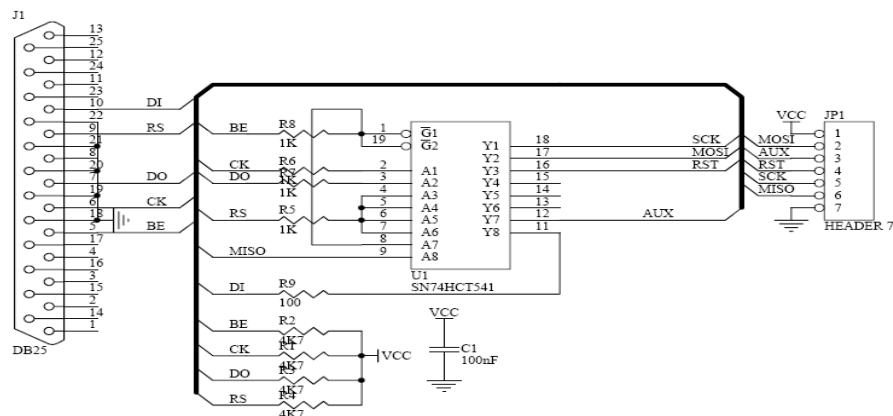


Gambar 2.31. Tampilan Model BASCOM AVR

Selanjutnya pada layar dapat dilihat besarnya *memory* dari program yang dibuat (pada contoh di atas besarnya 4460 *bytes*). Selanjutnya program dapat diisi ke-*microcontroller* secara langsung dengan menekan *Auto Program Chip*, tunggu beberapa waktu sampai selesai proses pengisian.

2.4.3. Pengisi Program *Microcontroller*

Untuk pengisi *microcontroller* menggunakan media kabel AT89ISP. Alat pengisi ini menggunakan IC 74HCT541 yang terhubung antara soket dB 25 dan pada JP1 terhubung pada *pin* ATMEGA16. Berikut skema dari alat pengisi *microcontroller* :

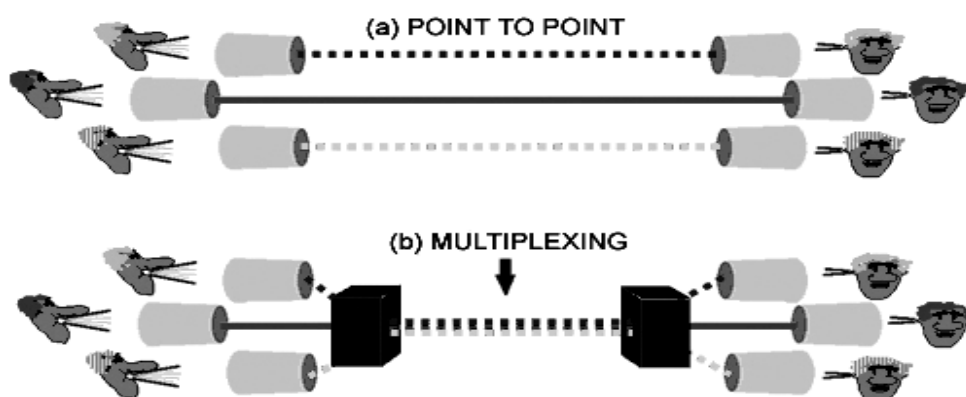


Gambar 2.32. Skema Rangkaian Pengisi AVR

Sumber: ATMEL Corporation, *AT89ISP Programmer Cable Datasheet*, Juli 2005
 <http://www.Atmel.com/dyn/recources/pro_documents/DOC0839.pdf>

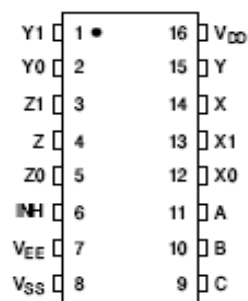
2.5. Multiplexing

Transmisi sinyal dapat dilakukan dengan 2 cara yaitu secara *point to point* dan secara *multiplexing*, transmisi *point to point* memerlukan satu buah jalur transmisi untuk masing – masing sinyal sehingga kurang efisien. *Multiplexing* adalah tehnik transmisi dua atau lebih sinyal pada satu buah jalur transmisi. *Multiplexing* mulai digunakan karena kesederhanaannya dan efektifitas transmisi sinyal yang tinggi. Transmisi berikut ini adalah perbedaan transmisi sinyal antara *point to point* dengan *multiplexing* :



Gambar 2.33. Transmisi Sinyal secara *Point to Point* (a) dan *Multiplexing* (b)

IC 4053 merupakan salah satu IC yang menerapkan dari sistem *multiplexing*. Pada 4053 memiliki 6 *input* dan 3 *output* yang pengaturan transmisi diatur oleh *pin* C, B, A. Berikut gambar dari konfigurasi *pin* dan tabel kebenaran dari IC MC14053 :



Gambar 2.34. Konfigurasi *Pin* MC14053

Sumber: On Semiconductor, *MC14053 Data Sheet*, 22 Okt 2005
 <<http://www.onsemi.com/resources/datasheet/MC14051.pdf>>.

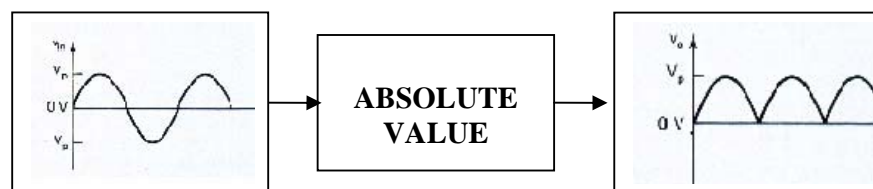
| E | C | B | A | On Switch | | |
|---|---|---|---|-----------|----|----|
| | | | | Z0 | Y0 | X0 |
| 0 | 0 | 0 | 0 | Z0 | Y0 | X0 |
| 0 | 0 | 0 | 1 | Z0 | Y0 | X1 |
| 0 | 0 | 1 | 0 | Z0 | Y1 | X0 |
| 0 | 0 | 1 | 1 | Z0 | Y1 | X1 |
| 0 | 1 | 0 | 0 | Z1 | Y0 | X0 |
| 0 | 1 | 0 | 1 | Z1 | Y0 | X1 |
| 0 | 1 | 1 | 0 | Z1 | Y1 | X0 |
| 0 | 1 | 1 | 1 | Z1 | Y1 | X1 |
| 1 | X | X | X | | | |

Gambar 2.35. Tabel Kebenaran MC14053

Sumber: On Semiconductor, *MC14053 Data Sheet*, 22 Okt 2005
<http://www.onsemi.com/recources/datasheet/MC14051.pdf>.

