BAB II

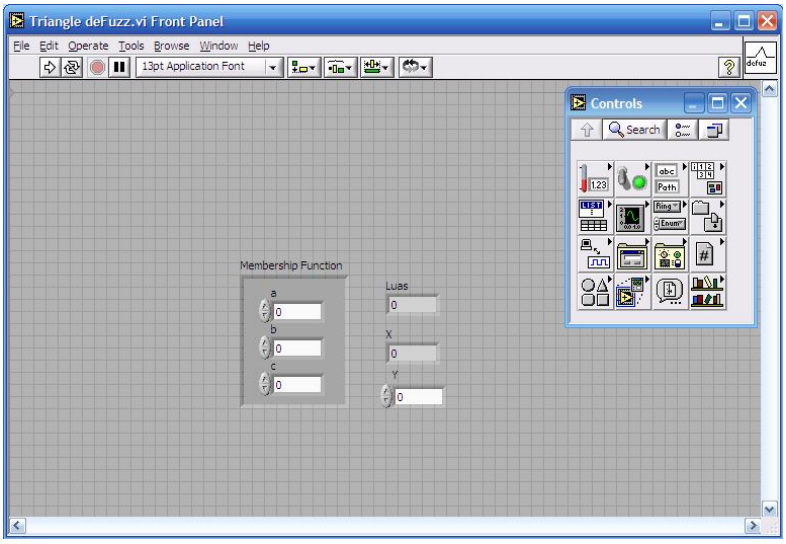
DASAR TEORI

2.1 Pengenalan LabVIEW

LabVIEW adalah sebuah *software* pemograman yang diproduksi oleh National *Instruments*. Seperti bahasa pemograman lainnya yaitu C++, matlab atau *visual basic* , LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya adalah LabVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sedangkan bahasa pemrograman lainnya menggunakan 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 kontrol 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:

2.1.1 *Front Panel*

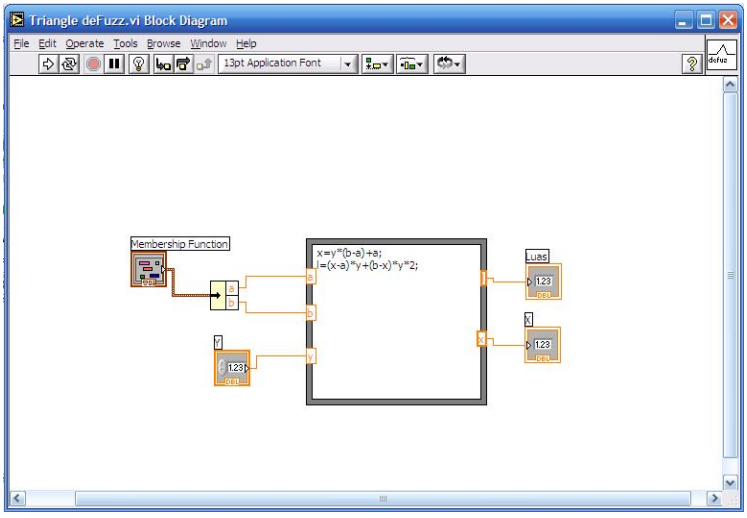
Front Panel adalah bagian *window* yang mempunyai *background* abu-abu serta mengandung kontrol dan indikator. *front panel* digunakan untuk membangun sebuah VI, menjalankan program dan men*debug* program. Tampilan dari *front panel* dapat di lihat pada gambar 2.1.



Gambar 2.1 *Front Panel*

2.1.2 Blok Diagram

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*

2.1.3 *Control* dan *Function Pallete*

Control dan *function pallete* digunakan untuk membangun sebuah VI.

a. *Control pallete*

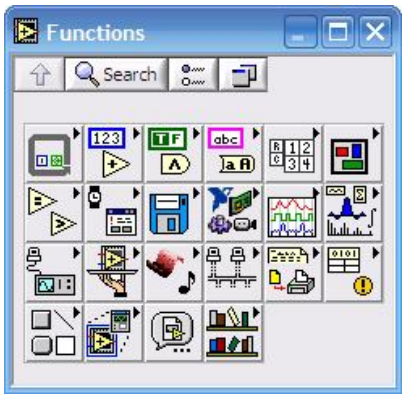
Control pallete merupakan tempat beberapa *control* dan indikator pada *front panel*, *control pallete* hanya tersedia di *front panel*, untuk menampilkan *control pallete* dapat dilakukan dengan mengklik **windows >> show control pallete** atau klik kanan pada *front panel*. Contoh control pallete ditunjukkan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



Gambar 2.3 *Control Palette*

b. *Function pallete*

Fungction pallete di gunakan untuk membangun sebuah blok diagram, *fungction 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 *fungction pallete* ditunjukkan pada gambar 2.4.



Gambar 2.4 *Function Pallete*

2.2 *Adaptive Neuro-Fuzzy Interference System (ANFIS)*

Adaptive Neuro-Fuzzy Interference System (ANFIS) merupakan jaringan adaptif yang berbasis pada sistem kesimpulan *fuzzy* (*fuzzy interference system*). Dengan penggunaan suatu prosedur *hybrid learning*, *ANFIS* dapat membangun suatu mapping *input-output* yang keduanya berdasarkan pada pengetahuan manusia (pada bentuk aturan *fuzzy if-then*) dengan fungsi keanggotaan yang tepat.

