

Purwarupa Sistem Peringatan Dini dan Kendali Pintu Air Bendungan dengan Kendali PID

Benni Sahputra*¹, Panggih Basuki²

¹ Prodi Elektronika dan Instrumentasi Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM

² Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM

e-mail: *¹ benz_linkins@yahoo.co.id, ² panggihbasuki11@gmail.com

Abstrak

Tujuan dari penelitian ini adalah untuk membuat rancang bangun alat Purwarupa Sistem Peringatan Dini dan Kendali Pintu Air Bendungan Dengan Kendali P.I.D (Proportional, Integratif, Derivatif). Untuk mendeteksi ketinggian air dan pintu air digunakan sensor SRF04. Terdapat dua sensor, sensor SRF04 (1) untuk mendeteksi ketinggian pintu dan sensor SRF04 (2) untuk mendeteksi ketinggian air. Sensor SRF04 (2) akan mendeteksi ketinggian air berdasarkan range setpoint yang telah ditentukan kemudian akan di proses oleh mikrokontroler. Keluaran dari mikrokontroler berupa PWM yang dihasilkan oleh kendali PID yang tertanam pada mikrokontroler akan diteruskan ke rangkaian driver motor untuk menggerakkan pintu air secara buka tutup. Motor DC berfungsi untuk menggerakkan pintu air secara buka tutup berdasarkan setpoint-setpoint tertentu. Adapun setpoint dari ketinggian pintu air adalah 5cm, 10cm, dan 15cm. Sistem juga akan mengirimkan informasi ketinggian dan status air berupa SMS berdasarkan range setpoint dari ketinggian air menggunakan modul GSM. Konstanta P.I.D yang dihasilkan dari penelitian ini adalah $K_p = 100$, $K_i = 0,2$, $K_d = 50$. Ketika mencapai setpoint 10cm, konstanta ini mempunyai respon time paling baik dibandingkan dengan konstanta-konstanta lainnya dimana rise time = 7,98 detik, settling time = 11,64 detik, steady state time = 11,76 detik.

Kata kunci—Rangkaian driver, modul GSM, SRF04, Kendali P.I.D

Abstract

The purpose of this research is to create a prototype of early warning system and dam sluiceway control by P.I.D control (proporsional, integratif, derivatif). To detect the water level and sluiceway used sensor SRF04. There are two sensors, sensor SRF04 (1) to detect the height of the sluiceway and sensor SRF04 (2) to detect water level. SRF04 sensor (2) detects the water level based on range of setpoint, then will be processed by the microcontroller. PWM output of the microcontroller by the PID control is embedded in the microcontroller will be forwarded to the motor driver circuit to drive the sluiceway are open and closed. Motor DC is used to drive the floodgates are open and close based on setpoint. The setpoint of the sluiceway is 5cm, 10cm, and 15cm. The system also will transmit information level and the status of water in the form of SMS based setpoint range of water levels used GSM modul. Result of PID constants from this research is $K_p = 100$, $K_i = 0.2$, $K_d = 50$. When reaching setpoint 10cm, the constants have the best response time compared with the other constants which rise time = 7,98 sec, settling time = 11,64 sec, steady state time = 11,76 sec.

Keywords—Driver circuit, GSM modul, SFR04, P.I.D control

1. PENDAHULUAN

Ketinggian permukaan air merupakan suatu parameter yang banyak dipantau dan dianalisa perubahannya, terutama pada musim dan keadaan tertentu. Hal ini berkaitan erat dengan banyaknya kerusakan yang mungkin disebabkan olehnya, seperti banjir, tsunami dan lain sebagainya. Selama ini pemantauan ketinggian air yang dilakukan masih menggunakan alat-alat manual berupa skala ketinggian air yang diletakkan di pinggir sungai, jembatan, atau pintu air. Hal ini memiliki keterbatasan terutama terhadap penupukan sedimen di dasar sungai, sehingga mengurangi akurasi dari pengukuran. Masalah lain adalah masih manualnya pintu air di suatu bendungan. Untuk membuka dan menutup pintu masih manual menggunakan tenaga manusia. Adapun penelitian-penelitian sebelumnya yang hanya melakukan pemantauan ketinggian air tanpa ada respon ketika air yang dipantau mengalami ketinggian maksimum.

Mengingat akan pentingnya pemantauan dan pengendalian terhadap ketinggian air sungai/bendungan terutama pada daerah-daerah dengan tinggi daratan yang rendah dari permukaan laut, maka dibuat dan dirancang suatu sistem pemantauan serta pengendalian ketinggian air secara elektrik yang dapat digunakan untuk peringatan dini akan terjadinya musibah banjir yang disebabkan meluapnya air pada bendungan melalui sms dan pengendalian ketinggian air menggunakan pintu air yang bekerja secara otomatis.

