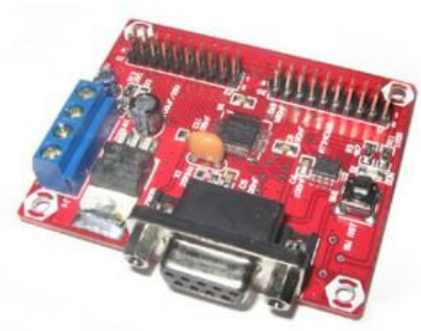
BAB II
LANDASAN TEORI

Sistem irigasi adalah salah satu hal terpenting dalam bidang pertanian dan perkebunan. Sistem irigasi yang baik akan menunjang hasil panen yang baik di bidang pertanian dan perkebunan. Kebanyakan sistem irigasi masih di kerjakan secara manual atau oleh manusia secara langsung, misalnya saja memonitoring ketinggian air dan pengendalian pintu air. Agar sistem irigasi dapat bekerja secara *efisien* dibutuhkan sebuah sistem untuk mengatur semua kegiatan di dalam sistem irigasi tersebut. Dalam bab ini dijelaskan mengenai teori-teori tentang perancangan sistem irigasi yang meminimalisir campur tangan manusia. Berikut ini teori perangkat keras dan perangkat lunak yang digunakan.

2.1 Perangkat Keras

2.2.1 Mikrokontroler Basic Stamp (BS2P40)



Gambar 2.1 Modul basic stamp (BS2P40)

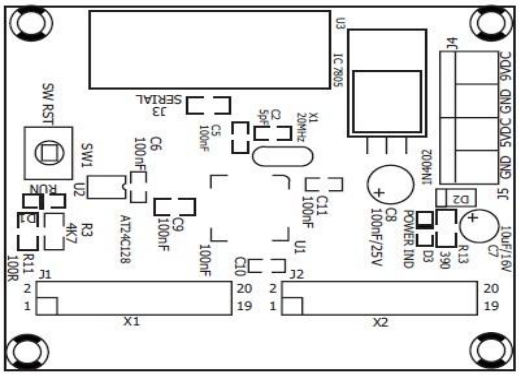
Mikrokontroler adalah sebuah IC yang berfungsi sebagai pengendali perangkat-perangkat yang terhubung dengan mikrokontroler tersebut. *Basic stamp* adalah mikrokontroler yang dikembangkan oleh *Parallax Inc* yang mudah

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.

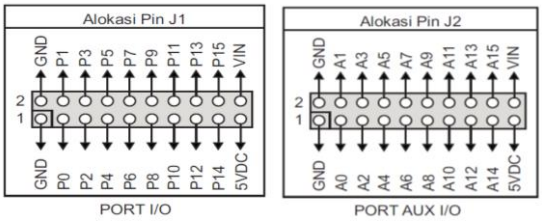
Pada mikrokontroler *basic stamp* ini memiliki spesifikasi sebagai berikut:[4]

1. Mikrokontroler *basic stamp* BS2P40 *Interpreter Chip* (PBASIC48W/P40)
2. 8 x 2K*byte* 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}.



Gambar 2.2 Tata Letak modul basic stamp (BS2P40)

Berikut ini adalah alokasi pin yang terdapat pada mikrokontroler *basic stamp* BS2P40.



Gambar 2.3 Alokasi PIN

Tabel 2.1. 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

2.2.2 YS-1100U Transceiver

Radio Frekuensi yang digunakan yaitu YS-1100U RF data *Transceiver* dirancang untuk mengirim dan menerima data dengan jarak pancar sampai 500 meter. Modul dapat langsung berhubungan dengan PC, perangkat RS232, dan lain-lain. [2]

Berikut ini adalah gambar dan alokasi pin yang terdapat pada radio YS-1100U.



