BAB II
LANDASAN TEORI

Dalam ilmu kesehatan perlu dipantau perkembangan gizi seseorang agar selalu terpantau, tidak kekurangan serta tidak kelebihan. Untuk memantau perkembangan gizi tersebut diperlukan sebuah pengukuran yaitu antara tinggi badan dan berat badan yang akan dimasukkan dalam sebuah rumus sehingga didapat berat badan ideal. Selama ini pemantauan dilakukan secara manual, perhitungan berat badan ideal ini pun di lakukan secara manual pula. Sehingga cukup mempersulit bidang yang bersangkutan untuk melakukan riset mereka. Ditambah lagi dengan sulitnya menjangkau lokasi yang terpencil, sehingga laporan dari daerah terpencil itu lama sampai ke pusat dan riset pun akan berlangsung lama. Dalam BAB ini akan dijelaskan beberapa teori tentang perancangan alat pengukur berat badan ideal sehingga mempercepat proses riset serta pengambilan data. Termasuk juga tentang perancangan *hardware* dan *software* yang digunakan.

2.1 Teori Berat Badan Ideal

Pengertian berat badan ideal adalah seseorang yang mempunyai bentuk tubuhnya tidak terlalu kurus , tidak terlalu gemuk terlihat serasi antara berat badan dan tinggi badan. Untuk menunjang kehidupan, di dalam tubuh harus ada lemak minimal 3% dari berat badan ideal baik pada wanita maupun pria, yang disebut sebagai *lemak esensial.* Lemak dalam tubuh yang jumlahnya melebihi 3% dari berat badan dianggap sebagai timbunan lemak (Budiyanto,2002). Secara medis postur tubuh ideal dapat dinilai dari pengukuran antropometri. Pengukuran antropometri yang paling sering digunakan adalah rasio antara berat badan (kg) dan tinggi badan (m) kuadrat, yang disebut Indeks Massa Tubuh (IMT). Antropometri adalah ilmu yang berkaitan dengan pengukuran manusia, termasuk di dalamnya cara menghitung berat badan ideal.

$$\text{IMT} = \frac{BB\ (Kg)}{TB \times TB\ (m)}$$

BB = Berat Badan

TB = Tinggi badan

(2.1)

Cara lain yang lebih mudah adalah menggunakan perhitungan BB ideal menggunakan metode Brocca. Brocca membuat definisi berat badan ideal dengan rumus: (Tinggi Badan - 100) - 10% (Tinggi Badan - 100). Cara pengukuran metode brocca ini sudah terkenal di kalangan orang awam karena paling mudah dimengerti dan digunakan. Batas ambang berat badan ideal adalah jika kelebihan berat badannya ± 10% dari berat badan ideal. Bila > 10% dikatakan mengalami kegemukan dan bila diatas 20% sudah terjadi obesitas (Depkes,2004).

Tabel II.1 Kriteria Status Gizi berdasarkan Index Massa Tubuh (Despkes,2004)

Status Gizi	Wanita	Laki-laki
Normal	17 -23	18 –25
Kegemukan	23 – 27	25 – 27
Obesitas	> 27	> 27

2.2 Jenis-jenis Timbangan

Timbangan merupakan sebuah alat ukur yang menghitung massa atau berat suatu benda. Timbangan dibagi dalam beberapa jenis sesuai dengan kebutuhan yang diperlukan. Timbangan dalam skala kecil digunakan untuk mengukur benda yang maksimum beratnya sesuai dengan timbangannya. Timbangan dalam skala besar disesuaikan dengan benda yang akan diukur dan dibuat dengan daya tahan yang besar pula.

2.2.1 Timbangan Mekanik

Timbangan/neraca mekanik biasanya menggunakan pegas sebagai mekanik utamanya. Salah satu contoh timbangan adalah neraca pegas (dinamometer). Neraca pegas adalah timbangan sederhana yang menggunakan pegas sebagai alat untuk menentukan massa benda yang diukurnya. Neraca pegas (seperti timbangan badan) mengukur berat, defleksi pegasnya ditampilkan dalam skala massa (label angkanya sudah dibagi gravitasi).

Persamaan matematis suatu neraca pegas dinyatakan dalam:

$$k \times X = m \times g \tag{2.2}$$

dengan

k = konstanta pegas

X = defleksi

m = massa

g = gravitasi

Neraca/timbangan dengan bandul pemberat (seperti yang terdapat di pasar ikan/sayur) menimbang massa. Biasanya menggunakan massa pembanding yang lebih kecil dengan lever (tuas) yg panjang. Mengikuti hukum tuas (persamaan momen).

$$m_1 \times g \times L_1 = m_2 \times g \times L_2 \tag{2.3}$$

dengan

m_1,m_2 = massa benda pertama, massa benda kedua

L_1,L_2 = panjang tuas pertama, panjang tuas kedua

g = gravitasi

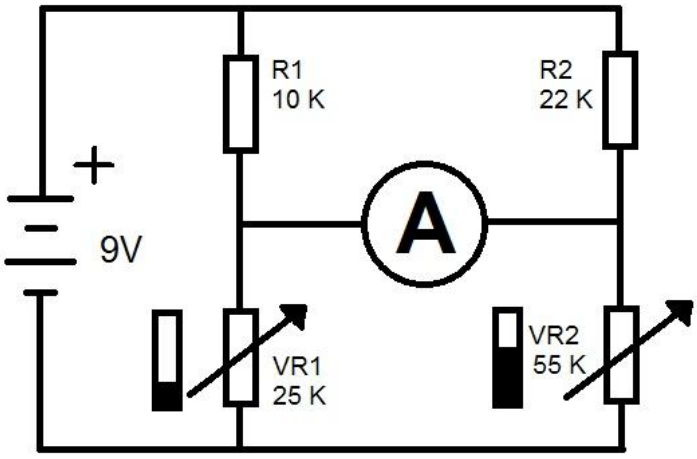
Neraca pegas menunjukkan angka yang berbeda di bumi dan bulan, atau di daerah yang gravitasinya berbeda. Timbangan bandul menunjukkan angka yg sama di mana pun, asal masih ada gravitasi untuk menggerakkan timbangan.

2.2.2 Timbangan Digital

Seiring dengan perkembangan teknologi yang membutuhkan keakuratan dan presisi yang tinggi, maka timbangan pun ikut masuk kedalamnya. Timbangan digital dibuat dengan sensor tekanan yang sebelumnya telah dikalibrasi dengan timbangan mekanik. Banyak jenis timbangan digital yang ada dipasaran dan juga sesuai dengan kebutuhannya. Itu juga didukung dengan sensor yang beragam dan sesuai dengan apa yang dibutuhkan. Keunggulan timbangan digital ini mampu menghitung massa dengan ukuran yang kecil dan akurat. Pada beberapa aplikasi timbangan digital mampu mengukur massa dengan hitungan skala 0.001 Kg bahkan lebih kecil lagi. Sehingga keakuratan pengukuran juga akan tercapai dengan baik.

2.3 Teori Jembatan Wheatstone

Prinsip dasar dari jembatan wheatstone adalah keseimbangan. Sifat umum dari arus listrik adalah arus akan mengalir menuju polaritas yang lebih rendah. Jika terdapat persamaan polaritas antara kedua titik maka arus tidak akan mengalir dari kedua titik tersebut. Dalam rangkaian dasar jembatan wheatstone penghubung kedua titik tadi disebut sebagai jembatan wheatstone.



