BAB II

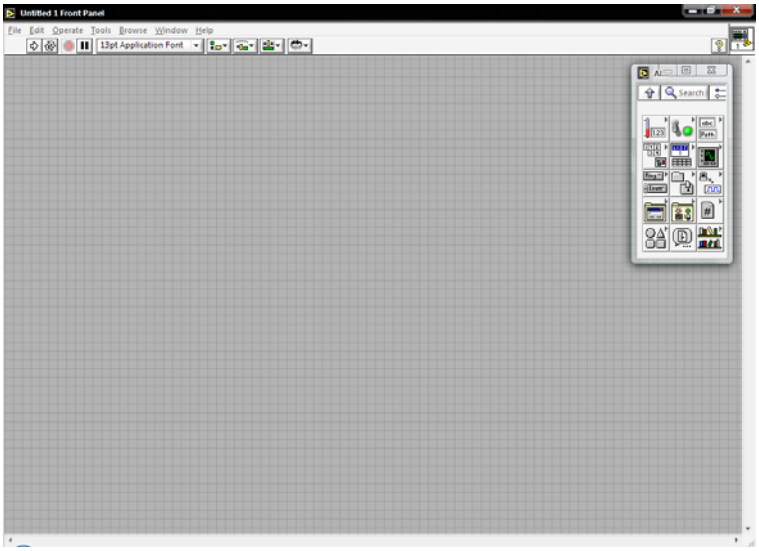
DASAR TEORI

2.1 Pengenalan LabVIEW

LabVIEW adalah sebuah *software* pemograman yang diproduksi oleh National *instruments* dengan konsep yang berbeda. Seperti bahasa pemograman lainnya yaitu C++, matlab atau *Visual basic* , LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa labVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis text. Program labVIEW dikenal dengan sebutan Vi atau *Virtual instruments* karena penampilan dan operasinya dapat meniru sebuah *instrument*. Pada labVIEW, *user* pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push buttons*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LEDs dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu *user* menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*. *Software* LabVIEW terdiri dari tiga komponen utama, yaitu :

1. *front panel*

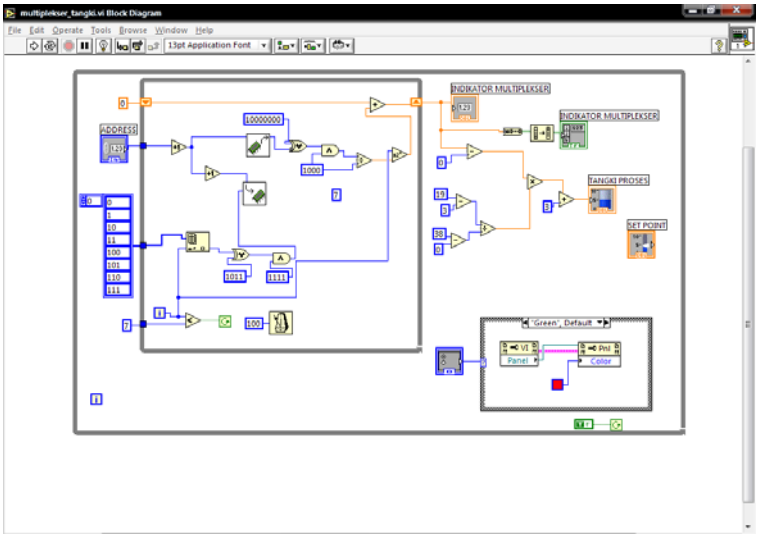
   front panel adalah bagian *window* yang berlatar belakang abu-abu serta mengandung *control* dan indikator. *front panel* digunakan untuk membangun sebuah VI, menjalankan program dan men*debug* program. Tampilan dari *front panel* dapat di lihat pada Gambar 2.1



Gambar 2.1 *Front Panel*

2. Blok diagram dari Vi

Blok diagram adalah bagian *window* yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai instruksi untuk *front panel*. Tampilan dari blok diagram dapat lihat pada Gambar 2.2.



Gambar 2.2 *Blok Diagram*

3. *Control* dan *Functions Pallete*

Control dan *Functions Pallete* digunakan untuk membangun sebuah Vi.

a. *Control Pallete*

Control Pallete merupakan tempat beberapa *control* dan indikator pada *front panel*, *control pallete* hanya tersedia di *front panel*, untuk menampilkan *control pallete* dapat dilakukan dengan mengkilk ***windows*** >> ***show control pallete*** atau klik kanan pada *front panel*. Contoh *control pallete* ditunjukkan pada Gambar 2.3.



Gambar 2.3 *Control Palette*

4. *Functions Pallete*

Functions Pallete di gunakan untuk membangun sebuah blok diagram, *functions pallete* hanya tersedia pada blok diagram, untuk menampilkannya dapat dilakukan dengan mengklik ***windows*** >> ***show control pallete*** atau klik kanan pada lembar kerja blok diagram. Contoh dari *functions pallete* ditunjukkan pada Gambar 2.4.



Gambar 2.4 *Functions pallete*

2.2 *Fuzzy Logic*

Fuzzy logic adalah suatu cara untuk memetakan suatu ruang input ke dalam suatu ruang output. *Fuzzy logic* menyediakan cara sederhana untuk menggambarkan kesimpulan pasti dari informasi yang ambigu, samar -samar, atau tidak tepat. Sedikit banyak, *fuzzy logic* menyerupai pembuatan keputusan pada manusia dengan kemampuannya untuk bekerja dari data yang ditafsirkan dan mencari solusi yang tepat. *Fuzzy logic* pada dasarnya merupakan logika bernilai banyak (*multivalued logic*) yang dapat mendefinisikan nilai diantara keadaan konvensional seperti ya atau tidak, benar atau salah, hitam atau putih, dan sebagainya. Penalaran *fuzzy* menyediakan cara untuk memahami kinerja dari sistem dengan cara menilai *input* dan *output system* dari hasil pengamatan.

