

## BAB II

### LANDASAN TEORI

#### 2.1 Mobile Robot

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu. Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan “*search and rescue*”, dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, dan pemotong rumput.

Ketika para teknisi siap untuk mencoba robot berjalan kembali, mereka mulai dengan *heksapoda* dan *platform* berkaki banyak lainnya. Robot-robot tersebut meniru serangga dan *arthropoda* dalam bentuk dan fungsi. Tren menuju jenis badan tersebut menawarkan fleksibilitas yang besar dan terbukti dapat beradaptasi dengan berbagai macam lingkungan, tetapi biaya dari penambahan kerumitan mekanikal telah mencegah pengadopsian oleh para *user*. Dengan lebih dari empat kaki, robot-robot ini stabil secara statis yang membuat mereka bekerja lebih mudah.

Ketika para pencipta robot pertama kali mencoba meniru manusia dan hewan, mereka menemukan bahwa hal tersebut sangatlah sulit; membutuhkan tenaga penghitungan yang jauh lebih banyak dari yang tersedia pada masa itu.

Jadi, penekanan perkembangan diubah ke bidang riset lainnya. Robot sederhana beroda digunakan untuk melakukan eksperimen dalam tingkah laku, navigasi, dan perencanaan jalur. Teknik navigasi tersebut telah berkembang menjadi sistem kontrol robot otonom yang tersedia secara komersial, contoh paling mutakhir dari sistem kontrol navigasi otonom yang tersedia sekarang ini termasuk sistem navigasi berdasarkan-laser dan *Visual Simultaneous Localization and Mapping* (VSLAM) dari ActivMedia Robotics dan Evolution Robotics.

Beberapa jenis Robot:

### **2.1.1. Robot Mobil (Bergerak) yang bisa berpindah tempat.**

*Mobile robot* adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. *Mobile robot* ini sangat disukai bagi orang yang mulai mempelajari robot. Hal ini karena membuat robot mobil tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah *mobile robot* minimal diperlukan pengetahuan tentang *microcontroller* dan sensor-sensor elektro

### **2.1.2. Robot Manipulator ( tangan )**

Robot ini hanya memiliki satu tangan seperti tangan manusia yang fungsinya untuk memegang atau memindahkan barang, contoh robot ini adalah robot las di industri mobil, robot perakitan elektronik dan lain-lain.

### **2.1.3. Robot Humanoid**

Robot yang memiliki kemampuan menyerupai manusia, baik fungsi maupun cara bertindak, contoh robot ini adalah Ashimo yang dikembangkan oleh Honda.

#### 2.1.4. Robot Berkaki

Robot ini memiliki kaki seperti hewan atau manusia, yang mampu melangkah, seperti robot serangga, robot kepiting dll.

#### 2.1.5. Flying Robot (Robot Terbang)

Robot yang mampu terbang, robot ini menyerupai pesawat model yang deprogram khusus untuk memonitor keadaan di tanah dari atas, dan juga untuk meneruskan komunikasi.

#### 2.1.6. Under Water Robot (Robot dalam air)

Robot ini digunakan di bawah laut untuk memonitor kondisi bawah laut dan juga untuk mengambil sesuatu di bawah laut (Irwan, 2011).

### 2.2 ATMEGA8535

ATMega8535 merupakan salah satu *microcontroller* keluarga ATMEL dari perkembangan terakhir, yaitu generasi AVR (*Alf and Vegard's Risc Processor*). *Microcontroller* AVR memiliki RISC (*Reduced Instruction Set Computing*) 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-bit Word) dan sebagian besar instruksi di eksekusi dalam 1 (satu) siklus clock, serta mempunyai kecepatan maksimal 16MHZ. Selain itu, ATMega8535 mempunyai 6 pilihan *modesleep* untuk menghemat daya listrik.

Fitur-fitur yang diberikan oleh *microcontroller* ATMega8535 adalah sebagai berikut:

1. Saluran I/O sebanyak 32 buah, yaitu Port A sampai Port D (port A, B, C, D)

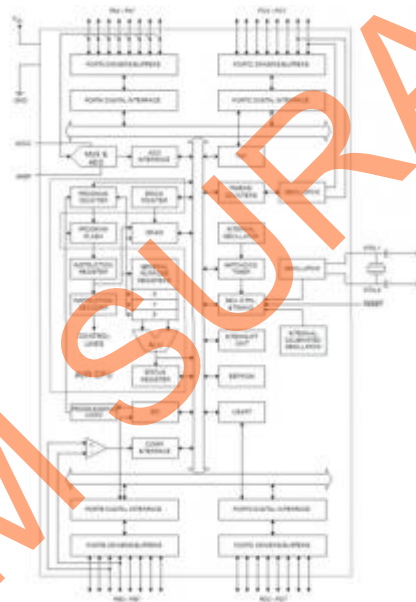
2. ADC (*Analog to Digital Converter*) 10 bit sebanyak 8 chanel.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan, yaitu 2 buah timer/counter 8 bit, dan 1 buah timer/counter 16 bit.
4. CPU yang memiliki 32 buah *register*
5. 131 Instruksi yang hanya membutuhkan 1 siklus *clock*
6. *Watchdog Timer* dengan *osilator internal*
7. Tegangan operasi 2,7 V – 5,5 V
8. Internal SRAM sebesar 512 byte
9. Memori *Flash* sebesar 8 KB dengan kemampuan *Read While Write*.
10. Unit interupsi internal dan eksternal
11. Port antarmuka SPI (*Serial Peripheral Interface*).
12. Kecepatan hampir mencapai 16 MPIS pada Kristal 16 Mhz
13. *Internal downloader* USB AVR (*In-system Programming* dilengkapi *LED programming indicator*)
14. Tidak membutuhkan power tambahan saat melakukan download program
15. EEPROM (*Electrically Erasable Programmable Read Only Memory*). sebesar 512 byte yang dapat diprogram saat operasi.
16. Antarmuka komparator analog.
17. Port USART untuk komunikasi serial.

Beberapa karakteristik ADC Internal Yang terdapat pada *microcontroller*

ATMega8535 adalah:

1. Mudah dalam pengoperasian.
2. Resolusi 10 bit.
3. Memiliki 8 masukan analog.
4. Konversi pada saat CPU sleep.
5. Interrupt waktu konversi selesai

Berikut diagram blok ATMega8535 ditunjukkan pada Gambar 2.1



Gambar 2.1 Blok Diagram ATMEGA8535

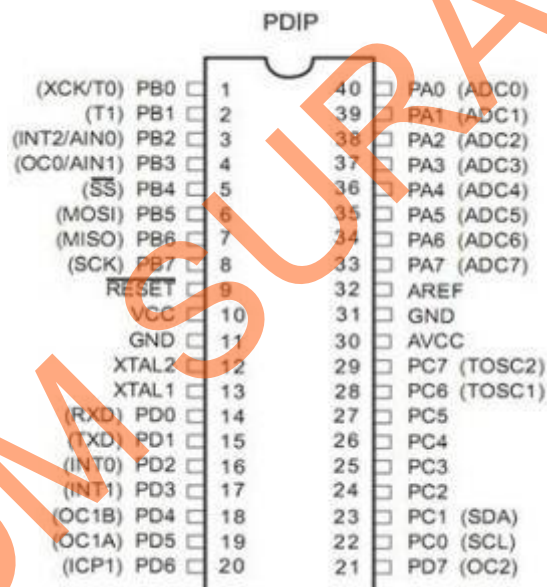
Berikut skema *minimum system* ATMEGA 8535 seperti ditunjukkan pada

Gambar 2.2.



Gambar 2.2 Skema Minimum System ATMEGA8535

Konfigurasi pin ATmega8535, pada Gambar 2.3 berikut ini merupakan susunan kaki standar 40 pin *Microcontroller* AVR ATMEGA 8535.



Gambar 2.3 Konfigurasi Port I/O ATMEGA8535

Berikut ini adalah penjelasan umum susunan kaki dari ATMEGA8535 :

1. VCC merupakan pin masukan positif catu daya. Setiap piranti elektronika digital membutuhkan sumber daya yang umumnya sebesar 5V. Oleh karena itu, biasanya di PCB kit *microcontroller* selalu ada IC regulator 7805

2. GND sebagai pin *ground*
3. Port A (PA0..PA7) merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC
4. Port B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog, dan SPI
5. Port C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog dan *Timer Osilator*
6. Port D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analaog, interupsi eksternal, dan komunikasi serial
7. Reset merupakan pin yang digunakan untuk me-*reset microcontroller*
8. XTAL1 dan XTAL2 sebagai pin masukan *clock* eksternal. Suatu *microcontroller* membutuhkan sumber detak (*clock*) agar dapat mengeksekusi instruksi yang ada di memeori. Semakin tinggi nilai kristalnya, semakin cepat *microcontroller* tersebut
9. AVCC sebagai pin masukan tegangan untuk ADC
10. AREF sebagai pin masukan tegangan referensi

Organisasi memori *microcontroller* ATMega8535 dapat dibagi atas dua bagian berbeda berdasarkan fungsinya dalam menyimpan data program, yaitu memori program dan memori data. Memori program digunakan untuk instruksi yang akan dijalankan oleh *microcontroller*. Memori jenis ini biasanya bertipe *Read Only Memory* (ROM), yang digunakan untuk menyimpan program.