Dengan menggunakan modul GSM sebagai perantara peringatan dini terhadap penduduk sekitar dan pengendalian ketinggian air menggunakan pintu air yang bekerja secara buka tutup, sensor ultrasonik sebagai detektor guna mendeteksi jarak ketinggian permukaan air [1]. Hasil penelitian ini diharapkan dapat menjadi suatu cara alternatif untuk memantau dan mengendalikan ketinggian air sungai/bendungan dengan kendali PID.

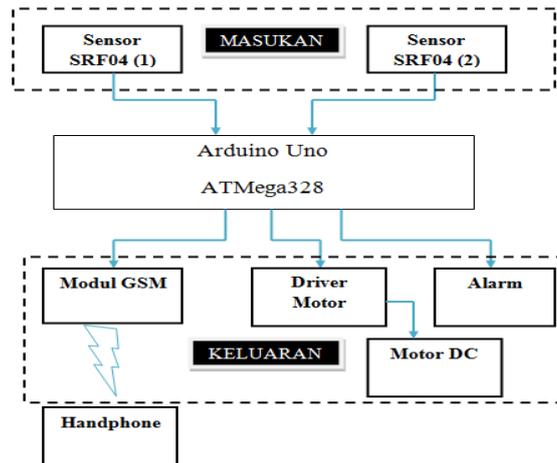
2. METODE PENELITIAN

2.1 Analisis Sistem

Sistem ini merupakan purwarupa sistem peringatan dini dan kendali pintu air bendungan dengan kendali PID. Kendali PID digunakan untuk mengendalikan pintu air yang digerakan oleh motor DC. Kendali PID akan mengoreksi *error* yang terjadi yang disebabkan oleh selisih antara *proses variable* dan *setpoint* sehingga pintu air dapat mencapai *setpoint-setpoint* yang telah ditentukan dengan baik.

2.2. Perancangan Sistem

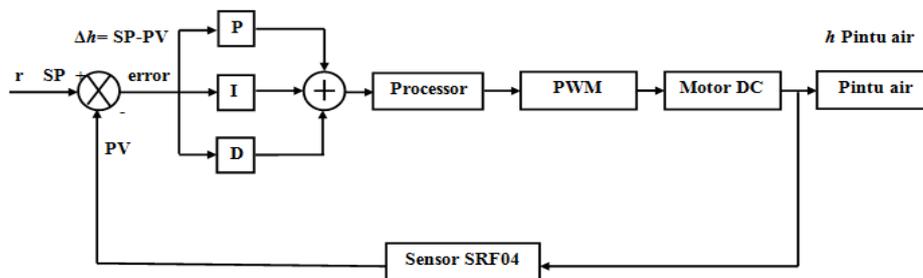
Rancangan perangkat keras dari purwarupa ini terdiri dari atas beberapa blok. Blok diagram sistem secara umum ditunjukkan pada Gambar 1. Pada bagian masukan terdapat dua sensor SRF04, sensor ini digunakan untuk mengukur ketinggian air dan ketinggian pintu air. Sebagai mikrokontroler digunakan arduino uno sebagai otak dari sistem yang berfungsi mengendalikan semua kerja sistem. Mikrokontroler menerima data dari sensor yang kemudian mengolah, menyimpan, dan memberikan instruksi ke bagian keluaran. Pada bagian keluaran terdiri atas beberapa bagian yaitu modul GSM, motor DC, dan alarm. Modul GSM berfungsi sebagai indikator sekaligus antarmuka ke pengguna untuk mengetahui ketinggian air yang sudah ditentukan sebelumnya (*setpoint*). Motor DC berfungsi sebagai aktuator penggerak pintu air yang bergerak secara buka tutup menggunakan *driver mosfet*. Alarm berfungsi sebagai indikator ketika ketinggian air sudah mencapai keadaan maksimum.



Gambar 1 Blok diagram sistem

2.2.1 Rancangan Sistem Pengontrol P.I.D

Pengontrol PID adalah pengontrol utama yang digunakan pada sistem ini ditunjukkan pada Gambar 2. Parameter masukan utama yang digunakan dalam pengontrol PID adalah nilai *error* atau selisih nilai yang luaran sistem (variabel proses) dengan nilai yang diharapkan (*set point*).



Gambar 2 Blok diagram kontrol kecepatan motor dengan PID

Parameter yang digunakan dalam mengontrol kecepatan motor adalah ketinggian pintu air yang terdapat sensor SRF04 terhadap nilai *desire state* dari masing-masing setpoint. Sedangkan parameter *outputnya* adalah pulsa PWM dengan lebar tertentu yang akan digunakan untuk mengatur kecepatan putar pada motor dc.