Konsep *Fuzzy Logic* diperkenalkan oleh Prof. Lotfi Zadeh dari Universitas California di Berkeley pada 1965, dan dipresentasikan bukan sebagai suatu metodologi kontrol, tetapi sebagai suatu cara pemrosesan data dengan memperkenankan penggunaan *partial set membership* dibanding *crisp set membership* atau *non-membership*. Pendekatan pada set teori ini tidak diaplikasikan pada sistem kontrol sampai tahun 70an karena kemampuan

komputer yang tidak cukup pada saat itu. Profesor Zadeh berpikir bahwa orang tidak membutuhkan kepastian, masukan informasi numerik, dan belum mampu terhadap kontrol adaptif yang tinggi.

Konsep *fuzzy logic* kemudian berhasil diaplikasikan dalam bidang kontrol oleh E.H. Mamdani. Sejak saat itu aplikasi *fuzzy* berkembang kian pesat. Di tahun 1980an negara Jepang dan negara – negara di Eropa secara agresif membangun produk nyata sehubungan dengan konsep logika *fuzzy* yang diintegrasikan dalam produk–produk kebutuhan rumah tangga seperti *vacum cleaner*, mesin cuci, *microwave oven* dan kamera Video.

Ada beberapa alasan mengapa orang menggunakan logika *fuzzy* :

1. Konsep logika *fuzzy* mudah dimengerti, konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah di mengerti
2. Logika *fuzzy* sangat *fleksibel*.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi *non linear* yang sangat *kompleks*
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerja sama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

Sebelum menggunakan logika *fuzzy*, biasanya komputasi menggunakan himpunan *crisp*. Pada himpunan tegas (*crips*), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan $\mu_A [x]$, memiliki 2 kemungkinan, yaitu :

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

Tetapi penggunaan himpunan *crips* untuk menyatakan suatu keanggotaan terkadang sangat tidak adil, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Contoh jika variabel umur dibagi menjadi 2 kategori yaitu umur < 40 tahun disebut muda dan ≥ 40 disebut tua, maka bagaimana jika seseorang berusia 40 tahun kurang 1 hari? Himpunan *fuzzy* digunakan untuk mengantisipasi hal tersebut, seberapa besar eksistensi dalam suatu himpunan dapat dilihat pada nilai keanggotaannya.

2.2.1 Fungsi Keanggotaan

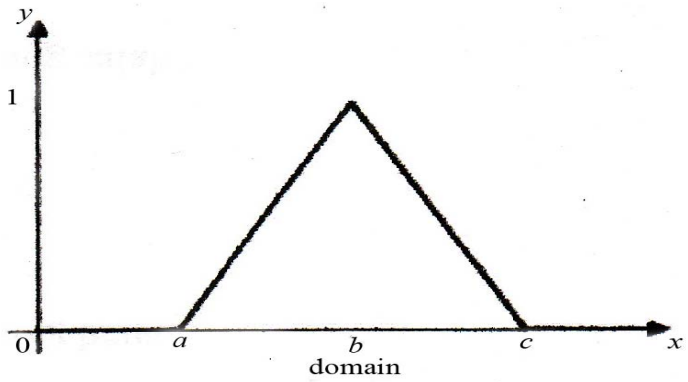
Fungsi keanggotaan (*membership functions*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga di sebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1, salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi ada beberapa fungsi yang digunakan dalam proyek tugas akhir ini sebagai perhitungan *input output* sistem, diantaranya :

1. Representasi Kurva *Triangel*

Representasi kurva *Triangel* merupakan gabungan antara dua buah garis linier naik dan garis linier turun, seperti persamaan (*2.1) fungsi* keanggotaan *Triangel* dispesifikasikan oleh tiga parameter (*a, b, c*).

$$\text{triangle}(x,a,b,c) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x \le c \\ 0 & c \le x \end{cases} \qquad \text{……...…….……..….(2.1)}$$

Representasi fungsi keanggotaan *Triangel* ditunjukkan pada Gambar 2.5



Gambar 2.5 *Representasi Fungsi Keanggotaan Triangel*

Alternatif lain dari persamaan (*2.1*) yaitu menggunakan *min* dan *max* seperti berikut.

μ[x] = max — — ……..……..……...(*2.2*)

parameter {*a*, *b*, *c*} (dengan $a < b < c$) ditentukan oleh koordinat *x* yang menentukan sudut dari fungsi keanggotaannya.

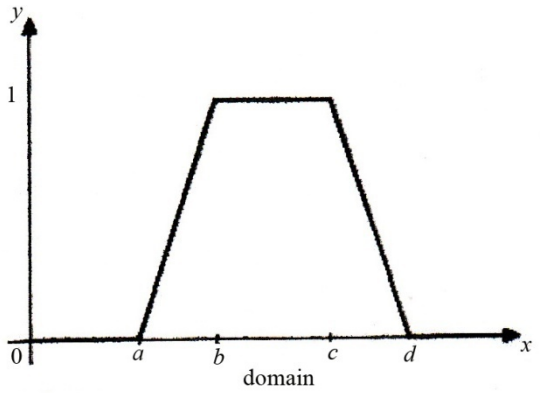
2. Representasi kurva trapezoid

Pada representasi kurva trapezoid kenaikan himpunan dimulai pada nilai domain dengan derajat keanggotaan nol [0] dan bergerak naik menuju

kekanan dengan nilai domain yang lebih tinggi, kemudian setelah beberapa saat mencapai derajat keanggotaan tertinggi dengan nilai doamin yang sama, kemudian nilai domain turun kembali menuju kekiri dengan nilai domain yang lebih rendah, seperti terlihat pada Gambar 2.6.

