

Aplikasi Scada System pada Miniatur Water Level Control

Antoni Susiono, Handy Wicaksono, Hany Ferdinando

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra Surabaya
e-mail: {handy,hanyf}@petra.ac.id

Abstrak

Makalah ini menjelaskan pembuatan Miniatur *Water Level Control* yang digunakan sebagai salah satu bentuk simulasi dari proses yang terjadi dalam dunia industri. *Project* ini juga dilengkapi dengan SCADA sistem, yang akan memberikan gambaran lebih nyata tentang proses yang dimaksud. Untuk lebih memudahkan proses kontrol dan *monitoring*, maka bentuk dan ukurannya-pun dibuat dalam model miniatur yang mudah dibawa (*portable*). Miniatur *water level control* dibuat sebagai *plant* yang dikontrol oleh PLC. PLC yang digunakan ada 2 macam yaitu: OMRON CPM1 dan MODICON TSX Micro 3721. Sedangkan komputer dalam hal ini SCADA *software* akan memvisualisasikan proses yang terjadi pada *plant*. Dari pengujian yang dilakukan, sistem dapat berjalan sesuai dengan yang diinginkan yaitu mempertahankan level ketinggian air. Di mana pencapaian kestabilan terjadi apabila nilai sensor sama dengan *set_point*. Untuk menjalankan sistem digunakan metode kontrol PID dengan *bias* ($teg_pompa = bias + error$).

Kata kunci: miniatur *water level control*, PLC, SCADA software, kontroler PID.

Abstract

The paper discuss the implementation of Miniature Water Level Control which is made as a simulation from the industrial process. These project is equipped with SCADA to give a real visualisation of industrial process. To make the controlling and monitoring process easier, the project is made in small size (miniature model) which is easy to be brought (portable). Miniature water level control was made as a plant which is controlled by PLC. There are two PLCs which is used, i.e. OMRON CPM1 and MODICON TSX Micro 3721. The computer, in this case the SCADA software, will visualize the process which is happened on the plant. From the experiment which is done, the plant can give the good response and keep the water level on the desired height. The system is stable when the height of water is the same as the set point. PID Control bias method ($pump_voltage = bias + error$) is used to run these system.

Keywords: miniatur *water level control*, PLC, SCADA software, PID controller.

Pendahuluan

Dunia industri terus berkembang dengan sistem-sistem yang baru dalam bidang manufaktur, khususnya sistem kontrol. Saat ini banyak sekali ditawarkan suatu metode kontrol yang efektif dan mudah untuk diimplementasikan.

Perkembangan yang semakin pesat ini, menuntut seorang mahasiswa untuk dapat lebih mengenal bidang tersebut. Untuk itu akan sangat bermanfaat sekali, apabila dalam suatu kegiatan akademik ditunjukkan dan diberikan simulasi dari *module* yang menyerupai sistem yang ada dalam dunia industri. Karena akan banyak sekali dijumpai suatu sistem dan peralatan dalam bidang kontrol, yang tidak sesederhana gambaran yang diberikan dalam perkuliahan.

Water Level Control adalah satu dari sekian banyak sistem yang ada dalam dunia industri. Di samping sederhana, sistem tersebut banyak sekali digunakan dalam dunia industri. Misalkan saja dalam industri kimia.

Dengan dukungan SCADA sistem (proses industri bisa diawasi dan dikendalikan dari jauh, sehingga bisa menghemat biaya, waktu dan tenaga), akan semakin memberikan gambaran tentang kondisi sebenarnya yang ada dalam dunia industri.

Pada makalah ini akan dibahas tentang perancangan miniatur *water level control* mulai dari desain mekanik, hardware, dan software, sampai dengan pengujian menggunakan kontroler PID dengan *bias*.

SCADA System

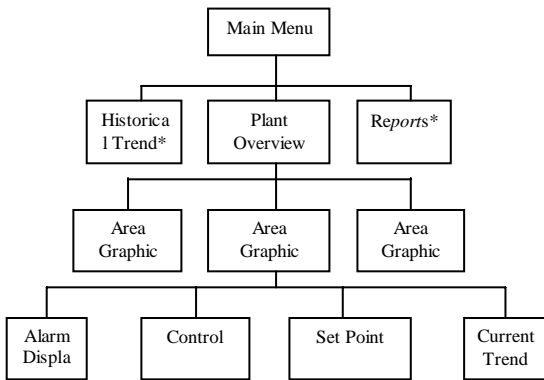
SCADA (*Supervisory Control and Data Acquisition*) sistem, adalah sistem yang memungkinkan pengguna/operator untuk melakukan:

Catatan: Diskusi untuk makalah ini diterima sebelum tanggal 1 Juni 2006. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 6, nomor 2, September 2006.

