BAB III
PERANCANGAN SISTEM

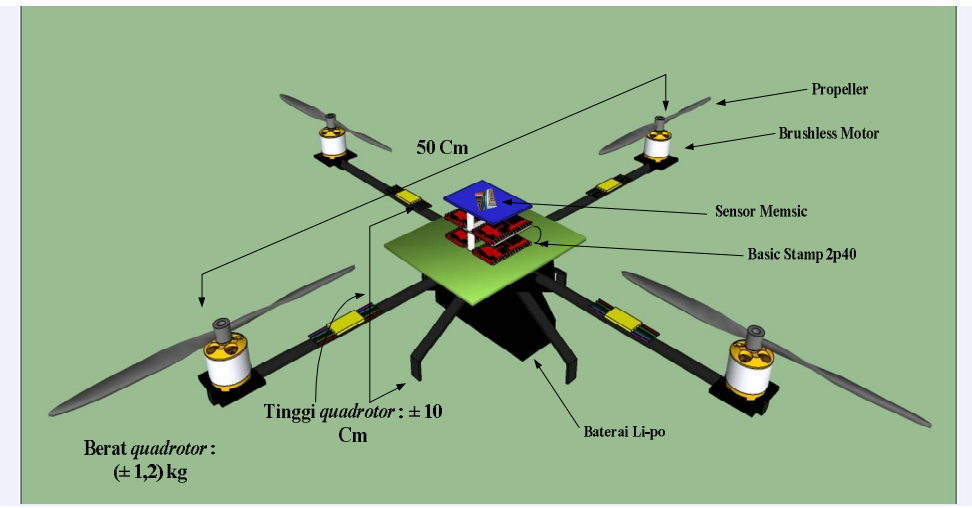
Bab ini akan membahas tentang perancangan sistem keseimbangan *quadrotor* yang didapat dari tiap-tiap rotor dengan didukung oleh sensor accelerometer. Pada perancangan sistem ini dibagi menjadi tiga bagian penting, yaitu:

- Perancangan mekanik,
- Perancangan perangkat keras, dan
- Perancangan perangkat lunak.

3.1 Perancangan Mekanik

Perancangan mekanik *quadrotor,* dirancang berdasarkan ukuran dan berat agar keseimbangan *quadrotor* dapat tercapai. Ukuran dan berat *quadrotor* yang dirancang adalah sebagai berikut:

- Diameter *quadrotor* : 50 cm
- Tinggi *quadrotor* : (± 10) cm
- Berat *quadrotor* : (± 1,2) kg



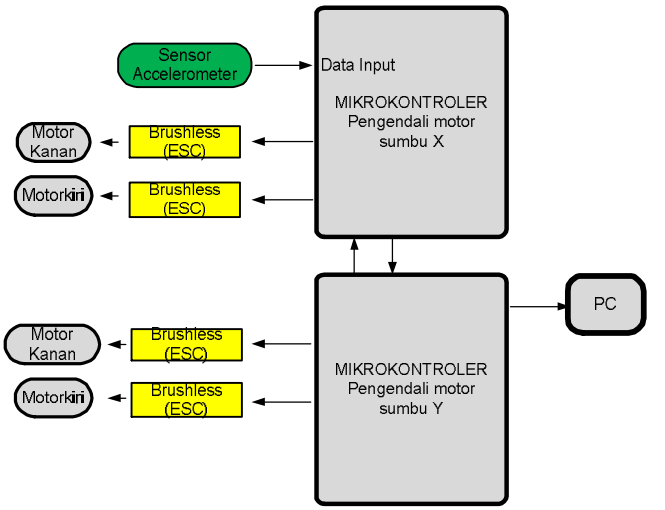
Gambar 3.1 Rancangan quadrotor tampak samping

Penempatan komponen:

- *Brushless* motor dan *propeller* dipilih sebagai aktuator agar *quadrotor* dapat melakukan pencarian keseimbangan dari rotasi yang ditentukan.
- Sensor ditempatkan pada bagian atas *quadrotor*.
- Mikrokontroler 1 diletakkan pada bagian bawah sensor *accelerometer*.
- Mikrokontroler 2 diletakkan pada bagian bawah mikrokontroler 1.
- Catu daya ditempatkan di bagian bawah quadrotor.

3.2 Perancangan Perangkat Keras

Diagram blok sistem yang akan dibuat ditunjukkan pada gambar 3.2.



