BAB II
LANDASAN TEORI

2.1. Perkembangan *Flying Robot*

Perkembangan industri penerbangan sekarang ini sudah semakin maju. Secara umum *flying robot* yang dikenal dalam industri penerbangan yaitu sejenis pesawat terbang dan helikopter, yang sistem navigasi pengontrolan serta pengendaliannya diatur oleh mesin serta perangkat-perangkat lainnya. Selain itu juga arah navigasinya diatur oleh sumber daya manusia. Namun seiring dengan waktu sekarang mulai berkembang *flying robot* yang sistem pengendalian serta navigasinya diatur dari jarak jauh, atau dengan kata lain sumber daya manusia dapat mengendalikan *flying robot* tersebut dari jarak jauh tanpa harus mengendalikannya di dalam *flying robot* tersebut. Dengan adanya *flying robot* dengan sistem ini diharapkan akan membantu pekerjaan manusia menjadi lebih aman dan tidak akan membahayakan dirinya. Jenis *flying robot* seperti ini banyak digunakan untuk peperangan, karena tingkat keamanan dan keselamatannya lebih terjamin.

Ada banyak jenis *flying robot* sesuai dengan kontruksi dan pemanfaatannya, diantaranya :

1. Helikopter Otonom yang dibuat untuk digunakan sebagai pembelajaran para pilot. Helikopter yang dikendalikan dengan jarak jauh ini sangat menunjang bagi pembelajaran pilot yang masih baru.[3]



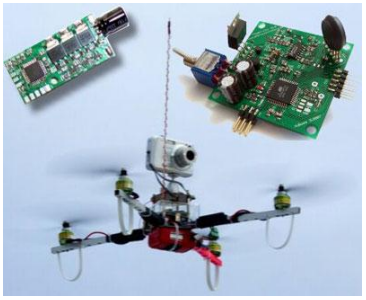
Gambar 2.1 Helikopter Otonom

2. *Flying Robot* Otonom dengan empat *Rotor* yang dapat menghemat energi. Tujuan dari proyek *flying robot* otonom berukuran kecil dengan empat *rotor* ini adalah dapat bermanuver terbang baik, digunakan di *indoor* atau *outdoor* di bawah kondisi apapun.[4]



Gambar 2.2 *Flying Robot* Otonom dengan empat *Rotor*

3. *Flying Robot* yang dipasang kamera. Digunakan sebagai pengintai jarak jauh dan dikendalikan olek PC serta menggunakan mikrokontroler.



Gambar 2.3 *Flying Robot* Otonom dengan kamera

2.2. Perangkat Keras (*Hardware*)

2.2.1. Mikrokontroler *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 di-*download* melalui *port* serial dengan menggunakan konverter *USB to Serial* untuk komputer yang tidak memiliki *port* serial, serta membutuhkan *power supply* saat men*download* 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.

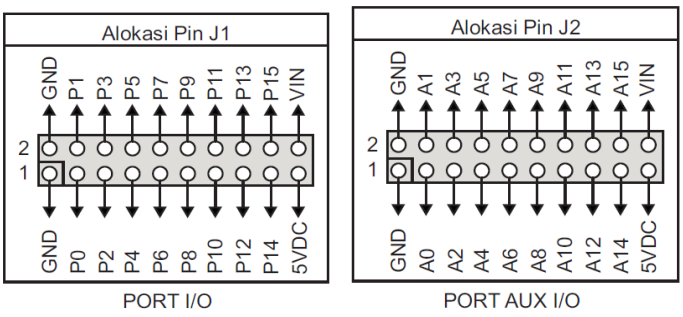


Gambar 2.4 Modul *basic stamp* (BS2P40)

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

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

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



Gambar 2.5 Alokasi PIN *basic stamp*

Diskripsi pin *Basic stamp*BS2P40

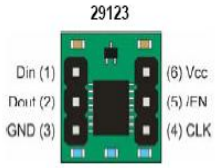
Tabel 2.1 Diskripsi Pin BS2P40

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. Kompas Digital Hitachi HM55B