1. *Monitoring* (pengawasan)
2. *Controlling* (pengendalian)
3. *Data Acquisition* (pengambilan dan perekaman data)

Ketiga fungsi di atas dapat dipenuhi dengan mewujudkannya dalam bentuk *hardware* maupun *software*. Salah satu *software* SCADA yang paling banyak digunakan di dunia ialah *Wonderware® InTouch™* yang berfungsi sebagai *Man Machine Interface* (MMI). Istilah MMI muncul untuk menjembatani jurang antara manusia (operator) dengan mesin (*Plant*), sehingga operator dapat mengawasi dan mengendalikan *Plant* dengan mudah.

Untuk mewujudkan suatu MMI (*display* untuk SCADA) yang baik, maka diperlukan batasan/standard dalam pembuatannya. Berikut ini hirarki dari display grafis suatu MMI:



Gambar 1. Hierarki MMI [1]

PLC (*Programmable Logic Controller*)

PLC yang digunakan ada 2 macam yaitu OMRON CPM1 yang dalam pemrogramannya menggunakan *software* SYSWIN 3.0 dengan struktur bahasa *Ladder Logic* dan PLC MODICON TSX Micro 3721 V.2 dibantu dengan *PL7 Junior Software* dengan struktur bahasa yang digunakan adalah *Structure Text*.

- **OMRON CPM1-20CDR-A**
Tegangan supply yang dibutuhkan adalah 100 – 240 VAC, 50/60Hz. Jumlah I/O OMRON CPM1-20CDR-A adalah 20 pin, 12 pin *input* dan 8 pin *output*. *Range* tegangan operasional OMRON CPM1-20CDR-A adalah 85 – 264V untuk tegangan AC dan 20.4 – 26.4V untuk tegangan DC. [2]
- **MODICON TSX Micro 3721 V.2.0**
PLC tersebut memiliki 10 rak yang bisa digunakan untuk *input/output module*. Dari 10

rak tersebut 6 diantaranya sebagai *built-in slot*, dan 4 lainnya sebagai *expansion slot*. Semua rak yang disediakan bisa dipasangkan dengan *I/O module discrete* maupun analog, Dengan sistem komunikasi yang digunakan adalah UNITELWAY. Dan untuk melakukan *programming* dilakukan dengan bantuan *PL7 software* yang merupakan *software* bawaan dari Schneider. [3] Di mana pada kesempatan ini digunakan TSX DMZ 28DR *discrete module*, yang mana *module* tersebut terdiri dari 16 *pin input module* (dengan karakteristik *sinking* dan *sourcing* yang bisa diatur sesuai kebutuhan) dan 12 *pin output module* (dengan karakteristik *output relay*)

Perencanaan Sistem

Pada perencanaan sistem tersebut ada tiga bagian yang perlu mendapat perhatian yaitu:

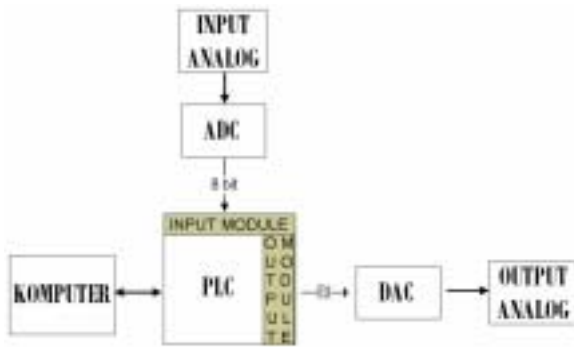
1. miniature water level control yang difungsikan sebagai plant
2. PLC yang merupakan “otak” dari system
3. SCADA *software* (*Wonderware® InTouch™*) yang akan memvisualisasikan proses yang terjadi pada plant

Gambar 2. Cara Kerja Sistem

Secara lengkap dapat dilihat pada gambar 3, di mana Data *input* diberikan oleh sensor *level* ketinggian yang menggunakan metode *float to resistance*. *Input* diterima oleh PLC dalam bentuk 8 bit data *digital*.

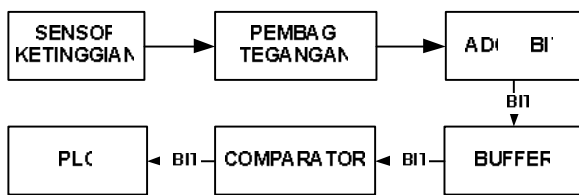
Dalam prosesnya PLC akan membandingkan data dari komputer (ketinggian yang diinginkan) dengan *input*. Di samping itu PLC juga mengirimkan data (*level* ketinggian air) sesuai data *input*-nya. Setelah proses selesai (1 *scan time*), PLC mengeluarkan *output* berupa 8 bit data. Ke-8 bit data tersebut akan diolah untuk mengatur kerja dari pompa air yang akan memompa air ke tandon.

Sebagai catatan sistem dijalankan dengan kondisi kran pada plant terbuka penuh. Jadi program yang dibuat telah dikondisikan khusus untuk pengaturan ketinggian air pada ketinggian tertentu dengan kondisi kran yang terbuka penuh.