Gambar II.1 Rangkaian Dasar Jembatan Wheatstone

Dari gambar di atas jembatan wheatstone terhubung dari dua buah rangkaian pembagi tegangan. Masing-masing menggunakan satu buah potensio sebagai pengatur pembagian tegangan. Besar tegangan pada titik jembatan pertama (titik A) adalah (75K / (75K + 10K)) × 9 volt = (75/85) 9 volt = 7.94 volt. Kemudian besar tegangan pada titik B adalah ((65K / (65K+10K)) × 9 volt = (65/75) × 9 volt = 7.8 volt. Maka dapat dari perhitungan tersebut dapat dipastikan bahwa arus mengalir dari titik A pada rangkaian pertama ke titik B pada jembatan rangkaian kedua, dikarenakan tegangan pada titik A lebih besar dari tegangan pada titik B. Jika kita pasang sebuah beban pada jembatan wheatstone maka untuk menentukan besar tegangan yang jatuh pada beban tersebut adalah selisih tegangan antara kedua titik jembatan tersebut (7.94 – 7.8 = 0.14 volt). Arus yang mengalir pada beban tersebut = 0.14 volt / Rbeban.

Analisa dari rangkaian diatas adalah :

VAB = VA-VB = 7.94 – 7.8 = 0.14 volt

VBA = VB-VA = 7.8 – 7.94 = - 0.14 volt

VA = VAB + VB (dikarenakan terhubung seri) = 0.14 + 7.8 = 7.94 volt

VB = VBA + VA (dikarenakan terhubung seri) = - 0.14 + 7.94 = 7.8 volt

Keterangan :

VAB : Tegangan yang diukur pada titik A dan B

VBA : Tegangan yang diukur pada titik B dan A

VA : Tegangan yang diukur pada titik A dan Ground

VB : Tegangan yang diukur pada titik B dan Ground.

Kesimpulan yang bisa kita tarik dari teori jembatan wheatstone adalah :

1. Tegangan yang diukur pada dua titik yang mempunyai polaritas yang sama adalah 0 volt.
2. Tegangan pada jembatan adalah selisih tegangan antara kedua polaritas tersebut.

3. Arus akan mengalir dari titik jembatan yang berpolaritas tinggi ke titik yang berpolaritas rendah.

Arus yang mengalir pada dua titik yang mempunyai polaritasyang sama adalah 0 ampere.

2.4 Komunikasi Serial RS232

Komunikasi serial merupakan hal yang penting dalam sistem *embedded*, karena dengan komunikasi serial kita dapat dengan mudah menghubungkan mikrokontroler dengan peralatan lainnya. *Port* serial pada mikrokontroler terdiri atas dua pin yaitu RXD dan TXD. RXD berfungsi untuk menerima data dari komputer atau perangkat lainnya, TXD berfungsi untuk mengirim data ke komputer atau perangkat lainnya. Standar komunikasi serial untuk komputer ialah RS-232. RS-232 mempunyai standar tegangan yang berbeda dengan serial port mikrokontroler, sehingga agar sesuai dengan RS-232 maka di butuhkan suatu rangkaian *level converter*, IC yang digunakan bermacam-macam, tetapi yang paling mudah dan sering digunakan ialah IC MAX232.

RS-232 merupakan standar komunikasi serial yang didefinisikan sebagai antarmuka antara perangkat terminal data (*data terminal equipment*) atau DTE dan perangkat komunikasi data (*data communications equipment*) atau DCE menggunakan pertukaran data biner secara serial. Standar RS232 ditetapkan oleh Electronic Industry Association and Telecomunication Industry Association pada tahun 1962. Nama lengkapnya adalah EIA/TIA-232 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange. Meskipun namanya cukup panjang tetapi standar ini hanya menyangkut komunikasi data antara komputer dengan alat-alat pelengkap komputer.

2.5 Mikrokontroler Basic Stamp

Mikrokontroler adalah sebuah IC yang berfungsi sebagai pengendali perangkat-perangkat yang terhubung dengan mikrokontroler tersebut. *Basic stamp* adalah mikrokontroler yang dikembangkan oleh Parallax Inc yang mudah diprogram menggunakan format bahasa pemrograman *basic*. Program yang dibuat diunduh melalui *port* serial dengan menggunakan *konverter USB to Serial* untuk komputer yang tidak memiliki *port* serial, serta membutuhkan *power supply* saat mengunduh program.

Beberapa macam versi dari *basic stamp* yaitu, *basic stamp* 1, *basic stamp* 2, *basic stamp 1e*, *basic stamp 2P*, *basic stamp 2Pe* dan *basic stamp 2sx*. *Basic stamp* bekerja pada tegangan DC 5 *volt* sampai 15 *volt*. *Basic stamp* yang di pakai adalah *basic stamp* BS2P40 yang mempunyai 40 pin I/O. Berikut adalah gambar dari *basic stamp* BS2P40.



Gambar II.2 Modul Basic Stamp (BS2P40)

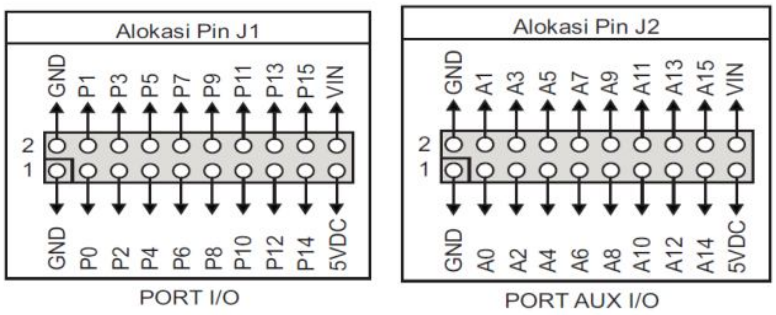
Pada mikrokontroler *basic stamp* ini memiliki spesifikasi sebagai berikut:

1. Mikrokontroler *basic stamp* BS2P40 *Interpreter Chip* (PBASIC48W/P40)
2. 8 x 2*Kbyte* EEPROM yang mampu menampung hingga 4000 instruksi.
3. Kecepatan prosesor 20MHz Turbo dengan kecepatan eksekusi program hingga 12000 instruksi per detik.
4. RAM sebesar 38*byte* (12 I/O, 26 variabel) dengan *Scratch Pad* sebesar 128 *byte*.
5. Jalur *input* / *output* sebanyak 32 pin.
6. Tersedia jalur komunikasi serial UART RS-232 dengan konektor DB9.

7. Tegangan *input* 9 – 12 V_{DC} dengan tegangan *output* 5 V_{DC}.

2.5.1 Konfigurasi PIN Mikro Basic Stamp

Berikut ini adalah alokasi pin yang terdapat pada mikrokontroler BS2P40.



Gambar II.3 Konfigurasi PIN Mikrokontroler Basic Stamp

Tabel II.2 Diskripsi Pin BS2P40

Pin	**Nama**	**Keterangan**
1	SOUT	serial *out* untuk pemrograman yang terkoneksi ke PC pada *port* di pin RX(DB9 PIN2/DB25 PIN3)
2	SIN	serial *input* untuk pemrograman yang terkoneksi ke PC pada *port* di pin TX(DB9 PIN3/DB25 PIN2)
3	ATN	serial data untuk pemrograman yang terkoneksi ke PC pada *port* di pin DTR (DB9 PIN4/DB25 PIN20)
4	VSS	serial data untuk pemrograman yang terkoneksi ke PC pada *port* di pin DTR (DB9 PIN5/DB25 PIN7)
5-20	P0-P15	PIN I/O dimana logika *high* = 5 V dan *Low*= 0V

21-36	X0-X15	PIN I/O sekunder (*Auxiliary*) dimana logika *high* = 5 V dan *Low* = 0 V
21&37	VDD	pin *input* tegangan sebesar 5V
22 & 38	RES	PIN *reset*
23 & 39	VSS	PIN *Ground* mikrokontroler
24 & 40	VIN	PIN *input* tegangan yang dilewatkan regulator 5V, membutuhkan sumber sebesar 5,5 -12VDC