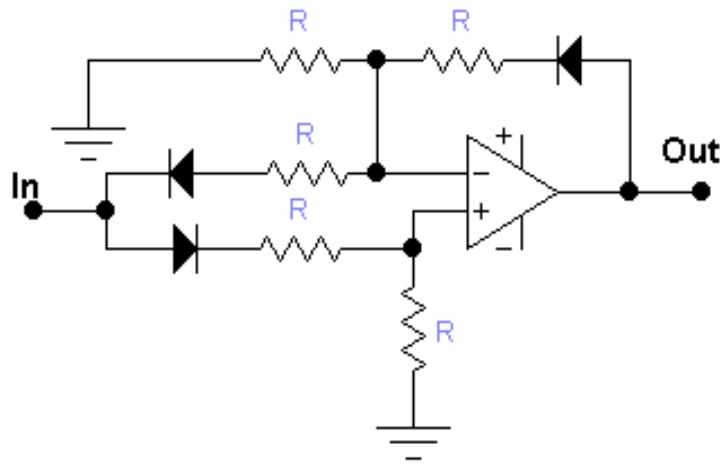
2.6. *Absolute Value*

Pada dasarnya *input ADC internal microcontroller ATMEGA 16* memiliki *input* berupa tegangan DC sedang *output audio* berupa *signal sinus*, maka untuk mengubah *signal sinus* menjadi bentuk DC salah satunya dapat menggunakan rangkaian *absolute-value*. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini :



Gambar 2.36. Fungsi Rangkaian *Absolute Value*

Rangkaian dari *absolute value* terdiri dari IC *Op-Amp*, dioda, dan resistor. Adapun rangkaian dari *absolute value* adalah sebagai berikut :

Gambar 2.37. Rangkaian *Absolute Value*

Dari rangkaian absolute value memiliki perhitungan dan persamaan rumus sebagai berikut :

$$V_1 = \frac{V_P - V_{D1}}{2} \quad (2.45)$$

$$V_2 = \frac{V_o(+)-V_{D3}}{2} \quad (2.46)$$

$$V_d = 0$$

$$V_1 = V_2$$

maka ;

$$\frac{V_P - V_{D1}}{2} = \frac{V_o(+)-V_{D3}}{2} \quad (2.47)$$

$$V_o(+)= V_p$$

dimana ;

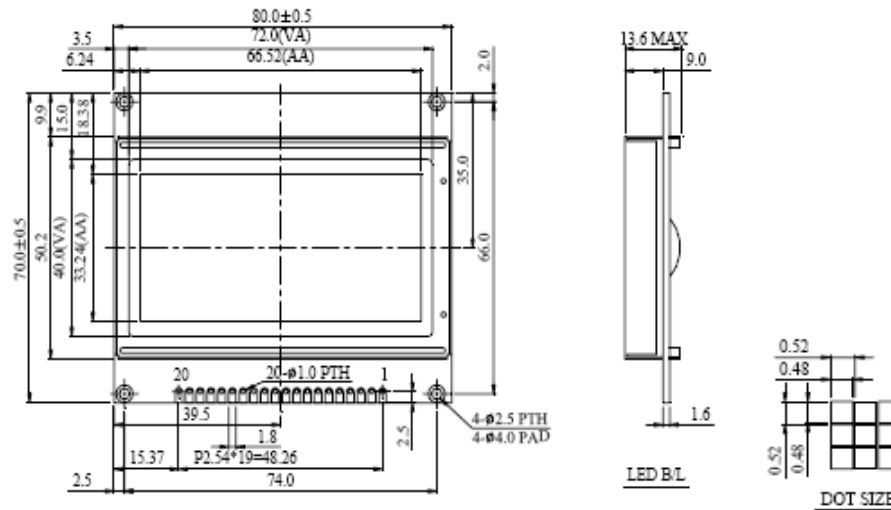
V_p = tegangan *input* positif

V_o = tegangan *output* positif *half-cycle*

$V_{D1} = V_{D3} = drop$ tegangan diode = 0.7 V

2.7. Graphic Liquid Crystal Display 128 X 64

Graphic Liquid crystal display ini memiliki *dot* sebanyak 128 X 64, sehingga memungkinkan untuk membentuk tampilan gambar yang bervariasi. Berikut dimensi dari *Graphic LCD* tipe G12864C-A dan besarnya *dots* atau *pixel* pada LCD:



Gambar 2.38. Dimensi *Graphic LCD*

Sumber: LCD *Graphic*. CGM12864CB-4. *Users Manual*.2005. p.6

Kegunaan dan penjelasan dari masing – masing *pin* sebagai berikut :

- *Pin* 1 dan 2 terhubung pada *power supply* dengan besar tegangan *input* sebesar 5 volt
- *Pin* 3 berfungsi sebagai pengatur *contrast* dari *display LCD* memiliki *range* tegangan antara 0 sampai -5 volt. Untuk mempermudah pengaturannya diperlukan sebuah potensio yang terhubung pada *ground* dan *pin* 18 (*Vout*)
- *Pin* 4 berfungsi untuk memilih perintah melakukan data atau instruksi. Jika diberi logic *high*, maka *pin* akan berfungsi sebagai data. Apabila diberi logic *low*, maka *pin* berfungsi sebagai instruksi.
- *Pin* 5 berfungsi untuk memberi kondisi pada *Graphic LCD*. Jika *high*, maka *Graphic LCD* melakukan pembacaan data pada LCD. Jika *low*, maka LCD berfungsi sebagai *write*
- *Pin* 6 merupakan *enable signal* untuk mengaktifkan fungsi program

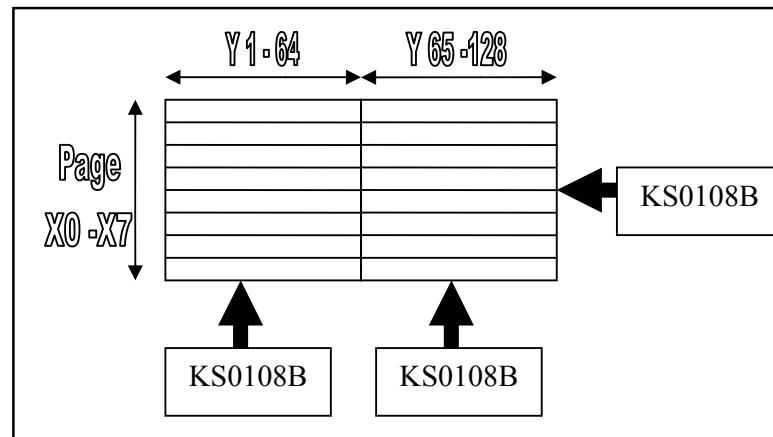
- *Pin 7* sampai 14 merupakan 8 *bit* untuk *input* data
- *Pin 15* (CS1) merupakan *chip select* untuk kolom 1 sampai 64
- *Pin 16* (CS2) merupakan *chip select* untuk kolom 64 sampai 128
- *Pin 17* berfungsi sebagai *reset* untuk program
- *Pin 18* menghasilkan tegangan – 5, berfungsi untuk pemberi *power supply* *pin 3* (*V_o*) sebagai pengatur *contrast* LCD
- *Pin 19* dan 20 dihubungkan dengan *power supply*, besarnya tegangan mempengaruhi terang redupnya *backlight* LCD grafik. Untuk *pin 19* dihubungkan pada positif *power supply* sedang *pin 20* terhubung pada *ground*.

