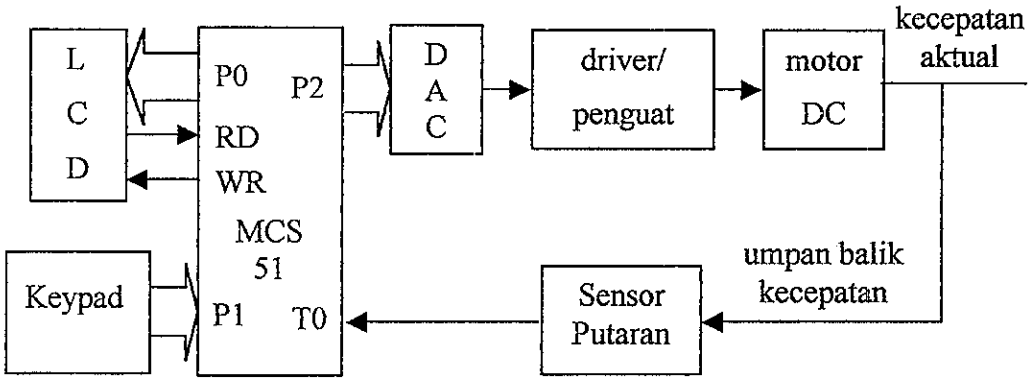
BAB III

PERANCANGAN DAN REALISASI

3.1 Diagram Blok Rancangan

Berikut ini diagram blok alat pengatur kecepatan motor DC hasil realisasi yang diperlihatkan pada gambar 3.1.



Gambar 3.1 Diagram blok alat pengatur kecepatan motor DC.

Prinsip kerja dari diagram blok rancangan tersebut adalah sebagai berikut :

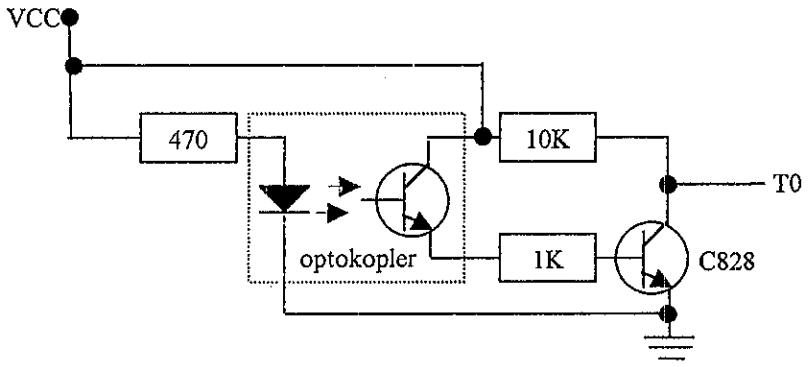
Ketika *setpoint* atau kecepatan acuan dimasukkan dari *keypad* dan ditampilkan ke LCD, mikrokontroler akan mengirim sinyal ke penguat melalui DAC untuk menggerakkan motor DC dan piringan berlubang. Jumlah pulsa yang masuk dari sensor putaran akan dibaca lewat kaki T0 sebagai kecepatan aktual. Selisih kedua kecepatan ini digunakan sebagai faktor koreksi dari fungsi kontrol PID dalam program pengaturan kecepatan. Mikrokontroler akan melakukan pengendalian sampai motor DC berputar sesuai dengan kecepatan acuan dan kecepatan terukur ditampilkan ke LCD.

3.2 Sensor Putaran

3.2.1 Rangkaian Sensor Putaran

Komponen yang digunakan sebagai sensor pada rangkaian ini adalah optokopler yang berfungsi mengindera jumlah putaran dalam bentuk pulsa tiap satuan waktu. Optokopler ini terdiri dari sebuah LED dan sebuah foto-transistor dalam satu kemasan. Diantara LED dan foto-transistor terdapat piringan berlubang yang dipasang pada poros motor.

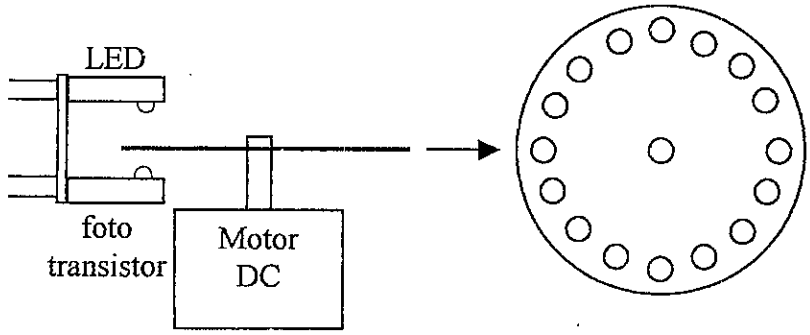
Ketika piringan dalam keadaan terbuka, cahaya infra merah dari LED akan diterima oleh foto-transistor. Ini berakibat sambungan pada foto-transistor terhubung dan seakan bersifat seperti saklar yang tertutup. Saat piringan berputar maka akan mengakibatkan terpotongnya cahaya dari LED. Cahaya LED yang terputus-putus oleh gerak piringan akan berakibat foto-transistor terkondisi hidup dan mati yang membentuk gelombang kotak. Pulsa dari foto-transistor sangat kecil sehingga perlu diperkuat dengan penambahan transistor C828. Keluarannya dihubungkan dengan kaki T0 mikrokontroler. Adapun rangkaian sensor putaran dari motor DC ditunjukkan dalam gambar 3.2.



Gambar 3.2 Rangkaian sensor putaran

3.2.2 Sistem Mekanik Sensor

Bagian mekanik dari sensor berupa piringan berlubang yang berfungsi meneruskan dan menahan cahaya LED ke foto-transistor pada optokopler. Piringan ini terbuat dari PCB dengan diameter 6 cm, jumlah lubang 16 dan jarak antar lubang 0,65 cm. Sistem mekanik dari sensor putaran motor DC diperlihatkan pada gambar 3.3.



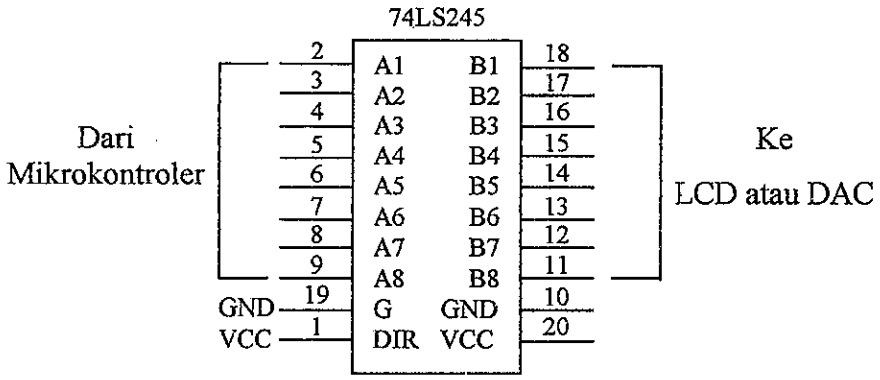
Gambar 3.3 Sistem mekanik sensor

3.3 Penyangga

Penyangga merupakan piranti yang menghasilkan keluaran data *digital* sama dengan masukannya. Penyangga 74LS245 digunakan untuk menghubungkan antara mikrokontroler dengan LCD maupun dengan DAC. Penyangga ini berfungsi untuk memperkuat sinyal keluaran dari mikrokontroler yang terlalu kecil dan sebagai penyangga rangkaian yang dibebani, agar keluaran rangkaian tidak terganggu oleh efek pembebanan dari rangkaian lain yang dikendalikan.