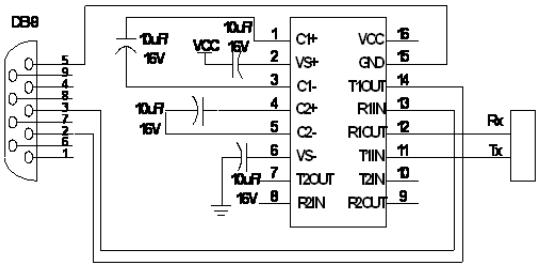
Gambar 2.4 YS-1100U

2.2.3 Radio Frekuensi YS-1100U RF mempunyai spesifikasi sebagai berikut :

1. *Power supplay* DC 3,3 - 5 V.
2. Ukuran 47mm X 24mm X 6mm (tanpa port antena).
3. Jarak pancar 300m – 500m. [2]

2.2.4 Rangkaian RS232

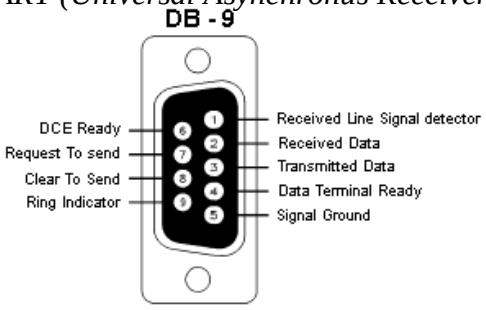
Rangkaian ini berfungsi untuk komunikasi antara alat dengan komputer. Komponen utama menggunakan IC MAX232 yaitu sebuah IC yang dapat mengubah format digital ke dalam sebuah format atau *level* RS232 dimana pada *level* RS232, tegangan *high* diwakili dengan tegangan +3 sampai +25 V. Diantara -3 dan +3 merupakan tegangan *invalid* atau tidak sah.



Gambar 2.5 Rangkaian RS232 dengan Rx,Tx

TxD dan RxD dihubungkan dengan modem pin Rx dan Tx sementara R1*IN* dan T1*OUT* dihubungkan dengan komputer melalui serial RS232. *Port* serial lebih sulit ditangani dari pada *port* paralel karena peralatan yang dihubungkan ke *port* serial harus berkomunikasi dengan menggunakan transmisi serial, sedangkan

data dikomputer diolah secara paralel. Sehingga, data dari *port* serial harus dikonversikan ke bentuk paralel untuk bisa digunakan secara *hardware* hal ini bisa digubakan oleh *UART (Universal Asynchronus Receiver Transmitter).*



Gambar 2.6 Konfigurasi port serial male

Tabel 2.2. Konfigurasi pin dan nama sinyal konektor serial DB9

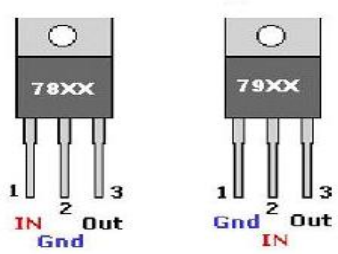
Nama Pin	Nama Sinyal	*Direction*	Keterangan
1	DCD	*In*	Data *Carrier Detect/Received Line Signal Detect*
2	RxD	*In*	*Received* Data
3	TxD	*Out*	*Transmite* Data
4	DTR	*Out*	Data Terminal *Ready*
5	GND	-	*Ground*
6	DSR	*In*	Data *Set Ready*
7	RTS	*Out*	*Request to Sent*
8	CTS	*In*	*Clear to Sent*
9	RI	*In*	*Ring Indicator*

2.2.5 Regulator

Rangkaian penyearah sudah cukup bagus jika tegangan *ripple*-nya kecil, namun ada masalah stabilitas. Jika tegangan PLN naik/turun, maka tegangan *output*nya 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 2.7 IC Regulator 78xx/79xx dan alokasi pin

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* regulatornya. 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.