| Pin No. | Symbol | Level | Description |
|---------|-----------------|------------|--|
| 1 | VSS | 0V | Ground |
| 2 | V _{DD} | 5.0V | Supply voltage for logic |
| 3 | V _o | (Variable) | Operating voltage for LCD |
| 4 | D/I | H/L | H: Data , L: Instruction |
| 5 | R/W | H/L | H: Read (MPU←Module) , L :Write (MPU→Module) |
| 6 | E | H | Enable signal |
| 7 | DB0 | H/L | Data bit 0 |
| 8 | DB1 | H/L | Data bit 1 |
| 9 | DB2 | H/L | Data bit 2 |
| 10 | DB3 | H/L | Data bit 3 |
| 11 | DB4 | H/L | Data bit 4 |
| 12 | DB5 | H/L | Data bit 5 |
| 13 | DB6 | H/L | Data bit 6 |
| 14 | DB7 | H/L | Data bit 7 |
| 15 | CS1 | L | Select Column 1 ~ Column 64 |
| 16 | CS2 | L | Select Column 65 ~ Column 128 |
| 17 | RST | L | Reset signal |
| 18 | Vout | | Negative Voltage |
| 19 | A | — | Power supply for LED backlight (+) |
| 20 | K | — | Power supply for LED backlight (-) |

Gambar 2.39. Fungsi Masing – Masing *Pin Graphic* LCD

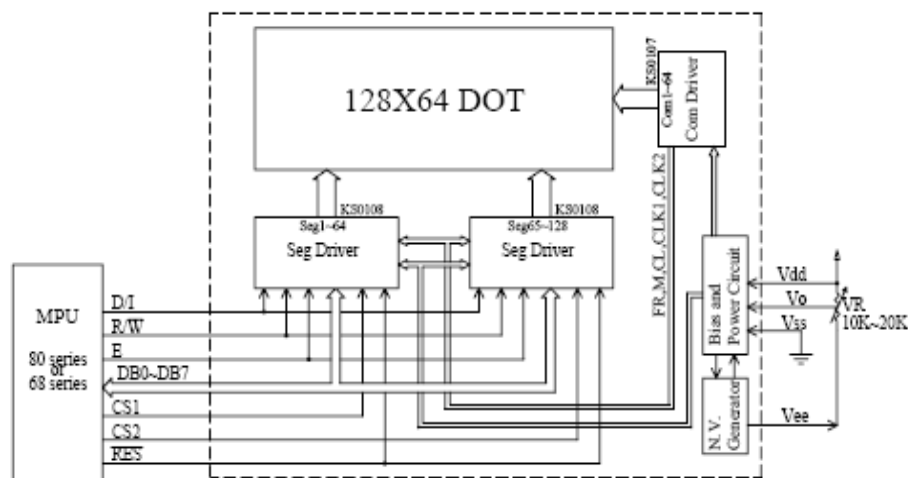
Sumber: LCD *Graphic*. CGM12864CB-4. *Users Manual*.2005. p.5

Pada *graphic LCD* ini ini menggunakan 3 buah IC *driver* KS0108B (64CH *Segment Driver for Dot Matrix LCD*) yang memiliki fungsi untuk membentuk *dot* sebanyak 128 X 64. Untuk IC pertama berfungsi mengatur banyaknya kolom 1 sampai 64, sedang IC kedua mengatur kolom 65 sampai 128. Untuk IC ketiga berfungsi untuk mengatur baris 1 sampai 64. untuk pembagian kerja IC KS0108B dan keseluruhan blok diagram dari *graphic LCD* adalah sebagai berikut :



Gambar 2.40. Pembagian IC KS0108B

Sumber: LCD *Graphic*. CGM12864CB-4. *Users Manual*.2005. p.6



Gambar 2.41. Blok Diagram *Graphic LCD*

Sumber: LCD *Graphic*. CGM12864CB-4. *Users Manual*.2005. p.6

Untuk pemrograman dari LCD grafik pengoperasian dari *graphic* LCD ini dibagi menjadi dua bagian yaitu bagian kiri (kolom 1 sampai 64) dan bagian kanan (kolom 65 sampai 128)

Untuk tahapan dalam mengoperasikan LCD sebagai berikut :

- Aktifkan *display*
- Aktifkan bagian *display* (kolom 1 – 64 atau kolom 65 - 128)
- Atur letak Y
- Atur letak X
- Atur mulai dari sumbu Y
- Kirim intruksi
- Kirim data
- tampilkan

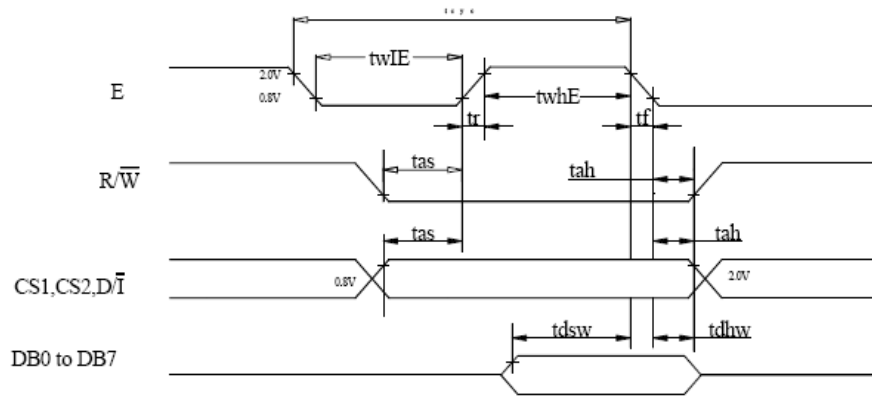
Untuk lebih jelasnya perintah atau intruksi dari LCD dapat dilihat pada tabel di bawah ini :