IC 74LS245 merupakan penyangga yang dapat mengirim dan menerima (*transceiver*) 8 saluran data dengan 3 status keluaran dan dikemas dalam kemasan *DIP* 20 pena. Konfigurasi pena 74LS245 dapat dilihat pada gambar 3.4.

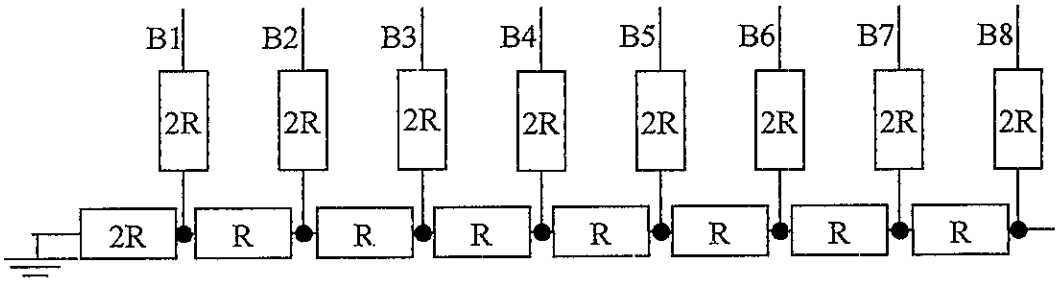
Penyangga ini digunakan untuk menyalurkan data dari A ke *bus* B karena kontrol arah (*DIR*) diberi logika tinggi sedangkan masukan enabel G diberi logika rendah.



Gambar 3.4 Konfigurasi pena 74LS245

3.4 Rangkaian DAC

Keluaran dari mikrokontroler masih berupa sinyal *digital* yang harus diubah menjadi sinyal *analog*, untuk itu diperlukan rangkaian *konverter D/A*. Adapun rangkaian DAC yang digunakan adalah metode tangga *biner* atau R-2R yang terdiri atas rangkaian resistor dengan nilai 10 KΩ dan 20 KΩ, yang mengubah setiap *bit* sinyal masukan menjadi arus yang sesuai. Jumlah masing-masing arus tersebut mengalir ke resistor yang mengubahnya menjadi tegangan yang sesuai dan keluarannya dihubungkan ke Op-Amp. Berikut adalah rangkaian DAC R-2R.

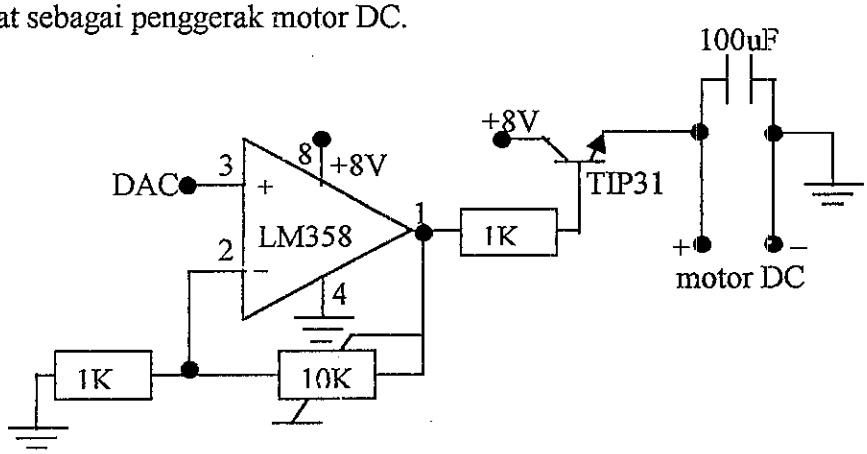


Gambar 3.5 Rangkaian DAC.

3.5 Rangkaian Penguat

Tegangan keluaran *DAC* yang terlalu kecil harus dikuatkan dengan penguat tegangan LM358 sekitar 1,05 kali. Hal ini diperlukan karena tegangan keluaran maksimum *konverter D/A* kadang-kadang untuk kecepatan maksimum motor tertentu tidak mencukupinya. Dalam hal ini digunakan LM358 yang dioperasikan secara tak membalik karena penguat ini tidak membutuhkan tegangan catu negatif, tetapi hanya menggunakan catu daya positif saja. Dengan menggunakan penguat operasional, maka faktor penguatan dapat diatur dan rentang tegangan keluaran dapat diperlebar.

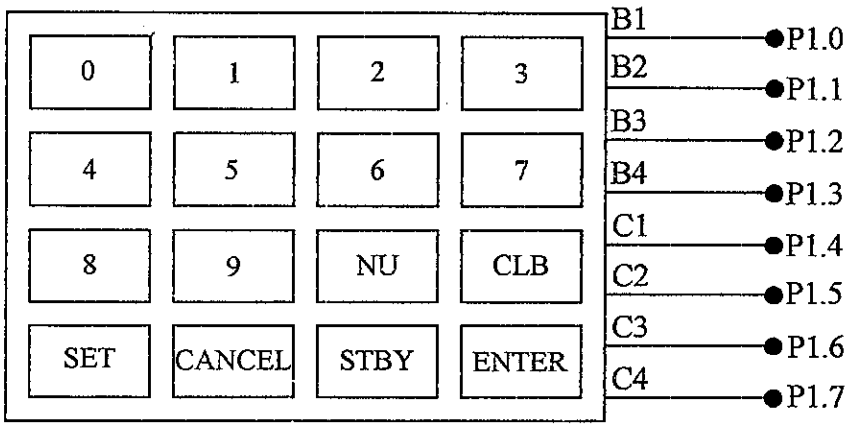
Karena arus keluaran penguat operasional kecil, hal ini belum cukup untuk menggerakkan motor DC, sehingga diperlukan penguat arus dengan transistor TIP31. Penyesuaian ini sangat perlu karena percikan api motor dapat menyebabkan gangguan yang mencapai *konverter D/A* dan dapat menyebabkan kesalahan. Sebagian gangguan ini sudah dinetralisasi oleh kondensator yang dipasang paralel dengan motor DC tipe BDS38. Gambar 3.6 merupakan rangkaian penguat sebagai penggerak motor DC.