Gambar 3. Blok Diagram Kerja Sistem

Perencanaan Input Sistem



Gambar 4. Desain Input Sistem

Sensor *level* ketinggian air akan memberi data berupa perubahan resistansi, yang kemudian diubah menjadi tegangan menggunakan prinsip pembagi tegangan. Data *analog* dari pembagi tegangan diubah menjadi data *digital* 8 bit dengan ADC. Data 8 bit dari ADC diproses kembali dengan menggunakan *buffer*. Supaya PLC bisa menerima data dari ADC, maka dibuatlah penyalaras tegangan yang menggunakan prinsip *comparator*.

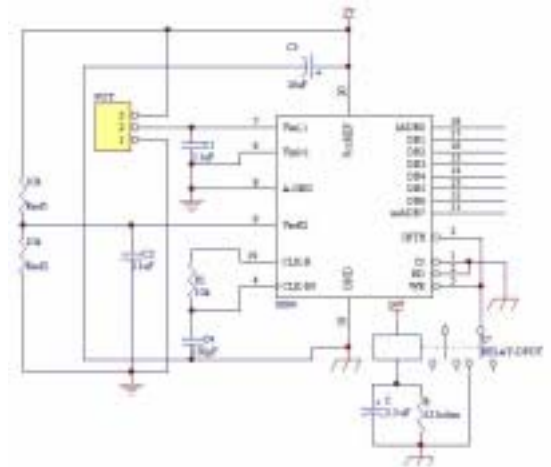
• **Sensor Level Ketinggian dan Pembagi Tegangan**



Gambar 5. Pembagi Tegangan

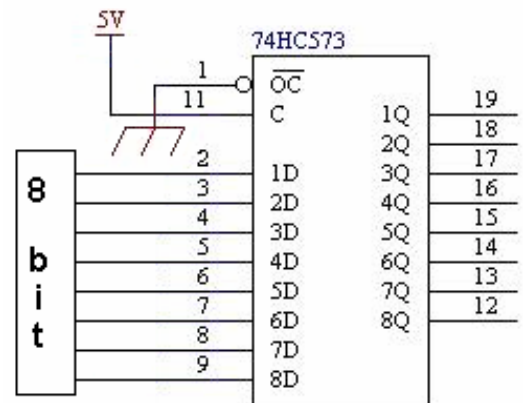
Sensor level ketinggian memiliki prinsip seperti potensiometer, penjelasan ada pada bab berikutnya. Sensor ini perlu dihubungkan dengan pembagi tegangan untuk mendapatkan level tegangan yang sesuai.

• **ADC dan Buffer**



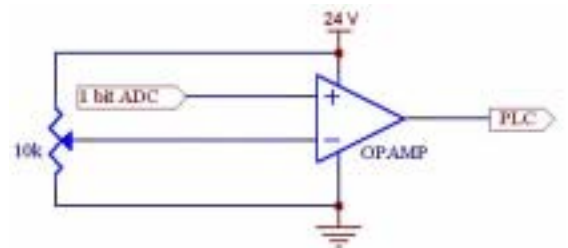
Gambar 6. Rangkaian ADC [4]

ADC 0804 Difungsikan sebagai pengubah data analog dari sensor level ketinggian menjadi data digital 8 bit yang bisa di baca PLC. Output dari ADC dihubungkan ke 74HCT573 yang difungsikan sebagai buffer.



Gambar 7. Desain Buffer ADC

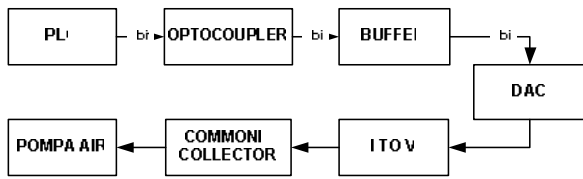
• **Penyalaras Tegangan**



Gambar 8. Comparator

Penyalaras tegangan adalah rangkaian pengkondisi yang menggunakan prinsip dari *comparator*, di mana rangkaian tersebut digunakan untuk mensinkronkan tegangan ADC “5 Vdc” dengan tegangan PLC “24 Vdc”.

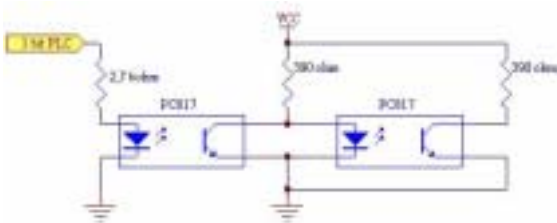
Perencanaan *Output* Sistem



Gambar 9. Desain *Output* Sistem

Output PLC dikondisikan dengan *optocoupler* dan *buffer* supaya bekerja pada kondisi yang sesuai dengan DAC. DAC menghasilkan arus yang akan diubah menjadi tegangan menggunakan prinsip *I to V* (konverter arus ke tegangan). Untuk men-*drive* pompa tegangan dihubungkan dengan transistor (*common collector*).

