

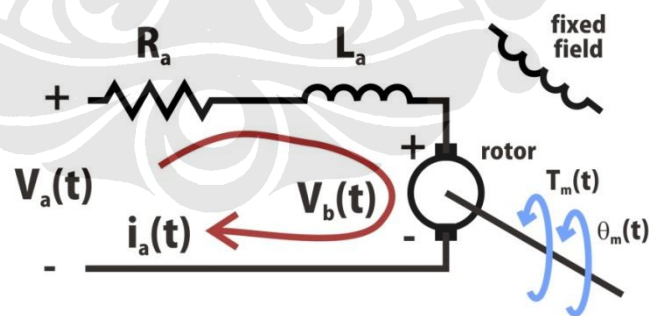
BAB 3

PERANCANGAN KENDALI MOTOR DC

Perancangan kendali motor DC dalam skripsi ini meliputi perancangan motor DC, perancangan blok kendali, perancangan kendali PID, perancangan perangkat lunak, dan perancangan perangkat keras.

3.1 Perancangan Motor DC

Motor DC adalah aktuator listrik yang menggunakan tegangan input sebagai variabel pengendali. Pengendalian motor DC ada dua jenis yaitu pengendalian medan (*field-controlled*) dan pengendalian jangkar (*armature controlled*). Motor DC yang digunakan pada skripsi ini adalah motor DC dengan pengendalian jangkar. Rangkaian pengganti untuk motor DC dengan pengendalian jangkar dapat dilihat pada Gambar 3.1.



Gambar 3.1 Rangkaian Ganti Motor DC

Motor DC dengan pengendalian jangkar menggunakan arus jangkar i_a sebagai variabel pengendali. Medan stator dapat ditimbulkan dengan kumparan berarus atau dengan magnet permanen.

Ketika arus medan konstan mengalir pada kumparan medan, torsi motor yaitu

$$T_m(s) = (K_1 K_f I_f) I_a(s) = K_m I_a(s) \quad (3.1)$$

Jika menggunakan magnet permanen, maka

$$T_m(s) = K_m I_a(s) \quad (3.2)$$

Dimana K_m adalah fungsi permeabilitas dari bahan magnet. Arus jangkar berhubungan dengan tegangan input pada rangkaian jangkar sesuai dengan persamaan

$$V_a(s) = (R_a + L_a s) I_a(s) + V_b(s) \quad (3.3)$$

Dimana $V_b(s)$ adalah tegangan *back electromotive-force* (emf) yang sebanding dengan kecepatan motor. Karena itu, kita mendapatkan persamaan

$$V_b(s) = K_b \omega(s) \quad (3.4)$$

Dan arus jangkar

$$I_a(s) = \frac{V_a(s) - K_b \omega(s)}{(R_a + L_a s)} \quad (3.5)$$

Torsi pada motor adalah sama dengan torsi yang dikirimkan ke beban.

$$T_m(s) = T_L(s) + T_d(s) \quad (3.6)$$

Dimana $T_L(s)$ adalah torsi beban dan $T_d(s)$ adalah torsi gangguan yang biasanya diabaikan. Bagaimanapun juga, torsi gangguan harus diperhatikan pada sistem yang bersubjek pada gaya eksternal seperti hembusan angin yang keras pada antenna. Torsi beban untuk benda berputar dituliskan sebagai

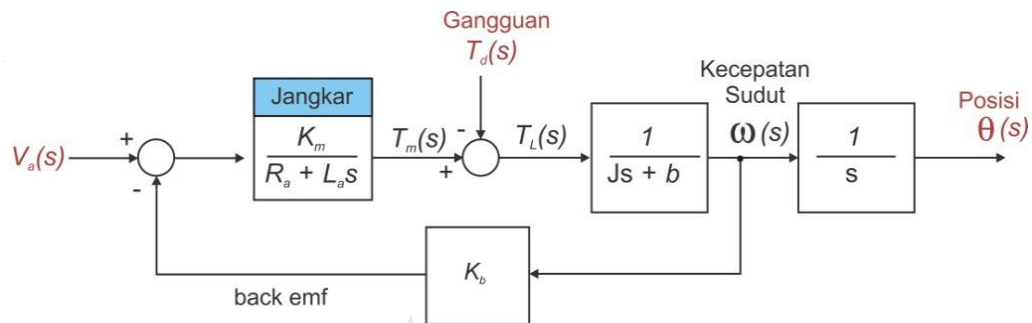
$$T_L(s) = J s^2 \theta(s) + b s \theta(s) \quad (3.7)$$

