

BAB II

TINJAUAN PUSTAKA

II.1 Perancangan

Perancangan adalah kajian mengenai penentuan kerangka dasar kegiatan pengumpulan informasi terhadap objek yang memiliki variasi, berdasarkan prinsip-prinsip statistika. Bidang ini merupakan salah satu cabang penting dalam statistika, inferensial dan diajarkan di banyak cabang ilmu pengetahuan di perguruan tinggi karena berkaitan erat dengan pelaksanaan percobaan. Perancangan dapat dikatakan sebagai "jembatan" bagi peneliti untuk bergerak dari hipotesis menuju pada percobaan agar memberikan hasil yang valid secara ilmiah. Dengan demikian, perancangan dapat dikatakan sebagai salah satu instrumen dalam metode ilmiah. Kajian perancangan adalah pelaksanaan percobaan terkendali. Desain atau perancangan dalam perangkat lunak merupakan upaya untuk mengkontruksi sebuah sistem yang memberikan kepuasan akan spesifikasi kebutuhan fungsional untuk memenuhi target, memenuhi kebutuhan secara jelas dari segi performasi maupun pengguna sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu dan perangkat[5].

II.2 Robot Keseimbangan

Robot keseimbangan adalah sebuah robot yang mampu berdiri tegak seimbang yang pada umumnya bertumpu pada dua roda atau satu. Proses keseimbangan pada robot ini biasa disebut sebagai stabilitas. Dua roda yang diletakkan pada dasar permukaan dan memungkinkan *chassis* robot untuk mempertahankan posisi tegak dengan bergerak ke arah kemiringan, baik maju ataupun mundur dalam upaya untuk menjaga pusat massa di atas as roda[4].

II.3 Sensor

Sensor adalah suatu piranti yang digunakan untuk melakukan suatu pengamatan terhadap suatu rangsangan dan mengubahnya ke dalam bentuk isyarat sehingga bisa diukur. Rangsangan dapa berupa akustik, elektrik, optic, termal maupun mekanik.

Ada dua jenis sensor, yaitu sensor analog dan sensor digital. Hal ini didasarkan pada jenis isyarat keluaran yang dihasilkan. Sensor analog berarti bahwa sensor menghasilkan isyarat analog, sedangkan sensor digital membangkitkan isyarat digital. Pada sensor digital, keluaran hanya berupa salah satu dari keadaan, yakni *HIGH* dan *LOW*. Pada sensor analog keluaran lebih bervariasi[2].

II.4 PID (*Proportional, Integral, Derivative*)

Kontroler PID (dari singkatan bahasa Inggris: *Proportional–Integral–Derivative controller*) merupakan kontroler mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri. Sebuah kontroler PID secara kontinyu menghitung *nilai kesalahan* sebagai beda antara setpoint yang diinginkan dan variabel proses terukur. Kontroler mencoba untuk meminimalkan nilai kesalahan setiap waktu dengan penyetelan *variabel kontrol*, seperti posisi keran kontrol, damper, atau daya pada elemen pemanas, ke nilai baru yang ditentukan oleh jumlahan:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Dengan :

K_p : Gain proposional, parameter turning

K_i : Gain integral, parameter turning

K_d : Gain Derivatif, parameter turning

e : error = $Y_{sp} - Y_m$

y_{sp} : setpoint

Y_m : Variabel proses

t : waktu

τ : Variabel integrasi ; nilainya dambil dari waktu nol sampai τ

Dengan K_p , K_i , dan K_d , semuanya positif, menandakan koefisien untuk term proporsional, integral, dan derivatif, secara berurutan (atau P , I , dan D). Pada model ini,

- **P** bertanggung jawab untuk nilai kesalahan saat ini. Contohnya, jika nilai kesalahan besar dan positif, maka keluaran kontrol juga besar dan positif.
- **I** bertanggung jawab untuk nilai kesalahan sebelumnya. Contoh, jika keluaran saat ini kurang besar, maka kesalahan akan terakumulasi terus menerus, dan kontroler akan merespon dengan keluaran lebih tinggi.
- **D** bertanggung jawab untuk kemungkinan nilai kesalahan mendatang, berdasarkan pada rate perubahan tiap waktu.