Gambar 3.6 Rangkaian penguat.

3.6 Keypad

Keypad merupakan papan ketik yang digunakan untuk memberikan masukan kecepatan acuan dan fungsi perintah lainnya ke layar LCD. *Keypad* ada beberapa jenis, tetapi yang digunakan disini mempunyai struktur 4 baris dan 4 kolom. *Keypad* ini dihubungkan ke *Port* 1 dari mikrokontroler, seperti yang ditunjukkan pada gambar 3.7.



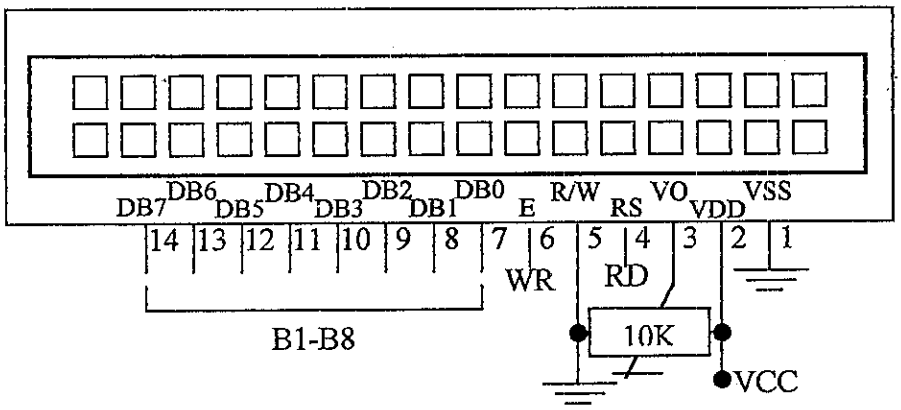
Gambar 3.7 Keypad.

Keypad ini terdiri dari 16 tombol yang fungsinya adalah sebagai berikut:

- Angka 0 sampai 9, sebagai nilai masukan
- NU (not use), tidak digunakan
- CLB (calib), untuk kalibrasi dengan memasukkan nilai DAC
- SET (setting), untuk masukan setpoint kecepatan acuan
- CANCEL, untuk membatalkan/mengganti nilai *setpoint*
- STBY (standby), untuk menghentikan putaran motor DC
- ENTER, untuk melanjutkan ke instruksi berikutnya

3.7 LCD

LCD merupakan bentuk tampilan keluaran berbentuk seperti 7 segmen namun memiliki keunggulan dengan tampilan dotmatriknya. Pada perancangan ini digunakan LCD tipe STN yang terdiri dari 16 karakter dengan 2 baris tampilan kristal cair dan 5x8 matrik titik. LCD ini memiliki ROM pembangkit karakter untuk 192 tipe karakter, RAM pembangkit karakter dan RAM tampilan yang dapat dibaca dari unit mikroprosesor. Berikut ini merupakan gambar dari penyemat LCD tersebut.

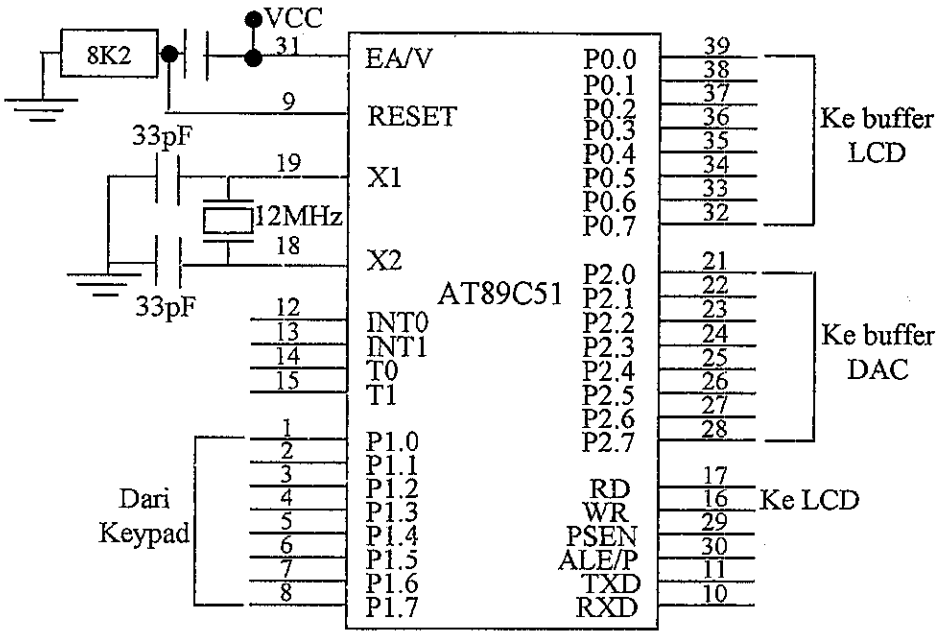


Gambar 3.9 Penyemat LCD.

3.8 Sistem Minimum

Dalam perancangan ini digunakan mikrokontroler AT89C51 buatan Atmel yang merupakan keluarga MCS-51. Pada AT89C51 sudah terdapat EPROM di dalamnya sehingga tidak memerlukan EPROM luar untuk menyimpan programnya. Untuk perancangannya *Port* 1 digunakan sebagai masukan dari *keypad*, *Port* 0 sebagai keluaran ke LCD, dan *Port* 2 sebagai keluaran ke DAC.

Pena X1 dan X2 dihubungkan ke rangkaian osilator eksternal dengan osilator kristal sebesar 12 MHz. Berikut ini gambar sistem minimum AT89C51 tersebut.



Gambar 3.10 Sistem minimum AT89C51.

3.9 Perangkat Lunak

3.9.1 Pengukuran Kecepatan

Untuk mengukur kecepatan putar digunakan sensor optokopler dan piringan berlubang yang menghasilkan 16 pulsa tiap putaran. Mikrokontroler akan menghitung banyaknya pulsa yang dihasilkan selama 50000 μS. Sehingga diperoleh perumusan program dalam menghitung jumlah putaran permenit, yaitu:

$$RPM = \frac{JML_TH0TL0}{16} X \frac{60}{50000\mu s} = JML_TH0TL0X75, \qquad (3.1)$$

$$RPM = JML_TH0TL0X4B \text{ Heksadesimal.} \qquad (3.2)$$

dengan JML_TH0TL0 adalah jumlah pulsa yang masuk / diterima oleh T0.

3.9.2 Pengaturan Kecepatan

Karena kecepatan ini tidak lagi diambil nilainya secara kontinyu tetapi setiap jangka waktu tertentu dan tidak lagi dikoreksi secara kontinyu, maka komponen integral dan diferensial dalam persamaan dapat digantikan dengan jumlah dan selisih. Sehingga diperoleh persamaan untuk menghitung kecepatan dengan kontrol PID sebagai algoritma kendalinya, yaitu:

$$Y_k = K_p[\ e_k + \frac{T}{T_i}\sum_{i=0}^{k} e_i + T_d \frac{e_k - e_i}{T}\ ] + y_0, \qquad (3.3)$$

dengan Y_k adalah kecepatan terukur, K_p adalah penguatan proporsional, e_k adalah simpangan pengaturan,T adalah periode sampling, T_i adalah waktu integral, T_d adalah waktu diferensial dan y_0 adalah kecepatan awal.