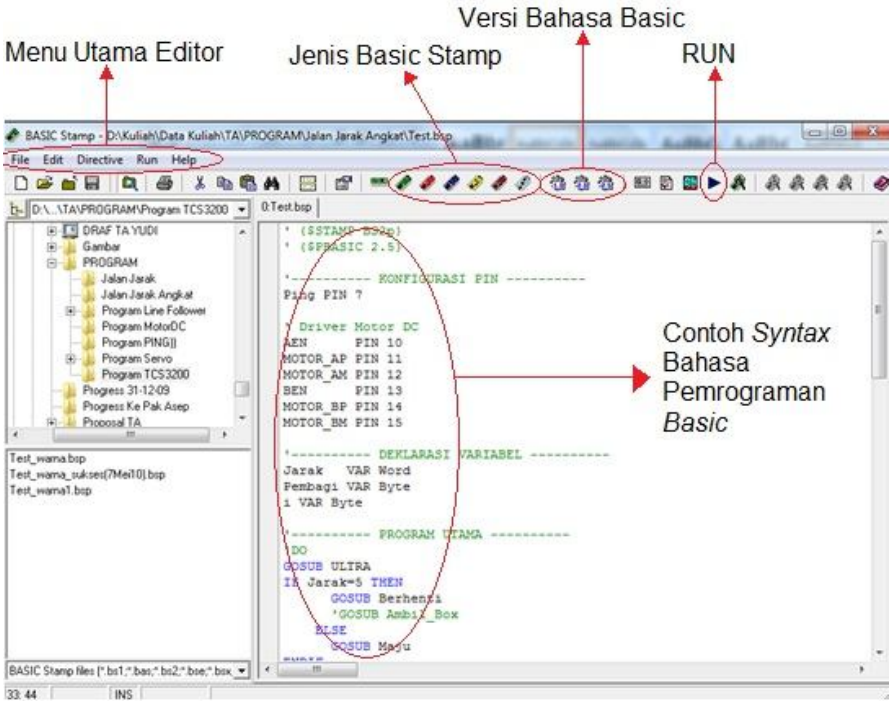
2.5.2 Basic Stamp Editor

Perangkat lunak ini merupakan algoritma gerak dan tugas *flying robot* dalam bentuk listing program yang disimpan kedalam mikrokontroler. Mikrokontroler BS2P40 menggunakan bahasa pemrograman *basic*. *Software* yang digunakan adalah *basic stamp editor*. *Basic stamp editor* adalah sebuah *editor* yang dibuat oleh Paralax Inc untuk menulis program, menjalankan dan mengunduhnya ke mikrokontroler keluarga *basic stamp*. Program ini memungkinkan penggunanya memprogram *basic stamp* dengan bahasa *basic* yang relatif ringan dibandingkan bahasa pemrograman lainnya. Berikut ini beberapa instruksi-instruksi dasar yang dapat digunakan pada mikrokontroler *basic stamp*.

Tabel II.3 Beberapa instruksi dasar basic stamp

Instruksi	**Keterangan**
DO...LOOP	Perulangan
GOSUB	Memanggil prosedur
IF..THEN	Percabangan

FOR...NEXT	Perulangan
PAUSE	Waktu tunda milidetik
IF...THEN	Perbandingan
PULSOUT	Pembangkit pulsa
PULSIN	Menerima pulsa
GOTO	Loncat ke alamat memori tertentu
HIGH	Membuat pin I/O menjadi logika 1
LOW	Membuat pin I/O menjadi logika 0
PWM	Konversi suatu nilai digital ke keluaran analog lewat *pulse width modulation*



Gambar II.4 Tampilan Basic Stamp Editor

2.5.3 Memprogram Basic stamp

1. *Directive*

Directive ditulis di awal program. Bagian ini menentukan tipe prosesor yang digunakan dan versi *dari compiler PBASIC* yang digunakan untuk meng*kompile* bahasa basic menjadi bahasa mesin. Tampilannya adalah seperti gambar berikut :

```
' {$STAMP BS2sx}     ' pemilihan tipe prosesor
' {$PBASIC 2.5}      ' pemilihan versi kompiler PBASIC
```

Gambar II.5 Tampilan bagian directive

2. Menentukan Variabel

Menentukan PIN mikrokontroler yang digunakan serta membuat variabel. Ada beberapa ketentuan untuk mendeklarasikan variabel yaitu :

1. PIN : PIN dari mikrokontroler (0-15)
2. VAR : Variabel
3. CON : Konstanta

PIN yang digunakan sudah ditentukan sesuai dengan konfigurasi *hardware* /*mainboard* yang digunakan adalah BS2P40. Selain itu dapat membuat variabel bebas yang nantinya dapat digunakan untuk keperluan perulangan atau yang lainnya.

Setelah menentukan variabel dan PIN yang digunakan, selanjutnya membuat program utama. Pada bagian program utama bisa melakukan dua mode program, yaitu program dengan pengetikan langsung atau program dengan pemanggilan prosedur. Program pengetikan lebih efektif jika program tidak terlalu banyak dan hanya untuk menangani kasus yang sederhana. Sedangkan untuk program yang banyak, rumit dan lebih dari satu slot, maka sebaiknya menggunakan program prosedur.

3. Memeriksa Sintaks Program

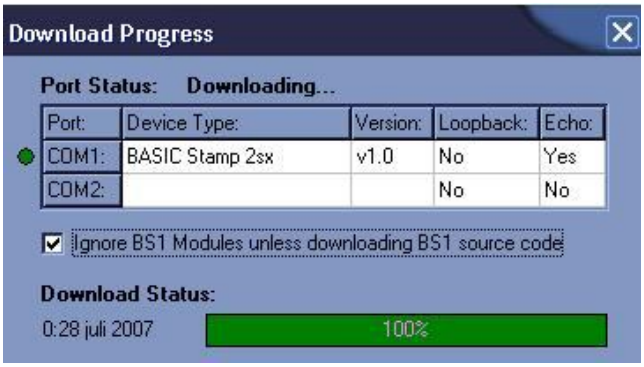
Memeriksa sintaks program lakukan untuk memastikan semua sintaks sudah benar. Untuk memeriksa sintaks ini bisa pilih menu *RUN*, Cek Sintaks atau kombinasi tombol CTRL+T. Tampilan jika *listing program* yang kita buat sudah benar.



Gambar II.6 Hasil pemeriksaan sintak yang sukses (Tokenize Successful)

4. Menjalankan Program

Setelah program selesai, program siap di unduh ke modul *basic stamp*. Cara untuk menjalankan program dapat memilih menu *RUN* atau kombinasi tombol CTR+R. Tampilan jika pengunduhan program sukses.



Gambar II.7 Tampilan jika pengunduhan program sukses

2.6 Penyearah (Rectifier)

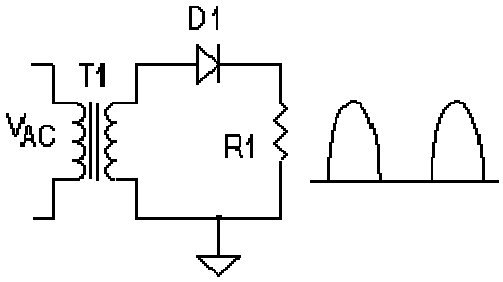
Penyearah adalah proses merubah atau menyearahkan arus bolak-balik menjadi arus searah. Arus bolak balik-balik ini berasal dari tegangan jala-jala. Secara umum penyearah dibagi menjadi tiga kategori yaitu :

1. Penyearah setengah gelombang
2. Penyearah gelombang penuh sistem CT
3. Penyearah gelombang penuh sistem jembatan

Sedangkan komponen utama yang diperlukan dalam penyearahan adalah transformator,dioda dan kapasitor elektrolit.