Sedangkan memori data digunakan sebagai tempat penyimpanan data-data yang sedang diakses oleh *microcontroller*.

Memori data terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O, dan 512 byte *Static Random Access Memory* (SRAM) Internal. Register keperluan umum menempati space data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan control terhadap *microcontroller* menempati 64 alamat berikutnya, yaitu mulai dari \$20 sampai \$5F. Register tersebut merupakan register khusus yang dipergunakan untuk mengatur fungsi terhadap berbagai fitur *microcontroller*, seperti *control register*, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. Alamat memori berikutnya digunakan untuk SRAM 512 byte, yaitu pada lokasi \$60 sampai dengan \$25F. Selain itu, AVR ATmega8535 juga memiliki memori data berupa *Electrically Erasable Programmable Read-Only Memory* (EEPROM) 8-bit sebanyak 512 byte. Alamat EEPROM dimulai dari \$000 sampai dengan \$1FF.

Memori program yang terletak dalam *flash Programmable Erasable Read Only Memory* (PEROM) tersusun dalam word atau 2 byte karena setiap instruksi memiliki lebar 16-bit atau 32-bit. Flash PEROM adalah PROM yang dapat ditulis ulang beberapa kali, dan dapat dihapus secara elektrik atau dengan tegangan listrik. AVR ATmega8535 memiliki 4Kbyte x 16-bit *Flash PEROM* dengan alamat mulai dari \$000 sampai dengan \$FFF. AVR tersebut memiliki 12-bit *Program Counter*(PC) sehingga mampu mengamati isi Flash.

Status register adalah register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu intruksi dieksekusi SREG merupakan bagian dari inti pemroses pada *microcontroller*.



1. Bit 7-1: Global Interrupt Enable.

Bit harus diset untuk meng-enable interupsi. Hal ini dilakukan dengan cara mengaktifkan interupsi yang akan digunakan, melalui meng-enable bit control register yang bersangkutan secara individu. Bit akan di-clear apabila terjadi suatu interupsi yang dipicu oleh hardware, dan bit tidak akan mengizinkan terjadinya interupsi, serta akan diset kembali oleh RET1.

2. Bit 6-T: Bit Copy Storage.

Instruksi *Bit Load* (BLD) dan *Bit Store* (BST) menggunakan bit-T sebagai sumber atau tujuan dalam operasi bit. Suatu bit dalam sebuah register *General Purpose Register* (GPR) dapat disalin ke bit T menggunakan instruksi BST, dan sebaliknya bit T dapat disalin kembali ke suatu bit dalam register GPR menggunakan instruksi BLD.

3. Bit 5-h: Half carry flag.

Menunjukkan *Half-carry* pada beberapa operasi aritmatika.

4. Bit 4-s: Sign Bit.

Bit S merupakan hasil operasi *Exclusive OR* (EOR) antara flag-N (negative) dan Flag V (complement dua overflow).

5. Bit 3-V: Two's Complement Overflow Flag.

Bit yang berguna untuk mendukung operasi aritmatika.

6. Bit 2-N: Negative Flag.

Apabila suatu operasi menghasilkan bilangan negative, maka flag-N akan diset.

7. Bit 1-Z: Zero Flag.

Bit akan diset bila hasil operasi yang diperoleh adalah nol.

## 8. Bit 0-C: Carry Flag.

Apabila suatu operasi menghasilkan carry, maka bit akan diset.

Port I/O pada *microcontroller* ATmega8535 dapat difungsikan sebagai input ataupun output dengan keluaran *high* atau *low*. Untuk mengatur fungsi port I/O sebagai input ataupun output, perlu dilakukan setting pada DDR dan Port.

Tabel 2.1 merupakan tabel konfigurasi pengaturan untuk Port I/O.

