

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Penelitian mengenai pengolahan citra menggunakan mikrokontroler diawali oleh *Carnegie Mellon University* yang membuat modul kamera *CMUCam* untuk pemakaian pada *mobile robot* (Kusuma, Suwastono, Dewanto 2012). Modul *CMUCam* ini didesain sebagai modul pengolah citra yang murah dan bisa dihubungkan dengan berbagai mikrokontroler lewat jalur *serial*. Sensor kamera yang digunakan pada *CMUCam* adalah OV6620. *CMUCam* menggunakan mikrokontroler 8 bit SX28 sebagai pengolah citranya. Karena keterbatasan RAM, maka total citra yang bisa diproses hanya 80x143 *pixel* (horisontal x vertikal). Akuisisi dan pemrosesan citra dilakukan secara per baris karena RAM yang ada tidak mencukupi untuk menampung semua *pixel* dari kamera. *CMUCam* seri ini memiliki fitur penjejak warna dengan metode pengambangan terhadap nilai *RGB pixel* yang dideteksi. *frame rate* maksimal yang dapat dicapai oleh *CMUCam* adalah 16.7 *frame* per detik [1].

*Pixy CMUCam 5* merupakan *image sensor* generasi ke-5 dari *CMUCam* dengan prosesor yang sangat *powerful*, *CMUCam 5* menggunakan prosesor NXP LPC4330, 204 MHz, dual core dengan Image sensor *Omnivision*, OV9715, 1/4", 1280x800 yang dapat diprogram untuk mengirimkan informasi berupa data gambar, sehingga proses pembacaan data olah gambar pada master mikrokontroler tidak terbebani. Sudut pandang jangkauan kamera *CMUCam 5* sebesar 75° untuk bidang horisontal dan 47° untuk bidang vertikal, Proses pengiriman data pada *Pixy CMUCam 5* dapat dilakukan dengan berbagai jalur komunikasi data, diantaranya *UART serial*, *SPI*, *I2C*, *digital out* maupun *analog out*.

Penelitian mengenai penjejukan objek berdasarkan warna dalam citra (Mada Sanjaya WS, Ph.D, 2015) melakukan penjejukan objek berdasarkan warna. Penelitian ini diterapkan pada komputer PC dengan bantuan *library* Matlab.

Untuk mendeteksi warna suatu objek, digunakan parameter *hue* dari suatu warna objek. Kemudian dilakukan konversi sistem warna dari nilai *RGB* ke *HSV* (*hue-saturation-value*). Warna suatu objek pada sistem *RGB* sangat dipengaruhi oleh intensitas pencahayaan sekitar. Sedangkan parameter *hue* pada sistem warna *HSV* tidak dipengaruhi intensitas pencahayaan. Kondisi pencahayaan hanya mempengaruhi parameter *saturation* dan *value* pada sistem *HSV* [2].

Penelitian mengenai pengolahan citra (Ramadhan Singgih Pradipta, UMS 2016) menggunakan *CMUCam 5* yaitu sistem *prototipe* troli bergerak secara otomatis mengikuti objek di depannya. Menggunakan sensor kamera *Pixy CMUCam 5* sebagai pendeteksi objek dengan warna orange. Warna pada objek disimulasikan dengan menggunakan papan persegi panjang dengan lingkaran warna berdiameter 10 cm. Pengolahan citra berupa pengenalan warna menjadi dasar pergerakan 4 motor *DC*. Robot bergerak mengikuti objek secara *real time* dengan teknik *color tracking*. *Board Arduino Uno* digunakan sebagai kendali seluruh sistem dan sensor ultrasonik *SRF05* berada di depan dan belakang sebagai penjaga jarak *prototipe* troli [3].

Pada penelitian (Faikul umam, Universitas Trunojoyo) dikembangkan sebuah sensor menggunakan 2 buah kamera yang dipasang sejajar dan dipisahkan oleh jarak untuk estimasi pengukuran jarak dan posisi orientasi objek. Untuk mengetahui keakuratan dari sensor ini, digunakan dua objek yang berbeda, yakni identifikasi objek diam menggunakan *Hough Transform* dan identifikasi objek bergerak menggunakan *Haar Cascade*. Struktur biologis manusia sudah banyak mengilhami para peneliti untuk terus melakukan riset, misalnya *Stereo Camera*. Terbentuknya algoritma *Stereo Camera* diilhami oleh penglihatan kedua mata pada manusia (*Binocular vision*), dimana data atau objek yang dilihat oleh kedua mata kita sebenarnya letaknya berbeda dan tumpang tindih. Hasil penglihatan mata kanan memiliki nilai posisi sendiri ( $x$ ,  $y$ ) begitu juga dengan mata kiri, ini disebabkan karena kedua mata memiliki perbedaan sudut pandang dalam melihat sebuah objek [4].

Pada penelitian sebelumnya telah digunakan beberapa metode untuk memperoleh informasi mengenai kedalaman menggunakan *stereo vision system*

(Suharlim, Putra, Budiman, Kartowosasto 2013). Perangkat keras yang digunakan terdiri dari satu buah kamera *stereo* dan satu buah *notebook*. Kamera yang digunakan adalah minoru 3D *Webcam* dengan *baseline* sebesar  $\pm 60\text{mm}$  dan dua buah kamera identik. Kamera ini digunakan untuk menghasilkan gambar input kiri dan kanan. Ukuran gambar yang diambil adalah  $640 \times 480$ . *Notebook* digunakan untuk memproses gambar input yang diperoleh dari minoru 3D *Webcam* untuk memperoleh informasi mengenai jumlah objek dan posisi objek yang terdeteksi terhadap kamera [5].

Masalah yang ditemui pada saat menggunakan *stereo vision* adalah estimasi *disparity map*. Nilai *disparity* ini sangat bergantung pada hasil pencarian pasangan antara *pixel* gambar kiri dan kanan. Proses pencarian ini adalah suatu proses yang sulit dan membutuhkan waktu yang tidak sedikit, tentunya proses ini juga dapat menghasilkan kesalahan. Para peneliti sebelumnya telah menerapkan beberapa cara untuk mengatasi masalah ini, seperti membatasi proses pencarian pasangan *pixel* yang merupakan tepi (*edge*) dan mengeliminasi *pixel* yang memiliki pasangan yang mirip dengan *pixel* acuan).

Penelitian Tugas Akhir mengenai pengolahan citra ini menggunakan modul *CMUCam 5* dengan pengontrol *Arduino Nano V3* dan 2 buah aktuator *motor servo* yang di desain menjadi lengan robot agar dapat bergerak *2-axis* untuk menjejak objek sebesar  $120^\circ$  horisontal (*tilting*) dan  $120^\circ$  vertikal (*panning*). Ketika objek berada pada posisi *center*, pembacaan perkiraan jarak antar robot dan objek akan dikerjakan dengan menampilkan nilai pada *LCD* ukuran  $16 \times 2$ . Robot bergerak mengikuti objek secara *real time* dengan teknik *color tracking*, ketika robot tidak mendeteksi objek warna yang ditentukan, ketika *frame* kamera robot menemukan objek warna yang ditentukan. Warna yang disimulasikan dengan berbagai macam bentuk objek yang telah ditentukan dan asumsi untuk pengaturan pembacaan pada modul *CMUCam 5* adalah sama, sehingga kemampuan dalam mendeteksi, menjejak, dan memprediksi jarak objek dapat diuji dan dapat diperoleh bagaimana hasil deteksi *CMUCam* dalam variasi perbedaan bentuk, perubahan warna, perubahan pencahayaan, dan perubahan jarak pada objek.

