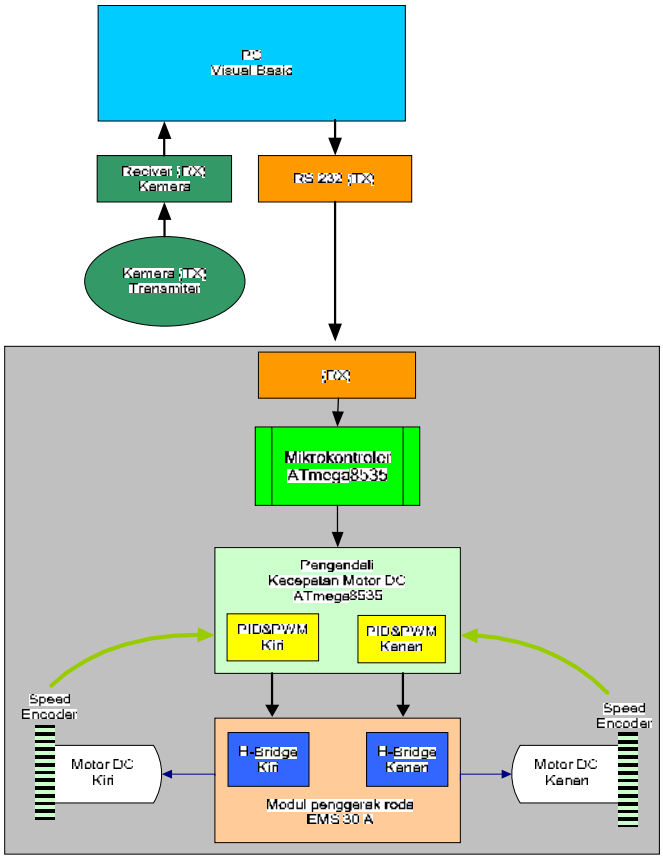
BAB III
PERANCANGAN SISTEM

Pada bab ini akan di jelaskan mengenai komponen-komponen apa saja yang akan digunakan untuk membangun prototipe *mobile robot* penjinak bom berdasarkan landasan teori. Secara umum sistem kontrol yang terdapat pada prototipe *mobile robot* penjinak bom terdapat dua bagian dasar, yaitu PC dan *mobile robot*. Dimana setiap bagian terbagi atas perangkat keras *(hardware)* dan perangkat lunak *(software)*. Gambar 3.1 merupakan diagram blok keseluruhan dari sistem.



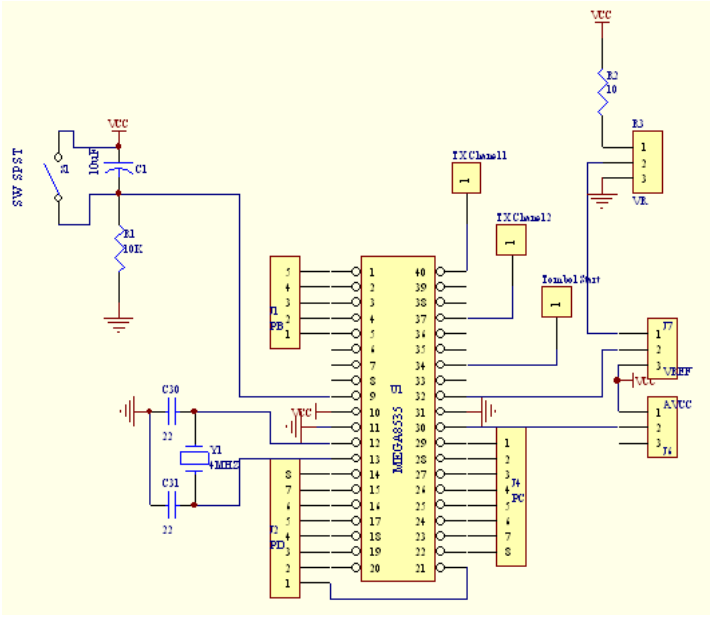
Gambar 3.1 *Diagram Blok Keseluruhan Sistem*

Diagram blok di atas dapat dijelaskan, PC berfungsi untuk mengirimkan data yang digunakan untuk mengendalikan jalannya *mobile robot*. PC ini akan mengendalikan *mobile robot* dengan memberikan data ke mikrokontroler utama melalui komunikasi serial RS-232 dan menampilkan gambar dari kamera yang diletakkan pada *mobile robot* melalui tampilan jendela Visual Basic. Data yang diterima melalui RS-232 diolah untuk menentukan aksi yang harus dilakukan oleh *mobile robot*.

3.1 PERANGKAT KERAS *(HARDWARE)*

3.1.1 Mikrokontroler ATmega8535

Pemilihan mikrokontroler ATmega8535 didasarkan pada fasilitas yang berada di dalam mikrokontroler dan ketersediaan komponen di pasaran. Gambar 3.3 merupakan gambar rangkaian sistem minimum dari mikrokontroler ATmega8535.



Gambar 3.2 *Rangkaian Sistem Minimum ATmega8535*

Table 3.1 menunjukan penggunaan pin-pin pada mikrokontroler ATmega8535 *(Master)*.