Tabel 2.1 Konfigurasi pengaturan untuk Port I/O

	DDR bit = 1	DDR bit = 0
Port bit = 1	Output High	Input pull-up
Port bit = 0	Output Low	Input Floating

Logika Port I/O dapat diubah-ubah dalam program secara 8-bit I/O atau hanya bit tertentu. Mengubah sebuah keluaran bit I/O dapat menggunakan perintah cbi (clear bit I/O) untuk menghasilkan output low atau perintah sbi (set bit I/O) untuk menghasilkan output high. Perubahan secara 8-bit I/O dengan perintah in atau out yang menggunakan register bantu. Port I/O sebagai output hanya memberikan arus *sourcing* sebesar 20mA sehingga untuk menggerakkan motor servo perlu diberikan penguat tambahan atau dengan konfigurasi port sebagai *sinking current*, seperti port untuk menyalakan LED yang akan menyala saat port diberikan logika *low*, dan mati saat port diberikan logika *high*. Berikut ini instruksi yang terdapat pada I/O:

### 1. In

Membaca data I/O *Pot* atau *internal peripheral* register (Timers, UART) ke dalam register.

### 2. Out

Menulis data sebuah register ke I/O port atau internal peripheral register.

### 3. Idi (load immediate)

Untuk menulis konstanta ke register sebelum konstanta itu dituliskan ke I/O Port.

### 4. Sbi (set bit in I/O)

Untuk membuat Logika high satu bit I/O register.

### 5. Cbi (clear bit in I/O)

Untuk membuat Logika low satu bit I/O register.

### 6. Sbic (skip if bit in I/O I cleared)

Untuk mengecek apakah bit I/O register clear. Jika clear, skip satu perintah dibawahnya.

### 7. Sbis (skip if bit in I/O is set)

Untuk mengecek apakah bit I/O register set. Jika set, skip satu perintah dibawahnya (Heryanto,2008).

## 2.3 Komunikasi Serial

Ada 2 macam cara komunikasi data serial yaitu sinkron dan asinkron. Pada komunikasi data serial sinkron, *clock* dikirimkan bersama sama dengan data serial, tetapi *clock* tersebut dibangkitkan sendiri-sendiri baik pada sisi pengirim maupun penerima. Sedangkan pada komunikasi serial asinkron tidak diperlukan *clock* karena data dikirimkan dengan kecepatan tertentu yang sama baik pada pengirim / penerima.

Pada IBM PC kompatibel *port* serialnya termasuk jenis asinkron. Komunikasi data serial ini dikerjakan oleh UART (*Universal Asynchronous*

*Receiver Transmitter*). IC UART dibuat khusus untuk mengubah data parallel menjadi data serial dan menerima data serial yang kemudian dirubah lagi menjadi data parallel. IC UART 8250 merupakan salah satunya. Selain berbentuk IC mandiri berbagai macam mikrokontroller juga ada yang dilengkapi dengan UART, misalnya AT89S51/52/53 atau PIC16F877.

Pada komunikasi UART, kecepatan pengiriman data ( atau yang sering disebut dengan *Baud Rate* ) dan fase *clock* pada sisi *transmitter* dan sisi *receiver* harus sinkron.

Untuk itu diperlukan sinkronisasi antara *transmitter* dan *receiver*. Hal ini dilakukan oleh bit “*Start*” dan bit “*Stop*”. Ketika saluran transmisi dalam keadaan *idle*, output UART adalah dalam keadaan logika “1”.

Ketika *transmitter* ingin mengirimkan data, output UART akan diset dulu ke logika “0” untuk waktu satu bit. Sinyal ini pada *receiver* akan dikenali sebagai sinyal “*Start*” yang digunakan untuk menyinkronkan fase *clock*-nya sehingga sinkron dengan fase *clock transmitter*.

Selanjutnya data akan dikirimkan secara serial dari bit yang paling rendah (bit0) sampai bit tertinggi. Selanjutnya akan dikirimkan sinyal “*Stop*” sebagai akhir dari pengiriman data serial.

Sebagai contoh misalnya akan dikirimkan data huruf “A” dalam format *American Standard Code for Information Interchange* (ASCII) (atau sama dengan 41 heksa atau 0100 0001. Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. *Baud rate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/detik). Dalam komunikasi data serial, *baud rate* dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama.

Selanjutnya harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil, atau tanpa paritas), dan jumlah bit “*Stop*” (1, 1 ½, atau 2 bit)( Mujahidin,2009).

## 2.4 Motor DC

Motor DC merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya, memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (mixer, bor listrik, fan angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” dalam dunia industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industri.

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

1. Tegangan dinamo – meningkatkan tegangan dinamo akan meningkatkan kecepatan.
2. Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan rolling mills, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya. Motor DC juga relatif mahal dibanding motor AC.