## **2.2 Dasar Teori**

### **2.2.1 Robot**

Menurut kamus bahasa, robot berasal dari kata *robota* yang artinya pekerja. Oleh karena itu robot dapat didefinisikan sebagai sebuah alat mekanik yang bekerja terus menerus untuk membantu pekerjaan manusia, saat ini dunia robotika terus berkembang. Bukan hanya robot *humanoid*, robot berkembang dalam berbagai kondisi dan kebutuhan manusia seperti *arm robots, automated guided vehicle, dish washer and cloth washer, wheelLED robot, unmanned vehicle, stand-alone CNC machine millers, humanoid robots, legged robots, caterpillar-tracked robots* [2]. (Mada Sanjaya WS, Ph.D, 2015)

### **2.2.2 Perkembangan Komputer dan Robot Vision**

Dalam teknologi canggih dewasa ini, teknologi komputer dan *robot vision* mengambil peranan dan perhatian yang cukup besar. Komputer dan *robot vision* telah diaplikasikan secara nyata dalam berbagai bidang seperti absensi sidik jari perkantoran, deteksi retina, deteksi wajah di bandara, pembaca barcode, pengolahan data media, sistem sortasi kualitas produk, pesawat pengintai, bahkan untuk proyek eksplorasi luar angkasa. *robot vision* adalah teknologi robot cerdas yang akan terus berkembang, dalam penelitian *robot vision*, *Webcam* merupakan perangkat yang berfungsi sebagai mata robot. *Webcam* dapat menggantikan peranan berbagai jenis sensor seperti sensor warna, sensor jarak, sensor kecepatan, sensor cahaya, bahkan sensor suhu objek [2]. (Mada Sanjaya WS, Ph.D, 2015)

### **2.2.3 Pengolahan Citra**

Pengolahan citra adalah setiap bentuk pengolahan sinyal dimana *input* adalah gambar, seperti foto atau video bingkai, sedangkan *output* dari pengolahan gambar dapat berupa gambar atau sejumlah karakteristik atau parameter yang berkaitan dengan gambar. Kebanyakan gambar-teknik pemrosesan melibatkan atau memperlakukan foto sebagai dimensi dua sinyal dan menerapkan standar-teknik pemrosesan sinyal untuk itu, biasanya hal tersebut mengacu pada pengolahan gambar *digital*, tetapi dapat juga digunakan untuk optik dan

pengolahan gambar *analog*. Akuisisi gambar atau yang menghasilkan gambar *input* di tempat pertama disebut sebagai pencitraan.

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.

Konsep dasarnya cukup sederhana. Sebuah gambar pada dasarnya adalah kumpulan titik-titik warna pada sebuah layar yang sering disebut dengan istilah *pixel*. Semakin banyak *pixel* yang dimiliki oleh sebuah gambar, maka kualitas gambar ini akan semakin baik. Walaupun mata manusia tidak terlalu mampu untuk membedakan gambar-gambar di atas 5 *megapixel* dengan ukuran yang sama, namun besarnya jumlah *pixel* pada gambar masih berguna keika ukuran gambar yang bersangkutan. Untuk gambar dengan jumlah *pixel* yang banyak, maka pembesaran ukuran gambar tidak akan terlalu banyak mengurangi kualitas gambar. Sementara pada gambar dengan *pixel* sedikit, pembesaran ukuran gambar akan menyebabkan gambar terlihat pecah dan kotak-kotak. Setiap *pixel* gambar ini pada hakikatnya adalah perwakilan dari sebuah vektor warna pada layar.

Umumnya pada layar berwarna yang ada sekarang, maka vektor warna itu terdiri dari 3 layer, yang masing-masing layer mewakili sub warna merah, hijau, dan biru (*red, green, blue*). sebuah elemen *vektor* warna akan bernilai antara nol sampai 255 sehingga nantinya warna hitam pekat akan mewakili vektor  $[0\ 0\ 0]$ , sedangkan warna putih bersih akan mewakili  $[255\ 255\ 255]$ . Warna-warna lainnya merupakan variasi ketiga elemen tersebut dengan nilai-nilai yang berbeda. Dalam pengolahan citra pada dasarnya dengan menampilkan nilai *pixel* pada gambar, ekstraksi nilai *pixel RGB*, konversi *RGB* ke Grayscale, konversi *RGB* ke *biner* [2]. (Mada Sanjaya WS, Ph.D, 2015)

### 2.2.4 Gerakan Kamera (Camera Movement)

Gerakan Kamera yaitu teknik pengambilan video dengan menggerakkan kamera dengan tujuan memberi kesan dan arti tersendiri. Ada beberapa gerakan kamera yang dapat dilakukan dalam pengambilan gambar. Ini merupakan teknik-teknik yang biasa digunakan dalam pengambilan gambar.

*Pan* atau *panning* merupakan teknik menggerakkan kamera secara mendatar (horizontal) dari kanan ke kiri atau sebaliknya.

- *Pan right* : Gerakan kamera dengan memutar ke kanan
- *Pan left* : Gerakan kamera dengan memutar ke kiri

*Tilt* atau *tilting* merupakan teknik menggerakkan kamera secara mendongak dari bawah ke atas (vertikal) atau sebaliknya.

- *Tilt up* : Gerakan kamera dengan mendongak ke atas
- *Tilt down* : Gerakan kamera dengan mendongak ke bawah

*Dolly* atau *Tracking* merupakan teknik menggerakkan kamera di atas tripod atau *dolly* mendekati atau menjauhi subyek.

- *Dolly in* : Gerakan kamera mendekati objek
- *Dolly out* : Gerakan kamera menjauhi objek

*Zoom* merupakan teknik gerakan lensa zoom yang mendekati atau menjauhi objek secara optik, Dalam *Zooming* ini yang bergerak bukannya kamera tetapi lensa kamera yang bergerak maju atau mundur.

- *Zoom in* : Lensa bergerak maju atau pandangan mendekati objek
- *Zoom out* : Lensa bergerak mundur atau pandangan menjauhi objek

*Crab* merupakan gerakan kamera secara lateral atau menyamping, berjalan sejajar dengan subyek yang sedang berjalan.

- *Crab left* : kamera bergerak secara lateral, berjalan sejajar ke kiri
- *Crab right* : kamera bergerak secara lateral, berjalan sejajar ke kanan

### 2.2.5 Pixy CMUCam 5

*Pixy CMUCam 5* merupakan *image sensor* generasi ke-5 dari *CMUCam* dengan prosesor yang sangat *powerful*, *CMUCam 5* menggunakan prosesor NXP LPC4330, 204 MHz, dual core dengan *Image sensor Omnivision, OV9715, 1/4"*, 1280x800 yang dapat diprogram untuk mengirimkan informasi berupa data gambar, sehingga proses pembacaan data olah gambar pada master mikrokontroler tidak terbebani. Sudut pandang jangkauan kamera *CMUCam 5* sebesar 75° untuk bidang horisontal dan 47° untuk bidang vertikal, Proses pengiriman data pada *Pixy CMUCam 5* dapat dilakukan dengan berbagai jalur komunikasi data, diantaranya *UART serial, SPI, I2C, digital out* maupun *analog out.*, Setiap *Pixy CMUCam 5* dilengkapi dengan 6 – 10 *pin* kabel IDC.