Gambar 3.2 Diagram blok sistem quadrotor

Berikut penjelasan blok-blok yang terdapat pada gambar 3.2.

a. Mikrokontroler Satu

Bagian blok ini memuat Mikrokontroler *Basic Stamp 2p40*. Bagian blok ini berfungsi sebagai pengolah data yang diterima dari sensor *accelerometer* dan juga berfungsi sebagai pengendali motor di sumbu x.

b. Mikrokontroler Dua

Bagian blok ini memuat Mikrokontroler *Basic Stamp 2p40*. Bagian blok ini berfungsi sebagai pengendali motor di sumbu y. Dan pengolahan data keseluruhan system untuk dikirim ke PC.

c. Sensor Accelerometer

Bagian blok ini memuat sensor Memsic 2125. Bagian blok ini berfungsi sebagai pendeteksi akselerasi yang dialami *quadrotor* yang kemudian datanya akan diolah pada mikrokontroler satu dan dua.

3.2.1 Mikrokontroler (Basic Stamp 2p40)

Mikrokontroler yang digunakan pada perancangan *quadrotor* ini adalah mikrokontroler jenis *basic stamp 2P40* dengan 32 *port* yang dapat digunakan sebagi *input* atau *output*. Pemilihan mikrokontroler jenis ini didasari kemampuannya yang cukup handal dan pemrograman yang tidak terlalu sulit. Pada perancangan *quadrotor* yang dibuat, penulis menggunakan dua buah mikrokontroler *basic stamp 2p40,* yaitu sebagai mikrokontroler satu dan mikrokontroler dua.

Berikut ini merupakan konfigurasi *port* yang digunakan mikrokontroler satu, seperti yang terlihat pada tabel 3.1.

Tabel 3.1 Konfigurasi port yang digunakan pada mikrokontroler satu.

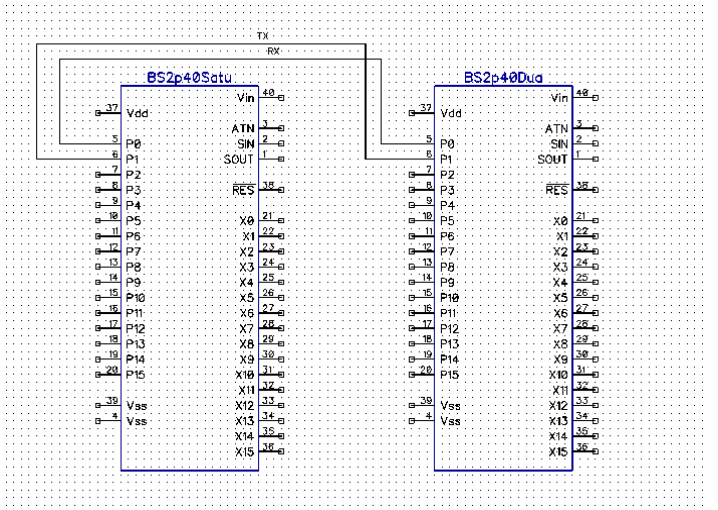
Nama Pin	No. Pin	Fungsi	Keterangan
MAIN I/O	0	*Output*	Penerima (RX) dari mikrokontroler *dua*
MAIN I/O	1	*Input*	Pengirim (TX) ke mikrokontroler *dua*
MAIN I/O	3	*Output*	Memsic 2125 (Yout)
MAIN I/O	2	*Output*	Memsic 2125 (Xout)
AUX I/O	0	*Output*	Motor kanan di sumbu x
AUX I/O	1	*Output*	Motor kiri di sumbu x

Berikut ini merupakan konfigurasi *port* yang digunakan mikrokontroler satu, seperti yang terlihat pada tabel 3.2.

Tabel 3.2 Konfigurasi port yang digunakan pada mikrokontroler dua.