Karena kontroler PID hanya mengandalkan variabel proses terukur, bukan pengetahuan mengenai prosesnya, maka dapat secara luas digunakan. Dengan penyesuaian (*tuning*) ketiga parameter model, kontroler PID dapat memenuhi kebutuhan proses. Respon kontroler dapat dijelaskan dengan bagaimana responnya terhadap kesalahan, besarnya overshoot dari setpoint, dan derajat osilasi sistem. penggunaan algoritma PID tidak menjamin kontrol optimum sistem atau bahkan kestabilannya [8].

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

1. Kontrol Proporsional

Kontrol P jika $G(s) = kp$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka $u = K_p \cdot e$ dengan K_p adalah Konstanta Proporsional. K_p 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

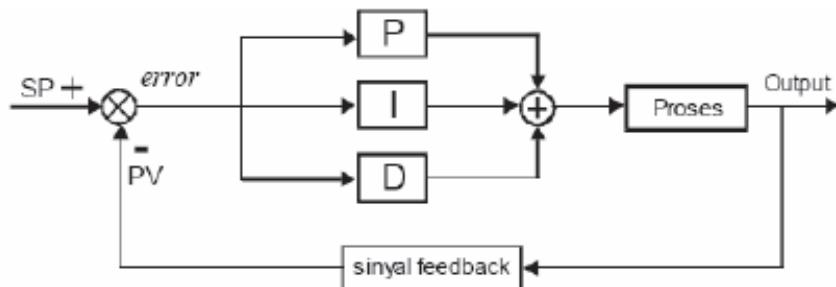
Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = (\text{integral}(t) \, dT) \, Ki$ dengan Ki adalah konstanta Integral, dan dari persamaan diatas, $G(s)$ dapat dinyatakan sebagai $u = Kd.[\delta e / \delta t]$. Jika $e(T)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat 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 Ki yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan Ki 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 $G(s) = s.Kd$

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

Karakteristik PID *controller* sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta K_p, T_i, dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diset lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan. Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler PID. Gambar 1 menunjukkan blok diagram pengontrol PID loop tertutup.



Gambar II.1. Blok Diagram PID controller Loop Tertutup

Dari blok diagram diatas dapat dijelaskan sebagai berikut:

1. $SP = Set\ Point$, secara simple maksudnya ialah suatu parameter nilai acuan atau nilai yang kita inginkan.
2. $PV = Present\ Value$, maksudnya ialah nilai bobot pembacaan sensor saat itu atau variabel terukur yang diumpan balikkan oleh sensor (sinyal *feedback* dari sensor).
3. Error = nilai kesalahan, yakni deviasi atau simpangan antar variabel terukur atau bobot sensor (PV) dengan nilai acuan (SP).

$$error = SP - PV$$

Proposional controller (K_p) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus kesalahan keadaan tunak, *Integral controller* (K_i) akan memberikan efek menghapus keadaan tunak, tetapi berakibat memburuknya respon transien, *Diferensial controller* (K_d) akan memberikan efek meningkatnya stabilitas sistem, mengurangi over-shot, dan menaikan respon transfer. Efek dari setiap kontroler (K_p , K_i , K_d) dalam sistem loop tertutup diperlihatkan pada Tabel II.1

Tabel II.1. Efek Dari Setiap Kontroler (K_p , K_i , K_d) Dalam Loop Tertutup.

| Respon Loop tertutup | Waktu naik | Over-shoot | Waktu turun | Kesalahan keadaan tunak |
|----------------------|-----------------|------------|-----------------|-------------------------|
| K_p | Menurun | Meningkat | Perubahan kecil | Menurun |
| K_i | Menurun | Meningkat | Meningkat | Hilang |
| K_d | Perubahan Kecil | Menurun | Menurun | Perubahan kecil |

II.5 Mikrokontroler

Adalah suatu chip atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Program yang direkam bertujuan agar rangkaian elektronik dapat membaca input, memproses, dan kemudian menghasilkan output sesuai yang diinginkan. Outputnya itu bisa berupa sinyal, besaran tegangan, lampu, suara, getaran, gerakan, dan sebagainya [3].

