

**PENGONTROLAN NUTRISI PADA SISTEM TOMAT
HIDROPONIK MENGGUNAKAN KONTROLER PID**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



FAIZAL BENI AKBAR

NIM. 125060307111005

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2016

LEMBAR PENGESAHAN

**PENGONTROLAN NUTRISI PADA SISTEM TOMAT
HIDROPONIK MENGGUNAKAN KONTROLER PID**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan

memperoleh gelar Sarjana Teknik



FAIZAL BENI AKBAR

NIM. 125060307111005

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
pada tanggal 12 Agustus 2016

Dosen Pembimbing I

Dosen Pembimbing II

Goegoes Dwi Nusantoro, S.T., M.T.

NIP. 19711013 200604 1 001

Dr. Ir. Erni Yudaningtyas, M.T.

NIP. 19650913 199002 2 001

**Mengetahui,
Ketua Jurusan Teknik Elektro**

M. Azis Muslim, S.T., M.T., Ph.D.

NIP. 19741203 200012 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi ini untuk dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, Agustus 2016

Mahasiswa,

FAIZAL BENI AKBAR

NIM. 125060307111005



RINGKASAN

Faizal Beni Akbar, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Agustus 2016, Pengontrolan Nutrisi pada Sistem Tomat Hidroponik Menggunakan Kontroler PID, Dosen Pembimbing: M. Aziz Muslim dan Purwanto.

Sistem hidroponik merupakan sebuah metode bercocok tanam modern tanpa menggunakan media tanah melainkan air dengan menekankan pemenuhan kebutuhan nutrisi pada tanaman. Sistem *Nutrient Film Technique* (NFT) merupakan sistem yang mendukung media tanam hidroponik dimana aliran air sangat dangkal dan mengandung semua nutrisi terlarut. Pengontrolan nutrisi pada sistem tomat hidroponik diperlukan, karena pertumbuhan tanaman bergantung pada pemberian nutrisi yang ideal. Pemberian nutrisi dapat dilihat dari kadar keasaman (pH) air tomat hidroponik yang idealnya berkisar antara 5.5 hingga 6.5, namun, akan lebih maksimal untuk penyerapan nutrisi larutan makro dan larutan mikro pada pH 6. Sedangkan, air hidroponik biasanya bersifat basa.

Dengan menggunakan kontroler PID, sistem akan menghasilkan *error steady state* yang kecil dan mendapatkan *settling time* yang lebih cepat. Proses perancangan kontroler PID menggunakan metode pertama *Ziegler-Nichols* didapatkan nilai parameter $K_p = 4.26$, $K_i = 0.1065$, dan $K_d = 42.6$. Dari hasil pengujian nutrisi dengan menggunakan kontroler PID didapatkan nilai *settling time* (ts) 520 detik, *peak time* (tp) 312 detik, *rise time* (tr) 157 detik, *delay time* (td) 107 detik, *maximum overshoot* (Mp) 10.67%, *error steady state* 5%, dan *recovery time* sebesar 118 detik.

Kata Kunci: hidroponik, NFT, tomat, PID, *Ziegler-Nichols*.

SUMMARY

Faizal Beni Akbar, *Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, August 2016, Nutrition Controlling on Tomatoes Hydroponic's System with PID Controller, Academic Supervisor: M. Aziz Muslim and Purwanto.*

Hydroponic's system is one of the modern farming method without using soils as a media but waters by emphasizing the fulfillment of the nutrient to the plant. Nutrient Film Technique's system is a system that support the hydroponic's system where the flow of water is very shallow and contains all of the dissolved nutrients. Controlling nutrients in the tomatoes hydroponic's system is necessary, because for plants growth depends on the ideal nutrition. Nutrition can be seen from the tomatoes hydroponic's water acidity (pH) which ideally ranges between 5.5 to 6.5, but, it would be optimal for the nutrient absorption solution of macro and micro solution at range 6. Meanwhile, the water of hydroponic is usually too alkaline.

With using the PID's controller, the system will gain small error steady state and get the fastest settling time. Design process that using the PID's controller with Ziegler-Nichols method obtained the PID's parameter such $K_p = 4.26$, $K_i = 0.1065$, and $K_d = 42.6$. According to the result of the nutrition experiment with using PID's controller, 520 seconds are the settling time (t_s), 312 seconds are the peak time (t_p), 157 seconds are the rise time (t_r), 107 seconds are the delay time (t_d), 10.67% are the value of the Maximum Overshoot (M_p), 5% are the value of the error steady state, and 118 seconds are the recovery time.

Keywords: *hydroponics, NFT, tomato, PID, Ziegler-Nichols.*

PENGANTAR

Bismillahirrohmanirrohim. Alhamdulillahirobbil'alamin, puji syukur terhadap Allah SWT atas berkat, rahmat serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengontrolan Nutrisi pada Sistem Tomat Hidroponik Menggunakan Kontroler PID” dengan baik dan lancar. Shalawat serta salam juga tak lepas untuk penulis panjatkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan yang baik bagi yang mengharapkan rahmat dan hidayah-Nya.

Penulis menyadari bahwa penyusunan skripsi ini tidak akan lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

- Allah SWT yang telah memberikan kelancaran, kemudahan dan hidayah-Nya.
- Keluarga tercinta, kedua orang tua Tjepi Subagdja dan Nurhayati yang selalu memberikan kasih sayang dan doanya yang tiada akhir. Adik tercinta Faizah Putri Rahmadani yang selalu memberikan dukungan.
- Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya sekaligus sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran, dan masukan yang telah diberikan.
- Bapak Hadi Suyono, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ali Mustofa, ST., MT. selaku Ketua Program Studi Strata-1 (S1) Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ir. Purwanto, MT. selaku Ketua Konsentrasi Teknik Kontrol Jurusan Teknik Elektro Universitas Brawijaya sekaligus sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran, dan masukan yang telah diberikan.
- Laboratorium Dasar Elektrik dan Pengukuran Teknik Elektro Universitas Brawijaya atas segala alat serta sarana dan prasarana yang dimanfaatkan penulis dalam melakukan penelitian.

- Laboratorium Sistem Kontrol Teknik Elektro Universitas Brawijaya atas segala alat serta sarana dan prasarana yang dimanfaatkan penulis dalam melakukan penelitian.
- Pranata Laboratorium Dasar Elektrik dan Pengukuran, Pak Dedi dan Keluarga besar asisten Laboratorium Dasar Elektrik dan Pengukuran Fajar, Ferdian, Gadis, Hesa, Zakiyah, Arif, Bobby, Citra, Fitrah, Ina, Milky, Najar, Bima, Dini, Nola, Okto, Rahmat, Taka, dan Topan terima kasih telah memberikan bantuan, dukungan, dan motivasi.
- Pranata Laboratorium Sistem Kontrol, Mbak Eka dan teman-teman asisten Laboratorium Sistem Kontrol Andri, Avif, Dennis, Faris, Hilmy, Indra, Suro, dan Yudha yang telah memberikan banyak dukungan dan bantuan.
- Sahabat Gabriel, Hanif, Hesa, Odi, Suro, Tyo, dan Zakiyah.
- Teman-teman Carina, Dhea, Dimas, Elisa, Fanny, Lia, Racha, Onil, Tio, Razan, dan Versita.
- Keluarga besar Konsentrasi Teknik Kontrol Angkatan 2012 dan Keluarga Besar “Voltage” Angkatan 2012 atas doa, semangat, serta dukungan yang diberikan untuk penulis.
- Semua pihak yang memberikan dukungan dan bantuan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa masih jauh dari sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaannya. Oleh karena itu, penulis mengharapkan untuk kritik dan saran untuk penyempurnaan penulisan di masa mendatang. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Agustus 2016

Penulis

DAFTAR ISI

RINGKASAN	i
SUMMARY	ii
PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Manfaat	3
BAB II TINJAUAN PUSTAKA	5
2.1 Tanaman Tomat	5
2.2 Hidroponik	6
2.2.1 Sistem Hidroponik <i>Nutrient Film Technique</i> (NFT)	8
2.3 Mesin <i>Direct Current</i> (DC)	9
2.3.1 Motor <i>Direct Current</i> (DC)	9
2.3.2 Motor <i>Direct Current</i> (DC) <i>Peristaltic Pump</i> 12V.....	11
2.3.3 <i>Pulse Width Modulation</i>	12
2.4 <i>High Current</i> 30 A DC Motor Driver dengan <i>Half-Bridge</i> Infineon BTS 7960 chip	12
2.5 Mikrokontroler Arduino Mega 2560	13
2.5.1 Daya	14
2.5.2 Memori	15
2.5.3 Input & Output	15
2.5.4 Komunikasi	16
2.6 Sensor pH	16
2.7 Sensor Ultrasonik	18
2.8 Kontroler	19
2.9 Jenis-jenis Kontroler	20
2.9.1 Kontroler Proporsional (P)	20

2.9.2	Kontroler Integral (I)	21
2.9.3	Kontroler Differensial (D)	22
2.9.4	Kontroler Proporsional Integral Differensial (PID)	22
2.10	Ziegler Nichols	23
2.11	Transformasi Z	26
2.12	Diskritisasi	27
2.13	Parameter Performansi Sistem	28
BAB III METODE PENELITIAN		29
3.1	Perancangan Blok Diagram Sistem	30
3.2	Spesifikasi Desain	31
3.3	Karakterisasi Setiap Blok	32
3.3.1	Karakterisasi Motor DC <i>Peristaltic Pump</i> 12V	32
3.3.2	Karakterisasi <i>Driver</i> Motor (<i>High Current 30 A DC Motor Driver</i>)	35
3.3.3	Karakterisasi Motor Pompa Mini DC 12V	37
3.3.4	Karakterisasi Sensor pH SKU SEN0161	40
3.3.5	Karakterisasi Sensor Ultrasonik SRF05	44
3.4	Pembuatan Perangkat Keras	47
3.4.1	Desain Media Hidroponik	47
3.4.2	Prinsip Kerja Alat	48
3.4.3	Rangkaian Catu Daya	49
3.4.4	Konfigurasi Pin Arduino Mega 2560	49
3.4.5	Konfigurasi Pin <i>Driver</i> Motor	50
3.4.6	Konfigurasi Pin <i>Relay</i>	51
3.5	Perancangan Algoritma	52
3.5.1	Penentuan Karakteristik Pengontrolan pH	52
3.5.2	Parameter Kontroler PID dengan Metode <i>Ziegler-Nichols</i>	52
3.5.3	Diskritisasi	55
3.5.4	<i>Flowchart</i> Program	56
BAB IV ANALISIS DAN PEMBAHASAN		57
4.1	Pengujian pH Larutan Nutrisi tanpa Gangguan	57
4.2	Pengujian pH Larutan Nutrisi dengan Gangguan	58
BAB V PENUTUP		59
5.1	Kesimpulan	59
5.2	Saran	59
DAFTAR PUSTAKA		61