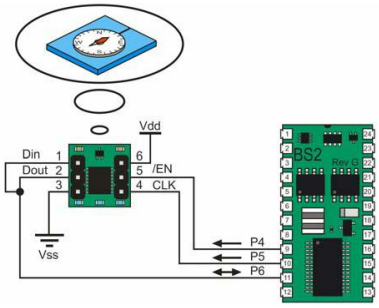
Sensor kompas berfungsi sebagai penunjuk arah dari gerak *flying robot*, sensor yang digunakan adalah sensor kompas digital HM55B. Modul kompas digital HM55B diproduksi oleh *Parallax*, yaitu sensor dual-sumbu magnetik. Sensor ini memiliki keluaran digital sebanyak 2 *axis* yaitu *axis* Y dan *axis* X.

Memiliki resolusi sampai 6 bit data dengan kecepatan sensivitas pengukuran antara 30 – 40 ms setelah program dijalankan dan dapat bekerja pada tegangan pada 5V. Konfigurasi pin sensor kompas HM55B.[7].



Gambar 2.6 Kompas Digital HM55B

Cara kerja sensor kompas digital HM55B



Gambar 2.7 koneksi kompas digital dengan modul *basic stamp* (BS2P40)

Kompas-kompas digital yang ada di pasaran banyak macamnya. Di antaranya yaitu HM55B yang cukup sensitif untuk mendeteksi medan magnet bumi. Kompas digital ini cukup di*supply* tegangan sebesar 5 Vdc dengan konsumsi arus 15mA. Pada CMPS03, arah mata angin dibagi dalam bentuk derajat yaitu : Utara (0^o), Timur (90^o), Selatan (180^o) dan Barat (270^o).

Ada dua cara untuk menperoleh informasi arah dari kompas digital ini yaitu dengan membaca sinyal PWM *(Pulse Width Modulation)* atau dengan membaca data *interface* I2C. Sinyal PWM adalah sebuah sinyal yang telah dimodulasi lebar pulsanya. Pada HM55B, lebar pulsa positif merepresentasikan sudut arah. Lebar pulsa bervariasi antara 1mS (00) sampai 36.99mS (359.90). Dengan kata lain lebar pulsa berubah sebesar 100uS setiap derajatnya. Sinyal akan *low* selama 65mS diantara pulsa, sehingga total periodanya adalah 65mS + lebar pulsa positif (antara 66mS sampai 102mS). Pulsa tersebut dihasilkan oleh *timer* 16 bit di dalam prosesornya, yang memberikan resolusi 1uS.

2.2.3. YS 1020K *Transciver*

Radio Frekuensi yang digunakan yaitu YS-1020K *Transceiver* dirancang untuk data sistem transmisi *UART* dalam jangkauan pendek. Modul dapat langsung berhubungan dengan PC, perangkat RS232, dan lain-lain.



Gambar 2.8 YS-1020K

Tabel 2.2 Susunan pin radio YS-1020K

PIN	NAMA PIN	FUNGSI	LEVEL
1	GND	*Ground*	
2	VCC	Tegangan *input*	+7,5V
3	RXD/TTL	*Input* serial data	TTL
4	TXD/TTL	*Output* serial data	TTL
5	DGND	Digital *Ground*ing	
6	A(TXD)	A of RS-485 or TXD of RS-232	
7	B(RXD)	B of RS-485 or RXD of RS-232	
8	*SLEEP*	*Sleep control* (*input*)	TTL
9	*TEST*	*EX-Factory testing*	

Modem radio ini mempunyai spesifikasi sebagai berikut :

1. Mempunyai 8 *channel* untuk pengiriman/penerimaan data
2. Tipe modulasi yang dipakai adalah GFSK (*Gaussian Frequensy Shift Keying*). Menggunakan *Gaussian filter* untuk memperhalus penyimpangan frekuensi yang terjadi.

