

3. PERENCANAAN

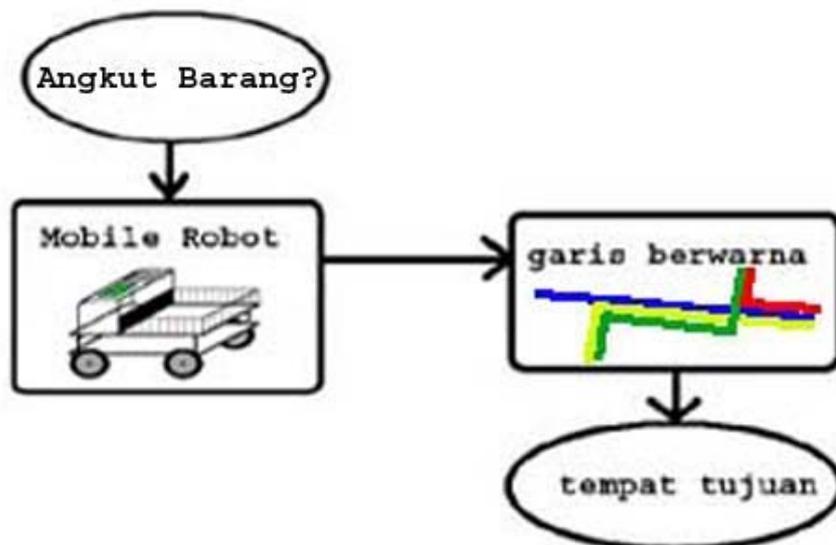
Dalam bab ini akan dibahas mengenai perencanaan sistem secara keseluruhan. Pada bagian pertama akan dibahas mengenai perancangan sistem, kemudian pada bagian kedua yang menjelaskan mengenai perangkat keras penunjang yang digunakan. Terakhir akan dibahas mengenai perencanaan *software* yang digunakan untuk mengolah informasi yang diperoleh dari perangkat keras.

3.1 Perencanaan Sistem

Mobile robot pengangkut barang ini dirancang agar dapat bergerak fleksibel di tempat yang berkeluk-luk dan sempit karena hanya menggunakan garis dengan warna tertentu sebagai jalur.

Sebagai sensor jalur berwarna, sistem ini menggunakan *CMU camera* sebagai *input* warna pada mikrokontroler. Untuk menggerakkan *mobile robot* ini, sistem menggunakan motor *gearbox* yang dikontrol oleh IC L298.

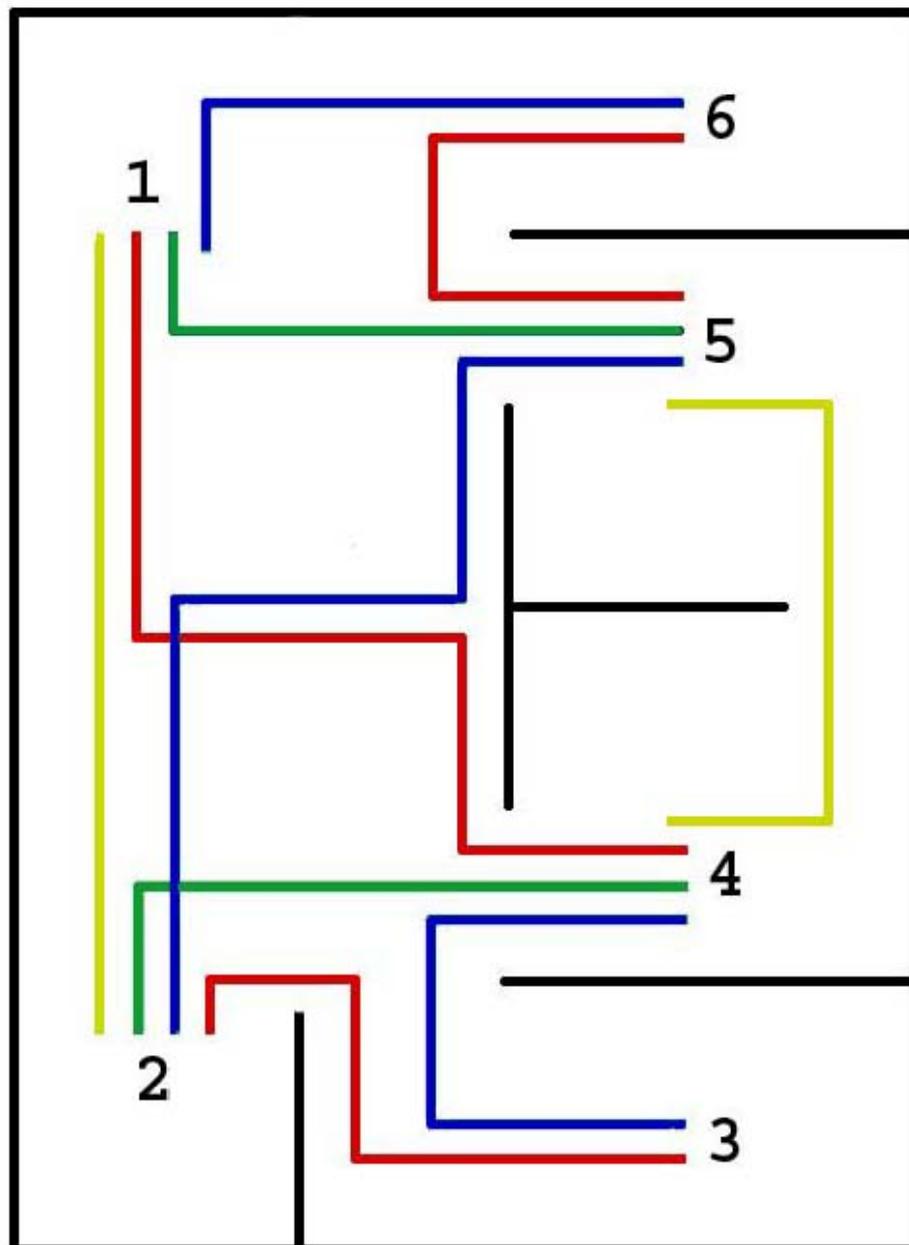
Selain itu, sistem ini juga menggunakan *keypad* sebagai *input* pilihan warna yang akan digunakan sebagai jalur dari *mobile robot* serta LCD sebagai tampilan warna yang dipilih.



Gambar 3.1 Gambaran Mobile Robot Secara Keseluruhan

Pada saat pertama kali mengaktifkan sistem, mikrokontroler akan menunggu *input*-an warna jalur (dari *keypad*) dan kemudian ditampilkan di LCD. Setelah itu mikrokontroler akan menunggu perintah dari *keypad* untuk menjalankan *mobile robot* sesuai dengan jalur yang sudah di-*input*-kan dari *keypad*.

