

## BAB II TINJAUAN PUSTAKA

### 2.1 Definisi 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 (kecerdasan buatan). Istilah robot berawal bahasa Cheko “*robota*” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. 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 "cari dan tolong" (*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.

Saat ini hampir tidak ada orang yang tidak mengenal robot, namun pengertian robot tidaklah dipahami secara sama oleh setiap orang. Sebagian membayangkan robot adalah suatu mesin tiruan manusia (*humanoid*), meski demikian *humanoid* bukanlah satu-satunya jenis robot, jenis robot yang lain yaitu robot *mobile*, robot *manipulator* (tangan), robot berkaki, robot *flying*, robot *under water*. Pada kamus *Webster* pengertian robot adalah : *An automatic device that performs function ordinarily ascribed to human beings* (sebuah alat otomatis yang melakukan fungsi berdasarkan kebutuhan manusia). Dari kamus *Oxford* diperoleh pengertian robot adalah: *A machine capable of carrying out a complex series of actions automatically, especially one programmed by a computer*. (Sebuah mesin yang mampu melakukan serangkaian tugas rumit secara otomatis, terutama yang diprogram oleh komputer).

Pengertian dari *Webster* mengacu pada pemahaman banyak orang bahwa robot melakukan tugas manusia, sedangkan pengertian dari *Oxford* lebih umum, beberapa organisasi di bidang robot membuat definisi tersendiri. *Robot Institute*

*of America* memberikan definisi robot sebagai: *A reprogrammable multifunctional manipulator designed to move materials, parts, tools or other specialized devices through variable programmed motions for the performance of a variety of tasks* (Sebuah *manipulator* multifungsi yang mampu diprogram, didesain untuk memindahkan material, komponen, alat, atau benda khusus lainnya melalui serangkaian gerakan terprogram untuk melakukan berbagai tugas) *International Organization for Standardization (ISO 8373)* mendefinisikan robot sebagai: *An automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either Fixed in place or mobile for use in industrial automation applications* (Sebuah *manipulator* yang terkendali, multifungsi, dan mampu diprogram untuk bergerak dalam tiga aksis atau lebih, yang tetap berada di tempat atau bergerak untuk digunakan dalam aplikasi otomasi industri). Dari beberapa definisi di atas, kata kunci yang ada yang dapat menerangkan pengertian robot adalah:

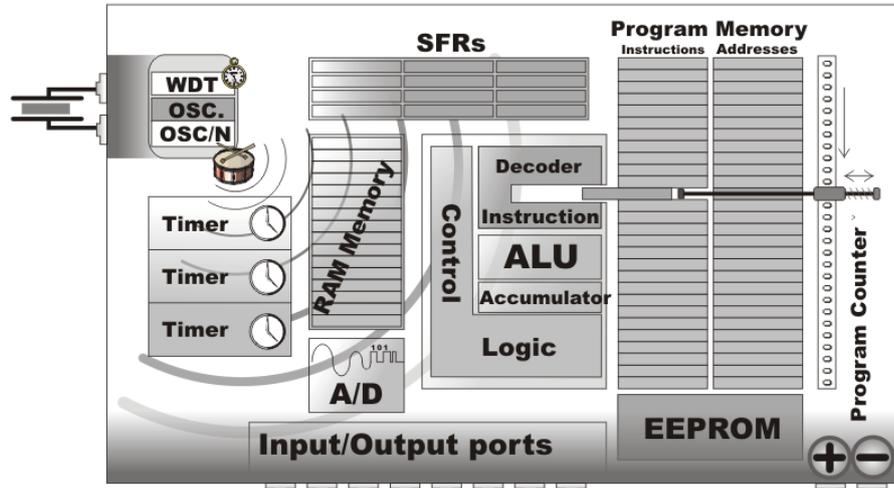
1. Dapat memperoleh informasi dari lingkungan (melalui sensor).
2. Dapat diprogram.
3. Dapat melaksanakan beberapa tugas yang berbeda.
4. Bekerja secara otomatis
5. Kecerdasan buatan (artificial intelligence)
6. Digunakan di industri (Membuat Robot itu Gampang, hal 5 2006)

## **2.2 Definisi Mikrokontroler**

Mikrokontroler merupakan suatu IC yang didalamnya berisi *CPU*, *ROM*, *RAM*, dan *I/O* yang dapat diprogram yang dapat disimpan didalam memory sehingga dapat mengendalikan perintah secara otomatis. Mikrokontroler terdiri dari beberapa bagian yaitu :

1. *CPU (Central Processing Unit)*
2. *RAM (Random Access Memory)*
3. *EEPROM/EPROM/PROM*
4. *I/O, Serial & Parallel*
5. *Timer*

## 6. Interrupt Controller



**Gambar 2.1:** Struktur Dari Mikrokontroler

(Sumber : Hendrawan (2007). *Sistem Mikrokontroler*. Makalah Teknik Elektro Politeknik Batam )

### 1. *Central Processing Unit (CPU)*

*CPU* merupakan bagian utama dalam suatu mikrokontroler. *CPU* pada mikrokontroler ada yang berukuran 8 bit ada pula yang berukuran 16 bit. *CPU* ini akan membaca program yang tersimpan di dalam *ROM* dan melaksanakannya.

### 2. *Read Only Memory (ROM)*

*ROM* merupakan suatu memori (alat untuk mengingat) yang sifatnya hanya dibaca saja. Dengan demikian *ROM* tidak dapat ditulisi. Dalam dunia mikrokontroler *ROM* digunakan untuk menyimpan program bagi mikrokontroler tersebut. Program tersimpan dalam format biner ('0' atau '1'). Susunan bilangan biner tersebut bila telah terbaca oleh mikrokontroler akan memiliki arti tersendiri.

### 3. *Random Acces Memory (RAM)*

Berbeda dengan *ROM*, *RAM* adalah jenis memori selain dapat dibaca juga dapat ditulis berulang kali. Tentunya dalam pemakaian mikrokontroler ada semacam data yang bisa berubah pada saat mikrokontroler tersebut bekerja. Perubahan data tersebut tentunya juga akan tersimpan ke dalam memori. Isi pada *RAM* akan hilang jika catu daya listrik hilang.

#### 4. *Electrically Erasable Programmable Read Only Memory (EEPROM)*

Beberapa mikrokontroler memiliki *EEPROM* yang terintegrasi pada chipnya. *EEPROM* ini digunakan untuk menyimpan sejumlah kecil parameter yang dapat berubah dari waktu ke waktu. Jenis memori ini bekerja relatif pelan, dan kemampuan untuk dihapus/tulisnya juga terbatas.

#### 5. *FLASH (EPROM)*

*FLASH* memberikan pemecahan yang lebih baik dari *EEPROM* ketika dibutuhkan sejumlah besar memori non-volatile untuk program. *FLASH* ini bekerja lebih cepat dan dapat dihapus/tulis lebih sering dibanding *EEPROM*.