Pada prinsipnya mikrokontroler adalah sebuah komputer berukuran kecil yang dapat berinteraksi dengan peranti-peranti eksternal, seperti sensor ultrasonik untuk mengukur jarak terhadap suatu objek, penerima GPS untuk memperoleh data posisi kebumian dari satelit, dan motor untuk mengontrol gerakan pada robot. Sebagai komputer yang berukuran kecil, mikrokontroler cocok diaplikasikan pada benda-benda yang berukuran kecil, misalnya sebagai pengendali pada *QuadCopter* ataupun robot[2].

II.5.1 Arduino Uno ATmega328

Arduino adalah suatu perangkat prototipe elektronik berbasis mikrokontroler yang fleksibel dan *open-source*, perangkat keras dan perangkat lunaknya mudah digunakan. Perangkat ini ditujukan bagi siapapun yang tertarik/memanfaatkan mikrokontroler secara praktis dan mudah. Bagi pemula dengan mennggunakan *board* ini akan lebih mudah mempelajari pengendalian dengan mikrokontroler, bagi desainer pengontrol menjadi lebih mudah dalam membuat prototipe ataupun implementasi; demikian juga bagi para hobi yang mengembangkan mikrokontroler. Arduino dapat digunakan ‘mendeteksi’ lingkungan dengan menerima masukan dari berbagai sensor (misal: cahaya, suhu, inframerah, ultrasonic, jarak, tekanan, kelembaban) dan dapat ‘mengendalikan’ peralatan sekitarnya (misal: lampu, berbagai jenis motor, dan akuator lainnya) [1].



Gambar II.2 Bentuk Fisik Arduino Uno

Sumber:<http://www.arduino.cc/>

II.5.2 Mikrokontroler ATMega328

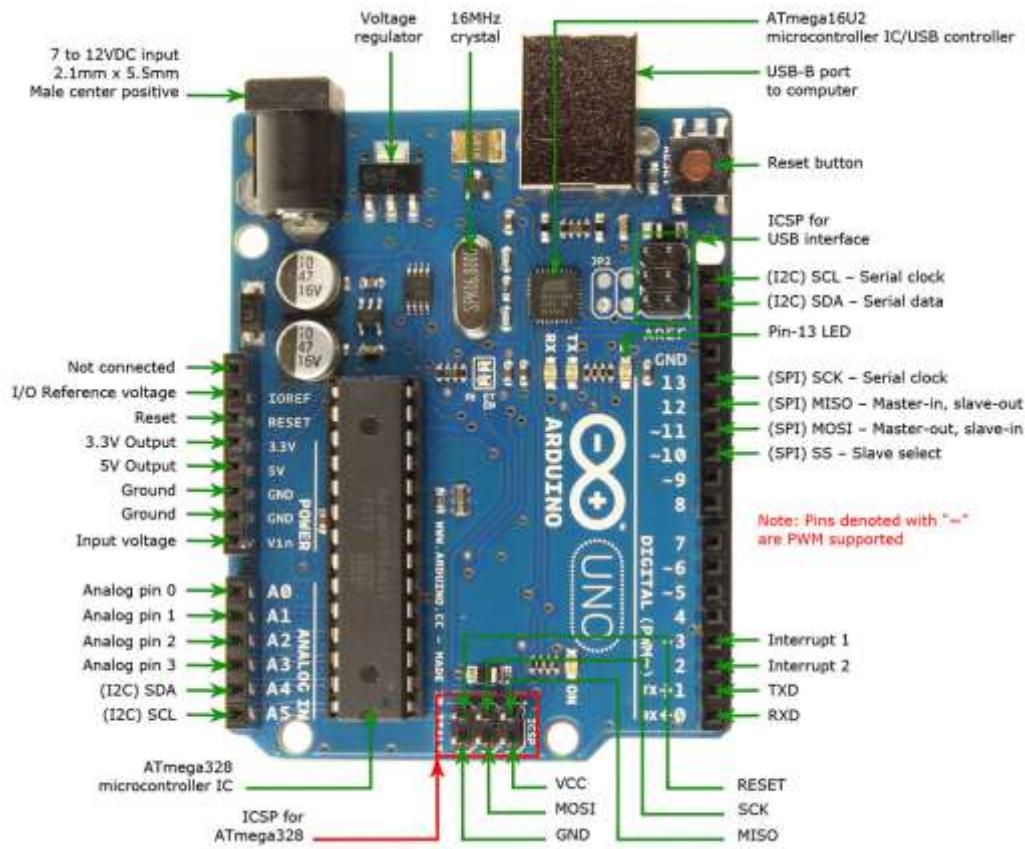
ATMega328 adalah mikrokontroller keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).