Hubungan antara kecepatan, flux medan dan tegangan dinamo ditunjukkan dalam persamaan berikut:

$$\text{Gaya elektromagnetik: } E = K\phi N \quad (1)$$

$$\text{Torque: } T = K\phi I_a \quad (2)$$

Dimana:

$E$  = gaya elektromagnetik yang dikembangkan pada terminal dinamo (volt)

$\Phi$  = flux medan yang berbanding lurus dengan arus medan

$N$  = kecepatan dalam RPM (putaran per menit)

$T$  = *torque* elektromagnetik

$I_a$  = arus dinamo

$K$  = konstanta persamaan

Mekanisme kerja untuk seluruh jenis motor secara umum sama

1. Arus listrik dalam medan magnet akan memberikan gaya
2. Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran/*loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
3. Pasangan gaya menghasilkan tenaga putar/ *torque* untuk memutar kumparan.
4. Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban mengacu kepada keluaran tenaga

putar/ *torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan kedalam tiga kelompok :

1. *Beban torque konstan* adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torque* nya tidak bervariasi. Contoh beban dengan *torque* konstan adalah *conveyors*, *rotary kilns*, dan pompa *displacement* konstan.
2. *Beban dengan variabel torque* adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa sentrifugal dan fan (*torque* bervariasi sebagai kwadrat kecepatan).
3. *Beban dengan energi konstan* adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan *torque* yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas. Sebuah motor DC yang memiliki tiga komponen utama:

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 dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar

melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.

## 2. Dinamo

Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo 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 dinamo.

## 3. *Commutator*

Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.

Motor DC memiliki beberapa jenis, antara lain :

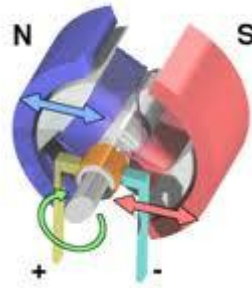
### 1. Motor DC Sumber Daya Terpisah/ *Separately Excited*

Jika arus medan dipasok dari sumber terpisah maka disebut motor DC sumber daya terpisah/*separately excited*.

### 2. Motor DC Sumber Daya Sendiri/ *Self Excited*: motor *shunt*

Pada motor *shunt*, gulungan medan (medan *shunt*) disambungkan secara paralel dengan gulungan dinamo. Oleh karena itu total arus dalam jalur merupakan penjumlahan arusmedan dan arus dinamo(Fathurohim,2010).





Gambar 2.4 Motor DC

## 2.5 WebCam

WebCam adalah kamera video sederhana berukuran relatif kecil. sering digunakan untuk konferensi video jarak jauh atau sebagai kamera pemantau. WebCam pada umumnya tidak membutuhkan kaset atau tempat penyimpanan data, data hasil perekaman yang didapat langsung ditransfer ke komputer.

Defenisi yang lain tentang WebCam adalah sebuah periferal berupa kamera sebagai pengambil citra/gambar dan mikropon (*optional*) sebagai pengambil suara/audio yang dikendalikan oleh sebuah komputer atau oleh jaringan komputer. Gambar yang diambil oleh *WebCam* ditampilkan ke layar monitor, karena dikendalikan oleh komputer maka ada *interface* atau *port* yang digunakan untuk menghubungkan *WebCam* dengan komputer atau jaringan. Ada beberapa orang mengartikan *WebCam* sebagai *Web pages + Cam era*, karena dengan menggunakan *WebCam* untuk mengambil gambar video secara aktual bisa langsung di *upload* bila komputer yang mengendalikan terkoneksi internet. Webcam (singkatan dari *web camera*), merupakan sebutan bagi kamera *real-time* (bermakna keadaan pada saat ini juga) yang gambarnya bisa diakses atau dilihat melalui internet, program *instant messaging* seperti Yahoo Messenger, AOL Instant Messenger (AIM), Windows Live Messenger, dan Skype. Istilah

"webcam" mengarah pada jenis kamera yang digunakan untuk kebutuhan layanan berbasis web(Majid,2007).



Gambar 2.5 WebCam

## 2.6 Pengolahan Citra

Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu.

Umumnya citra digital berbentuk persegi panjang atau bujur sangkar (pada beberapa sistem pencitraan ada pula yang berbentuk segienam) yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau *pixel* sehingga ukuran citra selalu bernilai bulat. Setiap titik memiliki koordinat sesuai posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem

yang digunakan. Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh titik tersebut.