- **Optocoupler**

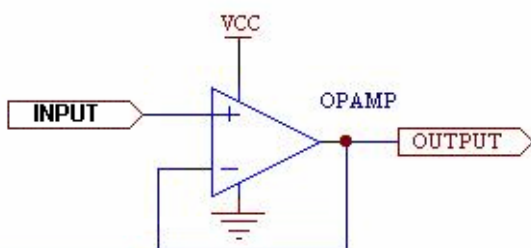


Gambar 10. Desain *Optocoupler* [5]

Rangkaian tersebut digunakan sebagai penghasil logika “0” (0 volt) dan “1” (5 volt) di mana PLC memiliki *output relay* dengan karakteristik *sourcing* 24 volt. Dengan demikian jika *output* PLC memberi logika “1” maka akan dihasilkan tegangan 24 volt sedangkan untuk logika “0” PLC tidak mengeluarkan tegangan.

Dengan rangkaian di atas maka akan didapatkan logika “1” 5 volt jika PLC memberikan logika 1, dan jika PLC memberi logika “0” maka akan didapatkan logika “0” 5 volt. Kondisi tersebut yang nantinya mensinkronkan sistem karena DAC yang akan digunakan bekerja pada level tegangan 5 volt.

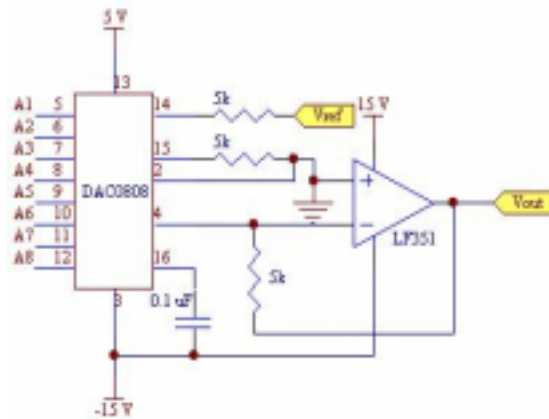
- **Buffer**



Gambar 11. Desain *Buffer Optocoupler*

Rangkaian *buffer* di atas digunakan untuk menjaga supaya tidak terjadi *drop* tegangan dari *output* rangkaian *optocoupler* sebelum masuk ke rangkaian DAC.

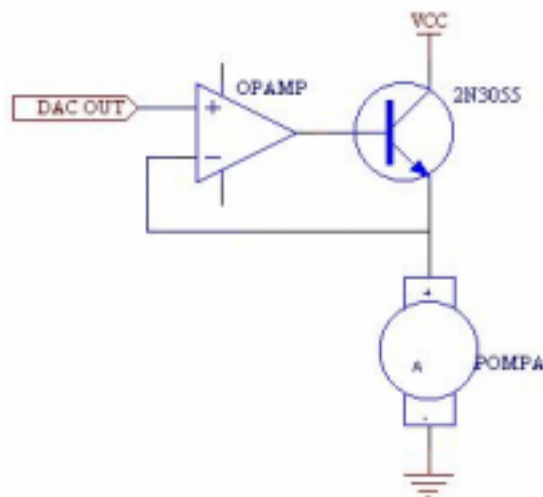
- **DAC dan *I to V***



Gambar 12. Desain Rangkaian DAC [6]

DAC 0808 difungsikan sebagai pengubah data digital dari PLC 8 bit menjadi data analog yang digunakan untuk menggerakkan pompa. Sebuah OpAmp LF351 berfungsi untuk mengubah arus yang dihasilkan menjadi tegangan.

- **Common Collector dan Pompa Air**



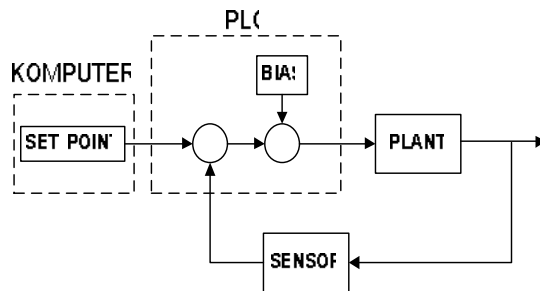
Gambar 13. *Driver* Pompa Air [7]

Dengan rangkaian tersebut maka pompa air akan mendapat tegangan sesuai dengan output dari DAC. Selain itu rangkaian tersebut juga digunakan untuk mengatasi kekurangan arus dari output DAC untuk menggerakkan pompa air.

Perencanaan Software

Pada bagian ini ada 2 perencanaan yaitu: desain *software* untuk PLC OMRON CPM1 lengkap dengan SCADA *system*-nya. Dan desain *software* untuk PLC MODICON TSX Micro 3721 V.2.