#### 6. *Input/Output*

- *UART (Universal Asynchronous Receiver Transmitter)* adalah adapter serial port adapter untuk komunikasi serial asinkron.
- *USART (Universal Synchronous/Asynchronous Receiver Transmitter)* merupakan adapter serial port untuk komunikasi serial sinkron dan asinkron. Komunikasi serial sinkron tidak memerlukan *start* atau *stop* bit dan dapat beroperasi pada clock yang lebih tinggi dibanding asinkron.
- *SPI (serial peripheral interface)* merupakan port komunikasi serial sinkron.
- *SCI (serial communications interface)* merupakan enhanced *UART* (asynchronous serial port)
- *I2C bus (Inter-Integrated Circuit bus)* merupakan antarmuka serial 2 kawat yang dikembangkan oleh Philips. Dikembangkan untuk aplikasi 8 bit dan banyak digunakan pada konsumen elektronik, otomotif dan industri. *I2C* bus ini berfungsi sebagai antarmuka jaringan multi-master, multi-slave dengan deteksi tabrakan data. Jaringan dapat dipasangkan hingga 128 titik dalam jarak 10 meter. Setiap titik dalam jaringan dapat mengirim dan menerima data. Setiap titik dalam jaringan harus memiliki alamat yang unik.
- *ADC (Analog to Digital Conversion)* Fungsi *ADC* adalah merubah besaran analog (biasanya tegangan) ke bilangan digital. Mikrokontroler dengan fasilitas ini dapat digunakan untuk aplikasi-aplikasi yang memerlukan informasi analog (misalnya voltmeter, pengukur suhu dll). Terdapat beberapa

tipe dari ADC yaitu *Succesive Approximation A/D converters*, *Single Slope A/D converters*, *Delta-Sigma A/Ds converters*, *Flash A/D*.

- *DAC (Digital to Analog Converters)* Kebalikan dari *ADC* yaitu merubah besaran digital (biasanya tegangan) ke bilangan analog.

#### 7. *Interrupt Controller*

Interrupt merupakan metode yang efisien bagi mikrokontroler untuk memproses periperalnya, mikrokontroler hanya bekerja memproses periperall tersebut hanya pada saat terdapat data diperiperall tersebut. Pada saat terjadi interrupt, mikrokontroler menunda operasi yang sedang dilakukan kemudian mengidentifikasi interupsi yang datang dan menjalankan rutin pelayanan interupsi. Rata-rata mikrokontroler memiliki setidaknya-tidaknya sebuah interupsi eksternal, interupsi yang dimiliki bisa dipicu oleh "edge" atau "level". Edge triggered interrupt bekerja tidak tergantung pada waktu terjadinya interupsi, tetapi interupsi bisa terjadi karena glitch. Sedangkan Level triggered interrupt harus tetap pada logika high atau low sepanjang waktu tertentu agar dapat terjadi interupsi, interupsi ini tahan terhadap glitch.

#### 8. *Timer*

*Timer /Counter* adalah suatu peripheral yang tertanam didalam microcontroller yang berfungsi pewaktu. Dengan peripheral ini pengguna microcontroller dapat dengan mudah menentukan kapan suatu perintah dijalankan (delay), tentu saja fungsi timer tidak hanya untuk penundaan perintah saja, timer juga dapat berfungsi sebagai oscillator, PWM, ADC, dan lain-lain. (*Sistem Mikrokontroler*, hal 5, 2007)

### 2.2.1 **Definisi Arduino Mega**

Arduino Mega 2560 adalah papan pengembangan mikrokontroler yang berbasis Arduino dengan menggunakan chip ATmega2560. Board ini memiliki pin *I/O* yang cukup banyak, sejumlah 54 buah digital *I/O* pin (15 pin diantaranya adalah *PWM*), 16 pin analog input, 4 pin *UART (serial port hardware)*. Arduino Mega 2560 dilengkapi dengan sebuah osilator 16 Mhz, sebuah *port USB*, *power jack DC*, *ICSP header*, dan tombol reset. Board ini sudah sangat lengkap, sudah

memiliki segala sesuatu yang dibutuhkan untuk sebuah mikrokontroler. Dengan penggunaan yang cukup sederhana, anda tinggal menghubungkan power dari *USB* ke PC anda atau melalui adaptor *AC/DC* ke *jack DC*.

Pemrograman board Arduino Mega 2560 dilakukan dengan menggunakan *Arduino Software (IDE)*. Chip ATmega2560 yang terdapat pada Arduino Mega 2560 telah diisi program awal yang sering disebut *bootloader*. *Bootloader* tersebut yang bertugas untuk memudahkan anda melakukan pemrograman lebih sederhana menggunakan *Arduino Software*, tanpa harus menggunakan tambahan hardware lain. Cukup hubungkan Arduino dengan kabel *USB* ke PC atau *Mac/Linux* anda, jalankan *Arduino Software (IDE)*, dan anda sudah bisa mulai memprogram chip ATmega2560. Lebih mudah lagi, di dalam *Arduino Software* sudah diberikan banyak contoh program untuk belajar mikrokontroler.

Untuk pengguna mikrokontroler yang sudah lebih mahir, anda dapat tidak menggunakan *bootloader* dan melakukan pemrograman langsung via *header ICSP (In Circuit Serial Programming)* dengan menggunakan *Arduino ISP*.

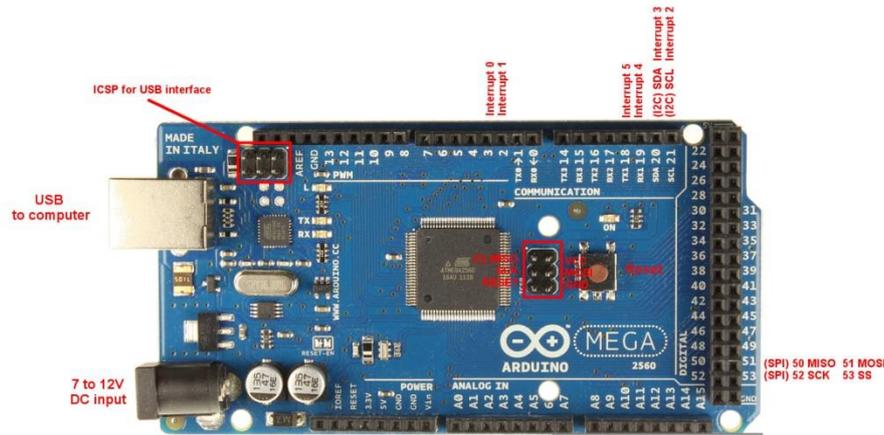
Arduino Mega 2560 Rev 3 telah dilengkapi dengan chip ATmega16U2 yang telah diprogram sebagai konverter *USB to Serial*. *Firmware* ATmega16U2 di *load* oleh *DFU bootloader*, dan untuk merubahnya anda dapat menggunakan *software Atmel Flip (Windows)* atau *DFU programmer (Mac OSX dan Linux)*, atau menggunakan *header ISP* dengan menggunakan *hardware external programmer*.

*Development board Arduino Mega 2560 R3* telah dilengkapi dengan *polyfuse* yang dapat direset untuk melindungi *port USB* komputer/laptop anda dari korsleting atau arus berlebih. Meskipun kebanyakan komputer telah memiliki perlindungan port tersebut didalamnya namun sikring pelindung pada Arduino Uno memberikan lapisan perlindungan tambahan yang membuat anda bisa dengan tenang menghubungkan Arduino ke komputer anda. Jika lebih dari 500mA ditarik pada *port USB* tersebut, sirkuit proteksi akan secara otomatis memutuskan hubungan, dan akan menyambung kembali ketika batasan aman telah kembali.