*Pixy CMUCam 5* juga menggunakan warna dan saturasi sebagai sasaran utama pada pendeteksi gambar. Sensor ini mampu mendeteksi tujuh warna yang berbeda dan menemukan beberapa benda pada saat yang sama dengan kecepatan 50 fps. *Pixy CMUCam 5* memiliki aplikasi *GUI open source* yang disebut *PixyMon*.



**Gambar 2.1** *Pixy CMUCam 5* [6]  
( Sumber : [www.adafruit.com](http://www.adafruit.com), 2017 )

### 2.2.5.1 Spesifikasi *PixyCMUCam 5*

Prosesor : NXP LPC4330, 204 MHz, dual core

sensor kamera : Omnivision, OV9715, 1/4", 640x400

Jangkauan pandang : 75<sup>0</sup> horisontal dan 47<sup>0</sup> vertikal

Tipe Lensa : standard M12

Konsumsi daya : 140mA

Daya Masuk : *USB input (5V)* atau *unregulate input (6V sampai 10V)*

RAM : 264 Kb

Flash : 1 Mb

Komunikasi Data : *UART serial, SPI, I2C, USB, digital, analog*

Dimensi : 2,1" x 2,0" x 1,4"

Berat : 27 gram

### 2.2.5.2 Mikrokontroler

*CMUCam 5* dapat dengan mudah terhubung ke banyak kontoler dengan sistemkomunikasi yang berbeda karena telah mendukung beberapa pilihan antarmuka seperti *serial UART, SPI, I2C, USB*, tetapi *Pixy* pertama-tama dikembangkan untuk dapat berkomunikasi dengan *Arduino*. Selama beberapa bulan terakhir *Pixy* telah dikembangkan dengan menambahkan dukungan untuk *Arduino Due, Raspberry Pi, dan BeagleBone Black*. *library* perangkat lunak telah disediakan untul semua *platform* sehingga proses pemrograman dapat berjalan dengan mudah dan cepat. Selain itu pada *Pixy* juga disediakan *API Python* jika menggunakan kontroler berbasis *Linux* (misalnya *Raspberry Pi, BeagleBone*).

### 2.2.5.3 Purple Dinosaur (Dan Objek Lainnya)

*Pixy* menggunakan algoritma berbasis penyaringan warna untuk mendeteksi benda-benda. Metode penyaringan berdasarkan warna yang populer karena cepat, efisien, dan relatif kuat. Penyaringan warna yang biasa dikenal menggunakan *RGB* (merah, hijau, dan biru) untuk mewakili warna. *Pixy* menghitung warna (*hue*) dan saturasi dari setiap *pixel RGB* dari sensor gambar dan menggunakan inti sebagai parameter penyaringan utama. Perubahan

pencahayaan dan paparan dapat memiliki *frustating effect* pada algoritma penyaringan warna.

#### 2.2.5.4 Tujuh Deteksi Warna

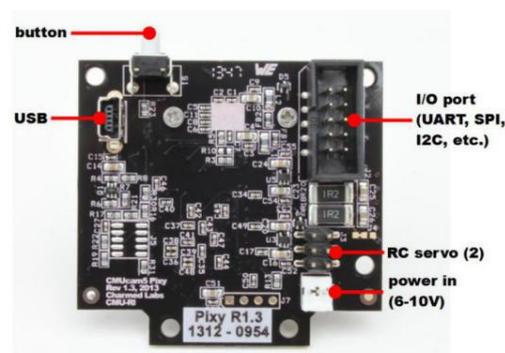
*Pixy* dapat mengingat ingat hingga 7 deteksi warna yang berbeda, yang berarti bahwa jika memiliki 7 objek yang berbeda dengan warna yang unik, algoritma warna penyaringan *Pixy* tidak akan memiliki masalah mengidentifikasi hal tersebut. Jika membutuhkan lebih dari tujuh, maka dapat menggunakan kode warna.

#### 2.2.5.5 Objek

*Pixy* dapat menemukan ratusan benda pada suatu waktu. Ini menggunakan algoritma komponen yang terhubung untuk menentukan di mana satu objek dimulai dan berakhir lain. *Pixy* kemudian mengkompilasi ukuran dan lokasi dari setiap objek dan laporan melalui salah satu *interface* (misalnya *SPI*).

#### 2.2.5.6 Deteksi Gambar Hingga 50 *frame* per detik

*Pixy* memproses seluruh bingkai 640x400 gambar setiap 1/50 detik (20 milidetik). Ini berarti bahwa kita akan mendapatkan pemberharuan posisi semua objek terdeteksi setiap 20 ms. Pada tingkat ini, pelacakan jalan jatuh / memantul bola adalah mungkin.



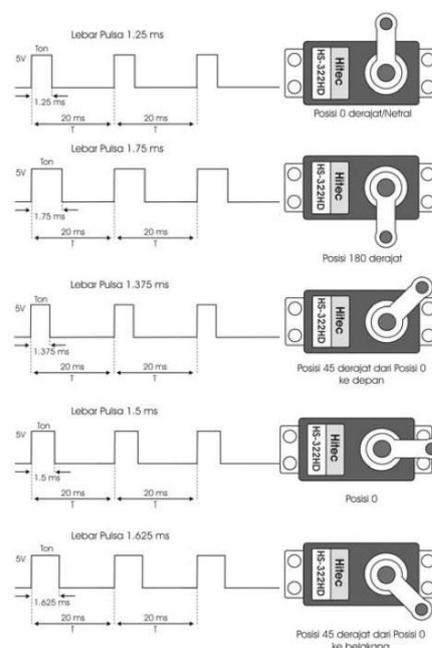
**Gambar 2.2** Bagian-Bagian *Pixy CMUCam 5* [6]

( Sumber : [http:// CMUCam.org/](http://CMUCam.org/), 2017)

### 2.2.6 Motor Servo

Motor *servo* adalah motor yang mampu bekerja dua arah (*CW* dan *CCW*) dimana arah dan sudut pergerakan rotornya dapat dikendalikan dengan memberikan variasi lebar pulsa (*duty cycle*) sinyal *PWM* pada bagian *pin* kontrolnya. Jenis Motor *servo* Standar 180° hanya mampu bergerak dua arah (*CW* dan *CCW*) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°. Motor *servo continuous* mampu bergerak dua arah (*CW* dan *CCW*) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu). Pulsa kontrol motor *servo* operasional dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari *range* sudut maksimum. Apabila motor *servo* diberikan pulsa dengan besar 1.5 ms mencapai gerakan 90°, maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati 0° dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati 180° [7].

(Nur Ahmad Effendi, 2014).