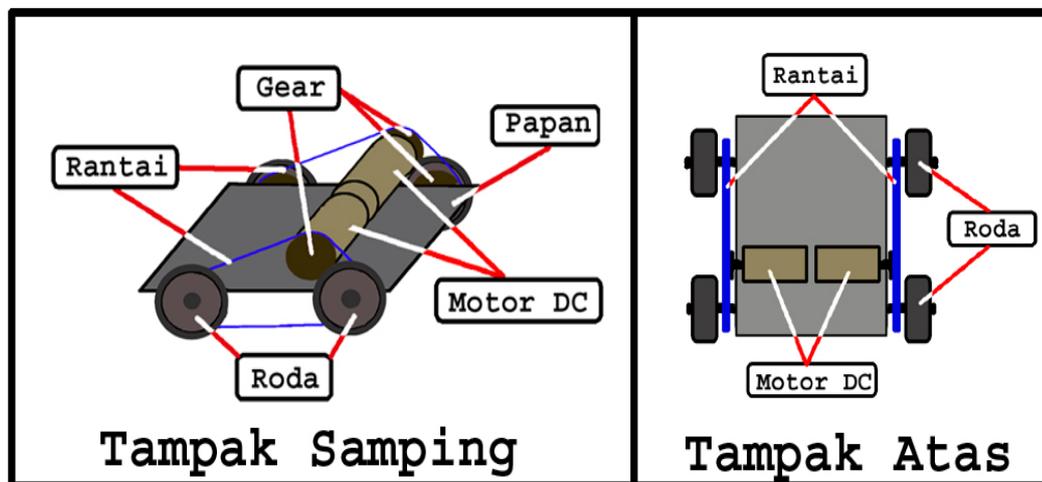
Setelah mencapai tempat tujuan (bagian ujung dari jalur warna tersebut), maka *mobile robot* akan berhenti dan menunggu untuk perintah selanjutnya. Pada gambar 3.2 dapat dilihat contoh gambar denah jalur *mobile robot*.



Gambar 3.2 Denah Jalur Mobile Robot

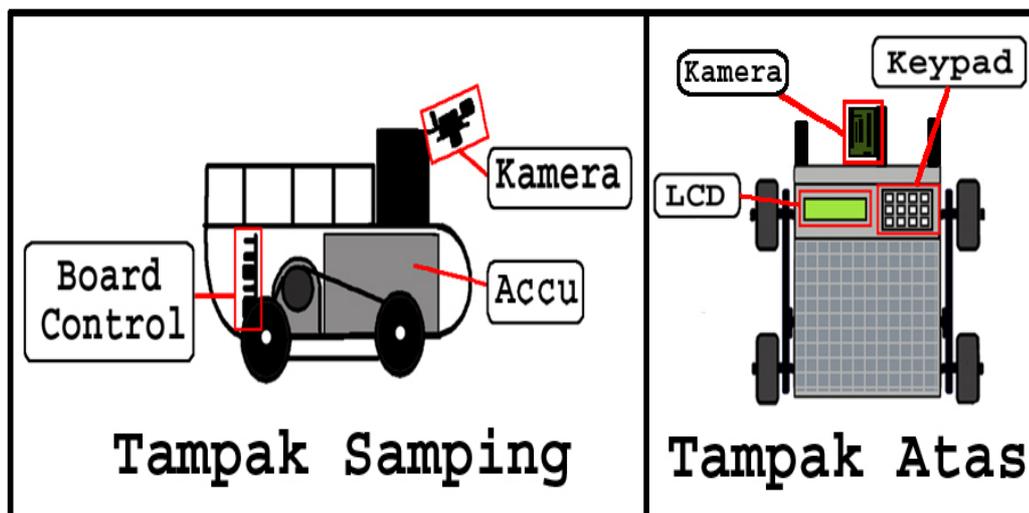
3.2 Perencanaan Perangkat Keras

Mobile robot ini akan digerakkan oleh 2 buah motor DC. Motor pertama akan mengatur putaran roda sebelah kiri (depan dan belakang) dan motor kedua akan mengatur putaran roda sebelah kanan (depan dan belakang). Untuk menghindari slip pada roda, maka roda depan dan belakang saling dihubungkan (agar kecepatan putar dari kedua roda sama). Motor dan kedua roda dihubungkan dengan menggunakan rantai, sehingga pada motor dan tiap-tiap roda dipasangi *gear*. Gambaran penempatan roda dan motor dapat dilihat pada gambar 3.3.



Gambar 3.3 Penempatan Roda Dan Motor

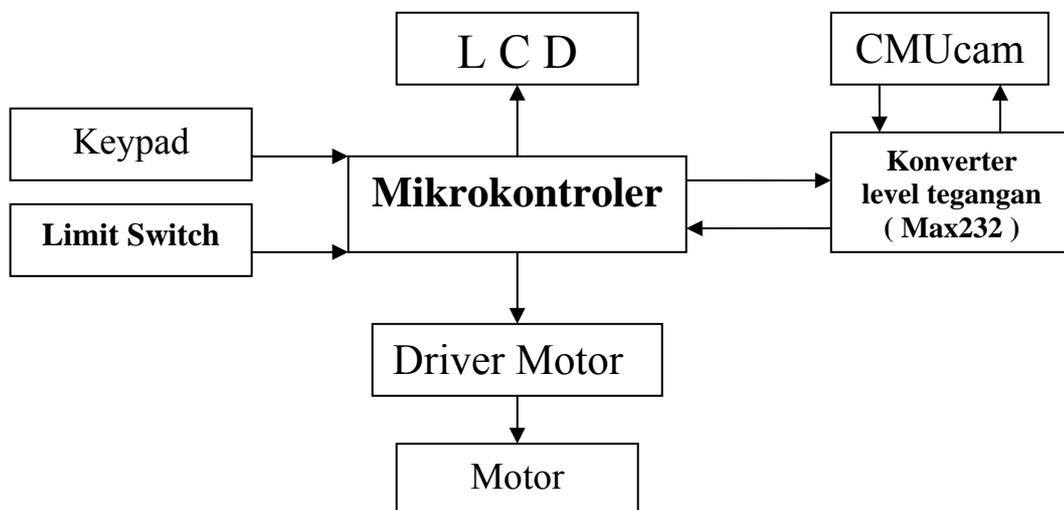
Mobile robot ini berfungsi untuk mengangkut barang sehingga bentuk *mobile robot* ini dirancang menyerupai *truck* pengangkut barang (gambar 3.4).



Gambar 3.4 Bentuk Mobile Robot

Selain itu, secara keseluruhan dalam perencanaan dan pembuatan perangkat keras terbagi-bagi menjadi 7 bagian yaitu sebagai berikut:

- Mikrokontroler
- Driver motor
- Kamera (CMUcam)
- Keypad
- Limit switch
- LCD
- Lampu penerang



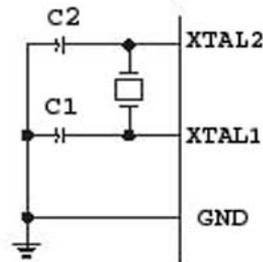
Gambar 3.5 Blok Diagram Mobile Robot

3.2.1 Mikrokontroler

Pada perangkat keras digunakan IC MCS51 yaitu tipe AT89S51 sebagai pusat kontrol dari sistem ini. Untuk mengontrol gerakan dari motor, kamera, keypad, limit switch dan LCD; mikrokontroler hanya membutuhkan *single chip* AT89S51 karena IC ini memiliki 32 *pin* I/O yang cukup untuk mengontrol semuanya. Selain itu, faktor utama dalam pemilihan IC AT89S51 adalah karena mudah di-program dan dihapus karena hanya memerlukan *ISP Cable Programming*. Untuk bahasa pemrograman, digunakan bahasa *Assembly* kompatibel dengan keluarga MCS-51.