*Board Arduino Mega 2560* dapat ditenagai dengan power yang diperoleh dari koneksi kabel *USB*, atau via *power supply eksternal*. Pilihan power yang digunakan akan dilakukan secara otomatis

*External power supply* dapat diperoleh dari adaptor AC-DC atau bahkan baterai, melalui *jack DC* yang tersedia, atau menghubungkan langsung GND dan pin *Vin* yang ada di *board*. *Board* dapat beroperasi dengan *power dari external power supply* yang memiliki tegangan antara 6V hingga 20V. Namun ada beberapa hal yang harus anda perhatikan dalam rentang tegangan ini. Jika diberi tegangan kurang dari 7V, pin 5V tidak akan memberikan nilai murni 5V, yang mungkin akan membuat rangkaian bekerja dengan tidak sempurna. Jika diberi tegangan lebih dari 12V, *regulator* tegangan bisa over heat yang pada akhirnya bisa merusak papan. Dengan demikian, tegangan yang di rekomendasikan adalah 7V hingga 12V. Beberapa pin power pada Arduino :

1. GND. Ini adalah ground atau negatif.
2. Vin. Ini adalah pin yang digunakan jika anda ingin memberikan power langsung ke board Arduino dengan rentang tegangan yang disarankan 7V - 12V
3. Pin 5V. Ini adalah pin output dimana pada pin tersebut mengalir tegangan 5V yang telah melalui *regulator*
4. 3V3. Ini adalah pin output dimana pada pin tersebut disediakan tegangan 3.3V yang telah melalui *regulator*
5. IOREF. Ini adalah pin yang menyediakan referensi tegangan mikrokontroler. Biasanya digunakan pada *board shield* untuk memperoleh tegangan yang sesuai, apakah 5V atau 3.3V



**Gambar 2.2** Bentuk Fisik Arduinio Mega

(Sumber : <http://ecadio.com/belajar-dan-mengenal-arduino-mega> diakses pada 1 Juni 2016)

Chip ATmega2560 pada *Arduino Mega 2560 Revisi 3* memiliki memori 256 KB, dengan 8 KB dari memori tersebut telah digunakan untuk *bootloader*. Jumlah *SRAM* 8 KB, dan *EEPROM* 4 KB, yang dapat di baca-tulis dengan menggunakan *EEPROM library* saat melakukan pemrograman.

*Arduino Mega 2560* memiliki jumlah pin terbanyak dari semua papan pengembangan Arduino. Mega 2560 memiliki 54 buah *digital pin* yang dapat digunakan sebagai input atau output, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus sebesar 20mA, dan memiliki tahanan *pull-up* sekitar 20-50k ohm (secara *default* dalam posisi *disconnect*). Nilai maximum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan chip mikrokontroler

Beberapa pin memiliki fungsi khusus :

1. Serial, memiliki 4 serial yang masing-masing terdiri dari 2 pin. Serial 0 : pin 0 (RX) dan pin 1 (TX). Serial 1 : pin 19 (RX) dan pin 18 (TX). Serial 2 : pin 17 (RX) dan pin 16 (TX). Serial 3 : pin 15 (RX) dan pin 14 (TX). RX digunakan untuk menerima dan TX untuk transmit data *serial TTL*. Pin 0

dan pin 1 adalah pin yang digunakan oleh chip *USB-to-TTL ATmega16U2*

2. *External Interrupts*, yaitu pin 2 (untuk *interrupt 0*), pin 3 (*interrupt 1*), pin 18 (*interrupt 5*), pin 19 (*interrupt 4*), pin 20 (*interrupt 3*), dan pin 21 (*interrupt 2*). Dengan demikian *Arduino Mega 2560* memiliki jumlah *interrupt* yang cukup melimpah: 6 buah. Gunakan fungsi “*attachInterrupt()*” untuk mengatur *interrupt* tersebut.
3. *PWM*: Pin 2 hingga 13 dan 44 hingga 46, yang menyediakan *output PWM 8-bit* dengan menggunakan fungsi “*analogWrite()*”.
4. *SPI* : Pin 50 (*MISO*), 51 (*MOSI*), 52 (*SCK*), dan 53 (*SS*) mendukung komunikasi *SPI* dengan menggunakan *SPI Library*
5. *LED* : Pin 13. Pada pin 13 terhubung *built-in led* yang dikendalikan oleh digital pin no 13. *Set HIGH* untuk menyalakan led, *LOW* untuk memadamkannya.
6. *TWI* : Pin 20 (*SDA*) dan pin 21 (*SCL*) yang mendukung komunikasi *TWI* dengan menggunakan *Wire Library*

*Arduino Mega 2560 R3* memiliki 16 buah *input analog*. Masing-masing pin analog tersebut memiliki resolusi *10 bits* (jadi bisa memiliki 1024 nilai). Secara default, pin-pin tersebut diukur dari *ground* ke 5V, namun bisa juga menggunakan pin *AREF* dengan menggunakan fungsi “*analogReference()*”. Beberapa in lainnya pada *board* ini adalah :

1. *AREF*. Sebagai referensi tegangan untuk input analog.
2. *Reset*. Hubungkan ke *LOW* untuk melakukan reset terhadap mikrokontroler. Sama dengan penggunaan tombol reset yang tersedia.

*Arduino Mega R3* memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, berkomunikasi dengan *Arduino* lainnya, atau dengan mikrokontroler lain nya. *Chip Atmega2560* menyediakan komunikasi serial *UART TTL (5V)* yang tersedia di pin 0 (*RX*) dan pin 1 (*TX*). *Chip ATmega16U2* yang terdapat pada board berfungsi menterjemahkan bentuk komunikasi ini melalui

*USB* dan akan tampil sebagai *Virtual Port* di komputer. *Firmware 16U2* menggunakan *driver USB* standar sehingga tidak membutuhkan *driver* tambahan.

Pada *Arduino Software (IDE)* terdapat monitor serial yang memudahkan data textual untuk dikirim menuju Arduino atau keluar dari Arduino. *Led TX* dan *RX* akan menyala berkedip-kedip ketika ada data yang ditransmisikan melalui chip *USB to Serial* via kabel *USB* ke komputer. Untuk menggunakan komunikasi serial dari digital pin, gunakan *Software Serial library*

Chip *ATmega2560* juga mendukung komunikasi *I2C (TWI)* dan *SPI*. Di dalam *Arduino Software (IDE)* sudah termasuk *Wire Library* untuk memudahkan anda menggunakan bus *I2C*. Untuk menggunakan komunikasi *SPI*, gunakan *SPI library*. (ecadio.com.2016)

### **2.3 Definisi Sensor**

Sensor adalah komponen yang digunakan untuk mendeteksi suatu besaran fisik menjadi besaran listrik sehingga dapat dianalisa dengan rangkaian listrik tertentu. Sensor sangat berperan penting dalam dunia robotika yang berfungsi sebagai input navigasi suatu robot, adapun jenis jenis sensor yaitu: sensor suara, sensor cahaya, sensor tekanan sensor api, sensor suhu, sensor kelembapan, sensor ultrasonic, sensor magnet. Pada dasarnya sensor dengan penting di dunia robotik tergantung kebutuhan dari robot itu sendiri. (Wikipedia.org.2015)..

#### **2.3.1 Sensor Suara**

Sensor suara adalah sebuah alat yang mampu mengubah gelombang *Sinusioda*, suara menjadi gelombang sinus energi listrik (*Alternating Sinusioda Electric Current*). Sensor suara berkerja berdasarkan besar/kecilnya kekuatan gelombang suara yang mengenai membran sensor yang menyebabkan Bergeraknya membran sensor yang juga terdapat sebuah kumparan kecil di balik membran tadi naik & turun. Oleh karena kumparan tersebut sebenarnya adalah ibarat sebuah pisau berlubang-lubang, maka pada saat ia bergerak naik-turun, ia juga telah membuat gelombang magnet yang mengalir melewatinya terpotong-

potong. Kecepatan gerak kumparan menentukan kuat-lemahnya gelombang listrik yang dihasilkannya.

