

## BAB II

### TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan skripsi ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- Kontrol PID ( *Proporsional Integral Differensial* )
- Modul Arduino Uno
- Motor DC ( *Direct Current* )
- Modul Pengendali motor DC ( *Driver EMS 5A H-Bridge* )
- *Soil Moisture Sensor YL69/LM393*
- PWM ( *Pulse Width Modulation* )

#### 2.1 Kontroler

Keberadaan kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu sub sistem, yaitu kontroler.

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Hal ini sesuai dengan tujuan sistem kontrol dimana mendapat nilai sinyal keluaran sama dengan nilai yang diinginkan referensi. Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Apabila perbedaan antara nilai referensi dengan nilai keluaran relatif besar, maka kontroler yang baik seharusnya mampu mengamati perbedaan ini untuk segera menghasilkan sinyal keluaran untuk mempengaruhi *plant*. Dengan demikian sistem secara cepat mengubah keluaran *plant* sampai diperoleh selisih dengan nilai referensi sekecil mungkin.

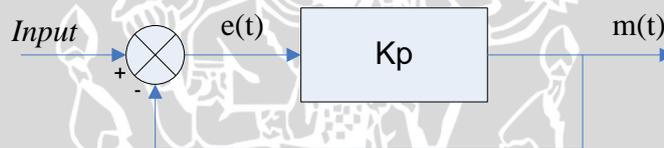
Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran *plant* dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya

menghasilkan sinyal kontrol untuk menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata.K.,1996).

### 2.1.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

Pada Gambar 2.1 menunjukkan blok diagram yang menggambarkan hubungan antara *input* (besaran referensi yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.



**Gambar 2.1** Diagram blok kontroler proporsional

Sumber: Ogata,1995: 157

Pada pengendali proporsional hubungan antara  $m(t)$  dan  $e(t)$  adalah

$$m(t) = K_p e(t) \quad (2.1)$$

Keterangan :

$K_p$  = Kepekaan proporsional atau penguatan

$m(t)$  = Keluaran kontroler

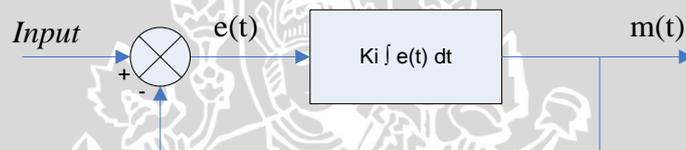
$e(t)$  = Sinyal kesalahan

Dengan  $K_p$  adalah penguatan proporsional. Keluaran  $m(t)$  hanya tergantung pada  $K_p$  dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan. Penambahan  $K_p$  akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respon dan mengurangi kesalahan keadaan mantap.

### 2.1.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah plant tidak memiliki unsur integrator (1/s), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.2 menunjukkan blok diagram kontroler integral.



**Gambar 2.2** Diagram Blok Kontroler Integral  
Sumber: Ogata, 1995: 158

Nilai keluaran kontroler  $m(t)$  sebanding dengan integral sinyal kesalahan  $e(t)$ ,  
Sehingga

$$\frac{dm(t)}{dt} = Ki \cdot e(t) \tag{2.2}$$

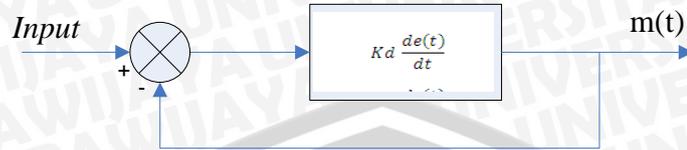
$$m(i) = Ki \int_0^t e(t)dt \tag{2.3}$$

dengan  $K_i$  adalah konstanta integral. Jika sinyal kesalahan  $e(t)=0$ , maka laju perubahan sinyal kendali integral  $\frac{dm(t)}{dt} = 0$  atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi control integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respon.

### 2.1.3 Kontroler Differensial

Kontroler differensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan

perubahan yang sangat besar dan cepat. Gambar 2.3 berikut menunjukkan blok diagram pada kontroler differensial.



**Gambar 2.3** Blok diagram kontroler differensial  
 Sumber: Ogata, 1995: 177

Nilai keluaran kontroler  $m(t)$  sebanding laju sinyal kesalahan  $\frac{de(t)}{dt}$ .

Hubungan ini dapat ditulis sebagai:

$$m(t) = Kd \frac{de(t)}{dt} \tag{2.4}$$

Dengan  $Kd$  adalah konstanta diferensial yang digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem serta dapat meredam osilasi. Kontroler differensial akan memberikan sinyal kendali keluaran  $m(t) = 0$ , untuk sinyal kesalahan  $e(t)$  yang konstan sehingga kontroler differensial tidak mempengaruhi keadaan mantap.