Persamaan (3.7) dan (3.8) merepresentasikan torsi beban

$$T_L(s) = J s^2 \theta(s) + b s \theta(s) = T_m(s) - T_d(s) \quad (3.8)$$

Secara skematis, hubungan-hubungan antara motor DC dengan

pengendalian jangkar ditunjukkan pada Gambar 3.2.



Gambar 3.2 Blok Diagram Fungsi Alih Motor DC

Pada skripsi ini, digunakan persamaan 3.9 untuk mengetahui fungsi alih dari motor. Nilai ζ dan ω_n didapatkan dari grafik step response yang dilakukan dalam uji coba.

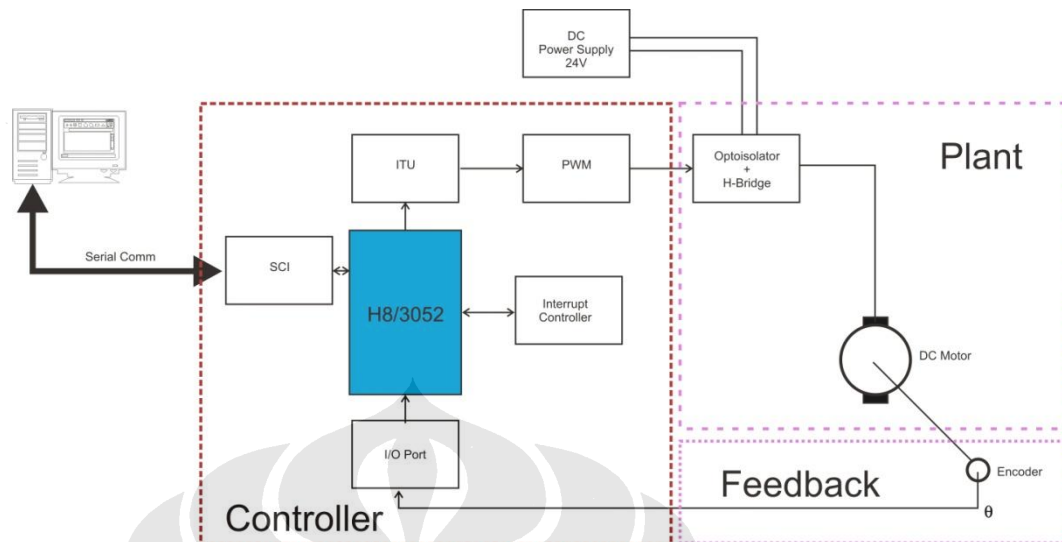
$$\frac{\omega(s)}{V_a(s)} = \frac{1/k_E}{1 + \frac{2s\zeta}{\omega_n} + s^2/\omega_n^2} \quad (3.9)$$

Dari persamaan fungsi alih yang telah didapat, kemudian digunakan untuk mendesain pengendali PID pada motor DC.

3.2 Perancangan Blok Kendali

Perancangan blok kendali motor DC yang terdiri atas plant, feedback, controller, dan pengambil data dapat digambarkan sebagai seperti pada Gambar 3.3 .

PC memberikan nilai setpoint berupa kecepatan ke mikrokontroler H8/3052. Dengan algoritma kendali PID yang telah dibuat, akan menghasilkan input untuk plant berupa nilai duty cycle. Aktuator pada plant yaitu motor DC merespon input tersebut dengan menghasilkan perubahan posisi. Perubahan posisi tersebut diukur menggunakan encoder. Encoder kemudian memberikan feedback ke mikrokontroler.