Tabel 3.1 *Konfigurasi Penggunaan Pin Mikrokontroler ATmega8535(Master)*

Nama Port	No Pin	Nama Pin	Type Pin	Keterangan
Port A	40	Port A.0	I/O	TX (#1)
	39	Port A.1	I/O	Ping
	38	Port A.2	I/O	Motor Servo1
	37	Port A.3	I/O	Motor Servo2
	36	Port A.4	I/O	Motor Servo3
	35	Port A.5	I/O	
	34	Port A.6	I/O	Tombol Start
	33	Port A.7	I/O	
Port D	14	Port D.0	I/O	Rx (PC)
	15	Port D.1	I/O	Tx (PC)
	16	Port D.2	I/O	
	17	Port D.3	I/O	
	18	Port D.4	I/O	
	19	Port D.5	I/O	
	20	Port D.6	I/O	
	21	Port D.7	I/O	

Keterangan :

Port A.0 pada mikrokontrler ATmega8535 digunakan sebagai *output* komunikasi serial *(Transmiter)* ke kendali motor.

Port A.1 pada mikrokontrler ATmega8535 digunakan sebagai *input* dari sensor ultrasonik

Port A.1,2,3 digunakan untuk memberi *input* pulsa pada motor servo.

Port A.6 digunakan sebagai *input* tombol mulai.

Port D.0 digunakan sebagai *input* komunikasi serial (*Receiver*) dengan PC.

Port D.1 digunakan sebagai *output* komunikasi serial (*Transmiter*) dengan PC.

3.1.2 Modul Driver Motor EMS 30A H-Bridge

Embedded Module Series (EMS) 30 A H-Bridge merupakan *driver H-Bridge* berbasis VNH3SP30 yang dirancang untuk menghasilkan *drive* 2 arah dengan arus kontinyu sampai dengan 30 A pada tegangan 5,5 Volt sampai 36 Volt (IC VNH2SP30 hanya sampai 16 V). Modul ini dilengkapi dengan rangkaian sensor

arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini mampu men-*drive* beban-beban induktif seperti misalnya relay, solenoida, motor DC, motor *stepper*, dan berbagai macam beban lainnya.

Spesifikasi:

- Terdiri dari 1 *driver full H-Bridge* beserta rangkaian *current sense.*
- Mampu melewatkan arus kontinyu 30 A.
- *Range* tegangan output untuk beban: 5,5 V sampai 16 V.
- Input kompatibel dengan *level* tegangan TTL dan CMOS.
- Jalur catu daya *input* (VCC) terpisah dari jalur catu daya untuk beban (V Mot).
- *Output tri-state.*
- Frekuensi PWM sampai dengan 20 KHz.
- *Fault Detection.*
- Proteksi hubungan singkat.
- Proteksi *overtemperature.*
- *Undervoltage* dan *Overvoltage Shutdown.*
- *Reverse Battery Protection.*



Gambar 3.3 *Modul EMS H-Bridge 30 A*

3.1.2.1 Deskripsi Pin Modul EMS H-Bridge 30 A

Tabel 3.2 *Deskripsi Pin EMS H-Bridge 30 A*

No. Pin	Nama	I/O	Fungsi
1	MIN1	I	Pin *input* untuk menentukan *output* **MOUT1**
2	MIN2	I	Pin *input* untuk menentukan *output* **MOUT2**
3	MEN1	I/O	Pin *enable* untuk *output* **MOUT1** Diberi logika *High* untuk mengaktifkan *half* HBridge 1, diberi logika *Low* secara eksternal untuk menonaktifkan *half* H-Bridge 1 Jika terjadi kondisi *Fault* (*thermal shutdown*, *undervoltage*, *overvoltage*, dsb.), maka pin ini akan ditarik *Low* secara internal oleh modul *H-Bridge* untuk melaporkan adanya kondisi *Fault.*
4	MEN2	I/O	Pin *enable* untuk *output* **MOUT2** Diberi logika *High* untuk mengaktifkan *half* HBridge 2, diberi logika *Low* secara eksternal untuk menonaktifkan *half* H-Bridge 2 Jika terjadi kondisi *Fault* (*thermal shutdown*, *undervoltage*, *overvoltage*, dsb.), maka pin ini akan ditarik *Low* secara internal oleh modul *H-Bridge* untuk melaporkan adanya kondisi *Fault*
5	MCS	O	Output tegangan analog yang berbanding lurus dengan arus beban (*Range* output 0 – 5 Volt) Tersedia untuk IC VNH2SP30
6	MPWM	I	Pin *input* untuk mengatur kerja modul *H-Bridge* secara PWM
7,9	VCC	-	Terhubung ke catu daya untuk *input* (5 Volt)
8,10	PGND	-	Titik referensi untuk catu daya *input*

3.1.2.2 Tabel Kebenaran

Tabel 3.3 *Tabel Kebenaran EMS H-Bridge 30 A*

Status kerja modul H-Bridge	Input dan Status					Output	
	MPWM	MIN1	MIN2	MEN1	MEN2	MOUT1	MOUT2
Forward	H	H	L	H	H	V MOT	MGND
Reverse	H	L	H	H	H	MGND	V MOT
Brake to GND	H	L	L	H	H	MGND	MGND
Brake to VCC	X	H	H	H	H	V MOT	V MOT
Free Running Stop	L	L	L	H	H	OPEN	OPEN
Free Running Stop	L	H	L	H	H	V MOT	OPEN
Free Running Stop	L	L	H	H	H	OPEN	V MOT
Fault di OUT1 dan OUT2	X	X	X	L	L	OPEN	OPEN
Fault di OUT1	H	X	H	L	H	OPEN	V MOT
Fault di OUT1	H	X	L	L	H	OPEN	MGND
Fault di OUT2	H	H	X	H	L	V MOT	OPEN
Fault di OUT2	H	L	X	H	L	MGND	OPEN