$$\text{trapezoid}(x;a,b,c,d) = \begin{cases} 0 & x \le a \\ \frac{x-a}{b-a} & a \le x \le b \\ 1 & b \le x \le c \\ \frac{c-x}{c-b} & c \le x \le d \\ 0 & d \le x \end{cases} \quad \text{.....(2.3)}$$

Fungsi keanggotaan trapezoid , dispesifikasikan oleh empat parameter, yaitu (*a*, *b*, *c*, *d*) seperti terlihat dalam persamaan (*2.3*), representasi fungsi keanggotaan trapezoid ditunjukkan pada Gambar 2.6



Gambar 2.6 *Representasi Fungsi Keanggotaan Trapezoid*

Alternatif lain dari persamaan (*2.3*) adalah dengan menggunakan *min* dan *max* seperti berikut :

$$\mu[x] = \max\left(min\left(\frac{x-a}{b-a}, 1, \frac{x-d}{c-d}\right), 0\right) \text{.....(2.4)}$$

2.2.2 Operator Dasar Untuk Operasi Keanggotaan Fuzzy

Untuk memodifikasi himpunan *fuzzy* , ada beberapa operasi yang didefinisikan, yaitu :

a. *fuzzy complement*

salah satu operasi yang digunakan pada *fuzzy complement* adalah operator NOT. Operasi ini mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

Persamaan matematis *fuzzy complement*

$\mu_A(x) = 1 - \mu_A(x)$……...…….……... (*2.5*)

b. *fuzzy interaction*

salah satu operasi umum yang digunakan pada *fuzzy interaction* adalah operator MIN atau operator AND, operasi ini diperoleh dengan mengambil nilai keanggotaaan terkecil diantara elemen-elemen dari himpunan yang bersangkutan.

Persamaan matematis *fuzzy interaction*

$\mu_{A\cap B}(x) = \min [\mu_A(x), \mu_B(x)]$..…….……...…….(*2. 6*)

c. *fuzzy union*

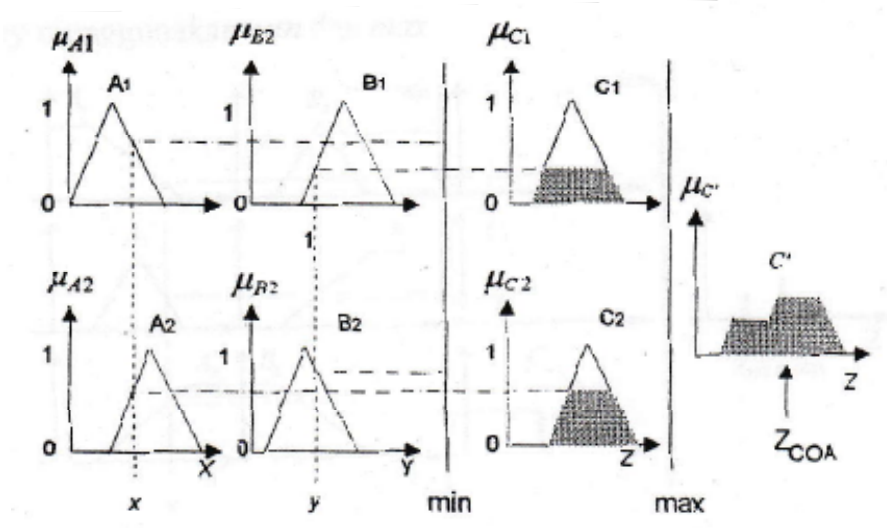
salah satu operasi umum yang digunakan pada *fuzzy interaction* adalah operator MAX atau operator OR. Operasi ini diperoleh dengan mengambil nilai keanggotaan terbesar diantara elemen-elemen dari himpunan yang bersangkutan.

Persamaan matematis *fuzzy union*

$\mu_{A\cup B}(x) = \max [\mu_A(x), \mu_B(x)]$……...……...…….(*2.7*)

2.2.3 Sistem Inferensi *Fuzzy* Metode Mamdani

Mekanisme proses pengontrolan *fuzzy* menggunakan sistem inferensi *fuzzy* metode mamdani adalah seperti terlihat pada Gambar 2.7.



Gambar 2.7 *Mekanisme Inferensi Fuzzy Model Mamdani*

Gambar 2.7 menunjukkan inferensi *fuzzy* model mamdani untuk dua rule. Proses dari inferensi tersebut dapat dijelaskan sebagai berikut :

1. terdapat dua masukan yaitu x dan y, masukan x akan dipetakan pada *membership function* A_1 dan A_2 sehingga diperoleh nilai μ_{A1} dan μ_{A2}, demikian juga masukan y dipetakan pada *membership function* B_1 dan B_2.
2. dilakukan operasi *interaction* (operasi min) pada μ_{A1} dan μ_{B1} sehingga diperoleh nilai $\mu_{C'1}$, demikian juga pada μ_{A1} dan μ_{B2} sehingga diperoleh nilai $\mu_{C'2}$.

3. dari nilai $\mu_{C'1}$. dan $\mu_{C'2}$. akan diperoleh besarnya daerah yang terarsir dari *membership function* C_1 dan C_2.

4. kedua daerah yang diarsir tersebut lalu digabungkan menjadi kurva C' yang kemudian akan dilakukan *defuzzifikasi* untuk memperoleh nilai Z_{COA}.