Gambar 3.3 Skema Perancangan Kendali PID pada Motor DC

Mikrokontroler mengolah nilai perubahan posisi tersebut menjadi kecepatan, kemudian membandingkan nilai kecepatan dari feedback encoder dengan kecepatan dari setpoint. Perbedaan nilai tersebut menghasilkan nilai error yang kemudian diolah pada program kontrol untuk menjadi input pada plant kembali. Nilai-nilai pengukuran yang didapat direkam dan ditampilkan pada komputer untuk selanjutnya dapat dianalisa.

3.3 Perancangan Kendali PID

Salah satu bentuk pengendali yang secara luas digunakan pada pengendalian proses industri adalah pengendali PID. Pengendali ini mempunyai fungsi alih sebagai berikut.

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s \quad (3.10)$$

Dengan K_P = proportional gain

K_I = integral gain

K_D = differential gain

Persamaan untuk output pada domain waktu adalah

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (3.11)$$

Jika kita setting $K_D = 0$, maka kita mempunyai pengendali proportional plus integral (PI).

$$G_c(s) = K_P + \frac{K_I}{s} \quad (3.12)$$

Ketika $K_I=0$, didapatkan

$$G_c(s) = K_P + K_D s \quad (3.13)$$

yang disebut sebagai pengendali proportional plus derivative (PD).

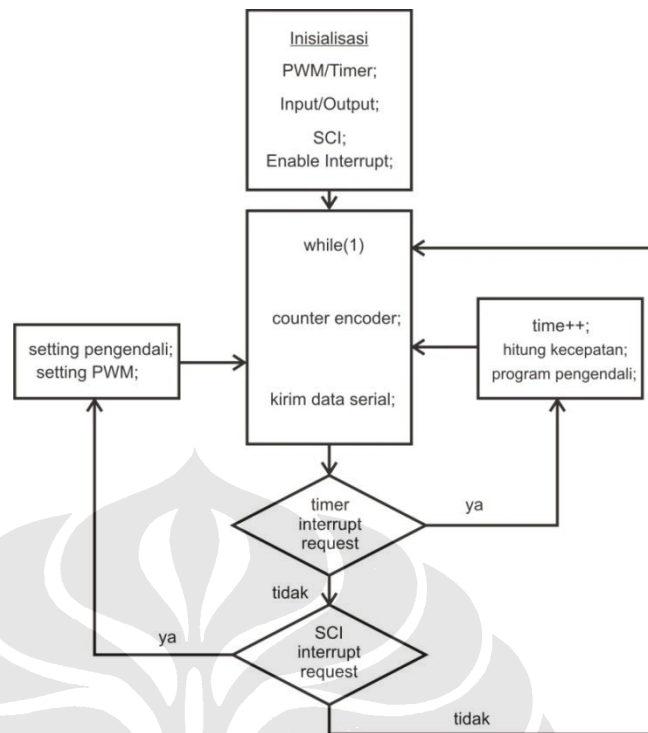
Program pengendali tersebut kemudian diterapkan pada pemrograman C pada mikrokontroler.

3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri atas dua bagian yaitu perancangan perangkat lunak pada mikrokontroler dan perancangan perangkat lunak pada PC.

Pada perancangan perangkat lunak mikrokontroler, penulisan program dengan menggunakan bahasa C dengan menggunakan software HEW (*High-performanced Embedded Workshop*) dari Renesas. Kemudian program C yang telah dibuat kemudian di-build hingga menjadi file dalam .mot. File .mot inilah yang di-*burn* ke mikrokontroler dengan menggunakan program H8Writer.

Untuk program yang dijalankan pada mikrokontroler, seperti ditunjukkan pada diagram alir pada Gambar 3.4.



Gambar 3.4 Diagram Alir Program pada Mikrokontroler

Pada awal program dimulai dengan inisialisasi program yang terdiri atas inisialisasi PWM/Timer, inisialisasi input/output, inisialisasi SCI, dan inisialisasi interrupt. Pada program utama dilakukan program penghitungan posisi motor dengan menggunakan sinyal dari encoder serta program pengiriman data serial yang terdiri atas data posisi, waktu, setpoint, dan kecepatan motor.