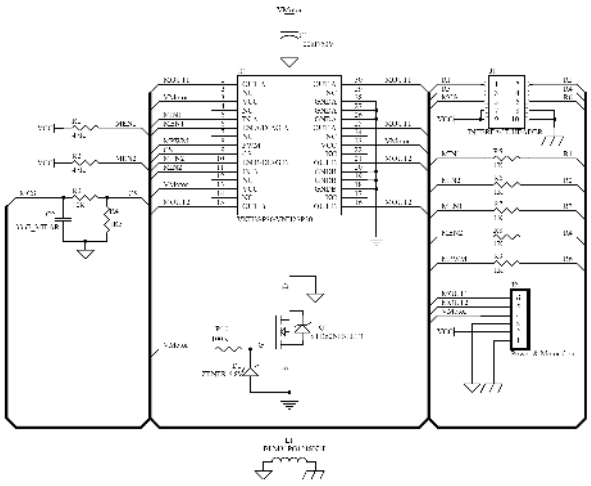
Deskripsi lebih jelas tentang status kerja tersebut dapat dilihat pada datasheet IC

Keterangan :

H = High atau logika 1 *L = Low atau logika 0*

X = don't care *Z = High Impedance (Tri-state)*

3.1.2.3 Skema EMS H-Bridge EMS 30 A

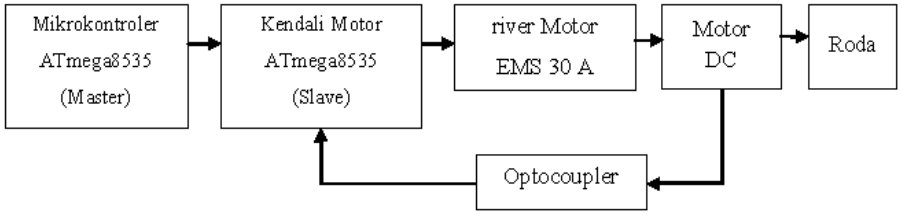


Gambar 3.4 *Skema EMS H-Bridge 30 A*

Gambar 3.4 merupakan gambar rangakaian pada modul EMS *H-Bridge* 30 A dan hubungan antara pin *input* dan *output* dengan komponen yang di dalamnya.

3.1.3 Modul Kendali Motor

Keluaran dari *optocoupler* akan dihubungkan ke kendali motor yaitu dua buah mikrokontroler ATmega8535 *(Slave)* yang berfungsi membaca pulsa yang dihasilkan oleh *optocoupler*, yang nantinya kendali motor ini akan mengolah data masukan dari *optocoupler* sehingga dihasilkan laju pergerakan robot yang lebih stabil. Pengaturan kecepatan putaran motor dapat dilakukan dengan cara memberikan data serial dari Mirkokontroler ATmega8535 *(Master)* ke kendali motor melalui pin-pin Rx (*reciever*) dan Tx (*transmiter*) masing-masing. Data serial tersebut berupa nilai frekuensi, nilai frekuensi ini akan menjadi acuan/*setpoint* bagi kendali motor untuk memutarkan motor melalui *driver* motor dengan membaca frekuensi sebelumnya yang didapatkan dari hasil bacaan *optocoupler*. Jika frekuensi yang dikirimkan oleh mikrokontroler lebih besar dari pada frekuensi yang dihasilkan oleh *optocouper* maka, kendali motor akan meningkatkan frekuensi tersebut sehingga mendekati nilai yang diinginkan dan sebaliknya jika frekeunsi yang diinginkan lebih kecil maka kendali motor akan menurunkan kecepatan dari putaran motor melalui *driver* motor. Berikut merupakan gambar diagram blok kendali motor:



Gambar 3.5 *Diagram Blok Kendali Motor*

Table 3.4 menunjukkan penggunaan pin-pin pada mikrokontroler ATmega8535 *(Slave)*.

Tabel 3.4 *Konfigurasi Penggunaan Pin Mikrokontroler (Slave)*

Nama Port	No Pin	Nama Pin	Type Pin	Keterangan
Port B	1	Port B.0	I/O	Timer0/Counter)
Port D	14	Port D.0	I/O	Rx *(Master)*
	15	Port D.1	I/O	
	16	Port D.2	I/O	
	17	Port D.3	I/O	Motor Dir1
	18	Port D.4	I/O	Motor Dir2
	19	Port D.5	I/O	Motor PWM
	20	Port D.6	I/O	Motor Enable
	21	Port D.7	I/O	Optocoupler

Keterangan :

Port B.0 *Timer0/Counter0* digunakan untuk menghitung jumlah pulsa yang dihasilkan sesor *optocoupler*.

Port D.0 pada mikrokontrler ATmega8535 *(Slave)* digunakan sebagai *input* komunikasi serial *(Receiver)* ke kendali motor kanan.

Port D.3 digunakan sebagai *output* untuk *direction1 driver* motor.

Port D.4 digunakan sebagai *output* untuk *direction2 driver* motor.

Port D.5 digunakan sebagai *output* untuk PWM *driver* motor.

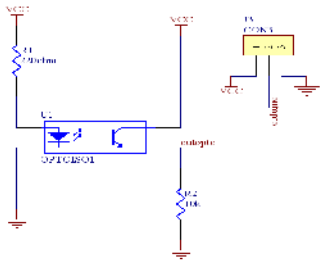
Port D.6 digunakan sebagai *output* untuk *enable driver* motor.

Port D.7 digunakan sebagai *input* dari sensor *optocoupler*.