Sistem kesimpulan *fuzzy* yang memanfaatkan aturan *fuzzy if-then* dapat memodelkan aspek pengetahuan manusia yang kwalitatif dan memberi *reasoning processes* tanpa memanfaatkan analisa kwantitatif yang tepat. Ada beberapa aspek dasar dalam pendekatan ini yang membutuhkan pemahaman lebih baik, secara rinci:

a. Tidak ada metoda baku untuk men-*transform* pengetahuan atau pengalaman manusia ke dalam aturan dasar (*rule base*) dan database tentang *fuzzy interference system*.

b. Ada suatu kebutuhan bagi metoda efektif untuk mengatur (*tuning*) fungsi keanggotaan (*membership function*/MF) untuk memperkecil ukuran kesalahan keluaran atau memaksimalkan indeks pencapaian.

ANFIS dapat bertindak sebagai suatu dasar untuk membangun satu kumpulan aturan *fuzzy if-then* dengan fungsi keanggotaan yang tepat, yang berfungsi untuk menghasilkan pasangan *input-output* yang tepat.

2.2.1 Aturan *Fuzzy If-Then*

Pernyataan aturan *fuzzy if-then* atau *fuzzy conditional* adalah ungkapan dengan format *IF A THEN B*, di mana A dan B adalah label dari himpunan *fuzzy* yang ditandai oleh fungsi keanggotaan yang sesuai. Dalam kaitan dengan formatnya, aturan *fuzzy if-then* sering digunakan untuk menangkap "*mode*" yang tidak tepat dalam memberi alasan yang digunakan dalam kemampuan manusia untuk membuat keputusan dalam suatu lingkungan yang tidak pasti dan tidak tepat.

Suatu contoh yang menguraikan fakta sederhana adalah

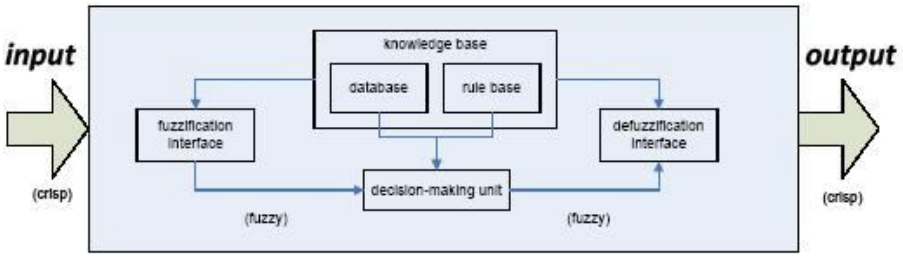
If pressure is high, then volume is small

dimana *pressure* dan *volume* adalah variabel bahasa *(linguistic variables)*, *high* dan *small* adalah *linguistic values* atau label linguistik yang merupakan karakterisrik dari fungsi keanggotaan.

Melalui penggunaan fungsi keanggotaan dan label bahasa, suatu aturan *fuzzy if-then* dapat dengan mudah menangkap “peraturan utama” yang digunakan oleh manusia.

2.2.2 Sistem Kesimpulan *Fuzzy*

Pada dasarnya suatu sistem kesimpulan *fuzzy* terdiri atas 5 (lima) blok *fungsional* (blok diagram *fuzzy interference system*) digambarkan pada gambar di bawah :



Gambar 2.5 *Sistem Fuzzy*

Sebuah aturan dasar (*rule base*) yang berisi sejumlah aturan *fuzzy if-then*

- Suatu *database* yang menggambarkan fungsi keanggotaan dari himpunan *fuzzy* yang digunakan pada aturan *fuzzy*.
- Suatu unit pengambilan keputusan (*decision making-unit*) melakukan operasi *interference* (kesimpulan) terhadap aturan.
- Suatu *fuzzification interface* mengubah bentuk masukan derajat crisp atau tingkat kecocokan dengan nilai bahasa.
- Suatu *defuzzification interface* yang mengubah bentuk hasil *fuzzy* dari kesimpulan ke dalam suatu *output crisp*.

Sistem inferensi *fuzzy* yang digunakan adalah sistem inferensi *fuzzy* model Tagaki-Sugeno-Kang (TSK) orde satu dengan pertimbangan kesederhanaan dan kemudahan komputasi.