Seperti mikrokontroler lainnya, AT89S51 juga membutuhkan *clock*. Dalam hal ini yang dipilih adalah menggunakan *internal clock* dengan tambahan

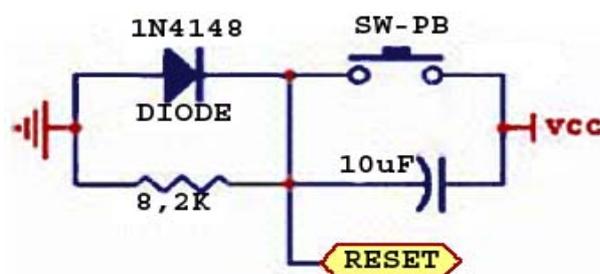
sebuah *crystal* dan dua buah *capacitor*. *Capacitor* yang digunakan menurut *datasheet* bernilai antara 30 pF hingga 40 pF. Sedangkan *crystal* yang digunakan memiliki nilai 22,1184 Mhz. Nilai ini diambil dengan pertimbangan agar mikrokontroler mampu mengimbangi kecepatan dari CMUcam.



Gambar 3.6 Rangkaian Osilator

Selain rangkaian *clock*, AT89S51 juga membutuhkan rangkaian *reset*. *Reset* (aktif “*high*”) diperlukan untuk memastikan agar CPU tersebut mengeksekusi instruksi mulai dari *address* 0000_H. *Reset* yang dibutuhkan menurut *datasheet* adalah sekitar dua *cycle machine* (24 osilator periode) . Dengan menggunakan *crystal* 22,1184 Mhz dapat dihitung waktu yang diperlukan untuk melakukan *reset* minimal adalah $1/22,1184 \times 24 = 1,08 \mu\text{s}$.

Rangkaian *reset* terdiri dari sebuah dioda, kapasitor senilai 10 uF, dan resistor 8,2 K Ω dan sebuah *push button* (untuk *reset manual*). Susunan rangkaian seperti tampak pada gambar 3.7 memungkinkan terjadinya *power-on reset*. Saat mikrokontroler mendapatkan *supply* untuk pertama kali, tidak diperlukan *reset* secara manual dengan menggunakan tombol. Menurut *datasheet*, V_{IH} minimum adalah 2,9 V_{DC}, untuk amannya diambil 4 V_{DC}. Berdasarkan rangkaian *power-on reset* didapatkan $V_R = 5 e^{-t/(8,2K \times 10 \mu)}$. Sesuai dengan rumus, jika diambil V_R sebesar 4V_{DC}, maka akan didapatkan $t = 18\text{ms}$. Sehingga waktu tersebut cukup untuk melakukan *power-on reset*.



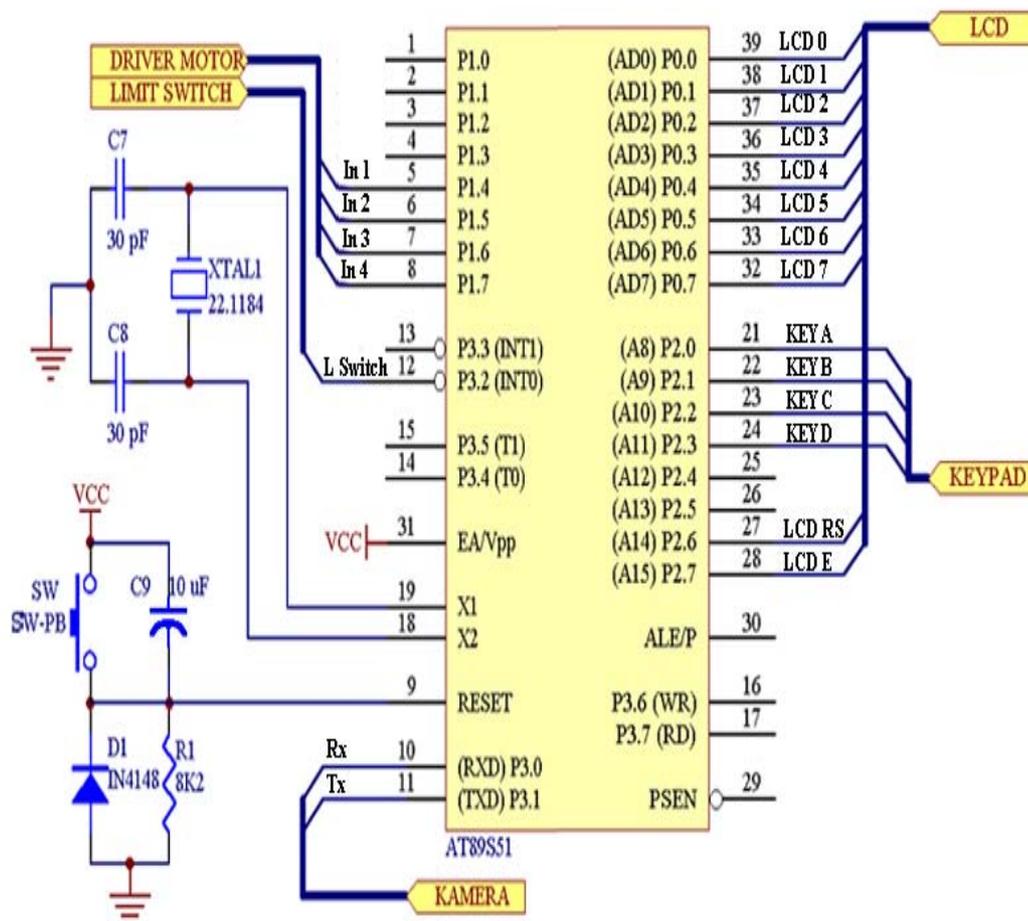
Gambar 3.7 Rangkaian Power On Reset

Dengan adanya rangkaian *reset* dan rangkaian osilator, maka syarat minimal agar sebuah mikrokontroler bekerja telah dipenuhi. Karena dengan adanya dua rangkaian ini, mikrokontroler AT89S51 telah dapat bekerja sendiri atau biasa diistilahkan *stand alone*.

Koneksi sistem dengan mikrokontroler dapat dilihat pada tabel 3.1 dan rangkaian mikrokontroler dapat dilihat pada gambar 3.8.

Tabel 3.1 Koneksi Sistem dengan Mikrokontroler

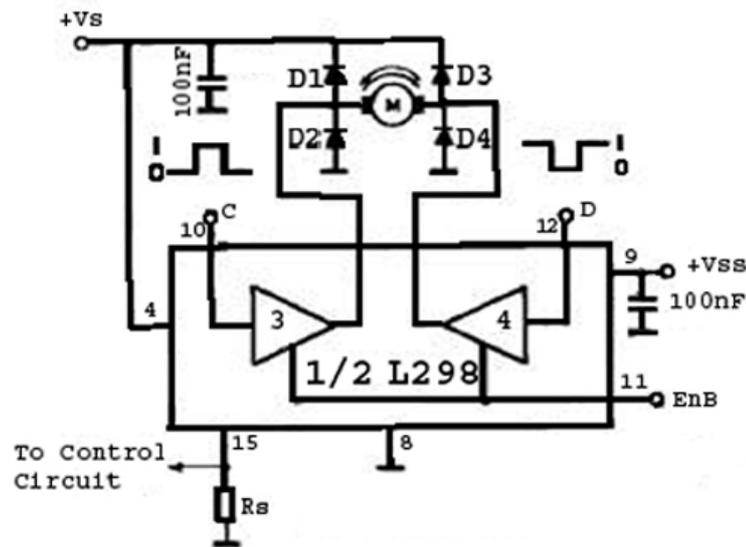
No.	Sistem	Mikrokontroler
1.	Driver Motor	P1.4, P1.5, P1.6, P1.7
2.	Kamera	P3.0, P3.1
3.	Keypad	P2.0, P2.1, P2.2, P2.3, P3.3
4.	Limit Switch	P3.0
5.	LCD	P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P.7, P2.6, P2.7



Gambar 3.8 Rangkaian Mikrokontroler

3.2.2 Driver Motor

Sebagai *driver* dari motor, digunakan IC L298. IC ini dipilih karena mudah dalam aplikasi serta dapat men-*supply* arus total sampai dengan 4 A. Kelebihan lain dari IC L298 ini adalah mampu menangani tegangan *supply* sampai dengan 46 V. Selain itu, IC ini dapat digunakan untuk mengontrol 2 motor sehingga untuk menjalankan *mobile robot*, hanya menggunakan 1 buah IC L298.