- **Desain software pada PLC OMRON CPM1**



Gambar 14. Diagram Sistem Pada PLC OMRON CPM1

PLC menunggu *input*-an dari sensor dan *set_point*, kemudian membandingkan kedua variabel tersebut. Jika $set_point \geq sensor$, maka $error = set_point - sensor$ dan $teg_pompa = bias + error$. Sedangkan jika $set_point < sensor$, maka $error = sensor - set_point$ dan $teg_pompa = bias - error$.

- **Desain SCADA System Pada PLC OMRON CPM1**

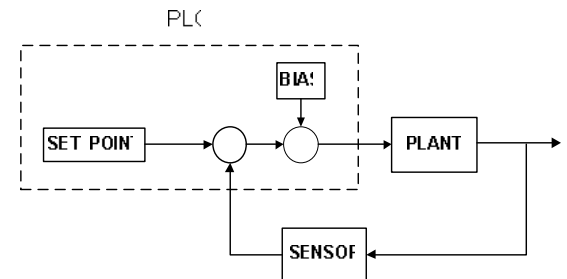


Gambar 15. Blok Diagram SCADA System Pada PLC OMRON CPM1

Untuk desain SCADA-nya, ada 2 *mode* yaitu: *mode control* dan *mode monitoring*. Untuk *mode monitoring user* hanya bisa melihat simulasi *plant* dan simulasi *trend* serta menentukan *set_point*. Sedangkan untuk *mode control*, *user* bisa melakukan fungsi pada *mode monitoring*

serta bisa menjalankan fungsi *manual control* berupa *speed* dan *emergency control*.

- **Desain software pada PLC MODICON TSX Micro 3721**



Gambar 16. Diagram Sistem Pada PLC MODICON TSX Micro 3721

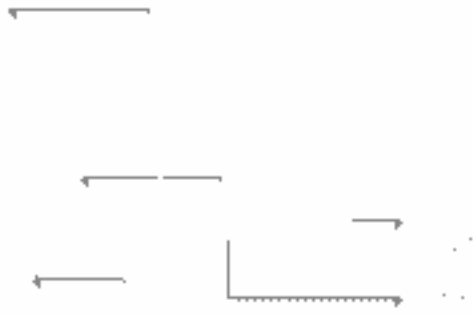
Desain pada PLC MODICON, tidak dilengkapi dengan SCADA *system* karena SCADA *software* yang ada (*Wonderware InTouch*) tidak mendukung layanan komunikasi dengan PLC MODICON TSX Micro 3721. Sehingga semua proses dilakukan dan dikerjakan oleh PLC.

Secara umum desain *software* pada kedua PLC yang digunakan adalah sama. Di mana pada desain *system*-nya dibagi menjadi 4 bagian seperti gambar 16.

Gambar 17. Desain software Pada PLC OMRON CPM1 Dan MODICON TSX Micro 3721

1. *Main* yang berisi program untuk mengontrol *input* dan *output* serta pengaturan *subroutine*.
2. Inisialisasi berfungsi untuk memberikan nilai awal pada semua *variable* yang digunakan.
3. *Soft_start* digunakan supaya pompa bekerja mulai dari kecepatan paling rendah. Hal ini dilakukan untuk menghindari terjadinya lonjakan arus yang besar saat pompa bekerja.
4. *Control* posisi adalah program untuk mempertahankan ketinggian berdasarkan *set_point* dan sensor.

Desain Plant



Gambar 18. Miniatur Water Level Control

• Desain Tandon dan Sumber

1. Tandon
 - Panjang = 15 cm
 - Lebar = 10 cm
 - Tinggi = 20 cm
2. Sumber
 - Panjang = 30 cm
 - Lebar = 20 cm
 - Tinggi = 12,5 cm

• Desain Sensor Level Ketinggian

Sensor level ketinggian di desain dengan menggunakan metode *float to resistance* yang memanfaatkan prinsip dari hukum Archimedes.

Sensor level ketinggian terdiri dari:

- a. beban 1 (W_1)
- b. beban 2 (W_2)
- c. katrol
- d. potensiometer sebagai sensor posisi



Gambar 19. Sensor Level Ketinggian

Desain sensor tersebut dibuat sedemikian rupa supaya benda 1 tetap terapung ($F_A + W_2 > W_1$). Namun dengan tetap memperhatikan posisi benda 1 pada saat naik dan turun.