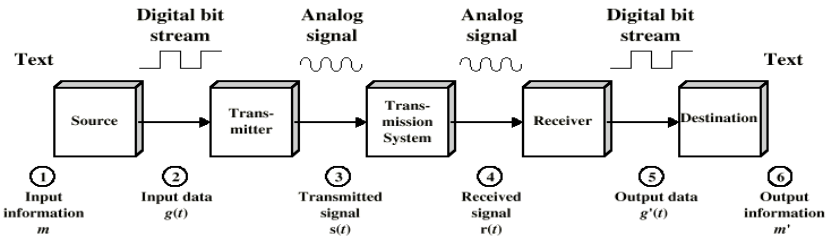
3. Dapat menggunakan *level* TTL (*Transistor-Transistor Logic*).
4. Integrasi antara *receiver* dan *transmiter* memerlukan waktu 10 ms antara pengiriman dan penerima.
5. Saat penerimaan data (*received*) arus yang diperlukan < 25 mA, sedangkan saat pengiriman data (*transmitted*) arus yang diperlukan < 350 mA.
6. Power Supply yang dibutuhkan yaitu 7,5V.[8].

2.2.3.1. Komunikasi data

Komunikasi data adalah bagian dari komunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer-komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berupa informasi yang disajikan oleh isyarat digital.

1. Komponen komunikasi data

a. Data / Sumber (*Source*)
b. Pengirim (*Transmiter*)
c. Media Transmisi (*Transmission media*)
d. Penerima (*Receiver*)
e. Tujuan (*Destination*)



Gambar 2.9 Blok diagram model komunikasi sederhana

Pada diagram model komunikasi data sederhana dapat dijelaskan :

1. **Sumber *(source)*** : dapat membangkitkan data sehingga dapat ditransmisikan.
2. **Pengiriman *(Transmiter*)** : pada bagian ini data yang dibangkitkan dari sistem, sumber tidak ditransmisikan secara langsung dalam bentuk aslinya

namun pada sebuah *transmiter* cukup memindahkan informasi dengan menghasilkan sinyal elektromagnetik yang dapat ditransmisikan dengan beberapa sistem transmisi berurutan.

3. **Media Transmisi *(transmission media)* :** merupakan jalur transmisi yang menghubungkan antara sumber dan tujuan.
4. **Penerima *(Receiver)*** : pada bagian ini sinyal dari pengirim diterima dari sistem transmisi dan memindahkan bentuk sinyal elektromagnetik menjadi digital yang dapat ditangkap oeh tujuan.
5. **Tujuan *(destination)*** : alat ini menerima data yang dihasilkan oleh penerima.[6]

2. Metode Transmisi

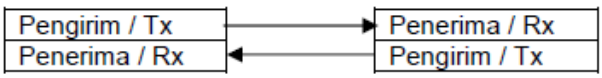
Ada tiga macam metode transmisi data, tiga metode tersebut adalah :

1. *Simplex*
   a. Sinyal ditransmisi dalam satu arah
   b. Stasiun yang satu bertindak sebagai *transmiter* dan yang lain sebagai *receiver*, tugasnya adalah tetap.
   c. jarang digunakan untuk sistem komunikasi data



2. *Half Duplex*

   a. Sinyal ditransmisikan ke dua arah secara bergantian
   b. Kedua stasiun dapat melakukan transmisi tetapi hanya sekali dalam suatu waktu.
   c. Terdapat "*turn around time*" (waktu untuk mengubah arah)



3. *Full Duplex*

   a. Sinyal ditransmisikan ke dua arah secara simultan / bersamaan
   b. Dua arah pada waktu yang sama.



3. Gangggguan transmisi

Gangguan yang ada pada transmisi data yaitu :

1. *Atenuasi* dan *distorsi atenuasi*

   Kekuatan sinyal berkurang bila jaraknya terlalu jauh melalui media transmisi, sehingga mengurangi tingkat kejelasan.