2.2.6 Motor DC

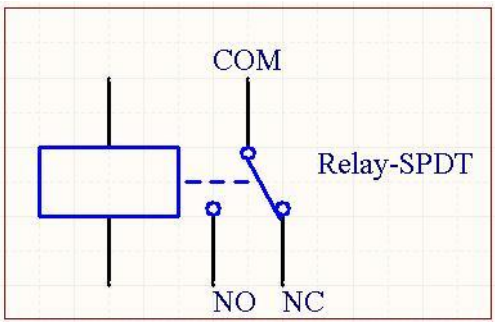
Motor DC merupakan peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik. Secara umum, kecepatan putaran poros motor DC akan meningkat seiring dengan meningkatnya tegangan

yang diberikan. Dengan demikian, putaran motor DC akan berbalik arah jika polaritas tegangan yang diberikan juga dirubah.

Motor DC tidak dapat dikendalikan langsung oleh mikrokontroler, karena kebutuhan arus yang besar sedangkan keluaran arus dari mikrokontroler sangat kecil. Motor *driver* merupakan alternatif yang dapat digunakan untuk menggerakkan motor DC.

2.2.7 Relay

Relay adalah komponen yang terdiri dari sebuah kumparan berinti besi yang akan menghasilkan elektromagnetik ketika kumparannya dialiri oleh arus listrik. Elektromagnetik ini kemudian menarik mekanisme kontak yang akan menghubungkan kontak *Normally Open (NO)* dan membuka kontak *Normally Closed (NC)*.



Gambar 2.8 Simbol dari Relay SPDT (Single Pole Dual Totem)

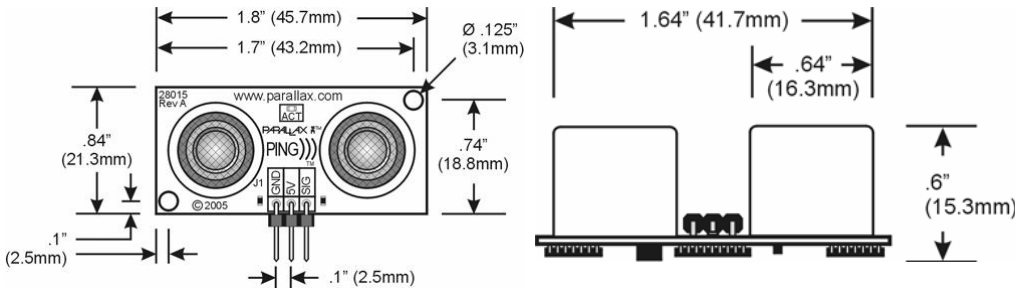
Gambar diatas adalah simbol dari relay SPDT *(Single Pole Dual Totem)* yang berarti memiliki sebuah kontak NO dan sebuah kontak NC dengan sebuah COMMON. Pada saat kumparan tidak dialiri arus, maka kontak NC akan terhubung dengan COM. Jika kumparan dialiri arus, maka kontak akan bergerak dari NC ke NO, sehingga NO akan terhubung dengan COM.

2.2.8 Sensor PING Ultrasonik

Ping adalah suatu modul yang berfungsi untuk mengukur jarak, dengan cara memancarkan sinyal *ultrasonic*. Dengan ukuran yang cukup kecil yaitu (2.1cm x

4.5cm) sensor ini dapat mengukur jarak antara 3cm sampai 300cm. keluaran dari Ping berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115µs sampai 18.5ms.

Pada dasarnya, Ping terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah pemancar sinyal *ultrasonic* dan sebuah penerima sinyal *ultrasonic*. Pemancar sinyal *ultrasonic* mengubah sinyal 40KHz menjadi sinyal, sedangkan penerima *sinyal ultrasonic* berfungsi untuk mendeteksi pantulan sinyalnya. Pada modul Ping terdapat tiga pin yang digunakan untuk jalur *power supply* (+5V), *ground* dan *signal.* Pin *signal* dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun.