Parameter y_0 ditentukan dengan mengoperasikan motor DC tanpa dilakukan pengaturan kecepatan. Dalam hal ini dengan memberikan variasi masukan nilai DAC dan mengukur besar RPM yang terbaca, sehingga diperoleh data berikut:

Tabel 3.1 Penentuan parameter y_0.

DAC (x)	60	100	140	180	200	240	255
RPM (y)	525	1725	2850	3900	4425	5400	5700

Dari data pada tabel 3.1 tersebut dapat diperoleh persamaan garis lurus, yaitu:

$$RPM = 26.DAC - 840, \qquad (3.4)$$

Jika RPM diganti dengan SETPOINT (kecepatan acuan dalam RPM) akan diperoleh parameter y_0 sebagai kecepatan awal dalam program, yaitu :

$$DAC = [SETPOINT + 840]/26, \qquad (3.5)$$

atau $DAC = [SETPOINT + 348]/1A$ Heksadesimal, (3.6)

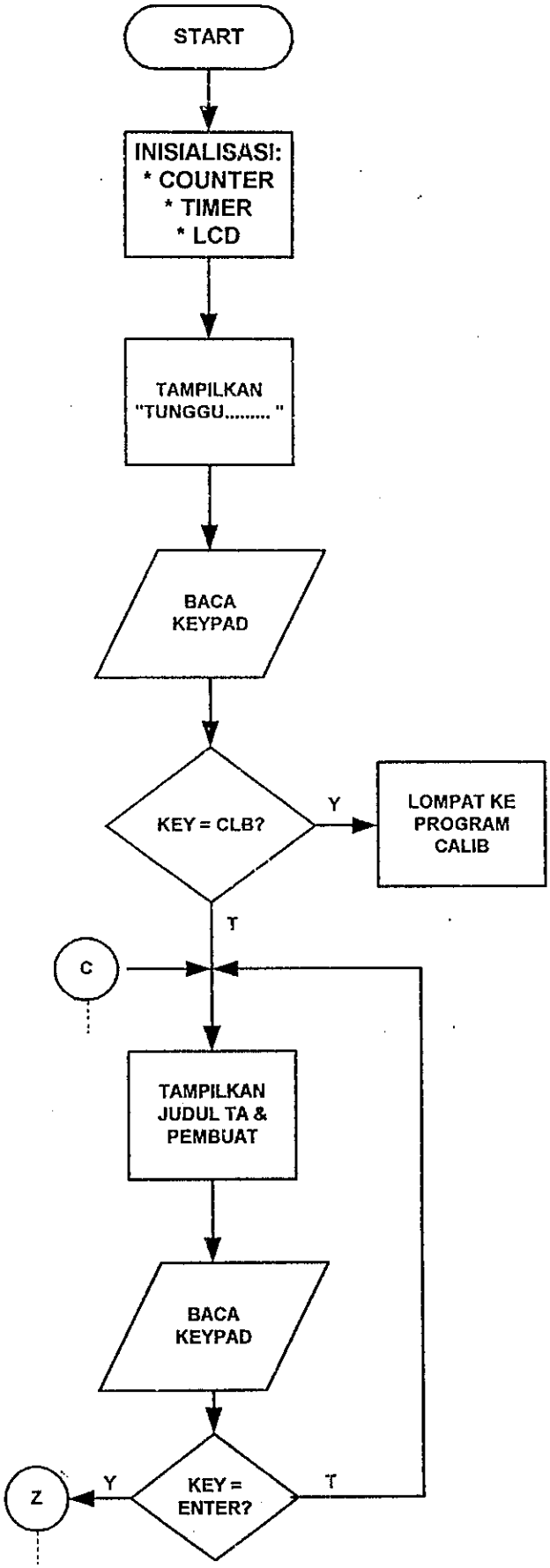
Pertama-tama motor DC dioperasikan sebagai pengatur P, dengan menghilangkan bagian integral dan diferensial. Kemudian penguatan K_p dinaikkan terus hingga motor mulai berputar secara periodik. Pada keadaan ini nilai K_p disebut K_{cr} dan diperoleh nilai penguatan K_{cr} = 2. Dari aturan penyepadanan Ziegler-Nichols metode kedua, bahwa K_p= 0,5 x K_{cr} , sehingga diperoleh penguatan K_p = 1. Karena pada pengatur P sudah diperoleh proses yang optimal dari percobaan dengan kecepatan maksimal 5700 RPM, maka pada pengatur PID hal ini hampir tidak mungkin sehingga tidak dilakukan. Sehingga diperoleh persamaan dalam program pengaturan kecepatan motor DC menggunakan kontrol proporsional sebagai algoritma kendalinya, yaitu:

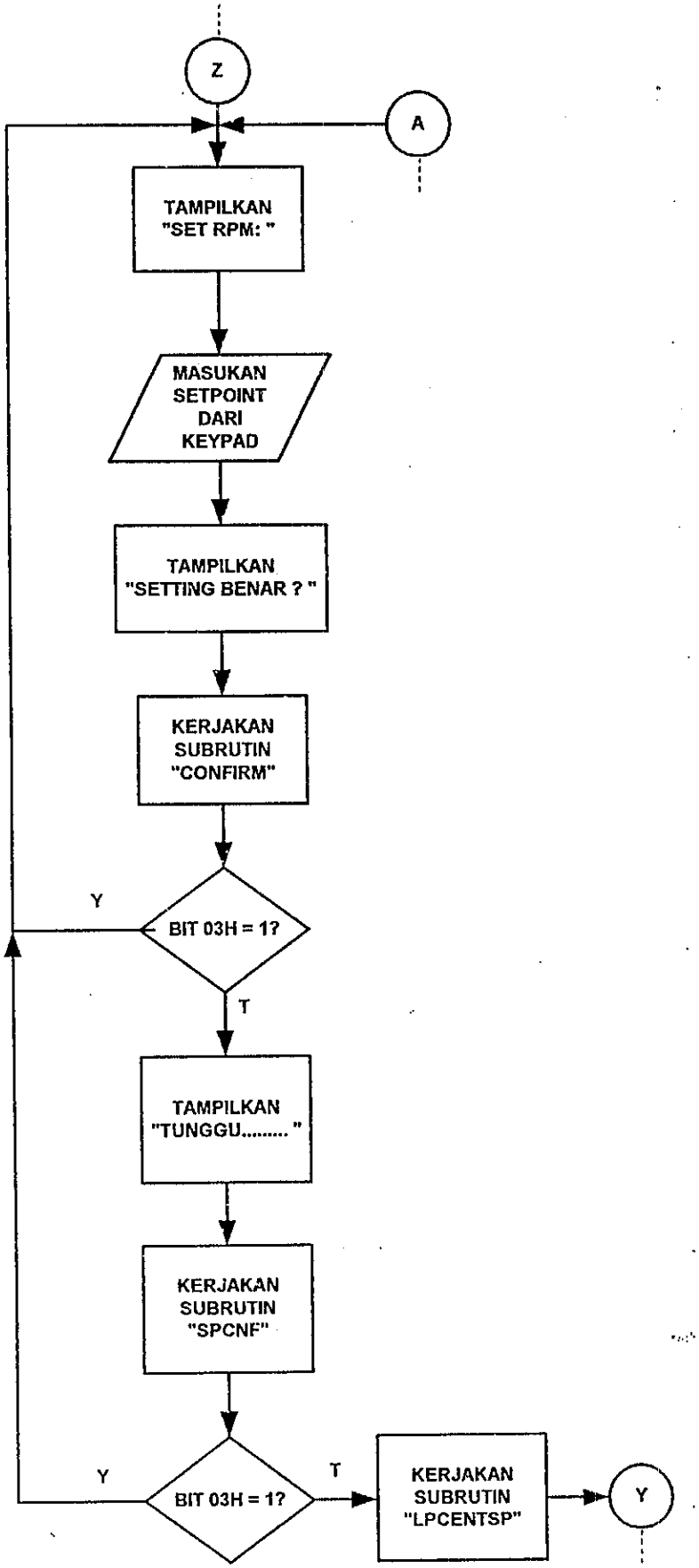
$$Y_k = e_k + \{[SETPOINT + 348]/1A\} \ . \qquad (3.7)$$

