

## BAB III

### TEORI PENUNJANG

Pada bab tiga penulis menjelaskan tentang teori penunjang kerja praktek yang telah dikerjakan.

#### 3.1 Mikrokontroler ATmega16

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur *Reduced Instruction Set Computer (RISC)*. Hampir semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 register *general-purpose*, *timer/counter* fleksibel dengan mode *compare*, *interrupt internal* dan *eksternal*, serial UART, *programmable watchdog timer*, dan *mode power saving*, *Analog to Digital Converter (ADC)* dan *Pulse With Modulation (PWM)* internal.

AVR juga mempunyai *In-system programmable flash on-chip* yang mengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan *Serial Peripheral Interface (SPI)*. ATmega16. ATmega16 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya *versus* kecepatan proses. Beberapa keistimewaan dari AVR ATmega16 antara lain:

##### 1. *Advanced RISC Architecture*

- *130 Powerful Instructions—Most Single Clock Cycle Execution*
- *32 x 8 General Purpose Fully Static Operation*

- *Up to 16 MIPS Throughput at 16 MHz*
- *On-chip 2-cycle Multiplier*

### **2. Nonvolatile Program and Data Memories**

- *8K Bytes of In-System Self-Programmable Flash*
- *Optional Boot Code Section with Independent Lock Bits*
- *512 Bytes EEPROM*
- *512 Bytes Internal SRAM*
- *Programming Lock for Software Security*

### **3. Peripheral Features**

- *Two 8-bit Timer/Counters with Separate Prescalers and Compare Mode*
- *Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes*
- *One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode*
- *Real Time Counter with Separate Oscillator*
- *Four PWM Channels*
- *8-channel, 10-bit ADC*
- *Byte-oriented Two-wire Serial Interface*
- *Programmable Serial USART*

### **4. Special Microcontroller Features**

- *Power-on Reset and Programmable Brown-out Detection*
- *Internal Calibrated RC Oscillator*
- *External and Internal Interrupt Sources*

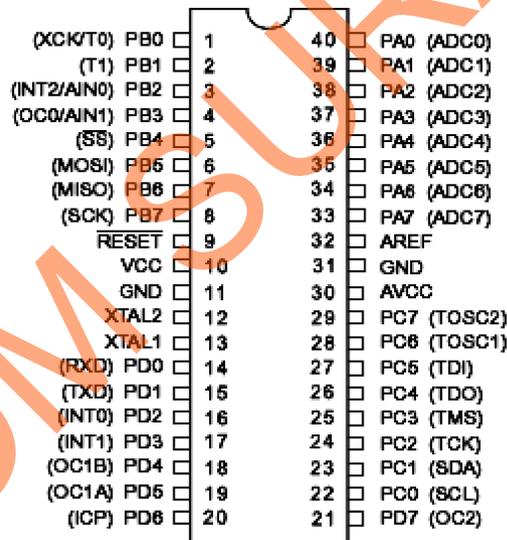
- *Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Powerdown, Standby and Extended Standby*

### 5. I/O and Package

- *32 Programmable I/O Lines*
- *40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MFL*

### 6. Operating Voltages

- *2.7-5.5V for ATmega16L*
- *4.5-5.5V for ATmega16*



Gambar 3.1 Mikrokontroler ATmega16

Sumber: ATMEL, 2011a

Pin-pin pada ATmega16 sebanyak 40-pin DIP (*dual inline package*) ditunjukkan oleh gambar 3.1. Guna memaksimalkan performa, AVR menggunakan arsitektur *Harvard* (dengan memori dan bus terpisah untuk program dan data).