Format data citra digital berhubungan erat dengan warna. Pada kebanyakan kasus, terutama untuk keperluan penampilan secara visual, nilai data digital merepresentasikan warna dari citra yang diolah. Format citra digital yang banyak dipakai adalah Citra Biner (monokrom), Citra Skala Keabuan ( *grayscale* ), Citra Warna ( *true color* ), dan Citra Warna Berindeks (Jihan,2010).

## 2.7 OpenCV

OpenCV adalah *library Open Source* untuk *Computer Vision* untuk C/C++, OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk image/video. OpenCV juga menyediakan *interface* ke *Integrated Performance Primitives* (IPP) Intel sehingga jika anda bisa mengoptimasi aplikasi *Vision* anda jika menggunakan prosesor Intel.

*Feature* yang dimiliki OpenCV antara lain :

1. *Manipulation* data citra (alokasi, copying, setting, konversi).
2. Citra dan video I/O (*file* dan kamera *based input*, *image/video file output*).
3. Manipulasi matriks dan vektor beserta rutin-rutin aljabar linear (*products, solvers, eigenvalues, SVD*).
4. Data struktur dinamis (*lists, queues, sets, trees, graphs*).
5. Pemroses citra fundamental (*filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids*).

6. Analisis struktur (*connected components, contour processing, distance transform, various moments, template matching, Hough transform, polygonal approximation, line fitting, ellipse fitting, Delaunay triangulation*).
7. Kalibrasi kamera (*calibration patterns, estimasi fundamental matrix, estimasi homography, stereo correspondence*).
8. Analisis gerakan (*optical flow, segmentation, tracking*).
9. Pengenalan obyek (*eigen-methods, HMM*).
10. Graphical User Interface (*display image/video, penanganan keyboard dan mouse handling, scroll-bars*).
11. Pelabelan citra (*line, conic, polygon, text drawing*).

OpenCV terdiri dari 3 library, yaitu

CV : untuk algoritma Image processing dan Vision

Highgui : untuk GUI, image dan video I/O

CXCORE : Untuk struktur data, support XML dan fungsi-fungsi grafis.

Selain itu OpenCV juga dilengkapi dengan Machine Learning library yang memiliki algoritma berikut :

1. *Naive bayes classifier*
2. *k-nearest neighbor algorithm*
3. *Support vector machine*
4. *Decision trees*
5. *Boosting*
6. *Random forest*
7. *Expectation maximization*
8. *Neural Networks*

*Library* ini bisa di-*compile* dengan Visual C++ 6.0, Visual C++.Net 2003/2005/2008 baik *Standard*, *Professional* maupun *express edition*(Adhiguna, 2008). Setelah mendapatkan fungsi-fungsi dari OpenCV maka selanjutnya menerapkan logika *RGB to Grayscale* pada OpenCV.

## 2.8 Konversi RGB ke Grayscale

Dalam fotografi dan komputasi, *grayscale* atau *greyscale* dari suatu gambar digital adalah gambar dimana nilai setiap *pixel* sampel tunggal, yang, itu hanya membawa informasi intensitas. Gambar semacam ini, juga dikenal sebagai hitam-putih, secara eksklusif terdiri dari warna abu-abu, bervariasi dari hitam di bagian intensitas paling lemah untuk putih di terkuat(Johnson ,2006).



Gambar 2.6 contoh gambar *greyscale* Padang Cement Indonesia.

Ada kasus bagaimana representasi nilai RGB (Red, Green, Blue) diubah menjadi gambar yang terdiri dari warna putih dan gradasi warna hitam yang biasa disebut *greyscale*. Untuk mengubah RGB menjadi *greyscale* dapat digunakan rumus sebagai berikut:

$$\text{greyscale} = 0.299R + 0.587G + 0.114B \quad (1)$$

atau

$$\text{greyscale} = 0.333R + 0.333G + 0.333B \quad (2)$$

Perhitungan nilai *greyscale* yang sebenarnya adalah dengan menggunakan persamaan pertama. Untuk lebih mudah digunakan untuk perhitungan manual dan mudah diingat digunakan persamaan yang kedua.

Misal ada warna RGB dengan pixel 3043918. Bagaimana mengkonversi menjadi warna *greyscale*? Gunakan prinsip geser dan logika AND. Konversi nilai decimal 3043918 ke bilangan biner menjadi 10111001110010011110. Kemudian lakukan proses AND 8 digit sehingga

$$\begin{array}{r} 10111001110010011110 \\ 0000000000000011111111 \\ \hline \text{AND} \\ 0000000000000010011110 = \text{Red.} \end{array}$$

Kemudian untuk *Green*. Geser 8 kali ke kanan kemudian lakukan proses AND.

$$\begin{array}{r} 10111001110010011110 \\ \text{Geser 8 Kali ke kanan} \\ 0000000010111001110010 \\ 0000000000000011111111 \\ \hline \text{AND} \\ 000000000000001110010 = \text{Green} \end{array}$$

Kemudian untuk *Blue*. Geser 16 kali ke kanan kemudian lakukan proses AND.