Berdasarkan karakteristik kontroler tersebut, kontroler differensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler differensial hanyalah efek dari lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler differensial tidak bisa digunakan tanpa ada kontroler lain.

Dari ketiga aksi kontrol dasar di atas dapat dibuat kombinasi dari ketiganya, yaitu:

#### 2.1.4 Kontroler Proporsional Integral (PI)

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = Kp \cdot e(t) + \frac{Kp}{Ti} \int_0^t e(t) dt \tag{2.5}$$

Kontroler ini menghasilkan sinyal kesalahan  $\int e(t) dt$  kemudian ditambahkan dengan sinyal kesalahan  $e(t)$ .



### 2.1.5 Kontroler Proporsional Diferensial (PD)

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.6)$$

Kontroler PD selalu mengukur kemiringan (*slope*) sinyal kesalahan  $\frac{de(t)}{dt}$  dan memperkirakan akan besar *overshoot* yang akan terjadi serta memberikan koreksi sebelum terjadi lewatan sebenarnya sehingga diperoleh *maximum overshoot* yang kecil.

Jika kesalahan keadaan mantap tidak berubah terhadap waktu maka turunannya terhadap waktu sama dengan nol, sehingga kontroler PD tidak mempunyai pengaruh terhadap kesalahan keadaan mantap, tetapi jika terdapat perubahan kesalahan, kontroler PD akan mengurangi besar kesalahan keadaan mantap. Jadi kontroler PD digunakan untuk memperbaiki suatu sistem pengendalian yang tanggapan peralihannya mempunyai *maximum overshoot* yang berlebihan tanpa memperhitungkan kecepatan responnya.

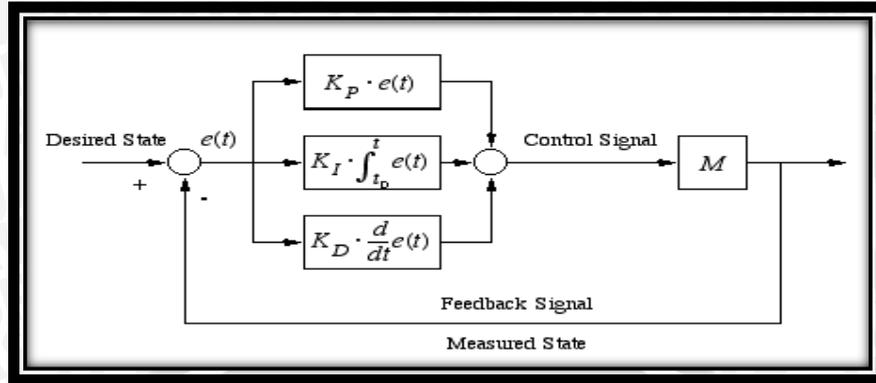
### 2.1.6 Kontroler Proporsional Integral Diferensial (PID)

Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral diferensial (PID). Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam Gambar 2.4.

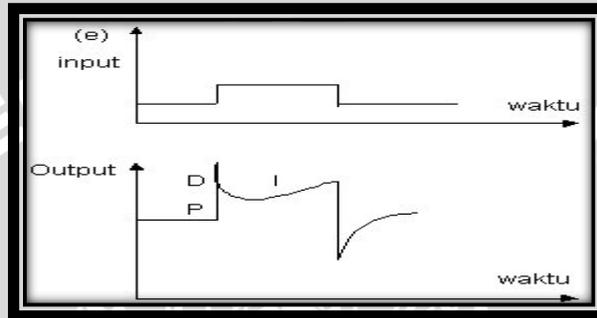
Aksi kontrolnya dinyatakan sebagai:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.7)$$

Jenis kontroler ini digunakan untuk memperbaiki kecepatan respon, mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.



Gambar 2.4 Diagram blok kontroler PID



Gambar 2.5 Hubungan fungsi waktu antara sinyal keluaran dan sinyal masukan kontroler PID