Selanjutnya untuk melakukan proses stabilisasi sistem, digunakan pengontrol PID. Rumus umum dari pengontrol PID ialah sebagai berikut:

$$\begin{aligned}
 e(t) &= x_d(t) - x(t), \\
 u(t) &= K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}
 \end{aligned}
 \tag{1}$$

dimana $u(t)$ adalah kontrol input dan $e(t)$ adalah perbedaan antara keadaan yang diinginkan $x_d(t)$ dan keadaan sekarang $x(t)$. K_P , K_I dan K_D adalah parameter atau kontanta untuk elemen *proporsional*, *integratif* dan *derivatif* dari control PID [2]. Bentuk umum dari rumus kontroler PD pada persamaan (1) adalah sebagai berikut:

$$u(k) = K_p \cdot e_k + \frac{1}{T} K_D (e_k - e_{k-1})
 \tag{2}$$

Selanjutnya pada Persamaan (2) tersebut ditambahkan komponen *Integratif* untuk mengoreksi *steady state error* yang terjadi pada sistem sehingga dihasilkan kontroler PID yang dirumuskan sebagai berikut [3] :

$$u(k) = K_p \cdot e_k + K_I \cdot T \sum_0^k e_k + \frac{1}{T} K_D (e_k - e_{k-1}) \quad (3)$$

Persamaan (3) tersebut kemudian diterapkan pada program yang akan dibuat dengan rumus pada persamaan (4):

$$U(k) = K_p * error + K_i * integralError + K_d*(error - last_error); \quad (4)$$

dengan,

$$Error = SP - PV$$

$$integralError = last_integralError + error * Ts;$$

dimana,

- $U(k)$ = Nilai *plant PID* berupa nilai PWM yang akan diolah menjadi kecepatan motor untuk mengatur ketinggian pintu air
- K_p = Nilai *Konstanta Proporsional*
- K_i = Nilai *Konstanta Integratif*
- K_d = Nilai *Konstanta Derivatif*
- $last_error$ = Nilai *error* terakhir yang dideteksi sistem (nilai *error* t^{-1})
- $last_integralError$ = Nilai *integral error* terakhir
- SP = *Set Point*
- PV = *Process Variable*
- Ts = *Time sampling* sistem, digunakan 0,06s

Mikrokontroler akan mengolah sinyal masukan dari sensor jarak SRF04. Selanjutnya mikrokontroler akan membandingkan nilai jarak tersebut dengan nilai *desire state* atau *set point* yang telah ditentukan pada saat kalibrasi sistem. nilai *set point* yang digunakan ada 4, yaitu :

- *setpoint* 2cm saat ketinggian air pada bak penampung 0 hingga 5cm
- *setpoint* 5cm saat ketinggian air pada bak penampung 5 hingga 10cm
- *setpoint* 10cm saat ketinggian air pada bak penampung 10 hingga 15cm
- *setpoint* 15cm saat ketinggian air pada bak penampung 15 hingga 20cm

Penentuan *setpoint* disesuaikan dengan rancangan mekanik. Penentuan *setpoint* 2cm sebagai *setpoint* terendah dikarenakan supaya pintu tidak tertutup penuh ketika terjadi osilasi yang menyebabkan pintu akan menyentuh dasar dari kedudukan pintu yang akan menyebabkan pintu rusak. Sedangkan penentuan *setpoint* 5cm dan 10cm adalah dengan tujuan mempertahankan tinggi air agar tetap stabil dan tidak melewati batas maksimum ketika air dimasukkan ke penampung air. Keluaran air yang keluar dari pintu air dengan tinggi pintu 5cm dan 10cm di anggap proporsional terhadap masukan air. Penentuan *setpoint* 15cm sama halnya dengan penentuan *setpoint* 5cm dan 10cm. Tetapi penentuan *setpoint* 15cm juga berdasarkan tinggi maksimum pintu air dari *plant* dimana apabila melebihi 15cm maka pintu akan mengalami pergerakan yang tidak stabil dikarenakan tinggi pintu sudah melebihi jalur atau rel pintu yang didesain untuk pergerakan pintu air pada *plant*.

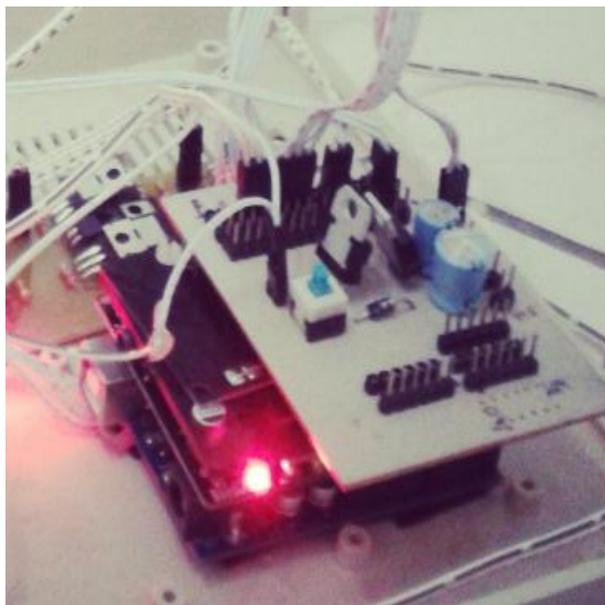
Apabila terdapat selisih (*Error*) antara nilai jarak yang terbaca dengan nilai *setpoint* yang telah ditetapkan sebelumnya, maka mikrokontroler akan menganggap bahwa sistem terganggu dan akan melakukan pengkoreksian terhadap *error* tersebut dengan cara menggerakkan motor untuk mengatur ketinggian pintu air.