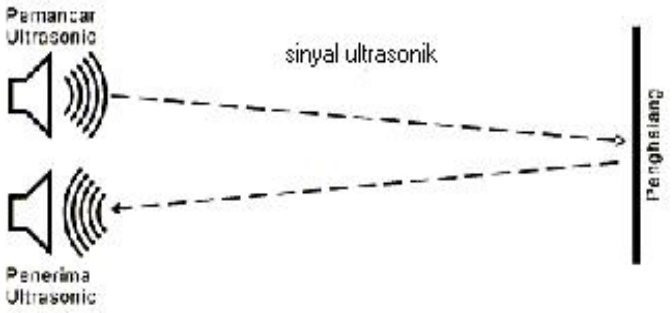


Gambar 2.9 Dimensi sensor Ping ultrasonic[3]

Sensor Ping *ultrasonic* ini memiliki spesivikasi sebagai berikut :[3]

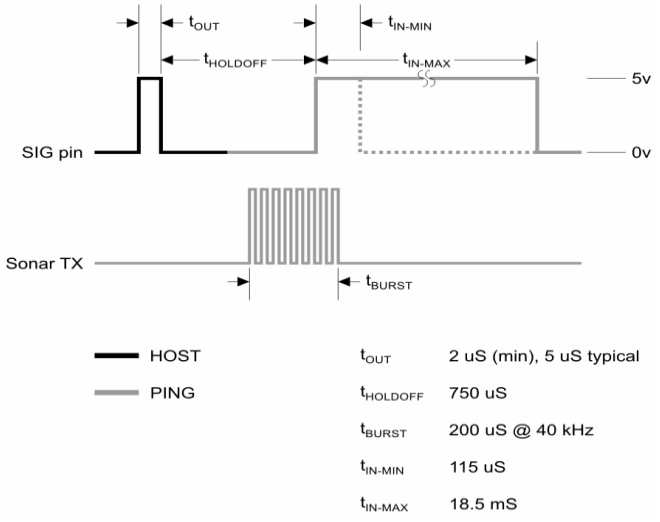
1. *Power supplay* DC +5 V.
2. *Supply Current* 30 mA typ; 35 mA max.
3. *Range* 2cm sampai 300cm.
4. *Burst Frequency* 40 kHz untuk 200 µs.

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



Gambar 2.10 Prinsip kerja sensor ultrasonic[3]

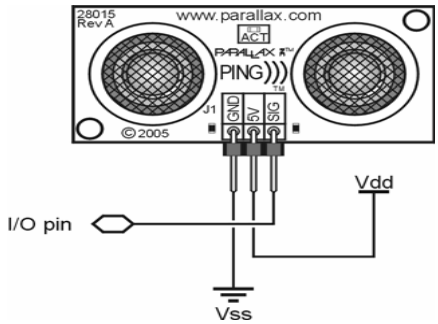
Ping bekerja dengan cara memancarkan sinyal ultrasonik kemudian menangkap pantulannya, ketika ada pulsa *trigger* dari mikrokontroler (pulsa high selama 5µs) sinyal ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama 200µs. sinyal tersebut akan merambat di udara dengan kecepatan 344.424m/s (1cm setiap 29.034µs). selama menunggu pantulan, Ping akan menghasilkan sebuah pulsa. Pulsa tersebut akan berhenti ketika sinyal pantulan tertangkap oleh Ping. Maka lebar pulsa dapat merepresentasikan jarak Ping dengan objek. Selanjutnya mikrokontroler cukup mengukur lebar pulsa tersebut dan mengkonversinya dalam bentuk jarak.