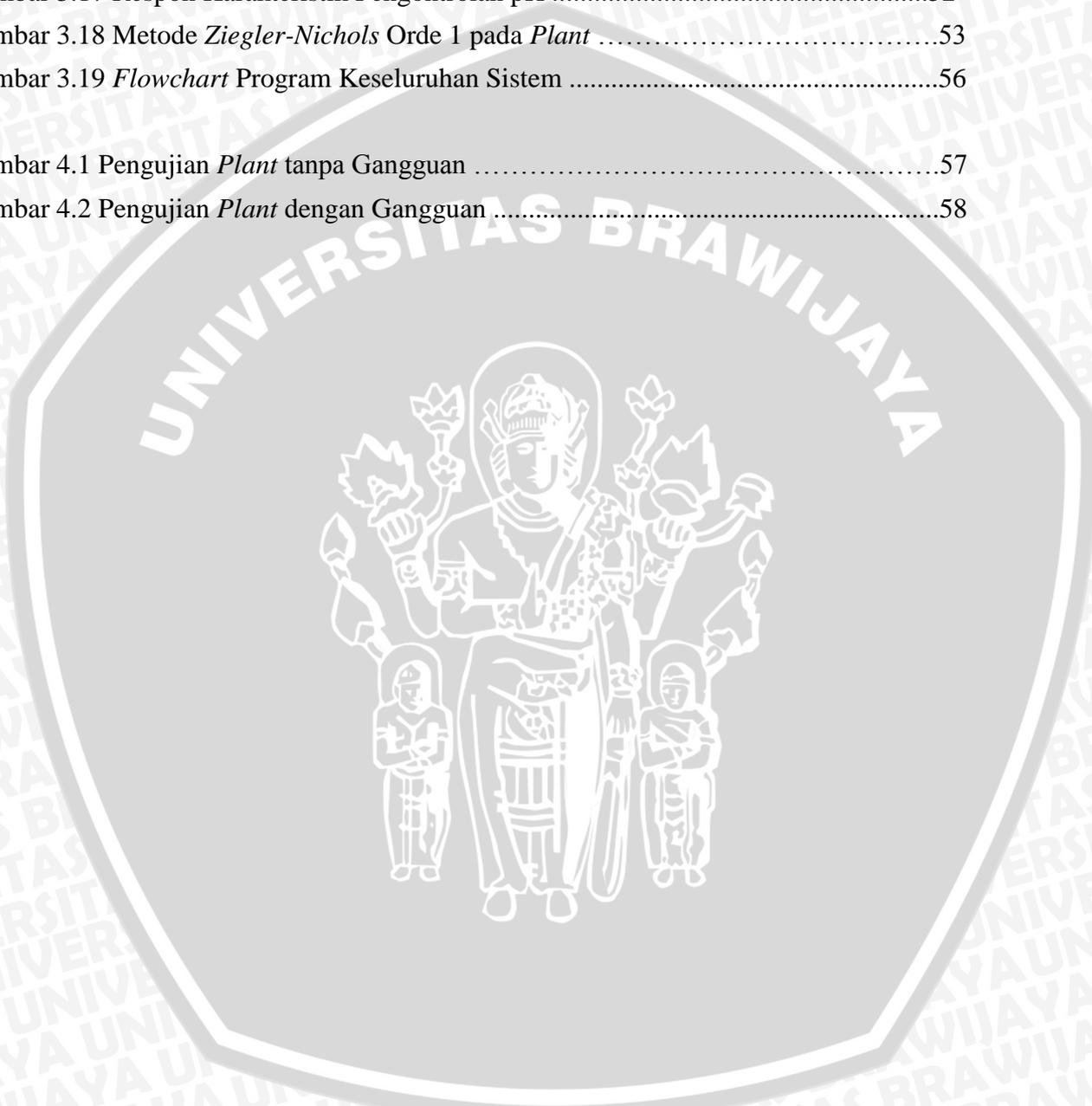
LAMPIRAN I.....	63
LAMPIRAN II.....	67
LAMPIRAN III.....	69



DAFTAR GAMBAR

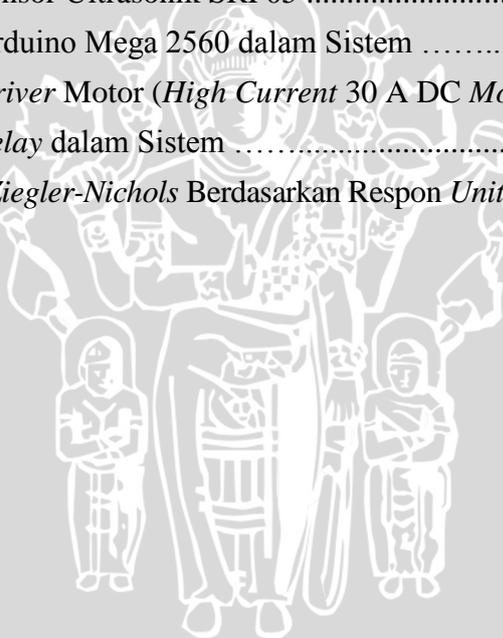
Gambar 2.1 Tanaman Hidroponik	7
Gambar 2.2 <i>Nutrient Film Technique</i> (NFT)	9
Gambar 2.3 <i>Motor Direct Current</i> (DC)	11
Gambar 2.4 Motor DC <i>Peristaltic Pump</i> 12V.....	11
Gambar 2.5 Prinsip Kerja <i>Peristaltic Pump</i>	11
Gambar 2.6 Sinyal PWM	12
Gambar 2.7 <i>High Current 30 A Motor Driver</i>	13
Gambar 2.8 Arduino Mega 2560	14
Gambar 2.9 Sensor pH SKU SEN0161	17
Gambar 2.10 Prinsip Sensor pH	17
Gambar 2.11 Sensor Ultrasonik SRF05	18
Gambar 2.12 Prinsip Kerja Sensor Ultrasonik SRF05	18
Gambar 2.13 Diagram Blok Kontroler Proporsional	20
Gambar 2.14 Diagram Blok Kontroler Proporsional	21
Gambar 2.15 Diagram Blok Kontroler Integral	22
Gambar 2.16 Diagram Blok Kontroler Differensial	22
Gambar 2.17 Diagram Blok Kontroler PID	23
Gambar 2.18 Kurva Respon <i>Unit Step</i> yang Menunjukkan 25% <i>Maximum Overshoot</i>	24
Gambar 2.19 Respon <i>Plant</i> Terhadap Masukan Berupa <i>Unit Step</i>	24
Gambar 2.20 Kurva Respon yang Berbentuk S	24
Gambar 2.21 Sistem <i>Loop</i> Tertutup dengan Kontroler Proporsional	26
Gambar 2.22 Osilasi Berkesinambungan dengan Periode <i>Pcr</i>	26
Gambar 3.1 Blok Diagram Sistem Tangki (I)	30
Gambar 3.2 Blok Diagram Sistem Tangki (II)	31
Gambar 3.3 Blok Pengujian Motor DC <i>Peristaltic Pump</i> 12V	33
Gambar 3.4 Grafik Hasil Pengujian Motor DC <i>Peristaltic Pump</i> 12V.....	34
Gambar 3.5 Blok Pengujian <i>Driver</i> Motor (<i>High Current 30 A DC Motor Driver</i>)	36
Gambar 3.6 Grafik Hasil Pengujian <i>Driver</i> Motor (<i>High Current 30 A DC Motor Driver</i>)	37
Gambar 3.7 Blok Pengujian Motor Pompa Mini DC 12V	38
Gambar 3.8 Grafik Hasil Pengujian Motor Pompa Mini DC 12V	39
Gambar 3.9 Blok Pengujian Sensor pH SKU SEN0161	41
Gambar 3.10 Grafik Hasil Pengujian Sensor pH SKU SEN0161.....	42
Gambar 3.11 Grafik Hasil Pengujian Sensor pH SKU SEN0161 setelah kalibrasi	44

Gambar 3.12 Blok Pengujian Sensor Ultrasonik SRF05	45
Gambar 3.13 Grafik Hasil Pengujian Sensor Ultrasonik SRF05	46
Gambar 3.14 Gambar Desain Media Hidroponik	47
Gambar 3.15 <i>Switching Power Supply</i>	49
Gambar 3.16 Skematik Konfigurasi I/O Arduino Mega 2560	49
Gambar 3.17 Respon Karakteristik Pengontrolan pH	52
Gambar 3.18 Metode <i>Ziegler-Nichols</i> Orde 1 pada <i>Plant</i>	53
Gambar 3.19 <i>Flowchart</i> Program Keseluruhan Sistem	56
Gambar 4.1 Pengujian <i>Plant</i> tanpa Gangguan	57
Gambar 4.2 Pengujian <i>Plant</i> dengan Gangguan	58



DAFTAR TABEL

Tabel 2.1 Kandungan Nutrisi Tomat	5
Tabel 2.2 Daftar Kebutuhan Nutrisi Ideal pada Tanaman	7
Tabel 2.3 Fungsi Pin pada <i>Interface Header High Current 30 A DC Motor Driver</i> .13	
Tabel 2.4 Aturan Penalaan <i>Ziegler-Nichols</i> Berdasarkan Respon <i>Unit Step</i> pada <i>Plant</i>25	
Tabel 2.5 Aturan Dasar <i>Ziegler-Nichols</i> Berdasarkan <i>Critical Gain</i> Kcr dan <i>Critical Period</i> Pcr.....26	
Tabel 3.1 Hasil Pengujian Motor DC <i>Peristaltic Pump 12V</i>	34
Tabel 3.2 Hasil Pengujian <i>Driver Motor (High Current 30 A DC Motor Driver)</i>	36
Tabel 3.3 Hasil Pengujian Motor Pompa Mini DC 12V	39
Tabel 3.4 Hasil Pengujian Sensor pH SKU SEN0161	41
Tabel 3.5 Hasil Pengujian Sensor pH SKU SEN0161 setelah Kalibrasi	43
Tabel 3.6 Hasil Pengujian Sensor Ultrasonik SRF05	46
Tabel 3.7 Konfigurasi Pin Arduino Mega 2560 dalam Sistem	50
Tabel 3.8 Konfigurasi Pin <i>Driver Motor (High Current 30 A DC Motor Driver)</i>	51
Tabel 3.9 Konfigurasi Pin <i>Relay</i> dalam Sistem	51
Tabel 3.10 Aturan Penalaan <i>Ziegler-Nichols</i> Berdasarkan Respon <i>Unit Step</i> pada <i>Plant</i> .54	



BAB I

PENDAHULUAN

1.1 Latar Belakang

Hidroponik merupakan sebuah metode bercocok tanam modern tanpa menggunakan media tanah melainkan air, dengan menekankan pemenuhan kebutuhan nutrisi pada tanaman. Sistem ini sangatlah cocok untuk teknologi bercocok tanam yang dapat diadopsi untuk menghasilkan pangan (Reno Suryani, 2015). Penggunaan air yang lebih efisien dibandingkan dengan menggunakan media tanah, menjadikan metode ini sangat cocok untuk daerah-daerah yang memiliki pasokan air yang terbatas. Jadi, selain untuk mengaplikasikan sistem daur ulang, dengan menggunakan sistem hidroponik juga dapat menghijaukan lingkungan sekitar. Tinggal di perkotaan dengan memiliki luas tanah yang kecil sekalipun tidak menutup kemungkinan untuk dapat menanam beragam tanaman seperti sayuran dan buah-buahan. Tanaman holtikultura seperti tanaman tomat sangat cocok untuk dibudidayakan menggunakan sistem hidroponik.

Tomat merupakan tanaman yang memiliki banyak manfaat bagi kehidupan manusia. Tomat dapat dijadikan sebagai bahan makanan, obat-obatan, dan bahkan dapat dijadikan sebagai bahan kosmetik. Tanaman ini banyak memiliki kandungan vitamin A, B, B2, dan C yang cukup tinggi, juga kaya akan Zat Besi (Fe), Kalsium (Ca), Fosfor (P), protein, karbohidrat, dan lemak (Direktorat Gizi Departemen Kesehatan RI, 1972). Kadar kandungan tomat juga bervariasi tergantung dari nutrisi yang didapat oleh tanaman tomat. Ada banyak mineral yang dibutuhkan oleh tanaman tomat, namun ada 6 yang paling penting dalam proses pertumbuhan tanaman tomat yakni Nitrogen (N), Fosfor (P), Kalium (K), Sulfur (S), Magnesium (Mg), dan Kalsium (Ca). Kombinasi dari beberapa unsur tersebut dapat memenuhi kebutuhan nutrisi tanaman hingga 95% (Sainju, et.al, 2003). Oleh karena itu, dengan melakukan pengontrolan nutrisi yang baik diharapkan dapat memenuhi kebutuhan nutrisi tanaman tomat tersebut.

Sistem hidroponik sangat bergantung pada larutan nutrisi yang diberikan. Pemberian nutrisi yang berlebihan dapat menyebabkan keracunan pada tanaman. Sedangkan, jika pemberian nutrisi terlalu sedikit maka dapat menyebabkan pertumbuhan tanaman menjadi tidak optimal (Novizan, 2002). Inti dari nutrisi yang diberikan terhadap tanaman adalah kadar molaritas, kadar ion, kadar *Electro Conductivity* (EC), dan kadar *potential of hydrogen* (pH) dari setiap unsur hara yang diberikan (Reno Suryani, 2015).