2.2.4 *Defuzzifikasi*

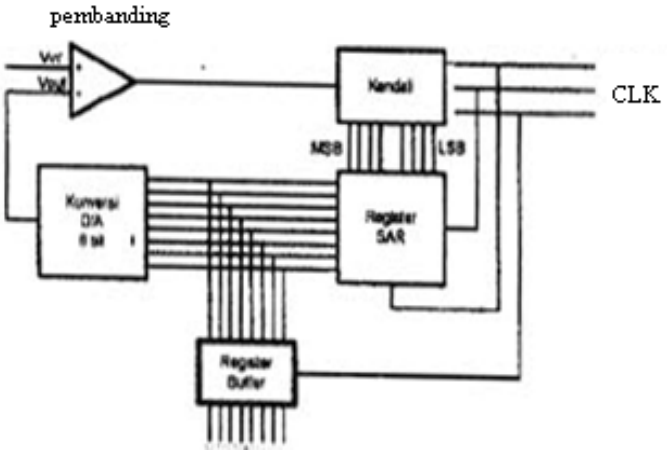
Proses *defuzzifikasi* berfungsi untuk merubah nilai *fuzzy* menjadi nilai *crisp*. Proses *defuzzifikasi* dilakukan dengan menghitung koordinat pusat massa dari bentuk fungsi keanggotaan C'. *Defuzzifikasi* pada komposisi aturan mamdani dengan menggunakan metode *centroid*. Dimana pada metode ini, solusi crisp diperoleh dengan cara mengambil titik pusat daerah *fuzzy*. Secara umum persamaan *defuzzifikasi* dituliskan pada persamaan 2.8 dibawah ini.

$$\mu(x) = \frac{\int_a^b x\mu(x)dx}{\int_a^b \mu(x)dx} \quad \text{atau} \quad \mu(x) = \frac{\sum_{i=1}^n x_i\mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \qquad (2.8)$$

2.3 *Analog To Digital Converter* (ADC)

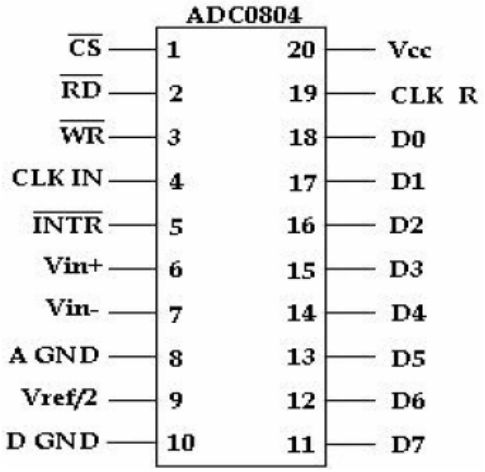
Analog To Digital Converter (ADC) adalah sebuah piranti yang dirancang untuk mengubah sinyal-sinyal analog menjadi bentuk sinyal digital. IC ADC 0804 dianggap dapat memenuhi kebutuhan dari rangkaian yang akan dibuat. IC jenis ini bekerja secara cermat dengan menambahkan sedikit komponen sesuai dengan spesifikasi yang harus diberikan dan dapat mengkonversikan secara cepat suatu masukan tegangan. Hal-hal yang juga perlu diperhatikan dalam penggunaan ADC ini adalah tegangan maksimum yang dapat dikonversikan oleh ADC dari rangkaian pengkondisi sinyal, *resolusi*, pewaktu *eksternal* ADC, tipe keluaran,

ketepatan dan waktu konversinya. Ada banyak cara yang dapat digunakan untuk mengubah *sinyal analog* menjadi *sinyal digital* yang nilainya proposional. Jenis ADC yang biasa digunakan dalam perancangan adalah jenis *Successive Approximation Convertion* (SAR) atau pendekatan bertingkat yang memiliki waktu konversi jauh lebih singkat dan tidak tergantung pada nilai masukan analognya atau sinyal yang akan diubah. Gambar 2.8, memperlihatkan diagram blok ADC tersebut.



Gambar 2.8 *Diagram Blok ADC*

Secara singkat prinsip kerja dari konverter *analog to digital* adalah semua bit-bit diset kemudian diuji, dan bilamana perlu sesuai dengan kondisi yang telah ditentukan. Dengan rangkaian yang paling cepat, konversi akan diselesaikan sesudah 8 clock, dan keluaran *digital to analog* merupakan nilai *analog* yang *ekivalen* dengan nilai *register SAR*. Apabila konversi telah dilaksanakan, rangkaian kembali mengirim sinyal selesai konversi yang berlogika rendah. Sisi turun sinyal ini akan menghasilkan data digital yang *ekivalen* ke dalam *register buffer*. Dengan demikian, *output* digital akan tetap tersimpan sekalipun akan dimulai siklus konversi yang baru.



Gambar 2.9 *ADC 0804*

Seperti terlihat pada Gambar 2.9, IC ADC 0804 mempunyai dua *input analog*, $Vi_{n(+)}$ dan $Vi_{n(-)}$, sehingga dapat menerima *input diferensial*. *Input analog* sebenarnya (Vin) sama dengan selisih antara tegangan-tegangan yang dihubungkan dengan ke dua pin input yaitu $V_{in} = V_{in(+)} - V_{in(-)}$. Kalau *input analog* berupa tegangan tunggal, tegangan ini harus dihubungkan dengan $V_{in(+)}$ sedangkan $V_{in(-)}$ ditanahkan. Untuk operasi normal, ADC 0804 menggunakan V_{cc} = +5 Volt sebagai *referens*. Dalam hal ini jangkauan *input analog* mulai dari 0 Volt sampai 5 Volt (skala penuh), karena IC ini adalah SAC 8-bit, maka resolusinya adalah sebagai berikut :

$$\text{Resolusi} = \left(\frac{tegangan\ skala\ penuh}{2^n - 1}\right) = \frac{5\ Volt}{255} = 19.6\ \text{mVolt} \quad \text{.....(2.9)}$$

2.3.1 Mode Operasi ADC0804

1. Mode operasi kontinyu

   Agar ADC0804 dapat dioperasikan pada mode operasi kontinyu (proses membaca terus menerus dan tanpa proses operasi jabat tangan), maka penyemat CS dan RD ditanahkan, sedangkan penyemat WR dan INTR tidak dihubungkan kemanapun. Prinsip kerja operasi kontinyu ini yaitu ADC akan memulai konversi ketika INTR kembali tidak aktif (logika '1'). Setelah proses konversi selesai, INTR akan aktif (logika '0'). Untuk memulai konversi pertama kali WR harus ditanahkan terlebih dahulu, hal ini digunakan untuk me-*reset* SAR. Namun pada konversi berikutnya untuk me-*reset* SAR dapat menggunakan sinyal INTR saat aktif (logika '0') dan mulai konversi saat tidak aktif (logika '1'). Ketika selesai konversi data hasil konversi akan dikeluarkan secara langsung dari *buffer* untuk dibaca karena RD ditanahkan. Saat sinyal INTR aktif, sinyal ini digunakan untuk me-*reset* SAR. Saat INTR kembali tidak aktif (logika '1') proses konversi dimulai kembali.