2. *Distorsi* tunda

   Kejadian ini disebabkan oleh sebuah sinyal yang melewati *guided* berbeda frekuensi.

3. *Derau*

   Adalah sinyal – sinyal yang tidak diinginkan yang terselip atau terbangkitkan dari suatu tempat diantara transmisi dan penerima. [6].

4. Sistem komunikasi radio untuk transmisi digital

Teknik modulasi sinyal analog :

a. *Amplitudo modulation (AM)*

   Amplitudo modulation (AM) merupakan proses modulasi yang mengubah *amplitudo* sinyal pembawa sesuai dengan sinyal pemodulasian atau sinyal informasinya. Sehingga dalam kondisi modulasi AM. Frekuensi dan *fasa* yang dimiliki sinyal pembawa tetap, tetapi *amplitudo* sinyal pembawa berubah sesuai dengan informasi.

b. *Frequency modulation (FM)*

   Frequency Modulation (FM) proses modulasi yaitu sinyal informasi ditumpangkan ke sinyal *carrier* atau sinyal pembawa, modulasi frekuensi merupakan suatu proses modulasi dengan cara mengubah frekuensi gelombang pembawa *sinusoidal*, yaitu dengan cara menyelipkan sinyal informasi pada gelombang pembawa tersebut.[6]

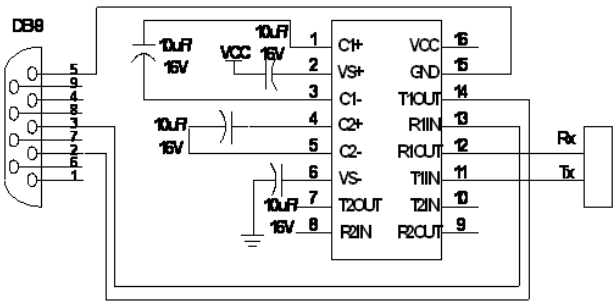
5. Modulator – Demodulator FSK

Alat untuk melakukan modulasi dan demodulasi disebut modem (*modulator–demodulator). Modulator* adalah bagian yang berfungsi untuk menggabungkan sunyal informasi dengan sinyal pembawa *(carrier)* dimana sinyal

hasil penggabungan inilah yang akan dikirim oleh pemancar *(transmitter)* . Sedangkan *demodulator* kebalikan dari *modulator* yaitu bagian yang berfungsi untuk memisahkan sinyal informasi dengan sinyal pembawa *(carrier)* dan dilakukan pada penerima *(receiver).* Alat yang menggunakan *port* serial untuk berkomunikasi dibagi menjadi 2 kategori, yaitu DTE *(Data Terminal Equiment)* dan *(Data Communication Equipment),* modem adalah perangkat DCE, perangkat yang berhubungan langsung dengan medium transmisi, sedangkan perangkat DTE contohnya adalah terminal atau komputer. Modem FSK umumnya memiliki kecepatan 300 bps sampai 1200 bps dan sering digunakan untuk komunikasi data antar komputer dan PSTN yang memiliki rangkaian *switching* yang sederhana dan memiliki *bandwidth* yang rendah.

2.2.4. Rangkaian RS232

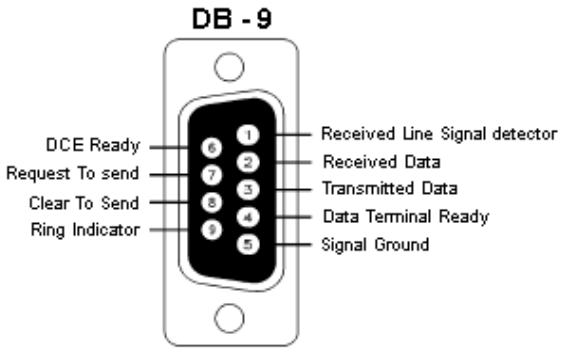
Rangkaian ini berfungsi untuk komunikasi antara alat dengan komputer. Komponen utama menggunakan IC MAX232 yaitu sebuah IC yang dapat mengubah format digital ke dalam sebuah format atau *level* RS232 dimana pada *level* RS232, tegangan *high* diwakili dengan tegangan +3 sampai +25 V. Diantara -3 dan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