Larutan nutrisi dalam sistem hidroponik dibagi menjadi 2 larutan, yakni larutan unsur Makro (N, P, K, Mg, Ca, S) dan larutan unsur Mikro (B, Cu, Zn, Fe, Mo, Mn, Cl, Na, Co, Si, Ni). Kedua larutan ini memiliki kadar pH tertentu, sehingga dapat menjadi indikator dalam menentukan nutrisi yang ideal pada tanaman tomat hidroponik. Dalam proses pemeliharaan tanaman tomat hidroponik dibutuhkan larutan nutrisi dengan indikator pH ideal berkisar antara 5.5 hingga 6.5 agar pertumbuhan tanaman menjadi berkembang. Pada pH bernilai 6 pertumbuhan akan dapat maksimal karena kisaran ≤ 5.5 larutan akan terlalu asam dan kisaran ≥ 6.5 larutan akan beresiko terlalu basa (Yati Supriyati, 2015).

Oleh karena itu, untuk memecahkan masalah di atas diantaranya adalah dengan meningkatkan kualitas produksi yang baik pada proses pemeliharaan tanaman tomat dengan sistem hidroponik, dibuatlah skripsi ini dengan judul pengontrolan nutrisi pada sistem tomat hidroponik dengan menggunakan kontroler PID. Kontroler merupakan suatu sistem yang dinamis yang sengaja ditambahkan untuk mendapatkan karakteristik sistem keseluruhan yang diinginkan (Ogata K, 2010).

Alat yang digunakan untuk pemberian nutrisi secara otomatis pada sistem tomat hidroponik ini menggunakan kontroler PID berbasis Arduino Mega 2560. Proporsional Integral Differensial (PID) adalah kontroler yang merupakan gabungan dari kontroler proporsional, kontroler integral, dan kontroler differensial. Gabungan dari ketiga kontroler ini diharapkan dapat memberikan keluaran sistem yang stabil dimana masing-masing kontroler bisa saling melengkapi satu dengan yang lainnya. Keuntungan dari kontroler PID adalah sebuah sistem yang sederhana sehingga dapat lebih cepat untuk mencapai keluaran sistem dengan *error* yang kecil. Diharapkan dengan menggunakan kontroler PID kadar larutan nutrisi pada sistem hidroponik tanaman tomat dapat dikendalikan sesuai dengan yang diinginkan.

1.2 Rumusan Masalah

Dari latar belakang yang telah diangkat memunculkan permasalahan yaitu :

1. Bagaimana merancang dan membuat alat pengontrolan nutrisi pada sistem tomat hidroponik?
2. Bagaimana merancang *software* untuk pengontrolan nutrisi dan mendapatkan nilai-nilai parameter kontroler PID agar sesuai dengan yang diinginkan?

1.3 Batasan Masalah

Karena luasnya objek kajian di dalam penelitian ini, maka diperlukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah yang diterapkan pada penelitian ini yaitu :

1. Sistem hidroponik menggunakan sistem aktif dengan kultur air yaitu metode *Nutrient Film Technique* (NFT) dengan desain sendiri.
2. Pembahasan ditekankan pada penggunaan kontroler PID pada sistem pengontrolan nutrisi.
3. Pengontrolan nutrisi yang ideal menggunakan indikator *potential of hydrogen* (pH).
4. Motor *peristaltic pump* DC 12V sebagai aktuator sistem.
5. Kontroler yang digunakan adalah mikrokontroler Arduino Mega 2560.
6. Kinerja rangkaian pengondisi sinyal, *driver* dan elektronika tidak dibahas secara mendetail.
7. Gangguan pada pengontrolan pH berupa penambahan larutan asam sebanyak 100 ml.

1.4 Tujuan

Tujuan dari skripsi ini adalah:

1. Menentukan parameter PID dan merancang algoritma kontroler PID dari *plant* sistem tomat hidroponik.
2. Mengetahui performansi *output plant* sistem tomat hidroponik dengan menggunakan algoritma kontroler PID sebelum dan setelah sistem diberi gangguan.

1.5 Manfaat

Manfaat skripsi ini adalah dapat dipahaminya pengontrolan nutrisi pada sistem tomat hidroponik dengan menggunakan kadar keasaman air atau *potential of hydrogen* (pH) sebagai indikator untuk menentukan pemberian nutrisi yang baik dengan menggunakan metode kontroler Proporsional Integral Differensial (PID).

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan teori-teori yang mendukung skripsi, yang terdiri atas:

2.1 Tanaman Tomat

Tomat merupakan satu dari sekian banyak jenis sayuran yang telah dikenal masyarakat. Tomat memiliki banyak manfaat. Sebagai salah satu komoditas pertanian, tomat memiliki kandungan vitamin dan mineral yang berguna untuk pertumbuhan dan kesehatan manusia. Tomat juga mengandung zat pembangun jaringan tubuh dan zat yang menghasilkan energi untuk bergerak dan berpikir, antara lain karbohidrat, protein, lemak, dan kalori. Kandungan nutrisi yang terdapat pada tomat terdapat pada Tabel 2.1 berikut ini.

Tabel 2.1 Kandungan Nutrisi Tomat (Direktorat Gizi Departemen Kesehatan RI).

No.	Kandungan	Sari Air Tomat	Tomat Muda	Tomat Masak
1.	Kalori (Kal)	15	23	20
2.	Protein (g)	1	2	1
3.	Lemak (g)	0.2	0.7	0.3
4.	Karbohidrat (g)	3.5	2.3	4.2
5.	Vitamin A (SI)	600	320	1500
6.	Vitamin B (mg)	0.5	0.07	0.6
7.	Vitamin C (mg)	10	30	40
8.	Kalsium (mg)	7	5	5
9.	Fosfor (mg)	15	27	26
10.	Besi (mg)	0.4	0.5	0.5
11.	Air (g)	94	93	94

Sebagai sumber vitamin, tomat kaya akan vitamin C yang berguna untuk meningkatkan kekebalan tubuh dari serangan berbagai macam penyakit. Tomat juga kaya akan vitamin A yang berguna untuk mencegah dan mengobati penyakit pada mata. Sebagai sumber mineral, tomat mengandung zat besi (Fe) yang berguna untuk pembentukan sel darah merah. Tomat juga mempunyai serat yang berguna bagi pencernaan dan mengandung potasium yang bermanfaat untuk menurunkan tekanan darah tinggi.

2.2 Hidroponik

Hidroponik merupakan sebuah metode bercocok tanam modern tanpa menggunakan media tanah melainkan air, dengan menekankan pemenuhan kebutuhan nutrisi pada tanaman. Sistem ini sangatlah cocok untuk teknologi bercocok tanam yang dapat diadopsi untuk menghasilkan pangan (Reno Suryani, 2015). Penggunaan media air yang lebih efisien dibandingkan dengan menggunakan media tanah, menjadikan metode ini sangat cocok untuk daerah-daerah yang memiliki pasokan air yang terbatas. Jadi, selain untuk mengaplikasikan sistem daur ulang, dengan menggunakan sistem hidroponik juga dapat menghijaukan lingkungan sekitar. Tinggal di perkotaan dengan memiliki luas tanah yang kecil sekalipun tidak menutup kemungkinan untuk dapat menanam beragam tanaman seperti sayuran dan buah-buahan. Tanaman hortikultura seperti tanaman tomat sangat cocok untuk dibudidayakan menggunakan sistem hidroponik.

Hasil panen dengan menggunakan sistem budidaya hidroponik memiliki kualitas yang lebih higienis dan sehat karena dalam pengelolaannya sedikit menggunakan bahan-bahan kimia. Lain halnya dengan bercocok tanam menggunakan media tanah, karena harus mengelola tanah yang biasanya menggunakan bahan-bahan kimia sebagai perawatannya sehingga sering menimbulkan permasalahan. Selain itu, keuntungan menggunakan sistem hidroponik lainnya adalah perawatan lebih praktis serta gangguan hama akan lebih terkontrol. Sistem ini juga sangat ramah lingkungan dimana penggunaan air yang efisien dan pertumbuhan tanaman kualitasnya dapat terjaga dengan baik. Teknik hidroponik juga tidak membutuhkan banyak tenaga kasar karena metode kerjanya lebih hemat dan memiliki standarisasi, serta nutrisi dan air yang selalu tersedia.

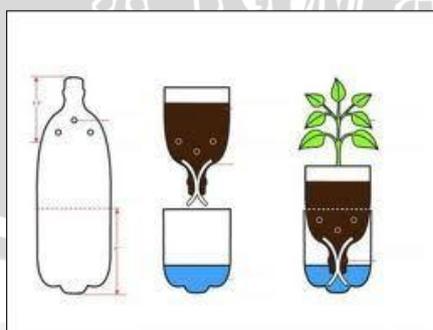
Dalam salah satu sistem hidroponik, terdapat sirkulasi nutrisi yang dapat digunakan ulang selama beberapa waktu, sesuai dengan kebutuhan tanaman yang dimaksud. Sebagian akar tanaman akan terendam di dalam larutan nutrisi, sementara bagian lainnya berada di atas permukaan. Faktor utama yang memengaruhi perkembangan tanaman tersebut adalah tersedianya nutrisi penunjang yang sesuai dengan jenis dan umur tanaman tersebut. Kadar nutrisi yang akan diserap oleh tanaman nantinya dapat dilihat dari berbagai macam indikator seperti halnya, kadar keasaman air (pH), *electro conductivity* (EC), kadar molaritas dan lain sebagainya. Daftar kebutuhan nutrisi ideal pada tanaman dijelaskan pada Tabel 2.2 berikut ini.

Tabel 2.2 Daftar Kebutuhan Nutrisi Ideal pada Tanaman (Suryani, Reno. 2015).

No.	Tanaman	pH	cF	EC	PPM
1.	Kubis	6.5-7.5	14-24	1.4-2.4	980-1680
2.	Wortel	5.8-6.4	14-22	1.4-2.2	980-1540
3.	Tomat	5.5-6.5	18-25	1.8-2.5	1260-1750
4.	Strawberry	5.5-6.5	18-25	1.8-2.5	1260-1750
5.	Bayam	6.0-7.5	18-35	1.8-3.5	1260-2450
6.	Mentimun	5.8-6.0	16-24	1.6-2.4	1120-1680
7.	Terong	5.5-6.5	18-24	1.8-2.4	1260-1540
8.	Bawang	6.5-7.0	18-22	1.8-2.2	1260-1540
9.	Asparagus	6.0-8.0	14-18	1.4-1.8	980-1260
10.	Brokoli	6.0-6.5	14-24	1.4-2.4	980-1680

Hidroponik sendiri dapat dikategorikan menjadi dua, yaitu sistem aktif dan sistem pasif. Hidroponik sistem aktif yaitu dimana aliran air beserta nutrisi dibuat bergerak dan bersirkulasi menggunakan pompa air. Contohnya adalah *nutrient film technique* (NFT), *deep flow technique* (DFT), dan juga *aeroponics*. Sedangkan, hidroponik sistem pasif adalah larutan nutrisi diserap oleh medium dan diteruskan ke akar tanaman tanpa tersirkulasi. Contohnya adalah sistem *wick* (sumbu) yang ditunjukkan pada Gambar 2.1.

Sistem hidroponik berdasarkan media tanam yang digunakan dibagi menjadi dua, yaitu kultur air dan kultur agregat. Sistem hidroponik memiliki berbagai macam sistem kultur air diantaranya adalah *deep flow technique* (DFT), teknologi hidroponik sistem terapung (THST), *aeroponics*, dan *nutrient film technique* (NFT). Kemudian juga terdapat di dalam media agregat seperti *sand culture*, *gravel culture*, *rockwool*, dan *bag culture*.