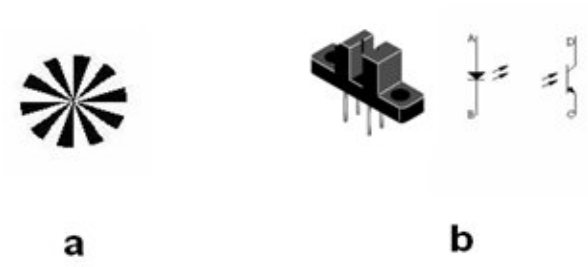
3.1.4 Sensor Optocoupler

Untuk mengatasi pergerakkan robot yang masih dipengaruhi oleh perubahan tegangan, solusi yang digunakan adalah dengan memasangkan *speed encoder* pada *shaft* motor. Sensor sederhana yang biasanya digunakan untuk *speed encoder* adalah *optocoupler*. *Optocoupler* terdiri dari pemancar dan penerima, pada bagian pemancar terdiri atas sebuah led dengan intensitas pancaran dapat diatur dengan merubah-rubah besaran resistor (220Ω-470Ω) yang terhubung ke anodanya. Bagian penerima *optocoupler* adalah sebuah trasnsistor, ketika pemancar tidak terhalangi maka *output* pada transistor akan berlogika 1 dan ketik pemancar terhalangi *output* pada transistor akan berlogika 0. Pada *shaft* tersebut

dipasangkan suatu plastik dengan diameter tertentu yang memiliki pola hitam dan transparan. Ketika pemancar mengenai hitam (terhalang) *output* transistor akan berlogika 0, ketika pemancar mengenai transparan (tidak terhalang) *output* transistor akan berlogika 1. Bila motor diberikan catu daya dengan tegangan tertentu yang akan memutarkan motor, bacaan dari *optocoupler* akan mengasilkan pulsa kotak persatuan waktu. Kerapatan dari pulsa yang dihasilkan bergantung kepada kecepatan putaran motor, semakin cepat putaran motor semakin rapat pulsa yang dihasilkan.



Gambar 3.6 *Rangkaian Optocoupler*

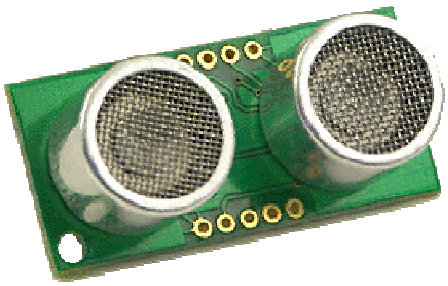


Gambar 3.7 *(a) Lingkaran plastik dengan pola tertentu*
(b) Rangkaian internal optocoupler dan bentuk fisik optocoupler

3.1.5 Sensor Ultrasonik

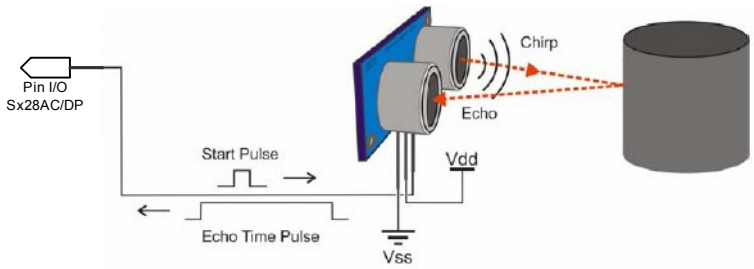
Sensor *ultrasonik* yang digunakan adalah SRF05 yang merupakan produk dari Devantech. Sensor ini mendukung untuk digunakan dengan mikrokontroler ATmega8535. Beberapa fitur yang menjadi pertimbangan pemilihan sensor ini sebagai pengukur jarak adalah:

1. Tegangan *input* (Vcc) adalah 5 Vdc.
2. Tipe arus *input* adalah 30 mA dengan arus maksimum 35 mA.
3. Jarak yang dapat diukur mulai 1 cm sampai 4 m.



Gambar 3.8 *Sensor Ultrasonik*

Gambar 3.10 menunjukkan cara kerja sensor *ultrasonik*, SRF mengirim gelombang *ultrasonik* singkat dan menunggu *echo* (suara pantulan) dari gelombang *ultrasonik* yang dikirim terlebih dahulu. Terdapat dua langkah untuk mengetahui besarnya jarak antara robot dengan objek. Pertama mikrokontroler harus mengirimkan pulsa (*trigger time*) selama 10 µS<$t_{trigger}$ untuk memulai pengukuran. Pada waktu yang hampir bersamaan, sensor *ultrasonik* mengirimkan gelombang *ultrasonik* pada level 40 kHz selama 200 µS melalui *speaker ultrasonik* dan mengirimkan sinyal *high* (1) ke mikrokontroler. Ketika mikrofon *ultrasonik* mendeteksi adanya *echo*, sinyal *high* (1) berubah menjadi *low* (0). Waktu yang dibutuhkan gelombang *ultrasonik* menuju objek dan kembali ke mikrofon sensor *ultrasonik* bisa dijadikan sebagai variabel untuk menghitung jarak antara robot dengan objek.