2. Mode Operasi Hand-Shaking

   ADC0804 dioperasikan pada *mode hand-shaking* . Agar ADC dapat bekerja, CS harus berlogika '0'. Ketika WR berlogika '0', *register* SAR akan direset, sedangkan ketika sinyal WR kembali '1', maka proses konversi segera dimulai. Selama konversi sedang berlangsung, sinyal INTR akan tidak aktif (berlogika '1') sedangkan saat konversi selesai ditandai dengan aktifnya sinyal INTR (logika'0'). Setelah proses konversi selesai data hasil konversi tetap tertahan pada buffer ADC. Data hasil konversi tersebut akan

dikeluarkan dengan mengirim sinyal RD berlogika ‘0’. Setelah adanya sinyal sinyal RD ini, maka sinyal INTR kembali tidak aktif.

ADC digunakan sebagai rangkaian yang mengubah sinyal analog menjadi sinyal digital. Dengan menggunakan ADC, kita dapat mengamati perubahan sinyal analog seperti : perubahan temperatur , kepekaan asap, temperatur udara, kecepatan angin, berat benda, kadar asam (pH) dan lain-lain, yang semuanya dapat di amati melalui sensornya masing-masing. Rangkaian analog ke digital ini dimaksudkan untuk mengubah data tegangan yang dihasilkan oleh rangkaian sensor menjadi data digital agar dapat dieksekusi oleh rangkaian mikrokontroler.

2.4 Motor Stepper

Motor stepper adalah alat yang mengubah pulsa listrik yang diberikan menjadi gerakan motor *discret* (berlainan) yang disebut *step* (langkah). Satu putaran motor memerlukan 360 derajat dengan jumlah langkah yang tertentu perderjatnya. Ukuran kerja dari motor stepper biasanya diberikan dalam jumlah langkah per-putaran per-detik. Motor stepper mempunyai kecepatan dan torsi yang rendah namun memiliki kontrol gerakan posisi yang cermat, hal ini dikarenakan motor stepper memiliki beberpa segment kutub kumparan. Motor ini sering kita gunakan untuk menggerakan lengan robot, gerak linier plotter. Gambar dari motor stepper terlihat pada Gambar 2.10.



Gambar 2.10 *Motor Stepper*

Pada dasarnya ada dua jenis motor stepper yaitu bipolar dan unipolar, sebuah motor stepper berputar 1 step apabila terjadi perubahan arus pada koil-koilnya, mengubah *pole-pole* magnetik disekitar *pole-pole stator*.

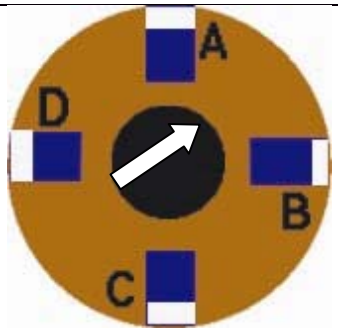
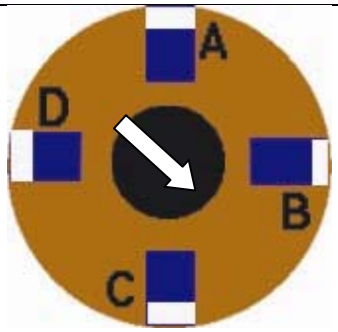
Perbedaan utama antara bipolar dan unipolar adalah :

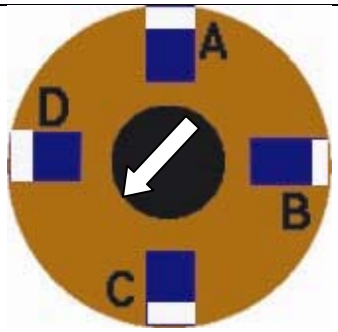
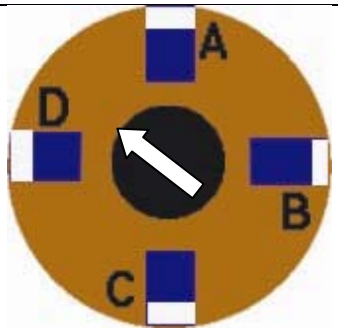
1. Bipolar :
   - Arus pada *koil* dapat berbolak-balik untuk mengubah arah putaran motor.
   - Lilitan motor hanya satu dan dialiri arus dengan arah bolak-balik.
2. Unipolar :
   - Arus mengalir satu arah, dan perubahan arah motor tergantung dari lilitan (*koil*) yang dialiri arus.
   - Lilitan terpisah dalam dua bagian dan masing-masing bagiannya hanya dilewati arus dalam satu arah saja.

Kelemahan jenis bipolar adalah bahwa rangkaian drivernya lebih kompleks, karena harus dapat menglirkan arus dalam dua arah melalui koil yang sama. Sedangkan jenis unipolar, selain motor stepper tersebut lebih mudah diperoleh di pasaran juga memerlukan rangkaian driver yang lebih sederhana.