2.6.1 Penyearah Setengah Gelombang

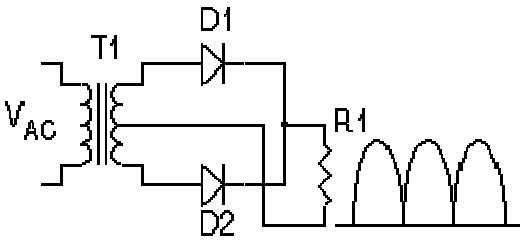
Penyearah ini bekerja dengan menggunakan satu dioda, sehingga hanya pulsa positif yang dapat terambil . Penyerah ini praktis sederhana, tetapi kekurangannya adalah bahwa gelombang keluaran bukan gelombang penuh sehingga rentan sekali akan *ripple.*



Gambar II.8 Penyearah setengah gelombang

2.6.2 Penyearah Gelombang Penuh Sistem CT

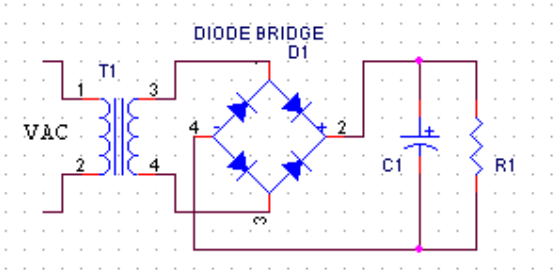
Penyearah ini menggunakan transformator jenis CT dengan dua buah dioda sebagai penyearah. Dioda bekerja secara bergantian untuk mengambil pulsa positif dan negatif, sehingga keluaran berupa gelombang penuh.



Gambar II.9 Penyearah sistem CT

2.6.3 Penyearah Gelombang Penuh Sistem Jembatan

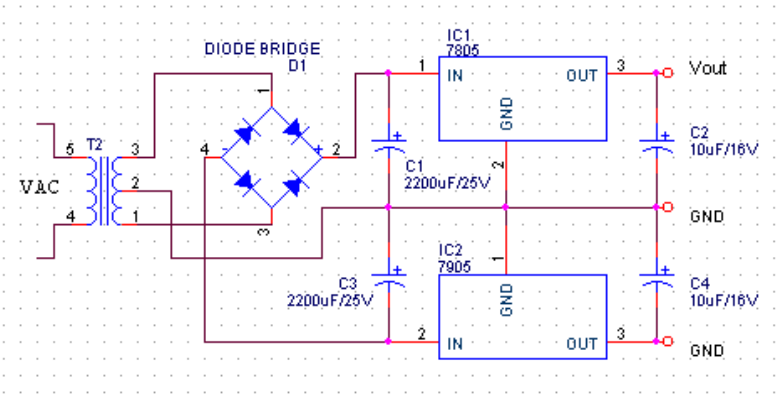
Penyearah ini menggunakan 4 buah dioda sebagai penyearah. Pada siklus pertama dua dioda bekerja untuk menyearahkan atau mengambil pulsa positif. Siklus selanjutnya dua dioda berikutnya yang bekerja untuk mengambil pulsa negatif. Keuntungan penyearah ini adalah bahwa keluaran berupa gelombang penuh dan jika salah satu dioda rusak, maka dioda yang satunya lagi akan tetap bekerja.



Gambar II.10 Penyearah sistem jembatan

2.6.4 Penyearah Teregulasi

Tegangan hasil penyearah belum tentu stabil pada satu titik yang diinginkan, untuk itu harus ada proses untuk menstabilkan tegangan tersebut. Hal ini dapat dilakukan dengan menambahkan komponen pada keluaran penyearah, diantaranya menggunakan dioda zener, penguat operasional atau dengan IC regulator.



Gambar II.11 Penyearah teregulasi menggunakan IC

2.7 Regulator

Rangkaian penyearah sudah cukup bagus jika tegangan *ripple*-nya kecil, namun ada masalah stabilitas. Jika tegangan PLN naik/turun, maka tegangan keluarannya juga akan naik/turun. Untuk beberapa aplikasi perubahan tegangan

ini cukup mengganggu, sehingga diperlukan komponen aktif yang dapat meregulasi tegangan keluaran ini menjadi stabil.

Saat ini sudah banyak dikenal komponen seri **78XX** sebagai regulator tegangan tetap positif dan seri **79XX** yang merupakan *regulator* untuk tegangan tetap negatif. Bahkan komponen ini biasanya sudah dilengkapi dengan pembatas arus (*current limiter*) dan juga pembatas suhu (*thermal shutdown*). Komponen ini hanya tiga pin dan dengan menambah beberapa komponen saja sudah dapat menjadi rangkaian catu daya yang ter-regulasi dengan baik.



Gambar II.12 IC Regulator 78xx/79xx

Misalnya 7805 adalah *regulator* untuk mendapat tegangan 5 *volt*, 7812 *regulator* tegangan 12 *volt* dan seterusnya. Sedangkan seri 79XX misalnya adalah 7905 dan 7912 yang berturut-turut adalah *regulator* tegangan negatif 5 dan negatif 12 *volt.*

Selain dari *regulator* tegangan tetap ada juga IC *regulator* yang tegangannya dapat diatur. Prinsipnya sama dengan *regulator* OP-amp yang dikemas dalam satu IC misalnya LM317 untuk *regulator variable* positif dan LM337 untuk *regulator variable* negatif. Bedanya resistor R1 dan R2 ada di luar IC, sehingga tegangan keluaran dapat diatur melalui resistor *eksternal* tersebut.

Hanya saja perlu diketahui supaya rangkaian *regulator* dengan IC tersebut bisa bekerja, tengangan *input* harus lebih besar dari tegangan *output regulator*-nya. Biasanya perbedaan tegangan V_{IN} terhadap V_{OUT} yang direkomendasikan ada di dalam *datasheet* komponen tersebut. Pemakaian *heatshink* (aluminium

pendingin) dianjurkan jika komponen ini dipakai untuk men-catu arus yang besar. Di dalam *datasheet,* komponen seperti ini maksimum bisa dilewati arus mencapai 1 A.

2.8 Penguat Operasional (OP – AMP)

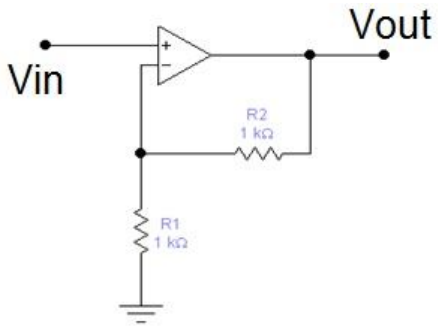
Penguat sinyal berfungsi untuk menguatkan sinyal dari tingkat sebelumnya agar keluaran dari bagian penguat ini dapat diterima oleh tingkat selanjutnya. Terdapat beberapa teknik untuk membentuk sebuah penguat sinyal, misalnya dengan menggunakan transistor atau IC, dalam hal ini penguat sinyal dibentuk oleh penguat operasional *(Operational Amplifier).* Terdapat dua teknik penguatan sinyal di dalam penguat operasional , yaitu dengan penguatan *inverting* dan penguatan *non–inverting*. Masing – masing konfigurasi memiliki kekurangan serta kelebihan yang dapat digunakan sesuai kebutuhan rangkaian. Cara kerja dari masing- masing konfigurasi adalah sebagai berikut .

2.8.1 Penguat Non - Inverting

Penguat *non-inverting* mempunyai impedansi *input* yang tinggi, impedansi *output* yang rendah dan penguatan tegangan yang stabil.