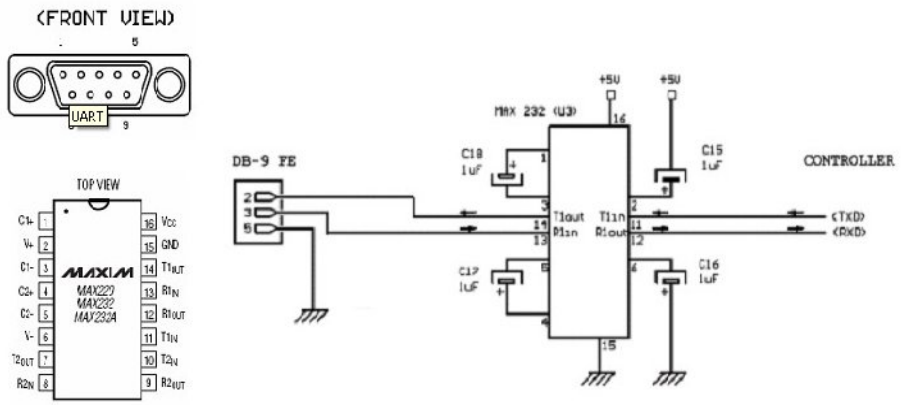
Gambar 3.9 *Prinsip Kerja Sensor Ultrasonik*

3.1.6 Modul Komunikasi

Untuk dapat mengirimkan instruksi dari PC ke *mobile robot* dibutuhkan modul komunikasi sesuai dengan spesifikasi yang dibutuhkan adapun modul komunikasi yang dipergunakan diantaranya:

3.1.6.1 Port Serial/RS-232

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* RS-232 dimana pada *level* RS-232, tegangan *high* diwakili dengan tegangan +3 sampai +25 V. Diantara -3 dan +3 merupakan tegangan *invalid* atau tidak sah. Pada Gambar 3.10 merupakan rangkaian RS-232.



Gambar 3.10 *Rangkaian RS-232*

Nilai kapasitor disesuaikan dengan saran dari pabrik pembuat (Maxim) dimana IC MAX232 mempunyai dua saluran untuk komunikasi serial namun pada alat hanya digunakan satu saja. TxD dan RxD dihubungkan dengan mikrokontroler ATmega8535 pin Rx dan Tx sementara R1IN dan T1OUT dihubungkan dengan komputer melalui serial RS-232.

3.1.6.2 Media Komunikasi

Media komunikasi yang digunakan pada *mobile robot* yaitu melalui frekuensi radio (RF). Untuk komunikasi melalui frekuensi radio menggunakan modul YS-1020U.



Gambar 3.11 *YS-1020U*

YS-1020U merupakan modul komunikasi yang sangat aman, mempunyai 8 kanal dengan frekuensi yang berbeda. Jarak jangkauan komunikasi sekitar 500 meter pada *baudrate* 9600 bps dan maksimum 800 meter dengan *baudrate* 1200 bps

Tabel 3.5 *Deskripsi pin-pin YS-1020U*

Pin	Nama Pin	Fungsi	Level
1	GND	*Ground*	
2	Vcc	Tegangan input	+3.3~5.5V
3	RXD/TTL	Input Serial data	TTL
4	TXD/TTL	Output Serial data	TTL
5	DGND	Digital grounding	
6	A(TXD)	A of RS-485 or TXD of RS-232	
7	B(RXD)	B of RS-485, RXD of RS-232	
8	SLEEP	Sleep control (input)	TTL
9	RESET	Reset (input)	TTL

3.1.7 Gripper

Komponen pendukung pada *mobile robot* diantaranya adalah sebuah *gripper* yang digunakan untuk mengambil sebuah objek. Pada perancangan *mobile robot* ini dipasangkan sebuah *gripper* yang terbuat dari bahan *fiberglass* yang digerakkan dua buah motor servo. *Gripper* ini memiliki gerak dua derajat kebebasan yang artinya *gripper* dapat bergerak dua arah. Berikut merupakan gambar tampilan *gripper* pada *mobile robot*.



Gambar 3.12 *Gripper dan Motor Servo*

3.1.8 Catu Daya / Regulator

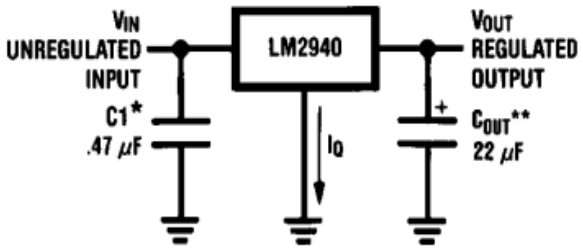
Catu daya adalah faktor pendukung yang sangat penting, karena mikrokontroler, sensor-sensor dan aktuator dapat bekerja karena adanya tegangan. *Power supply* yang dibutuhkan sebanyak 2 buah, pertama untuk mencatu mikrokontroler dan sensor-sensor yang membutuhkan level tegangan 5 volt. Kedua, tegangan 19.2 volt sebagai tegangan kerja aktuator berupa motor *DC*. Pemisahan tegangan ini dimaksudkan agar tegangan yang masuk ke mikrokontroler tidak terganggu oleh tegangan dari motor.

Tegangan yang digunakan berasal dari baterai *rechargeable* yang mempunyai nilai tegangan 1,2 volt per satu baterai. Untuk *supply* tegangan ke mikrokontroler dan sensor-sensor digunakan 8 buah baterai yang disusun seri sehingga menghasilkan tegangan total sebesar 9,6, karena mikrokontroler dan sensor-sensor hanya mampu bekerja pada level tegangan 5 volt, untuk itu diperlukan regulator tegangan untuk mendapatkan tegangan 5 volt, dengan demikian digunakan IC regulator LM2940.