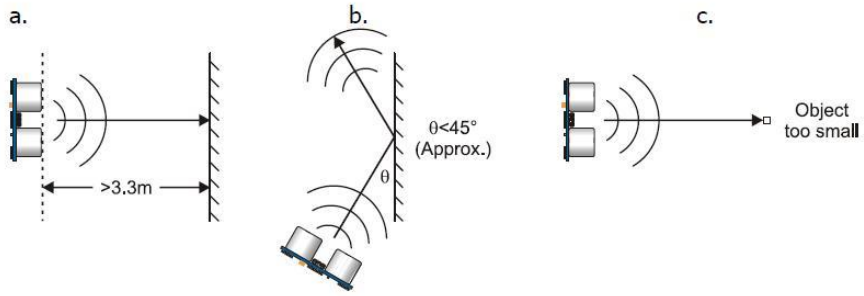
Gambar 2.11 Cara kerja sensor Ping ultrasonic

Dari gambar di atas, cara kerja sensor Ping ultrasonik adalah:

Kondisi awal SIG pin adalah logika 0, akan memberikan logika 1 selama 2µs-5µs (*t-out*) untuk mengaktifkan Ping dan mulai pengukuran. Tunggu hingga 750µs (*t-holdoff*) yaitu saat Ping mulai memancarkan sinyal ultrasonic. Kemudian merubah SIG pin sebagai input yang akan membaca jarak dengan cara membuat SIG pin berlogika 1, apabila SIG pin berubah kondisi dari logika 1 menjadi logika 0 maka itulah jarak yang terukur. Jika tidak terdapat dinding atau penghalang maka waktu bernilai 18.5ms yang berarti jarak yang terukur adalah 300cm.



Gambar 2.12 Konfigurasi pin Sensor Ping Ultrasonik.[3]



Gambar 2.13 Objek posisi benda.[3]

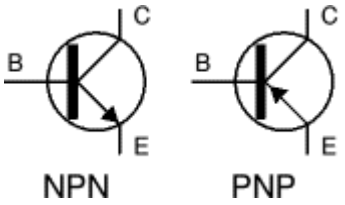
Sensor PING tidak dapat secara akurat mengujur jarak dengan objek pantul jika:

a. Objek lebih dari 3 meter.
b. Objek memiliki permukaan reflektif, sehingga suara tidak dipantulkan kembali menuju sensor.
c. Objek terlalu kecil.

Material objek yang tidak dapat memantulkan suara ultrasonic adalah objek yang menyerap suara, memiliki permukaan yang lembut atau tidak teratur seperti boneka hewan. Namun PING ultrasonic dapat mendeteksi permukaan air, namun tidak disarankan untuk penggunaan *outdoor* atau penggunaan terus menerus dilingkungan yang basah.[3]

2.2.9 Transistor Sebagai Saklar

Transistor adalah komponen elektronika yang tersusun dari bahan semi konduktor yang memiliki tiga kaki yaitu basis (B), kolektor (C), dan emiter (E). Berdasarkan susunan semikonduktor yang membentuknya, transistor dibedakan menjadi dua tipe, yaitu transistor PNP dan NPN.



Gambar 2.14 Simbol transistor

Transistor dapat difungsikan sebagai sakelar elektronika dengan memanfaatkan dua keadaan yaitu, keadaan saturasi (sebagai saklar tertutup) dan keadaan *cut off* (sebagai saklar terbuka).

Pada saat saturasi maka arus kolektor adalah

$$Ic(sat) = \frac{\text{VcC}}{\text{Rc}} \quad \text{..... (1)}$$

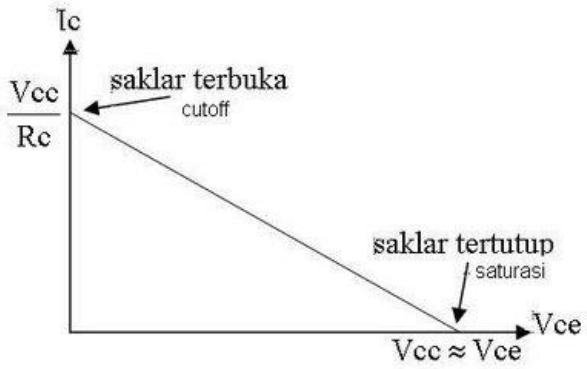
Pada saat *cut off* tegangan kolektor emiter sama dengan tegangan sumber, kolektor dan arus basis mendekati nol.