| Instruction | D/I | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Function |
|----------------------|-----|-----|------------------|-----|--------------------------|-----------------------|-----|--|-----|--|--|
| Display ON/OFF | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0/1 | Controls the display on or off. Internal status and display RAM data are not affected. 0:OFF, 1:ON |
| Set Address | 0 | 0 | 0 | 1 | Y address (0~63) | | | | | Sets the Y address in the Y address counter. | |
| Set Page (X address) | 0 | 0 | 1 | 0 | 1 | 1 | 1 | Page (0~7) | | | Sets the X address at the X address register. |
| Display Start Line | 0 | 0 | 1 | 1 | Display start line(0~63) | | | | | Indicates the display data RAM displayed at the top of the screen. | |
| Write Display Data | 1 | 0 | Display Data | | | | | Writes data (DB0:7) into display data RAM. After writing instruction, Y address is increased by 1 automatically. | | | |
| Read Display Data | 1 | 1 | Display Data | | | | | Reads data (DB0:7) from display data RAM to the data bus. | | | |
| Status Read | 0 | 1 | B U S Y | 0 | O N/ O F F | R E S E T | 0 | 0 | 0 | 0 | Read status. BUSY 0: Ready 1: In operation ON/OFF 0: Display ON 1: Display OFF RESET 0: Normal 1: Reset |

Gambar 2.42. Instruksi Control *Display*

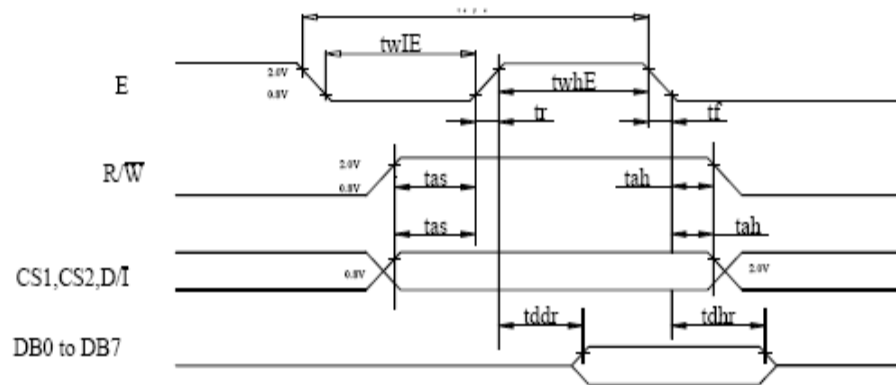
Sumber: LCD *Graphic*. CGM12864CB-4. *Users Manual*.2005. p.8

Grafik LCD memiliki 2 kondisi yaitu *write* dan *read*. Untuk melakukan tampilan ke LCD menggunakan instruksi *write*, sedang untuk pembacaan dari LCD menggunakan instruksi *read*. Berikut urutan timing perintah *write* dan *read* pada LCD grafik :



Gambar 2.43. Urutan *Write Timing*

Sumber: LCD *Graphic*. CGM12864CB-4. *Users Manual*.2005. p.7



Gambar 2.44. Urutan *Read Timing*

Sumber: LCD *Graphic*. CGM12864CB-4. *Users Manual*.2005. p.8

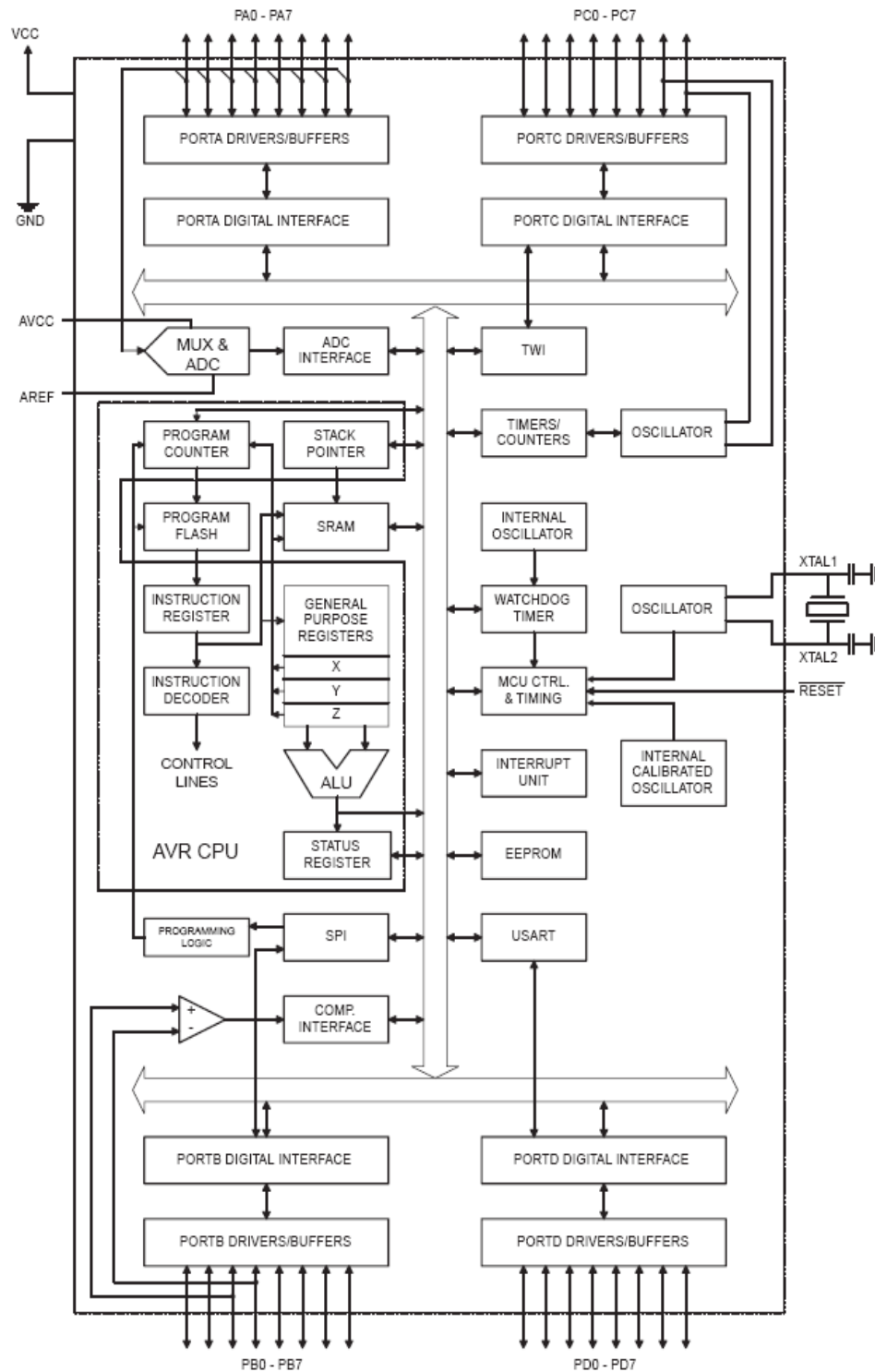
Untuk pengoperasian LCD ini langkah awal adalah mengirim instruksi lalu kemudian mengirim data. Untuk membedakan antara instruksi dan data adalah *pin* D/I. Jika diberi logic 0, maka akan berfungsi sebagai instruksi. Jika logic 1, maka berfungsi sebagai data. CS1 dan CS2 berfungsi untuk mengaktifkan sisi kanan atau sisi kiri yang aktif.