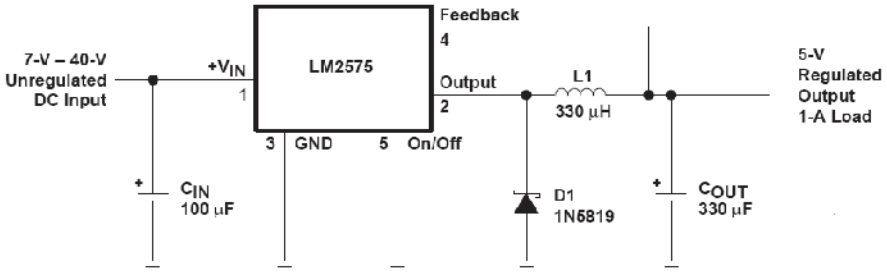
Gambar 3.13 *(a) IC Regulator; (b) Kapasitor polar*

Gambar 3.14 menunjukkan catu daya untuk tegangan *output* 5 Vdc (nilai terukur 4.9 volt) menggunakan IC LM2940 sebagai regulator tegangan positif yang dihasilkan dari tegangan sumber Vin sebesar 9.6 Vdc. Kapasitor digunakan untuk menghilangkan *ripple* yang berasal dari tegangan *input*.



Gambar 3.14 *Schematic regulator 5 volt dengan LM2940*

IC VNH3SP30 sebagai driver motor membutuhkan tegangan logic 5 volt sama halnya dengan sensor-sensor dan mikrokontroler. Untuk menghasilkan tegangan 5 volt tersebut dibutuhkan IC yang berbeda dari regulator LM2940, hal ini dikarenakan arus yang masuk ke driver motor ± 2A, dengan arus sebesar itu IC LM2940 tidak bekerja dengan baik, IC yang digunakan sebagai penggantinya adalah LM2575, mampu bekerja dengan teganan sumber sebesar 40 volt dengan arus 3 A



Gambar 3.15 *Schematic regulator 5 volt dengan LM2575*

Dari gambar 3.15 L1 , D1 dan Cout berfungsi untuk menghilangkan *ripple* di tegangan keluaran.

3.2 PERANGKAT LUNAK *(SOFTWARE)*

3.2.1 Algoritma Gerak Dasar

Perancangan algoritma dasar merupakan hal yang sangat penting dalam perancangan sistem kendali pada sebuah sistem robot, sebab algoritma dasar inilah yang akan menjadi panduan dalam pengembangan/perancangan algoritma kendali secara keseluruhan. Algoritma-algoritma dasar tersebut diantaranya :

3.2.1.1 Maju dan Mundur

Gerakan maju dilakukan dengan menggerakkan roda kiri dan roda kanan ke arah depan secara bersama-sama, begitu pula dengan gerakkan mundur, yaitu dengan menggerakkan roda kiri dan roda kanan kearah belakang secara bersama-sama. Untuk dapat menggerakkan robot dengan arah maju, ada 3 mode intruksi yang diterapkan kepada mikrokontroler sebagai berikut :

3.2.1.1.1 Mode SetDC

Mode SetDC merupakan mode dimana motor akan berputar terus-menerus sampai ada instruksi selanjutnya.

Untuk dapat menggerakan motor, bergerak maju atau mundur kita dapat memberikan input logika *high* (1) atau *low* (0) pada *driver* motor sesuai dengan tabel kebenaran *driver* motor. Berikut penggalan program untuk menggerakan motor maju:

```
Program_maju:
Motor_dir1 = 1        'MIN1 (Pin input untuk menentukan output MOUT1)
Motor_dir2 = 0        'MIN2 (Pin input untuk menentukan output MOUT2)
Motor_enable = 1      'MEN1 & MEN2 (Pin enable untuk output MOUT1
                      & MOUT2)
Motor_pwm = 255       'MPWM (Pin input untuk mengatur kerja modul H-
                      Bridge secara PWM)
```

3.2.1.1.2 Mode Count

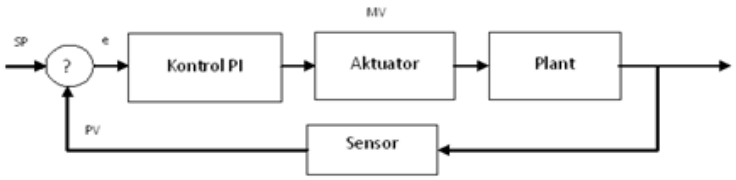
Mode *count* adalah mode untuk menggerakkan motor dengan putaran yang ditentukan. Dapat digunakan untuk bergerak dengan jarak pasti.

Mode *count* juga digunakan untuk belok kanan dan belok kiri. Penggunaan mode *count* untuk belok hanya dengan merubah arah putaran motor. Pada mikrokontroler penggunaan mode *count* ini memanfaatkan fasilitas *Timer0*/*Counter0* untuk menghitung banyaknya putaran dari pembacaan sensor *optocoupler* terhadap klip *shaft encoder* yang dipasang pada poros motor. Berikut adalah penggalan program untuk mode *count*:

```
Config Timer0 = Counter , Edge = Rising   'Setting Timer0 sebagai Counter
Tcnt0 = 0                                 'Inisialisari register Timer0
Motor_dir1 = 1                            'Gerak maju atau mundur
Motor_dir2 = 0
Motor_enable = 1
Motor_pwm = 255
For X = 1 To 5 Step 1                     'Ulang sebanyak 5 kali
    Do                                    'Baca pulsa sebanyak 255 kali
    Loop Until Tcnt0 >= 255
Next
```

3.2.1.1.3 Mode PI

Mode PI digunakan dengan memberikan *setpoint* berupa nilai kecepatan pada mikrokontroler sehingga motor akan berjalan sesaui dengan *input* kecepatan dan dapat memperbaiki *error* kecepatan, kembali menju *setpoint* dengan memanfaatkan umpan balik dari sensor. Berikut gambar diagram blok kontrol PI:



Gambar 3.16 *Diagram Blok Kontrol PI*

Respon dari kontrol PI ditentukan oleh konstanta Kp dan Ki. Selain itu perlu dilakukan pembatasan *output* agar tidak terjadi *overflow output*. Berikut penggalan program untuk mode PI:

```
Mode_PI:
Pwm_pi = 00                    ‘PWM di set 0
Const Kp = 5                   ‘konstanta Kp = 5
Const Ki = 1                   ‘konstanta Ki = 1
Const Setpoint = 30            ‘Setpoint = 30
Error = Setpoint – Rpm         ‘cek error
P = Kp * Error                 ‘kendali P
Temp = Ki * Error
I = Temp + Ki_error_acc        ‘kendali I
Pi = P + I
If Pi < -150 Then              ‘Pembatasan output
    Ki_error_acc = -150 – P
Elseif Pi > 150 Then
    Ki_error_acc = 150 – P
Else
    Ki_error_acc = I
End If
Pi = P + Ki_error_acc
Temp = Pi + 150
Temp = Temp * 180
Temp = Temp / 300
Motor_pwm = 75 + Temp          ‘output kendali PI
```

3.2.2 Visual Basic V6.0

Pada bagian ini menjelaskan tentang perancangan perangkat lunak untuk PC, yang berfungsi untuk mengirim data ke mikrokontroler. Untuk perangkat lunak disini menggunakan program *Visual Basic V6.0* yang memiliki tampilan *visual* dan grafis, sehingga akan lebih menarik untuk dilihat. Untuk program *Visual Basic* ini sebetulnya terbagi menjadi beberapa bagian, tapi disini akan dibahas secara langsung tentang perancangan perangkat lunak yang paling utama, dimana dapat menerima data dan kode dari pengirim.

Berikut ini adalah daftar kode karakter beserta penjelasan fungsinya seperti yang terlihat pada tabel 3.5:

Tabel 3.6 *Kode Karakter Beserta Fungsinya*

Kode	Fungsi
“q”	Maju Mode PI
“w”	Mundur Mode PI
“e”	Kanan 45
“a”	Kanan 90
“s”	Kiri 45
“d”	Kiri 90
“z”	Stop
“c”	Rem
“r”	Kamera Depan
“t”	Kamera Bawah
“j”	Buka Gripper
“k”	Ambil Gripper
“l”	Tutup Griper
“f”	Maju Pelan
“g”	Mundur Pelan
“v”	Maju Sedikit
“b”	Mundur Sedikit
“n”	Kiri Dikit
“m”	Kanan Dikit

Dalam program ini dibuat suatu fasilitas komunikasi antara program aplikasi yang dibuat dengan *port* serial. Untuk mengirim dan menerima data melalui *port* serial. Setiap MSComm hanya menangani satu *port* serial. Untuk penulisan program dapat dilihat pada program dibawah ini:

Comm1.CommPort = 1	*port* yang digunakan adalah COM1
Comm1.Settings = "9600,n,8,1"	*Setting Comm1* adalah *baudrate* 9600, tanpa paritas, jumlah data 8 bit dan jumlah bit *stop* adalah 1 bit.
Comm1.PortOpen = True	Membuka *port* serial

Berikut adalah contoh *listing* program untuk mengirim karakter “w” ke mikrokontroler menggunakan tombol Command1_Click() di Visual Basic 6.0:

```
Private Sub Command1_Click()
Comm1.CommPort = 1
Comm1.Settings = "9600,n,8,1"
Comm1.PortOpen = True
Comm1.Output = "w"
End Sub
```

Untuk penulisan program secara keseluruhan dapat dilihat pada bagian lampiran.

3.2.2.1 Perancangan Tampilan Utama

Perancangan menu utama *software* Visual Basic sebagai tampilan di PC dan untuk memasukkan data-data yang akan dijalankan oleh *mobile robot*. Menu utama meliputi objek-objek seperti *form*, *button* (tombol), *textbox* (kotak isian) *label* dan *activeX.*

Perancangan Tampilan Visual Basic terbagi atas tiga bagian utama yaitu:

1. *Panel input* mikrokontroler untuk pemilihan gerak.
2. *Activex video capture* untuk menampilkan gambar dari kamera yang di pasang pada *mobile robot*.
3. *Panel setting* warna gambar *video capture*

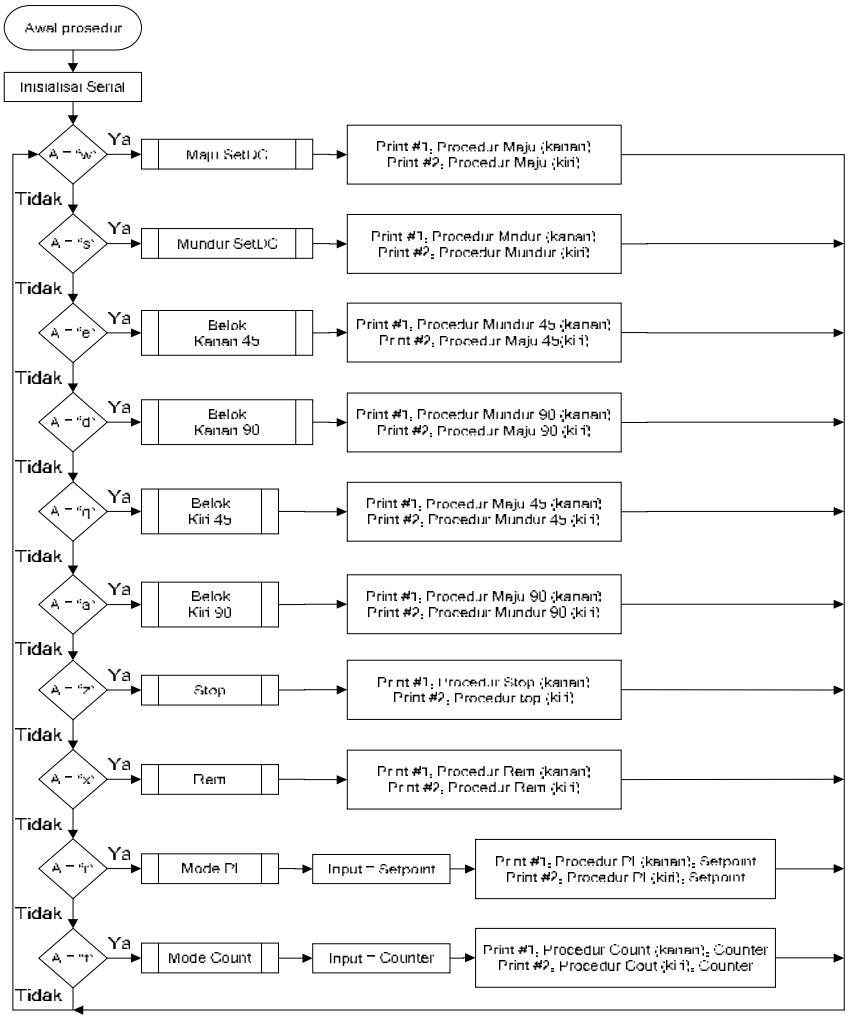
3.2.3 Perancangan Perangkat Lunak Mikrokontroler