Pada saat terjadi permintaan interrupt timer, mikrokontroler melaksanakan program penanganan interrupt yang terdiri atas penambahan nilai waktu, perhitungan kecepatan, dan aplikasi program pengendalian. Pada saat terjadi permintaan interrupt penerimaan SCI, mikrokontroler menjalankan program penanganan interrupt yang terdiri atas setting nilai duty cycle PWM, serta setting nilai K_p , K_i , dan K_d untuk pengendalian. Jika program penanganan interrupt selesai, maka program kembali ke program utama.

Perancangan perangkat lunak pada PC terdiri atas program pembacaan data

serial menggunakan gcc cygwin, pembuatan grafik data dengan gnuplot, dan tampilan secara realtime menggunakan aplikasi graph232.

3.5 Perancangan Perangkat Keras

Perancangan blok diagram pengendalian kemudian diaplikasikan ke perancangan perangkat keras. Masing-masing komponen dirangkai hingga menjadi rangkaian perangkat keras seperti ditunjukkan pada Gambar 3.5.



Gambar 3.5 Rangkaian Perangkat Keras

Rangkaian perangkat keras yang digunakan yaitu terdiri atas:

- a. Motor DC Tsukasa Electric TG-38A-AM-50-KA



Gambar 3.6 Motor DC Tsukasa Electric

Motor DC yang digunakan yaitu motor DC produksi Tsukasa Electric dengan nomor seri TG-38A-AM-50-KA. Motor ini mempunyai tegangan rating 24V.

b. *Evaluation Kit* AKI-H8/3052-LAN

Evaluation kit AKI-H8/3052-LAN ini digunakan sebagai perangkat pengendali. *Evaluation kit* multifungsi ini memiliki beberapa dukungan perangkat keras tambahan yaitu LCD, 4 buah LED, kanal untuk motor servo, koneksi RS232, koneksi LAN, dan koneksi power USB, dan koneksi power 5V.

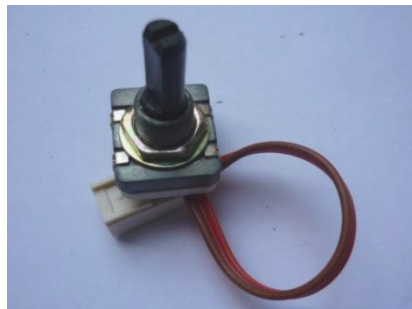


Gambar 3.7 AKI-H8/3052-LAN

LCD digunakan untuk debugging dan menampilkan data. Akan tetapi pada skripsi ini LCD tidak digunakan karena operasi LCD cukup menyita waktu yang lama. Kanal servo digunakan untuk output sinyal PWM serta output pin CS. Komunikasi serial dapat memilih menggunakan SCI0 atau SCI1. Untuk catu daya digunakan power USB dengan tegangan kurang lebih 5V.

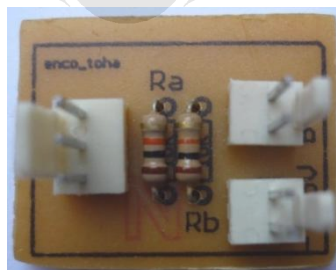
c. Rotary Encoder EC16B

Encoder yang digunakan yaitu EC16B yang dibuat oleh ALPS *Electricity Company* di Jepang. Bentuk hardware dari encoder EC16B dapat dilihat pada Gambar 3.8.