Nama Pin	No. Pin	Fungsi	Keterangan
MAIN I/O	0	*Input*	Penerima (RX) dari mikrokontroler satu
MAIN I/O	1	*Output*	Pengirim (TX) ke mikrokontroler satu
MAIN I/O	2	*Input*	Penerima (RX) dari PC
MAIN I/O	3	*Output*	Pengirim (TX) ke PC
AUX I/O	0	*Output*	Motor kanan di sumbu y
AUX I/O	1	*Output*	Motor kiri di sumbu y

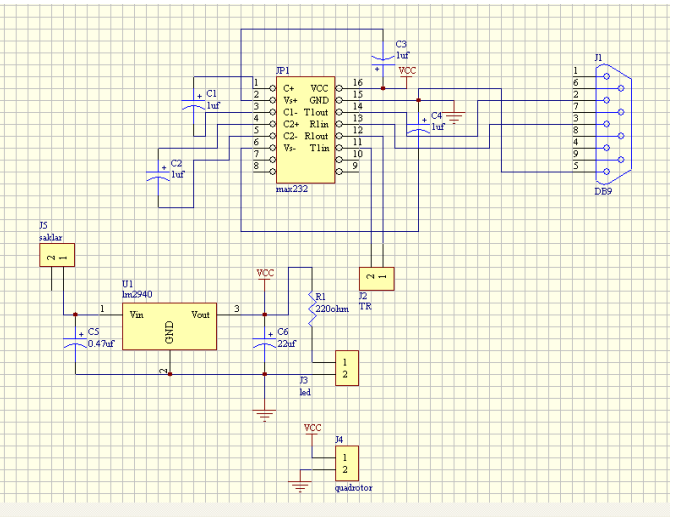
Mikrokontroler satu berfungsi untuk mengolah data pengukuran sensor *accelerometer* dan sebagai pengendali motor di sumbu x. Sedangkan mikrokontroler dua berfungsi sebagai pengendali motor di sumbu y dan penerima data pengukuran dari mikrokontroler satu. Dua mikrokontroler ini akan saling berkomunikasi. Mikrokontroler dua mengirimkan hasil data pengukuran ke PC. Rangkaian komunikasi mikrokontroler satu dengan mikrokontroler dua terlihat pada gambar 3.3



Gambar 3.3 Rangkaian komunikasi mikrokontroler satu dengan mikrokontroler dua

3.2.2 Pengubah Level Digital ke RS232 (MAX232)

Rangkaian RS232 berfungsi sebagai pengintegrasian antara *quadrotor* dan PC. Dengan menggunakan komponen IC MAX 232, dimana besar level tegangannya adalah antara -25 volt sampai dengan +25 volt. Untuk bit berlogika 1 maka besar level tegangannya adalah antara -3 volt sampai -25 volt, sedangkan untuk bit dengan logika 0 maka besar level tegangannya antara +3 sampai dengan +25 volt. Rangkaian RS232 ditunjukan pada gambar 3.

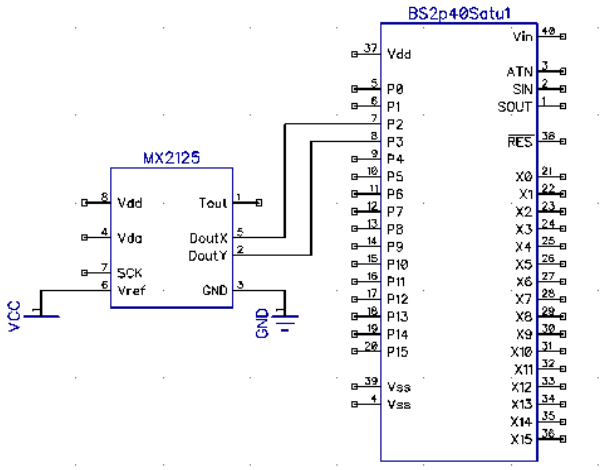


Gambar 3.4 Rangkaian RS232

3.2.3 Sensor MX2125 *Dual-Axis Accelerometer*

Pada perancangan *quadrotor* ini menggunakan sensor MX2125 yang digunakan sebagai sensor *accelerometer,* yaitu pengukur data rotasi yang terjadi pada *quadrotor.*

Pemilihan sensor ini didasari kemampuannya yang dapat mendeteksi akselerasi pada dua sudut. Dan sensor ini memiliki keluaran berupa pulsa digital yang secara kontinyu dengan frekuensi sebesar 100Hz, jadi tidak perlu adanya rangkaian ADC. seperti yang terlihat pada gambar 3.5