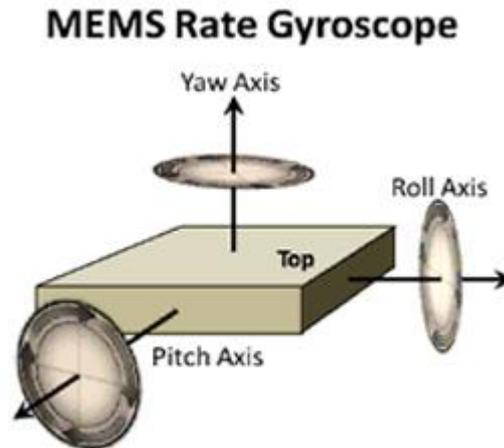
Komponen yang termasuk dalam Sensor suara yaitu *electric condenser microphone* atau mic kondenser. Intensitas suara adalah ukuran dari "aliran energi melewati satuan luas per satuan waktu" dan unit pengukuran adalah W/m<sup>2</sup> Probe intensitas suara mikrofon ini dirancang untuk menangkap intensitas suara bersama dengan unit arah aliran sebagai besaran vektor. Hal ini dicapai dengan menggabungkan lebih dari satu mikrofon di probe untuk mengukur aliran energi suara. mikrofon konvensional dapat mengukur tekanan suara (unit: Pa), yang mewakili intensitas bunyi di tempat tertentu (satu titik), tetapi dapat mengukur arah aliran. Mikrofon intensitas bunyi Oleh karena itu digunakan untuk sumber suara memeriksa dan untuk mengukur kekuatan suara. (<http://komponenelektronika.biz/sensor-suara.html>.2016)

### 2.3.2. Sensor Gyroscope dan Accelerometer

Gyroscope berfungsi untuk mengukur/menentukan orientasi suatu benda berdasarkan pada ketetapan momentum sudut. Dari pengertian lain gyroscope berfungsi untuk menentukan gerakan sesuai dengan gravitasi yang dilakukan oleh pengguna. Gyroscope ini memiliki peranan yang sangat penting dalam hal mempertahankan keseimbangan suatu benda seperti penggunaannya pada pesawat terbang yang dapat menentukan kemiringan pada sumbu x,y, dan z.

Output yang dihasilkan oleh gyroscope berupa kecepatan sudut yang pada sumbu x akan menjadi *phi* ( $\Phi$ ), sumbu y menjadi *theta* ( $\theta$ ), dan sumbu z menjadi *psi* ( $\Psi$ ). Sebelum digunakan biasanya gyroscope di kalibrasi terlebih dahulu dengan menggunakan bandul yang fungsinya untuk menentukan nilai faktor ataupun dapat juga melihat pada datasheet sensor yang digunakan.

*Gyroscope* adalah suatu alat berupa sensor gyro untuk menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi dengan cepat pada sumbu.



**Gambar 2.3** Prinsip Kerja Gyroscope

(Sumber : <http://id.wikipedia.org/wiki/Giroskop> diakses pada tanggal 10 Juni 2016)

*Gyroscope* memiliki output yang peka terhadap kecepatan sudut dari arah sumbu x yang nantinya akan menjadi sudut  $\phi$  (roll), dari sumbu y nantinya menjadi sudut  $\theta$  (pitch), dan sumbu z nantinya menjadi sudut  $\psi$  (yaw).

Penggunaan *gyroscope* dalam kehidupan sehari-hari belum banyak. *Gyroscope* lazimnya digunakan pada Pesawat terbang, Kapal, Helikopter, dll. Hal tersebut untuk mengurangi getaran yang ditimbulkan mesin agar keseimbangan tidak goyah. *Gyroscope* baru-baru ini juga diterapkan pada mainan Helikopter dengan tujuan menyeimbangkan saat diterbangkan.

### 2.3.2.1 MPU-6050 6-Axis Gyroscope+Accelerometer Module

Sensor ini adalah modul sensor gerak terpadu yang merupakan giroskop 3 sumbu sekaligus pengukur percepatan (akselerometer) 3 sumbu / **3-axis MEMS gyroscope + 3-axis MEMS accelerometer**. Sensor ini sangat akurat, dengan ADC (*analog-to-digital converter*) internal beresolusi 16-bit. Modul ini menggunakan IC InvenSense **MPU-6050** yang merupakan komponen elektronika pertama dan satu-satunya di dunia yang memadukan fungsi giroskop dan akselerometer dalam satu sirkuit terpadu (IC / *integrated circuit*).



**Gambar 2.4** Gambar Fisik MPU-6050 6-Axis Gyroscope+Accelerometer  
(Sumber : [http://www.vcc2gnd.com/2014/02/mpu-6050-6-axis-gyroscopeaccelerometer\\_73.html](http://www.vcc2gnd.com/2014/02/mpu-6050-6-axis-gyroscopeaccelerometer_73.html))

MPU-6050 menerapkan teknologi MotionFusion™ dan *run-time calibration firmware* yang menjamin kinerja optimal bagi pengguna. Dengan adanya Digital Motion Processor™ modul ini dapat diintegrasikan dengan magnetometer atau sensor lainnya lewat antarmuka I<sup>2</sup>C untuk memproses algoritma gerakan yang kompleks secara internal tanpa membebani kerja mikroprosesor / mikrokontroler utama.



**Gambar 2.5** Diagram Modul MPU 6050

(Sumber [http://www.vcc2gnd.com/2014/02/mpu-6050-6-axis-gyroscopeaccelerometer\\_73.html](http://www.vcc2gnd.com/2014/02/mpu-6050-6-axis-gyroscopeaccelerometer_73.html))

Produsen IC ini mendukung pengembangan perangkat keras dan perangkat lunak untuk menggunakan chip ini, antara lain dengan merilis API (*Application Programming Interface*) yang dapat diakses dari situs web resmi Invensense.

1. Catu daya IC dari 2,375 V ~ 3,46 V namun modul ini sudah dilengkapi dengan LDO / *low drop-out voltage regulator* (untuk pengguna Arduino,

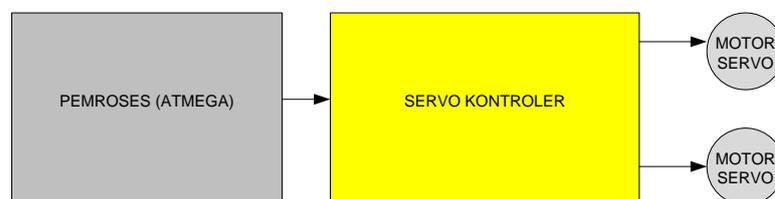
Anda dapat menyambungkan pin Vcc dari modul ini dengan pin 5V pada Arduino Anda)

2. Antarmuka kendali dan pengumpulan data lewat protokol I<sup>2</sup>C berkecepatan tinggi (*Fast Mode*, 400 kHz), pada modul ini sudah dipasangkan *pull-up resistor* 2K2 sehingga Anda bisa menyambungkan pin SDA dan SCL dari modul ini dengan mikrokontroler / Arduino Board tanpa resistor eksternal tambahan
3. Pilihan rentang skala giroskop: 250° (sensitivitas 13,1), 500° (65,6), 1000° (32,8), 2000° (16,4) per detik; sensitivitas dalam satuan LSB/°/detik
4. Pilihan rentang skala akselerometer: ±2g (sensitivitas 16384), ±4g (8192), ±8g (4096), ±16g (2048); sensitivitas dalam LSB/g
5. Data keluaran MotionFusion sebanyak 6 atau 9 sumbu dalam format matriks rotasi, quaternion, sudut Euler, atau data mentah (*raw data format*).
6. Memori penampung data (*buffer memory*) sebesar 1 Kb, FIFO (*First-In-First-Out*)
7. Dengan digabungkannya akselerometer dan giroskop dalam satu sirkuit terpadu menyebabkan pendeteksian gerakan menjadi lebih akurat (*reduced settling effects and sensor drift*) karena faktor kesalahan penyesuaian persilangan sumbu antara akselerometer dan giroskop dapat dihilangkan
8. DMP™ Engine mengambil alih komputasi rumit dari prosesor utama sehingga sistem tidak terbebani kalkulasi yang kompleks (*red*: sebelum adanya IC ini, perancang rangkaian elektronika biasanya menggunakan chip PLD eksternal untuk mengerjakan komputasi semacam ini karena perhitungan matematika dalam kalkulasi gerak sangatlah kompleks dan terlalu membebani kerja mikrokontroler yang biasanya bertenaga terbatas).
9. Tersedia platform pengembangan perangkat lunak MotionApps™ untuk sistem operasi Android, Linux, dan Windows
10. Algoritma untuk menghitung bias dan kalibrasi kompas sudah terpasang dan siap digunakan, tidak perlu intervensi dari pemakai