$$\text{VCE(cut)} = \text{Vcc}$$

$$\text{IB} \approx 0 \quad \text{..... (2)}$$

Untuk mencari arus basis pada keadaan resistor basis terpasang dapat dihitung dengan persamaan

$$IB = \frac{\text{VBB} - \text{VBE}}{\text{RB}} \quad \text{..... (3)}$$

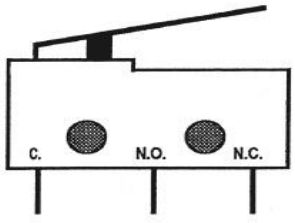


Gambar 2.15 Kurva daerah operasi transistor sebagai saklar

Daerah saturasi adalah mulai dari VCE = 0 Volt sampai ± 0.7 Volt (transistor silicon), yaitu akibat dari efek diode kolektor-basis yang mana tegangan VCE belum mencukupi untuk dapat menyebabkan aliran arus. Sedangkan daerah *cut off* adalah ketika tegangan VCC dinaikan perlahan-lahan, sampai tegangan VCE tertentu yang menyebabkan IC mulai konstan.

Saturasi adalah kondisi dimana kaki basis (B) pada transistor dialiri arus pemicu dan akan menghantarkan arus dari kolektor (C) ke emiter (E). sedangkan *cut off* adalah kondisi dimana kaki basis tidak mendapatkan arus pemicu dan tidak menghantarkan arus dari kolektor (C) ke emiter (E).

2.2.10 Micro Switch



Gambar 2.16 Konfigurasi pin micro switch

Istilah mikro pada *micro switch* atau sakelar mikro, tidak berarti bahwa sakelar ini berukuran kecil. Nama ini mengindikasikan bahwa tombol yang digunakan untuk mengoperasikan *micro switch* hanya bergeser dengan jarak

perpindahan yang sangat kecil. Sakelar jenis ini sangat sensitif, sedikit tekanan saja pada tuas dapat mengakibatkan sakelar berpindah dari satu posisi ke posisi lainnya.

Micro switch umumnya memiliki tiga buah kaki atau tag terminal, yaitu **Common** atau jalur bersama, **NO** *(Normally Open)* atau kontak normal terbuka, dan **NC** *(Normally Closed)* atau kontak normal tertutup. Kontak-kontak tersebut dilengkapi dengan pegas, dalam keadaan normal, kontak jalur bersama akan terhubung ke kontak normal tertutup.