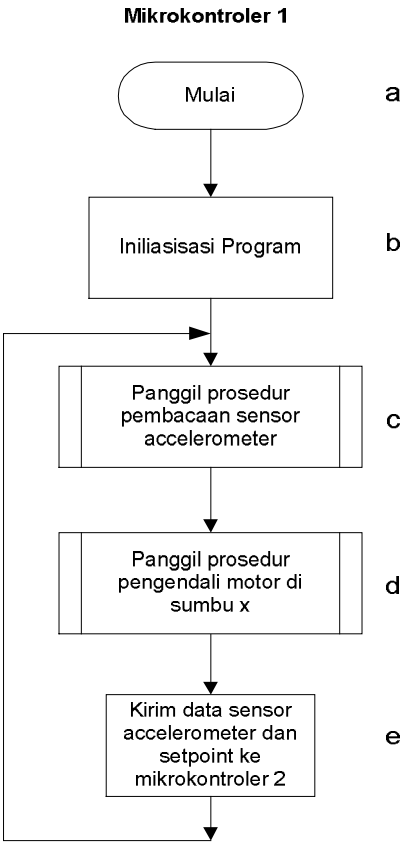
Gambar 3.5 Rangkaian MX2125 Dual-Axis Accelerometer

3.3 Perancangan Perangkat Lunak

Setelah dalam pembuatan perangkat keras selesai, bagian yang paling penting dalam pembuatan tugas akhir ini yaitu dalam merancang suatu perangkat lunak. Dimana perancangan perangkat lunak adalah pembuatan program utama untuk diimplemtasikan ke dalam pemograman *Basic* melalui *basic stamp editor* agar dapat diisi ke dalam mikrokontroler *basic stamp 2p40.* Perancangan perangkat lunak dibagi menjadi dua bagian yaitu:

- Perancangan perangkat lunak untuk mikrokontroler satu.
- Perancangan perangkat lunak untuk mikrokontroler dua.

3.3.1. Diagram Alir Program Mikrokontroler Satu

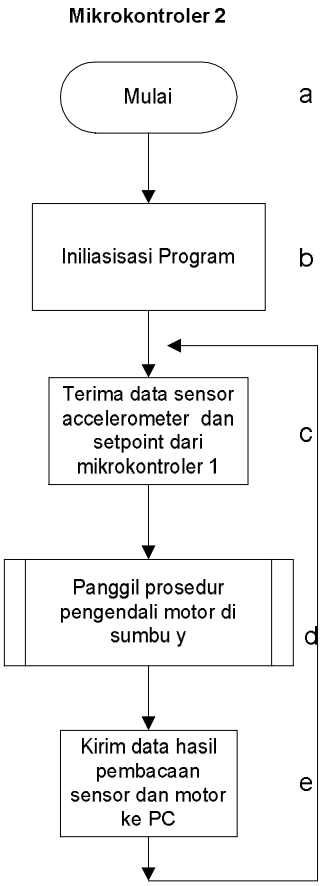


Gambar 3.6 Diagram alir program mikrokontroler satu

Berikut ini penjelasan dari diagram alir program mikrokontroler satu.

a. Memulai awal program
b. Pendeklarasian program, konfigurasi pin-pin dan deklarasi variabel.
c. Panggil procedure pembacaan sensor accelerometer
d. Panggil procedure pengendali motor di sumbu x dengan pengontrol PID.
e. Kirim data sensor accelerometer dan setpoint yang telah didapat ke mikrokontroler dua.

3.3.2. Diagram Alir Program Mikrokontroler Dua

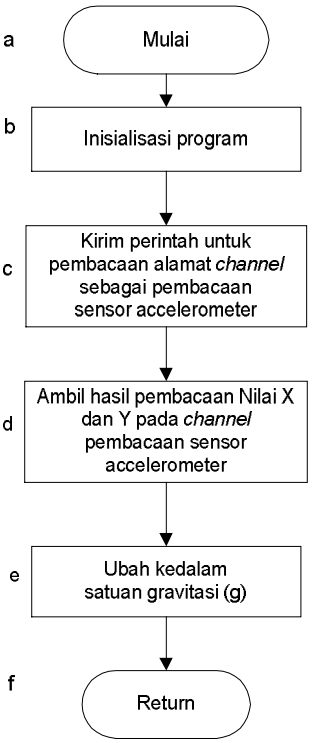


Gambar 3.7 Diagram alir program mikrokontroler dua

Berikut ini penjelasan dari diagram alir program mikrokontroler satu.

a. Memulai awal program
b. Pendeklarasian program, konfigurasi pin-pin dan deklarasi variabel.
c. Terima data dari mikrokontroler satu. Berupa data sensor accelerometer dan data pengendali motor yang terdapat di sumbu x.
d. Panggil procedure pengendali motor di sumbu y dengan pengontrol PID.
e. Kirim data hasil pengolahan secara keselurahan ke PC.