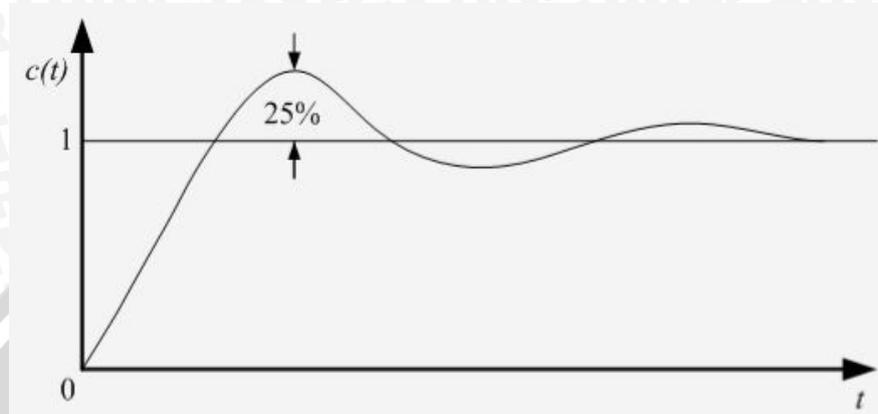
Sumber: Gunterus, 1994:8-11

Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional, integral dan differensial. Gambar 2.5 menunjukkan hubungan tersebut. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P,I dan D. Penyetelan konstanta  $K_p$ ,  $T_i$  dan  $T_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah yang akan memberikan konstribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

### 2.1.1.1 Ziegler-Nichols dan *Hand Tuning* Kontroler PID

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional  $K_p$ , waktu integral  $T_i$ , dan waktu derivatif  $T_d$  berdasarkan karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plan.(Ogata, K., 1997)

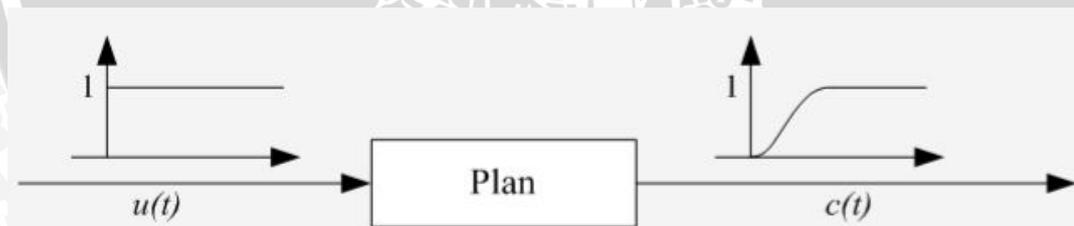
Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step, ditunjukkan dalam Gambar 2.6.



**Gambar 2.6** Kurva Respon Unit Step yang Menunjukkan 25% *Maximum Overshoot*  
Sumber: Ogata, K., 1997

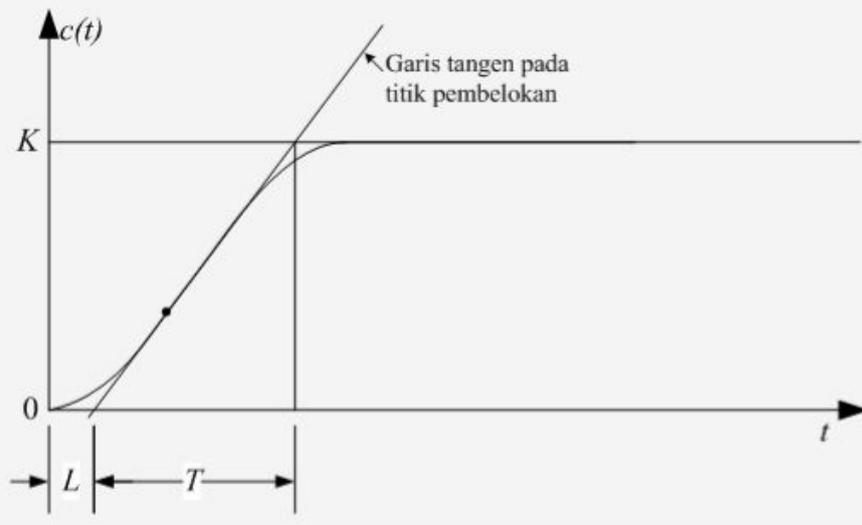
#### 1). Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dalam Gambar 2.7.



**Gambar 2.7** Respon Plant Terhadap Masukan Berupa Unit Step  
Sumber: Ogata, K. 1997

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti ditunjukkan dengan Gambar 2.8. jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku.(Ogata, K., 1997).



**Gambar 2.8** Kurva Respon yang Berbentuk S  
 Sumber: Ogata, K. 1997

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda  $L$  dan konstanta waktu  $T$ . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu  $t$  dan sumbu  $c(t) = K$ , seperti yang telah ditunjukkan dalam gambar 2.8

Fungsi alih  $C(s)/U(s)$  dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \dots\dots\dots(2.8)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari  $K_p$ ,  $T_i$  dan  $T_d$  berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1 (Ogata, K., 1997)

**Tabel 2.1** Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan  
 Sumber: Ogata, K. 1997

Type Kontroler	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	<b>0</b>
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	<b>0</b>