Gambar 2.1 Tanaman Hidroponik
Sumber: Suryani, Reno. (2015)

2.2.1 Sistem Hidroponik *Nutrient Film Technique* (NFT)

Dalam penelitian ini digunakan sistem hidroponik sistem aktif dengan kultur air yakni sistem hidroponik *Nutrient Film Technique* (NFT). Dengan menggunakan sistem ini, desain dapat dimodifikasi dengan mudah sesuai dengan yang diinginkan.

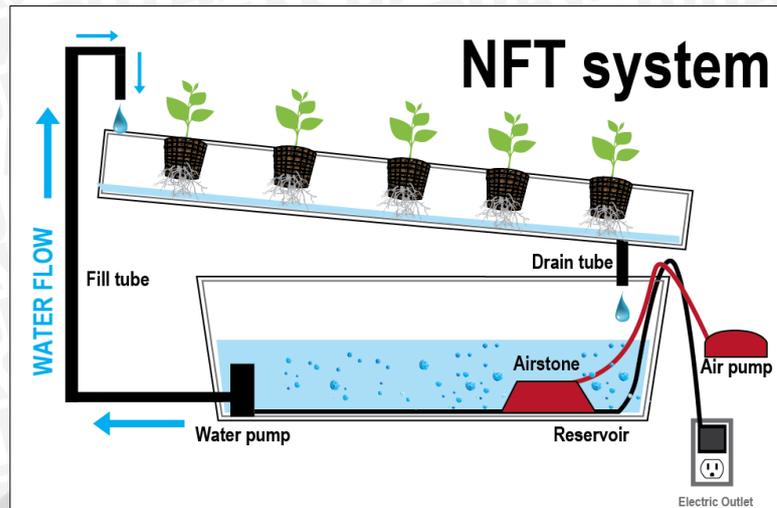
Cooper menyatakan bahwa *Nutrient Film Technique* (NFT) adalah sebuah sistem yang menggunakan ‘film’ larutan nutrisi. Film atau lapisan tipis setebal 1-3 mm ini dipompa dan dialirkan melewati akar tanaman secara terus-menerus dengan kecepatan aliran air sekitar 1-2 liter per menit.

Nutrient Film Technique (NFT) adalah teknik hidroponik dimana aliran air yang sangat dangkal mengandung semua nutrisi terlarut yang diperlukan untuk pertumbuhan tanaman. Larutan nutrisi mengalir melalui saluran kedap air seperti pipa paralon, dengan kedalaman sirkulasi aliran larutan nutrisi yang sangat dangkal. Dalam sistem yang ideal, kedalaman aliran sirkulasi harus sangat dangkal, sedikit lebih dari sebuah film air.

Keuntungan utama dari sistem ini adalah akar tanaman yang terkena aliran nutrisi akan mengalami kecukupan pasokan air, oksigen, dan nutrisi. Artinya dalam prinsip dasar pertanian konvensional, apabila sistem ini pada kondisi air berlebih, jumlah oksigen biasanya tidak memadai. Namun, dengan sistem ini ketersediaan nutrisi dan oksigen pada akar akan selalu berlimpah.

Sirkulasi nutrisi dapat digunakan ulang selama beberapa minggu sesuai dengan kebutuhan tanaman. Sebagian akar tanaman tumbuh di atas permukaan larutan nutrisi dan sebagian lagi terendam di dalamnya. Faktor utama yang mempengaruhi perkembangan tanaman hidroponik *Nutrient Film Technique* (NFT) adalah tersedianya nutrisi penunjang yang sesuai dengan jenis dan umur tanaman dan kestabilan kecepatan aliran nutrisi.

Banyak sekali digunakan untuk menanam tanaman dan sayuran. Sistem *Nutrient Film Technique* (NFT) dapat menghasilkan lebih tanaman dengan ruangan yang sedikit, sedikit air dan sedikit nutrisi. Selain itu, ada aerasi yang baik dan suplai oksigen di sebagian besar sistem hidroponik. Sistem *Nutrient Film Technique* (NFT) juga sangat mudah dalam pembuatan dan pemeliharaan. Seperti pada Gambar 2.2.



Gambar 2.2 Nutrient Film Technique (NFT) System.
 Sumber: Suryani, Reno. (2015)

2.3 Mesin Direct Current (DC)

Mesin *Direct Current* (DC) merupakan alat atau perangkat yang memiliki fungsi untuk mengkonversikan energi mekanik menjadi energi listrik begitu pula sebaliknya. Mesin DC memiliki 2 elemen yakni stator dan rotor. Stator merupakan elemen mesin DC yang berdiri tetap/statis, sedangkan rotor merupakan elemen mesin DC yang berputar/dinamis. Stator sendiri terdiri dari beberapa bagian yaitu gandar, kutub utama, kutub komutasi/kutub bantu, dan kumparan penguat. Sementara itu, rotor terdiri dari inti jangkar, kumparan jangkar, kawat baja pengikat, dan komutator. Selain itu, mesin DC juga terdiri dari celah-celah udara dan sikat-sikat.

2.3.1 Motor Direct Current (DC)

Mesin *Direct Current* (DC) yang memiliki fungsi untuk mengkonversikan energi listrik menjadi energi mekanik adalah motor DC. Motor DC merupakan perangkat listrik yang sangat mudah untuk dikontrol. Apabila arus dialirkan melalui kumparan jangkar dari sebuah mesin DC dan kumparan medannya diberi penguatan maka akan timbul gaya *Lorentz* pada setiap sisi kumparan jangkar tersebut (Soemarwanto, 2010). Hal tersebut merupakan prinsip dasar kerja motor DC.

Besar gaya *Lorentz* yang ditimbulkan motor DC adalah:

$$F = B.I.l \text{ (dyne)} \quad (2.1)$$

Keterangan:

B = kerapatan medan magnetik (Gauss)

I = arus listrik (Ampere)

l = panjang konduktor (cm)

Akibat adanya gaya *Lorentz*, maka akan timbul kopel yang besarnya adalah:

$$\text{Kopel: } M = F.r \text{ (dyne-cm)} \quad (2.2)$$

Keterangan:

F = gaya *Lorentz* (dyne)

r = jari-jari jangkar (cm)

$$M = B.I.l.r \text{ (dyne-cm)} \quad (2.3)$$

Apabila jumlah: $M = z.B.I.l.r$ (dyne-cm), karena $B = \frac{\theta}{A}$.

Keterangan:

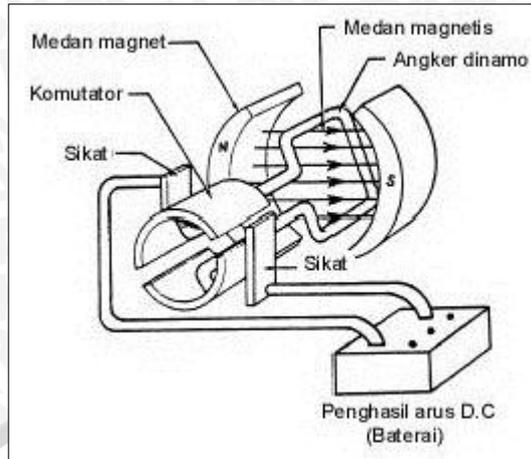
θ = fluksi tiap kutub (*Maxwell*)

A = luas penampang (cm²)

z = jumlah konduktor pada jangkar

$$\text{maka: } M = z. \frac{\theta}{A}. I.l.r \text{ (dyne-cm)} \quad (2.4)$$

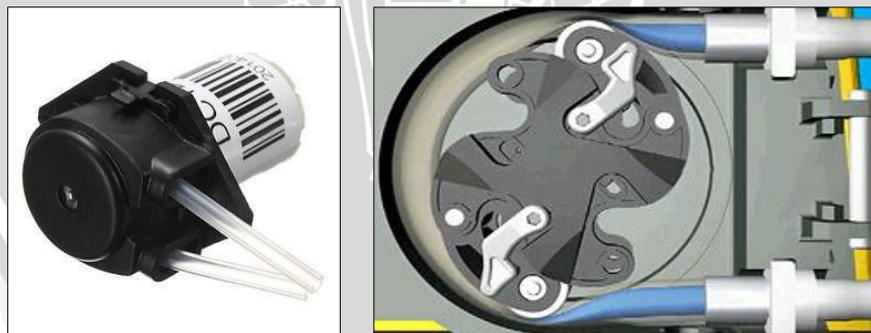
Motor DC ditunjukkan pada Gambar 2.3 membutuhkan catu daya tegangan yang searah pada kumparan medan. Catu tegangan tersebut akan menuju ke lilitan melalui sikat yang bersentuhan dengan komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan dalam satu lilitan yang berputar di antara medan magnet disebut dengan *Armature* (jangkar).



Gambar 2.3 Motor Direct Current (DC).

2.3.2 Motor Direct Current (DC) Peristaltic Pump 12V

Motor DC memiliki beberapa jenis di antaranya adalah motor DC pompa peristaltik. Motor DC ini berperan sebagai pompa yang beroperasi dengan prinsip membuat perbedaan tekanan antara bagian masuk (*suction*) dengan bagian keluar (*discharge*). Dengan kata lain, pompa berfungsi mengubah tekanan mekanis dari suatu sumber tenaga atau penggerak menjadi tenaga kinetis atau kecepatan. Tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada di setiap proses yang kita jalankan. Pompa ini digunakan untuk memindahkan cairan dari suatu tempat ke tempat lainnya melalui suatu media perpipaan dengan cara menambahkan suatu energi pada cairan yang dipindahkan dan berlangsung secara terus-menerus. Seperti pada Gambar 2.4 dan 2.5.



Gambar 2.4 dan 2.5 Motor DC Peristaltic Pump 12V, Prinsip Kerja Peristaltic Pump.
Sumber: Treutel, Chuck. (2009)

2.3.3 Pulse Width Modulation

Metode *Pulse Width Modulation* (PWM) adalah salah satu metode yang dapat digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle*, akan diperoleh keluaran yang diinginkan (Alfarisi.R.,2011). *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada motor. Persamaan untuk perhitungan *duty cycle* ditunjukkan dalam persamaan 2.5 dengan T_{ON} adalah periode logika tinggi dan T adalah periode keseluruhan.

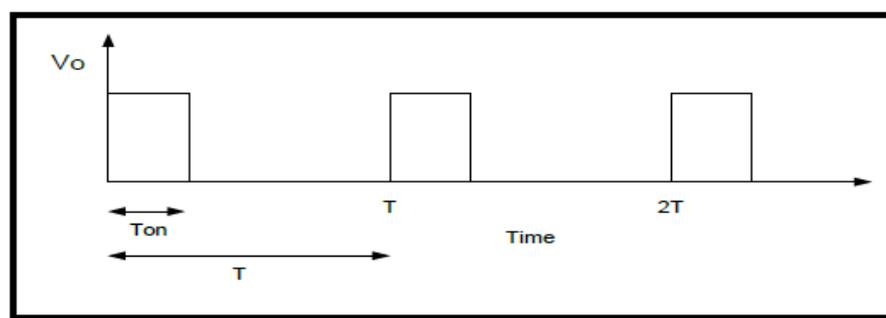
$$Duty\ cycle = \frac{T_{ON}}{T} \times 100\% \quad (2.5)$$

Keterangan:

T_{ON} = periode logika tinggi

T = periode keseluruhan

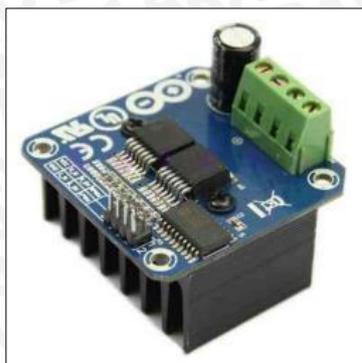
Sinyal PWM secara umum ditunjukkan oleh Gambar 2.6.



Gambar 2.6 Sinyal PWM.
Sumber: Ardyani, F. (2013)

2.4 High Current 30 A DC Motor Driver dengan Half-Bridge Infineon BTS 7960 chip

High Current 30 A DC Motor Driver (Gambar 2.7) adalah *driver* H-Bridge yang dirancang untuk dapat menghasilkan *drive* dengan 2 arah gerakan *clockwise* (cw) dan *counterclockwise* (ccw). Arus yang didapat juga kontinyu hingga 30 A dan tegangan antara 5 hingga 27 Volt. Modul ini mampu untuk menggerakkan motor DC 12V dan dilengkapi dengan rangkaian sensor arus beban. Rangkaian ini dihubungkan dengan mikrokontroler Arduino Mega 2560.