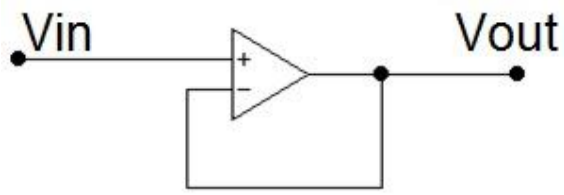
$$\frac{V_{OUT}}{V_{IN}} = \frac{R_2}{R_1} + 1 \qquad (2.4)$$

(Untuk V_{OUT} dan V_{IN} digunakan huruf besar karena penguat operatif dapat bekerja secara langsung dengan sinyal DC). Penguat *non-inverting* dapat populer karena penguat tersebut mendekati penguat tegangan ideal.



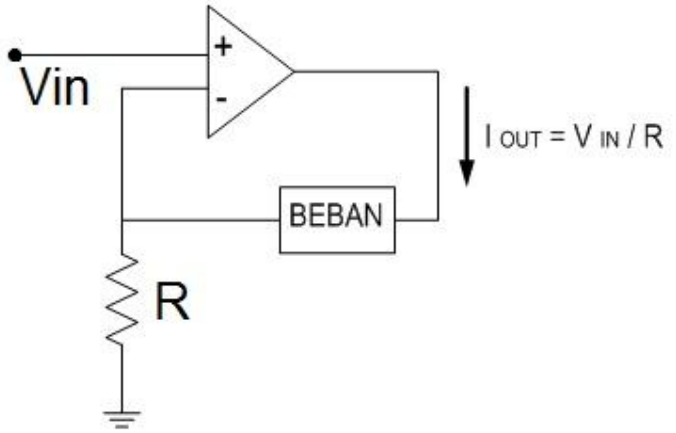
Gambar II.13 Penguat non-inverting

Gambar II.14 adalah pengikut tegangan, yang banyak digunakan karena kualitas *buffer*-nya yang baik sekali , dimana memiliki impedansi input ekstrim tinggi, impedansi output ekstrim rendah dan penguatan tegangan *unity*. Karena dalam sebuah pengikut tegangan umpan balik negatif adalah maksimum, maka lebar pita sama dengan f_{unity}



Gambar II.14 Pengikut tegangan

Pada kondisi tertentu ada kemungkinan perlu memberi arus dalam jumlah yang tetap melalui beban. Gambar II.15 menunjukkan satu cara untuk melakukan hal tersebut. Karena tegangan kesalahan kecil dapat diabaikan, pada dasarnya semua muncul pada R yang menimbulkan arus.



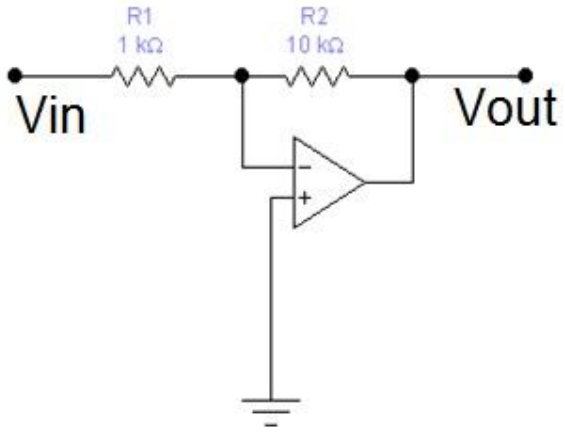
Gambar II.15 Sumber arus

$$I_{OUT} = \frac{V_{IN}}{R} \tag{2.5}$$

Semua arus ini harus mengalir melalui beban, karena arus yang dapat diabaikan mengalir ke dalam *input inverting* dari penguat operatif. Tergantung pada penggunaan, beban dapat berupa resistor, kapasitor, induktor atau gabungan.

2.8.2 Penguat Inverting

Gambar II.16 menunjukkan penguat *inverting*, rangkaian penguat operatif yang sangat populer. Terminal *inverting* pada pertanahan semu *(virtual ground)* yang berarti tegangan terhadap tanah mendekati nol. Tetapi karena pertanahan semu tidak dapat melepaskan arus, semua arus *input* didorong melalui R2.



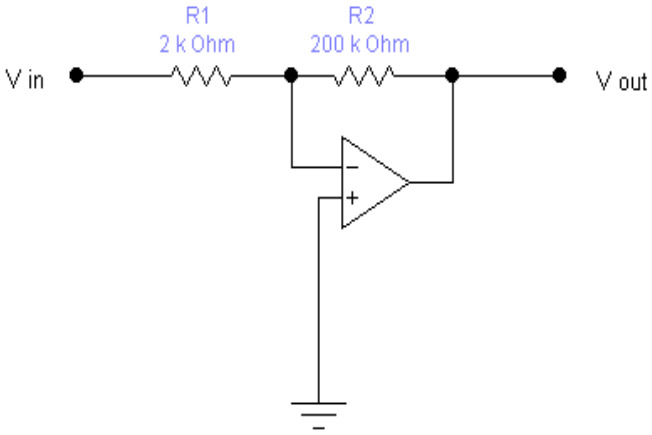
Gambar II.16 Penguat inverting

$$V_{IN} = I_{IN} \times R_1 \tag{2.6}$$

$$V_{OUT} = I_{IN} \times R_2 \tag{2.7}$$

Tanda minus terjadi karena inversi. Dengan mengambil rasio kedua persamaan diatas, diperoleh penguatan tegangan :

$$\frac{V_{OUT}}{V_{IN}} = -\frac{R_2}{R_1} \tag{2.8}$$



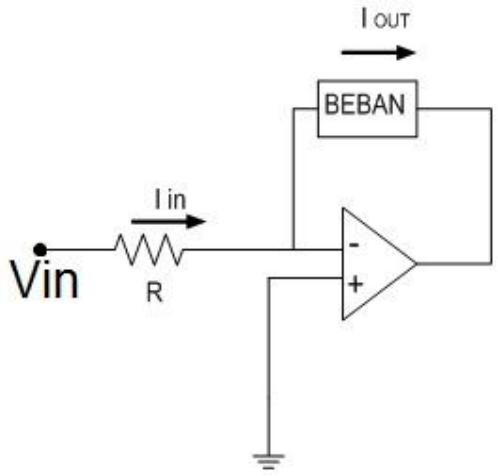
Gambar II.17 Contoh aplikasi penguat inverting

Pentanahan semua impedansi *input* adalah

$$Z_{in} = R_1 \quad (2.9)$$

Salah satu sebab kepopuleran dari penguat *inverting* adalah penguat tersebut memungkinkan kita men-*set* satu harga yang tepat dari impedansi *input*, demikian juga penguatan tegangan. Banyak penggunanan dimana kita ingin memastikan impedansi *input* bersama dengan penguatan tegangan. Sebagai contoh, misalkan kita memerlukan impedansi *input* sebesar 2 KΩ dan penguatan tegangan sebesar 100. Maka tugas ini dapat dilakukan oleh rangkaian seperti gambar II.17. Gambar II.18 berikut menunjukkan penguat *inverting* yang digunakan ke sumber arus melalui beban.

$$I_{OUT} = \frac{V_{IN}}{R} \quad (2.10)$$



Gambar II.18 Sumber Arus

2.9 Motor DC

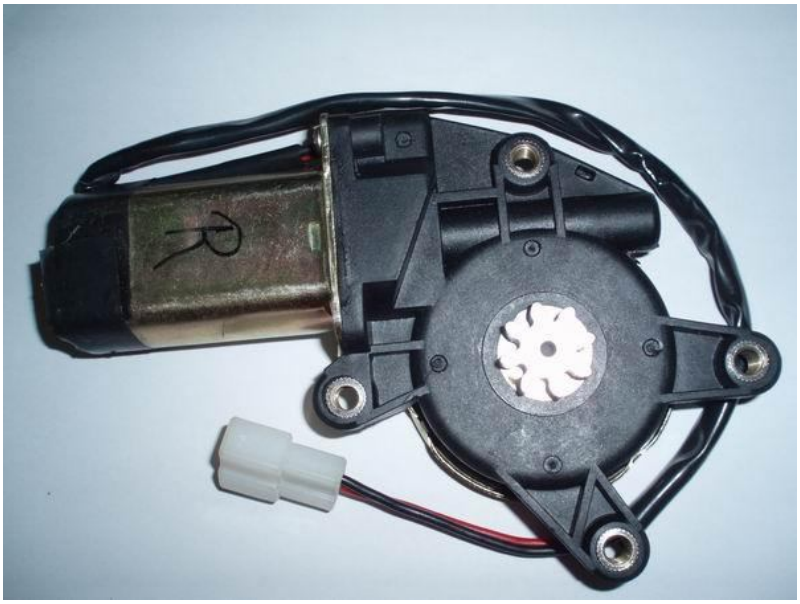
Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor

sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor.