Inputs		Function
V = H	C=H ; D=L	Turn Right
	C=L ; D=H	Turn Left

Gambar 3.9 Rangkaian Driver Motor (secara *default*)

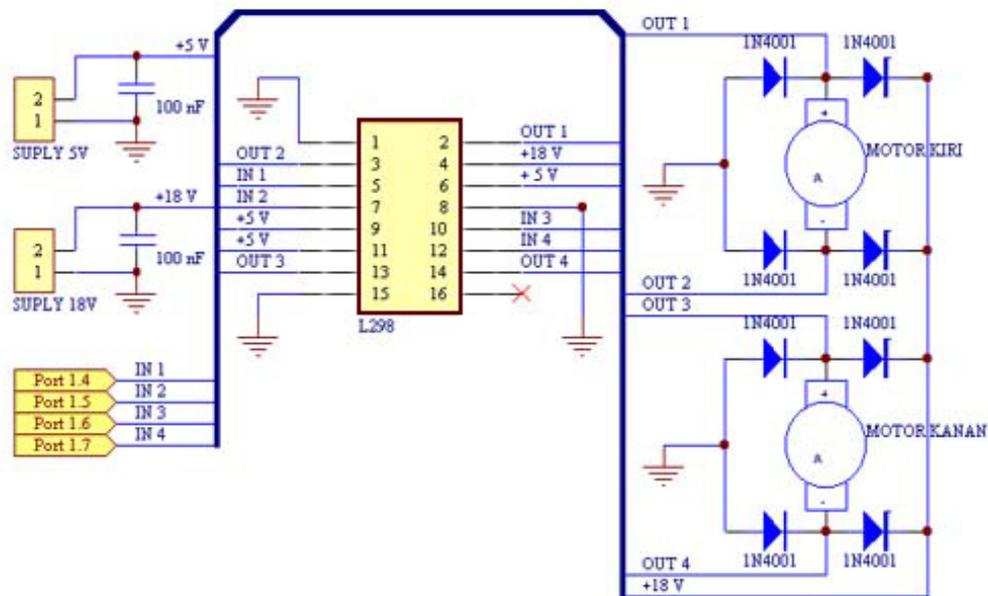
Sumber: SGS-Thomson Microelectronics. *Dual Full-Bridge Driver*. 2004. <<http://www.dialelec.com/477.html>>

Koneksi *driver* motor dengan Mikrokontroler dapat dilihat pada tabel 3.2.

Tabel 3.2 Koneksi Driver Motor dengan Mikrokontroler

No.	Input Driver Motor	Mikrokontroler
1.	Motor Kiri 1	P 1.4
2.	Motor Kiri 2	P 1.5
3.	Motor Kanan 1	P 1.6
4.	Motor Kanan 2	P 1.7

Mobile robot menggunakan motor *gearbox* +24 V_{DC} (arusnya 1,92 A) tetapi karena torsi yang dihasilkan terlalu kuat, maka dari percobaan diperoleh tegangan sebesar +18 V_{DC} sudah cukup untuk menggerakkan motor.



Gambar 3.10 Rangkaian Driver Motor Secara Keseluruhan

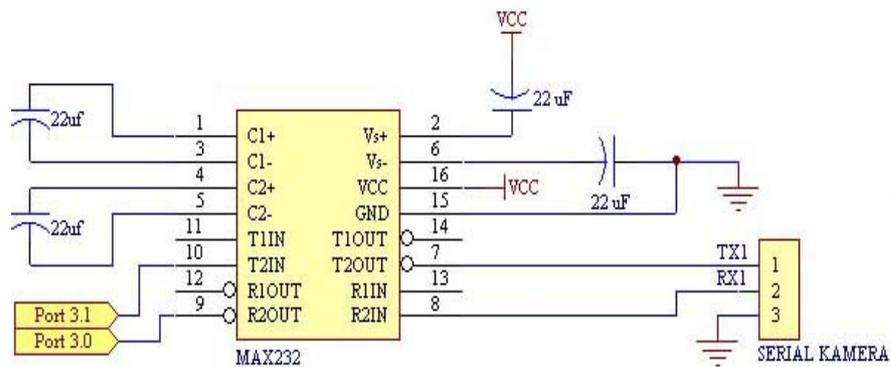
3.2.3 Kamera

Kamera digunakan dalam *mobile robot* untuk membedakan dan mendeteksi warna. Tipe kamera yang dipilih adalah CMUcam karena di dalam CMUcam sudah terdapat perintah-perintah yang memudahkan untuk melakukan *image processing* dan *track color*. Resolusi dari CMUcam adalah 80x143 pixels. Selain itu, komunikasi dari CMUcam berupa RS-232.

Karena komunikasi CMUcam berupa RS-232, maka sebelum mikrokontroler terhubung dengan CMUcam, AT89S51 harus dihubungkan dengan sebuah IC MAX-232 yang digunakan sebagai konverter *level* tegangan (gambar 3.11). Koneksi MAX-232 dengan mikrokontroler dapat dilihat pada tabel 3.3.

Tabel 3.3 Koneksi MAX-232 dengan Mikrokontroler

No.	MAX-232	Mikrokontroler
1.	Transmit In	P 3.1
2.	Receive Out	P 3.0

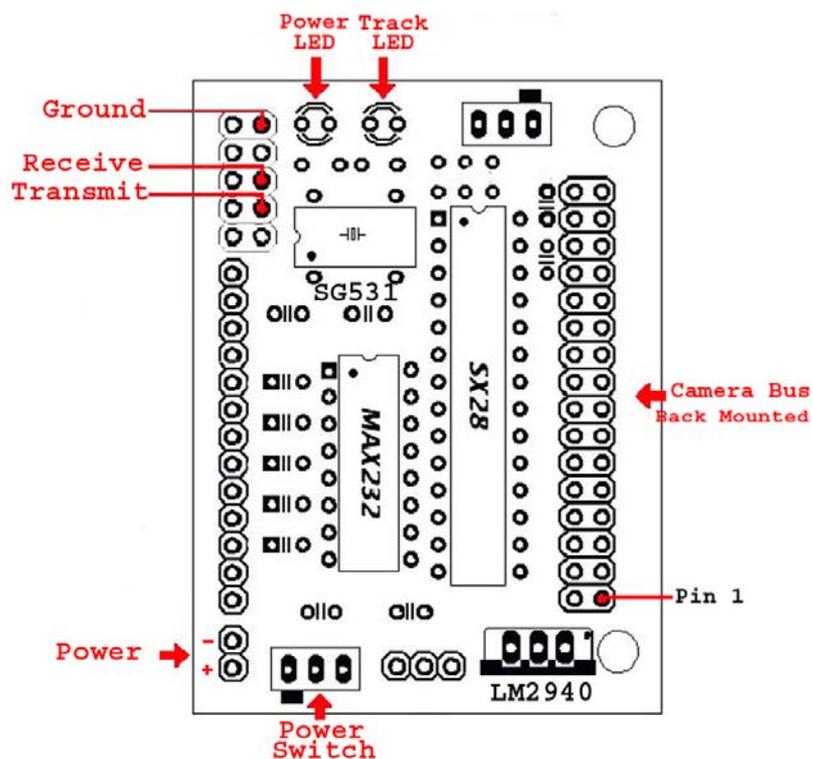


Gambar 3.11 Rangkaian Konverter Level Tegangan (MAX-232)

Data dari konverter *level* tegangan ini kemudian dihubungkan dengan RS-232 dari CMUcam. Koneksi MAX-232 dengan CMUcam dapat dilihat pada tabel 3.4.