Sub fungsi pokok dari program utama dalam pengendalian sistem adalah membaca nilai sensor jarak kemudian mempertahankan posisi ketinggian pintu air agar sesuai dengan referensi menggunakan pengontrol PID. Selisih nilai jarak dengan *desire state* atau *setpoint* kemudian akan diolah dengan *PID Controller* untuk menentukan kecepatan motor dc pada terhubung pada

pintu air untuk mengatur ketinggian pintu air. Langkah tersebut digunakan untuk mengatur ketinggian air pada bak penampungan air agar tidak tumpah. Metode untuk pencarian konstanta KP, KI, KD yang akan digunakan dalam sistem kontrol PID ini menggunakan metode eksperimen, dengan cara mencoba konstanta tersebut satu per satu untuk memperoleh nilai KP, KI, KD yang tepat pada system.

2.3 Implementasi Perangkat Keras Sistem

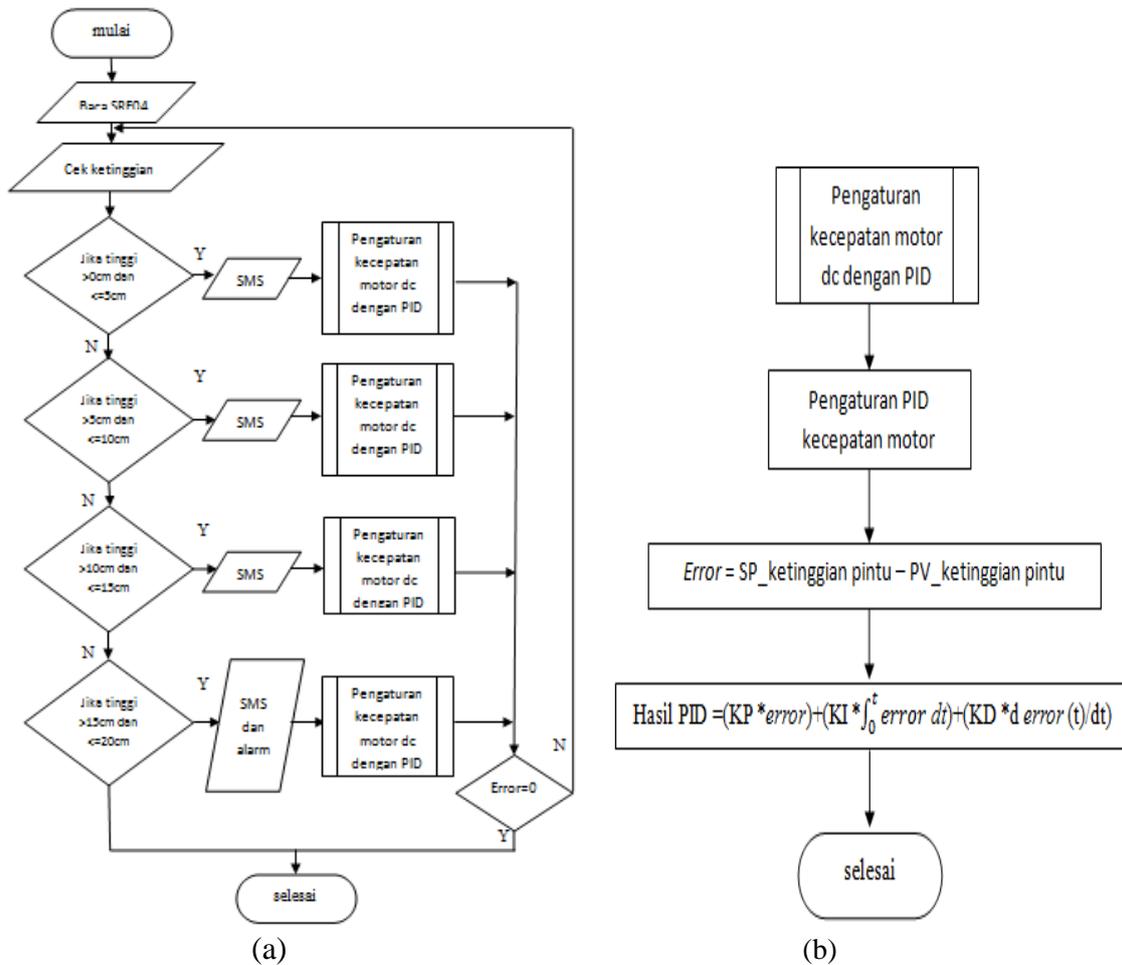
Rangkaian mikrokontroler Arduino Uno ATmega328 merupakan sistem utama yang mengendalikan seluruh aktivitas perangkat elektronik lainnya. Pada Arduino Uno ATmega328 tersebut dipasang dengan modul GPRS V2.0. modul tersebut merupakan modul GSM khusus untuk arduino. Setelah rangkaian Arduino Uno ATmega328 dan modul GPRS V2.0 terpasang, diberikan *shield* untuk mempermudah pemasangan antara Arduino Uno ATmega328 dan modul GPRS V2.0 dengan sensor SRF04, motor dc, dan alarm. Rangkaian mikrokontroler Arduino Uno ATmega328 dan modul GPRS V2.0 sudah tersedia dalam bentuk modul siap pakai sehingga pengguna hanya memerlukan *shield* tambahan yang terhubung dengan bagian *input/output* system [4]. Terdapat 2 sensor SRF04 yang terhubung dengan pin 5, 6 dan 12, 13 Arduino Uno ATmega. Untuk pin output PWM digunakan pin 9 dan 10 yang selanjutnya dihubungkan dengan motor dc, dan pin 2 terhubung dengan alarm. Perintah pengiriman *sms* digunakan pin software serial 7 dan 8 pada modul GPRS V2.0. Rangkaian mikrokontroler Arduino Uno ATmega328 dan modul GPRS V2.0 beserta *shield* ditunjukkan pada Gambar 3 berikut ini.