Gambar 2.7 High Current 30 A Motor Driver.
 Sumber: *datasheet High Current 30 A Motor Driver*

Fungsi masing-masing pin pada *Interface Header* dapat dilihat dalam Tabel 2.3.

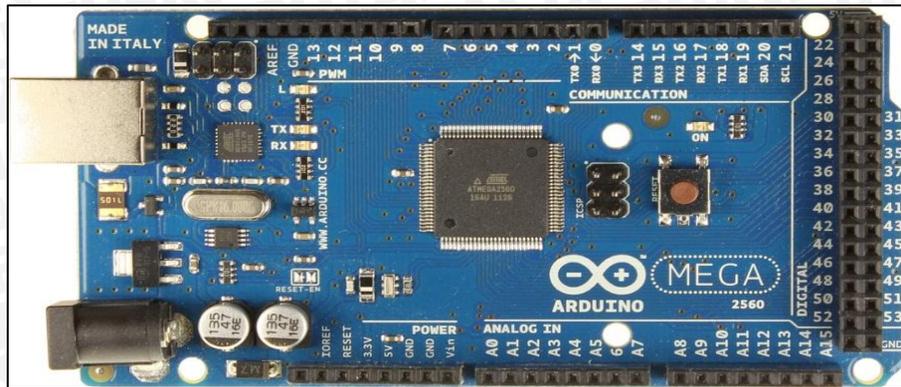
Tabel 2.3 Fungsi pin pada *Interface Header High Current 30 A DC Motor Driver (datasheet)*

No.	Pin	Fungsi
1.	RPWM	<i>Forward level atau input sinyal PWM, aktif HIGH</i>
2.	LPWM	<i>Reverse level atau input sinyal PWM, aktif LOW</i>
3.	R_EN	<i>Forward drive enable input, HIGH enable, LOW close</i>
4.	L_EN	<i>Reverse drive enable input, HIGH enable, LOW close</i>
5.	R_IS	<i>Forward drive-side current alarm output</i>
6.	L_IS	<i>Reverse drive-side current alarm output</i>
7.	VCC	<i>5V power input dari board arduino</i>
8.	GND	<i>Ground</i>

2.5 Mikrokontroler Arduino Mega 2560

Arduino Mega 2560 adalah *board* mikrokontroler berbasis ATmega 2560. Modul ini memiliki 54 digital *input/output* dimana 14 digunakan untuk PWM *output* dan 16 digunakan sebagai analog *input*, 4 untuk UART, 16 MHz osilator kristal, koneksi USB, *power jack*, ICSP *Header*, dan tombol *reset*. Modul ini memiliki segalanya yang dibutuhkan untuk merancang program mikrokontroler seperti kabel USB dan sumber daya melalui *adaptor* maupun *battery*.

Bahasa yang digunakan dalam modul arduino bukan bahasa *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, dapat dilihat dengan Gambar 2.8.



Gambar 2.8 Arduino Mega 2560.

Sumber: *electroschematics.com*

Oleh karena itu, sehubungan dengan pembahasan untuk saat ini *software* arduino yang akan digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat canggih ditulis dengan menggunakan *Java*. IDE arduino terdiri dari :

- Editor program, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
- *Compiler*, sebuah modul yang mengubah kode program (bahasa *processing*) menjadi kode biner.
- *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan arduino.

2.5.1 Daya

Arduino Mega 2560 dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menghubungkan *plug jack* pusat-positif ukuran 2.1mm konektor *power*. Ujung kepala dari baterai dapat dimasukkan ke dalam *Gnd* dan V_{IN} *pin header* dari konektor *power*. Kisaran kebutuhan daya yang disarankan untuk *board* Arduino Mega 2560 adalah 7 sampai dengan 12 Volt, jika diberi daya kurang dari 7 volt kemungkinan pin 6V Arduino Mega 2560 dapat beroperasi tetapi tidak stabil. Kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *board*.

Pin listrik pada Arduino Mega 2560 adalah sebagai berikut:

1. V_{IN} adalah tegangan masukan kepada *board* arduino ketika itu menggunakan sumber daya eksternal (sebagai pengganti dari 5 volt koneksi USB atau sumber daya lainnya).

2. 5V adalah catu daya digunakan untuk daya mikrokontroler dan komponen lainnya.
3. 3v3 adalah sebuah pasokan 3,3 Volt dihasilkan oleh *regulator on-board*. Maksimum arus adalah 50 mA.
4. *GND* adalah sebuah *ground pin*.

2.5.2 Memori

Atmega2560 memiliki 256 KB (dengan 8 KB digunakan untuk *bootloader*), 8 KB dari SRAM dan 4 KB EEPROM.

2.5.3 Input & Output

Masing-masing dari 54 pin digital pada Arduino Mega 2560 dapat digunakan sebagai *input* ataupun *output*, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*, beroperasi dengan daya 5 Volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki *internal pull-up* resistor (secara *default* terputus) dari 20-50 KOhms.

Selain itu, beberapa pin memiliki fungsi khusus:

1. *Serial*: 0(RX) dan 1(TX); *Serial 1*: 19(RX) dan 18(TX); *Serial 2*: 17(RX) dan 16(TX); *Serial 3*: 15(RX) dan 14(TX) digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data *serial*. Pin ini dihubungkan ke pin yang berkaitan dengan *chip Serial ATmega8U2 USB-to-TTL*.
2. *Eksternal menyela*: 2(*interrupt 0*), 3(*interrupt 1*), 18(*interrupt 5*), 19(*interrupt 4*), 20(*interrupt 3*), 21(*interrupt 2*). Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.
3. *PWM*: 0 hingga 13. Menyediakan *output PWM* 8-bit dengan fungsi *analogWrite()*.
4. *SPI*: 53 (SS), 51 (MOSI), 50 (MISO), 52 (SCK). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.
5. *LED*: 13. Ada *built-in LED* terhubung ke pin digital 13. Ketika pin bernilai nilai HIGH, LED *on*, ketika pin bernilai LOW, LED *off*.
6. *I²C*: 20(*SDA*) dan 21(*SCL*). Mendukung komunikasi *I²C (TWI)* menggunakan *Wire Library*.

repository.ub.ac.id

Arduino Mega 2560 memiliki 16 masukan analog, berlabel A0 sampai dengan A15, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda). Selain itu, beberapa pin memiliki fungsi khusus :

1. *Aref*. Tegangan referensi (0 sampai 5V saja) untuk input analog. Digunakan dengan fungsi *analogReference()*.
2. *Reset*. Bawa baris ini LOW untuk me-*reset* mikrokontroler.

2.5.4 Komunikasi

Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, arduino lain, atau mikrokontroler lainnya. Atmega2560 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi *serial* melalui USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada *windows* diperlukan sebuah *file* inf. Perangkat lunak arduino terdapat monitor *serial* yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke ataudari *board* arduino. LED RX dan TX di layar akan berkedip ketika data sedang dikirim melalui *chip* USB-*to*-*serial* dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *software serial library* memungkinkan untuk berkomunikasi secara *serial* pada salah satu pin digital pada *board* Arduino Mega 2560. Atmega2560 juga mendukung I2C (TWI) dan komunikasi SPI.

2.6 Sensor pH

Sensor pH (Gambar 2.9) merupakan elektroda gelas yang terdiri dari gelembung gelas sensitif pH pada ujungnya, berisi larutan klorida yang diketahui pHnya dan elektroda referensi. Biasanya berupa perak atau perak klorida atau *calomel*. Bahan gelas terbuat dari komposisi gelas khusus, yaitu sebagian dihidrasi alumunium silikon yang mengandung ion sodium atau kalsium dan sebagian kecil ion lantanida.

Bagian dalam dari gelas akan kontak dengan larutan 0,1 m HCl dan bagian luar kontak dengan larutan yang diukur pHnya. Pada setiap permukaan, membran gelas akan menyerap air dan membentuk lapisan gel. Ion hidrogen dari larutan dapat berdifusi melalui lapisan gel dan mengganti (pertukaran ion) ion sodium atau logam lain dalam struktur gelas. Hasil akhir dari proses difusi dan pertukaran ion adalah fase pengikatan

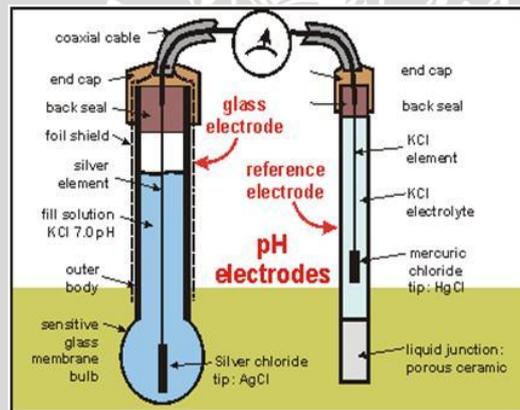
UNIVERSITAS BRAWIJAYA

tegangan pada kedua sisi membran gelas akan berubah, di mana besarnya ditentukan oleh aktivitas ion hidrogen dalam larutan yang diukur.



Gambar 2.9 Sensor pH SKU SEN0161.
 Sumber: datasheet Sensor pH SKU SEN0161

Ciri khas pH *probe* modern adalah kombinasi yang menggabungkan kedua elektroda kaca dan elektroda referensi menjadi satu bagian. Prinsip sensor pH dapat dilihat dalam Gambar 2.10 berikut ini.



Gambar 2.10 Prinsip Sensor pH.
 Sumber: elprocus.com

Berikut adalah karakteristik elektroda pH ideal:

- $V_o = 0$ Volt pada pH netral (pH=7)
- Tegangan keluaran positif pada kondisi asam (pH<7)
- Tegangan keluaran negatif pada kondisi basa (pH>7)
- Rentang pengukuran pH adalah 0-14, asam kuat hingga basa kuat

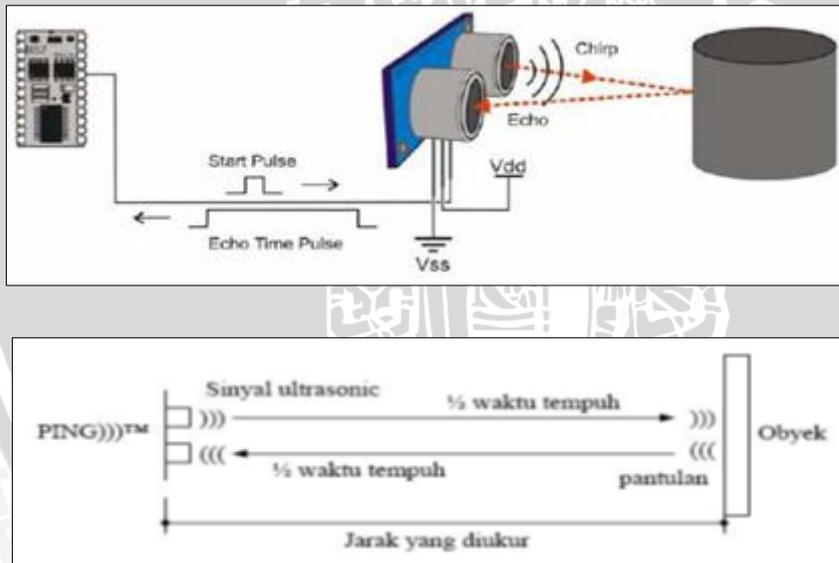
Menghasilkan tegangan sebesar -59,16 mV per satuan pH pada suhu ruang atau sekitar 25° C.

2.7 Sensor Ultrasonik

Sensor ultrasonik seperti pada Gambar 2.11 merupakan jenis sensor pengukur jarak yang menggunakan ultrasonik. Dimana prinsip kerja sensor ini pada Gambar 2.12 adalah pemancar (*transmitter*) mengirimkan seberkas gelombang ultrasonik, kemudian diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan objek, sehingga didapat jarak sensor dengan objek yang bisa ditentukan dengan persamaan.