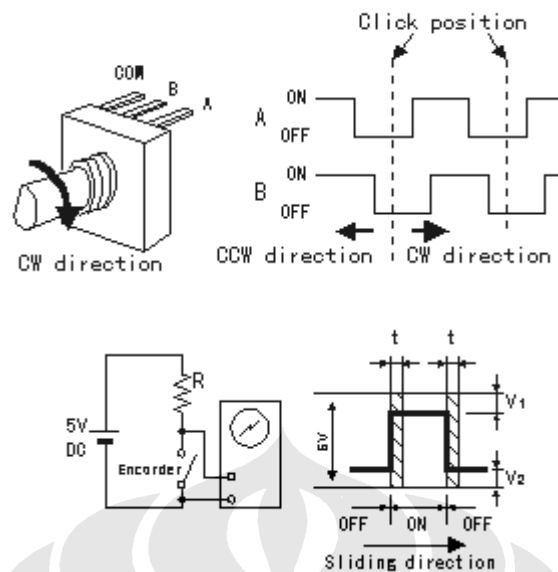


Gambar 3.8 Encoder EC16B

Encoder ini mempunyai 3 terminal, satu digunakan sebagai *common terminal* (COM), dan dua lainnya digunakan untuk output A dan B. Output dari terminal yaitu kondisi dimana terminal tersebut terhubung dengan *common terminal* atau tidak. Encoder EC16B ini tergolong sebagai encoder mekanik yaitu menggunakan sentuhan dalam menentukan output encoder. Dua terminal output mempunyai perbedaan waktu saat menyentuh *common terminal* ketika sumbu encoder berputar. Untuk interface encoder dengan mikrokontroler, digunakan rangkaian encoder seperti pada Gambar 3.9. Rangkaian ini terdiri atas dua buah resistor 10K. Rangkaian encoder ini menggunakan catu daya dari mikrokontroler yaitu sebesar 5V. Terdapat dua pin sebagai output dari encoder yaitu pin A dan pin B. Pin A dan pin B ini harus dijaga agar tidak menyentuh ground agar tidak terjadi kesalahan dalam perhitungan.



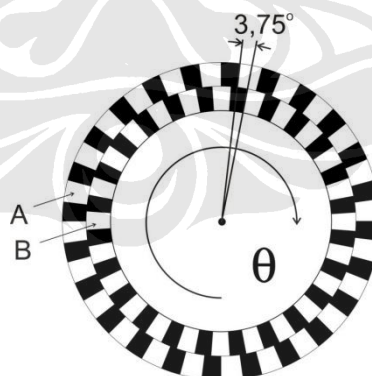
Gambar 3.9 Rangkaian Encoder



Gambar 3.10 Prinsip Kerja Encoder EC16B

Sumber: http://www.interq.or.jp/japan/se-inoue/e_ckt10.htm

Gambar 3.10 menunjukkan sinyal encoder yang dihasilkan pada pin A dan pin B saat diputar searah atau berlawanan arah jarum jam. Encoder ini dapat digunakan untuk menghitung putaran baik searah jarum jam maupun berlawanan arah jarum jam.



Gambar 3.11 Incremental Rotary Encoder

Encoder EC16B merupakan incremental rotary encoder dengan resolusi sebesar $3,75^\circ$. Resolusi encoder tersebut merupakan satuan terkecil perubahan posisi yang dapat diukur oleh encoder. Besar resolusi ini didapatkan dari besar putaran encoder dibagi dengan jumlah perubahan output biner dalam satu kali

putaran yaitu ditunjukkan pada persamaan 3.14.

$$Resolusi = \frac{\text{Sudut dalam 1 putaran}}{\text{Jumlah pulsa} \times \text{Jumlah perubahan biner per pulsa}} \quad (3.14)$$

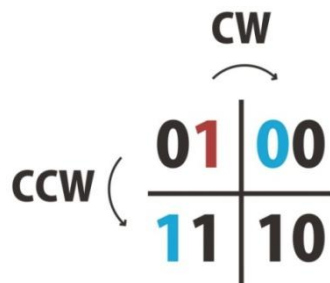
$$Resolusi = \frac{360^\circ}{24 \times 4} = 3,75^\circ \quad (3.15)$$

Sebuah incremental encoder menggunakan 2-bit output biner untuk menentukan posisi putaran motor. Encoder EC16B ini menggunakan sensor mekanik untuk mendapatkan output biner tersebut. Terdapat dua track yaitu track A dan track B sebagai 2-bit output encoder. Persentuhan antara dua pin yaitu pin A dan B encoder dengan pin common menghasilkan sinyal output encoder.