Gambar 2.10 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 2.11 Konfigurasi port serial *male*

Tabel 2.3. 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. Motor *Brushless*

Pada motor *brushless* terdapat enam bagian penting yaitu, *Rotor, Commutator, Brushless, Axle (sumbu), field magnet (medan magnet),* dan *dc power supply (arus dc)*. Motor menggunakan magnet untuk menghasilkan gerakan (putaran). Di dalam sebuah elektrik motor kondisi saling tarik menarik dan tolak menolak pada medan magnet ini akan menghasilakan gerakan berputar atau sering

disebut *rotational motion. Rotor* adalah sebuah elektromagnet (magnet yang dihasilkan dari arus listrik), sedangkan sebagai medan magnet digunakan magnet yang permanen, jika arus DC mengalir, maka *rotor* akan berputar 180 derajat karena perbedaan kutub antara elektromagnet dan permanen magnet. Untuk membuat agar *rotor* dapat berputar maka kutub di elektromagnet perlu diubah, hal ini akan dilakukan oleh *brushless*. Bagian *brushless* ini berputar pada *rotor* dan mengubah polaritas magnet pada elektromagnet pada saat berputar.

Beberapa kelamahan dari *Brushless*:

1. *Brushless* lama kelamaan akan menjadi rusak
2. *Brushless* membatasi kecepatan maksimum dari motor.
3. Karena posisi elektromegnet ada di tengah-tengah *(rotor)* maka pendinginan motor menjadi lebih sulit.

Beberapa keuntungan dari *brushless*:

1. Jumlah elektromagnet *distator* dapat sebanyak mungkin untuk mendapatkan kontrol yang lebih akurat.
2. Komputer juga dapat mengatur kecepatan motor lebih baik sehingga membuat *brushless* lebih efisien.



Gambar 2.12 Motor *Brushless*

2.2.6. *Electronic speed controller (ESC)*

ESC adalah sebuah rangkaian elektronik yang fungsinya mengatur putaran pada motor sesuai *ampere* yang dibutuhkan oleh motor, selain itu juga berfungsi untuk menaikan jumlah arus yang diperlukan oleh motor. Kecepatan untuk motor yang keluar dari ESC diatur melalui pulsa dari mikrokontroler. ESC bekerja dipengaruhi dua faktor:

1. Kuat arus *(ampere)* diberikan pada motor untuk mengontrol *speed*. *Ampere*ESC harus lebih besar dari pada motor, minimal A.ESC=A.Motor. Misalnya, motor mampu menyedot arus maksimal 30 *ampere*, ESC harus minimal 30 *ampere* atau lebih besar dari 30 *ampere*. Kalau *ampere* ESC lebih kecil dari motor maka daya kerja ESC akan semakin besar untuk men*supply* arus ke motor dan bisa mengakibatkan ESC cepat panas dan terbakar.
2. Di pengaruhi oleh bobot *flying robot*

   Jika beban *flying robot* semakin berat sebaiknya *ampere* ESC diberikan nilai yang besar, ini sangat mempengaruhi saat mengangkat beban *flying robot*.



Gambar 2.13 *Electronic Speed Controller* (ESC)

2.2.7. *Propeller* (baling-baling)

Propeller atau baling-baling digunakan pada *flying robot* sebagai alat gerak, biasanya dipasang di atas *flying robot* dan di ekornya. *Propeller* ini akan menghasilkan angin dari putarannya, sehingga membantu motor untuk bisa terbang. Semakin kencang putarannya maka semakin cepat *flying robot* terbang ke arah yang ditentukan.