Rule 1 : **if x is A1 and y is B1 then z1 = ax + by + c**

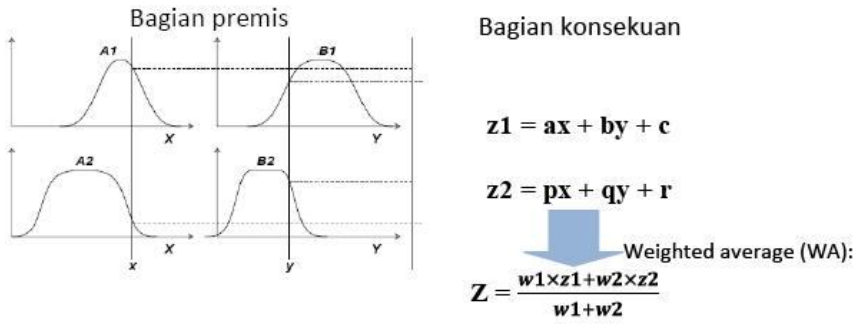
premis konsekuen

Rule 2 : **if x is A2 and y is B2 then z2 = px + qy + r**

premis konsekuen

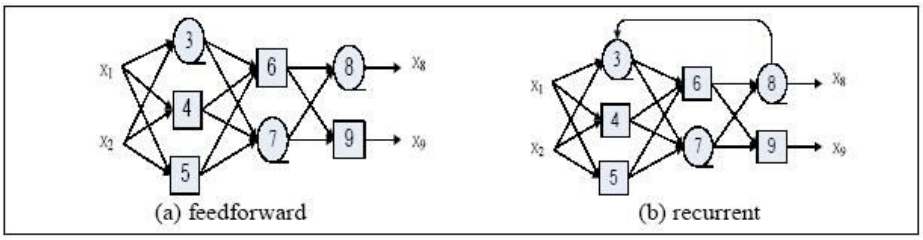
Input adalah x dan y. Konsekuen adalah z.

Langkah dari *fuzzy reasoning* dilakukan oleh sistem kesimpulan *fuzzy* adalah:



Gambar 2.6 *Fuzzy Reasoning*

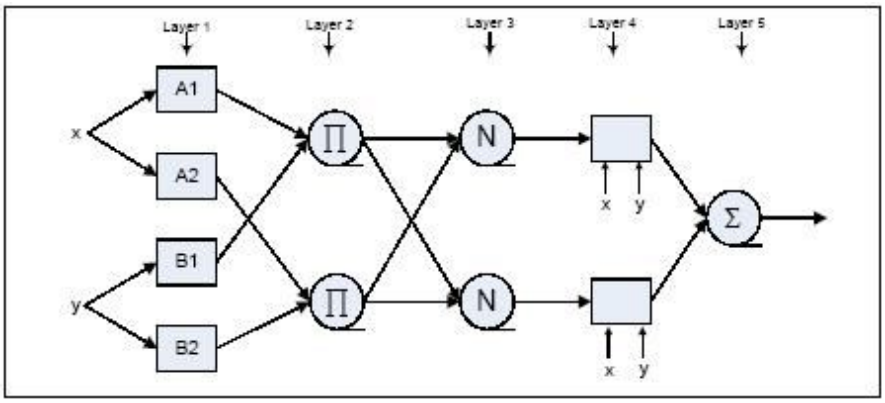
2.2.3 Jaringan Adaptif: Arsitektur dan Algoritma Pembelajaran



Gambar 2.7 *Jaringan Adaptif*

Sesuai dengan namanya, jaringan adaptif merupakan suatu struktur jaringan yang terdiri dari sejumlah simpul yang terhubung melalui penghubung secara langsung (*directional links*). Tiap simpul menggambarkan sebuah unit proses, dan penghubung antara simpul menentukan hubungan kausal antara simpul yang saling berhubungan. Semua atau sebagian simpul bersifat adaptif (bersimbol kotak) dapat dilihat pada gambar 2.7, yang artinya keluaran dari simpul tersebut bergantung pada parameter peubah yang mempengaruhi simpul itu.

2.2.4 Arsitektur *ANFIS*



Gambar 2.8 *Arsitektur ANFIS*

Simpul-simpul (neuron-neuron) pada arsitektur *ANFIS* (gambar 2.8):

- Nilai-nilai (Membership function).
- Aturan- aturan (T-norm yang berbeda, umumnya yang digunakan operasi *product)*.
- Normalisasi (Penjumlahan dan pembagian aritmatika).

- Fungsi-fungsi (*Regresi linier* dan pengalian dengan).
- *Output* (Penjumlahan aljabar).

Simpul-simpul tersebut dibentuk dalam arsitektur *ANFIS* yang terdiri dari 5 (lima) lapis. Fungsi dari setiap lapis tersebut sebagai berikut:

a. Lapis 1

Setiap simpul i pada lapis 1 adalah simpul adaptif dengan nilai fungsi simpul sebagai berikut:

$O_{1,i} = \mu_{Ai}(X)$ untuk i=1,2 …… (*2.1*)

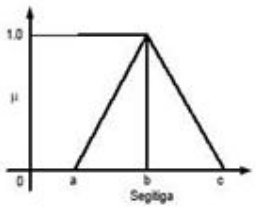
$O_{1,i} = \mu_{Ai}(X)$ untuk j=1,2 …… (*2.2*)

dengan : x dan y adalah masukan simpul ke i.

$A_i(x)$ dan $B_j(y)$ adalah label linguistik yang terkait dengan simpul tersebut. O1,i dan O1,j adalah derajat keanggotaan himpunan *fuzzy* A1, A2, atau B1, B2.