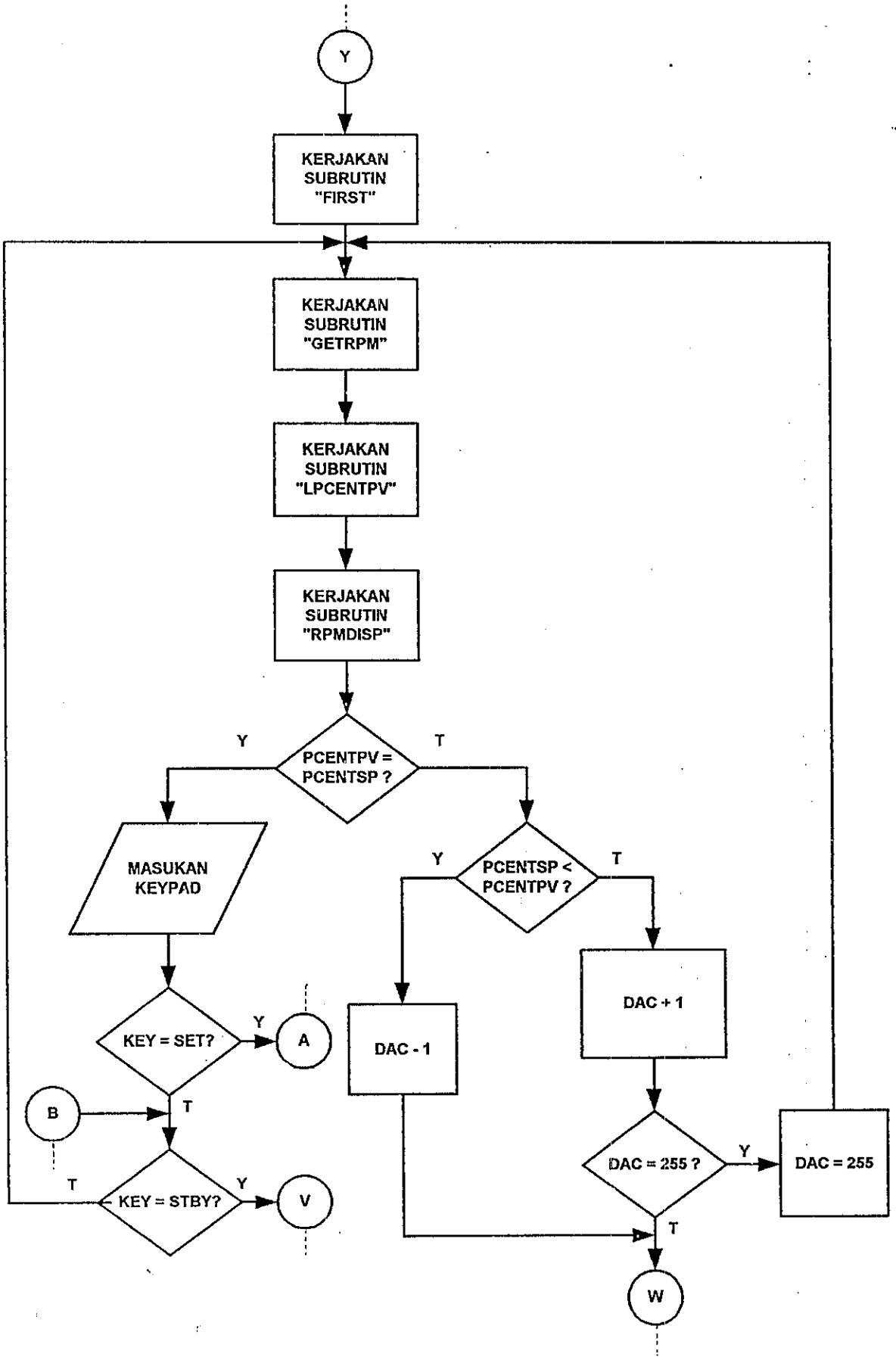
Ketika *setpoint* dimasukkan motor akan berputar dengan nilai DAC dan kecepatan aktual dibaca dan dibandingkan dengan *setpoint* sebagai simpangan pengaturan. Nilai DAC ini akan dinaikkan atau dikurangi sampai simpangan pengaturan mendekati nol. Sehingga motor DC akan berputar disekitar setpoint dan kecepatan terukur ditampilkan ke LCD dalam RPM setelah dikonversi.