2.8. *Microcontroller ATMEGA16*

Pada aplikasi ini digunakan *microcontroller* jenis AVR yaitu ATMEGA16 produksi *ATMEL*. Keunggulan dari jenis *microcontroller* ATMEGA16 ini adalah sebagai berikut :

- 16K *bytes In-System Programmable Flash*
- 512 *bytes EEPROM*
- 1 K *bytes SRAM*
- 32 *general purpose I/O pin*
- 32 *general purpose working register*
- 1 *JTAG interface for boundry scan*
- *On chip debugging support and programing*
- 3 buah *Flexible Timer/Counter* yang dilengkapi dengan mode pembanding
- *Internal dan eksternal interrupt*
- Serial UART yang dapat diprogram
- *Watchdog Timer* yang dapat diprogram dan memiliki osilator *internal*
- *Internal ADC (Analog to Digital Converter) 10 bit*
- *Port serial jenis SPI* untuk proses *flash Memory downloading*
- 6 mode *selectable program* untuk menghemat daya

Blok diagram dan *pin* dari ATMeEGA16 dapat dilihat pada gambar 2.45.

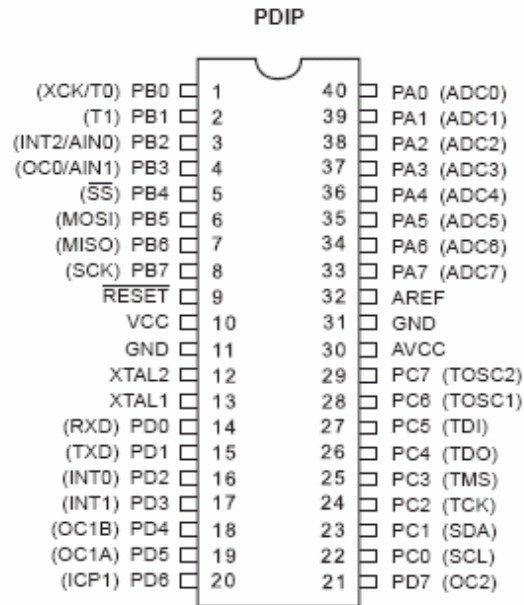


Gambar 2.45. Blok Diagram ATMEGA16

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.

<http://www.Atmel.com/dyn/recoources/pro_documents/DOC0839.pdf>

2.8.1. Fungsi *Pin* ATMEGA16



Gambar 2.46. Konfigurasi *Pin* ATMEGA16

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.

<http://www.Atmel.com/dyn/resources/pro_documents/DOC0839.pdf>

ATMEGA16 memiliki konfigurasi *pin* seperti tampak pada gambar di atas dengan fungsi masing-masing *pin* adalah sebagai berikut :

- a. VCC
 - Tegangan Sumber
- b. GND (*Ground*)
 - *Ground*
- c. *Port* ATMEGA16

ATMEGA16 memiliki 4 buah port I/O yaitu port A, port B, port C dan port D. Masing – masing port terdiri dari 8-bit *pin I/O* yang bersifat *bi-directional* dan setiap *pin* memiliki *internal pull-up resistor*. *Output buffer* masing - masing *port* dapat mengalirkan arus sebesar 20 mA. Ketika *port* digunakan sebagai *input* dan di *pull-down* secara langsung, maka *port* akan mengeluarkan arus (jika *internal pull-up resistor* diaktifkan). Pada masing – masing port memiliki fungsi khusus, antara lain :

- *Port A (PA7..PA0)*

Pin-pin dari *port A* memiliki fungsi khusus yaitu dapat berfungsi sebagai *channel ADC (Analog to Digital Converter)* sebesar 10 *bit*. Fungsi-fungsi khusus *pin-pin port A* dapat ditabelkan seperti yang tertera pada tabel 2.1.

Tabel 2.1. Fungsi khusus *port A* ATMEGA16

| Port Pin | Alternate Function |
|----------|----------------------------|
| PA7 | ADC7 (ADC input channel 7) |
| PA6 | ADC6 (ADC input channel 6) |
| PA5 | ADC5 (ADC input channel 5) |
| PA4 | ADC4 (ADC input channel 4) |
| PA3 | ADC3 (ADC input channel 3) |
| PA2 | ADC2 (ADC input channel 2) |
| PA1 | ADC1 (ADC input channel 1) |
| PA0 | ADC0 (ADC input channel 0) |

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.

<http://www.Atmel.com/dyn/recources/pro_documents/DOC0839.pdf>

- *Port B (PB7..PB0)*

Port B adalah 8-bit *port I/O* yang bersifat *bi-directional* dan setiap *pin* mengandung *internal pull-up resistor*. *Output buffer port B* dapat mengalirkan arus sebesar 20 mA. Ketika *port B* digunakan sebagai *input* dan di *pull-down* secara *eksternal*, *port B* akan mengalirkan arus (jika *internal pull-up resistor* diaktifkan). *Pin-pin port B* memiliki fungsi-fungsi khusus antara lain:

- *SCK-port B, bit 7 = Input pin clock* untuk *up/downloading Memory*.
- *MISO-port B, bit 6 = Pin output data* untuk *uploading Memory*.
- *MOSI-port B, bit 5 = Pin input data* untuk *downloading Memory*.

Fungsi-fungsi khusus *pin-pin port B* dapat ditabelkan seperti yang tertera pada tabel 2.2.

Tabel 2.2. Fungsi khusus *port* B ATMEGA16

| Port Pin | Alternate Functions |
|----------|---|
| PB7 | SCK (SPI Bus Serial Clock) |
| PB6 | MISO (SPI Bus Master Input/Slave Output) |
| PB5 | MOSI (SPI Bus Master Output/Slave Input) |
| PB4 | \overline{SS} (SPI Slave Select Input) |
| PB3 | AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output) |
| PB2 | AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input) |
| PB1 | T1 (Timer/Counter1 External Counter Input) |
| PB0 | T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output) |

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
<http://www.Atmel.com/dyn/resources/pro_documents/DOC0839.pdf>

- *Port* C (PC7..PC0)

Fungsi-fungsi khusus *pin-pin port* C dapat ditabelkan seperti yang tertera pada tabel 2.3.