Sistem perangkat lunak *(software)* merupakan suatu perangkat yang dibuat untuk mengendalikan sistem kerja dari masing-masing komponen, sistem ini dikendalikan oleh mikrokontroler yang juga mengendalikan perangkat keras lainnya dalam memberikan suatu *input* dan *output* data serta pertukaran informasi.

Perancangan perangkat lunak mikrokontroler ini dibagi menjadi dua bagian yaitu mikrokontroler utama *(master)* dan mikrokontroler kendali motor *(slave)*.

3.2.3.1 Diagram Alir Program Mikrokontroler *(Master)*

Mikrokontroler utama *(master)* digunakan untuk komunikasi antara kendali PC dengan *mobile robot*.

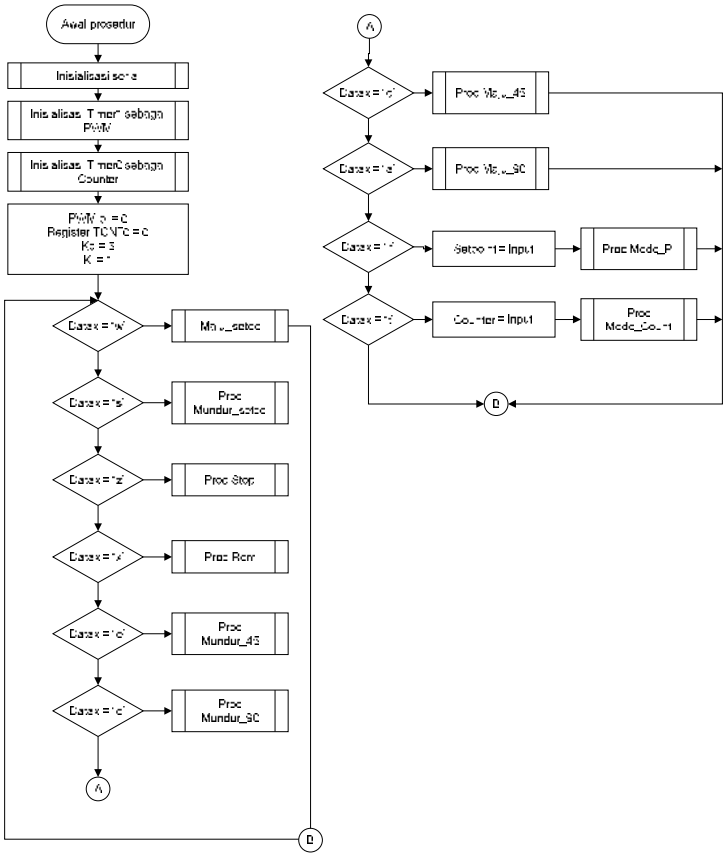


Gambar 3.17 *Diagram Alir Mikrokontroler (Master)*

Diagram alir di atas merupakan diagram alir program mikrokontroler utama *(master)*. Diagram alir program di atas diawali dengan penginisialisasian komunikasi di lanjutkan dengan pemilihan gerak dan pemanggilan prosedur gerak yang akan dikirimkan ke kendali motor.

3.2.3.1.1 Diagram Alir Program Mikrokontroler *(Slave)*

Mikrokontroler kendali motor *(slave)* digunakan untuk mengendalikan motor berdasarkan perintah dari mikrokontroler utama *(master)*



Gambar 3.18 *Diagram Alir Mikrokontroler Kendali Motor (Slave)*

Gambar 3.18 merupakan diagram alir program mikrokontroler kendali motor *(slave)*. Program kendali motor diawali dengan inisialisasi komunikasi serial, *Timer0/Counter0*, dan *PWM*. Dilanjutkan dengan pemilihan gerak sesuai dengan masukan yang diberikan oleh mikrokontroler utama *(master)*. Setelah pemilihan selesai, kemudian lakukan pemanggilan prosedur gerak sesuai dengan hasil pemilihan. Prosedur gerak yang dapat lakukan oleh masing-masing mikrokontroler kendali motor adalah maju_setdc, mundur_setdc, maju_45, maju_90, mundur 45, mundur_45, mode_pi dan mode_count. Untuk mode_pi dan mode_count diwali dengan memberikan *input*-an *setpoint* dan nilai *counter*.