Dengan prinsip hukum Archimedes didapatkan nilai dari F_A yang kemudian digunakan sebagai acuan untuk menentukan massa dari benda 1 dan benda 2.

$$F_A = V_f \cdot \rho_f \cdot g$$

$$\text{Dengan } V_f = V_1 = \pi r^2 T = 3,14(42,510^3)^2 \cdot 6.10^3$$

$$= 3,4.10^5 \text{ m}^2$$

$$\rho_f = 1000 \text{ kg/m}^3$$

$$g = 10 \text{ m/s}^2$$

$$F_A = 3,4.10^5 \cdot 1000 \cdot 10$$

$$F_A = 0,34 \text{ N}$$

$$\text{Jadi } W_1 - W_2 < F_A$$

$$W_1 - W_2 < 0,34 \text{ N}$$

Sehingga dipilih

$$W_1 = 55,3 \text{ g (acrylic)}$$

$$W_2 = 36,2 \text{ g (nilon)}$$

Pengujian Sensor

Pada pengujian ini digunakan nilai potensiometer sebagai acuan dari pergerakan sensor pada tiap ketinggian.

Pada saat pengisian tandon didapatkan *respon* seperti pada gambar 20.

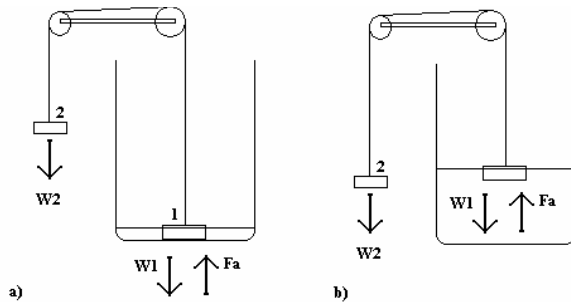


Gambar 20. Grafik *Respon* Potensiometer Terhadap Ketinggian Air Pada Saat Pengisian Tandon

Dari gambar dapat dilihat bahwa pada *mode* pengisian tandon saat ketinggian air 0–1 cm, hampir tidak ada *respon* baik dari potensiometer. Hal ini disebabkan karena pada saat pengisian pertama kali, pelampung (1) tidak bergerak karena untuk mendapat gaya ke-atas posisi pelampung harus mengapung.

Jadi ada jarak yang dibutuhkan oleh pelampung untuk mendapat gaya ke-atas yaitu setebal pelam-

pung itu sendiri. Gerakkan tersebutlah yang membuat sensor tidak memberi *respon*. Untuk lebih jelas bisa dilihat pada gambar 21.



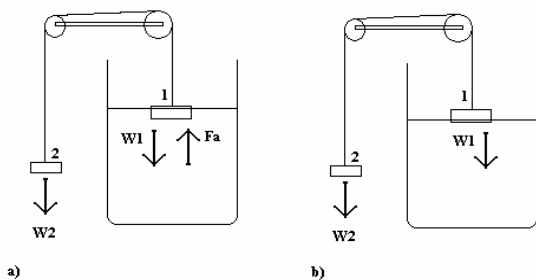
Gambar 21. *Respon* Pelampung Saat Pengisian Tandon

Sedangkan pada saat pengosongan tandon, *respon* yang dihasilkan sedikit berbeda yaitu kesalahan pembacaan terjadi pada ketinggian 11 – 11,5.



Gambar 22. Grafik *Respon* Potensiometer Terhadap Ketinggian Air Pada Saat Pengosongan Tandon

Pada saat pengosongan pertama kali, pelampung (1) tidak bergerak karena untuk bisa membuat posisinya turun, pelampung harus melepas gaya ke-atas yang didapat dari tekanan air. Hal ini dimungkinkan karena berat pelampung sedikit lebih besar daripada berat beban (2). Untuk melepas gaya ke-atasnya, posisi pelampung harus berada di atas air seperti pada gambar 23.

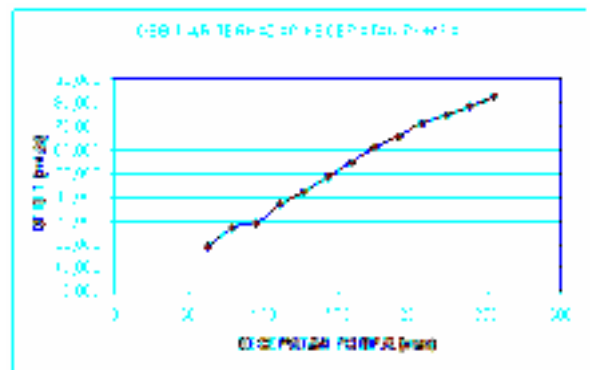


Gambar 23. *Respon* Pelampung Saat Pengosongan Tandon

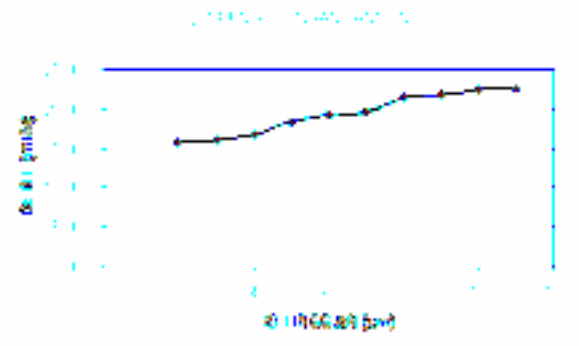
Jadi ada jarak yang dibutuhkan oleh pelampung untuk mendapat gaya ke-atas yaitu setebal pelampung itu sendiri. Gerakkan tersebutlah yang membuat sensor tidak memberi *respon*. Setelah ada pada posisi mengapung, barulah sensor memberikan data yang linier.