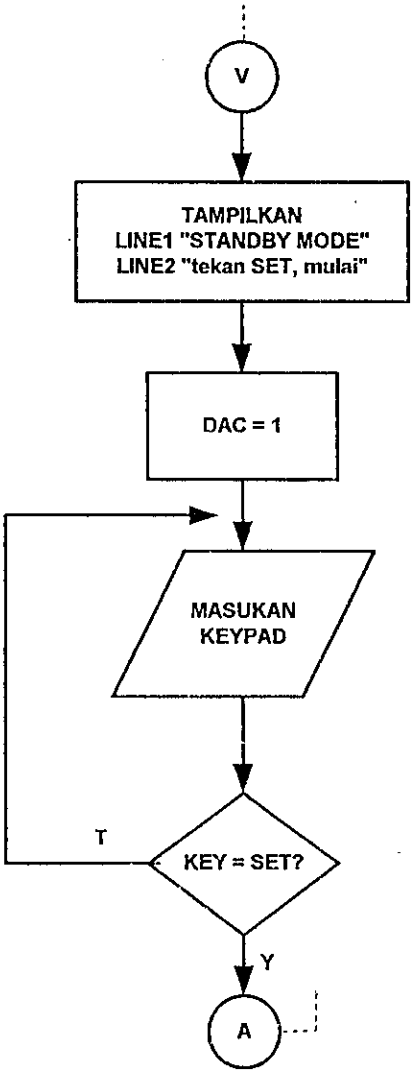
3.9.3 Diagram Alir

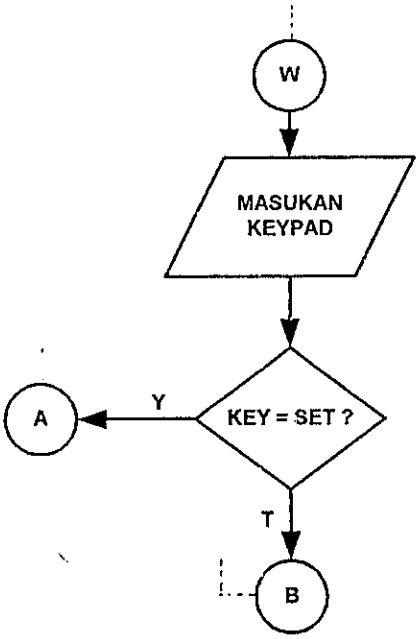
Diagram alir ini merupakan urutan kerja dari program pengaturan kecepatan motor DC dengan mikrokontroler MCS-51 menggunakan kontrol proporsional sebagai algoritma kendali. Dalam diagram alir ini meliputi pengukuran kecepatan, pengaturan kecepatan dan menampilkannya pada layar LCD. Berikut ini adalah algoritma program pengaturan kecepatan motor DC secara keseluruhan.

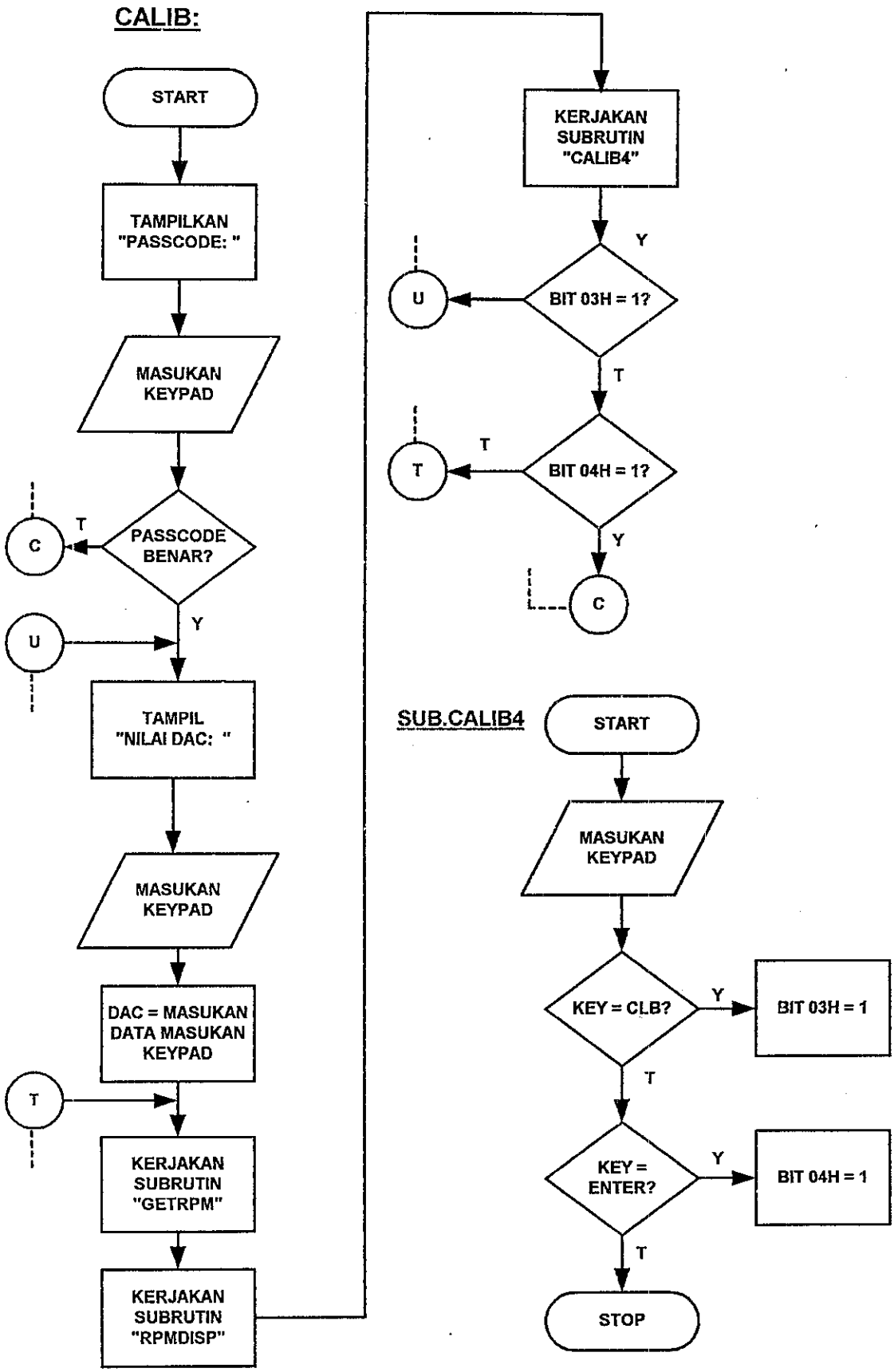

START
INISIALISASI:
* COUNTER
* TIMER
* LCD
TAMPILKAN
"TUNGGU..... "
BACA
KEYPAD
KEY = CLB?
Y
LOMPAT KE
PROGRAM
CALIB
T
C
TAMPILKAN
JUDUL TA &
PEMBUAT
BACA
KEYPAD
Z
Y
KEY =
ENTER?
T


Z
A
TAMPILKAN "SET RPM: "
MASUKAN SETPOINT DARI KEYPAD
TAMPILKAN "SETTING BENAR ? "
KERJAKAN SUBRUTIN "CONFIRM"
Y
BIT 03H = 1?
T
TAMPILKAN "TUNGGU..... "
KERJAKAN SUBRUTIN "SPCNF"
Y
BIT 03H = 1?
T
KERJAKAN SUBRUTIN "LPCENTSP"
Y