Gambar 3 Rangkaian Arduino Uno ATmega328+GPRS Shield V2.0+Shield

2.4 Implementasi Perangkat Lunak Sistem

Program dibuat dengan menggunakan bahasa pemrograman C++ dengan menggunakan *Arduino IDE (Integrated Development Environment)*. Dalam *Arduino IDE* ini juga telah terdapat fungsi dan *library* untuk menjalankan *feature-feature* khusus yang dimiliki mikrokontroler Aduino Uno ATmega328. *Arduino IDE* tersebut akan mengkompilasi bahasa C++ yang ditulis kedalam bahasa mesin yang dimengerti oleh mikrokontroler Arduino Uno ATmega328[3]. Program pada Arduino Uno ATmega328 terdiri dari 4 bagian, yaitu program yang menangani proses kalibrasi sensor SRF04 (2) untuk mendeteksi ketinggian air, program pengiriman SMS yang menggunakan intruksi AT-commands, program untuk menyalakan buzzer, program utama yang menangani sistem kecepatan motor dc dan sensor SRF04 (1) untuk menggerakkan pintu air menggunakan pengontrol PID.



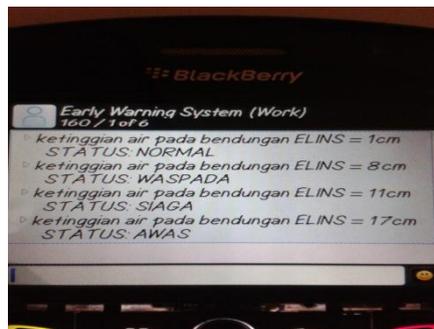
Gambar 4 Diagram alir sistem (a) ; diagram subfungsi (b)

Gambar 4 (a) merupakan diagram alir dari sistem yang terdapat subfungsi di dalamnya dimana pada diagram alir sistem terdapat kendali PID untuk mengendalikan kecepatan motor DC. Gambar 4 (b) merupakan diagram subfungsi dari diagram alir sistem.

3. HASIL DAN PEMBAHASAN

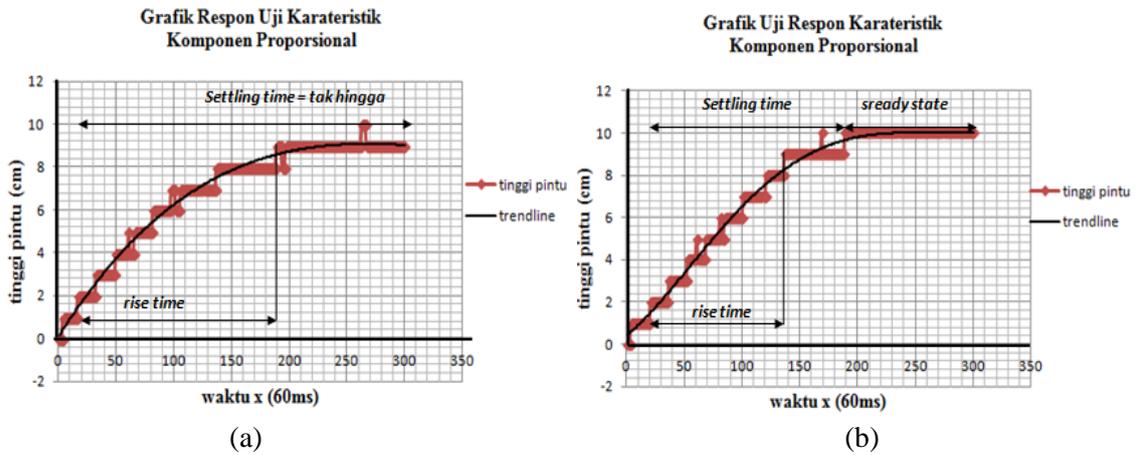
3.1 Hasil Penelitian

Gambar 5 merupakan wujud implementasi dari ketinggian air dari sistem yang ditampilkan pada handphone. Pengiriman sms berdasarkan setpoint dari ketinggian air.