### 3.1.1 Port Sebagai Input/Output Digital

ATmega16 mempunyai empat buah *port* yang bernama *PortA*, *PortB*, *PortC*, dan *PortD*. Keempat *port* tersebut merupakan jalur *bidirectional* dengan pilihan *internal pull-up*. Tiap *port* mempunyai tiga buah register bit, yaitu  $DDx_n$ ,  $PORTx_n$ , dan  $PINx_n$ . Huruf 'x' mewakili nama huruf dari *port* sedangkan huruf 'n' mewakili nomor bit. Bit  $DDx_n$  terdapat pada I/O address  $DDR_x$ , bit  $PORTx_n$  terdapat pada I/O address  $PORT_x$ , dan bit  $PINx_n$  terdapat pada I/O address  $PIN_x$ . Bit  $DDx_n$  dalam register *Data Direction Register* ( $DDR_x$ ) menentukan arah pin. Bila  $DDx_n$  diset 1 maka  $P_x$  berfungsi sebagai pin *output*.

Bila  $DDx_n$  diset 0 maka  $P_x$  berfungsi sebagai pin *input*. Bila  $PORTx_n$  diset 1 pada saat pin terkonfigurasi sebagai pin *input*, maka resistor *pull-up* akan diaktifkan. Untuk mematikan resistor *pull-up*,  $PORTx_n$  harus diset 0 atau pin dikonfigurasi sebagai pin *output*. Pin *port* adalah *tri-state* setelah kondisi reset. Bila  $PORTx_n$  diset 1 pada saat pin terkonfigurasi sebagai pin *output* maka pin *port* akan berlogika 1. Dan bila  $PORTx_n$  diset 0 pada saat pin terkonfigurasi sebagai pin *output* maka pin *port* akan berlogika 0.

Saat mengubah kondisi *port* dari kondisi *tri-state* ( $DDx_n=0$ ,  $PORTx_n=0$ ) ke kondisi *output high* ( $DDx_n=1$ ,  $PORTx_n=1$ ) maka harus ada kondisi peralihan apakah

itu kondisi *pull-up enabled* ( $DDxn=0, PORTxn=1$ ) atau kondisi *output low* ( $DDxn=1, PORTxn=0$ ).

Biasanya, kondisi *pull-up enabled* dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah *pull-up*. Jika ini bukan suatu masalah, maka bit PUD pada register SFIOR dapat diset 1 untuk mematikan semua *pull-up* dalam semua *port*.

Peralihan dari kondisi *input* dengan *pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama. Kita harus menggunakan kondisi *tri-state* ( $DDxn=0, PORTxn=0$ ) atau kondisi *output high* ( $DDxn=1, PORTxn=0$ ) sebagai kondisi transisi.

Tabel 3.1 Konfigurasi pin *port*

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (HI-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (HI-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Sumber : IT Telkom *Library*, 2008

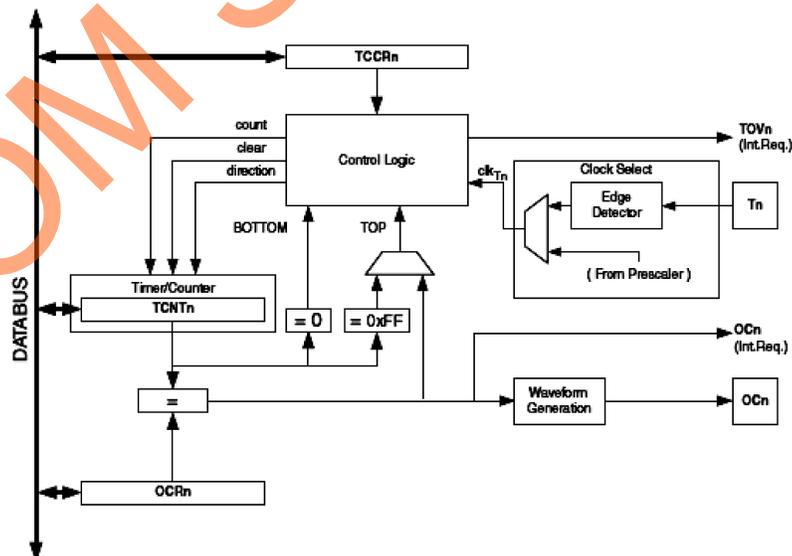
Bit 2 – PUD : *Pull-up Disable* Bila bit diset bernilai 1 maka *pull-up* pada *port* I/O akan dimatikan walaupun register DDxn dan PORTxn dikonfigurasi untuk menyalakan *pull-up* ( $DDxn=0, PORTxn=1$ ).