Gambar 2.11 Sensor Ultrasonik SRF05.
Sumber: datasheet Sensor Ultrasonik SRF05



Gambar 2.12 Prinsip Kerja Sensor Ultrasonik SRF05.
Sumber: <http://kuliahandifajar.com/sensor-ultrasonic/>

Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai objek dan memantul kembali ke sensor. Sensor ini mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi sensor akan membuat *output low* pada pin SIG. Lebar pulsa *High* (t_{IN}) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk dua kali jarak ukur dengan objek (Fajar, 2013). Maka jarak yang diukur adalah sebagai berikut:

$$S = \frac{(t_{IN} \times V)}{2} \quad (2.6)$$

S = Jarak antara sensor ultrasonik dengan objek yang dideteksi.

t_{IN} = Selisih waktu pemancaran dan penerimaan pantulan gelombang.

V = Cepat rambat gelombang ultrasonik di udara (344 m/s).

Sensor ultrasonik SRF05 dengan karakteristik sebagai berikut :

1. Bekerja pada tegangan DC 5 Volt.
2. Beban arus sebesar 30 mA – 50 mA.
3. Menghasilkan gelombang dengan frekuensi 40 KHz.
4. Jangkauan jarak yang dapat dideteksi 3 cm – 400 cm.
5. Membutuhkan *trigger* input minimal sebesar 10 μ s.

2.8 Kontroler

Salah satu komponen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*setpoint*) atau sinyal *error* mejadi sinyal kontrol. Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika diberi gangguan (Ogata K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespon *input* dan bagaimana peredaman terhadap adanya lonjakan (*overshoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *outputnya* berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi, maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respon transien berupa suatu osilasi teredam sebelum mencapai *steady state*. Dalam sistem pengendalian terdapat 2 macam loop:

1. Pengendalian dengan Loop Terbuka

Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi, pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpan balik untuk dibandingkan dengan masukan.

2. Pengendalian dengan Loop Tertutup

Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

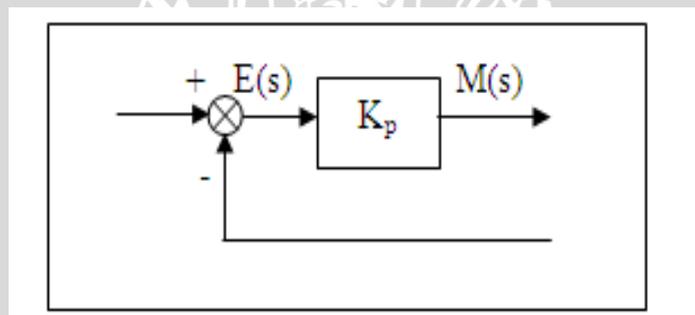
$$m(t) = K_p e(t)$$

atau, dalam besaran transformasi *Laplace*,

$$(M(s))/E(s) = K_p$$

dimana K_p adalah penguatan proporsional.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggeraknya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997). Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.13.



Gambar 2.13 Diagram Blok Kontroler Proporsional.

Sumber: Ogata K., (1997)

2.9 Jenis-jenis Kontroler

2.9.1 Kontroler Proporsional (P)

Kontroler proporsional adalah sebuah kontroler yang memiliki karakteristik mempercepat respon. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ ditunjukkan pada persamaan 2.7:

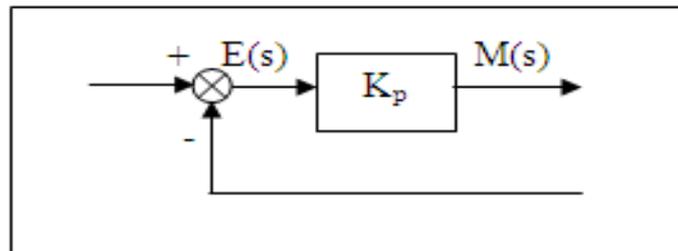
$$m(t) = K_p e(t) \quad (2.7)$$

atau, dalam besaran transformasi *Laplace* ditunjukkan pada persamaan 2.8:

$$\frac{M(s)}{E(s)} = K_p \quad (2.8)$$

Di mana K_p adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997). Diagram blok kontroler proporsional ditunjukkan pada Gambar 2.14.



Gambar 2.14 Diagram Blok Kontroler Proporsional.

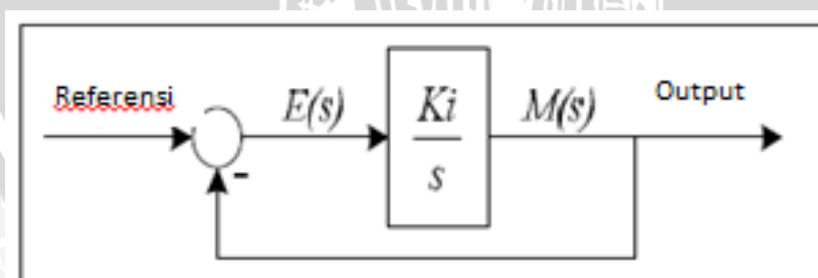
Sumber: Ogata K., (1997)

2.9.2 Kontroler Integral (I)

Kontroler integral memiliki kemampuan untuk mengurangi *offset* yang ditinggalkan oleh kontroler proporsional. Harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$. Persamaan kontroler K_i ditunjukkan dalam persamaan 2.9 (Ogata K.,1997).

$$\frac{dm(t)}{dt} = K_i e(t) \quad (2.9)$$

K_i adalah penguatan integral. Gambar 2.15 menunjukkan diagram blok kontroler integral.



Gambar 2.15 Diagram Blok Kontroler Integral.

Sumber: Ogata K., (1997)

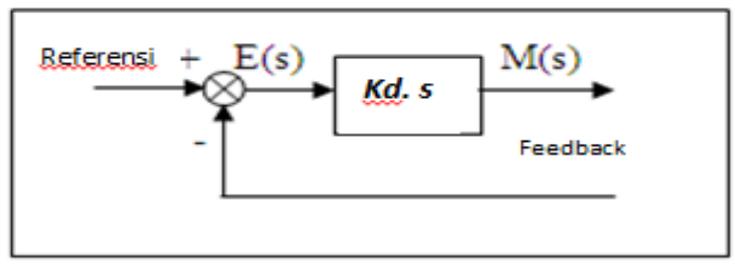
2.9.3 Kontroler Differensial (D)

Kontroler ini digunakan untuk memperbaiki atau mempercepat respon transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol

dan mengurangi *phase lag* penguatan tersebut (Ogata K.,1997). Kontroler differensial tidak dapat mengeluarkan *output* bila tidak ada perubahan *input*, selain itu kontroler differensial tidak dapat digunakan untuk proses yang mengandung *noise*. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ dinyatakan pada persamaan 2.10.

$$\frac{M(s)}{E(s)} = Kd \cdot s \quad (2.10)$$

Kd adalah penguatan differensial. Gambar 2.16 menunjukkan diagram blok kontroler differensial.



Gambar 2.16 Diagram Blok Kontroler Differensial.

Sumber: Ogata K., (1997)

2.9.4 Kontroler Proporsional Integral Differensial (PID)

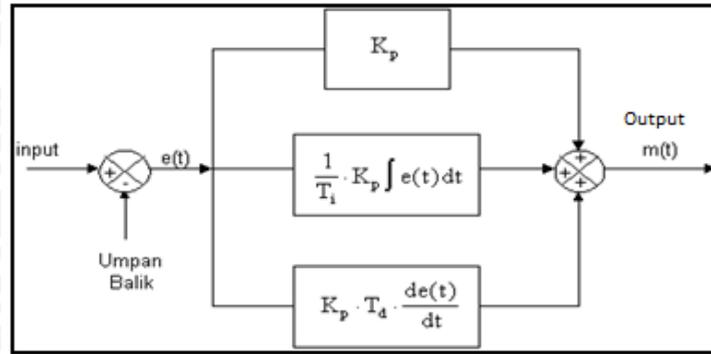
Gabungan aksi kontrol proporsional, integral, dan differensial mempunyai keunggulan dibandingkan dengan masing-masing dari tiga aksi kontrol tersebut. Masing – masing kontroler P, I, maupun D berfungsi untuk mempercepat reaksi sistem, menghilangkan *offset*, dan mendapatkan energi ekstra ketika terjadi perubahan *load*. Persamaan kontroler PID ini dapat dinyatakan dalam persamaan 2.11:

$$m(t) = Kp \cdot e(t) + \frac{Kp}{Ti} \cdot e(t)dt + Kp \cdot Td \frac{de(t)}{dt} \quad (2.11)$$

Dalam transformasi *Laplace* dinyatakan dalam persamaan 2.12:

$$\frac{M(s)}{E(s)} = Kp \left(1 + \frac{1}{Ti \cdot s} + Td \cdot s \right) \quad (2.12)$$

Ti adalah waktu integral dan Td adalah waktu derivatif. Gambar 2.17 menunjukkan diagram blok kontroler PID.



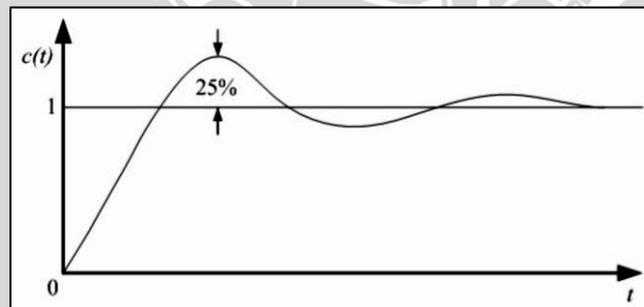
Gambar 2.17 Diagram Blok Kontroler PID.

Sumber: Ogata K., (1997)

2.10 Ziegler Nichols

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan *plant* (Ogata, K., 1997).

Terdapat dua metode yang disebut dengan aturan penalaan *Ziegler-Nichols*, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon *unit step*, seperti ditunjukkan dalam Gambar 2.18.

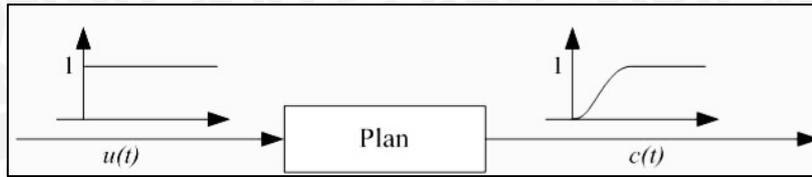


Gambar 2.18 Kurva Respon *Unit Step* yang Menunjukkan 25% *Maximum Overshoot*.

Sumber: Ogata K., (1997)

a. Metode Pertama

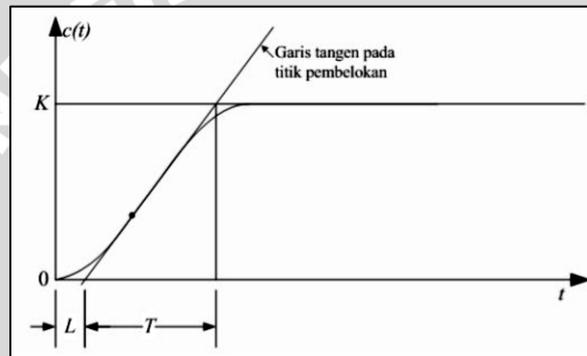
Metode pertama atau sering disebut metode kurva reaksi, respon dari *plant* dapat dapat diperoleh secara eksperimental dengan masukan berupa *unit step*, seperti yang ditunjukkan dalam Gambar 2.19.



Gambar 2.19 Respon *Plant* Terhadap Masukan Berupa *Unit Step*.

Sumber: Ogata K., (1997)

Jika dalam *plant* tersebut minimal tidak mengandung unsur integrator atau *dominan complex-conjugate poles*, maka kurva respon *unit step* berbentuk seperti huruf S, seperti dalam Gambar 2.20 jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku (Ogata, K., 1997).