Fungsi keanggotaan untuk A atau B dapat diparameterkan, misalnya fungsi segitiga:

$$\mu_A[x] = \begin{cases} 0; & x \le a \text{ atau } x \ge c \\ (x-a)/(b-a); & a < x \le b \\ (c-x)/(c-b); & b < x < c \end{cases} \quad (2.3)$$



Gambar 2.9 *Fungsi Keangotaan Triangel*

Dengan {a,b,c} adalah himpunan parameter. Parameter dalam lapis ini disebut parameter premis yang adaptif.

b. Lapis 2

Setiap simpul pada lapis ini adalah simpul tetap berlabel ∏ dengan keluarannya adalah produk dari semua sinyal yang datang.

$O_{2,i} = \mu_{Ai}(X) \times \mu_{Bi}(X)$, I = 1,2; …… (*2.4*)

Setiap keluaran simpul dari lapis ini menyatakan kuat penyulutan (*fire strength*) dari tiap aturan *fuzzy*. Fungsi ini dapat diperluas apabila bagian premis memiliki lebih dari dua himpunan *fuzzy*. Banyaknya simpul pada lapisan ini menunjukkan banyaknya aturan yang dibentuk. Fungsi perkalian yang digunakan adalah *interpretasi operator and.*

c. Lapisan 3

Setiap simpul pada lapis ini adalah simpul tetap berlabel N. Simpul ini menghitung rasio dari kuat penyulutan atau fungsi derajat pengaktifan aturan ke i pada lapisan sebelumnya terhadap jumlah semua kuat penyulut dari semua aturan pada lapisan sebelumnya.

$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1+w_2}$, i=1,2; …… (*2.5*)

Keluaran lapis ini disebut kuat penyulut ternormalisasi.

d. Lapis 4

Setiap simpul pada lapis ini adalah simpul adaptif dengan fungsi simpul:

$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i)$, i=1,2; …… (*2.6*)

dengan: wi adalah kuat penyulut ternormalisasi dari lapis 3.

parameter

$\{p_i x + q_i y + r_i\}$ …… (*2.7*)

Adalah himpunan parameter dari simpul ini. Parameter pada lapis ini disebut parameter konsekuensi.

e. Lapis 5

Simpul tunggal pada lapis ini adalah simpul tetap dengan label Σ yang menghitung keluaran keseluruhan sebagai penjumlahan semua sinyal yang datang dari lapisan 4.

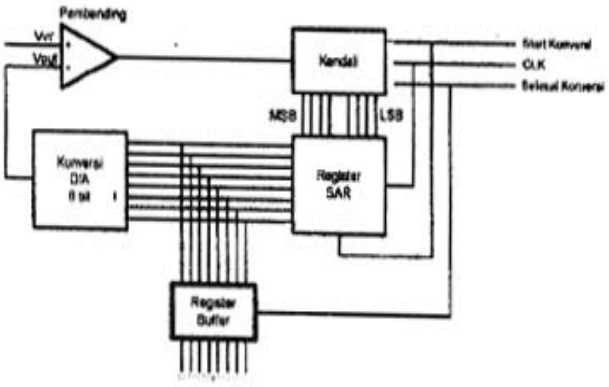
$$O_{5,i} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad \text{..... (2.8)}$$

Jaringan adaptif dengan lima lapisan diatas ekivalen dengan sistem inferensi *fuzzy* Takagi–Sugeno–Kang (TSK) atau yang lebih dikenal dengan Sugeno.

2.3 ADC (*Analog to Digital Converter*)

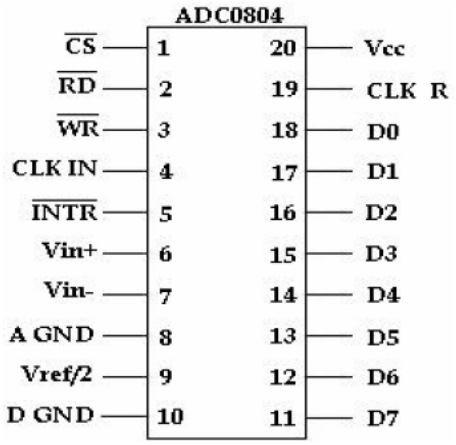
Analog to Digital Converter (ADC) adalah sebuah piranti yang dirancang untuk mengubah sinyal-sinyal *analog* menjadi bentuk sinyal digital. Dalam pembuatan tugas akhir ini, penulis menggunakan IC ADC 0804 yang dapat bekerja secara cermat dengan menambahkan sedikit komponen sesuai dengan spesifikasi yang harus diberikan dan dapat mengkonversikan secara cepat suatu masukan tegangan. Dalam penggunaan ADC ini, hal-hal yang perlu diperhatikan 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 *analog*nya atau sinyal yang akan diubah. Pada gambar 2.10 memperlihatkan diagram blok ADC tersebut.



Gambar 2.10 *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 2.11 *ADC 0804*

Pada gambar 2.11 IC ADC 0804 mempunyai dua *input analog*, $V_{in(+)}$ dan $V_{in(-)}$, 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 :

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

Dengan fungsi ADC yang digunakan sebagai rangkaian yang mengubah sinyal *analog* menjadi sinyal digital, maka 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.