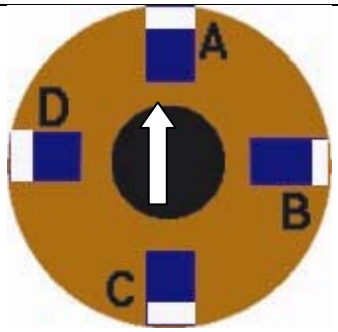
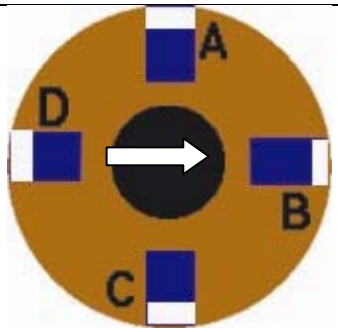
Proses pengendalian motor stepper unipolar dilakukan dengan menghubungkan kutub-kutub motor ke *ground* secara begantian. Kutub motor yang berhubungan dengan ground akan mengaktifkan koil yang bersangkutan. Maka dengan mengaktifkan urutan yang tepat, motor stepper dapat bergerak secara *full stepping* maupun *half stepping* baik searah maupun berlawanan dengan jarum jam. Jika motor stepper bergerak 1.8 derajat atau *step* pada *mode full stepping* , maka pada *mode half stepping* motor dapat digerakkan sebesar 0.9 derajat atau *step*. Pengaturan kutub-kutub motor dan proses gerak motor stepper dapat dipahami melalui contoh Tabel 2.1, Tabel 2.2 dan Tabel 2.3.

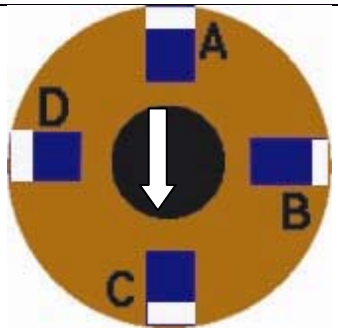
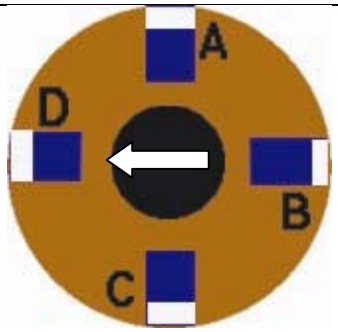
Tabel 2.1 *Mode Full Step Motor Stepper (a)*

Fasa	Kutub				Proses
	A	B	C	D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	

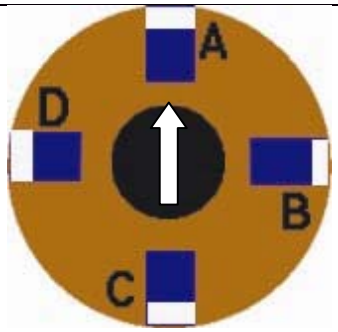
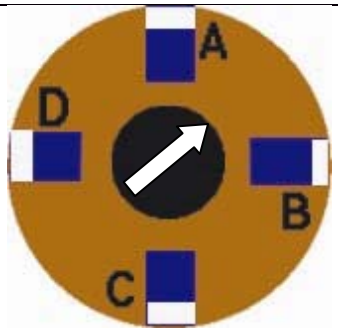
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

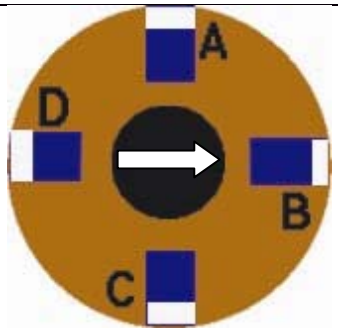
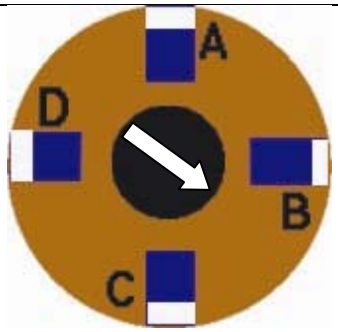
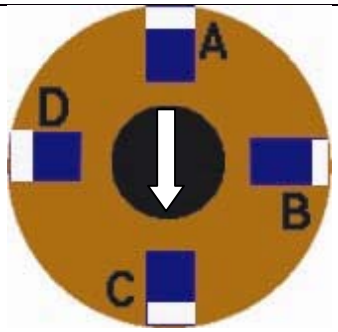
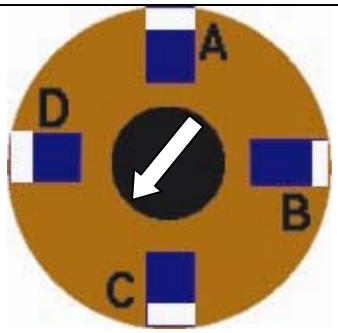
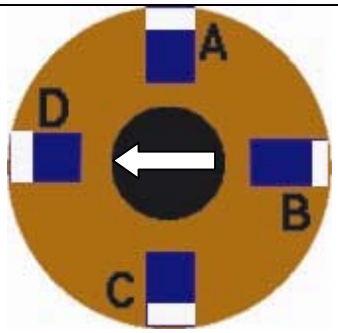
Tabel 2.2 *Mode Full Step Motor Stepper (b)*

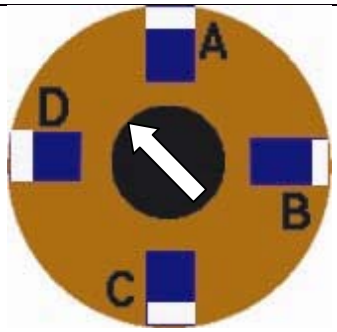
Fasa	Kutub				Proses
	A	B	C	D	
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	

3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

Tabel 2.3 *Mode Half Step Motor Stepper*

Fasa	Kutub				Proses
	A	B	C	D	
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	