11. Interupsi yang dapat diprogram untuk mendeteksi pengenalan gestur (*gesture recognition*), pergeseran (*panning*), *zooming*, *scrolling*, dan *shake detection*
12. Konsumsi arus giroskop hanya sebesar 3,6 mA, giroskop + akselerometer hanya 3,8 mA (tenaga penuh, 1 kHz *sample rate*).
13. Moda siaga hemat daya hanya mengkonsumsi arus sebesar 5 $\mu$ A
14. Dapat menoleransi guncangan hingga 10000g
15. Modul dengan PCB berkualitas dengan *gold immersion welding* untuk menjamin kualitas
16. Akses sangat mudah menggunakan pin standar dengan *pitch* 0,1" / 2,54 mm. (www.vcc2gnd.com.2014)

#### 2.4 Definisi Servo Kontroler

Servo Kontroler adalah sebuah perangkat yang menerima sinyal perintah dari sistem kontrol pemrosesan atau mikrokontroler, dan mengirimkan arus listrik ke motor servo untuk menghasilkan gerak sebanding dengan sinyal diperintahkan. Biasanya sinyal perintah merupakan kecepatan yang diinginkan dan juga dapat mewakili torsi atau posisi yang diinginkan. Servo controller memiliki tujuan utama yaitu untuk mempermudah dalam mengendalikan servo yang jumlahnya lebih dari satu, tujuan lain dari servo kontroler yaitu dapat memperkecil penggunaan port pada chip pemrosesan atau mikrokontroler, karena servo kontroler terhubung dengan cara I2C pada chip pemrosesan (mikrokontroler).

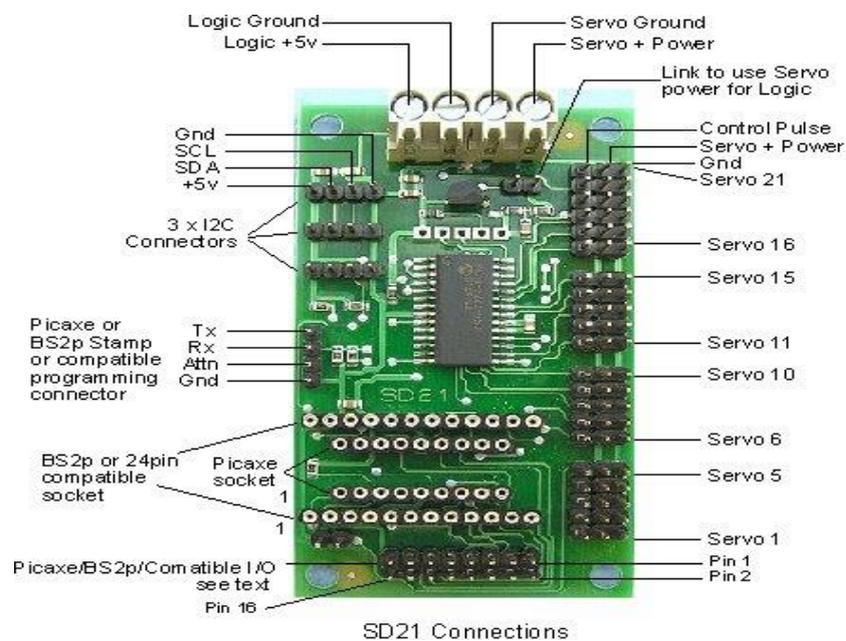


**Gambar 2.6 :** Struktur Diagram Servo Kontroler

(Sumber : Diolah Penulis)

### 2.4.1 Servo Kontroler SD21

Servo Kontroler SD21 adalah modul servo kontroler yang dapat mengontrol sebanyak 21 motor servo secara bersama - sama. Ini akan mendorong hingga 21 RC servo dan mempertahankan 20mS *refresh rate*, terlepas dari jumlah servo yang digunakan atau posisi mereka (pulsa lebar). Ini akan mengontrol kedua posisi dan kecepatan servo. Ini dikendalikan dengan mengirimkan perintah ke *PIC18F2220* melalui bus *I2C*. Ada 3 konektor *I2C* di papan, salah satu dari tiga tersebut dapat digunakan untuk terhubung ke Microcontroller. Atau, banyak pengendali seperti *Picaxe*, *BS2p*, *Atom*, *BX-24* dll. ([robot-electronics.co.uk/htm/sd21tech.htm](http://robot-electronics.co.uk/htm/sd21tech.htm))



**Gambar 2.7** Servo Kontroler SD 21

(Sumber : <http://www.robot-electronics.co.uk/htm/sd21tech.htm>. diakses pada tanggal 25 Mei 2016)

### 2.5. Definisi 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. Motor servo memiliki dua tipe yaitu servo standard dan servo *rotation*

(*continuous*). Dimana biasanya untuk tipe standar hanya dapat melakukan pergerakan sebesar  $180^\circ$  sedangkan untuk tipe *continuous* dapat melakukan rotasi atau  $360^\circ$ .

Pada dasarnya motor servo tersusun dari motor DC, rangkaian kontrol, *gearbox* dan potensiometer. Tampak seperti gambar motor servo beserta komponen internal motor servo dibawah ini.



**Gambar 2.8** : Bentuk Motor Servo (Kiri), Komponen Internal Servo (kanan)  
(Sumber : <http://Wikipediaindonesia.co.id/Motor-Servo/2012.html>. diakses pada tanggal 10 Juni 2016)

Terlihat jelas bahwa motor DC yang digunakan sangat kecil sehingga motor servo memiliki dimensi yang cukup kecil jika dibandingkan dengan motor DC pada umumnya. Rangkaian kontrol pada motor servo digunakan untuk mengontrol motor DC yang ada pada motor servo, dikarenakan untuk mengakses motor servo kita harus memberikan pulsa-pulsa kepada sinyal kontrol tersebut. Gearbox berfungsi untuk meningkatkan torsi dari motor servo, sebenarnya terdapat dua macam bahan penyusun *gearbox* yang digunakan untuk motor servo yaitu metal gear (biasanya untuk torsi yang sangat besar) dan nylon gear (berwarna putih seperti gambar diatas). Dan potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirimkan. (Wikipediaindonesia.co.id. 2012)

### 2.5.1. Motor Servo HS 7954 SH

Hitec HS-7954 merupakan jenis motor servo DC standar yang dapat beroperasi 180° dari sudut -90° sampai dengan 90° derajat, adapun spesifikasi motor servo yaitu:

1. Jenis servo *standard*
2. Torsi :
  - 6V dengan torsi 24 Kg
  - 7,4 deengan torsi 29Kg
3. Kecepatan
  - 6V = 0,15 sec/60°
  - 6.0V = 0,12 sec/60°
4. Berat = 55,0 g
5. Dimensi
  - Lebar = 19,8 mm
  - Tinggi = 42,9 mm
  - Panjang =40,6 mm
6. Pulsa kontrol :
  - 90° dengan pulsa 800u sec
  - 0° dengan pulsa 1500u sec
  - +90° dengan pulsa 2200u sec



**Gambar 2.9.** Bentuk Fisik Servo Hitec HS-7954 SH  
(Sumber : [http:// Datasheet Servo Hitec HS-7954 SH.2007](http://Datasheet%20Servo%20Hitec%20HS-7954%20SH.2007))