2.3.1 Mode Operasi ADC 0804

Ada beberapa mode operasi yang dapat dilakukan oleh ADC 0804 yaitu :

a. Mode operasi kontinyu

Agar ADC 0804 dapat dioperasikan pada mode operasi kontinyu (proses membaca terus menerus dan tanpa proses operasi jabat tangan), maka penyemat CS dan RD dihubungkan ke *ground*, 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 di-*ground*-kan 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 di-*ground*-kan. Saat sinyal INTR aktif, sinyal ini digunakan untuk me-*reset* SAR. Saat INTR kembali tidak aktif (logika ‘1’) proses konversi dimulai kembali.

b. Mode Operasi *Hand-Shaking*

ADC 0804 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.

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 perderajatnya. 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 beberapa segment kutub kumparan. Gambar dari motor stepper terlihat pada gambar 2.12.



Gambar 2.12 *Motor Stepper*

Pada dasarnya ada dua jenis motor stepper yaitu bipolar dan unipolar. Sebuah motor stepper akan berputar 1 (satu) *step* apabila terjadi perubahan arus pada *koil-koil*nya, mengubah *pole-pole* magnetik disekitar *pole-pole stator*. Perbedaan utama antara bipolar dan unipolar adalah :

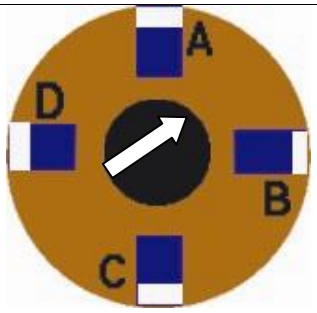
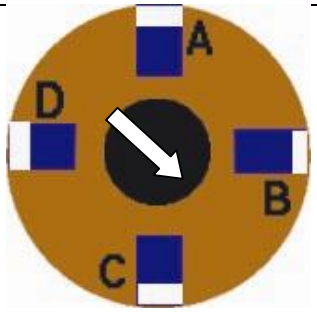
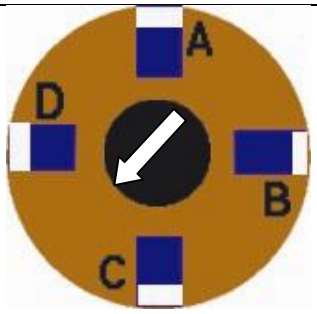
a. Bipolar :
    - Arus pada *koil* dapat berbolak-balik untuk mengubah arah putaran motor.
    - Lilitan motor hanya satu dan dialiri arus dengan arah bolak-balik.
b. Unipolar :
    - Arus mengalir satu arah, dan perubahan arah motor tergantung dari lilitan (*koil*) yang dialiri arus.
    - Lilitan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

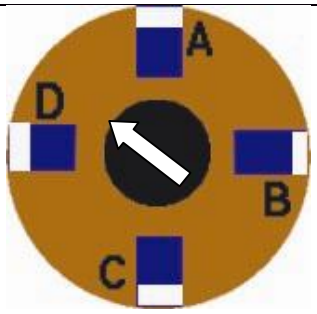
Kelemahan jenis bipolar adalah bahwa rangkaian *driver*nya lebih kompleks, karena harus dapat menglirkan arus dalam dua arah melalui *koil* yang sama. Sedangkan jenis unipolar, selain motor stepper tersebut lebih mudah diperoleh di pasaran juga memerlukan rangkaian *driver* yang lebih sederhana.

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*

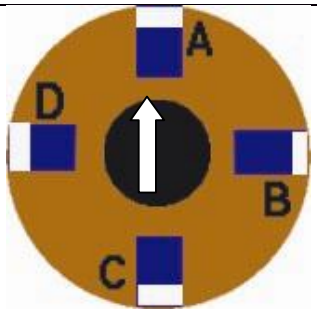
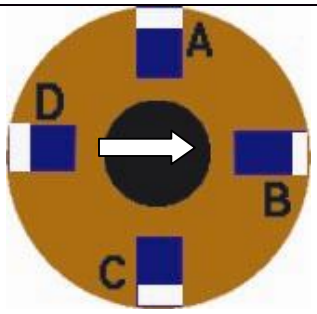
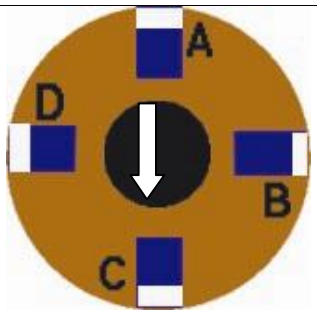
stepping , maka pada *mode half stepping* motor dapat digerakkan sebesar 0.9 derajat atau *step*. Pengaturan kutub-kutub motor dan proses gerak motor stepper dapat dilihat melalui contoh tabel 2.1, tabel 2.2 dan tabel 2.3.