Desain Bias

Desain *bias* ini didapatkan dari beberapa pengujian mulai dari pengujian *debit* air yang masuk dan keluar, lewat perhitungan rumusan serta menggunakan metode *trial and error*.



Gambar 24. Debit Air Terhadap Kecepatan Pompa



Gambar 25. Debit Pengosongan Tandon

Berdasarkan hasil pengujian terhadap debit air yang masuk (gambar 24) dan debit air yang keluar (gambar 25), didapatkan rumusan sebagai berikut (menggunakan regresi linier):

- Debit air yang masuk
 $= 0,33 \cdot \text{kecepatan pompa} - 0,765$ (i)
- Debit air yang keluar
 $= 0,87 \cdot \text{ketinggian} + 13,73$ (ii)

Dari kedua rumusan tersebut, jika ditarik suatu hubungan antara debit air yang masuk dengan debit air yang keluar pada tiap ketinggian maka didapatkan rumusan:

$$\text{Kecepatan pompa} = \frac{0,87 \cdot \text{ketinggian} + 14,495}{0,33} \quad (\text{iii})$$

Nilai dari kecepatan pompa inilah yang digunakan sebagai *bias*. Berdasarkan hasil pengujian (pengujian desain perhitungan *bias*), maka bisa ditarik kesimpulan bahwa desain *bias* lewat perhitungan di atas, dirasa kurang tepat, karena debit air yang keluar tidak konstan, sedangkan debit air yang masuk selalu konstan (lebih lengkap lihat pengujian desain perhitungan *bias*).

Selain rumusan di atas ada juga rumusan baru yang didapat dari rumus debit (debit = luas penampang. kecepatan aliran fluida) yaitu

$$\text{Debit (Q)} = \text{luas penampang (A)} \cdot \text{kecepatan aliran (v)}$$

$$A = \frac{1}{4} \cdot \pi \cdot d^2$$

$$v = \sqrt{2 \cdot g \cdot \text{ketinggian}}$$

$$g = 9,8 \text{ m/s}^2$$

$$= \sqrt{2 \cdot 9,8 \cdot \text{ketinggian}}$$

$$\text{Debit (Q)} = \frac{1}{4} \cdot \pi \cdot d^2 \cdot \sqrt{2 \cdot 9,8 \cdot \text{ketinggian}} \quad (\text{iv})$$

Rumusan diatas kemudian diterapkan pada tiap ketinggian dan didapatkan regresi linier:

$$\text{Debit air yang keluar} = 0,87 \cdot \text{ketinggian} + 13,73 \quad (\text{v})$$

Dari persamaan (i) dan (v) maka bisa didapatkan hubungan antara debit air yang masuk dan ketinggian seperti persamaan (vi)

$$\text{Kecepatan pompa} = \frac{0,28 \cdot \text{ketinggian (mm)} + 16,085}{0,34} \quad (\text{vi})$$

Berdasarkan rumusan tersebut kemudian dilakukan pengujian sistem dengan hasil sebagai berikut:

Tabel 1. *Bias* dan *set_point* (1)

KETINGGIAN (cm)	SET_POINT (dec)	BIAS (dec)	KESTABILAN (cm)	ERROR (cm)
2	24	64	2	0
3	49	72	3.1	0.1
4	74	80	4.1	0.1
5	99	88	5.2	0.2
6	124	97	6.3	0.3
7	149	105	7.5	0.5
8	174	113	8.7	0.7
9	199	121	9.8	0.8
10	224	130	10.9	0.9
11	249	138	12.5	1.5

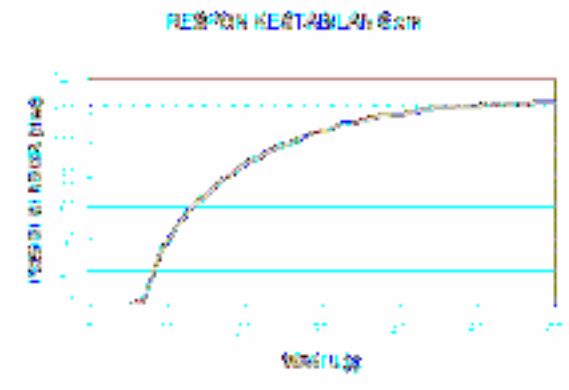
Dari tabel 1 dapat dilihat bahwa dengan menggunakan rumusan *bias* (iv) didapatkan error untuk tiap ketinggian. Oleh karena itu untuk menentukan nilai *bias* dilakukan dengan *trial and error*.