Gambar 2.14 Jenis - jenis *Propeller*

2.2.8. Catu Daya

Catu daya yang digunakan adalah baterai LIPO. Baterai ini memiliki daya sebesar 11,1 V dan arus sebesar 1800 mA. Baterai LIPO ini memiliki ketahanan pemakaian selama 10 menit, tidak boleh terpasang dengan motor maupun

komponen lain selama lebih dari 10 jam dan harus segera di*carger* sebelum batas minimal dari *voltase* baterai itu sendiri.

Berikut ini gambar dari baterai yang digunakan.

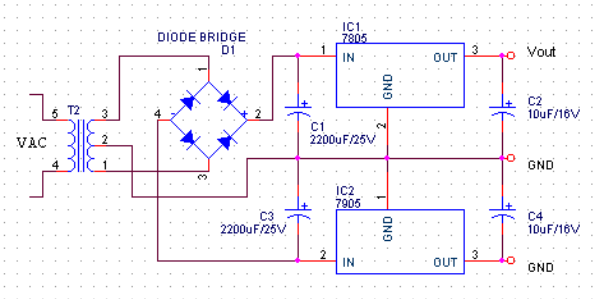


Gambar 2.15 Baterai LIPO *(Lithium Polimer)*

2.2.9. 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 2.16 Regulator Dengan IC 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 12 *volt*.

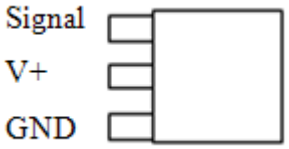
Selain dari regulator tegangan tetap ada juga IC regulator yang tegangannya dapat diatur. Prinsipnya sama dengan regulator OP-amp yang dikemas dalam satu IC misalnya LM317 untuk regulator *variable* positif dan LM337 untuk regulator *variable* negatif. Bedanya resistor R1 dan R2 ada di luar IC, sehingga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

2.2.10. Motor *Servo Mini Standard.*

Berbeda dengan motor DC dan motor *Stepper*, motor *servo* adalah sebuah motor dengan sistem *closed feedback* dimana posisi dari motor akan di informasikan kembali ke rangkaian kontrol yang ada di dalam motor *servo*.
Motor ini terdiri dari sebuah motor, serangkaian *gear*, *potensiometer* dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran *servo*. Sedangkan sudut dari sumbu motor *servo* diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Motor *servo* biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor *stepper*. Walau demikian, untuk beberapa keperluan tertentu, motor *servo* dapat dimodifikasi agar bergerak kontinyu.



Gambar 2.17 Pin *Out* kabel Motor *Servo*

Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagian-bagian lain yang mempunyai gerakan terbatas dan membutuhkan *torsi* cukup besar.



Gambar 2.18 Motor *Servo Mini standard*

2.2.11. Motor DC



Gambar 2.19 Motor DC

Motor DC merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

1. Tegangan dynamo → meningkatkan tegangan dinamo akan meningkatkan kecepatan.
2. Arus medan → menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan *rolling mills*, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya. Motor DC juga relatif mahal dibanding motor AC.

Mekanisme Kerja Motor DC

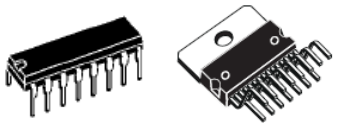
Mekanisme kerja untuk seluruh jenis motor secara umum sama

- Arus listrik dalam medan magnet akan memberikan gaya
- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran/*loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar/ *torque* untuk memutar kumparan.
- Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan. [9]

2.2.12. Driver motor L298

L293 dan L298 adalah contoh IC yang dapat digunakan sebagai *driver* motor DC. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. L293 dapat mengontrol 4 buah motor DC sedangkan L298 dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 VDC dan arus mencapai 2 A untuk setiap kanalnya.