Gambar 2.20 Kurva Respon yang Berbentuk S.

Sumber: Ogata K., (1997)

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu mati L dan waktu tunda T . Waktu mati dan waktu tunda ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.20 Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \tag{2.13}$$

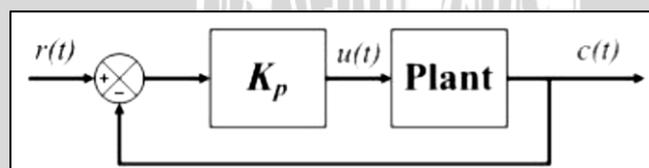
Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.4.

Tabel 2.4 Aturan Penalaan *Ziegler-Nichols* Berdasarkan Respon *Unit Step* dari *Plant* (Sumber: Ogata, K., 1997)

Type Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5L$

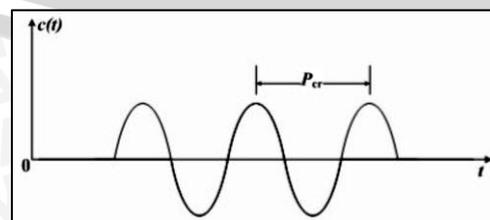
b. Metode Kedua

Dalam metode kedua *ziegler-nichols*, mula-mula yang dilakukan adalah membuat $T_i = \infty$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Diagram blok sistem *loop* tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.21 dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam Gambar 2.22 *Ziegler* dan *Nichols* menyarankan penyetelan nilai parameter K_p , T_i , T_d dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.5.



Gambar 2.21 Sistem *Loop* Tertutup dengan Kontroler Proporsional.

Sumber: Ogata K., (1997)



Gambar 2.22 Osilasi Berkesinambungan dengan Periode P_{cr} .

Sumber: Ogata K., (1997)

Tabel 2.3 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain Kcr* dan *Critical Period Pcr*
(Sumber: Ogata, K. 1997)

Tipe Kontroler	K_p	T_i	T_d
P	0.5 Kcr	∞	0
PI	0.45 Kcr	$\frac{1}{1,2} Pcr$	0
PID	0.60 Kcr	0.5 Pcr	0.125 r

2.11 Transformasi Z

Dalam suatu sistem yang berulang, transformasi *Laplace* dapat digunakan dalam sebuah analisis maupun desain. Sedangkan, transformasi Z memiliki fungsi untuk dapat dimanfaatkan dalam menganalisa sistem yang termodelkan dalam waktu diskrit dari persamaan differensial. Sebuah transformasi ditetapkan sebagai deret angka yang berurutan. Fungsi $E(z)$ ditetapkan sebagai sebuah *power series* dalam z^{-k} dengan koefisien yang sama untuk nilai dari deret angka $\{e(k)\}$. Transformasi ini disebut sebagai transformasi Z, yang kemudian diungkapkan dalam bentuk persamaan 2.13.

$$E(z) = z\{E(k)\} = e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots \quad (2.13)$$

$$\text{Dengan } e(k) = z^{-1}\{E(z)\} = \frac{1}{2\pi f} \oint_{\Gamma} E(z)z^{k-1} dz, f = \sqrt{-1}$$

Di mana $z(\dots)$ mengindikasikan operasi transformasi Z dan $Z^{-1}(\dots)$ mengindikasikan *invers* dari transformasi z. $E(z)$ dalam Persamaan 2.13 dapat dituliskan dalam notasi yang lebih sederhana seperti pada persamaan berikut:

$$E(z) = Z\{e(k)\} = \sum_{k=0}^{\infty} e(k)z^{-k} \quad (2.14)$$

Transformasi Z terdefiniskan untuk beberapa deret angka $\{e(k)\}$, dan mungkin digunakan dalam menganalisis dari beberapa jenis sistem yang diuraikan oleh persamaan differensial *time-invariant* linier.

2.12 Diskritisasi

Ada beberapa cara dalam menentukan proses diskritisasi diantaranya adalah *backward difference*, *forward difference*, dan *bilinear transform*. Semua metode adalah metode pendekatan (*approximations*), sehingga hasil yang diperoleh tidak akan sama

dengan bentuk analognya. Hal ini disebabkan karena bentuk diskrit akan ada bagian informasi yang hilang, karena formula matematika yang digunakan dalam proses diskritisasi juga diturunkan berdasarkan pendekatan yang lebih mudah digunakan.

Menurut teori *sampling Nyquist* dan *Shannon*, jika fungsi waktu $e(t)$ tidak terdapat unsur frekuensi yang lebih tinggi 1 Hz, dalam hal ini nilai dari nilai $e(t)$ dapat ditentukan dengan memberikan titik *sampling* berjarak $1/2f_0$ detik terpisah.

Berikut tahapan diskritisasi yang dapat digunakan:

1. Tulis algoritma analog bentuk transformasi *Laplace*.
2. Lakukan diskritisasi menjadi bentuk transformasi Z dengan mengganti operator s dengan menggunakan salah satu dari titik metode diskritisasi yaitu:

- *Back difference*:

$$s = \frac{1-z^{-1}}{T_s} \quad (2.15)$$

- *Forward difference*:

$$s = \frac{1-z^{-1}}{T_s z^{-1}} \quad (2.16)$$

- *Bilinear transform*:

$$s = \frac{2(1-z^{-1})}{T_s(1+z^{-1})} \quad (2.17)$$

Dimana T_s merupakan waktu *sampling*.

Hingga tahap 2, algoritma sudah didapat dalam bentuk diskrit yang dinyatakan dalam bentuk transformasi Z. Namun, pada implementasinya bentuk transformasi Z tersebut perlu diubah menjadi bentuk *time domain*, yaitu dengan mengubah operator z^n menjadi n kali waktu *delay* (z^{-1} berarti 1 kali waktu *delay*, z^{-2} berarti 2 kali waktu *delay* dan seterusnya).

2.13 Parameter Performansi Sistem

Parameter-parameter dibagi menjadi lima (performansi sistem) antara lain:

- Waktu penetapan (*Settling time*), T_s : waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah di sekitar nilai akhir yang ukurannya ditentukan dengan presentase mutlak dari nilai akhir (biasanya 5% atau 2%). Waktu penetapan ini dihubungkan dengan konstanta waktu terbesar

dari sistem pengontrolan. Kriteria presentase kesalahan yang akan digunakan ditentukan dari sasaran desain.

- Waktu tunda (*Delay time*), T_d : waktu yang diperlukan respon untuk mencapai setengah nilai akhir pada saat lonjakan yang pertama kali.
- Waktu naik (*Rise time*), T_r : waktu yang diperlukan respon untuk naik dari 10 sampai 90%, 5 sampai 95%, atau 0 sampai 100% dari nilai akhir. Untuk sistem orde dua redaman kurang (*Underdamped*) digunakan waktu naik 0 hingga 100% dan untuk sistem redaman lebih (*Overdamped*) digunakan waktu naik 10 hingga 90%.
- Waktu puncak (*Peak time*), T_p : waktu yang diperlukan respon untuk mencapai puncak lewatan (lonjakan maksimal) yang pertama kali.
- Lewatan maksimum (*Maximum overshoot*), M_p : nilai puncak maksimum kurva respon yang diukur dari satu. Jika nilai keadaan mantap respon tidak sama dengan satu, maka dapat digunakan persen lewatan maksimum.



BAB III

METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan informasi dengan tujuan dan manfaat tertentu. Dalam menyelesaikan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

- 3.1 Perancangan blok diagram sistem
- 3.2 Spesifikasi desain
- 3.3 Karakterisasi setiap blok

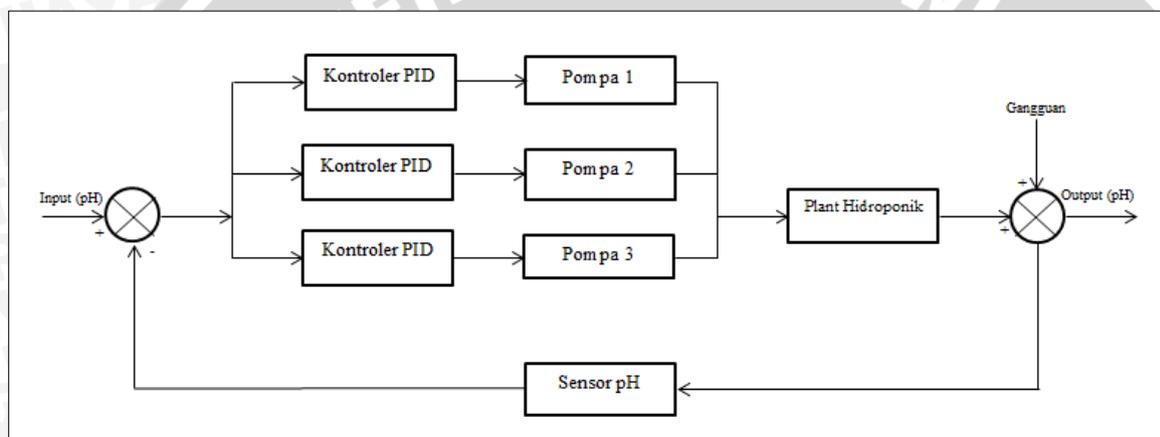
Karakterisasi setiap blok dilakukan untuk mempermudah analisis sistem.

Karakterisasi dibagi menjadi beberapa bagian, yaitu:

- 3.3.1 Karakterisasi Motor DC *Peristaltic Pump* 12V
- 3.3.2 Karakterisasi *Driver Motor (High Current 30 A DC Motor Driver)*
- 3.3.3 Karakterisasi Motor Pompa Mini DC 12V
- 3.3.4 Karakterisasi sensor pH SKU SEN1601
- 3.3.5 Karakterisasi sensor ultrasonik SRF05
- 3.4 Pembuatan perangkat keras
 - 3.4.1 Desain media hidroponik
 - 3.4.2 Prinsip kerja alat
 - 3.4.3 Rangkaian catu daya
 - 3.4.4 Konfigurasi pin Arduino Mega 2560
 - 3.4.5 Konfigurasi pin *driver motor*
 - 3.4.6 Konfigurasi pin *relay*
- 3.5 Perancangan algoritma
 - 3.5.1 Penentuan respon karakteristik pengontrolan pH
 - 3.5.2 Parameter kontroler PID dengan metode *Ziegler-nichols*
 - 3.5.3 Diskritisasi
 - 3.5.4 *Flowchart* Program

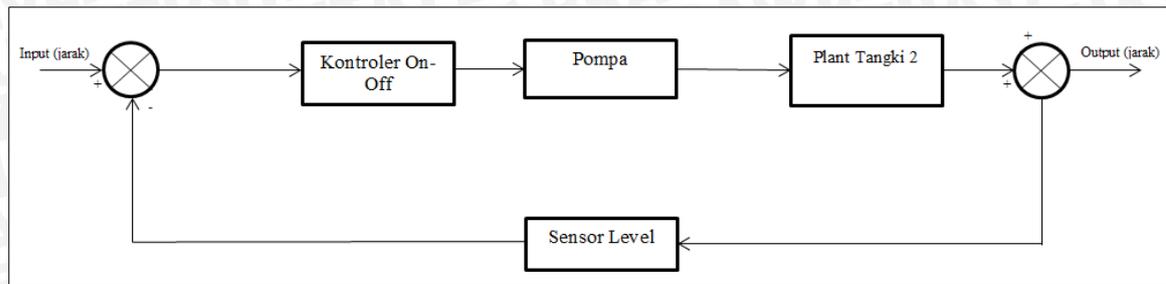
3.1 Perancangan Blok Diagram Sistem

Pada perancangan alat diperlukan perancangan diagram blok sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan yang diinginkan. Blok diagram sistem dapat dilihat pada Gambar 3.1 dan 3.2. Pada Gambar 3.1 merupakan blok diagram untuk sistem pengontrolan nutrisi pada tangki I. *Input* yang digunakan pada sistem ini adalah pH, dengan menggunakan kontroler PID, aktuator yang digunakan adalah motor DC pompa peristaltik 12V, *plant* dari sistem ini adalah sistem hidroponik tomat dengan metode *Nutrient Film Technique* (NFT), gangguan berupa penambahan larutan asam dengan volume 100 ml, sensor yang digunakan untuk mendapatkan *feedback* adalah sensor pH SKU SEN0161 dan *output* dari sistem ini juga berupa pH.