Tabel 3.4 Koneksi MAX-232 dengan CMUcam

No.	MAX-232	CMUcam
1.	Transmit Out	Receive CMUcam
2.	Receive In	Transmit CMUcam

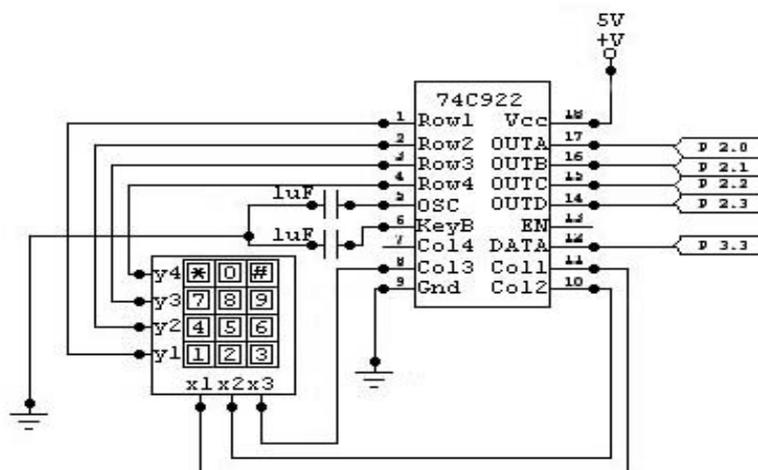


Gambar 3.12 Layout Board CMUcam

3.2.4 Keypad

Keypad digunakan sebagai *input*-an warna dari *mobile robot*. Sebelum terhubung dengan mikrokontroler, *keypad* terlebih dahulu dihubungkan IC 74C922 sebagai konverter *input* dari *keypad*. IC 74C922 digunakan untuk mengubah *input*-an dari *keypad* yang berupa matriks (*format x dan y*) menjadi 4-bit data dan 1-bit *available data*. *Available data* digunakan untuk mengetahui adanya penekanan tombol pada *keypad*. Selain *available data*, IC 74C922 ini juga memiliki kelebihan untuk mengatasi efek *bouncing*.

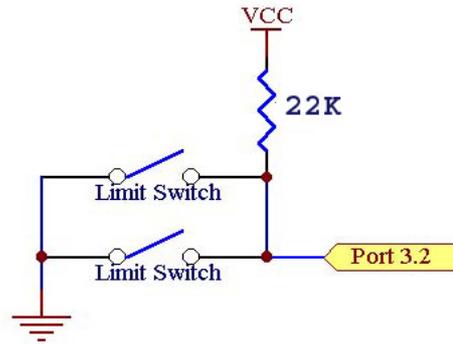
Rangkaian *keypad* secara keseluruhan dan koneksi dengan mikrokontroler dapat dilihat pada gambar 3.13.



Gambar 3.13 Rangkaian Keypad Dan Koneksi Dengan Mikrokontroler

3.2.5 Limit Switch

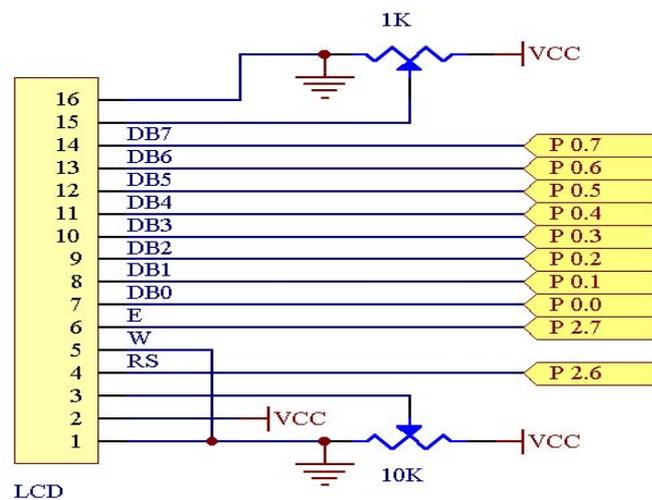
Limit Switch digunakan sebagai sensor untuk mendeteksi apakah ada penghalang pada jalur atau tidak. Jika ada penghalang, maka *mobile robot* akan berhenti dan menunggu hingga penghalang itu hilang. *Limit switch* ditempatkan 2 buah pada bagian depan dari *mobile robot* dan dirancang untuk aktif *low*, sehingga dalam keadaan normal akan mendapat nilai “1” (Gambar 3.14). *Limit switch* dihubungkan dengan resistor *pull-up* 22K Ω ke Vcc untuk mendapatkan logika “1” sebesar nilai Vcc. Efek *bouncing* yang terjadi pada *limit switch* ini akan dihilangkan dengan menambahkan *delay* pada *software*.



Gambar 3.14 Rangkaian Limit Switch Dan Koneksi Dengan Mikrokontroler

3.2.6 LCD

LCD 16 x 2 digunakan sebagai tampilan warna jalur yang dipilih dan tampilan perintah untuk menjalankan *mobile robot* serta keterangan-keterangan lainnya.

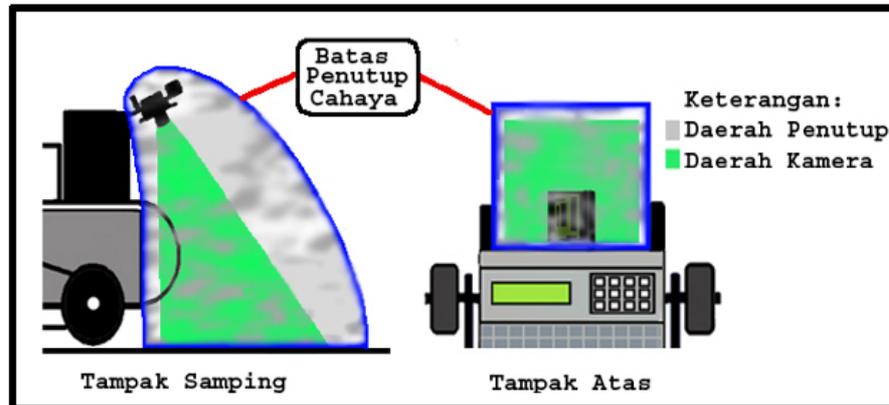


Gambar 3.15 Rangkaian LCD Dan Koneksi Dengan Mikrokontroler

3.2.6 Lampu Penerang

Dalam penerapannya, cahaya yang diterima oleh kamera akan berubah-ubah sesuai dengan pengaruh penerangan lingkungan. Ini menyebabkan nilai RGB dari warna yang dideteksi oleh kamera pun ikut berubah-ubah (penerangan matahari akan berbeda dengan penerangan lampu), maka akan sulit untuk menentukan nilai RGB minimum dan maksimum dari suatu warna.