Pengaturan kecepatan kedua motor dilakukan dengan cara pengontrolan lama pulsa aktif (metode PWM - *Pulse Width Modulation*) yang dikirimkan ke rangkaian *driver* motor oleh modul pengendali. *Duty cycle* PWM yang dikirimkan menentukan kecepatan putar motor DC. Gambar 2.10 menunjukkan bentuk IC L293 dan L298 *Multiwatt* 15 yang digunakan sebagai motor *driver.*



Gambar 2.20 IC L293 dan L298

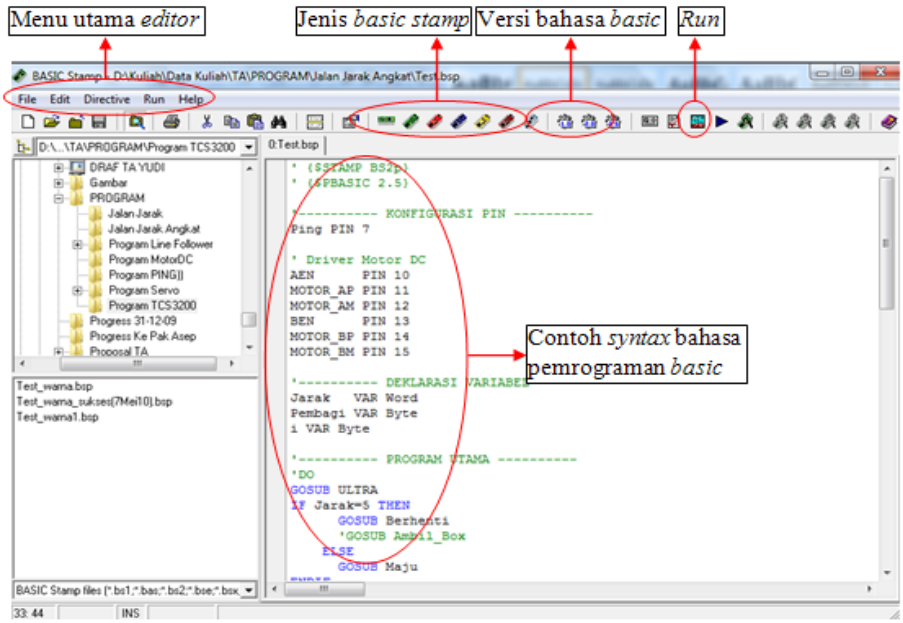
2.3. Perangkat Lunak (*Software*)

2.3.1. Pengenalan *Basic stamp Editor*

Perangkat lunak ini merupakan algoritma gerak dan tugas *flying robot* dalam bentuk listing program yang disimpan kedalam mikrokontroler. 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 2.4 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 2.21 Tampilan *basic stamp editor*

2.3.1.1 . 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 :

```
' {$STAMP BS2p}          ' tipe mikrokontroler yang digunakan
' {$PBASIC 2.5}          ' bahasa PBASIV kompiler yang digunakan
```

Gambar 2.22 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 Sintak Program

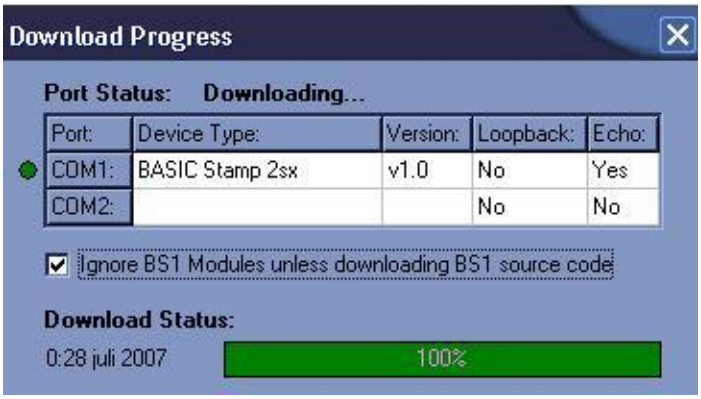
Memeriksa sintak program lakukan untuk memastikan semua sintak sudah benar. Untuk memeriksa sintak ini bisa pilih menu *RUN*, Cek Sintak atau kombinasi tombol CTRL+T. Tampilan jika listing program yang kita buat sudah benar.