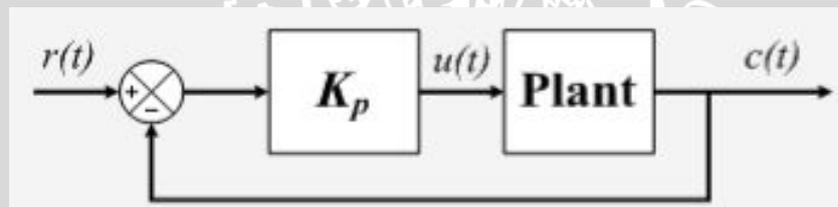


PID	$1,2 \frac{T}{L}$	$2L$	$0,5 L$
-----	-------------------	------	---------

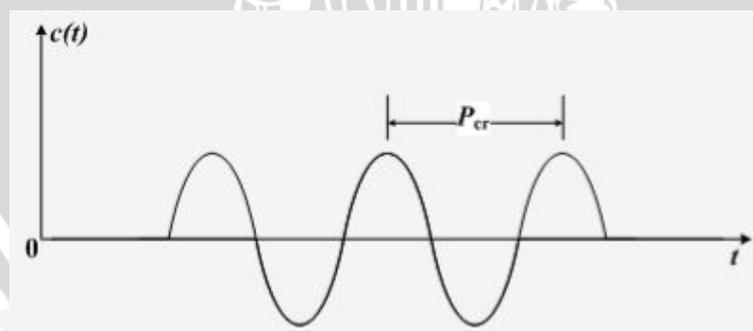
2). Metode Kedua

Dalam metode kedua *ziegler-nichols*, mula-mula yang dilakukan adalah membuat  $T_i = \infty$  dan  $T_d = 0$ . Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis  $K_{cr}$ , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai  $K_p$  manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis  $K_{cr}$  dan periode  $P_{cr}$  dapat ditentukan.

Diagram blok sistem loop tertutup dengan kontroler proporsional ditunjukkan dalam Gambar 2.9 dan untuk osilasi berkesinambungan dengan periode  $P_{cr}$  ditunjukkan dalam Gambar 2.10.



Gambar 2.9 Sistem Loop Tertutup dengan Kontroler Proporsional  
 Sumber: Ogata, K., 1997



Gambar 2.10 Osilasi Berkesinambungan dengan Periode  $P_{cr}$   
 Sumber : Ogata, K., 1997

Ziegler dan Nichols menyarankan penyetelan nilai parameter  $K_p, T_i, T_d$  dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.2. (Ogata, K., 1997)

**Tabel 2.2.** Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain* Kcr dan *Critical Period* Pcr  
Sumber: Ogata, K., 1997

Tipe Kontroler	$K_p$	$T_i$	$T_d$
P	0.5 Kcr	$\infty$	0
PI	0.45 Kcr	$\frac{1}{1,2} P_{cr}$	0
PID	0.60 Kcr	0.5 Pcr	0.125 Pcr

Selain dengan menggunakan Ziegler-Nichols, kontroler PID dapat di *tuning* dalam beberapa cara, antara lain *loop tuning*, metode analitis, optimisasi, *pole placement*, *auto tuning*, dan *hand tuning* (Smith, 1979; Astrom & Hagglund, 1995). Pada skripsi ini digunakan cara *hand tuning* untuk menentukan besar  $K_p$ ,  $K_i$ , dan  $K_d$ . Hal ini dilakukan karena ada kendala untuk melakukan cara lain yang disebutkan di atas. Kendala tersebut adalah tidak dapat melihat respon motor secara langsung karena tidak digunakannya sensor untuk mengukur kecepatan motor saat system berjalan. Selain itu tidak adanya model matematis dari motor membuat cara analitis sulit untuk dilakukan.

Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut:

1. Melepaskan kontroler integral dan deferensial dengan memberikan nilai  $K_i = 0$  dan  $K_d = 0$ .
2. Mengatur nilai  $K_p$  hingga didapatkan respons yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
3. Dengan terus menaikkan nilai  $K_p$ , nilai dari  $K_d$  dinaikkan untuk mengurangi *overshoot* yang terjadi.
4. Naikkan nilai  $K_i$  untuk mengurangi *offset*.

Keuntungan dari *hand tuning* adalah prosedur di atas dapat dilakukan dengan segera, *online* dan dapat melihat dengan cepat respons system setelah perubahan  $K_p$ ,  $K_i$ , dan  $K_d$ . Kerugian dari cara ini adalah kesulitan untuk melihat apakah *setting* akhir dari kontroler merupakan nilai optimal atau tidak (Jantzen, 2001).

## 2.2 Arduino Uno

Arduino Uno adalah *board* mikrokontroler berbasis ATmega328. Memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *board* Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.

Uno berbeda dengan semua *board* sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan *board* sebelumnya yang menggunakan chip FTDI *driver* USB-to-serial. Arduino Uno ditampilkan pada gambar 2.11.