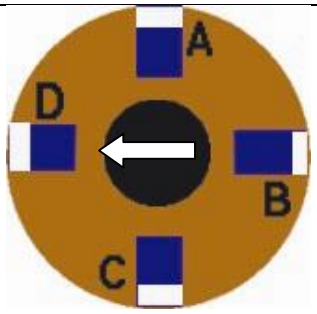
Tabel 2.1 *Mode Full Step Motor Stepper A*

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

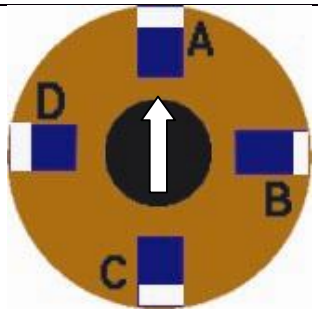
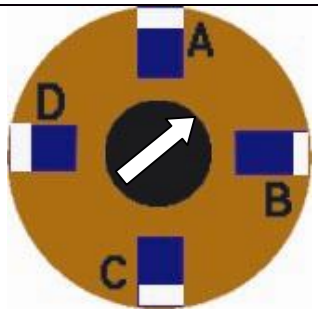
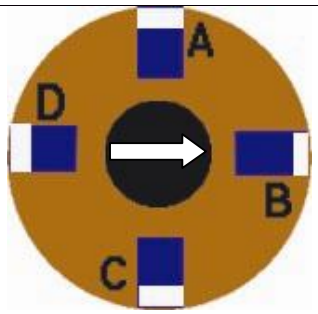
4	ON	OFF	OFF	ON	

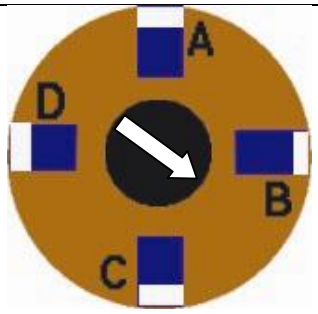
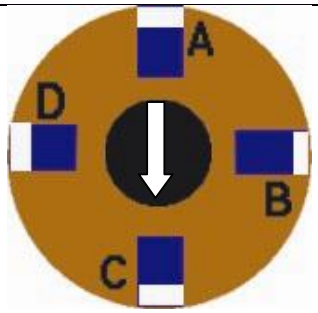
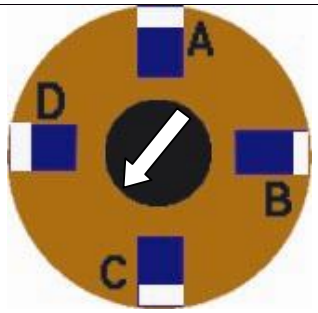
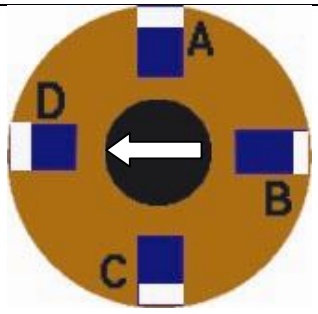
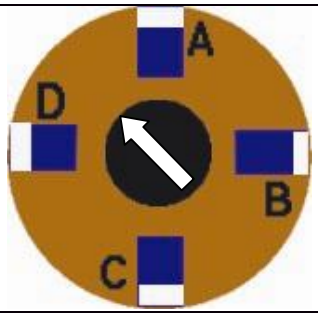
Tabel 2.2 *Mode Full Step Motor Stepper B*

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	

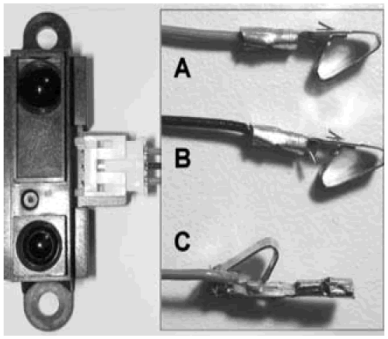
Tabel 2.3 *Mode Half Step Motor Stepper B*

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	

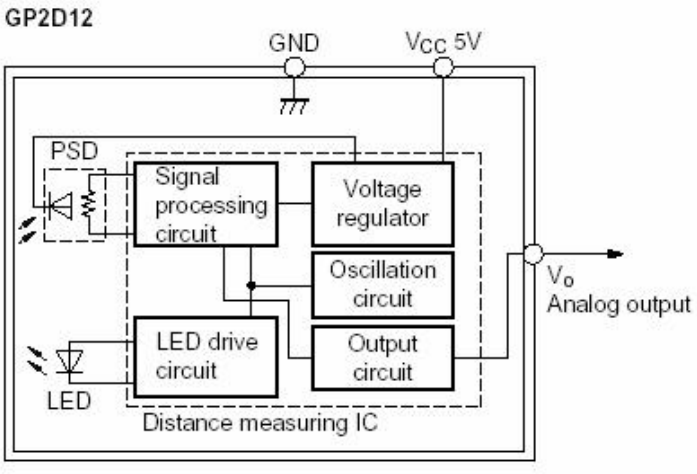
2.5 *Infrared Object Detector*

Dalam pembuatan tugas akhir ini, sensor yang digunakan adalah sensor GP2D12. Sensor Sharp GP2D12 ini merupakan sensor deteksi jarak Infra-Red dengan keluaran tegangan *analog*.