### 2.5.2. Motor Servo Turnigy TrackStar 910

Motor Servo Turnigy TrackStar 910 adalah motor servo DC Standard yang dapat beroperasi dengan sudut  $180^{\circ}$ , dari sudut  $-90^{\circ}$  sampai dengan sudut  $90^{\circ}$  adapun spesifikasinya antara lain :

1. Jenis Servo Standard

2. Torsi

25.8kg.cm (6.0V) –

30.6kg.cm (7.4V)

3. Kecepatan

0.16sec/60deg (6.0V)

0.14 sec/60deg (7.4V)

4. Berat

66 gram

5. Ukuran

Lebar : 39 mm

Tinggi : 19 mm

Panjang : 38,5 mm

6. Pulsa kontrol :

$-90^{\circ}$  dengan pulsa 800u sec

$0^{\circ}$  dengan pulsa 1500u sec

$+90^{\circ}$  dengan pulsa 2200u sec

([thehobbywarehouse.co.nz/collections/servo.2012](http://thehobbywarehouse.co.nz/collections/servo.2012))



**Gambar 2.10.** Servo TrackStar Turnigy

(Sumber : <https://www.thehobbywarehouse.co.nz/collections/servo>. diakses pada tanggal 25 Mei 2016)

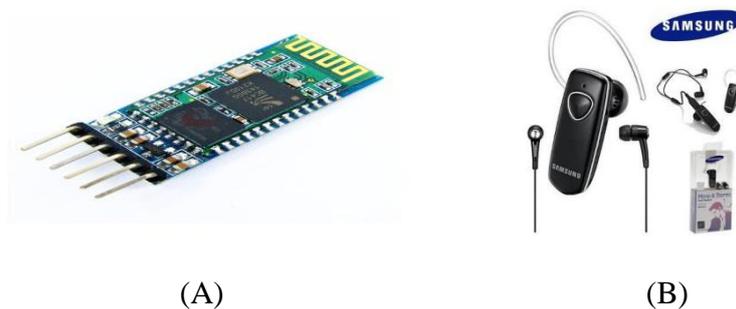
## 2.6 Definisi Bluetooth

Bluetooth adalah spesifikasi industri untuk jaringan kawasan pribadi (*personal area networks* atau *PAN*) tanpa kabel. Bluetooth menghubungkan dan dapat dipakai untuk melakukan tukar-menukar informasi di antara peralatan-peralatan. Spesifikasi dari peralatan Bluetooth ini dikembangkan dan didistribusikan oleh kelompok Bluetooth Special Interest Group. Bluetooth beroperasi dalam pita frekuensi 2,4 GHz dengan menggunakan sebuah *frequency hopping traceiver* yang mampu menyediakan layanan komunikasi data dan suara secara real time antara host-host bluetooth dengan jarak terbatas. Kelemahan teknologi ini adalah jangkauannya yang pendek dan kemampuan transfer data yang rendah. (Wikipedia.org/wiki/Bluetooth.2015)

### 2.6.1 Bluetooth HC 05 dan Headset Bluetooth

*Bluetooth Module HC-05* dan Headset Bluetooth merupakan module komunikasi nirkabel pada frekuensi 2.4GHz dengan pilihan koneksi bisa sebagai *slave*, ataupun sebagai *master*. Sangat mudah digunakan dengan mikrokontroler untuk membuat aplikasi *wireless. Interface* yang digunakan adalah serial *RXD*, *TXD*, *VCC* dan *GND*.

Tegangan input antara 3.6 ~ 6V, jangan menghubungkan dengan sumber daya lebih dari 7V. Arus saat *unpaired* sekitar 30mA, dan saat *paired* (terhubung) sebesar 10mA. 4 pin *interface* 3.3V dapat langsung dihubungkan ke berbagai macam mikrokontroler (khusus Arduino, 8051, 8535, AVR, PIC, ARM, MSP430, etc.). Jarak efektif jangkauan sebesar 10 meter, meskipun dapat mencapai lebih dari 10 meter, namun kualitas koneksi makin berkurang. (Martyncurrency.com.2015)



**Gambar 2.11.** Modul Bluetooth HC 05 (A), Headset Bluetooth (B)  
 (Sumber : [http://www.martyncurrey.com/hc-05-fc-114-and-headset Bluetooth-fc-114-first-look/](http://www.martyncurrey.com/hc-05-fc-114-and-headset-Bluetooth-fc-114-first-look/). diakses pada tanggal 25 Mei 2016)

## 2.7 Definisi Bahasa C

Akar dari bahasa C adalah bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Bahasa ini memberikan ide pada Ken Thompson yang kemudian mengembangkan bahasa yang disebut dengan B pada tahun 1970. Perkembangan selanjutnya dari bahasa B adalah bahasa C yang ditulis oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc. (sekarang adalah AT&T Bell Laboratories). Bahasa C pertama kali digunakan pada komputer Digital Equipment Corporation PDP-11 yang menggunakan sistem operasi *UNIX*. Standar bahasa C yang asli adalah standar dari *UNIX*. Sistem operasi, kompiler C dan seluruh program aplikasi *UNIX* yang esensial ditulis dalam bahasa C. Kepopuleran bahasa C membuat versi-versi dari bahasa ini banyak dibuat untuk komputer mikro. Untuk membuat versi-versi tersebut menjadi standar, ANSI (American National Standard Institutes) membentuk suatu komite (*ANSI Committee X3J11*) pada tahun 1983 yang kemudian menetapkan standar *ANSI* untuk bahasa C. Standar *ANSI* ini didasarkan kepada standar *UNIX* yang diperluas. Bahasa C mempunyai kemampuan lebih dibanding dengan bahasa pemrograman yang lain.

Bahasa C merupakan bahasa pemrograman yang bersifat portabel, yaitu suatu program yang dibuat dengan bahasa C pada suatu komputer akan dapat dijalankan pada komputer lain dengan sedikit (atau tanpa) ada perubahan yang berarti. Bahasa C merupakan bahasa yang biasa digunakan untuk keperluan

pemrograman sistem, antara lain untuk membuat Assembler, Interpreter Compiler, Sistem Operasi, Program bantu (*utility*), *Editor*, dan Paket program aplikasi. Beberapa program paket yang beredar seperti *dBase* dibuat dengan menggunakan bahasa C, bahkan sistem operasi *UNIX* juga dibuat dengan menggunakan bahasa C. Bahasa C sesungguhnya merupakan bahasa pemrograman yang serbaguna dan pemakaiannya tidak terbatas untuk pemrograman sistem, namun juga dapat digunakan untuk aplikasi bisnis, matematis maupun games, bahkan untuk aplikasi kecerdasan buatan.

Dalam beberapa literatur, bahasa C digolongkan sebagai bahasa tingkat menengah (*medium level language*). Penggolongan ini bukan berarti bahasa C kurang ampuh atau lebih sulit dibandingkan dengan bahasa tingkat tinggi (*high level language* - seperti *Pascal*, *Basic*, *Fortran*, *Java*, dan lain-lain), namun untuk menegaskan bahwa bahasa C bukanlah bahasa yang berorientasi pada mesin yang merupakan ciri dari bahasa tingkat rendah (*low level language*), yaitu bahasa mesin dan *assembly*.