Perbedaan fasa antara track A dan B digunakan untuk menghitung putaran encoder. Kita dapat menggunakan Gray code untuk menentukan posisi dan arah dari perputaran motor. Gray code atau juga dikenal *reflected binary code* adalah sistem penomoran biner dimana dua nilai yang bersebelahan tepat hanya memiliki satu digit beda. Dinamakan Gray code karena ditemukan oleh Frank Gray. Pada awalnya Frank Gray memperkenalkan *reflected binary code* dalam paten aplikasinya tahun 1947. Dia memberikan nama berawal dari fakta bahwa kode ini “mungkin dibentuk dari kode biner yang konvensional dengan urutan proses yang terbalik”.

Kode ini dikatakan *reflected* karena bisa dibentuk dengan cara dicerminkan. Ambil Gray code 0,1 kemudian dicerminkan dan dituliskan pada urutan setelahnya hingga menjadi 0, 1, 1, 0. Tambahkan 0 pada setengah pertama dan 1 pada setengah digit kedua setelah digit-digit tersebut menjadi 00, 10, 11, 01 (Gray code 2 digit). Untuk mendapatkan Gray code dengan jumlah digit ke-(n+1) yaitu dengan mencerminkan Gray code digit ke-n kemudian menambahkan digit 0 pada setengah pertama dan digit 1 pada setengah kedua.

Prinsip kerja Gray code dalam menentukan posisi dan arah perputaran motor dapat dijelaskan seperti pada **Error! Reference source not found.**



Gambar 3.12 Prinsip Gray Code 2bit

Gambar itu dapat kita sederhanakan dengan algoritma sebagai berikut

`if(bit_kanan_lama XOR bit_kiri_baru == 1) CW++;`

`if(bit_kanan_lama XOR bit_kiri_baru == 0) CCW++;`

dengan bit kanan = bit pada track B, dan bit kiri = bit pada track A.

Pada gambar di atas, posisi 01 kita anggap sebagai posisi lama. Jika berputar searah jarum jam (CW), maka posisi baru adalah 00. Dengan algoritma tersebut, maka:

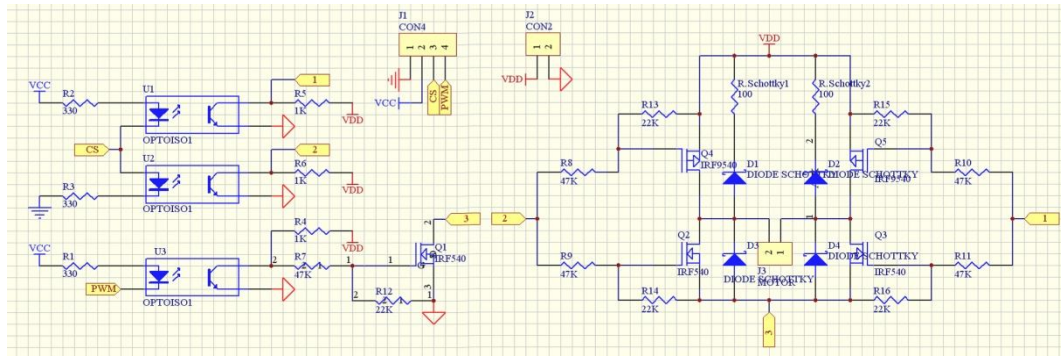
`bit_kanan_lama XOR bit_kiri_baru = 1 XOR 0 = 1`, maka CW++;

Sedangkan jika berputar berlawanan arah jarum jam, maka posisi baru adalah 11.

`bit_kanan_lama XOR bit_kiri_baru = 1 XOR 1 = 0`, maka CCW++;

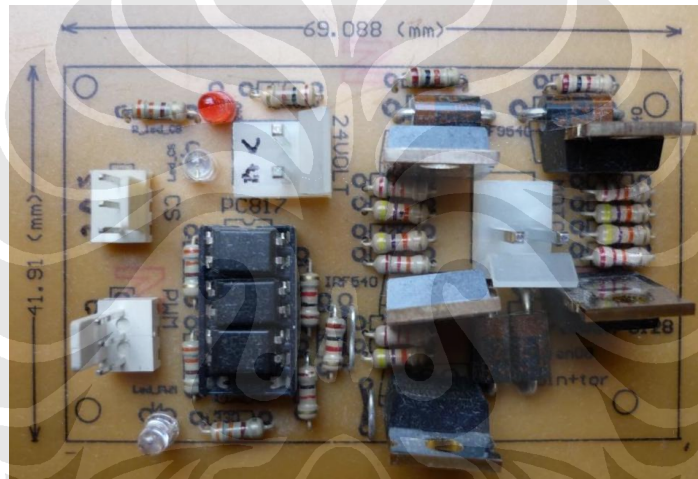
d. Rangkaian *driver* motor

Rangkaian *driver* motor di sini merupakan gabungan antara rangkaian optoisolator dan H-bridge. Skematik dan hardware rangkaian driver motor dapat dilihat pada Gambar 3.13 dan Gambar 3.14.