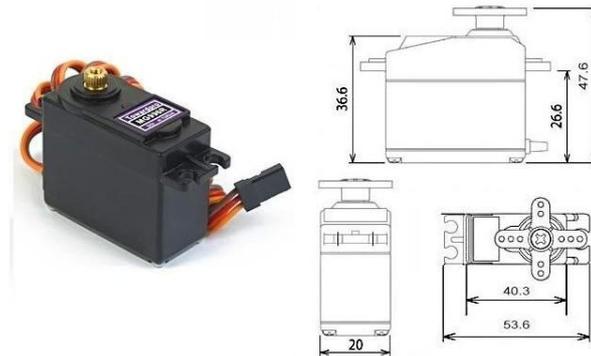


**Gambar 2.3** Pulsa Kendali Motor *servo* [7]

(Sumber : Nur Ahmad Effendi, Purwarupa Sistem Peringatan Dini Nirkabel Pada Jembatan Antar Pulau Oleh Angin Berbasis *Arduino Nano*, 2014)

### ***Servo TowerPro MG996R***

*TowerPro MG996R* merupakan *servo digital* dengan gear metal memiliki ukuran medium dan ringan dengan daya *output* tinggi . *servo* dapat memutar sekitar 120 derajat ( $60^\circ$  ke arah kanan dan  $60^\circ$  ke arah kiri ).

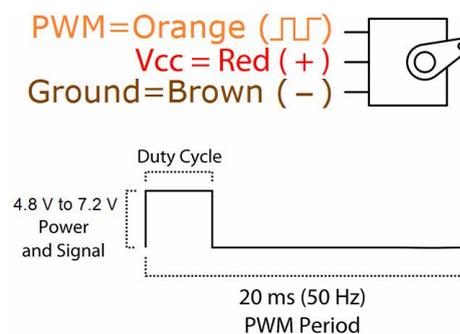


**Gambar 2.4** *MG996R Tower-Pro* [8]

( Sumber : [http:// electronicscaldas/datasheet/ MG996R\\_Tower-Pro.pdf](http://electronicscaldas/datasheet/MG996R_Tower-Pro.pdf), 2017 )

#### Spesifikasi motor

- a. Berat : 55 g
- b. Dimensi : 40.7 x 19.7 x 42.9 mm approx.
- c. Torsi Maksimal : 9.4 kgf·cm (4.8 V ), 11 kgf·cm (6 V)
- d. Kecepatan Operasi : 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
- e. Tegangan Kerja : 4.8 V a 7.2 V
- f. Kebutuhan Arus : 500 mA



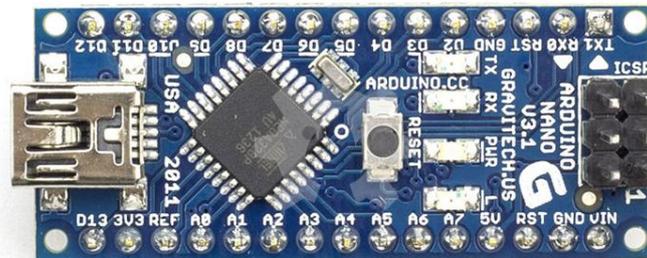
**Gambar 2.5** *Wiring MG996R Tower-Pro* [8].

( Sumber : [http:// electronicscaldas/datasheet/ MG996R\\_Tower-Pro.pdf](http://electronicscaldas/datasheet/MG996R_Tower-Pro.pdf), 2017 )

### 2.2.7 *Arduino*

*Arduino* merupakan pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. perangkat kerasnya memiliki prosesor *Atmel AVR* dan perangkat lunaknya memiliki bahasa pemrograman sendiri [9]. (wikipedia)

*Arduino Nano* adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan *breadboard*. *Arduino Nano V3* diciptakan dengan basis mikrokontroler *ATmega328* (untuk *Arduino Nano* versi 3.x) atau *ATmega 168* (untuk *Arduino* versi 2.x). *Arduino Nano* kurang lebih memiliki fungsi yang sama dengan *Arduino Duemilanove*, tetapi dalam paket yang berbeda. *Arduino Nano* tidak menyertakan colokan *DC* berjenis *Barrel Jack*, dan dihubungkan ke komputer menggunakan *port USB Mini-B*. *Arduino Nano* dirancang dan diproduksi oleh perusahaan *Gravitech*.



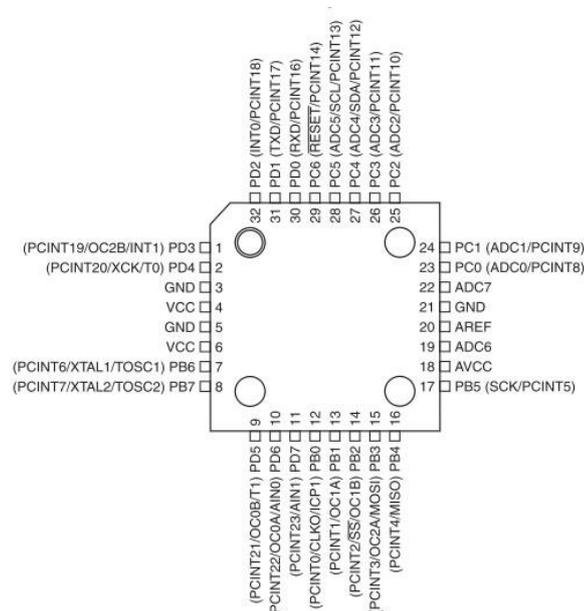
**Gambar 2.6** *Arduino Nano V3* [10].

( **Sumber** : [http:// ecadio/mengenal-dan-belajar- Arduino-Nano](http://ecadio/mengenal-dan-belajar-Arduino-Nano), 2017)

*Arduino Nano V3.0* ini memiliki 14 *pin digital* dan 8 *pin analog* yang kesemuanya dapat dijadikan *input* atau *output* yang disesuaikan dengan kebutuhan dari proyek yang dijalankan. *pin* ini akan mengeluarkan tegangan 5V untuk mode *HIGH* (logika 1) dan 0V untuk mode *LOW* (logika 0) jika dikonfigurasi sebagai *pin output*. Jika di konfigurasi sebagai *pin input*, maka ke 14 *pin* ini

juga dapat menerima tegangan 5V untuk mode *HIGH* (logika 1) dan 0V untuk mode *LOW* (logika 0). Beberapa dari *pin digital* memiliki fungsi khusus seperti *serial Tx/Rx* yang berfungsi untuk melakukan komunikasi data menggunakan *Transmitter/Receiver*, *External Interrupts* yang berfungsi untuk menjalankan perintah interupsi ketika program utama berjalan sesuai dengan algoritma yang diberikan, *PWM (Pulse Width Modulation)* yang berfungsi untuk memberikan kendali berupa pengaturan lebar pulsa yang dikeluarkan dari *pin* tersebut.

*pin SPI (serial Peripheral Interface)* suatu sistem komunikasi data yang memiliki prinsip kerja *Master/Slave*, artinya komunikasi yang dilakukan bersifat dua arah, sedangkan *pin analog* memiliki beberapa *pin* khusus yaitu *pin I<sup>2</sup>C (Inter Integrated Circuit)* yang didalamnya memiliki 2 *pin* yaitu *SCL (serial Clock Line)* dan *SDA (serial Data Line)* yang memiliki fungsi untuk melakukan komunikasi antara *Master* dengan beberapa *Slave* yang mengakses *Slave* sesuai dengan *Address* masing-masing. (Arduino, 2014).