Pada kenyataannya, bahasa C mengkombinasikan elemen dalam bahasa tingkat tinggi dan bahasa tingkat rendah, kemudahan dalam membuat program yang ditawarkan pada bahasa tingkat tinggi dan kecepatan eksekusi dari bahasa tingkat rendah. (aviscena-ary.blogspot.com.2016)

## **2.8 Definisi Software Arduino IDE**

Arduino bukan hanya sekedar sebuah alat pengembangan, tetapi juga merupakan kombinasi dari hardware, bahasa pemrograman, dan *Integrated Development Environment (IDE)* yang canggih. *IDE* adalah sebuah *software* untuk menulis program, meng-*compile* menjadi kode biner, dan meng-*upload* ke dalam memori mikrokontroler. Ada banyak projek dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena ia menjadi pilihan dan acuan bagi

banyak praktisi. *IDE* Arduino yang bisa di-download dan diinstal pada komputer secara gratis. (Rustamaji.net.2016)



**Gambar 2.12** Tampilan Menunggu Jendela Arduino IDE

(Sumber : Diolah Sendiri)

### 2.8.1 Konstruksi Bahasa C pada Arduino IDE

Struktur program sketch dapat dibagi menjadi 3 bagian yaitu.

#### 1. Bagian Deklarasi Awal

Bagian deklarasi awal digunakan untuk mendeskripsikan variabel-variabel yang akan digunakan dalam program utama dan juga untuk menambahkan file-file program yang dibutuhkan untuk menjalankan program utama. Berikut contoh deklarasi program

```
#include<AVR/io.h>
int tombol1 = 3;
char lampu1 = 0;
```

#### 2. Bagian *Setup*

Bagian *setup* digunakan untuk menginisiasi variabel, mengatur *mode* pin pada *board*, mengatur *timer*, mengatur *baudrate serial port* dan sebagainya. Bagian *setup* hanya dijalankan sekali saja yaitu pada saat awal program dijalankan atau ketika program direset. Berikut contoh program *setup*.

```
void setup()
```

```

{ pinMode(tomboll, INPUT);
pinMode(Lampul, OUTPUT);
serial.begin(9600);
}

```

### 3. Bagian *loop*

Bagian *loop* adalah fungsi utama program yang akan dijalankan berulang-ulang. Berikut ini contoh penulisan program sketch lengkap.

```

#include<AVR.io.h>
int lampul = 3;

void setup()
{ pinMode(lampul, OUTPUT);}

void loop()
{
    digitalWrite(lampul, HIGH);
    delay(1000);
    digitalWrite(lampul, LOW);
    delay(1000);
}

```

#### Penjelasan program :

```

#include<AVR/io.h>
int tomboll = 3;
char lampul = 0;

```

Bagian ini merupakan deklarasi awal yang menyatakan variabel “lampul” adalah pin no3 board arduino Uno dan menambahkan file io.h ke program.

```

void setup()
{ pinMode(lampul, OUTPUT);}

```

Bagian ini mengatur variabel “lampul” atau pin3 sebagai pin *output*. Karena pin no 3 adalah pin digital dan diatur sebagai pin *output*, maka pin ini hanya dapat mengeluarkan tegangan 5V ketika *HIGH* dan tegangan 0V ketika *LOW*.

```

void loop()
{
    digitalWrite(lampul, HIGH);
    delay(1000);
    digitalWrite(lampul, LOW);
}

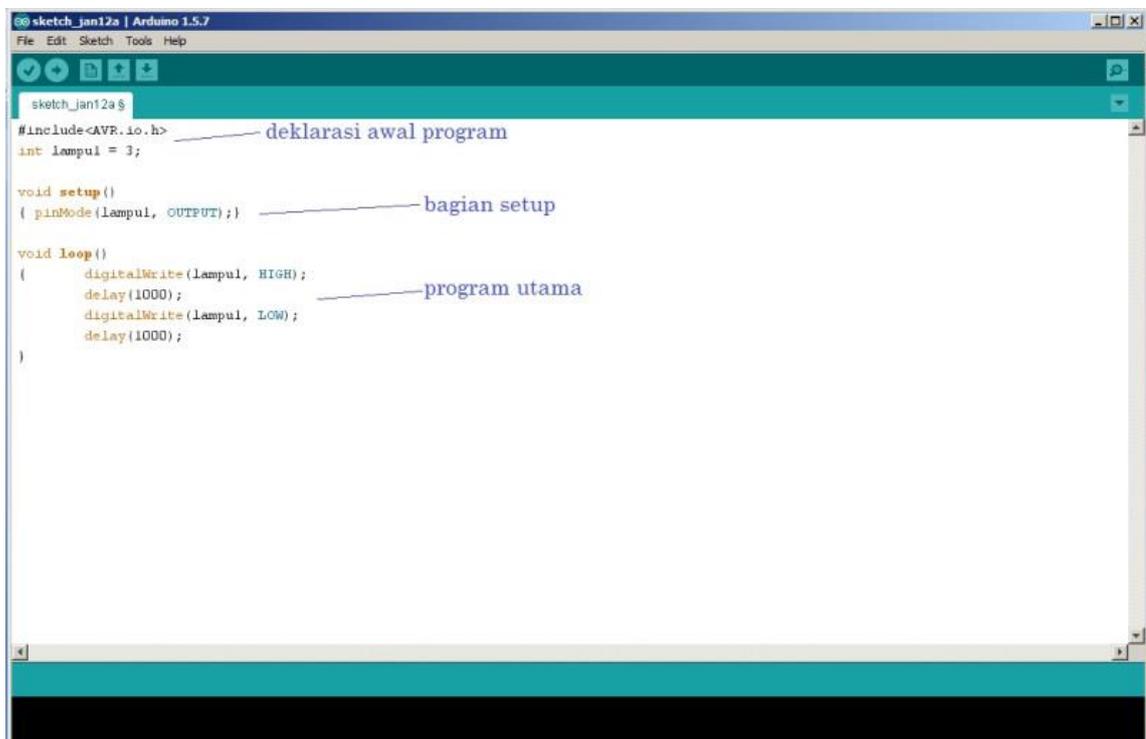
```

```

    delay(1000);
}

```

Bagian ini adalah program utama yang akan dijalankan berulang-ulang oleh mikrokontroler. Pertama-tama mikrokontroler akan mengeluarkan tegangan 5V (logika *HIGH*) pada pin3 (lampu1), kemudian menahan nyala lampu selama 1000 ms, setelah itu lampu akan dimatikan selama 1000ms, dan berulang kembali. Gambar berikut ini menunjukkan contoh penulisan *sketch* pada *IDE* program Arduino (Kelasrobot.com.2016)



```

sketch_jan12a $
#include<AVR.io.h>
int lampu = 3;

void setup()
{ pinMode(lampu, OUTPUT); }

void loop()
{
  digitalWrite(lampu, HIGH);
  delay(1000);
  digitalWrite(lampu, LOW);
  delay(1000);
}

```

**Gambar 2.13** Contoh Penulisan Sketch pada IDE Program Arduino

(Sumber : Diolah Sendiri)

### 2.8.2. Variabel

Variabel harus dipahami lebih dahulu sebelum membuat suatu program mikrokontroler, karena di dalam program mikrokontroler, banyak variabel yang bekerja di dalamnya. Variabel yang digunakan harus ditentukan juga tipe datanya.

Variabel adalah nama yang dibuat dan disimpan di dalam memori mikrokontroler. Deklarasi suatu variabel dapat dilakukan tanpa pemberian nilai awal atau dapat juga langsung diberikan nilai awal. (Kelasrobot.com.2016)

### 2.8.3. Tipe Data

Tipe data yang dapat digunakan di dalam program sketch bermacam-macam antara lain:

#### 1. Boolean

Tipe data boolean hanya memiliki 2 data yaitu benar (*true*) dan salah (*false*). Tipe data boolean hanya membutuhkan 1 byte memori.

#### 2. Byte

Tipe data byte memiliki 8 bit data (0 – 255). Tipe data byte tidak memiliki nilai negative.

#### 3. Char

Char atau karakter adalah tipe data untuk menyatakan suatu karakter seperti “A”, “B”, “C”, “+”, “.” dan sebagainya. Karakter ini disimpan dalam bentuk angka. Untuk mengkonversikan bentuk angka ke bentuk karakter dapat menggunakan tabel *ASCII*.