**Gambar 2.11** Tampak Depan Arduino Uno

Sumber : <http://arduino.cc/en/Main/arduinoBoardUno>

Nama “Uno” berarti *satu* dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. *Uno* adalah yang terbaru dalam serangkaian *board* USB Arduino, dan sebagai model referensi untuk platform Arduino, untuk perbandingan dengan versi sebelumnya,

### 2.2.1 Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal (otomatis). Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan

*plug jack* pusat-positif ukuran 2.1mm konektor POWER. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin header dari konektor POWER. Kisaran kebutuhan daya yang disarankan untuk *board uno* adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari 7 volt kemungkinan pin 5v *uno* dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *board uno*.

Pin listrik adalah sebagai berikut:

N. Tegangan masukan kepada *board* Arduino ketika itu menggunakan sumber daya eksternal (sebagai pengganti dari 5 volt koneksi USB atau sumber daya lainnya) 5V. Catu daya digunakan untuk daya mikrokontroler dan komponen lainnya. Sebuah pasokan 3,3 volt dihasilkan oleh regulator on-board.

GND. Ground pin.

### 2.2.2 Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk bootloader), 2 KB dari SRAM dan 1 KB EEPROM

### 2.2.3 Input dan Output

Masing-masing dari 14 pin digital di Uno dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode* (), *digitalWrite* (), dan *digitalRead* (), beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara default terputus) dari 20-50 kOhm. Selain itu, beberapa pin memiliki fungsi khusus:

- a. *Serial*: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.
- b. *Eksternal menyela*: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai. *PWM*: 3, 5, 6, 9, 10, dan 11. Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite* ().
- c. *SPI*: 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.

- d. *LED*: 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin bernilai nilai HIGH, LED on, ketika pin bernilai LOW, LED off.
- e. Uno memiliki 6 masukan analog, berlabel A0 sampai dengan A5, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda). Selain itu, beberapa pin memiliki fungsi khusus:
  1. *I2C*: A4 (*SDA*) dan A5 (*SCL*). Dukungan I2C (TWI) komunikasi menggunakan perpustakaan *wire*.
  2. *Aref*. Tegangan referensi (0 sampai 5V saja) untuk *input* analog. Digunakan dengan fungsi *analogReference* ().
  3. *Reset*. Bawa baris ini LOW untuk me-reset mikrokontroler.

### 2.2.1 Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi serial melalui USB dan sebagai port virtual com untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada Windows diperlukan, sebuah file '.inf'. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari *board* Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *SoftwareSerial library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada *board* Uno's. ATmega328 juga mendukung I2C (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan Kawat untuk menyederhanakan penggunaan bus I2C.

### 2.3 Motor DC (*Direct Current*)

Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorentz, yaitu membangkitkan fungsi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah

konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

$$F = B.I.L \text{ (Newton)} \quad (2.9)$$

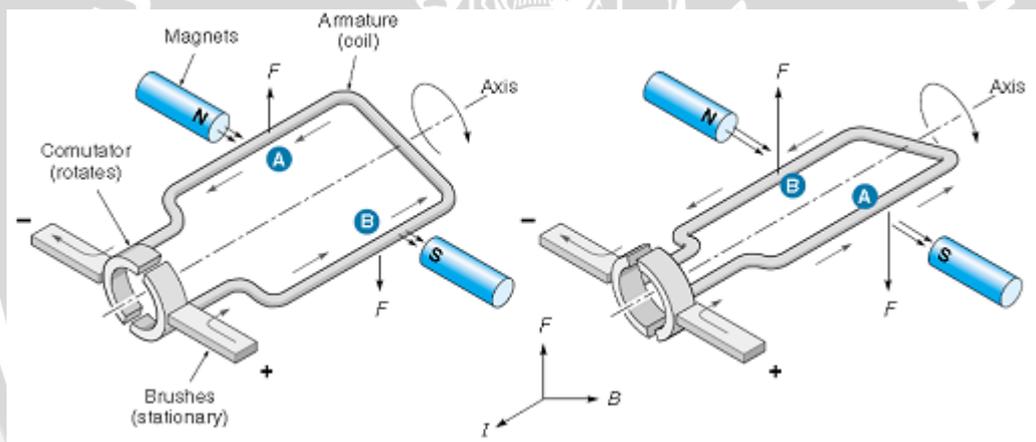
Dimana :

$B$  = kerapatan fluks magnet (weber)

$L$  = panjang konduktor (meter)

$I$  = arus listrik (ampere)

(Soemarwanto,1999)



**Gambar 2.12** Cara Kerja Motor DC

Sumber : Kilian, 1996

Gambar 2.12 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a-b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*commutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan

kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertical.