### 3.1.2 Timer

*Timer/counter* adalah fasilitas dari ATmega16 yang digunakan untuk perhitungan pewaktuan. Beberapa fasilitas *channel* dari *timer/counter* antara lain: *counter channel* tunggal, pengosongan data *timer* sesuai dengan data pembanding, bebas *glitch*, tahap yang tepat PWM, pembangkit *frekwensi*, *event counter external*..

#### 3.1.2.1 Gambaran Umum

Gambar diagram *block timer/counter* 8 bit ditunjukkan pada gambar 3.2 Untuk penempatan pin I/O telah dijelaskan pada bagian I/O di atas. CPU dapat diakses register I/O, termasuk dalam pin-pin I/O dan bit I/O. *Device* khusus register I/O dan lokasi bit terdaftar pada deskripsi *timer/counter* 8 bit.

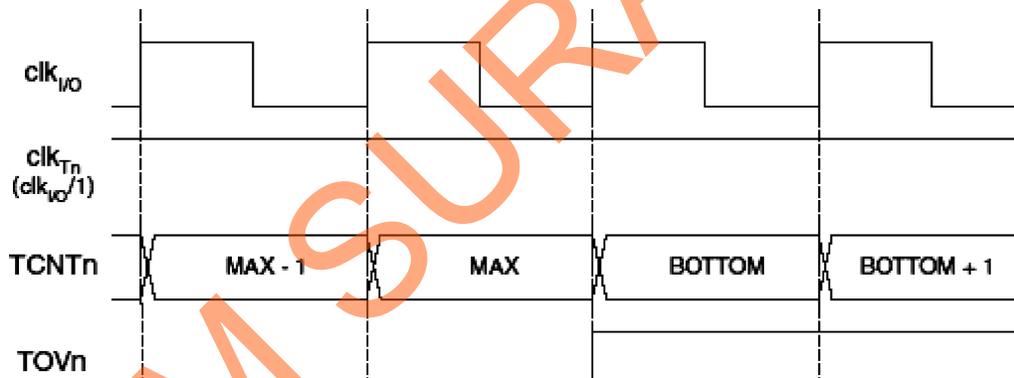


Gambar 3.2 Blok diagram *timer/counter*

Sumber : Atmel, 2011b

### 3.1.2.2 Timing Diagram Timer/Counter

Timer/counter didesain sinkron clock timer ( $clk_{T0}$ ) oleh karena itu ditunjukkan sebagai sinyal *enable clock* pada gambar 3.3. Gambar ini termasuk informasi ketika *flag interrupt* dalam kondisi set. Data *timing* digunakan sebagai dasar dari operasi *timer/counter*.



Gambar 3.3 Timing diagram timer/counter, tanpa prescaling

Sumber : Atmel, 2011c

Sesuai dengan gambar 3.3 timing diagram timer/counter dengan prescaling maksudnya adalah counter akan menambahkan data counter ( $TCNTn$ ) ketika terjadi pulsa clock telah mencapai 8 kali pulsa dan sinyal clock pembagi aktif clock dan ketika telah mencapai nilai maksimal maka nilai  $TCNTn$  akan kembali ke nol. Dan kondisi *flag timer* akan aktif ketika  $TCNTn$  maksimal.