Tabel 2.3. Fungsi khusus *port* C ATMEGA16

| Port Pin | Alternate Function |
|----------|--|
| PC7 | TOSC2 (Timer Oscillator Pin 2) |
| PC6 | TOSC1 (Timer Oscillator Pin 1) |
| PC5 | TDI (JTAG Test Data In) |
| PC4 | TDO (JTAG Test Data Out) |
| PC3 | TMS (JTAG Test Mode Select) |
| PC2 | TCK (JTAG Test Clock) |
| PC1 | SDA (Two-wire Serial Bus Data Input/Output Line) |
| PC0 | SCL (Two-wire Serial Bus Clock Line) |

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
<http://www.Atmel.com/dyn/resources/pro_documents/DOC0839.pdf>

- *Port D (PD7..PD0)*

Fungsi-fungsi khusus *pin-pin port D* dapat ditabelkan seperti yang tertera pada tabel 2.4.

Tabel 2.4. Fungsi khusus *port D* ATMEGA16

| Port Pin | Alternate Function |
|----------|---|
| PD7 | OC2 (Timer/Counter2 Output Compare Match Output) |
| PD6 | ICP1 (Timer/Counter1 Input Capture Pin) |
| PD5 | OC1A (Timer/Counter1 Output Compare A Match Output) |
| PD4 | OC1B (Timer/Counter1 Output Compare B Match Output) |
| PD3 | INT1 (External Interrupt 1 Input) |
| PD2 | INT0 (External Interrupt 0 Input) |
| PD1 | TXD (USART Output Pin) |
| PD0 | RXD (USART Input Pin) |

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
<http://www.Atmel.com/dyn/recources/pro_documents/DOC0839.pdf>

d. AVCC

Pin dari AVCC merupakan sumber tegangan untuk ADC. *Pin* AVCC harus tetap disambungkan ke VCC (meskipun tidak menggunakan konfigurasi dari ADC).

e. AREF

AREF merupakan *pin* tegangan *analog* referensi untuk ADC.

f. *Reset*

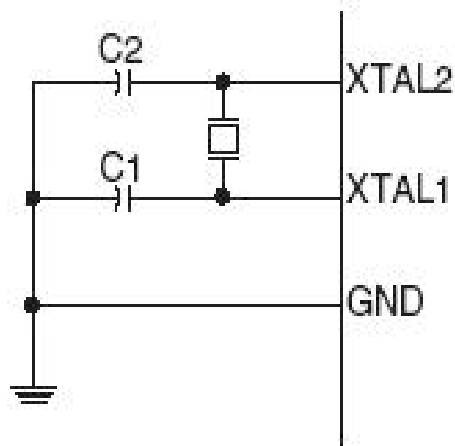
Input reset. Jika *pin* ini mendapat logika “*low*”, maka akan menjalankan *reset* (meskipun *clock* tidak aktif). Pulsa yang pendek tidak akan menjamin dapat menjalankan *reset* ini.

g. XTAL1

Merupakan *pin* untuk *inverting oscillator amplifier* dan *input* dari rangkaian *internal clock*.

h. XTAL2

Merupakan *output* dari *inverting oscillator amplifier*.

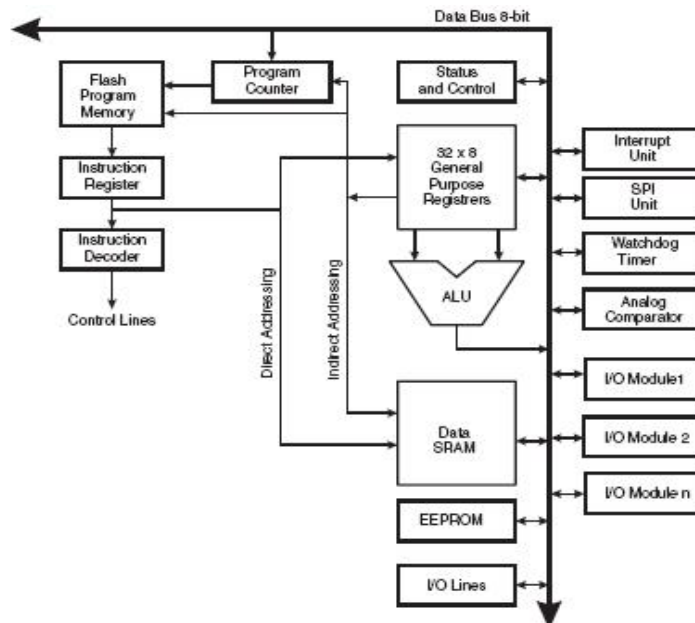


Gambar 2.47. Rangkaian *Crystal*.

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
 <http://www.Atmel.com/dyn/recources/pro_documents/DOC0839.pdf>

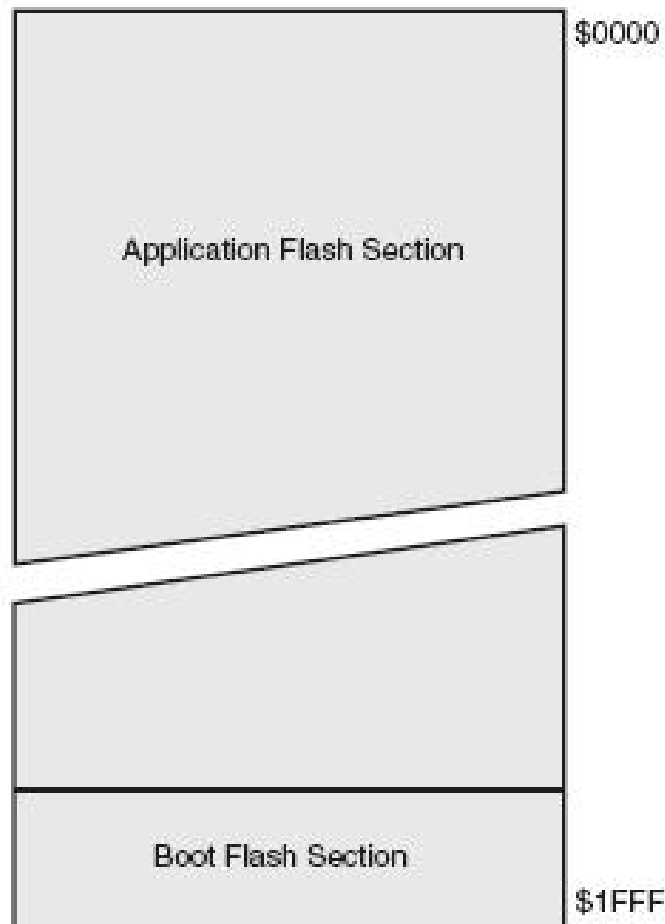
2.8.2. Arsitektur ATMEGA16

Struktur arsitektur dan peta *Memory* ATMEGA8 dapat digambarkan secara blok diagram seperti pada gambar di bawah ini.