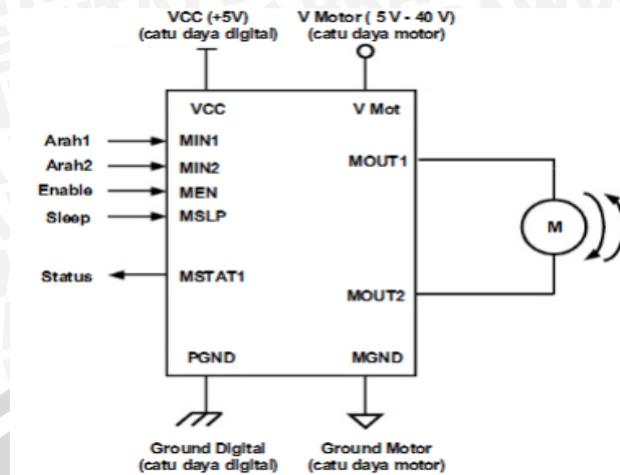
Bagian *commutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam gambar 2.12 arah fluks magnetik adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah. Gaya-gaya yang bekerjasama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar  $180^\circ$  arah arus berbalik, pada saat itu penghantar a dan b bertukar tempat. Akibatnya arah gerak putaran tidak berubah. us I2C.

#### 2.4 Modul Pengendali Motor DC EMS 5A *H-Bridge*

Modul pengendali motor DC yang digunakan adalah modul EMS 5A *H-Bridge* ditunjukkan pada gambar 2.13. Secara garis besar, fungsi modul pengendali motor adalah untuk mengendalikan arah dan kecepatan putaran motor DC sesuai instruksi kendali dari mikrokontroler pengguna. Gambar koneksi dari modul pengendali motor DC EMS 5A *H-bridge* ditunjukkan pada gambar 2.14.



**Gambar 2.13** Pengendali motor DC menggunakan EMS 5A *H-Bridge* Koneksi Modul  
EMS 5A *H-Bridge*



Gambar 2.14 Koneksi Modul EMS 5A H-Bridge

## 2.5 Soil Moisture Sensor YL69/LM393

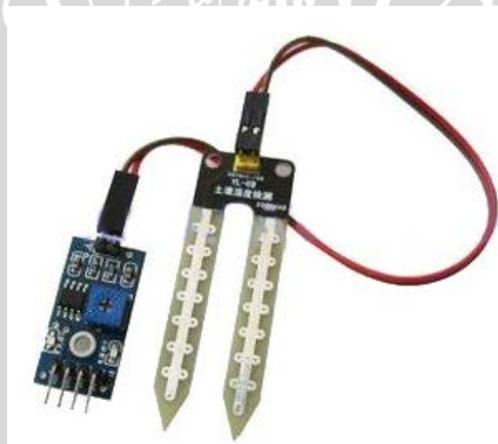
Sensor adalah sebuah piranti yang dapat mengubah besaran suhu menjadi besaran tegangan. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian dikelola oleh kontroler sebagai otaknya. Sensor dalam teknik pengukuran dan pengaturan secara elektronik berfungsi mengubah besaran fisik seperti temperatur, gaya, dan kecepatan putaran menjadi besaran listrik yang proporsional.

Modul pendeteksi kelembapan / kadar air dalam tanah (**Soil Moisture sensor**) yang ditunjukkan pada Gambar 2.15 ini menggunakan moisture probe tipe YL-69 yang diproses IC pembanding *offset* rendah LM393 (*low offset voltage comparator* dengan *offset* masukan lebih rendah dari 5mV) yang sangat stabil dan presisi.

Sensor ini bisa digunakan untuk mendeteksi kadar air dalam tanah, yang kemudian bisa menjadi acuan dalam sistem pengairan / penyiraman tanaman secara otomatis. Untuk pendeteksian secara presisi menggunakan mikrokontroler / Arduino, sensor ini dapat menggunakan pin keluaran analog (sambungkan dengan pin ADC / *Analog-Input* pada mikrokontroler) yang akan memberikan nilai kelembapan pada skala 0V (relatif terhadap GND) hingga Vcc (tegangan catu daya). Modul pemroses dapat menggunakan catu daya antara 3,3 Volt hingga 5 Volt sehingga fleksibel untuk digunakan pada berbagai macam *microcontroller / development board*.