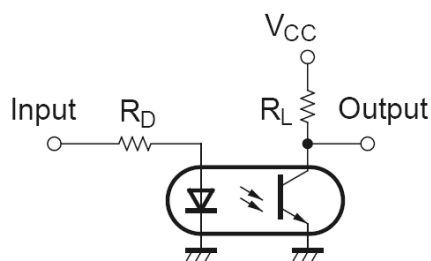
Gambar 3.13 Rangkaian skematik *Driver Motor: Optoisolator dan H-Bridge*

Sumber: Dokumen Tim Robot UI 2008



Gambar 3.14 Optoisolator dan Driver H-Bridge Motor DC

Rangkaian optoisolator digunakan untuk memisahkan sumber tegangan mikrokontroler dengan sumber tegangan motor, dan juga menghasilkan output PWM dan $\overline{\text{PWM}}$. Sedangkan input dari optoisolator ada dua yaitu PWM dan CS. CS inilah yang menentukan output optoisolator apakah PWM atau $\overline{\text{PWM}}$. Komponen utama optoisolator ini adalah PC817 SHARP.



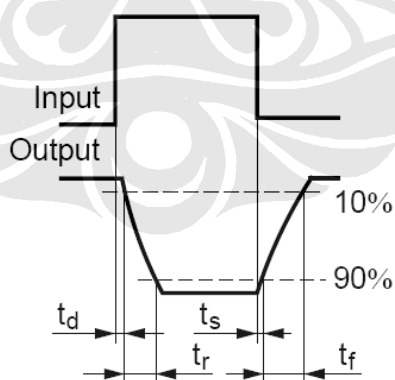
Gambar 3.15 Prinsip Kerja PC817

Sumber: <http://pdf1.alldatasheet.net/datasheet-pdf/view/43371/SHARP/PC817.html>

Prinsip kerja optoisolator pada Gambar 3.15 adalah sebagai berikut. Pada saat input bernilai HIGH, maka LED pada optoisolator akan menyala dan transistor pada optoisolator ON sehingga output dihubungkan dengan GROUND dan output tidak menyala.

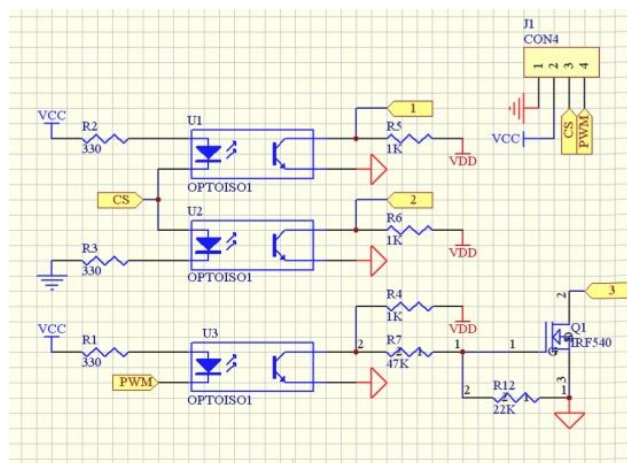
Sebaliknya saat input bernilai LOW, maka LED pada optoisolator tidak menyala dan transistor OFF. Akibatnya output mendekati nilai V_{CC} .

Dengan rangkaian seperti pada Gambar 3.15 maka hubungan antara input dan output dapat digambarkan seperti pada Gambar 3.16.