1011100111001001001110

Geser 16 kali ke kanan

00000000000000000101110

000000000000000011111111

————— AND

0000000000000000101110 = *Blue* (Samuel, 2012).

Setelah menjadi Greyscale maka selanjutnya diproses menjadi biner sehingga mendapatkan hasil yang mudah diolah.

## 2.9 Konversi dari *Grayscale* ke Biner (*Thresholding*)

*Thresholding* adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra secara jelas. Citra hasil *thresholding* biasanya digunakan lebih lanjut untuk proses pengenalan obyek serta ekstraksi fitur. Metode *thresholding* secara umum dibagi menjadi dua, yaitu :

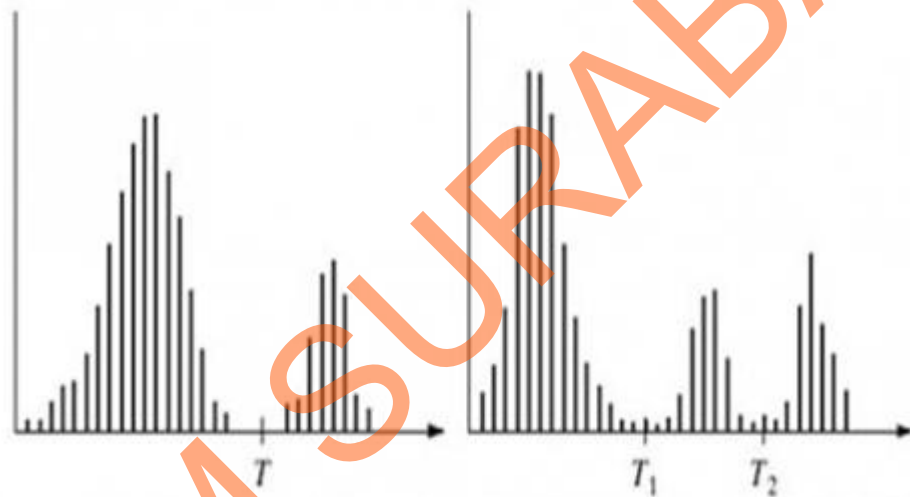
### 1. *Thresholding* global

*Thresholding* dilakukan dengan mempartisi histogram dengan menggunakan sebuah *threshold* (batas ambang) global T, yang berlaku untuk seluruh bagian pada citra.

### 2. *Thresholding* adaptif

*Thresholding* dilakukan dengan membagi citra menggunakan beberapa sub citra. Lalu pada setiap sub citra, segmentasi dilakukan dengan menggunakan *threshold* yang berbeda.

Yang menjadi fokus dalam tugas akhir ini adalah metode *thresholding* global. *Thresholding* diimplementasikan setelah dilakukan proses perbaikan kontras citra menggunakan fungsi *Contrast-limited adaptive histogram equalization* (CLAHE). *Thresholding* dikatakan global jika nilai *threshold*  $T$  hanya bergantung pada  $f(x,y)$ , yang melambangkan tingkat keabuan pada titik  $(x,y)$  dalam suatu citra. Berikut ini akan disajikan contoh partisi histogram untuk memperoleh *threshold* dalam Gambar 2.7.



Gambar 2.7 Contoh Partisi Histogram Untuk Memperoleh Nilai Threshold

Histogram yang berada pada sisi kiri Gambar 1 mewakili citra  $f(x,y)$  yang tersusun atas obyek terang di atas *background* gelap. *Pixel-pixel* obyek dan *background* dikelompokkan menjadi dua mode yang dominan. Cara untuk mengekstraks obyek dari *background* adalah dengan memilih nilai *threshold*  $T$  yang memisahkan dua mode tersebut. Kemudian untuk sembarang titik  $(x,y)$  yang memenuhi  $f(x,y) > T$  disebut titik obyek, selain itu disebut titik *background*. Kesuksesan metode ini bergantung pada seberapa bagus teknik partisi histogram.



Citra hasil thresholding dapat didefinisikan sebagaimana Persamaan di bawah ini (Nasir, 2012).