## 3.2 *Relay*

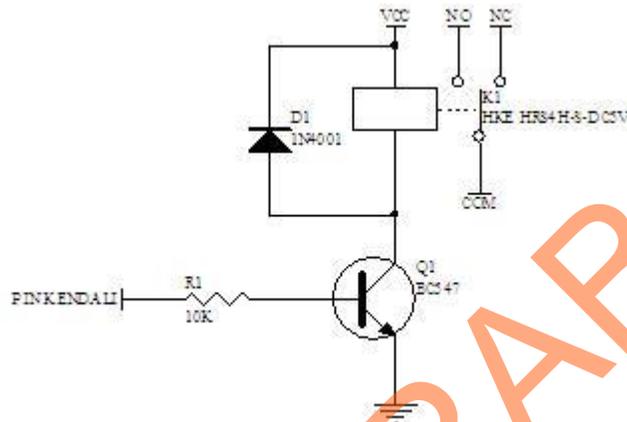
Pengertian *relay* yang ada pada wikipedia (2010) adalah “suatu peranti yang menggunakan elektromagnet untuk mengoperasikan seperangkat kontak saklar. Susunan paling sederhana terdiri dari kumparan kawat penghantar yang dililit pada inti besi. Bila kumparan ini dienergikan, medan magnet yang terbentuk menarik armatur berporos yang digunakan sebagai pengungkit mekanisme saklar”.

### 3.2.1 Prinsip Kerja *Relay*

*Relay* pada dasarnya sama dengan prinsip kerja pada saklar, namun pada *relay* saklar dijalankan secara otomatis dengan memberikan catu daya pada *coil* sebagai tegangan sistem pada *relay*. Ketika *coil* mendapat energi listrik, maka akan timbul gaya elektromagnetik yang akan menarik *armature* yang berpegas yang juga akan menutup kontak yang terhubung pada *output*.

*Relay* memiliki tiga jenis kutub yaitu *common* (kutub acuan), *normally close* (kutub yang dalam keadaan awal terhubung pada *common*), dan *Normally Open* (kutub yang pada awalnya terbuka dan akan terhubung dengan *common* saat kumparan *relay* diberi arus listrik).

Dalam kerja praktek ini *relay* digunakan sebagai *driver motor* tangan pada robot yang dikombinasikan dengan beberapa komponen lain yang membantu kinerja *relay*. Berikut rangkaian *relay*.



Gambar 3.4 Gambar rangkaian *relay*

Sumber : *depokinstrument*, 2010

### 3.3 *Driver H-Bridge*

“Sebuah *h-bridge* adalah suatu pengaturan transistor yang memungkinkan kontrol atas sirkuit motor DC. Artinya, dengan jembatan-H mikrokontroler, *chip*

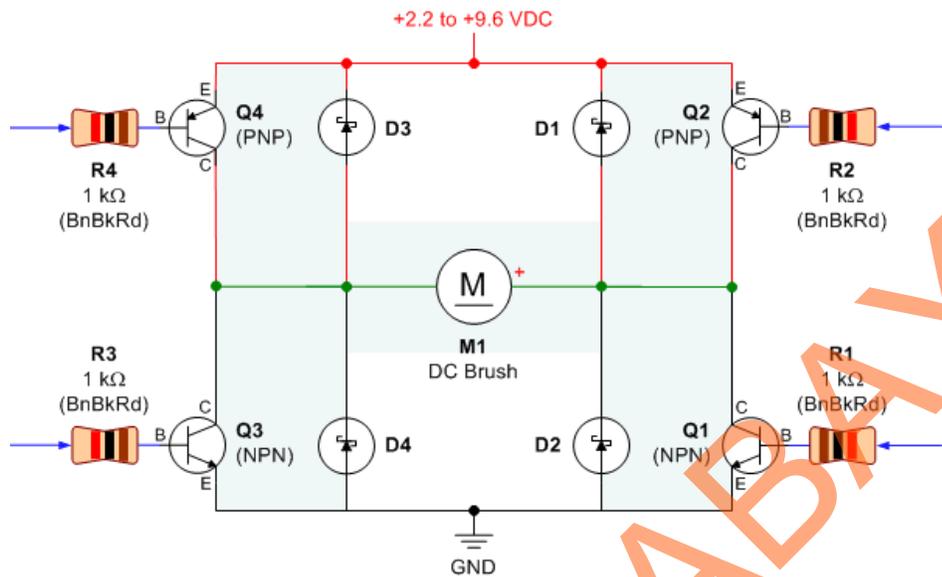
logika, atau remot kontrol elektronik dapat memerintah motor untuk maju, mundur, dan rem” (wikipedia, 2011a).