3.3.3. Diagram Alir Prosedur Pembacaan Sensor Accelerometer

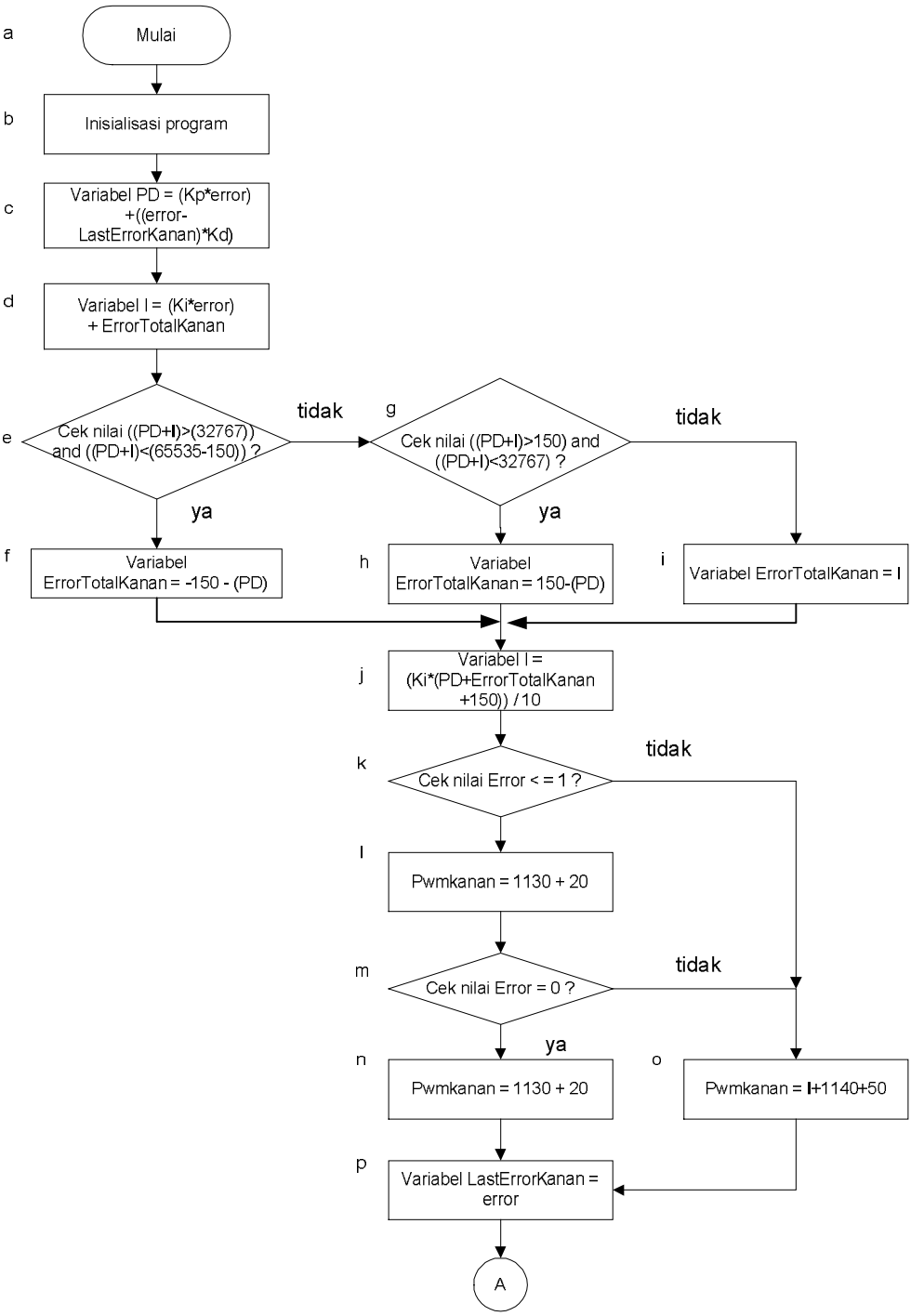


Gambar 3.8 Diagram alir prosedur pembacaan sensor accelerometer

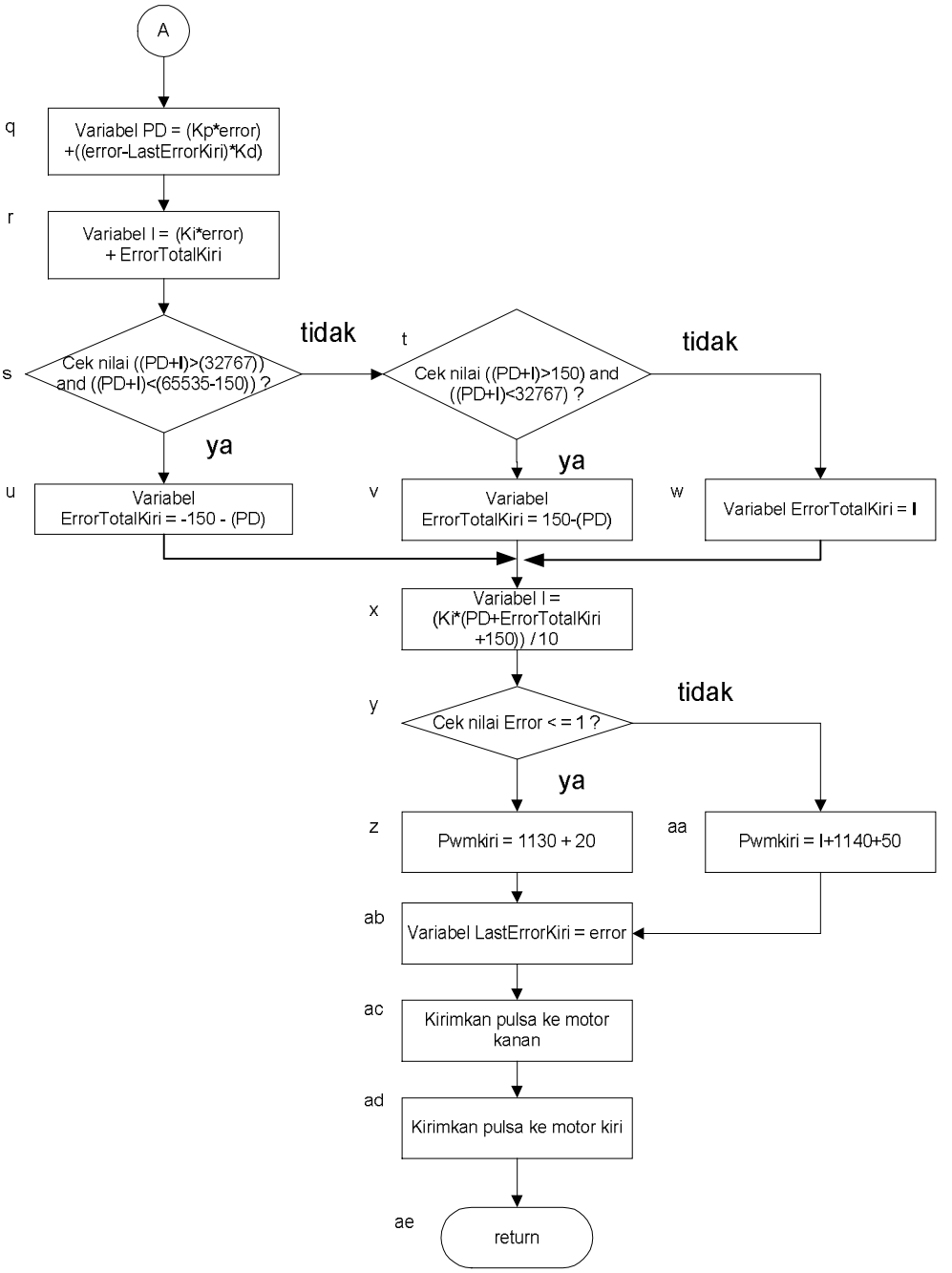
Berikut ini penjelasan diagram alir prosedur pembacaan sensor accelerometer

a. Memulai awal program.
b. Pendeklarasian program. Pengaturan nama prosesor, nama kompiler, konfigurasi pin-pin dan deklarasi variabel.
c. Mengirimkan perintah untuk membaca alamat *channel* yang digunakan untuk pembacaan sensor *accelerometer*.
d. Mengirim perintah untuk membaca hasil pengukuran dan menyimpannya pada variabel yang disediakan.
e. Hasil pengukuran diubah kedalam satuan gravitasi (g).
f. Keluar dari prosedur dan kembali mengeksekusi intruksi berikutnya pada program utama.