Spesifikasi *Soil Moisture Sensor YL69/LM393*:

- Sensitivitas dapat diatur melalui potensiometer
- Catu daya : 3,3V - 5V DC
- Keluaran tipe analog dapat dibaca sebagai representasi linear akurat dari tingkat kelembapan (resistansi) yang terdeteksi
- Keluaran tipe digital akan bernilai logika HIGH saat kelembapan rendah (tanah kering), atau sebaliknya bernilai LOW saat kelembapan terdeteksi melewati ambang batas / *moisture threshold*.
- Ukuran PCB modul sebesar 30 x 16 mm
- Panjang elemen pendeteksi kelembapan / *moisture sensing probe* adalah 5cm.
- Tersedia indikator LED untuk kondisi nyala (*power indicator*, LED berwarna merah) dan status keluaran (*digital switching output indicator*, LED berwarna hijau)
- Pemroses menggunakan IC komparator LM393



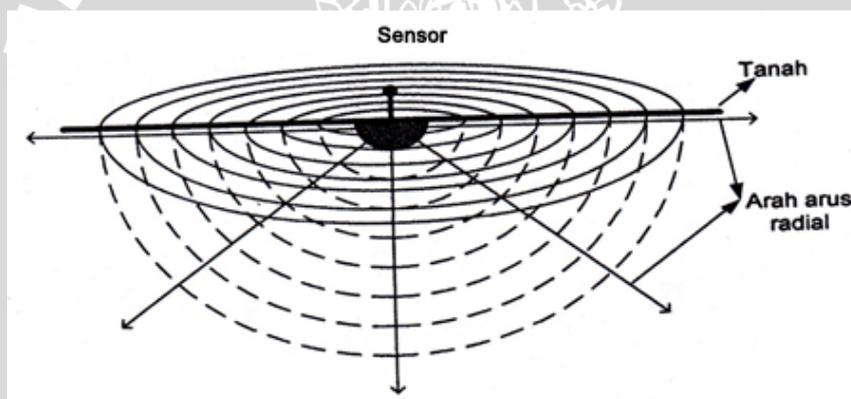
**Gambar 2.15** *Soil Moisture Sensor YL69/LM393*

Umumnya resistansi sensor dan resistansi kontak nilainya kecil dan dapat diabaikan dengan resistansi bagian tanah di sekitar sensor. Hal tersebut dapat diabaikan apabila sensor bebas dari minyak maupun cat dan kontak antara tanah dan sensor adalah sempurna (tidak ada rongga udara). Dengan demikian resistansi

yang paling menentukan harga resistansi sistem pentanahan adalah resistivitas tanah itu sendiri.

Komponen yang mempengaruhi resistivitas tanah adalah jenis tanah, komposisi kimia yang terkandung dalam tanah, konsentrasi garam yang terlarut dalam air yang berada di tanah, kelembaban udara, temperatur tanah, ukuran partikel tanah serta penyebaran partikel tersebut di dalam tanah, kepadatan dan tekanan tanah [G.F. Tagg, 1964; 4]

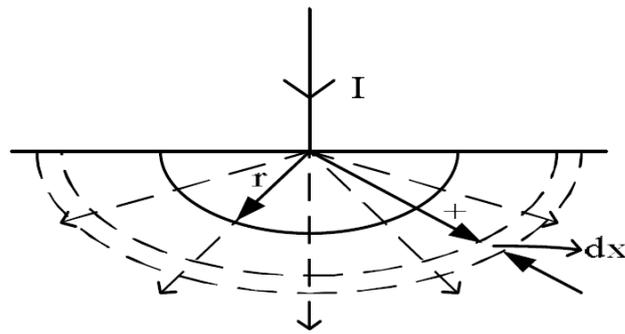
Sebuah sistem pentanahan membentuk konduktor pentanahan yang umumnya dimodelkan sebagai sebuah setengah lingkaran (*hemisphere*), setengah *ellips* atau setengah tabung dengan alas berupa setengah bola. Sensor ini digambarkan sebagai konduktor yang terdiri dari lapisan berupa sel-sel tanah yang tebalnya sama seperti pada Gambar 2.16 dimana penanaman satu *Soil Moisture Sensor* tegak lurus dengan permukaan tanah.



Gambar 2.16 Sel-sel tanah sebagai konduktor

Sumber: Mil-HDBK-419A, 1987

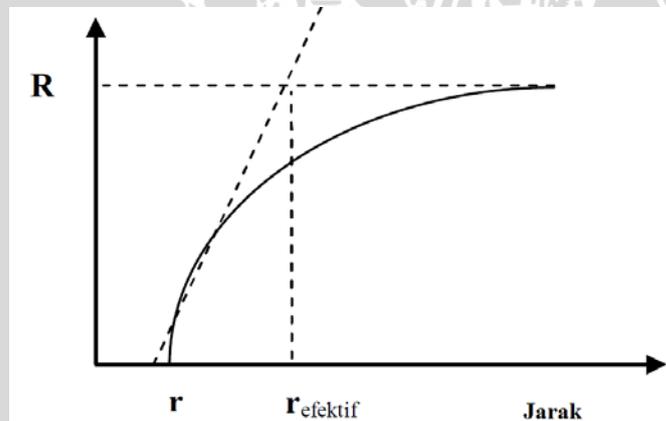
Arus yang mengalir dari pentanahan tersebut akan melintasi sel-sel ini ke semua arah. Sel tanah yang terdekat dengan batang sensor mempunyai permukaan paling kecil sehingga memberikan resistansi paling besar. Bila jarak dari sensor bertambah, maka luasan ini juga akan membesar. Pada beberapa titik yang menentukan jarak tertentu dari sel tanah tersebut, penambahan sel secara signifikan tidak menambah nilai resistansi tanah di sekitar *Soil Moisture Sensor*. Hal ini diketahui sebagai daerah resistansi efektif dan jarak ini ditentukan oleh kedalaman penanaman dan diameter *Soil Moisture Sensor* yang dipakai.