**Gambar 2.7** Pemetaan *pin AtMega 328 SMD* [10].

( Sumber : [http:// ecadio/mengenal-dan-belajar- Arduino-Nano](http://ecadio/mengenal-dan-belajar-Arduino-Nano), 2017)

### 2.2.7.1 Spesifikasi

**Tabel 2.1** Tabel Spesifikasi *Arduino Nano V3* [10].

Mikrokontroler	Atmel <i>ATmega168</i> atau <i>ATmega328</i>
Tegangan Operasi	5V
<i>input Voltage</i> (disarankan)	7-12V
<i>input Voltage</i> (limit)	6-20V
<i>pin digital I/O</i>	14 (6 <i>pin</i> digunakan sebagai <i>output PWM</i> )
<i>pins input analog</i>	8
Arus DC per <i>pin I/O</i>	40 mA
<i>flash memory</i>	16KB ( <i>ATmega168</i> ) atau 32KB ( <i>ATmega328</i> ) 2KB digunakan oleh Bootloader
<i>SRAM</i>	1 KB ( <i>ATmega168</i> ) atau 2 KB ( <i>ATmega328</i> )
<i>EEPROM</i>	512 byte ( <i>ATmega168</i> ) atau 1KB ( <i>ATmega328</i> )
Clock Speed	16 MHz
Ukuran	1.85cm x 4.3cm

( Sumber : [http:// ecadio/mengenal-dan-belajar- Arduino-Nano, 2017](http://ecadio/mengenal-dan-belajar-Arduino-Nano,2017))

### 2.2.7.2 Memory

*ATmega168* memiliki 16 KB *flash memory* untuk menyimpan kode (2 KB digunakan untuk *bootloader*), Sedangkan *ATmega328* memiliki *flash memory* sebesar 32 KB, (juga dengan 2 KB digunakan untuk *bootloader*). *ATmega168* memiliki 1 KB *memory* pada *SRAM* dan 512 byte pada *EEPROM* (yang dapat dibaca dan ditulis dengan perpustakaan *EEPROM*); Sedangkan *ATmega328* memiliki 2 KB *memory* pada *SRAM* dan 1 KB pada *EEPROM*.

### 2.2.7.3 *input dan output*

Masing-masing dari 14 *pin digital* pada *Arduino Nano* dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Semua *pin* beroperasi pada tegangan 5 volt. Setiap *pin* dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up internal* sebesar 20-50 KOhm. Selain itu beberapa *pin* memiliki fungsi khusus, yaitu:

- a. **serial** : 0 (*RX*) dan 1 (*TX*). Digunakan untuk menerima (*RX*) dan mengirimkan (*TX*) *TTL data serial*. *pin* ini terhubung ke *pin* yang sesuai dari chip *FTDI USB-to-TTL serial*.
- b. **External Interrupt** (Interupsi Eksternal): *pin* 2 dan *pin* 3 ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
- c. **PWM** : *pin* 3, 5, 6, 9, 10, dan 11. Menyediakan *output PWM 8-bit* dengan fungsi *analogWrite()*. Jika pada jenis papan berukuran lebih besar (misal: *Arduino Uno*), *pin PWM* ini diberi simbol *tilde* atau “~” sedangkan pada *Arduino Nano* diberi tanda titik atau strip.
- d. **SPI** : *pin* 10 (*SS*), 11 (*MOSI*), 12 (*MISO*), 13 (*SCK*). *pin* ini mendukung komunikasi *SPI*. Sebenarnya komunikasi *SPI* ini tersedia pada perangkat keras, tapi untuk saat belum didukung dalam bahasa *Arduino*.
- e. **LED** : *pin* 13. Tersedia secara *built-in* pada papan *Arduino Nano*. *LED* terhubung ke *pin digital* 13. Ketika *pin* diset bernilai *HIGH*, maka *LED* menyala, dan ketika *pin* diset bernilai *LOW*, maka *LED* padam.

*Arduino Nano* memiliki 8 *pin* sebagai *input analog*, diberi label A0 sampai dengan A7, yang masing-masing menyediakan resolusi 10 *bit* (yaitu 1024 nilai yang berbeda). Secara *default pin* ini dapat diukur/diatur dari mulai *ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi *analogReference()*. *pin analog* 6 dan 7 tidak dapat digunakan sebagai *pin digital*. Selain itu juga, beberapa *pin* memiliki fungsi yang dikhususkan, yaitu:

- a. **I2C** : *pin* A4 (*SDA*) dan *pin* A5 (*SCL*). Yang mendukung komunikasi *I2C* (*TWI*) menggunakan perpustakaan *Wire*.

Masih ada beberapa *pin* lainnya pada *Arduino Nano*, yaitu:

- b. **AREF** : Referensi tegangan untuk *input analog*. Digunakan dengan fungsi *analogReference()*.
- c. **reset** : Jalur *LOW* ini digunakan untuk me-*reset* (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol *reset* pada *shield* yang menghalangi papan utama *Arduino*.

#### 2.2.7.4 Komunikasi

*Arduino Nano V3* memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan *Arduino* lain, atau dengan mikrokontroler lainnya. *ATmega168* dan *ATmega328* menyediakan komunikasi *serial UART TTL* (5 Volt), yang tersedia pada *pin digital* 0 (*RX*) dan *pin* 1 (*TX*). Sebuah *chip FTDI FT232RL* yang terdapat pada papan *Arduino Nano* digunakan sebagai media komunikasi *serial* melalui *USB* dan *driver FTDI* (tersedia pada perangkat lunak *Arduino IDE*) yang akan menyediakan *COM Port Virtual* (pada *device* komputer) untuk berkomunikasi dengan perangkat lunak pada komputer. Perangkat lunak *Arduino* termasuk didalamnya *serial Monitor* memungkinkan data tekstual sederhana dikirim ke dan dari papan *Arduino*. *LED RX* dan *TX* yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip FTDI* dan koneksi *USB* yang terhubung melalui *USB* komputer (tetapi tidak untuk komunikasi *serial* pada *pin* 0 dan 1).

Sebuah perpustakaan perangkat lunak *serial* memungkinkan komunikasi *serial* pada beberapa *pin digital Nano*. *ATmega168* dan *ATmega328* juga mendukung komunikasi *I2C (TWI)* dan *SPI*. Perangkat lunak *Arduino* termasuk perpustakaan *Wire* digunakan untuk menyederhanakan penggunaan *bus I2C*. Untuk komunikasi *SPI* menggunakan empat *pin* untuk berkomunikasi, berikut ini penjelasan bagaimana *pin* dari *SPI*. *SPI* biasanya menggunakan empat *pin* untuk berkomunikasi yaitu *MOSI*, *MISO*, *SCK* dan *SS*.

### 2.2.8 LCD 2x16

*LCD (Liquid Crystal Display)* berfungsi untuk menampilkan karakter angka, huruf ataupun simbol dengan lebih baik dan dengan konsumsi daya yang rendah, *LCD* dilengkapi dengan panel *LCD* dan tingkat kontras yang cukup tinggi serta pengendalian *LCD CMOS* yang telah terpasang dalam modul tersebut. Ada dua macam ukuran *LCD* yang dapat ditemui di pasaran yaitu *LCD 2x16* (2 baris, 16 kolom) dan *4x20* (4 baris, 20 kolom), yang dipakai dalam alat ini adalah jenis *LCD 2x16*.