Mikrokontroller ini memiliki beberapa fitur antara lain :

- 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
- 32 x 8-bit register serba guna.
- Kecepatan mencapai 16 MIPS dengan clock 16 MHz.
- 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari flash memori sebagai *bootloader*.
- Memiliki *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi permanent karena *EEPROM* tetap dapat menyimpan data meskipun catu daya dimatikan.
- Memiliki *SRAM* (*Static Random Access Memory*) sebesar 2KB.
- Memiliki pin *I/O* digital sebanyak 14 pin 6 diantaranya *PWM* (*Pulse Width Modulation*) output dan memiliki 8 pin *I/O* analog.
- *Master / Slave SPI Serial interface.*

Mikrokontroller ATmega 328 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism* [9].

II.5.3 Konfigurasi Pin Arduino Uno ATmega328



Gambar II.3 Konfigurasi Pin Arduino Uno

Sumber:<http://www.arduino.cc/>

Pada gambar II.2 terlihat konfigurasi pin pada board arduino Uno R3. Pin-pin tersebut terdiri dari:

1. VIN

Adalah (*Supply Voltage*) input tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai ‘saingan’ tegangan 5 Volt dari koneksi USB atau sumber daya ter-regulator lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok

tegangan untuk papan melalui jack power, kita bisa mengakses/mengambil tegangan melalui pin ini.

2. 5V

Sebuah pin yang mengeluarkan tegangan ter-regulator 5 Volt, dari pin ini tegangan sudah diatur (ter-regulator) dari regulator yang tersedia (built-in) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari jack power DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada board (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3.3V secara langsung tanpa melewati regulator dapat merusak papan Arduino.

3. 3V3

Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (on-board). Arus maksimum yang dihasilkan adalah 50 mA.

4. GND

Pin Ground atau Massa.

5. AREF

Referensi tegangan untuk input analog. Digunakan dengan fungsi analogReference().

6. Input dan Output

Masing-masing dari 14 digital pin pada Arduino Nano dapat digunakan sebagai input atau output, menggunakan fungsi pinMode() , digitalWrite() , dan digitalRead(). Arduino Nano beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara default) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus contohnya Pin PWM. Selain pin digital arduino nano juga mempunyai 8 pin analog yang dapat mengeluarkan tegangan keluaran yang bervariasi[7].

II.6 Bluetooth

Bluetooth adalah protokol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain .

Besarnya jarak jangkauan tergantung pada kelas *Bluetooth*. Dalam kelas *transceiver* *Bluetooth* ada tiga kelas pembagian daya yaitu:

1. Daya kelas 1 beroperasi antara 100 Mw(20dBm) dan 1 Mw (0dBm), dirancang untuk perangkat dengan jangkauan yang jauh hingga mencapai 100 m
2. Daya kelas 2 beroperasi antara 2,5 Mw (4dBm) dan 0,25 Mw (-6dBm), dirancang untuk perangkat dengan jangkauan yang jauh hingga mencapai 10 m
3. Daya kelas 3 beroperasi antara 1 Mw(0dBm) dirancang untuk perangkat dengan jangkauan yang jauh hingga mencapai 1 m [3].

Beberapa jenis model *Bluetooth* yang dapat kita jumpai adalah modul HC – 03, HC – 04, HC – 05, dan HC – 06. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05. modul *Bluetooth* HC-05 merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda.



Gambar II.4. Modul Bluetooth HC – 05

Sumber:<http://www.google.com/>

HC-05 merupakan modul bluetooth *to serial* yang menggunakan protokol standar bluetooth V2.0 dan kebutuhan tegangan sebesar 3,3 V [10]

.Spesifikasi modul *bluetooth* HC – 05 :

- *Bluetooth* versi : Versi 2.0
- Frekuensi : 2.4 GHz
- *Power Supply* : 3,3 Volt
- Keburuan Arus : Pada saat pairing 30 – 40 mA , setelah pairing 8 mA
- Baudrate : 9600 (standar)

II.7 Android

Android adalah sistem operasi berbasis Linux yang digunakan pada perangkat *mobile* seperti *smartphone* dan komputer *tablet*. *Platform open source* yang dimiliki Android memungkinkan para *developer* untuk menciptakan aplikasinya sendiri.

Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Karena merupakan software yang berbasis open source banyak sekali developer yang mengembangkan sistem operasi dan software pendukung untuk aplikasi android ini.



Gambar II.5. Smartphone Berbasis Android dari Xiaomi

Sumber:<http://www.google.com/>

II.8 Perangkat Lunak (*Software*) dan Bahasa Pemrograman

Agar mikrokontroler dapat bekerja maka digunakan perangkat lunak dan pemrograman sebagai pengkondisian dan perintah-perintah yang diinginkan oleh pembuat alat. Perangkat lunak yang penulis gunakan adalah Arduino IDE.

Software IDE arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari platform *wiring*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang, *hardware*-nya menggunakan prosesor Atmel AVR dan *software*-nya memiliki bahasa pemrograman C++ yang sederhana dan fungsi-fungsinya yang lengkap, sehingga arduino mudah dipelajari oleh pemula [1].



```
Bluetooth
char val;
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if(Serial.available() > 0) {
    val = Serial.read();
    Serial.println(val);
  }

  if(val== '1' ) {
    digitalWrite(2,HIGH);
  }
  if(val== '2' ) {
    digitalWrite(3,HIGH);
  }
  if(val== '3' ) {
    digitalWrite(4,HIGH);
  }
}
```

Gambar II.6. Tampilan Software Editor Arduino
Sumber:<http://www.arduino.cc/>

Arduino IDE menggunakan bahasa pemrograman C++ dengan versi yang telah disederhanakan, sehingga lebih mudah dalam belajar pemrograman. Arduino menggunakan software processing yang digunakan untuk menulis program kedalam Arduino. Processing sendiri merupakan penggabungan antara bahasa C++ dan Java. Software IDE Arduino terdiri dari 3 (tiga) bagian:

1. *Editor program, untuk menulis dan mengedit program dalam bahasa processing.*
2. *Compiler, modul yang berfungsi mengubah bahasa processing (kode program) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontroler.*
3. *Uploader, modul yang berfungsi memasukkan kode biner kedalam memori mikrokontroler.*

Struktur perintah pada Arduino secara garis besar terdiri dari 2 (dua) bagian yaitu void setup dan void loop. Void setup berisi perintah yang akan dieksekusi hanya satu kali sejak Arduino dihidupkan sedangkan void loop berisi perintah yang akan dieksekusi berulang-ulang selama arduino dinyalakan.