3	OFF	ON	OFF	OFF	
4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	
6	OFF	OFF	ON	ON	
7	OFF	OFF	OFF	ON	

8	ON	OFF	OFF	ON	

2.5 Potensiometer

Resistor berasal dari kata *resist* yang berarti menahan (dalam elektronika artinya menahan tegangan dan arus). Resistor merupakan komponen pasif dalam elektronika yang berarti berfungsi sebagai hambatan atau penahan arus listrik, resistor terbagi dari beberapa jenis diantaranya:

1. resistor yang nilainya konstan (*fixed resistor*)
2. resistor variable (*variable resistor*)
3. resistor non linear (*non linear resistor*)



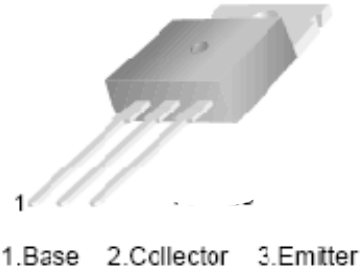
Gambar 2.11 *Potensiometer*

Variable resistor adalah resistor yang nilai tahanannya dapat diubah-ubah sesuai dengan nilai yang di kehendaki contoh resistor variable adalah *potensiometer*

karbon, present, *present potensio meter* yang lebih dikenal dengan nama *trimmer potensio (trimpot),* hal ini dapat diamati pada contoh Gambar 2.11.

2.6 Transistor TIP 122

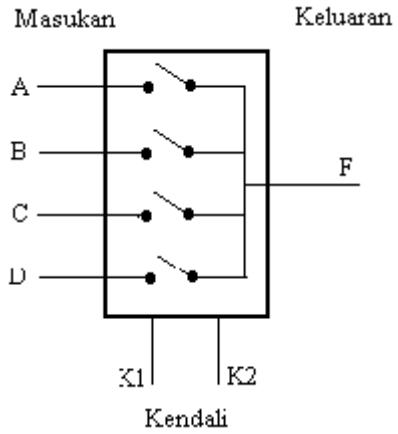
Transistor TIP 122 merupakan jenis transistor NPN, pada proyek tugas akhir ini transistor TIP 122 digunakan sebagai driver motor stepper, agar TIP 122 dapat digunakan sebagai driver motor stepper, maka TIP 122 harus rangkai secara darlington. Rangkaian darlington adalah rangkaian elektronika yang terdiri dari sepasang transistor bipolar (dua kutub) yang tersambung secara seri. Sambungan seri ini digunakan sebagai penguatan yang tinggi, karena hasil penguatan pada transistor pertama akan dikuatkan lebih lanjut oleh transistor kedua. Datasheet dari transistor NPN TIP 122 terlihat seperti Gambar 2.12 di bawah ini.



Gambar 2.12 *Transitor TIP 122*

2.7 Multiplekser

Multiplekser adalah suatu rangkaian elektronik yang mampu menyalurkan sinyal salah satu dari banyak masukan ke sebuah keluaran. Pemilihan masukan ini dilakukan melalui masukan penyeleksi. Secara bagan kerja Multiplekser ditunjukkan pada Gambar 2.13.



Gambar 2.13 *Bagan Multiplekser*

Kendali pada Multiplekser akan memilih saklar mana yang akan dihubungkan. Saluran kendali sebanyak "n" saluran dapat menyeleksi 2n saluran masukan. Sebagai contoh: sebuah Multiplekser 4 ke 1 dengan Kendali K1 dan K2. Ketika saluran Enable = 1, keluaran selalu bernilai nol. Tetapi ketika saluran Enable = 0, keluaran F diatur melalui K1 dan K2. Tabel kebenaran Multiplekser ini dinyatakan seperti terlihat pada Tabel 2.4.

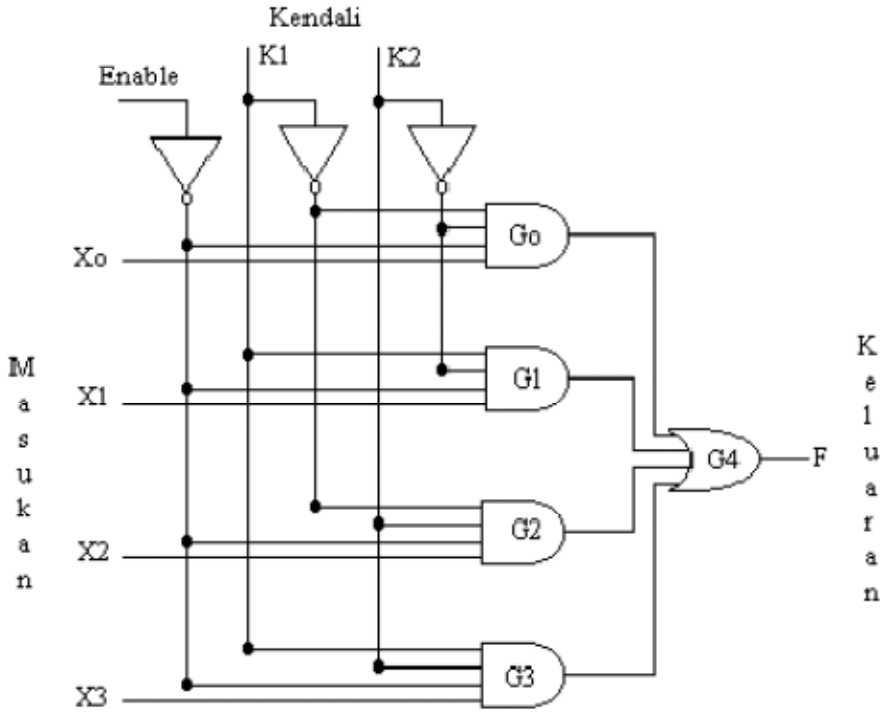
Tabel 2.4 *Tabel Kebenaran Multiplekser*

$\overline{\textbf{Enable}}$	K2	K1	F
1	x	X	0
0	0	0	Xo
0	0	1	X1
0	1	0	X2
0	1	1	X3

Jika E mewakili saluran Enable, maka berdasarkan Tabel kebenaran tersebut keluaran F dapat dinyatakan sebagai :

F = E.Xo.K1.K2 + E.X1.K1.K2 + E.X2.K1.K2 + E.X3.K1.K2……..(*2.10*)

Berdasarkan persamaan 2.10 disusunlah rangkaian logika Multiplekser 4 ke1 seperti ditunjukkan oleh Gambar 2.14.