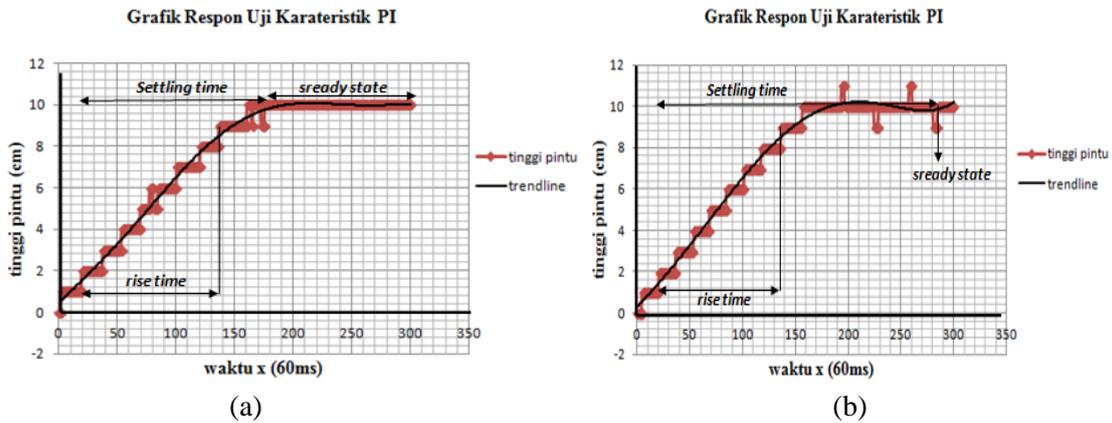


Gambar 5 Tampilan SMS pada handphone

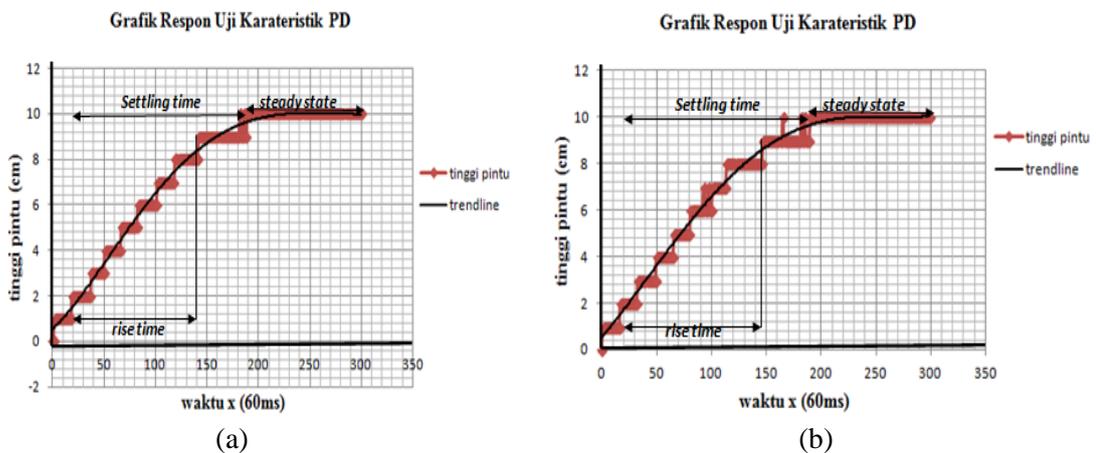
Pada sistem dilakukan pengujian kecepatan motor DC dengan kendali PID. Pengujian dilakukan dengan melihat respon time dari sistem dengan menguji kontrol masing-masing. Kontrol yang diuji pada pengujian ini adalah pengujian kontrol P, PI, PD, PID yang ditampilkan lewat grafik.