Gambar 2.48. Arsitektur ATMEGA16 AVR RISC

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
 <http://www.Atmel.com/dyn/recources/pro_documents/DOC0839.pdf>



Gambar 2.49. Peta *Memory* ATMEGA16

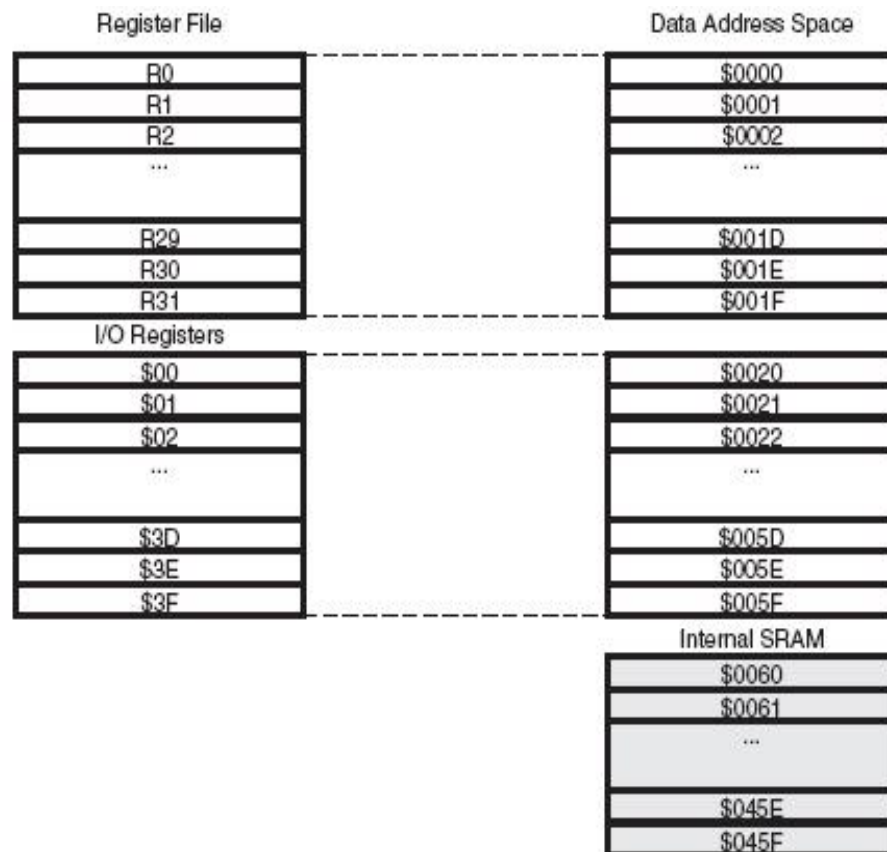
Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
 <http://www.Atmel.com/dyn/resources/pro_documents/DOC0839.pdf>

2.8.3. *In-Sistem Programmable Flash Program Memory*

ATMEGA16 memiliki 16K *byte On-chip In-Sistem Programmable Flash Memory* untuk menyimpan program. *Flash Memory* ini diatur pada ukuran 8K x 16bit karena semua instruksi AVR adalah 19-bit atau 32-bit *word*. *Flash Memory* dapat diisi atau dihapus sebanyak ± 10.000 kali.

2.8.4. SRAM Data Memory

Pada gambar 2.50 menunjukkan bagaimana *memory* data SRAM diatur.



Gambar 2.50. Pengaturan SRAM ATMEGA16

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
http://www.Atmel.com/dyn/recources/pro_documents/DOC0839.pdf

1120 lokasi data *Memory* digunakan sebagai alamat *register file*, *I/O Memory*, dan data SRAM. 96 lokasi *Memory* yang pertama digunakan untuk alamat *register file* dan *I/O Memory*, dan 1024 berikutnya meletakkan alamat data dari SRAM. Ada lima cara pengalamatan yang berbeda yaitu: *Direct*, *Indirect with Displacement*, *Indirect*, *Indirect with Pre-decrement*, dan *Indirect with Post-increment*.

Pengalamatan *Direct* mencapai seluruh alamat data yang kosong. 32 *register* yang bekerja, 64 *I/O register*, dan 1024 *bytes* data semuanya dapat diakses melalui semua cara pengalamatan ini.

2.8.5. EEPROM Data Memory

AT Mega 16 memiliki 512 *byte memory* data EEPROM . *Memory* ini diatur pada *data space* yang terpisah dimana setiap *byte*-nya dapat ditulis atau dibaca. EEPROM memiliki ketahanan sebesar 100.000 kali proses penulisan atau penghapusan.

2.8.6. Status Register – SREG

| | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | I | T | H | S | V | N | Z | C | SREG |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Gambar 2.51. Register SREG

- **Bit 7 – I: Global Interrupt Enable**
Global Interrupt Enable harus di-set (1) untuk memungkinkan menjalankan *interrupt*. Kontrol *interrupt* individu dijalankan dalam *register* kontrol yang berbeda. Jika *Global interrupt Enable* bernilai (0), tidak ada *interrupt* yang dilaksanakan, dan di-set oleh instruksi RETI untuk menjalankan *interrupt* berikutnya.
- **Bit 6 – T: Bit Copy Storage**
Instruksi *Bit Copy* BLD (*Bit Load*) dan BST (*Bit Store*) menggunakan T-*bit* sebagai sumber dan tujuan untuk *bit* yang dioperasikan. Sebuah *bit* dalam *register* dalam *register file* dapat disalin ke T oleh instruksi BST, dan *bit* di T dapat disalin ke *register* di dalam *register file* oleh instruksi BLD.
- **Bit 5 – H: Half Carry Flag**
Half Carry Flag H, mengindikasikan sebuah *half-carry* dalam beberapa operasi matematika.
- **Bit 4 – S: Sign Bit, $S = N + V$**
S-*bit* selalu terpisah atau diantara *Negative Flag* N dan *two's Complement Overflow Flag* V.

- *Bit 3 – V: Two's Complement Overflow Flag*
Two's Complement Overflow Flag V mendukung dua komplemen aritmatika
- *Bit 2 – N: Negative Flag*
Negative Flag mengindikasikan hasil negatif setelah operasi aritmatika dan logika dijalankan.
- *Bit 1 – Z: Zero Flag*
Zero Flag Z mengindikasikan hasil kosong (0) setelah operasi aritmatika dan logika dijalankan.
- *Bit 0 – C: Carry Flag*
Carry Flag C mengindikasikan sebuah *Carry* dalam operasi aritmatika atau logika.