Gambar 2.14 *Rangkaian Logika Multiplekser 4 ke 1*

2.8 Komunikasi Parallel

Komunikasi paralel adalah komunikasi yang mengirimkan data secara bersamaan. Pada penggunaan komunikasi paralel semua bit dikirim secara bersamaan pada waktu yang sama. Oleh karena itu pada komunikasi ini kita membutuhkan banyak kabel. Hal memang sering menjadi kelemahan komunikasi paralel akibat banyaknya kabel yang dibutuhkan, dan panjang kabel ini tidak boleh lebih dari 20m, untuk menjaga keaslian data. Namun kelebihan komunikasi paralel adalah lebih cepat dan kapasitas yang dibawa juga banyak serta pemrograman yang lebih mudah. Komunikasi paralel yang digunakan adalah komunikasi paralel lewat

kabel data untuk printer (saat mengeluarkan data). Pada keadaan normal (tidak aktif) tegangan pada pin-pin ini adalah 0 volt, namun bila kita beri high, maka tegangannya akan berubah menjadi 5 volt. Pada perancangan alat, komunikasi paralel hanya digunakan untuk mengeluarkan data, yang bisa berguna untuk menyalakan *relay* atau motor stepper untuk menjalankan atau mengontrol *hardware*. Pada port paralel ada tiga jalur data yaitu :

1. jalur kontrol memiliki arah bidirectional.
2. jalur kontrol memiliki satu arah yaitu : arah keluaran.
3. jalur data, memiliki 2 arah. Dapat juga berfungsi sebagai pengirim Address dan data, masing-masing 8 bit, dimana keduanya melakukan transfer data dengan protokol handshaking serta diakses dengan register yang berbeda. Bila kita menggunakan jalur LPT1 maka base address biasanya dalam bentuk hexadesimal , seperti 278, 378 atau 3BC, seperti terlihat pada Tabel 2.5.

Tabel 2.5 *Base Address*

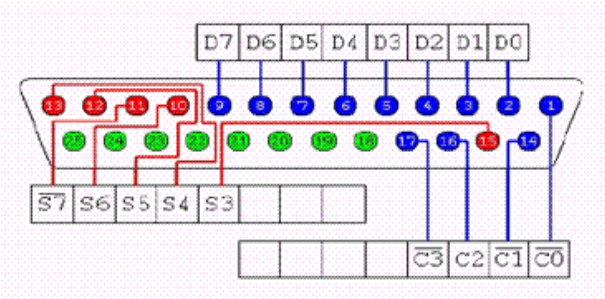
Ragister kontrol	Register status	Register data	Register address
37A	379	378	37B

Data output yang dituliskan pada register dapat diukur tegangannya dengan mengunakan multimeter, dan saat diberikan tregangan ke pin input pada paralel port , dapat dibaca oleh register.sebagai contoh jika salah satu pin out dialamatkan nilai “1” maka akan terbaca tegangan 5 Volt pada pin yang bersangkutan. Fungsi dari pin-pin DB 25 terlihat pada Tabel 2.6.

Tabel 2.6 *Fungsi Pin – Pin DB 25*

Pin No (DB25)	Signal name	Direction	Register-bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data 0	In/Out	Data-0	No
3	Data 1	In/Out	Data-1	No
4	Data 2	In/Out	Data-2	No
5	Data 3	In/Out	Data-3	No
6	Data 4	In/Out	Data-4	No
7	Data 5	In/Out	Data-5	No
8	Data 6	In/Out	Data-6	No
9	Data 7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	

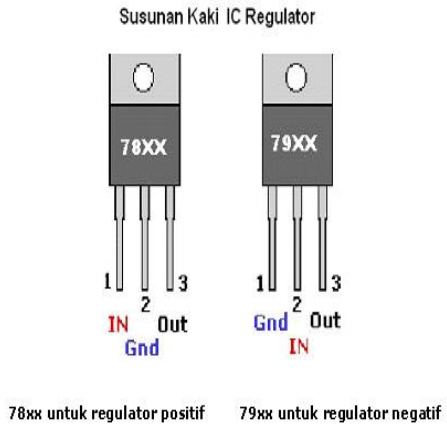
Port paralel dapat mentransmisi 8 bit data dalam sekali detak. Tata-letak dari ke-dua puluh lima pin (DB 25) parallel printer port, diperlihatkan dalam Gambar 2.15.



Gambar 2.15 *Konfigurasi Pin Parallel Port*

2.9 IC Regulator

IC regulator adalah IC yang digunakan untuk menghasilkan tegangan yang stabil, dari sumber tegangan yang tidak stabil. Besarnya tegangan keluaran tergantung dari jenis voltase regulator. Ada beberapa macam voltase regulator, yaitu 5V, 6V, 7V, 8V, 9V, 10V, 12V, 15V, dan 18V. Dan voltase yang diperbolehkan untuk input adalah maksimal 35V (tergantung pabrik). Sedangkan arus pada output untuk tipe L seperti 78L05 dan 7812 adalah 100 mA, sedangkan untuk tipe biasa seperti 7805 arus output antara 500 mA sampai dengan 1 A, Gambar 2.16 menunjukkan susunan kaki dari IC 7805.



Gambar 2.16 *Susunan Kaki IC 7805*