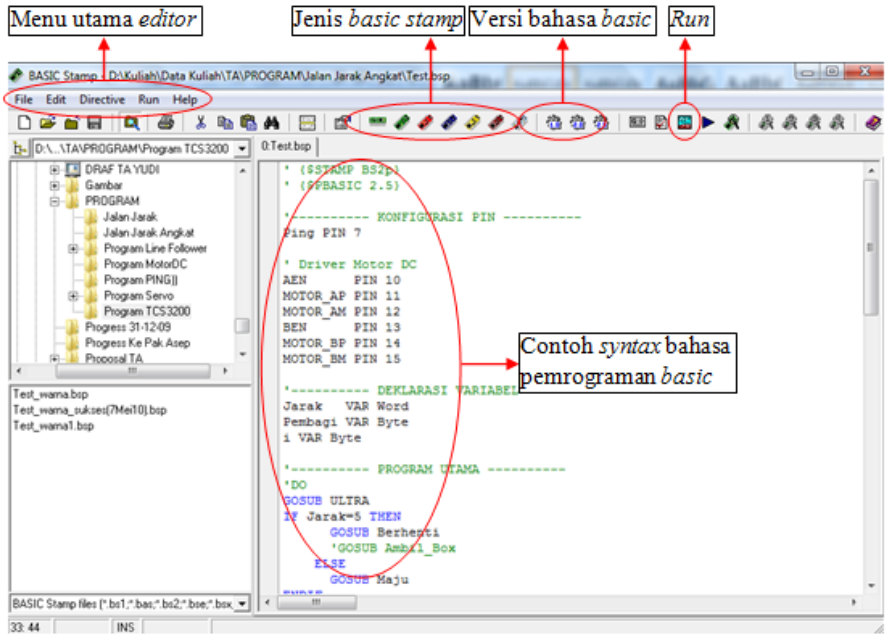
2.2 Perangkat Lunak

2.2.1 Pengenalan Basic Stam Editor

Mikrokontroler *basic stamp* (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, meng*kompile* dan men*download*nya 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 2.3. 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	Men*set* pin I/O menjadi 1
LOW	Men*set* pin I/O menjadi 0
PWM	Konversi suatu nilai digital ke keluaran analog lewat *pulse width modulation*



Gambar 2.17 Tampilan basic stamp editor

2.2.5.1 Program 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 2.18 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* atau *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 Sintak Program

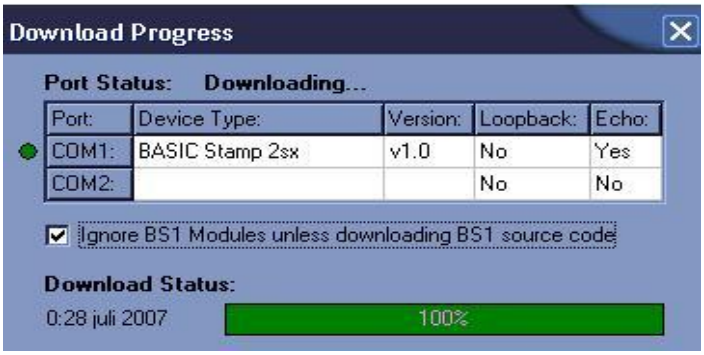
Memeriksa sintak program dilakukan untuk memastikan semua sintak sudah benar. Untuk memeriksa sintak ini bisa pilih menu *RUN*, Cek Sintak atau kombinasi tombol CTRL+T. Tampilan jika listing program yang kita buat sudah benar.