*Driver h-bridge* juga merupakan salah satu pengendali motor DC dengan arus yang lebih tinggi daripada menggunakan driver motor biasa seperti *L298* atau *L293*.

*Driver h-bridge* dapat mengendalikan motor DC lebih dari 3A.

Dimana besar arus yang dikeluarkan motor DC tergantung besar motor DC dan beban pada robot, maka dari itu *driver h-bridge* membantu menstabilkan arus yang berasal dari tegangan sumber. Berikut rangkaian *driver h-bridge*.

STIKOM SURABAYA



Gambar 3.5 Rangkaian *driver h-bridge*

Sumber : Cook, 2011

3.4 Motor DC

Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Motor DC digunakan pada penggunaan khusus di mana diperlukan penyalan *torque* yang tinggi atau percepatan yang tetap.

Komponen utama motor DC :

1. Kutub medan

Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan *dynamo* yang menggerakkan *bearing* pada ruang di antara kutub medan. Motor DC sederhana memiliki dua kutub medan yaitu kutub utara dan kutub selatan.

2. *Dynamo*

Bila arus masuk menuju *dynamo*, maka arus ini akan menjadi *dynamo* magnet. *Dynamo* yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, *dynamo* berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan *dynamo*.

3. *Commutator*

Kegunaan komponen ini adalah untuk membalikan arah arus listrik dalam *dynamo*. *Commutator* juga membantu dalam transmisi arus antara *dynamo* dan sumber daya.



Gambar 3.6 Motor DC

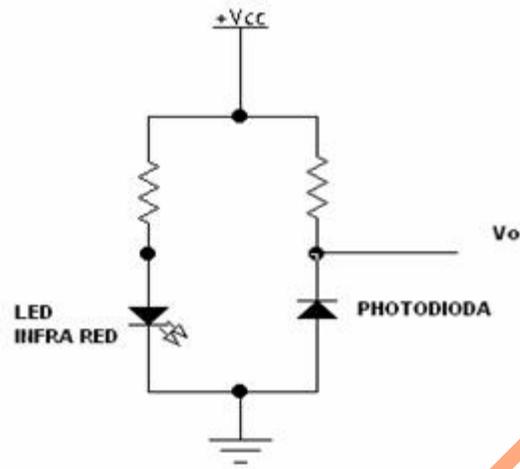
Sumber : Artha, 2011

### 3.5 Rotary Encoder

Fungsi *rotary encoder* pada robot KRI adalah untuk menghitung jumlah putaran yang telah terjadi pada motor ketika robot berbelok. Pengertian dari *rotary encoder* seperti adalah “*Rotary encoders* digunakan dalam berbagai aplikasi yang membutuhkan sensor perputaran yang presisi” (wikipedia, 2011b).

### 3.6 Sensor Cahaya

Sensor cahaya ini terdiri dari sebuah LED *ultrabright* dan *photodiode*. Di mana LED *ultrabright* digunakan sebagai *transmitter* dan *photodiode* sebagai *receiver*. Antara kaki *photodiode* dan resistor akan mengeluarkan tegangan sesuai dengan *input* tegangan yang diterima oleh *photodiode* dari pantulan LED.