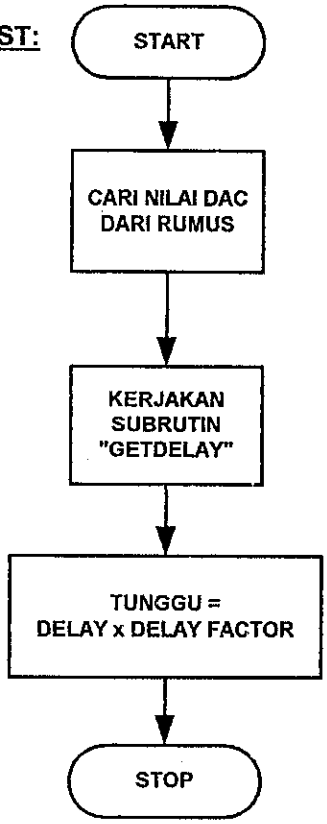

Y
KERJAKAN SUBRUTIN "FIRST"
KERJAKAN SUBRUTIN "GETRPM"
KERJAKAN SUBRUTIN "LPCENTPV"
KERJAKAN SUBRUTIN "RPMDISP"
PCENTPV = PCENTSP ?
Y
T
MASUKAN KEYPAD
PCENTSP < PCENTPV ?
Y
T
KEY = SET?
Y
A
DAC - 1
DAC + 1
B
T
DAC = 255 ?
Y
DAC = 255
T
KEY = STBY?
Y
V
T
W


V
TAMPILKAN
LINE1 "STANDBY MODE"
LINE2 "tekan SET, mulai"
DAC = 1
MASUKAN
KEYPAD
T
KEY = SET?
Y
A

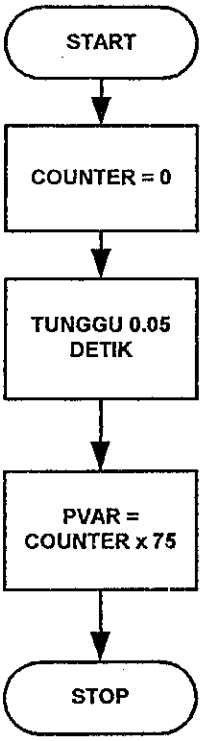

W
MASUKAN
KEYPAD
A
Y
KEY = SET ?
T
B


CALIB:
START
TAMPILKAN "PASSCODE: "
MASUKAN KEYPAD
PASSCODE BENAR?
T
C
Y
U
TAMPIL "NILAI DAC: "
MASUKAN KEYPAD
DAC = MASUKAN DATA MASUKAN KEYPAD
T
KERJAKAN SUBRUTIN "GETRPM"
KERJAKAN SUBRUTIN "RPMDISP"
KERJAKAN SUBRUTIN "CALIB4"
Y
U
BIT 03H = 1?
T
T
BIT 04H = 1?
T
Y
C
SUB.CALIB4
START
MASUKAN KEYPAD
KEY = CLB?
Y
BIT 03H = 1
T
KEY = ENTER?
Y
BIT 04H = 1
T
STOP

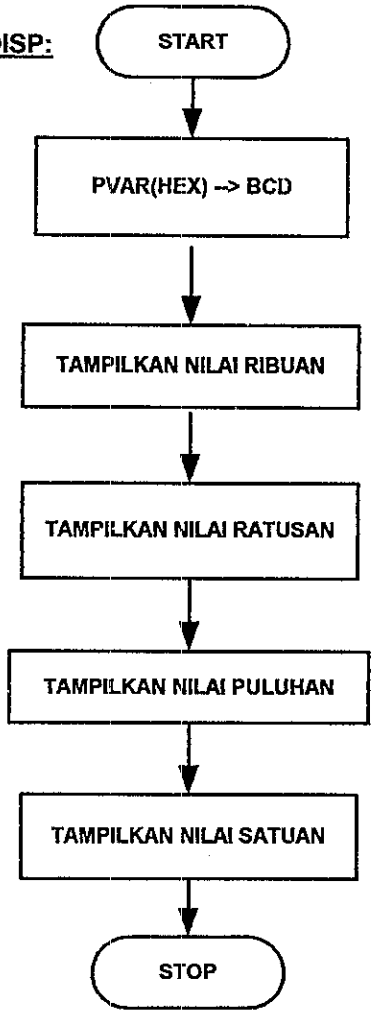
SUB.FIRST:



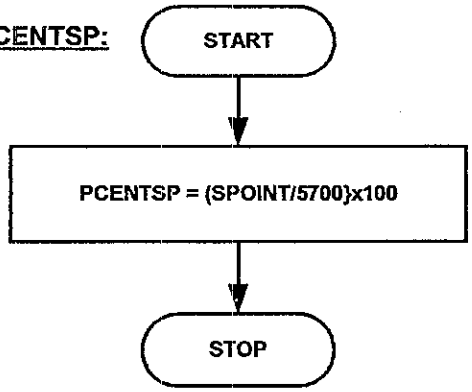
SUB.GETRPM:



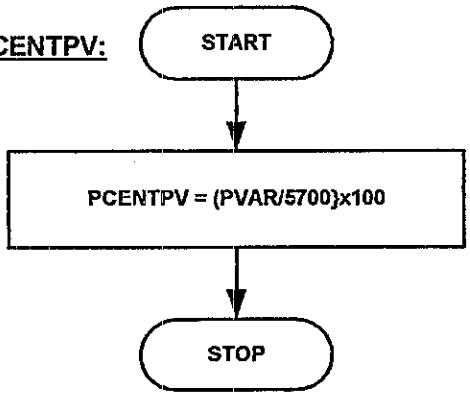
SUB. RPMDISP:

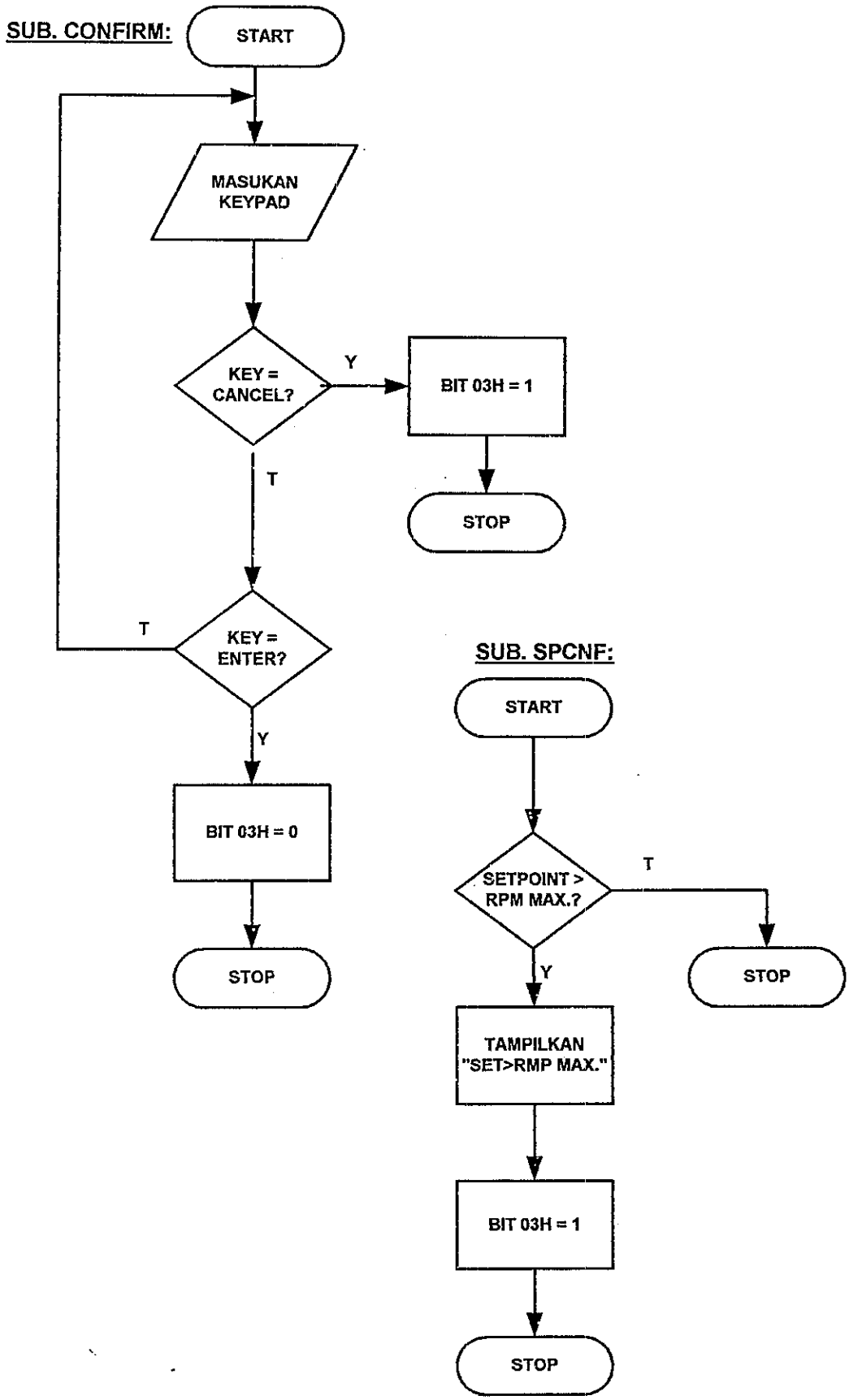


SUB. LPCENTSP:

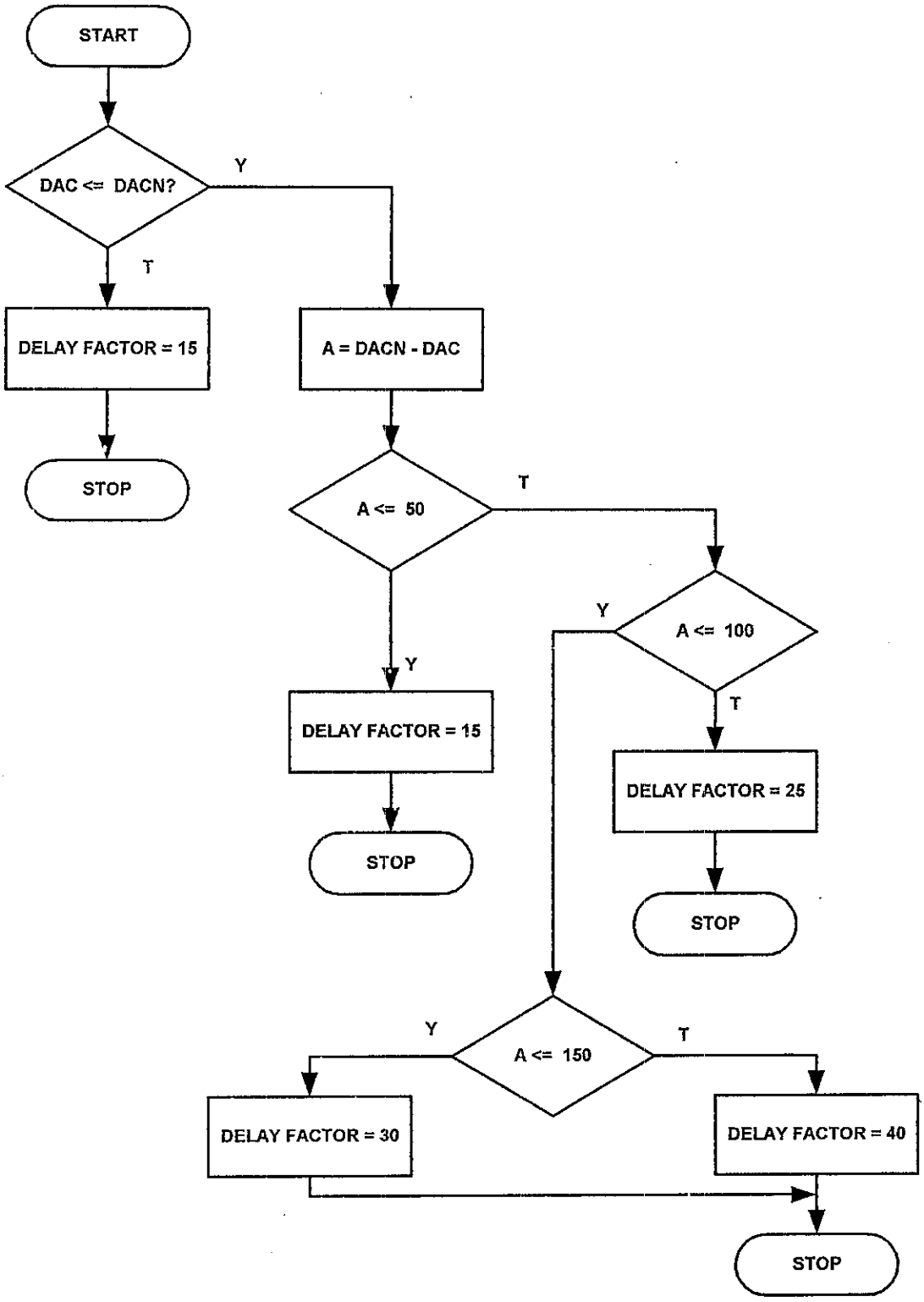


SUB. LPCENTPV:




SUB. CONFIRM:
START
MASUKAN KEYPAD
KEY = CANCEL?
Y
BIT 03H = 1
STOP
T
KEY = ENTER?
T
Y
BIT 03H = 0
STOP
SUB. SPCNF:
START
SETPOINT > RPM MAX.?
T
STOP
Y
TAMPILKAN "SET>RMP MAX."
BIT 03H = 1
STOP

SUB. GETDELAY:



Gambar 3.11 Diagram alir program pengaturan kecepatan motor DC.