Motor DC memiliki 2 bagian dasar :

1. Bagian yang tetap/*stasioner* yang disebut *stator*. *Stator* ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen.
2. Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir.

Gaya elektromagnet pada motor DC timbul saat ada arus yang mengalir pada penghantar yang berada dalam medan magnet. Medan magnet itu sendiri ditimbulkan oleh megnet permanen. Garis-garis gaya magnet mengalir diantara dua kutub magnet dari kutub utara ke kutub selatan



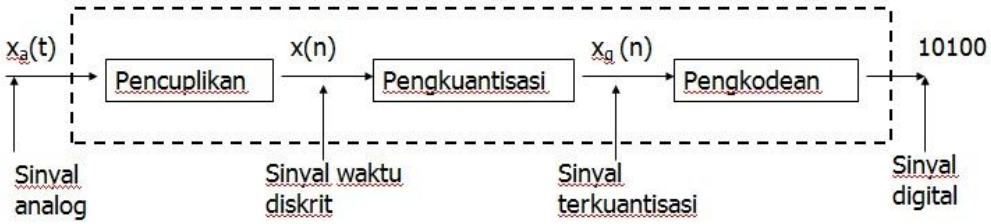
Gambar II.19 Motor DC

2.10 Konversi ADC (Analog to Digital Conventer)

Sebagian besar sinyal di alam ini adalah berbentuk sinyal kontinu (analog), misal : sinyal suara, sinyal seismik, sinyal radar, sinyal sonar, sinyal komunikasi audio dan video, dan lain-lain. Untuk memproses sinyal analog dengan peralatan

digital , maka perlu mengkonversikan ke dalam bentuk digital atau yang lazim disebut konversi analog ke digital (ADC)

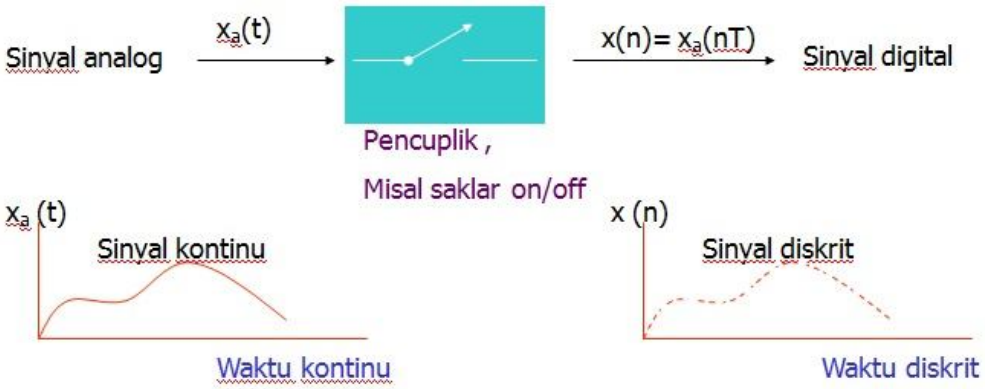
Pengkonversi data pada elektronika adalah suatu alat yang mengubah besaran sinyal dari analog ke digital atau sebaliknya. Umunya, sinyal analog berasal dari suatu sensor. Sinyal DC/AC lemah yang biasanya diperkuat oleh Op-Amp dan diubah menjadi sinyal digital oleh perangkat pengkonversi data (ADC) atau sinyal digital yang umumnya sekitar 8-32 bit yang diubah menjadi sinyal analog (DAC) untuk tujuan tertentu, misalnya pada pemutar musik MP4.



Gambar II.20 Bagian Dasar ADC

Ada 3 langkah konversi ADC seperti pada gambar II.20

1. Pencuplikan : konversi suatu sinyal fungsi waktu- kontinu menjadi suatu sinyal fungsi waktu –diskrit.
2. Kuantisasi : konversi sinyal yang ***bernilai-kontinu*** fungsi waktu diskrit menjadi sinyal ***bernilai-diskrit*** (digital 1 atau 0) fungsi waktu diskrit.
3. Pengkodean : dalam proses pengkodean, setiap nilai diskrit terkuantisasi x_q (n) dikodekan dengan suatu barisan biner (10100 . . .)

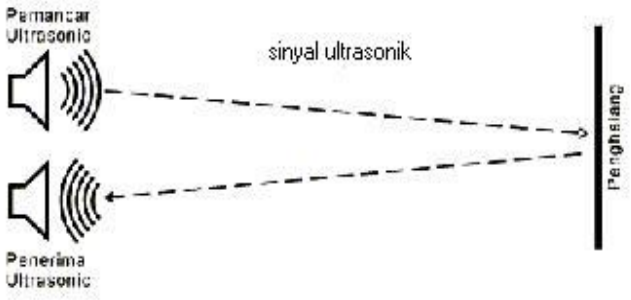


Gambar II.21 Pencuplikan dengan saklar

2.11 Sensor Jarak

Sensor jarak merupakan sensor yang dipakai untuk menghitung jarak antara modul sensor dengan bidang/benda yang ada dalam jangkauannya. Salah satu sensor jarak yang dipakai antara lain adalah sensor ultrasonik. Gelombang ultrasonik adalah gelombang dengan besar frekuensi diatas frekuensi gelombang suara yaitu lebih dari 20 KHz. Sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).

Prinsip kerja dari sensor ultrasonik dapat ditunjukkan dalam gambar dibawah ini :



Gambar II.22 Prinsip kerja sensor ultrasonik

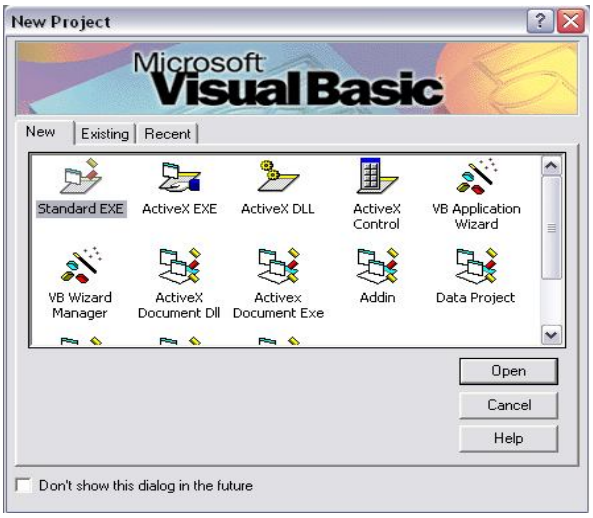
Prinsip kerja dari sensor ultrasonik adalah sebagai berikut :

1. Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20KHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40KHz. Sinyal tersebut di bangkitkan oleh rangkaian pemancar ultrasonik.
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal / gelombang bunyi dengan kecepatan bunyi yang berkisar 340 m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima Ultrasonik.
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya.