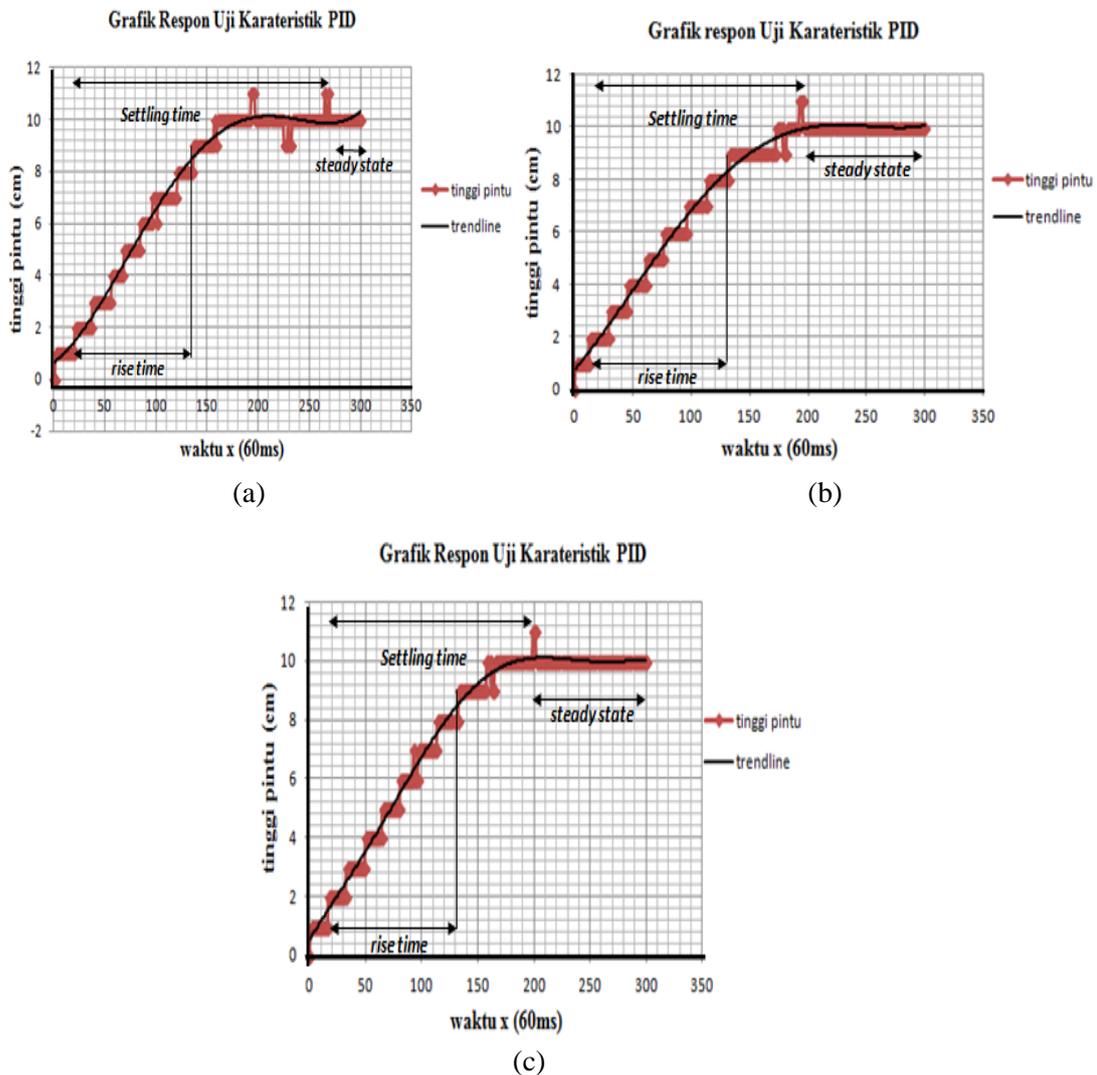
Gambar 6 Grafik perbandingan kontrol P dengan $K_p = 50$ dan $K_p = 100$



Gambar 7 Grafik perbandingan kontrol PI dengan $K_p = 100$ dan $K_i = 0,5$; $K_p = 100$ dan $K_i = 1$



Gambar 8 Grafik perbandingan kontrol PD dengan $K_p = 100$ dan $K_d = 30$; $K_p = 100$ dan $K_d = 50$



Gambar 9 Grafik perbandingan kontrol PID dengan ($K_p = 100$; $K_d = 50$; $K_i = 1$), ($K_p = 100$; $K_d = 50$; $K_i = 0,5$), dan ($K_p = 100$; $K_d = 50$; $K_i = 0,2$)

3.2 Pembahasan

Pada Gambar 6 (a) dan 6 (b) menunjukkan grafik respon uji kontrol P. *Respon time* dari kontrol P dengan $K_p = 50$ adalah *rise time* = 11,1 detik sedangkan *settling time* dan *steady state time* adalah tak hingga. Hal ini disebabkan karena karakteristik dari komponen proporsional yang tetap meninggalkan keadaan *error* tunak. *Respon time* dari kontrol P dengan $K_p = 100$ adalah *rise time* = 7,8 detik, *settling time* = 11,04 detik dan *steady state time* = 11,1 detik. Pada Gambar 7 (a) dan Gambar 7 (b) menunjukkan grafik respon uji kontrol PI. *Respon time* dari kontrol PI dengan $K_p = 100$ dan $K_i = 0,5$ adalah *rise time* = 8,22 detik, *settling time* = 9,78 detik dan *steady state time* = 9,96 detik. *Respon time* dari kontrol PI dengan $K_p = 100$ dan $K_i = 1$ adalah *rise time* = 8,22 detik, *settling time* = 16,78 detik dan *steady state time* = 17,1 detik. Efek K_i sendiri digunakan untuk mengatur kecepatan motor pada pintu air agar *error* yang dihasilkan mendekati 0. Pengaturan K_i yang salah akan mengakibatkan pintu air naik turun. Berdasarkan grafik, semakin besar K_i menunjukkan semakin besar pula osilasi yang dihasilkan. Pada Gambar 8 (a) dan 8 (b) menunjukkan grafik respon uji kontrol PD. *Respon time* dari kontrol PD dengan $K_p = 100$ dan $K_d = 30$ adalah *rise time* = 8,46 detik, *settling time* = 11,22 detik dan *steady state time* = 11,34 detik. *Respon time* dari kontrol PD