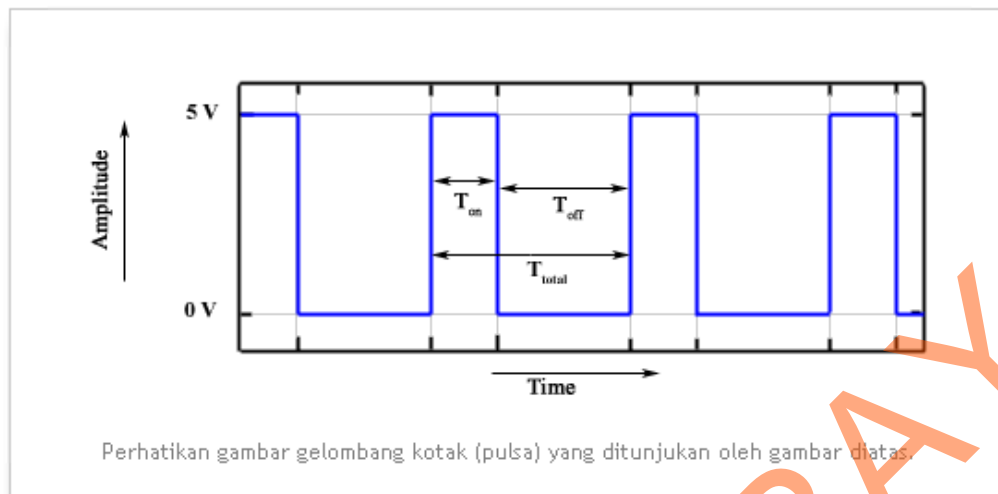
$$g(x,y) = \begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) \leq T \end{cases} \quad (1)$$

### 2.10 *Pulse Width Modulation (PWM)*

Pengaturan lebar pulsa modulasi atau PWM merupakan salah satu teknik yang “ampuh” yang digunakan dalam sistem kendali (*control system*) saat ini. Pengaturan lebar modulasi dipergunakan di berbagai bidang yang sangat luas, salah satu diantaranya adalah: *speed control* (kendali kecepatan), *power control* (kendali sistem tenaga), *measurement and communication* (pengukuran atau instrumentasi dan telekomunikasi).

Di dalam tutorial yang akan saya bahas kali ini akan mengajak anda mengenal dasar-dasar PWM dan implementasi PWM dengan menggunakan *microcontroller*.

Modulasi lebar pulas (PWM) dicapai/diperoleh dengan bantuan sebuah gelombang kotak yang mana siklus kerja (*duty cycle*) gelombang dapat diubah-ubah untuk mendapatkan sebuah tegangan keluaran yang bervariasi yang merupakan nilai rata-rata dari gelombang tersebut.



Gambar 2.8 Grafik PWM.

$T_{on}$  adalah waktu dimana tegangan keluaran berada pada posisi tinggi (baca: *high* atau *1*) dan  $T_{off}$  adalah waktu dimana tegangan keluaran berada pada posisi rendah (baca: *low* atau *0*).

Anggap  $T_{total}$  adalah waktu satu siklus atau penjumlahan antara  $T_{on}$  dengan  $T_{off}$ , biasa dikenal dengan istilah “periode satu gelombang”.

$$T_{total} = T_{on} + T_{off} \quad (1)$$

Siklus kerja atau *duty cycle* sebuah gelombang di definisikan sebagai,

$$D = \frac{T_{on}}{T_{on} + T_{off}} = \frac{T_{on}}{T_{total}} \quad (2)$$

Tegangan keluaran dapat bervariasi dengan *duty-cycle* dan dapat dirumuskan sebagai berikut,

$$V_{out} = D \times V_{in} \quad (3)$$

$$\text{sehingga: } V_{out} = \frac{T_{on}}{T_{total}} \times V_{in} \quad (4)$$

Dari rumus diatas dapat ditarik kesimpulan bahwa tegangan keluaran dapat diubah-ubah secara langsung dengan mengubah nilai  $T_{on}$ .

Apabila  $T_{on}$  adalah 0,  $V_{out}$  juga akan 0. Apabila  $T_{on}$  adalah  $T_{total}$  maka  $V_{out}$  adalah  $V_{in}$  atau katakanlah nilai maksimumnya.

Penjelasan tersebut diatas merupakan keseluruhan dari teori dasar dibalik sistem PWM. Sekarang, mari kita kita lihat pada implementasi praktek dari teori PWM tersebut diatas pada sebuah *microcontroller*(Ari, 2010).

STIKOM SURABAYA