Tabel 2. *Bias* dan *Set_point* (2)

KETINGGIAN	BIAS		SET_POINT	
	HEX	DEC	HEX	DEC
2	41	65	18	24
3	46	70	31	49
4	4B	75	4A	74
5	50	80	63	99
6	55	85	7C	124
7	5A	90	95	149
8	5F	95	AE	174
9	64	100	C7	199
10	69	105	E0	224
11	6E	110	F9	249

Pengujian Kestabilan Sistem

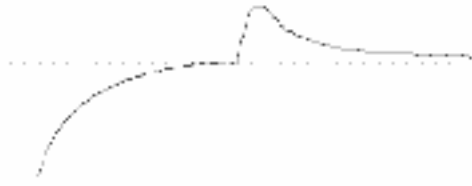
Pengujian kestabilan sistem dilakukan dengan menggunakan *bias* berdasarkan hasil pengujian *trial and error*. Di bawah ini adalah salah satu hasil pengujian sistem pada ketinggian 6 cm.



Gambar 26. Respon Kestabilan 6 cm

Dari gambar 26 dapat dilihat bahwa sistem dapat bekerja dengan baik, terbukti dengan kestabilan sistem yang tercapai pada saat sensor memberikan nilai yang sama dengan *set_point*.

Dari gambar 27 dapat dilihat, bahwa sistem akan kembali stabil setelah diberikan gangguan penambahan air. Setelah penambahan air kestabilan sistem tercapai tidak pada saat sensor sama dengan *set_point*. Melainkan ada error yang ditimbulkan, di mana error tersebut adalah setinggi bahan pelampung.



Gambar 27. Respon Kestabilan 6 cm Terhadap Gangguan Penambahan Air 500 ml

Kesimpulan

Berdasarkan perancangan dan pengujian sistem dapat diambil kesimpulan sebagai berikut:

- Tipe dan karakteristik *input* dan *output* pada PLC akan memberi pengaruh yang berbeda pada desain rangkaian pengkondisi sinyal.
- Karakteristik PLC yang membaca data dalam BCD dan Heksadesimal akan sedikit menyulitkan bila dibaca menggunakan SCADA. Karena SCADA *software* membaca dalam Heksadesimal dan menampilkan dalam decimal.
- Pelampung (*float*) dengan ketebalan 6mm akan memberikan *error* pembacaan sensor pada saat pengosongan tandon sebesar 6 mm.
- Metode kontrol yang digunakan adalah metode kontrol *bias* dengan rumusan $Bias = 2,5 \cdot ketinggian - 26$. Rumusan tersebut didapatkan dari *regresi linier* hasil pengujian *bias* pada tiap ketinggian.
- Untuk mempertahankan sistem pada ketinggian tertentu dijalankan dengan rumusan $teg_pompa = bias + error$. Di mana kestabilan sistem dicapai saat $teg_pompa = bias$.
- Pada saat kestabilan sistem tercapai kemudian pada sistem diberikan gangguan penambahan air sebanyak 500 ml, maka sistem akan kembali stabil pada ketinggian yang diinginkan dikurangi tebal dari pelampung.

- Penggunaan PLC dengan bahasa pemrograman *Ladder Diagram* sudah terlalu umum. Ada baiknya untuk mulai dikembangkan penggunaan bahasa pemrograman lain misalnya: *Structure Text* yang berbasis PASCAL, atau BASIC.
- Miniatur *Water Level Control* tersebut bisa digunakan untuk semua tipe PLC dengan syarat *input* dan *output* PLC minimal 8 bit.
- Untuk mengontrol sistem tidak hanya menggunakan metode bias. Bisa juga digunakan metode kontrol dengan PID atau *Fuzzy Logic*.

Daftar Pustaka

- [1]. *Automation & Control Volume3–Programming*. Canada: City of London Environmental Services, November 2002.
- [2]. *A Beginner's guide to PLC OMRON*. Singapore: OMRON, Agustus 1996.
- [3]. *TSX Micro PLCs TSX 3705/3708/3710/3720 Implementation manual - volume 2*. CD-ROM. TSX Micro PLCs TSX 3705/3708/3710/ 3720 Implementation manual. Germany: Schneider, 2002.
- [4]. *ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-bit μ P Compatible A/D Converters Datasheet*. California: National Semiconductor, Oktober 1989.
- [5]. *Sharp Device Specification for Photocoupler PC817*. Japan: Sharp Corporation, 1995. 25 Maret 2005 <<http://www.alldatasheet.com/datasheet-pdf/pdf/77784/AUK/SPC817M.html>>.
- [6]. *DAC0808/DAC0807/DAC0806 8-bit μ P Compatible D/A Converters Datasheet*. California: National Semiconductor, Oktober 1989.
- [7]. Allaboutcircuit, *Common Collector*. 04 Mei 2005. <http://www.allaboutcircuits.com/vol_3/chpt_4/6.html>.