**Gambar 2.8** *LCD 2x16* [11]

Dalam pasaran *LCD 2x16* terdapat tiga macam *back light* yaitu biru, putih dan hijau. *LCD 2x16* ini terdapat 16 kaki dimana masing-masing dari kaki tersebut mempunyai fungsi tersendiri. Berikut adalah tabel penjelasan dari masing-masing kaki *LCD 2x16* :

**Tabel 2.2** Fungsi *pin* Pada *LCD 2x16* [11]

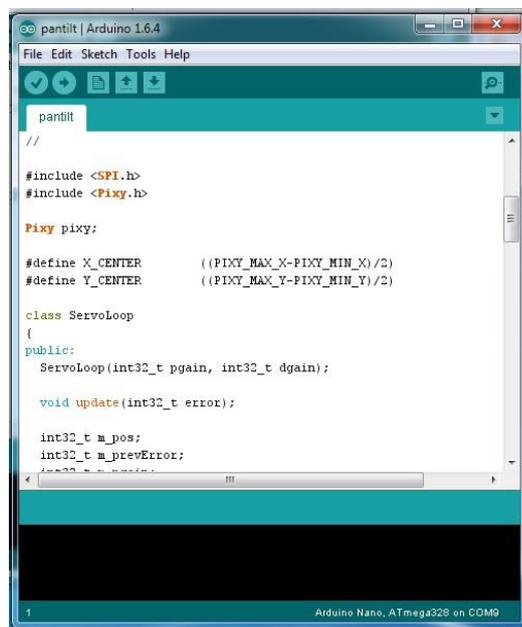
No <i>pin</i>	Name	Function
1	VSS	<i>ground</i> Voltage
2	VCC	5V
3	VEE	Contrast Voltage
4	RS	Register Select
5	R/W	Read/Write
6	E	Enable



lunak *PixyMon* terdapat *tab icon* yang berfungsi untuk mengatur pengolahan data gambar yang dibaca oleh *CMUCam 5* meliputi, kontras warna yang akan di eksekusi, pencahayaan *LED*, pelabelan objek, jalur komunikasi, dan gerak *motor*.

### 2.2.10 *Arduino IDE*

*IDE (Integrated Development Environment) Arduino* adalah perangkat lunak yang dijalankan dengan menggunakan *Java* dan terdiri dari beberapa fitur seperti *editor program*, *compiler*, dan *uploader*. *Editor program* adalah sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*. *Compiler* berfungsi mengubah kode program (bahasa *C Arduino*) menjadi Bahasa mesin dalam bentuk file *\*.hex (hexadecimal)*. *Uploader* adalah sebuah modul yang memuat kode *biner* dari komputer ke dalam memori didalam Board *Arduino* [12]. (Uswatun Hasanah, UMY 2016)



**Gambar 2.10** Tampilan *window Arduino IDE*

Program *Arduino* menggunakan bahasa pemrograman *C*. Ada tiga bagian utama dalam bahasa pemrograman *Arduino* yaitu struktur program, variabel dan fungsi. Struktur program meliputi kerangka program, sintaks program, kontrol

aliran program dan operator. Kerangka program *Arduino* terdiri dari dua blok yaitu:

1. Blok pertama adalah *void setup()* yang berisi kode program yang hanya sekali dijalankan saat *Arduino* dihidupkan dan merupakan inisialisasi program.
2. Blok kedua adalah *void loop()* yang berisi kode program yang akan dijalankan terus menerus dan merupakan program utama.

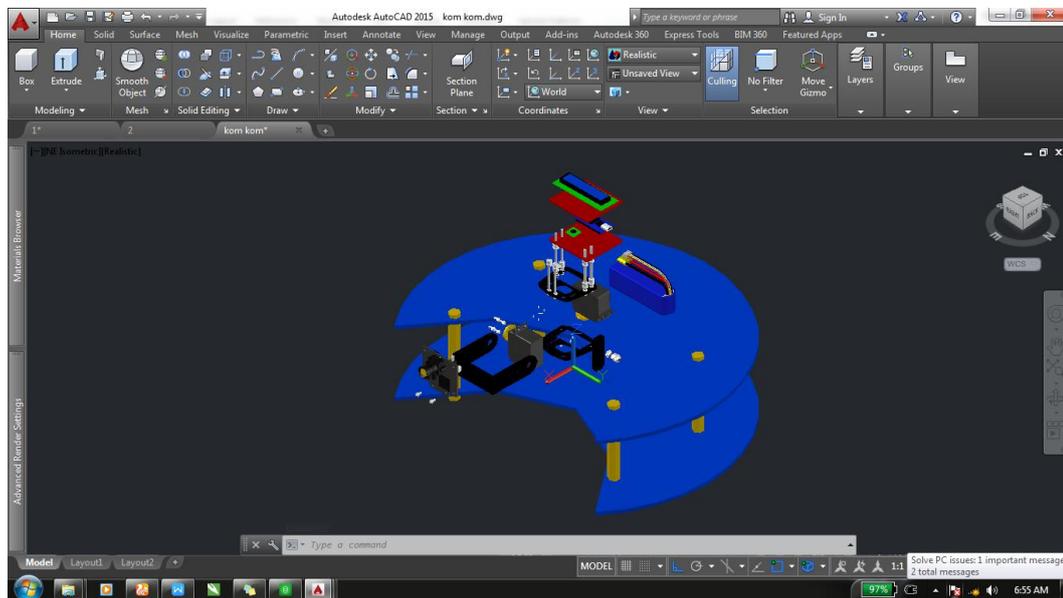
### 2.2.11 *AutoCAD*

*AutoCAD* adalah perangkat lunak komputer yang digunakan untuk menghasilkan gambar 2 dimensi dan 3 dimensi. Dengan perangkat lunak ini sangat memudahkan pekerjaan desain interior, insinyur mesin, arsitek dan berbagai profesi lainnya. Dengan berbagai fungsi *AutoCAD* yang mudah diaplikasikan, menjadikan perangkat lunak ini populer dan banyak digunakan.

Aplikasi *AutoCAD* sekarang ini hanya dapat dioperasikan pada komputer yang menggunakan sistem operasi *Microsoft*. Sebelumnya memang untuk sistem operasi *Machintos* dan *Linux* sempat bisa digunakan untuk menjalankan program *AutoCAD* [13]. Tapi fungsi perangkat lunak *AutoCAD* hanya ini berlangsung pada sekitar tahun 1980-an hingga 1990-an. Sayangnya fitur itu tidak lagi diteruskan *AutoDesk* sebagai pengembangnya. Berbagai versi dikembangkan *AutoDesk* sebagai perbaikan yang kreatif dan inovatif untuk memenuhi kebutuhan. Berbagai pengembangan ini semakin mempermudah penggunaanya dalam mengoperasikan segala macam fitur menarik.