Gambar 2.23 Hasil pemeriksaan sintak yang sukses *(tokenize successful)*

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 2.24 Tampilan jika pen*download*an program sukses

2.3.2. Pengenalan *Visual Basic 6.0*

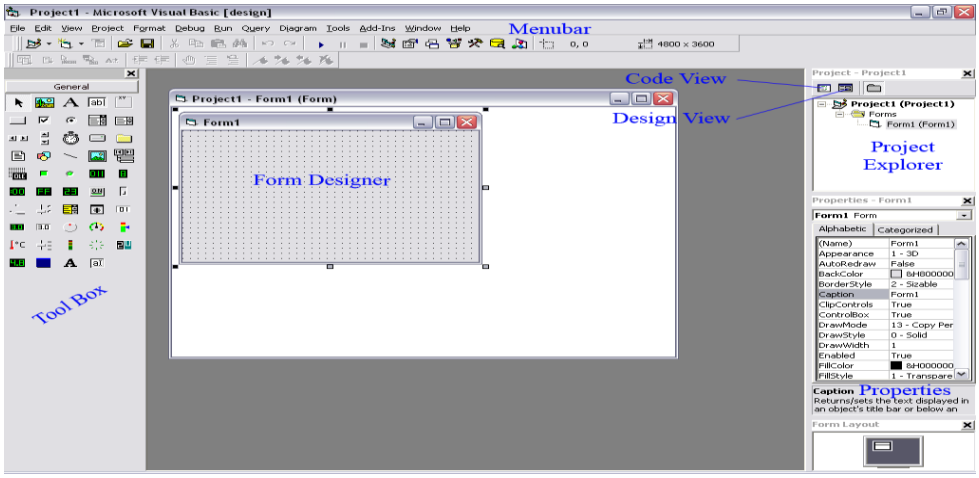
Visual Basic merupakan bahasa pemrograman komputer. Di dalam*Visual Basic 6.0* menyediakan sintak-sintak dalam pembuatan suatu aplikasi. Dengan menggunakan *Visual Basic 6.0* dapat menghasilkan berbagai macam jenis program. Aplikasi yang dibuat dapat diintegrasikan dengan *database*, *hardware* lain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

Pada layar awal akan muncul tampilan di bawah ini.



Gambar 2.25 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 2.26 Tampilan lembar kerja *Visual Basic 6.0*

2.3.3.Pengkodean pada *Visual Basic 6.0*

2.3.3.1.Tipe Data

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

Table 2.5 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*

2.3.3.2. 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.

2.3.3.3. Operator

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 2.6 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 2.7 Operator Komparasi

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

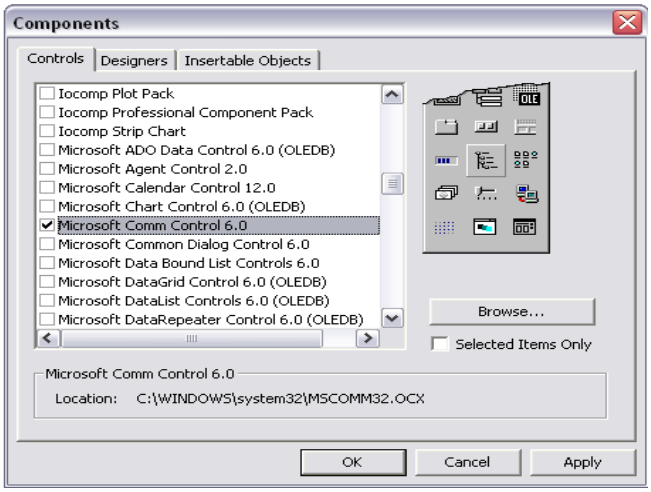
c. **Operator Logika**

Tabel 2.8 Operator Logika

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

2.3.4. 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 2.27 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 = 1
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 )
```