Gambar 3.7 Rangkaian sensor cahaya

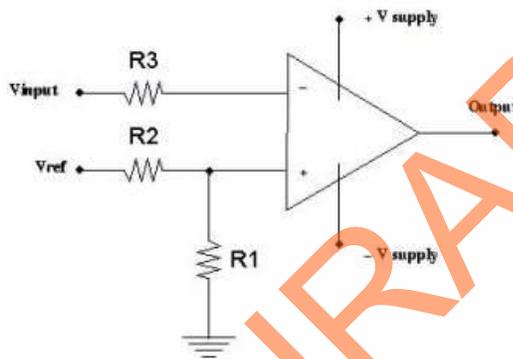
Sumber : Kuncoro, 2009

### 3.6.1 Photodiode

“*Photodiode* adalah jenis photodetektor yang mampu mengubah cahaya menjadi tegangan” (wikipedia, 2011c).

### 3.6.2 Komparator

Komparator adalah sebuah rangkaian yang dapat membandingkan besar tegangan masukan. Komparator biasanya menggunakan Op-Amp atau ic LM339 sebagai piranti utama dalam rangkaian.



Gambar 3.8 Rangkaian komparator

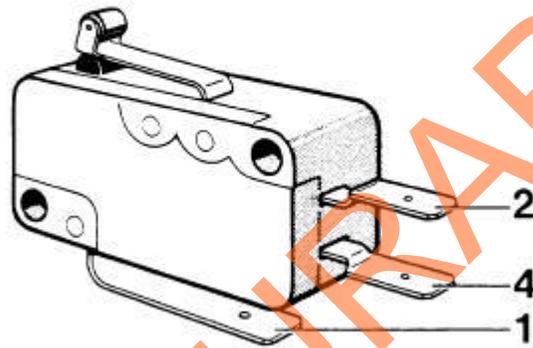
Sumber : Atmel mikrokontroler, 2010

### 3.7 Limit Switch

Pada dasarnya prinsip kerja limit *switch* sama dengan prinsip kerja *relay* yaitu sesuai dengan sakelar. Di mana pada kaki-kakinya memiliki 3 *inputan* yaitu *normally*

*close*, *normally open*, dan *common*. Pada keadaan awal kaki *normally close* tersambung dengan kaki *common* dan jika ditekan, maka kaki *normally open* akan tersambung dengan *common*. Adapun rangkaian limit switch seperti yang ditunjukkan pada gambar berikut.

### Late Limit Switch



Gambar 3.9 Limit switch

Sumber : Garage, 1998

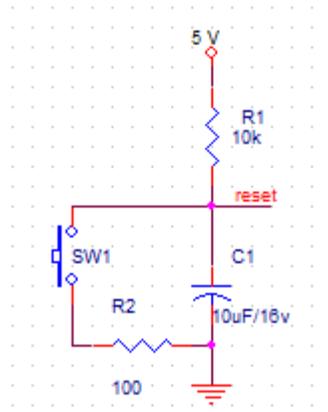
### 3.8 Rangkaian Reset

Chandra MDE (2011) menjelaskan sebagai berikut :

Rangkaian reset adalah Pin reset pada mikrokontroler ATMEL AVR adalah aktif low. Jika sebuah sinyal low diaplikasikan pada pin ini, maka mikrokontroler akan direset. Peresetan sistem dilakukan dengan tujuan :

1. Untuk 'melepas' semua pin (kecuali pin-pin XTAL) untuk masuk ke keadaan *tri-state*, menginisialisasi semua register I/O, dan mereset program *counter* (PC=0).
2. Untuk memasuki mode pemrograman paralel.

Jalur reset memiliki resistor *pull-up* internal berukuran 100k-500k ohm. Secara teori, resistor *pull-up* tersebut berfungsi menahan pin reset pada logika *high* dan tidak mengambang. Resistor *pull-up* juga berfungsi sebagai penahan dari *noise* yang diakibatkan oleh pengaruh lingkungan luar yang tinggi.



Gambar 3.10 Rangkaian reset AVR