*CAD* adalah kependekan dari *Computer-aided drafting and design program*. Sehingga aplikasi ini sangat bermanfaat dalam bidang desain grafis, arsitektur, teknik sipil, teknik mesin dan berbagai bidang lainnya yang berhubungan dengan penciptaan gambar tertentu. Dengan perangkat lunak ini, maka lebih mudah dalam menghasilkan model yang tepat, guna memenuhi kebutuhan khusus. Karena gambar bisa dibuat menyerupai bentuk aslinya, dengan ukuran yang disesuaikan. Sehingga dengan mengaplikasikan *AutoCAD*, maka bisa menghasilkan gambar dengan ketepatan tinggi, mudah dan hemat waktu. Karena

berbagai fitur dan *tool* yang dihadirkan sangat mudah digunakan. Sehingga tidak memerlukan waktu lama untuk bisa menguasainya.



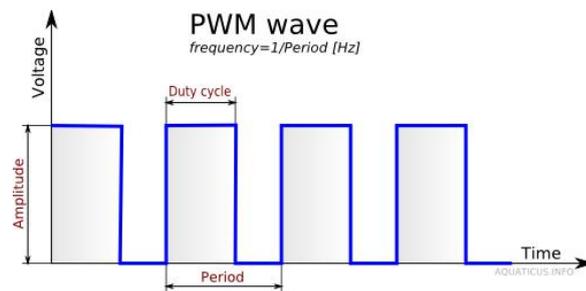
**Gambar 2.11** Tampilan *window AutoCAD*

( Sumber : <http://tips-tutorial/fungsi-AutoCAD-dan-tips-belajar-AutoCAD>, 2017 )

## 2.2.12 Sistem Komunikasi

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

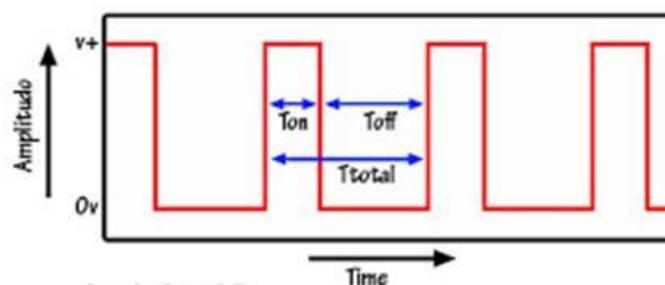
*PWM (Pulse Width Modulation)* adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu perioda, untuk mendapatkan tegangan rata-rata yang berbeda. Beberapa contoh aplikasi *PWM*, yaitu pemodulasi data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, *audio effect* dan penguat, serta aplikasi-aplikasi lainnya. Aplikasi *PWM* berbasis mikrokontroler biasanya untuk pengendalian kecepatan motor *DC*, pengendalian motor *servo* dan pengaturan nyala terang suatu *LED*.



**Gambar 2.12** Sinyal *PWM* [14]

(Sumber: <http://ow.ly/sH3U303dK9y>, 2016)

Sinyal *PWM* pada umumnya memiliki *amplitude* dan *frekuensi* dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar pulsa *PWM* berbanding lurus dengan *amplitude* sinyal asli yang belum termodulasi. Artinya, sinyal *PWM* memiliki *frekuensi* gelombang yang tetap namun *duty cycle* bervariasi (antara 0% hingga 100%).

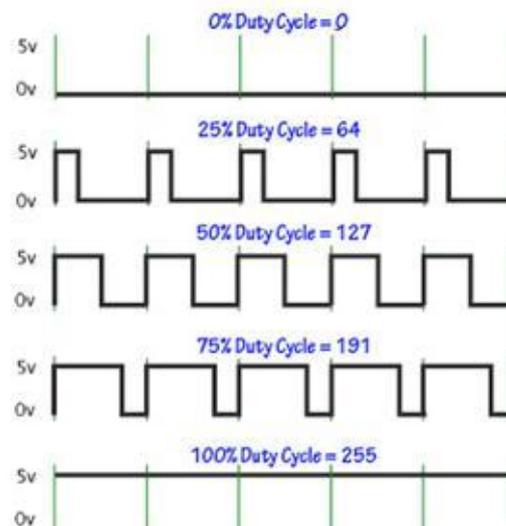


**Gambar 2.13** Sinyal *PWM* dan Persamaan  $V_{out\ PWM}$  [14]

(Sumber: <http://ini-robot.blogspot.com>, 2016)

*PWM* merupakan salah satu teknik untuk mendapatkan sinyal *analog* dari sebuah piranti *digital*. Sebenarnya sinyal *PWM* dapat dibangkitkan dengan banyak cara, dapat menggunakan metode *analog* dengan menggunakan rangkaian *op-amp* atau dengan menggunakan metode *digital*. Dengan metode *analog* setiap perubahan *PWM* yang terjadi sangat halus, sedangkan menggunakan metode *digital*, setiap perubahan *PWM* dipengaruhi oleh resolusi dari *PWM* itu sendiri. Resolusi adalah jumlah variasi perubahan nilai dalam *PWM* tersebut. Contoh, suatu *PWM* memiliki resolusi 8-bit yang berarti *PWM* ini memiliki variasi

perubahan nilai sebanyak = 256 variasi, mulai 0-255 perubahan nilai yang mewakili *duty cycle* 0-100% dari keluaran *PWM* tersebut.



**Gambar 2.14** *Duty Cycle* dan Resolusi *PWM* [14]  
(Sumber: <http://Arduino.cc/en/Tutorial/PWM>, 2016)

### 2.2.12.2 serial Peripheral Interface (SPI)

*serial Peripheral Interface* (SPI) adalah sebuah antarmuka bus yang biasa digunakan untuk mengirim data antara mikrokontroler dan perangkat kecil seperti register geser, sensor, dan kartu SD. *serial Peripheral Interface* (SPI) merupakan salah satu mode komunikasi *serial synchronous* kecepatan tinggi. Faktanya, komunikasi SPI lebih mudah jika dibandingkan dengan komunikasi USART ataupun UART karena sangat mudah dan sederhana.

SPI banyak diaplikasikan seperti:

1. Transmisi kabel data, (meski pilihan pertama adalah USART, tapi tidak Transmisi Wireless (nirkabel) seperti pada Zigby, 2,4 GHz transmission.

2. Pemrograman *AVR* (upload file `**hex` ke mikrokontroler) juga menggunakan *SPI* transfer.
3. Perangkat komunikasi data dari sensor maupun *peripheral device* yang lain, seperti: *IMU, Joystick, Compass*, dll.

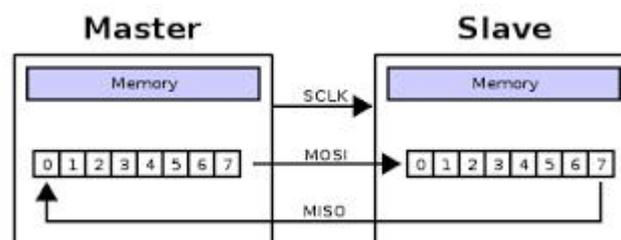
Berikut adalah Keuntungan menggunakan *bus SPI*:

1. Sangat mudah dalam *interfacenya* (*setup/inisialisasi* dan pengiriman data lebih mudah dibandingkan dengan *USART* atau *I2C*).
2. Komunikasi *Full duplex*.
3. Konsumsi daya yang rendah dibanding dengan *I2C* dan masih banyak lagi. Disamping itu, *SPI* juga memiliki kelemahan antara lain: Butuh sambungan atau koneksi yang lebih banyak, jadi lebih memakan *pin* dari mikrokontroler.