Gambar 2.13 *Sharp GP2D12 dan Kabel Pin*

Blok diagram dari Sharp GP2D12 berisi led pemancar dan penerima yang memiliki rangkaian pemroses, pengemudi, dan rangkaian osilasi serta rangkaian *output analog* seperti gambar berikut:

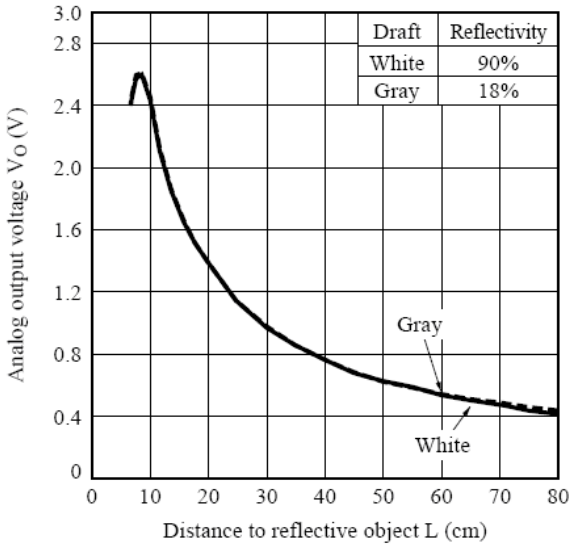


Gambar 2.14 *Blok Diagram GP2D12*

Sensor ini mempunyai *output* sensor 3 kabel yang terdiri dari (gambar 2.13) :

- Pin A sebagai tegangan *output* (Vo)
- Pin B, GND
- Pin C, Vcc

Sensor Sharp GP2D12 mendeteksi bacaan terus menerus ketika diberi daya. *Output*nya berupa tegangan *analog* yang sesuai dengan jarak yang diukur. Jarak yang bisa dideteksi oleh sensor GP2D12 mulai dari 8cm sampai 80cm. Nilai *output* tersebut akan terus diperbaharui setiap 32 ms. Berdasarkan pengukuran, tegangan yang dihasilkan pada jarak 10cm ialah 2.6V, dan menurun tidak secara linear pada jarak 80 cm beriksar pada tegangan 0.5 V. Sehingga dengan kata lain jarak berbanding terbalik dengan tegangan, jadi tegangan akan semakin tinggi pada saat jarak semakin dekat.



Gambar 2.15 *Karakteristik Sensor Sharp GP2D12*

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 2.16.

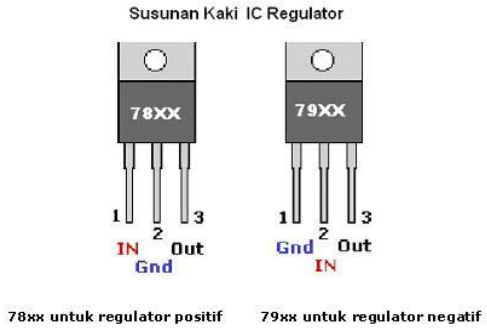


Gambar 2.16 *Transitor TIP 122*

2.7 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 adalah 100 mA, sedangkan untuk tipe

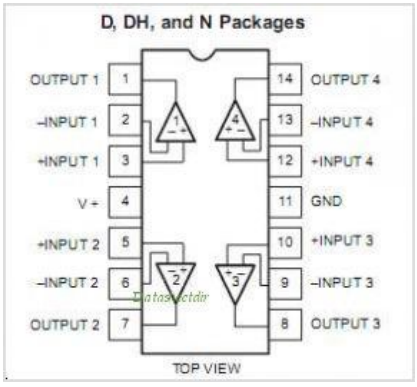
biasa seperti 7805 arus *output* antara 500 mA sampai dengan 1 A, gambar 2.17 menunjukkan susunan kaki dari IC 7805 dan 7812.



Gambar 2.17 *Susunan Kaki IC 7805 dan 7812*

2.8 Penguat Operasional

Penguat operasional (bahasa Inggris: *operational amplifier*) atau yang biasa disebut op-amp merupakan suatu jenis penguat elektronika dengan sambatan (bahasa Inggris: *coupling*) arus searah yang memiliki bati (*faktor penguatan* atau dalam bahasa Inggris: *gain*) sangat besar dengan dua masukan dan satu keluaran. Penguat operasional pada umumnya tersedia dalam bentuk sirkuit terpadu dan pada proyek tugas akhir ini op-amp yang dipakai adalah LM324N.

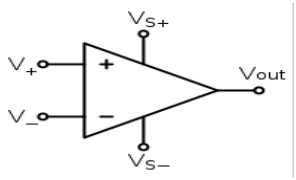


Gambar 2.18 *Datasheet LM324N*

Penguat operasional adalah perangkat yang sangat efisien dan serba guna. Contoh penggunaan penguat operasional adalah untuk operasi matematika sederhana seperti penjumlahan dan pengurangan terhadap tegangan listrik hingga dikembangkan kepada penggunaan aplikatif seperti komparator dan osilator dengan distorsi rendah.

Penguat operasional dalam bentuk rangkaian terpadu memiliki karakteristik yang mendekati karakteristik penguat operasional ideal tanpa perlu memperhatikan apa yang terdapat di dalamnya. Karakteristik penguat operasional ideal adalah:

1. Bati tegangan tidak terbatas.
2. Impedansi masukan tidak terbatas.
3. Impedansi keluaran nol.
4. Lebar pita tidak terbatas.
5. Tegangan ofset nol (keluaran akan nol jika masukan nol).