dengan $K_p = 100$ dan $K_d = 50$ adalah *rise time* = 8,22 detik, *settling time* = 11,22 detik dan *steady state time* = 11,4 detik. Pengaruh konstanta *derivative* adalah untuk menurunkan *overshoot* terjadi pada sistem. Atau dengan kata lain fungsi dari konstanta *derivative* adalah untuk meredam *overshoot* yang terjadi akibat dari kenaikan konstanta proporsional. Pada Gambar 9 (a), 9 (b), dan 9 (c) menunjukkan grafik respon uji kontrol PID. *Respon time* dari kontrol PID dengan $K_p = 100$; $K_d = 50$; $K_i = 1$ adalah *rise time* = 8,22 detik, *settling time* = 16,02 detik dan *steady state time* = 16,2 detik. *Respon time* dari kontrol PID dengan $K_p = 100$; $K_d = 50$; $K_i = 0,5$ adalah *rise time* = 8,22 detik, *settling time* = 12 detik dan *steady state time* = 9,36 detik. *Respon time* dari kontrol PID dengan $K_p = 100$; $K_d = 50$; $K_i = 0,2$ adalah *rise time* = 7,98 detik, *settling time* = 11,64 detik dan *steady state time* = 11,76 detik. Berdasarkan pengujian jenis pengontrol diatas, dapat dilihat bahwa jenis pengontrol yang paling efektif untuk digunakan untuk proses stabilisasi sistem adalah jenis pengontrol PID. Pada uji coba pengontrol jenis PID sudah didapatkan hasil yang memuaskan terutama pada pengaturan konstanta $K_P = 100$; $K_I = 0,2$; dan $K_D = 50$ didapatkan data yang memiliki respon time paling cepat.

4. KESIMPULAN

1. Telah dibuat suatu purwarupa sistem peringatan dini berbasis SMS menggunakan Arduino Uno ATmega dan bekerja dengan baik.
2. Purwarupa menggunakan motor dc untuk menggerakkan pintu air yang berfungsi untuk mengendalikan ketinggian air.
3. Penggunaan sensor SRF04 pada sistem bekerja dengan baik dan akurat dalam mendeteksi ketinggian air.
4. Tingkat akurasi sensor SRF04(1) adalah 99,86% dan sensor SRF(2) 99,74%
5. Pengiriman SMS menggunakan modul GSM dengan instruksi AT-comands.
6. Pengiriman SMS dilakukan dengan jeda 1sms/40s, 1sms/30s, 1sms/20s, 1sms/10s berdasarkan setpoint ketinggian air.
7. Pengontrol PID dapat digunakan sebagai metode pengendalian kecepatan motor dc yang berfungsi untuk menggerakkan pintu air.
8. Konstanta penguat yang paling stabil dari sistem adalah $K_p = 100$, $K_i = 0,2$, $K_d = 50$.
9. Ketika diset setpoint 10cm, *Respon time* dari konstanta PID adalah *rise time* = 7,98 detik, *settling time* = 11,64 detik, *steady state time* = 11,76 detik.

5. SARAN

1. Perlu pengembangan lebih lanjut terutama pada antarmuka sistem yaitu sebuah tampilan untuk menampilkan hasil ketinggian air secara realtime.
2. Perlu pengembangan lebih lanjut untuk penyampaian hasil ketinggian air menggunakan sosial media seperti facebook, twitter dan sosial media lainnya.

DAFTAR PUSTAKA

- [1] Pahlawan, R. 2010. *Pemanfaatan Jaringan GSM Untuk Penyampaian Informasi Pintu Air ke Pusat Kendali*, Laporan Tugas Akhir, Fakultas Teknik, Universitas Gunadharma, Depok.
- [2] Ardiyanto, I. 2008. *Penerapan Pengendali Proportional Integral Derivative Pada Robot Wall Follower Berbasis ATMEGA16 Dengan Menggunakan Sensor Ultrasonik*. Tugas Akhir S1, Jurusan Teknik Elektro, Fakultas Teknik, UGM, Yogyakarta.
- [3] Korkmaz, S. 2007. *PID Controllers*. Unpublished Journal. Ankara.
- [4] Djuandi, F. 2011. Pengenalan Arduino, www.tokobuku.com, diakses tanggal 18 Februari 2013).