Gambar 2.19 Hasil pemeriksaan sintak yang sukses

4. Menjalankan Program

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

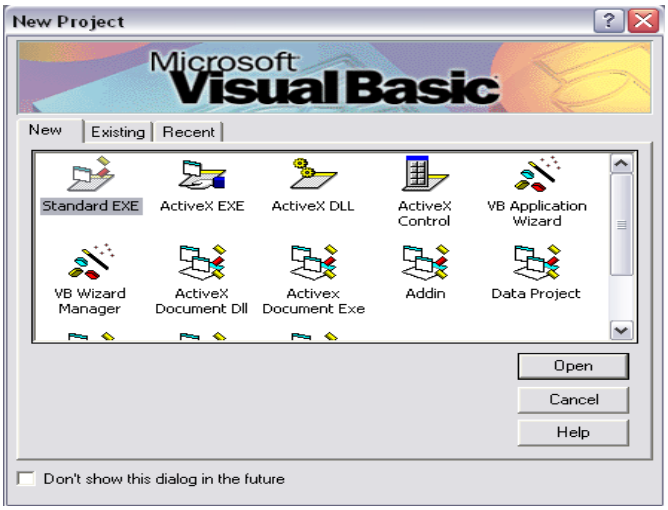


Gambar 2.20 Tampilan jika pendownloadan program sukses

Pengenalan dalam*Visual Basic 6.0*

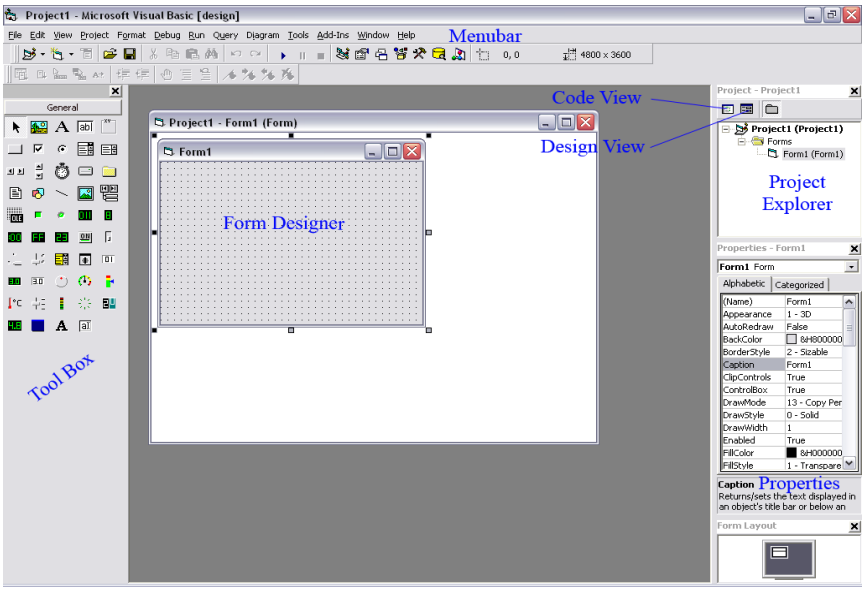
Visual Basic (atau sering disingkat VB) adalah perangkat lunak untuk menyusun program aplikasi yang bekerja dalam lingkungan sistem operasi windows. [6] Di dalam *Visual Basic 6.0* disediakan sintak-sintak dalam pembuatan suatu aplikasi. Dengan menggunakan *Visual Basic 6.0* dapat menghasilkan berbagai 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 2.21 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 2.22 Tampilan lembar kerja Visual Basic 6.0

Pengkodean *Visual Basic 6.0*

1. Tipe Data

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

Tabel 2.4. Ukuran dari 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	Panjangdari 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]

Gambar 2.23 Cara mendeklarasi variabel

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.

3. Operator

Operator digunakan pada Visual Basic untuk memanipulasi data maupun untuk melakukan perhitungan. Operator dalam visual besic dapat dikelompokan menjadi tiga kelompok yaitu operator *aritmatika*, *komparasi* dan logika.[6] 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 2.5. 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 2.6. Operator Komparasi

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

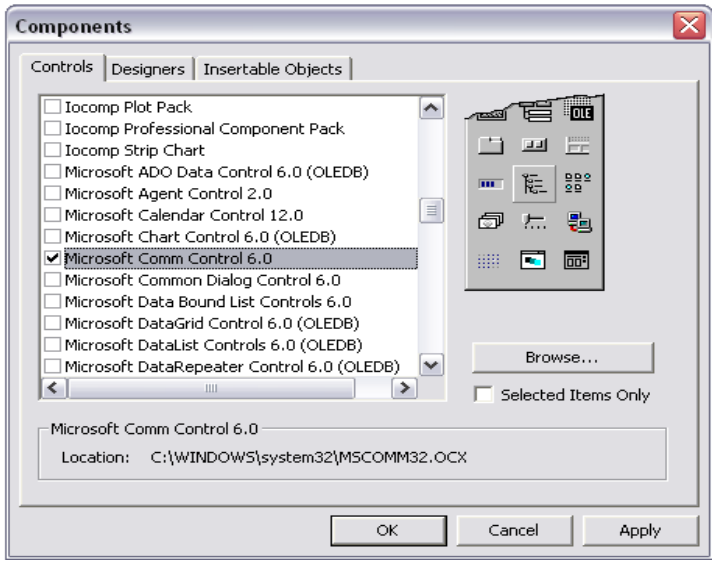
c. Operator Logika

Tabel 2.7. Operator Logika

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

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 2.24 Penambahan komponen pada Visual Basic 6.0

Untuk mengadakan suatu komunikasi serial antara dua 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*. 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* menggunakan *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

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

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

peralatan lain digunakan *poperty Output*. Contoh struktur kode untuk *Input* dan *Output*.