Gambar 2.19 *Simbol Penguat Operasional*

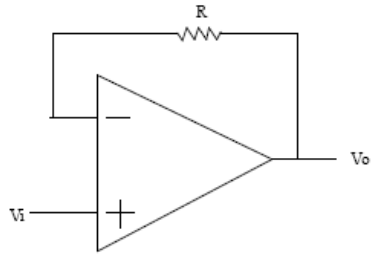
Simbol penguat operasional pada rangkaian seperti pada gambar 2.19, di mana:

1. V_+: masukan non-pembalik
2. V_-: masukan pembalik

3. V_{out}: keluaran
4. V_{S+}: catu daya positif
5. V_{S-}: catu daya negative

Catu daya pada notasi penguat operasional seringkali tidak dicantumkan untuk memudahkan penggambaran rangkaian.

Terdapat banyak sekali penggunaan dari penguat operasional dalam berbagai jenis sirkuit listrik. Oleh karena itu, pada tugas akhir ini hanya memakai penguat operasional sebagai buffer. Rangkaian buffer adalah rangkaian yang *input*nya sama dengan hasil *output*nya. Dalam hal ini seperti rangkaian common colektor yaitu berpenguatan = 1. Rangkaiannya seperti pada gambar berikut ini



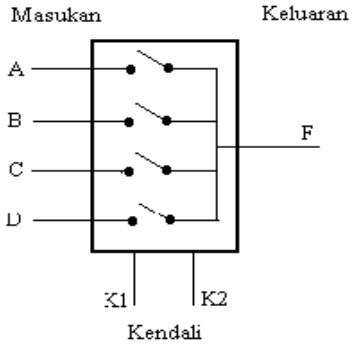
Gambar 2.20 *Rangkaian Buffer*

Nilai R yang terpasang gunanya untuk membatasi arus yang di keluarkan. Besar nilainya tergantung dari indikasi dari komponennya, biasanya tidak dipasang alias arus dimaksimalkan sesuai dengan kemampuan op-ampnya.

2.9 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 2.21.



Gambar 2.21 *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.

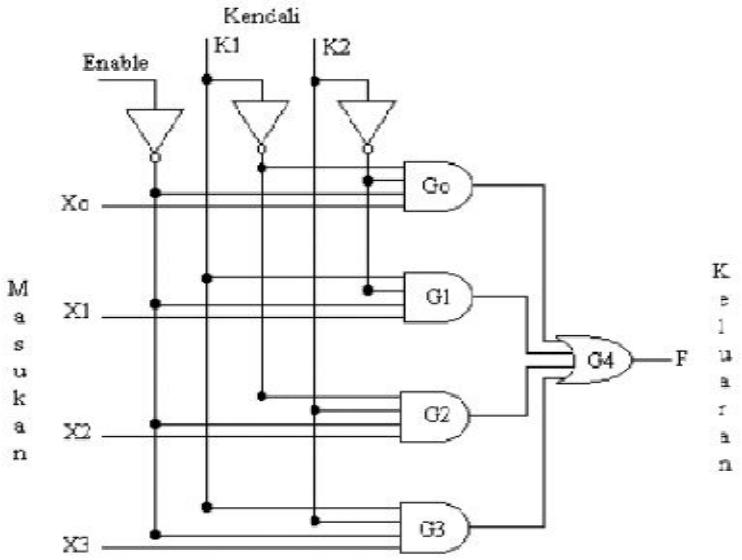
Tabel 2.4 *Tabel Kebenaran Multiplekser*

$\overline{\textbf{Enable}}$	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 2.22.



Gambar 2.22 *Rangkaian Logika Multiplekser 4 ke 1*

2.10 Komunikasi Paralel

Komunikasi paralel adalah komunikasi yang mengirimkan data secara bersamaan. Sehingga pada komunikasi ini kita membutuhkan banyak kabel. Hal ini yang sering menjadi kelemahan. Oleh karena itu, akibat banyaknya kabel yang dibutuhkan, maka panjang kabel yang digunakan tidak boleh lebih dari 20 m, untuk menjaga keaslian data. Sedangkan kelebihan komunikasi paralel adalah pengiriman data lebih cepat dan kapasitas yang dibawa juga banyak serta pemograman 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*

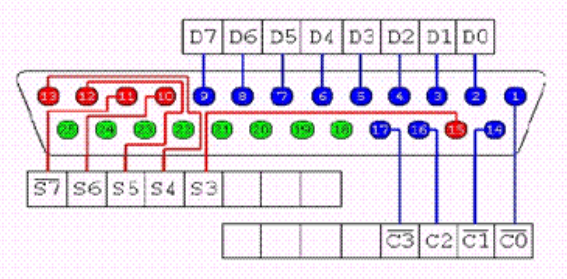
Ragister kontrol	**Register status**	**Register data**	**Register address**
37A	379	378	37B

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 2.23.



Gambar 2.23 *Konfigurasi Pin Paralel Port*