2.8.7. *Stack Pointer – SP*

Pada ATMEGA16 terdapat 16-bit *stack pointer* yang terdiri dari dua 8-bit register. *Stack pointer* menunjuk ke *stack* dari SRAM dimana *subroutine* dan *interrupt* diletakkan. Ruang *Stack* dalam data SRAM ini harus didefinisikan oleh program sebelum *subroutine* dipanggil atau *interrupt* dijalankan. *Stack pointer* akan berkurang satu ketika data dimasukkan ke dalam *stack* (dengan menggunakan instruksi PUSH) dan akan berkurang dua jika alamat dimasukkan ke dalam *stack* (dengan pemanggilan *subroutine* atau *Interrupt*). *Stack Pointer* akan bertambah satu jika data dikeluarkan dari *stack* (menggunakan instruksi POP), dan akan bertambah dua ketika alamat dikeluarkan dari *stack* (dengan kembalinya dari *subroutine* RET atau dari *interrupt* RETI).

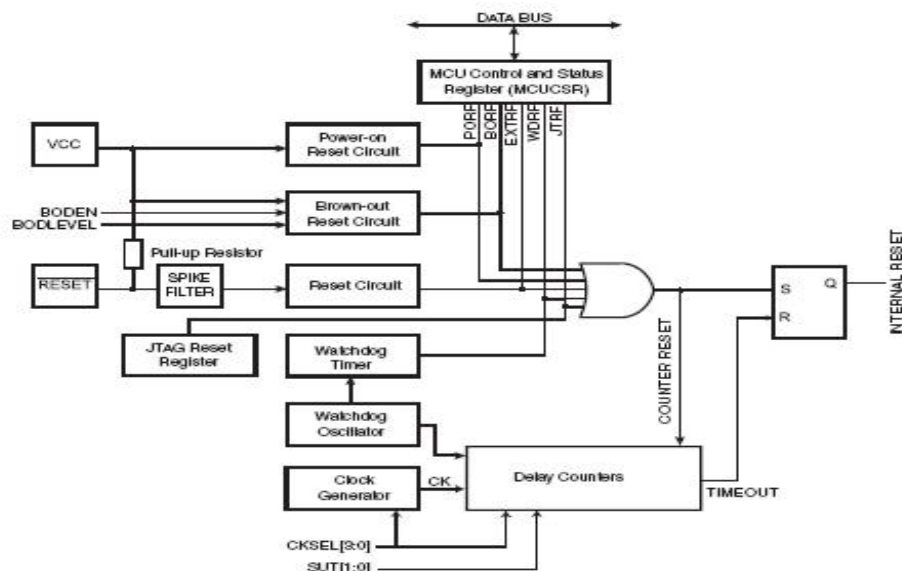
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---------------|------|------|------|------|------|------|-----|-----|-----|
| | SP15 | SP14 | SP13 | SP12 | SP11 | SP10 | SP9 | SP8 | SPH |
| | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | SPL |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Gambar 2.52. *Register Stack Pointer* ATMEGA16

2.8.8. Sumber *Reset*

Selama *reset*, semua *register* I/O dikembalikan pada nilai awalnya dan program mulai dieksekusi pada alamat 00_H. Instruksi yang diletakkan pada alamat 00_H harus berupa instruksi RJMP (*Relative Jump*) menuju rutin yang menangani *reset*. Apabila program tidak pernah menggunakan *interrupt*, maka *interrupt vector* tidak akan digunakan dan kode program biasa dapat diletakkan pada alamat vektor ini. Rangkaian dari logika *reset* dapat dilihat pada gambar 2.52. ATMEGA8 memiliki 4 sumber *reset*, yaitu:

- *Power-on Reset*: MCU dalam keadaan *reset* ketika tegangan *supply* berada di bawah tegangan *power-on reset*.
- *External Reset*: MCU dalam keadaan *reset* ketika sebuah *level* tegangan *low* terdapat pada *pin Reset* selama lebih dari 50nS.
- *Watchdog Reset*: MCU dalam keadaan *reset* ketika *watchdog* diaktifkan dan *watchdog timer* berakhir
- *Brown-out Reset*: MCU dalam keadaan *reset* ketika nilai dari sumber tegangan (VCC) berada di bawah *brown-out reset threshold* (V_{BOT}) dan *brown-out detector* dalam keadaan *enable*.

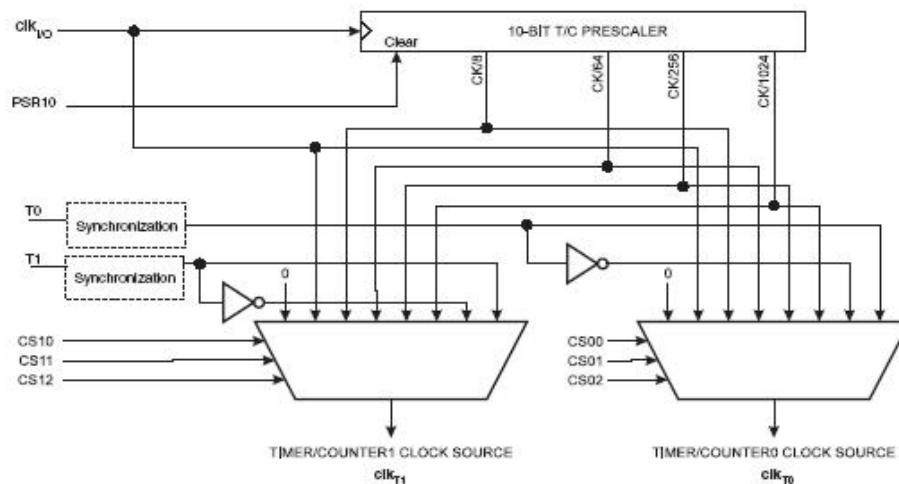


Gambar 2.53. Logika *Reset*

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
 <http://www.Atmel.com/dyn/resources/pro_documents/DOC0839.pdf>

2.8.9. Timer/Counter

ATMEGA16 menyediakan dua *general purpose timer/counter* (8 bit T/C dan 16 bit T/C). *Timer/counter* ini memiliki pilihan *prescale* sendiri-sendiri dari 10-bit *prescaling timer* yang sama.. Setiap *timer/counter* memiliki 4 pilihan *prescale* yang berbeda yaitu CK/8, CK/64, CK/256 dan CK/1024 dimana CK adalah *oscillator clock*. Gambar 2.53. menunjukkan *timer/counter prescaler*.



Gambar 2.54. *Timer/Counter Prescaler*

Sumber : Atmel Corporation. *ATmega16 Datasheet*. 30 Juli 2005.
 <http://www.Atmel.com/dyn/resources/pro_documents/DOC0839.pdf>