Gambar 3.16 Hubungan input dan output pada PC817

Sumber: <http://pdf1.alldatasheet.net/datasheet-pdf/view/43371/SHARP/PC817.html>

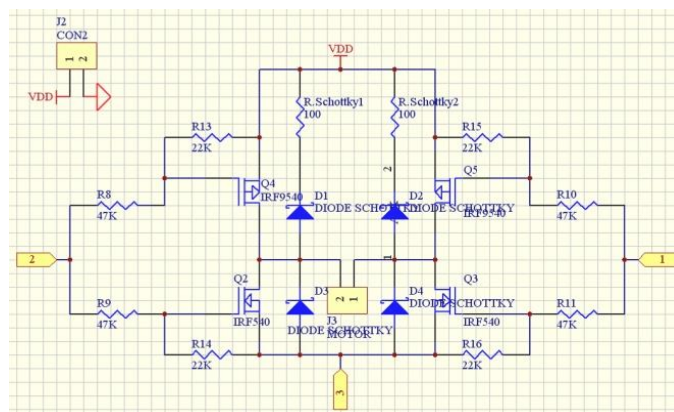


Gambar 3.17 Rangkaian Optoisolator

Jika diaplikasikan pada rangkaian optoisolator seperti pada Gambar 3.17, maka optoisolator dapat digunakan untuk menentukan arah motor. Nilai VCC kita anggap sebagai 1 dan nilai Ground kita anggap 0. Saat pin CS bernilai 1, maka terjadi perbedaan tegangan antara CS dengan Ground sehingga LED pada U3 akan menyala. Hal ini mengakibatkan transistor pada U3 ON dan pin 2 terhubung dengan Ground sehingga bernilai 0. Sebaliknya karena antara VCC dan CS tidak terdapat perbedaan tegangan maka LED pada U1 tidak menyala, transistor U1 OFF. Sehingga pin 1 tidak terhubung dengan Ground, dan tegangan pada pin 1 besarnya kurang lebih sama dengan VDD dan nilainya kita anggap 1. Sehingga saat pin CS bernilai 1, maka pin 1 bernilai 1 dan pin 2 bernilai 0.

Hal ini berlaku sebaliknya, yaitu saat pin CS bernilai 0, maka pin 1 bernilai 0 dan pin 2 bernilai 1. Pengaturan inilah yang menyebabkan motor dapat bergerak searah jarum jam atau berlawanan arah dengan jarum jam.

Rangkaian H-Bridge sendiri digunakan untuk mengendalikan gerak motor maju atau mundur. Komponen utama yang digunakan yaitu transistor IRF9540N dan IRF540N. Skematik Protel rangkaian H-Bridge dapat dilihat pada Gambar 3.18.



Gambar 3.18 Rangkaian H-Bridge Motor

Pin 1 dan 2 pada rangkaian merupakan pin yang menentukan arah putaran motor. Nilai pin ini ditentukan dari output rangkaian optoisolator. Pin 3 merupakan pin PWM yang menentukan besarnya kecepatan motor. Pin V_{DD} merupakan sumber tegangan untuk motor yaitu sebesar 24V. Dioda schottky digunakan untuk menangkap back emf motor yang dihasilkan oleh kumparan motor saat power on dan off. Tegangan flyback ini bisa berapa kali lebih tinggi daripada tegangan supply. Jika tidak digunakan diode, maka transistor dapat terbakar. Resistor $R_{Schottky}$ sebesar 100Ω digunakan sebagai pembagi tegangan. Resistor ini hanya dipasang pada bagian antara dioda dengan sumber tegangan sedangkan di sisi diode dengan PWM tidak dipasang karena sudah terpasang seri.

e. Perangkat keras lainnya

Perangkat keras lainnya terdiri atas PC, peralatan serial, dan digital tachometer. PC digunakan untuk menampilkan dan merekam data yang diperoleh melalui komunikasi serial dari mikrokontroler. Tachometer digunakan untuk mengukur kecepatan motor DC untuk dibandingkan dengan pengukuran menggunakan encoder. Tachometer yang digunakan pada skripsi ini yaitu Dual Digital Tachometer DT-2268 Lutron. Sedangkan peralatan komunikasi serial terdiri atas konektor USB ke serial dan kabel serial to serial.