Untuk itu, cahaya menuju kamera harus “distabilkan” terlebih dahulu agar dapat menentukan nilai RGB dari warna-warna jalur. Untuk “menstabilkan” cahaya yang diterima oleh kamera, dapat digunakan penutup yang berfungsi untuk “menahan” masuknya cahaya lingkungan (gambar 3.16) dan lampu penerang untuk memberi cahaya pada kamera.



Gambar 3.16 Kerudung Penahan Cahaya

Seperti yang dilihat pada gambar 3.16, daerah hijau merupakan daerah kamera sehingga lampu penerang digunakan untuk menerangi daerah tersebut. Lampu penerang yang dipakai berupa LED berwarna putih.

Diketahui bahwa sebuah LED ini membutuhkan tegangan *input* sebesar 3 Volt dan arus sekitar 25mA sehingga dengan tegangan *input* +18 Volt, LED dapat di-seri sebanyak 4 buah dan sebuah resistor yang menghasilkan arus sekitar 25mA.

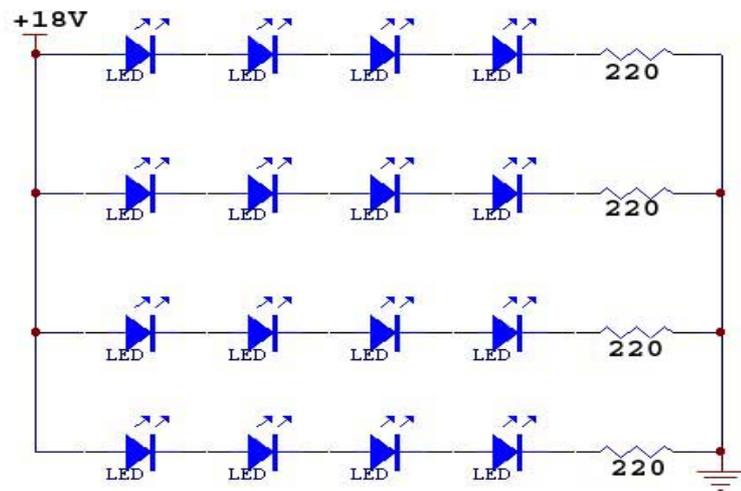
Perhitungannya adalah sebagai berikut:

$$\begin{aligned} V_{in} &= (4 \times V_{led}) + (R \times I) \\ 18 \text{ V} &= (4 \times 3 \text{ V}) + (R \times 25\text{mA}) \\ R &= (18 - 12) \text{ V} / 25\text{mA} \\ R &= 240 \Omega \end{aligned}$$

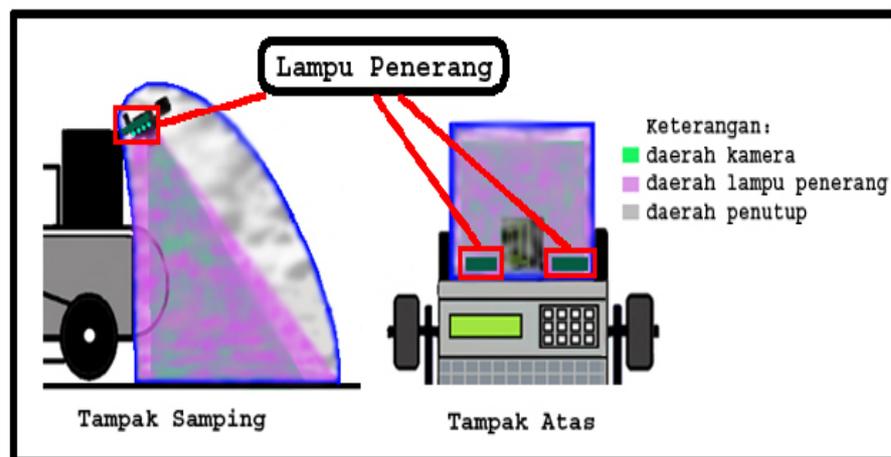
Maka diambil nilai resistornya adalah 220 Ω .

Dalam percobaan, LED sebanyak 16 buah cukup untuk menerangi daerah kamera. Gambar rangkaian lampu penerang dapat dilihat pada gambar 3.17. Selain itu, penempatan LED harus diatur agar cahaya dari LED tidak memantul kembali ke kamera. Jika cahaya dari LED memantul kembali ke kamera, akan menyebabkan kamera menjadi silau sehingga data yang dihasilkan tidak akurat.

Untuk itu, maka LED ini ditempatkan bersebelahan dengan kamera (gambar 3.18).



Gambar 3.17 Rangkaian Lampu Penerang



Gambar 3.18 Letak Lampu Penerang

3.3 Perencanaan Software

Mobile Robot ini menggunakan AT89S51 sebagai mikrokontrolernya sehingga *software* yang dipakai untuk memprogram adalah bahasa *assembly* MCS-51. Dalam penerapannya *mobile robot* akan mulai bekerja saat saklar *On/Off* diaktifkan. Setelah itu LCD akan menampilkan keterangan *input* warna dan mikrokontroler akan menunggu *input* dari *keypad*. Saat operator memberikan *input* warna melalui *keypad*, LCD akan menampilkan warna yang dipilih dan mikrokontroler akan menunggu *input*-an untuk menjalankan *mobile robot*.

Saat dijalankan, awalnya *mobile robot* akan berputar ditempat untuk mencari jalur yang di-*input*-kan oleh operator. Jika dalam 1 putaran *mobile robot* tidak menemukan jalur yang di-*input*-kan oleh operator, maka *mobile robot* akan berhenti dan menunggu *input*-an lagi. Pada saat bersamaan LCD akan menampilkan keterangan bahwa tidak ada jalur yang dimaksud.

Jika menemukan jalur yang di-*input*-kan, *mobile robot* akan mengikuti jalur tersebut. Pada ujung jalur, *mobile robot* akan berhenti dan menunggu *input*-an lagi. Saat tidak ada *input*-an, *mobile robot* tetap pada kondisi *stand-by*. Jika tombol *On/Off* dimatikan, maka *mobile robot* akan berhenti dan sistem akan mati.

Dalam men-*track* jalur, mikrokontroler akan memanfaatkan rutin yang tersedia pada kamera (CMUcam) yaitu rutin *track color*. Untuk menggunakan rutin ini dalam men-*track* suatu jalur, mikrokontroler harus mengirimkan perintah kepada kamera dengan *format*: *TC [Rmin Rmaks Gmin Gmaks Bmin Bmaks] \r*. “\r” disini adalah karakter ASCII dari *enter* yaitu 0D_H. Contoh, jika mikrokontroler akan men-*track* jalur dengan nilai Rmin = 15, Rmaks = 20, Gmin = 25 Gmaks = 30, Bmin = 35, Bmaks = 40, maka perintahnya adalah:

TC 15 20 25 30 35 40 \r

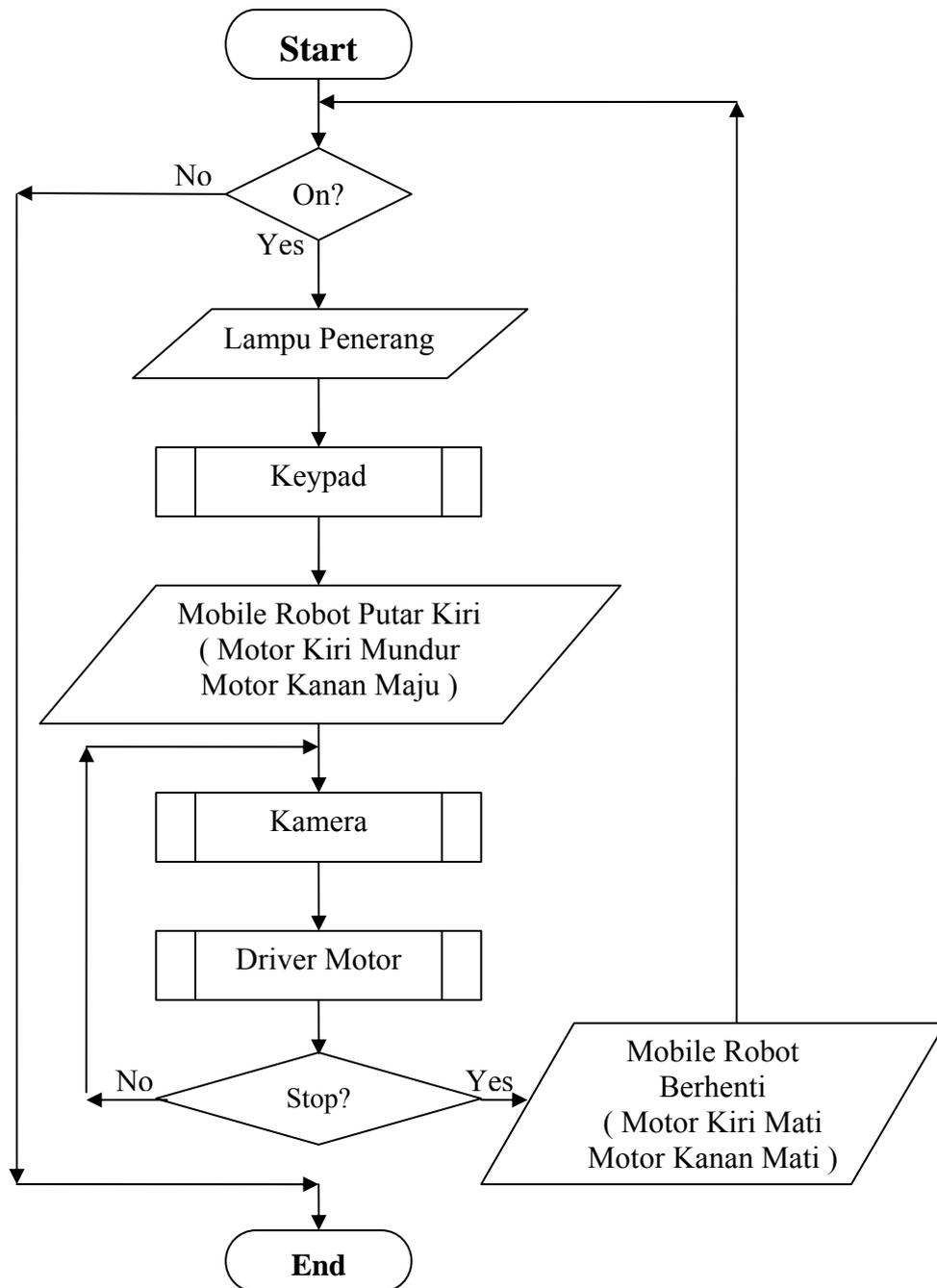
Perintah ini harus dikirimkan kamera berupa nilai *hexadecimal*, maka perintah yang harus dikirim ke kamera (dalam *hexadecimal*) adalah:

54 43 20 31 35 20 32 30 20 32 35 20 33 30 20 33 35 20 34 30 OD

Data dari kamera yang akan diterima oleh mikrokontroler akan ber-*format*: *M mx my x1 y1 x2 y2 pixels confidence \r*. *M* = awal data, *mx* = koordinat x dari titik berat, *my* = koordinat y dari titik berat, *x1* = nilai koordinat x minimum, *y1* = nilai koordinat y minimum, *x2* = nilai koordinat x maksimum, *y2* = nilai koordinat y maksimum, *pixels* = jumlah *pixels* daerah yang merupakan warna yang di-*track*, *confidence* = jumlah *pixels* warna yang di-*track*. Contoh data yang diterima adalah:

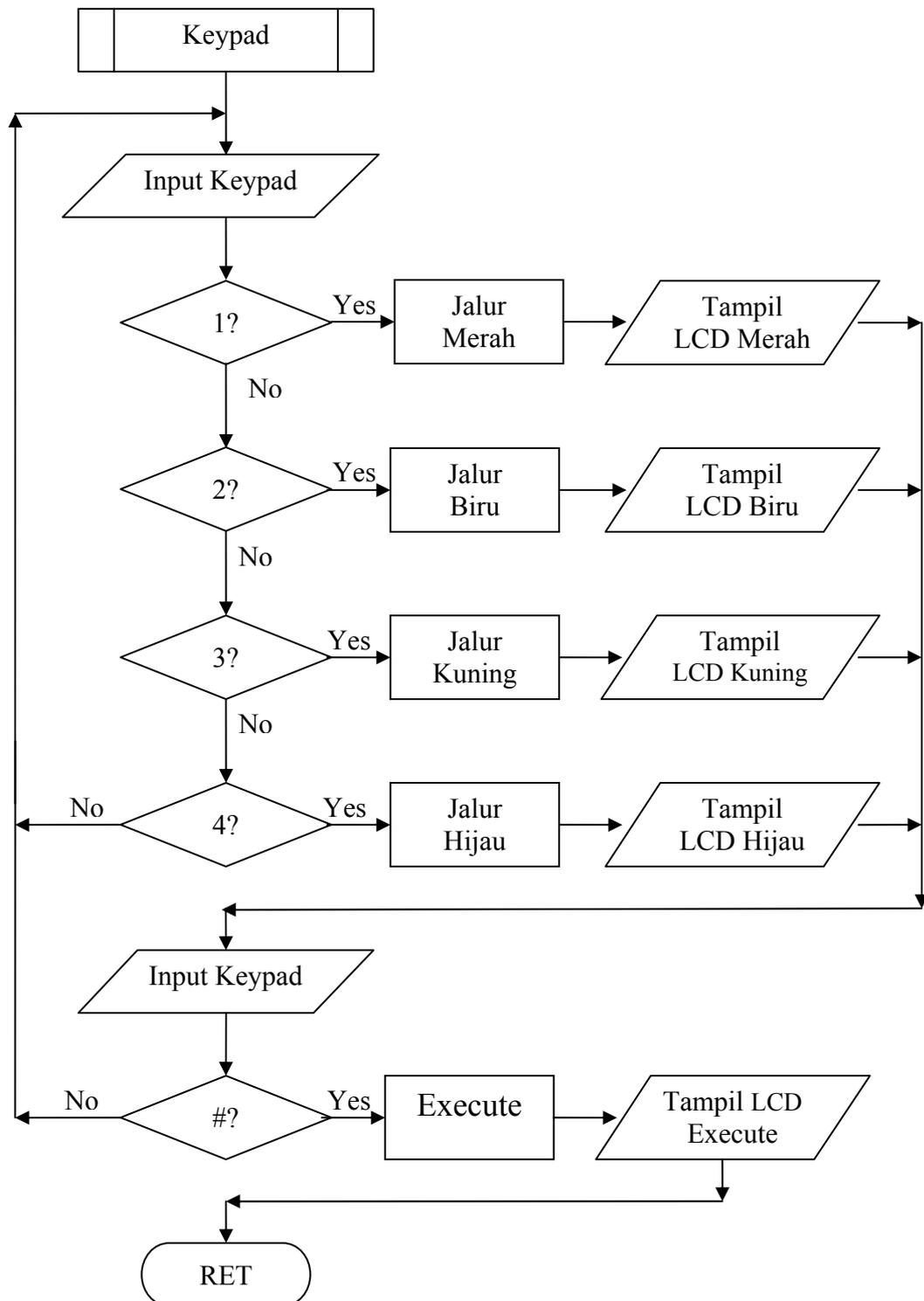
M 40 72 1 1 80 143 35 98

Secara umum *software* yang akan dibuat akan digambarkan pada *algoritma software* pada gambar 3.19.