#### 4. Unsigned Char

Unsigned Char serupa dengan tipe char tetapi tanpa nilai negatif sehingga unsigned char memiliki nilai dari 0 hingga 255 atau sama dengan tipe data byte.

#### 5. Int

Int digunakan untuk menyatakan tipe data integer (bilangan bulat). Tipe data integer berkisar antara -32768 sampai 32768 (-215 hingga (215-1)). Nilai bilangan ini membutuhkan 2 byte memori mikrokontroler.

#### 6. Unsigned Int

Unsigned Int digunakan untuk mendeklarasikan tipe data bilangan bulat positif atau bernilai 0 hingga 65535. Sama dengan integer, tipe data ini juga membutuhkan 2 byte memori mikrokontroler.

#### 7. Word

Tipe data Word sama dengan tipe data unsigned int. Besar data word adalah 16 bit atau membutuhkan 2 byte memori mikrokontroller.

#### 8. Long

Tipe data Long adalah tipe data untuk menampung bilangan bulat yang berkisar antara -2.147.483.648 hingga 2.147.483.647. variabel bertipe long ditulis dengan akhiran L atau l. Contoh 456789980L.

#### 9. Unsigned Long

Unsigned Long adalah tipe data yang sama dengan long, tetapi dihitung dari angka 0 atau mempunyai nilai berkisar 0 hingga 4.292.967.295. Nilai variabel ini ditulis dengan kode UL diakhir konstanta.

#### 10. Float

Tipe ini berguna untuk menyimpan bilangan real. Angka yang bisa disimpan dari  $-3,4028235 \times 10^{38}$  hingga  $3,4028235 \times 10^{38}$ . Angka dengan tipe float sangat besar sekali, sehingga sangat jarang digunakan karena akan memperlambat kerja prosessor mikrokontroler dan banyak memakai memori, kecuali bila memang sangat dibutuhkan dalam program.

#### 11. Double

Tipe data double dan float untuk arduino tidak ada bedanya. (Kelasrobot.com.2016)

### 2.8.4. Konstanta

Konstanta adalah nilai suatu besaran yang tidak berubah besarnya atau memiliki nilai yang tetap. Dalam pemograman arduino ditulis dengan kata “const” yang artinya konstanta. Contoh penulisan konstanta :

```
const int harga = 12;
```

Artinya variabel “harga” adalah sebuah konstanta bilangan bulat integer yang memiliki nilai tetap yaitu 12. Nilai ini hanya bisa dibaca dan tidak dapat diubah selama program dijalankan. (Kelasrobot.com.2016)

### 2.8.5. Operator

Pernyataan atau disebut juga ekspresi adalah suatu rangkaian dari operator, variabel, fungsi atau konstanta yang ditujukan untuk menghasilkan suatu nilai dengan tipe data tertentu. (Kelasrobot.com.2016)

## 2.9. Definisi *Complementary Filter*

*Complementary Filter* merupakan *recursive digital filter* yang dapat mengestimasi kondisi proses apapun dengan sangat efektif. *Complementary Filter* digunakan sebagai algoritma untuk menyediakan kondisi estimasi yang dapat dipercaya dari keadaan proses. *Complementary Filter* juga digunakan untuk mengontrol sistem yang sensitif terhadap noise dari lingkungan karena meminimalkan square error. Filter ini dapat mengurangi pengukuran yang terkena noise dari sensor-sensor sebelum masuk kedalam sistem kontrol. Dalam algoritma yang diterapkan pada mikrokontroler dengan perangkat lunak, digunakan persamaan matematika diskrit. Sistem yang akan diukur harus dimodelkan oleh sistem linier. (CreativeRobot.com.2016)

### 2.9.1. Low Pass Filter

Low Pass Filter (LPF) atau Filter Lolos Bawah adalah filter yang hanya melewatkan sinyal dengan frekuensi yang lebih rendah dari frekuensi cut-off ( $f_c$ ) dan akan melemahkan sinyal dengan frekuensi yang lebih tinggi dari frekuensi cut-off ( $f_c$ ). Pada filter LPF yang ideal sinyal dengan frekuensi diatas frekuensi cut-off ( $f_c$ ) tidak akan dilewatkan sama sekali (tegangan output = 0 volt). Rangkaian low pass filter RC merupakan jenis filter pasif, dengan respon frekuensi yang ditentukan oleh konfigurasi R dan C yang digunakan. (elektronika-dasar.web.id.2016)

### 2.9.2. High Pass Filter

Filter high-pass atau sering juga disebut dengan filter lolos atas adalah suatu rangkaian yang akan melewatkan suatu isyarat yang berada diatas frekuensi cut-off ( $\omega_c$ ) sampai frekuensi cut-off ( $\omega_c$ ) rangkaian tersebut dan akan menahan

isyarat yang berfrekuensi dibawah frekuensi cut-off ( $\omega_c$ ) rangkaian tersebut. (elektronika-dasar.web.id.2016)

## 2.10. Kontrol PID

**Sistem Kontrol PID** ( *Proportional–Integral–Derivative controller* ) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut ( Feed back ).

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol **P** (*Proportional*), **D** (*Derivative*) dan **I** (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran system terhadap masukan tertentu sebagaimana yang diinginkan

### 1. Kontrol Proporsional

Kontrol P jika  $G(s) = k_p$ , dengan k adalah konstanta. Jika  $u = G(s) \cdot e$  maka  $u = K_p \cdot e$  dengan **Kp** adalah **Konstanta Proporsional**. Kp berlaku sebagai **Gain (penguat)** saja **tanpa memberikan efek dinamik** kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya rise time dan settling time.

### 2. Kontrol Integratif

Jika  $G(s)$  adalah kontrol I maka u dapat dinyatakan sebagai  $u(t) = [\int e(t)dt]K_i$  dengan **Ki** adalah **konstanta Integral**, dan dari persamaan diatas,  $G(s)$  dapat dinyatakan sebagai  $u = K_i \cdot \frac{1}{s} \cdot e(s)$  Jika  $e(T)$  mendekati **konstan (bukan nol)** maka  $u(t)$  akan menjadi **sangat besar** sehingga diharapkan dapat **memperbaiki error**. Jika  $e(T)$  mendekati **nol** maka **efek kontrol I ini semakin kecil**. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan  $K_i$  yang tidak tepat dapat menyebabkan respon

transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan  $K_i$  yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde system

### 3. Kontrol Derivatif

Sinyal kontrol  $u$  yang dihasilkan oleh kontrol D dapat dinyatakan sebagai  $G(s) = s.K_d$  Dari persamaan di atas, nampak bahwa sifat dari **kontrol D** ini dalam konteks "**kecepatan**" atau **rate dari error**. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri

Untuk mendapatkan aksi kontrol yang baik diperlukan langkah coba-coba dengan kombinasi antara P, I dan D sampai ditemukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  seperti yang diinginkan.

- (1) Memahami cara kerja system,
- (2) Mencari model sistem dinamik dalam persamaan differensial,
- (3) Mendapatkan fungsi alih sistem dengan Transformasi Laplace,
- (4) Memberikan aksi pengontrolan dengan menentukan konstanta  $K_p$ ,  $K_i$  dan  $K_d$ ,
- (5) Menggabungkan fungsi alih yang sudah didapatkan dengan jenis aksi pengontrolan,
- (6) Menguji sistem dengan sinyal masukan fungsi langkah, fungsi undak dan impuls ke dalam fungsi alih yang baru,
- (7) Melakukan Transformasi Laplace balik untuk mendapatkan fungsi dalam kawasan waktu,
- (8) Menggambar tanggapan sistem dalam kawasan waktu. (*Sistem Instrumentasi dan Sistem Kontrol*, hal 116, 2006)