#### *Master Dan Slave Dalam SPI*

Dalam Komunikasi *SPI*, dua atau lebih *device* yang terhubung yang salah satunya akan bertindak sebagai *Master* dan yang lainnya akan bertindak sebagai *Slave*.

*Master device* adalah *device* yang memulai sambungan dan melakukan kontrol transmisi data. Saat kedua *device* sudah terkoneksi, *Master* dapat meminta *request* data baik mengirim data ataupun menerima data. Seperti pada penjelasan di atas, hal ini dinamakan koneksi *Full Duplex*, yaitu *Master device* dapat mengirim data dan juga *Slave device* juga dapat mengirim data pada saat yang bersamaan.



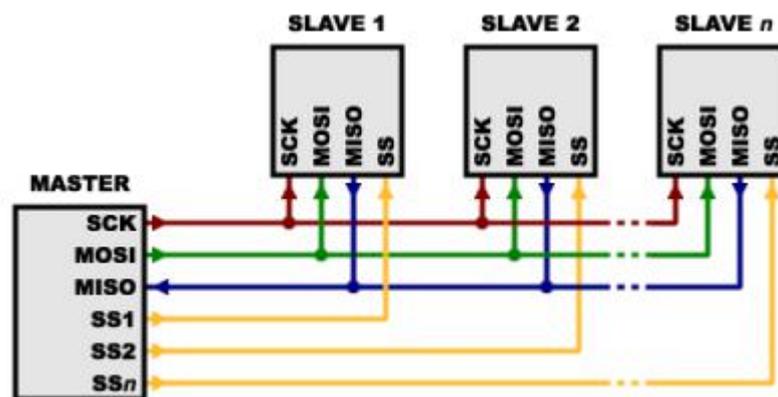
**Gambar 2.15** Koneksi *Full Duplex* [15]

(Sumber: <http://www.insinyoer.com/dasar-dasar-SPI/>, 2017)

Pada penjelasan sebelumnya *SPI* menggunakan empat *pin* untuk berkomunikasi, berikut ini penjelasan bagaimana *pin* dari *SPI*. *SPI* biasanya menggunakan empat *pin* untuk berkomunikasi, yaitu *MOSI*, *MISO*, *SCK* dan *SS*.

1. *MOSI* berarti *Master output Slave Input*, dimana *pin* ini digunakan sebagai *Master* mengirim data dan *Slave* sebagai penerima data.
2. *MISO* berarti *Master input Slave output*, dimana *pin* ini digunakan sebagai *Master* bertindak sebagai penerima data dan *Slave* sebagai pengirim data.
3. *SCK*, *SPI Clock Line* yang berarti *clock* yang digunakan untuk berkomunikasi (*Komunikasi synchronous*)
4. *SS* artinya *Slave Select*. Jika *SS* bernilai 0 atau *low* maka *device* bertindak sebagai *Slave* dan jika bernilai 1 atau *high* maka *device* bertindak sebagai *Master*.

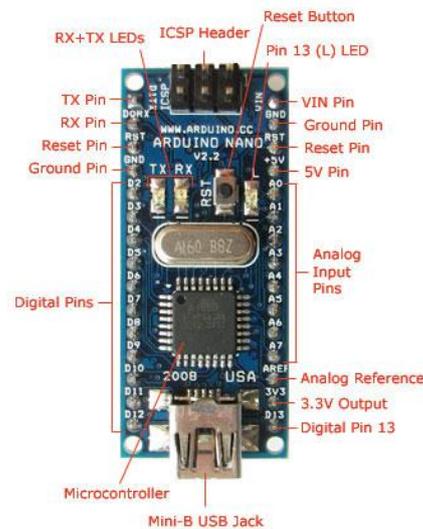
Ilustrasi cara kerja *protokol SPI*



Gambar 2.16 Komunikasi *SPI* [15]

(Sumber: <http://www.insinyoer.com/dasar-dasar-SPI/>, 2017)

Dari sisi perangkat lunak, *Arduino* sudah menyediakan *library internal* khusus untuk menangani komunikasi *serial sinkron* dengan *SPI*. Nama *library* dalam *Arduino IDE* yaitu *SPI.h*, *library* ini bisa langsung digunakan dalam *sketch Arduino* utama atau sebagai pendukung *library* yang lain



**Gambar 2.17** pin komunikasi ICSP *Arduino Nano V3*

(Sumber: <http://www.insinyoer.com/dasar-dasar-SPI/>, 2017)

### 2.2.13 Kesebangunan Segitiga

Kesebangunan segitiga ini diperlukan pada Trigonometri, khususnya membantu kita dalam mempelajari perbandingan trigonometri sudut ber-relasi. Menurut Wikipedia, dasar dari Trigonometri adalah Konsep kesebangunan segitiga siku-siku. Sisi-sisi yang bersesuaian pada dua bangun datar yang sebangun memiliki perbandingan yang sama. Pada geometri Euclid, jika masing-masing sudut pada dua segitiga memiliki besar yang sama, maka kedua segitiga itu pasti sebangun. Hal ini adalah dasar untuk perbandingan trigonometri sudut lancip. Konsep ini lalu dikembangkan lagi untuk sudut-sudut non lancip ( lebih dari 90 derajat dan kurang dari nol derajat ).

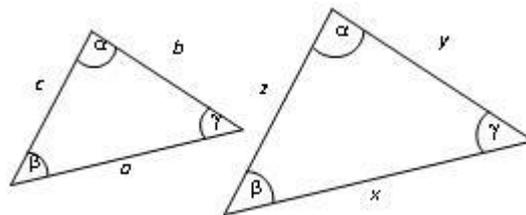
Kesebangunan merupakan salah satu konsep yang penting dalam matematika khususnya geometri. Kita dapat mengukur tinggi tiang bendera tanpa menurunkan tiang bendera. Kita dapat membuat model suatu benda dengan bentuk yang sangat mirip. Kita dapat mengetahui bentuk keseluruhan wilayah Indonesia melalui peta. Semua itu dapat dilakukan karena adanya konsep kesebangunan.

Dua buah atau lebih bangun datar dikatakan sebangun jika sisi-sisi yang bersesuaian memiliki perbandingan yang sama. Khususnya dalam segitiga,

definisi di atas akan dipenuhi jika segitiga memiliki sudut-sudut yang besarnya sama. Dengan kata lain, dua segitiga atau lebih dikatakan sebangun jika memiliki sudut-sudut yang besarnya sama. Konsep ini dapat digunakan untuk mencari panjang sisi-sisi pada segitiga jika diketahui panjang sisi-sisi yang lain.

Dua segitiga dikatakan sebangun jika memenuhi salah satu syarat berikut :

1. Perbandingan panjang sisi-sisi yang bersesuaian senilai.
2. Dua pasang sudut yang bersesuaian yang sama besar.



**Gambar 2.18** Segitiga sebangun

Kedua segitiga tersebut sebangun karena sudut-sudutnya sama besar. Oleh karena itu perbandingan sisi-sisi yang bersesuaian harus sama. Dengan kata lain :

$$\frac{a}{x} = \frac{b}{y} = \frac{c}{z}$$