2.12 Pengenalan Visual Basic 6.0

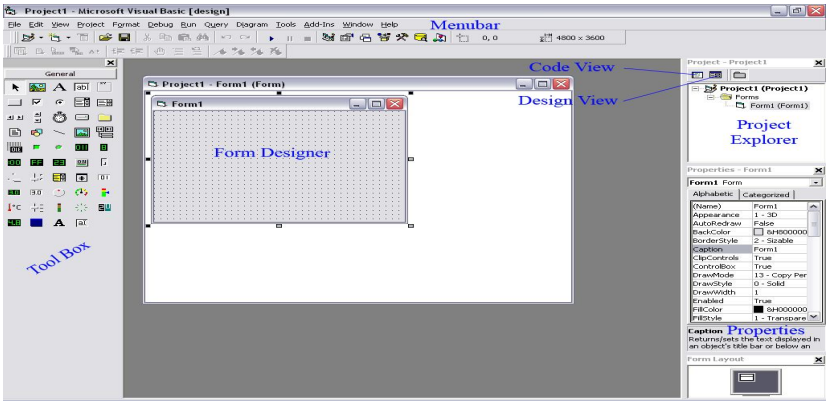
Visual Basic merupakan bahasa pemrograman komputer. Di dalam*Visual Basic 6.0* menyediakan sintak-sintak dalam pembuatan suatu aplikasi. Dengan menggunakan *Visual Basic 6.0* dapat menghasilkan berbagai macam jenis program. Aplikasi yang dibuat dapat diintegrasikan dengan *database*, *hardware* lain (*interface*) dan sebagainya.

Pada layar awal akan muncul tampilan di bawah ini.



Gambar II.23 Tampilan awal Visual Basic 6.0

Visual Basic 6.0 menyediakan banyak jenis modul aplikasi. Untuk memulai program standar pilihlah **Standar EXE**, kemudian klik *open*. Setelah itu akan muncul tampilan seperti berikut ini, yang menunjukan bagian-bagian dari IDE (*Integrated Development Environment*) yang akan digunakan.



Gambar II.24 Tampilan lembar kerja Visual Basic 6.0

2.12.1 Pengkodean pada Visual Basic 6.0

1. Tipe Data

Tipe data memiliki ciri – ciri tersendiri. Berikut bentuk dan ukuran dari tipe data:

Tabel II.4 Ukuran dan Tipe Data

Tipe Data	**Ukuran Storage**	**Jangkauan**
Byte	1 *Byte*	0 s/d 255
Boolean	2 *Byte*	*True* atau *False*
Integer	2 *Byte*	-32768 s/d 32767
Long	4 *Byte*	-2.147.483.648 s/d 2.147.483.647
Single	4 *Byte*	-3,40282e38 s/d -1,401296e-45 (-)
		1,401296e-45 s/d 3,402823e38 (+)
Double	8 *Byte*	-1,797691348623e308 s/d -4,9406564844127
Currency	8 *Byte*	-922.337.203.685.477,5808 s/d 922.337.203.685.477,5807
Decimal	14 *Byte*	7,92E+028
Date	8 *Byte*	1 Januari 100 s/d 31 desember 9999
Object	4 *Byte*	Mangacu pada objek tertentu
String	Panjang dari string	1 sampai ± 65400
Variant	16 *Byte*	Sembarang angka sampai jangkauan jenis *double* atau *string*

2. Variabel

Variabel digunakan untuk menampung nilai sementara di memori. Untuk membuat sebuah variabel terdapat ketentuan sebagai berikut :

a. Harus dimulai dengan suatu huruf
b. Tidak dapat mengandung titik atau spesial karakter
c. Tidak dapat lebih dari 255 huruf
d. Tidak dapat sama dengan *keyword* dari *visual basic*
e. Tidak membedakan huruf besar dan huruf kecil (*no case sensitive*)

Cara mendeklarasi variabel adalah sebagai berikut :

```
Dim [nama variabel] As [tipe data] atau
Public [nama variabel] As [tipe data] atau
Private [nama variabel] As [tipe data]
```

Public akan membuat suatu variabel dapat diakses dari segala tempat di dalam *project*, sedangkan *Dim* dan *Private* akan membuat suatu variabel yang hanya dapat diakses di dalam modul dimana variabel tersebut dideklarasikan.

2.12.2 Operator Pada Visual Basic

Visual basic menyediakan operator *aritmatika*, *komparasi* dan logika, salah satu hal yang harus dipahami adalah tata urut dari masing–masing operator, sehingga mampu membuat *ekspresi* yang akan menghasilkan nilai yang benar.

a. Operator *aritmatika*

Tabel II.5 Operator Aritmatika

Nama Operator	Tanda Operator
Pangkat	^

Negatif	-
Kali dan Bagi	* , /
Pembagian Bulat	\
Sisa Bagi	*Mod*
Tambah dan Kurang	+ , -
Penggabungan *string*	&

b. Operator Komparasi

Tabel II.6 Operator Komparasi

Nama Operator	**Tanda Operator**
Sama	=
Tidak Sama	<>
Kurang dari	<
Lebih dari	>
Kurang dari sama	<=
Lebih dari sama	>=
Like	*Like*

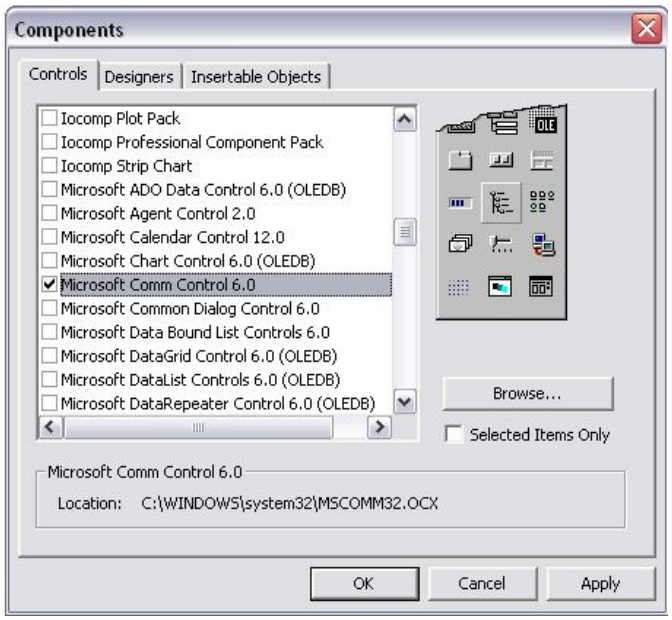
c. Operator Logika

Tabel II.7 Operator Logika

Nama Operator	Tanda Operator
Not	*Not*
And	*And*
Or	*Or*
Xor	*Xor*

2.12.3 Komunikasi serial *Visual Basic 6.0*

Visual Basic 6.0 menyediakan komponen *MS Comm Control 6.0*, sebagai media komunikasi. Untuk menambahkan komponen ini pada *Visual Basic*, pilih *Project ->Components.* Setelah itu akan muncul tampilan sebagai berikut:



Gambar II.25 Penambahan komponen pada Visual Basic 6.0

Untuk mengadakan suatu komunikasi serial antara 2 peralatan, kita harus melakukan beberapa langkah.