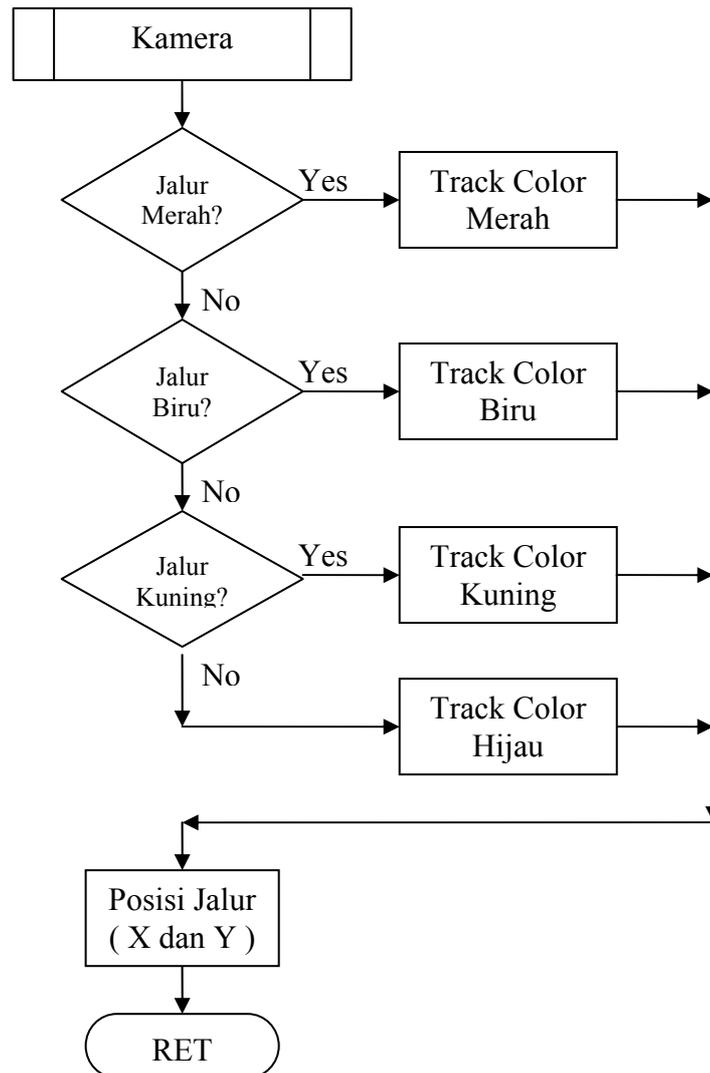
Gambar 3.19 Algoritma Software Mobile Robot

Pada gambar 3.20 dapat dilihat *algoritma input-an* dari *keypad* serta tampilan pada LCD. Sebagai *input mobile robot*, tombol *keypad* yang digunakan hanya 5 buah yaitu tombol 1 sebagai *input* warna merah, tombol 2 sebagai *input* warna biru, tombol 3 sebagai *input* warna kuning, tombol 4 sebagai *input* warna hijau, tombol # sebagai *input execute* untuk *mobile robot*.



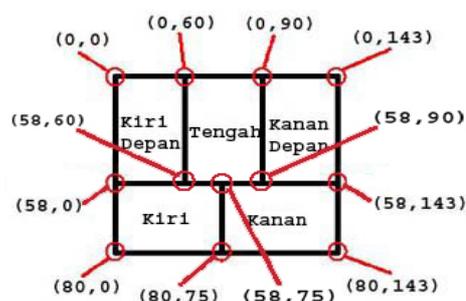
Gambar 3.20 Algoritma Input Keypad

Gambar 3.21 merupakan *algoritma* kamera. Data yang diterima dari kamera berupa posisi jalur yang berupa koordinat x dan y pada bagian kiri atas dan kanan bawah serta posisi titik berat dari jalur tersebut (berupa x dan y).

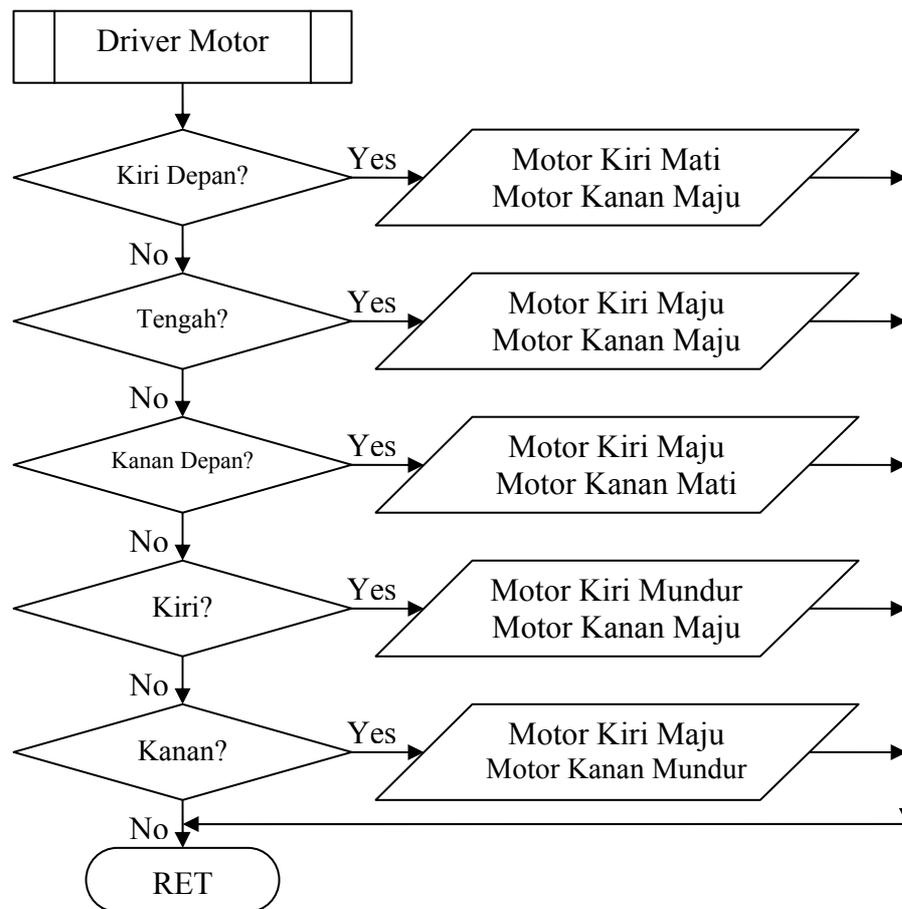


Gambar 3.21 Algoritma Software Kamera

Setelah menerima data koordinat titik berat dari kamera kemudian mikrokontroler menentukan gerakan dari *mobile robot*. Dalam menentukan gerakan, mikrokontroler membagi menjadi 5 daerah (gambar 3.22). Gambar 3.23 adalah *algoritma driver motor*.



Gambar 3.22 Pembagian Daerah Untuk Menentukan Gerakan



Gambar 3.23 Algoritma Driver Motor