**Gambar 2.17** Distribusi arus di dalam tanah melalui konduktor *hemisphere*

Sumber: G.F. Tagg, 1964: 90

Pada gambar 2.17 mendeskripsikan arus I mengalir ke dalam sel tanah melalui konduktor pentanahan yang dimodelkan *hemisphere* dan mengalir ke semua arah (radial) secara seragam. Semua permukaan sel tanah yang dilewati arus I tersusun secara seri. Jika jarak dari *Soil Moisture Sensor* bertambah maka luas permukaan sel tanah (*hemisphere*) juga meningkat, sedangkan nilai resistansinya perlahan berkurang sampai nilai resistansinya konstan yaitu pada saat mencapai nilai resistansi efektif ( $r_{\text{efektif}}$ ) seperti yang ditunjukkan pada Gambar 2.18.



**Gambar 2.18** Grafik fungsi resistansi terhadap jarak

Sumber: G.F. Tagg, 1964: 91

Jika sel individual pada radius  $x$ , mempunyai lapisan tipis setebal  $dx$ , mempunyai resistansi  $dR$  yang dinyatakan:

$$dR = \frac{\rho dx}{2\pi x^2} \tag{2.10}$$

Integrasi dari  $r$  menuju titik  $r_1$  menghasilkan:



$$R = \frac{\rho}{2\pi} \left( \frac{1}{r} - \frac{1}{r_1} \right) \quad (2.11)$$

Bila  $r_1$  berada dijauh tak berhingga ( $r_1 = \infty$ ), maka rumusan diatas menjadi:

$$R = \frac{\rho}{2\pi r} \quad (2.12)$$

Dengan R adalah resistansi, r adalah jari-jari luasan tanah, dan  $\rho$  merupakan massa dan kerapatan tanah.

**Tabel 2.3** Nilai resistivitas beberapa jenis tanah

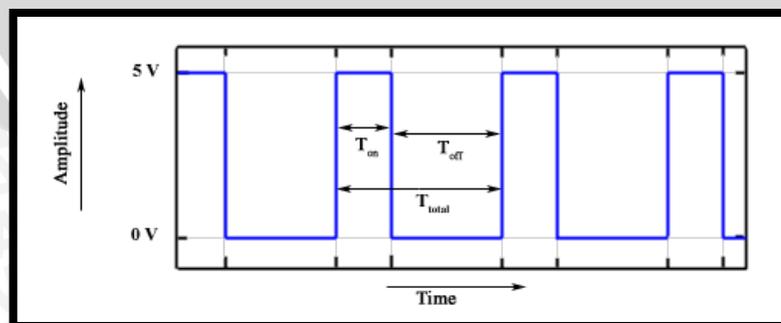
Sumber: G.F. Tagg, 1964: 4

Jenis Tanah	Resistivitas dalam (ohm/cm)
Tanah liat, tanah kebun, dll	500 – 5.000
Tanah liat	800 – 5.000
Campuran tanah liat, pasir dan kerikil	4.000 – 25.000
Pasir dan kerikil	6.000 – 10.000
Batu tulis, pasir berbatu, dll	1.000 – 50.000
Batu Karang	20.000 – 1.000.000

## 2.6 PWM (*Pulse Width Modulation*)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0% - 100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (*Pulse Width Modulation*) secara umum dapat dilihat dalam Gambar 2.19 berikut:



**Gambar 2.19** Sinyal PWM Secara Umum

Sumber: electronics-scheme.com

$$\text{Duty Cycle} = \frac{T_{on}}{T} \times 100\% (\%) \quad (2.13)$$

Dengan:

$T_{on}$  = Periode logika tinggi

$T$  = Periode Keseluruhan

$$V_{dc} = \text{duty cycle} \times V_{cc} (V) \quad (2.14)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

$$f_{On} = \frac{f_{clk I/O}}{N \cdot 256} \quad (2.15)$$

UNIVERSITAS BRAWIJAYA