1. **Membuka Serial *Port***

Pada komunikasi serial, bit-bit data yang masuk dari dunia luar ke dalam komputer melalui *serial port* akan ditampung dulu di *receive buffer* sebelum akan dieksekusi oleh *main controller*. Demikian pula sebelum dikirimkan ke luar, data akan ditampung dulu di *transmit buffer*. Gambar skema lengkapnya dapat dilihat pada gambar di bawah ini. Sebelum membuka *serial port*, dilakukan pengaturan protokol komunikasi serial dengan *property* MSComm. Menentukan nomor *port* komunikasi menggunakan *CommPort* dan menentukan *baud rate*, *parity*, data *bits*, *stop bits* mengguankan *property setting*. Sedangkan untuk membuka *serial port* cukup menggunakan perintah *Port Open*. Sehingga kode program akan tertulis sebagai berikut

```
MSComm1.ComPort = 2
MSComm1.Settings = “9600,N,8,1”
MSComm1.PortOpen = True
```

2. Mengatur Serial *Device*

Pada tahap ini kita perlu memastikan bahwa pengaturan protokol komunikasi serial yang digunakan pada peralatan lain yang kita akses, sesuai dengan pengaturan pada komputer yang kita pakai.

3. *Setting Receive* dan *Transmit Buffer Properties*

Ada beberapa *property* dari *receive buffer* dan *transmit buffer* (*porperty* dari **MSComm**) yang perlu kita atur.

a. *InBufferSize* : mengatur ukuran *receive buffer*
b. *OutBuffer Size* : mengatur ukuran *transmit buffer*
c. *Rthreshold* : menentukan jumlah karakter yang diterima oleh *receive buffer* sebelum *OnComm event* dipicu

d. *Sthreshold* : menentukan jumlah karakter yang diterima oleh *transmit buffer* sebelum *OnComm event* dipicu. Nilai 0 berarti tidak pernah dipicu, sedangkan nilai 1 berarti dipicu setiap satu karakter.
e. *InputLen* : menentukan jumlah karakter yang dibaca CPU dari *receive buffer*
f. *InputMode* : menentukan tipe data *input* yang akan dibaca CPU. *Com Input Mode Text* untuk data *string*/teks dan *com Input Mode Binary* untuk data biner.

4. *Managing Receive* dan *Transmit Buffer*

Untuk menampilkan data dari peralatan lain ke dalam aplikasi VB, digunakan properti *Input*, sedangkan untuk mengirim data dari aplikasi VB ke peralatan lain digunakan poperti *Output*. Contoh struktur kode untuk *Input* dan *Output*.

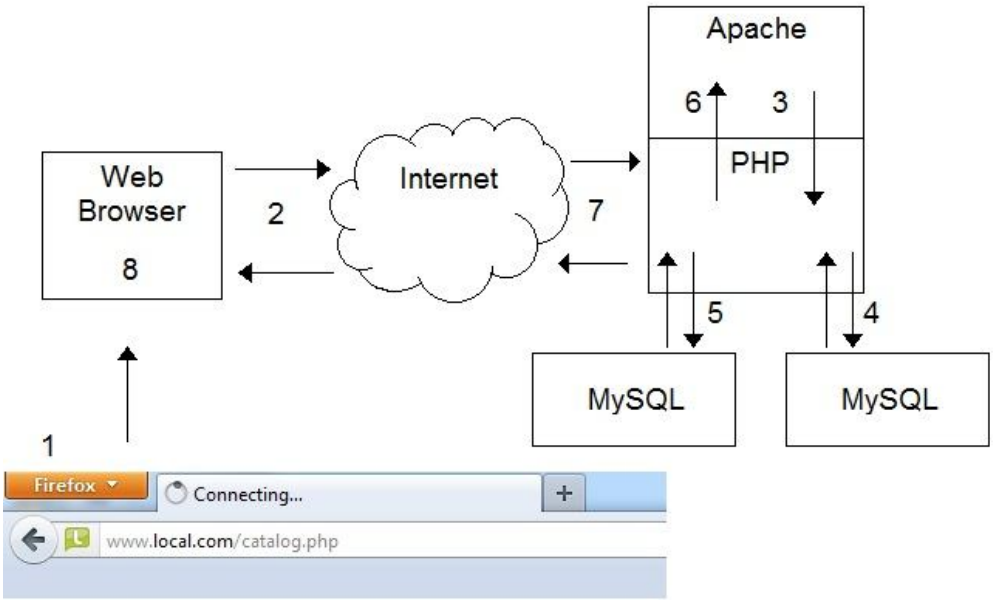
```
TxtDisplay.Text = MSComm1.Input   ( Contoh Input )

MSComm1.Output = "Data String"   ( Contoh Output )
```

2.13 Bahasa Pemrograman PHP

PHP Merupakan salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan di tampilkan kembali ke *web browser*, karena pemrosesan program PHP dilakukan di lingkungan *web server (Server-side)*, oleh sebab ituu seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah "*View Source"* pada *web browser* yang digunakan. [Budi Raharjo, Imam Heryanto, Enjang R.K (2010)."Modul Pemrograman Web (HTML,PHP,MySQL)"

Cara Kerja Aplikasi *Web* yang ditulis dengan PHP dapat di ilustrasikan dengan gambar dibawah ini.



Gambar II.26 Aplikasi Web Pada PHP

Keterangan dari gambar diatas:

1. User Menulis *www.local.com/catalog.php* kedalam *address bar* dari *web browser (IE, Mozilla Firefox, Opera*, dll*).*
2. *Web Browser* mengirimkan pesan di atas ke komputer *server* (*www.local.com*) melalui internet, meminta halaman *catalog.php*.
3. *Web Server* (Misalnya *Apache*), program yang berjalan di komputer *server*, akan menangkap pesan tersebut, lalu meminta *interpreter* PHP (program lain yang juga berjalan di komputer *server*) untuk mencari *file catalog.php* dalam *disk drive.*
4. *Interpreter* PHP membaca *file catalog.php* dari *disk drive.*
5. *Interpreter* PHP akan menjalankan perintah-perintah atau kode PHP yang ada dalam *file catalog.php*. Jika kode *file catalog.php* melibatkan akses terhadap *database* (misalnya *MySQL*) maka *interpreter* PHP juga akan berhubungan dengan *MySQL* untuk melaksanakan perintah-perintah yang berkaitan dengan *database*.
6. *Interpreter* PHP mengirimkan halaman dalam bentuk HTML ke *Apache*.
7. Melalui internet, *Apache* mengirimkan halaman yang diperoleh dari *interpreter* PHP ke komputer *user* sebagai respon atas permintaan yang diberikan.
8. *Web browser* dalam komputer user akan menampilkan halaman yang dikirimkan oleh *Apache.*

Berikut ini contoh program PHP :

```
<?

echo “Hello World!”;

?>
```

Keterangan: Perintah *echo* di dalam PHP berguna untuk mencetak nilai, baik teks maupun *numeric* ke dalam *web browser*. Setiap perintah atau statemen didalam kode PHP harus diakhiri dengan tanda titik koma (;).

2.14 Bahasa Pemrograman MySQL

MySQL merupakan sistem *database* yang banyak digunakan untuk pengembangan aplikasi *web*. Alasannya mungkin karena gratis, pengelolaan datanya sederhana, memiliki tingkat keamanan yang bagus, mudah diperoleh, dan lain-lain. [Budi Raharjo, Imam Heryanto, Enjang R.K (2010).”Modul Pemrograman Web (HTML,PHP,MySQL)”

Selain itu *MySQL* juga memiliki beberapa keistimewaan, antara lain :

1. *Portability*
   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
2. *Open Source*
   MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. *Multiuser*
   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance Tuning*
   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.
5. *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, text, date, timestamp,* dan lain-lain.

6. *Command* dan *Functions*
   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *Query.*
7. *Security*
   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask,* nama *host,* dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.
8. *Scalability* dan *Limits*
   MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. *Connectivity*
   MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).
10. *Localisation*
    MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.
11. *Interface*
    MySQL memiliki *interface* (antarmuka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API *(Application Programming Interface).*
12. *Clients* dan *Tools*

    MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database.*