3.3.4. Diagram Alir Prosedur Pengendali Motor di Sumbu X dan Y



Gambar 3.9 Diagram alir prosedur pengendali motor di sumbu x dan y



Gambar 3.10 Diagram alir prosedur pengendali motor di sumbu x dan y (lanjutan)

Berikut tabel penjelasan diagram alir prosedur pengendali motor di sumbu x dan y:

a. Memulai awal program.
b. Pendeklarasian program. Pengaturan nama prosesor, nama kompiler, konfigurasi pin-pin dan deklarasi variabel.
c. Hasil dari (Kp*error)+((error-LastErrorKanan)*Kd) simpan pada variabel PD.
d. Hasil dari (Ki*error)+(ErrorTotalKanan) simpan pada variabel I.
e. Pemeriksaan apakah nilai pada variabel ((PD+I) > 32767) dan ((PD+I) < (65535-150)), jika ya lanjutkan ke poin f, jika tidak lanjutkan ke poin g.
f. Kurangi nilai (-150) dengan nilai pada variabel PD, kemudian hasil pengurangan simpan pada variabel ErrorTotalKanan.
g. Pemeriksaan apakah nilai pada variabel ((PD+I) > 150) dan ((PD+I) < 32767), jika ya lanjutkan ke poin h, jika tidak lanjutkan ke poin i.
h. Kurangi nilai 150 dengan nilai pada variabel PD, kemudian hasil pengurangan simpan pada variabel ErrorTotalKanan.
i. Isi nilai pada variabel ErrorTotalKanan dengan nilai pada variabel I.
j. Hasil dari ((Ki*(PD+ErrorTotalKiri+150)) / 10) simpan pada variabel I.
k. Pemeriksaan apakah nilai pada variabel Error <= 1, jika tidak lanjutkan ke poin o.
l. Tambah nilai pulsa 1130 dengan 20 dan simpan di variabel pwmkanan.
m. Pemeriksaan apakah nilai pada variabel Error = 0, jika ya lanjutkan ke poin n, jika tidak lanjutkan ke poin o.
n. Tambah nilai pulsa 1130 dengan 20 dan simpan di variabel pwmkanan.

o. Tambah nilai I dengan 1140 dan 50 dan simpan di variabel pwmkanan.
p. Isi nilai pada variabel ErrorTotalKanan dengan nilai pada variabel error.
q. Hasil dari (Kp*error)+((error-LastErrorKiri)*Kd) simpan pada variabel PD.
r. Hasil dari (Ki*error)+(ErrorTotalKiri) simpan pada variabel I.
s. Pemeriksaan apakah nilai pada variabel ((PD+I) > 32767) dan ((PD+I) < (65535-150)), jika ya lanjutkan ke poin u, jika tidak lanjutkan ke poin t.
t. Pemeriksaan apakah nilai pada variabel ((PD+I) > 150) dan ((PD+I) < 32767), jika ya lanjutkan ke poin h, jika tidak lanjutkan ke poin i.
u. Kurangi nilai (-150) dengan nilai pada variabel PD, kemudian hasil pengurangan simpan pada variabel ErrorTotalKiri.
v. Kurangi nilai 150 dengan nilai pada variabel PD, kemudian hasil pengurangan simpan pada variabel ErrorTotalKiri.
w. Isi nilai pada variabel ErrorTotalKanan dengan nilai pada variabel I.
x. Hasil dari ((Ki*(PD+ErrorTotalKiri+150)) / 10) simpan pada variabel I.
y. Pemeriksaan apakah nilai pada variabel Error <= 1, jika ya lanjutkan ke poin z, jika tidak lanjutkan ke poin aa.
z. Tambah nilai pulsa 1130 dengan 20 dan simpan di variabel pwmkiri.
aa. Tambah nilai I dengan 1140 dan 50 dan simpan di variabel pwmkiri.
ab. Isi nilai pada variabel ErrorTotalKiri dengan nilai pada variabel error
ac. Mengirimkan pulsa ke *brushless* motor kiri sesuai dengan nilai pada variabel pwmkiri.

ad. Mengirimkan pulsa ke *brushless* motor kanan sesuai dengan nilai pada variabel pwmkanan.
ae. Keluar dari prosedur dan kembali mengeksekusi intruksi berikutnya pada program utama.

Algoritma di atas diimplementasikan pada bahasa pemograman *basic* melalui *basic stamp editor* agar dapat diisi ke dalam mikrokontroler *basic stamp 2p40.*