Gambar 3.1 Blok Diagram Sistem Tangki I).
Sumber: Perancangan

Pada Gambar 3.2 merupakan blok diagram untuk sistem pengontrolan volume larutan sisa nutrisi pada tangki II. *Input* yang digunakan pada sistem ini adalah jarak (cm), kontroler yang digunakan adalah dengan menggunakan kontroler *on off*, aktuator yang digunakan adalah motor pompa, *plant* dari sistem tangki II adalah larutan sisa pakai yang tidak diserap oleh tanaman pada *plant* tomat hidroponik, sensor yang digunakan adalah sensor ultrasonik SRF05 dan *output* berupa jarak (cm). Apabila sudah mencapai jarak batas yang telah ditentukan maka motor pompa akan menyala sehingga, larutan sisa nutrisi yang ada pada tangki II tidak akan tumpah dan dapat dipakai kembali.



Gambar 3.2 Blok Diagram Sistem Tangki (II).
Sumber: Perancangan

3.2 Spesifikasi Desain

Desain yang diinginkan pada perancangan pengontrolan nutrisi sistem tomat hidroponik yaitu:

1. Sistem hidroponik yang digunakan adalah dengan metode *Nutrient Film Technique* (NFT) di mana menggunakan dua buah tangki masing-masing untuk tangki I adalah 40x30x10 cm dan untuk tangki II adalah 40x30x25 cm dengan ketebalan 0.3 cm. Ukuran ini digunakan karena dapat menampung air maksimal hingga 12 liter untuk tangki I dan 30 liter untuk tangki II.
2. Pipa yang digunakan adalah pipa paralon berdiameter 4 inci (10.16 cm) dengan panjang 57 cm. Penggunaan pipa agar dapat menampung pot hidroponik dan akar tanaman dapat menyentuh dasar pipa tempat air mengalir dengan bantuan kain seperti prinsip sistem NFT.
3. Motor pompa yang digunakan untuk pengontrolan nutrisi adalah motor DC *peristaltic pump* dengan catu daya sebesar 12V. Motor DC ini digunakan karena catu daya yang digunakan adalah sebesar 12V DC.
4. Motor pompa yang digunakan untuk mengalirkan air ke *plant* hidroponik adalah motor pompa dengan catu daya sebesar 12V dengan debit maksimal hingga mencapai 3.5 liter/menit. Motor DC ini digunakan karena catu daya yang digunakan adalah 12V DC.
5. Sensor pH yang digunakan adalah sensor pH SKU SEN0161. Sensor ini digunakan karena dapat mengukur pH dari nilai 0 hingga 14.
6. Sensor jarak yang digunakan adalah sensor ultrasonik SRF05 untuk mengendalikan volume larutan nutrisi di tangki pembuangan agar dapat disalurkan kembali ke tangki pencampuran (tangki I). Sensor ini digunakan karena mampu mengukur jarak dari 0 hingga 400 cm.

7. Nutrisi dikontrol dengan menggunakan pH sebagai indikator. Sedangkan, pH sistem yang dikontrol pada nilai *setpoint* 6. Nilai ini digunakan karena pH ideal pada sistem hidroponik tomat adalah 5.5 hingga 6.5.
8. Jarak yang diukur sebagai batas tangki pembuangan larutan nutrisi adalah sebesar 10 cm dari dasar tangki di mana pompa akan menyala dan 2 cm dari dasar tangki di mana pompa akan mati. Hal ini dilakukan karena sifat pompa yang hanya bekerja di dalam air (*submersible*).
9. Volume air yang digunakan hingga 15 liter. Volume ini digunakan agar masing-masing tangki mampu menahan beban air yang diberikan sehingga tidak pecah.
10. *Error steady state* $\leq 5\%$. Jika nilai akhir sudah mencapai $\leq 5\%$ dari *setpoint* maka sistem sudah dapat digolongkan *steady state*.

3.3 Karakterisasi Setiap Blok

3.3.1 Karakterisasi Motor DC *Peristaltic Pump* 12V

Karakterisasi motor DC *Peristaltic Pump* 12V dilakukan untuk mengetahui karakter atau *gain* motor tersebut. Hal tersebut diperoleh dengan mengamati kecepatan putaran motor DC terhadap perubahan *duty cycle input* motor DC *peristaltic pump* 12V.

Peralatan yang digunakan terdiri atas:

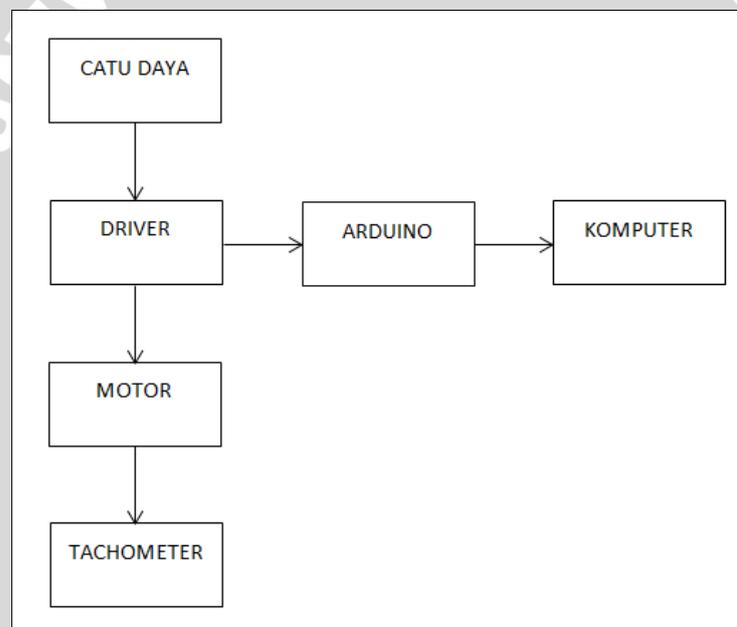
1. Motor DC *peristaltic pump* 12 Volt
2. *Driver motor (High Current 30 A DC Motor Driver)*
3. Komputer/Laptop
4. Arduino Mega 2560 (Mikrokontroler)
5. *Software* Arduino 1.6.7.
6. Tachometer digital DT-2235B
7. *Power supply* DC
8. Kabel penghubung

Langkah pengujian meliputi:

1. Merangkai *driver motor* ke Arduino Mega 2560 dengan cara menghubungkan PWM ke pin PWM arduino, *enable driver motor* ke pin digital arduino, *Vcc* ke pin 5V arduino, dan *GND* ke pin *GND* arduino.

2. Menghubungkan masukan sumber dari *power supply* DC ke masukan *driver* dan keluaran *driver* M+ ke (+) DC *peristaltic pump* 12V dan *driver* M- ke (-) DC *peristaltic pump* 12V.
3. Menghubungkan rangkaian *driver motor* dan arduino ke komputer/laptop.
4. Membuat program arduino yang sesuai dengan pengujian DC *peristaltic pump* 12V pada *software* Arduino 1.6.7.
5. Masukkan *input duty cycle* secara berurutan mulai dari 0% hingga 100% dan catat data kecepatan putaran motor yang keluar dari pembacaan tachometer digital DT-2235B.

Blok prosedur pengujian karakterisasi motor DC *peristaltic pump* 12V dapat dilihat pada Gambar 3.3.



Gambar 3.3 Blok Pengujian Motor DC *Peristaltic Pump* 12V.

Sumber: Perancangan

Data hasil pengujian kecepatan putaran motor DC (*rpm*) terhadap *duty cycle input* motor ditunjukkan dalam Tabel 3.1. Pada tabel tersebut, terdapat *duty cycle* sebagai *input driver*. Sedangkan, *output driver* berupa kecepatan putaran motor yang diukur dengan menggunakan tachometer digital.

Tabel 3.1 Hasil Pengujian Motor DC *Peristaltic Pump* 12V.

No.	Duty Cycle (%)	Kecepatan DC <i>Peristaltic Pump</i> (rpm)
1.	0	0
2.	1	0.1
3.	10	36.3
4.	20	77.8
5.	30	120.2
6.	40	159.8
7.	50	202.5
8.	60	244.1
9.	70	285.3
10.	80	342
11.	90	368.4
12.	95	402.9
13.	100	405.4

Grafik hasil pengujian kecepatan putaran motor DC (*rpm*) terhadap *duty cycle* input motor ditunjukkan dalam Gambar 3.4.



Gambar 3.4 Grafik Hasil Pengujian Motor DC *Peristaltic Pump* 12V.

Sumber: Perancangan

Dari grafik didapatkan *gain* motor DC *peristaltic pump* 12V adalah:

$$m = \frac{y_{0.5} - y_{0.4}}{x_{0.5} - x_{0.4}} = \frac{202.5 - 159.8}{50 - 40} = 4.27$$

3.3.2 Karakterisasi *Driver Motor (High Current 30 A DC Motor Driver)*

Karakterisasi *driver motor (High Current 30 A DC Motor Driver)* dilakukan untuk mengetahui karakter atau *gain* dari *driver motor* tersebut. Hal tersebut diperoleh dengan mengamati tegangan keluaran *driver motor* terhadap perubahan *duty cycle input driver motor*.

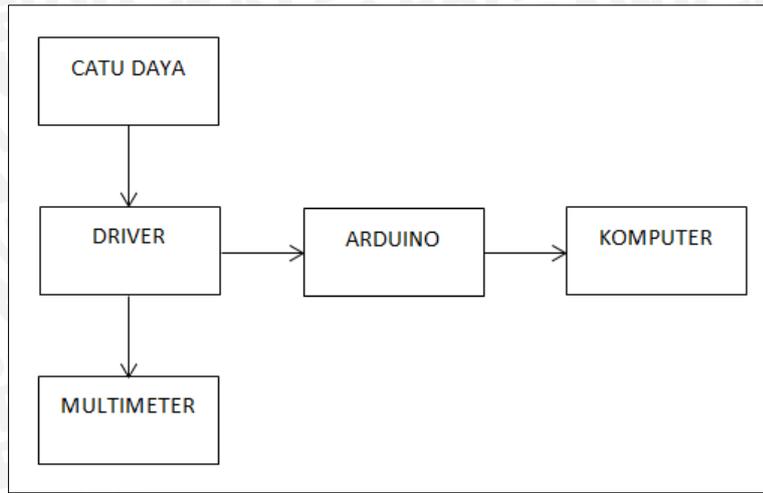
Peralatan yang digunakan terdiri atas:

1. *Driver motor (High Current 30 A DC Motor Driver)*
2. Komputer/laptop
3. Arduino Mega 2560 (Mikrokontroler)
4. *Software* Arduino 1.6.7.
5. Multimeter digital CD800a
6. *Power supply* DC
7. Kabel penghubung

Langkah pengujian meliputi:

1. Merangkai *driver motor* ke Arduino Mega 2560 dengan cara menghubungkan PWM ke pin PWM arduino, *enable driver motor* ke pin digital arduino, *Vcc* ke pin 5V arduino, dan *GND* ke pin *GND* arduino.
2. Menghubungkan masukan sumber dari *power supply* DC ke masukan *driver* dan keluaran *driver* M+ ke (+) multimeter CD800a dan *driver* M- ke (-) multimeter CD800a.
3. Menghubungkan rangkaian *driver motor* dan arduino ke komputer/laptop.
4. Membuat program arduino yang sesuai dengan pengujian *driver motor* pada *software* Arduino 1.6.7.
5. Masukkan *duty cycle* secara berurutan mulai dari 0% hingga 100% dan catat data tegangan yang keluar dari multimeter CD800a.

Blok prosedur pengujian karakterisasi *driver motor (High Current 30 A DC Driver Motor)* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Blok Pengujian *Driver Motor (High Current 30 A DC Motor Driver)*.

Sumber: Perancangan

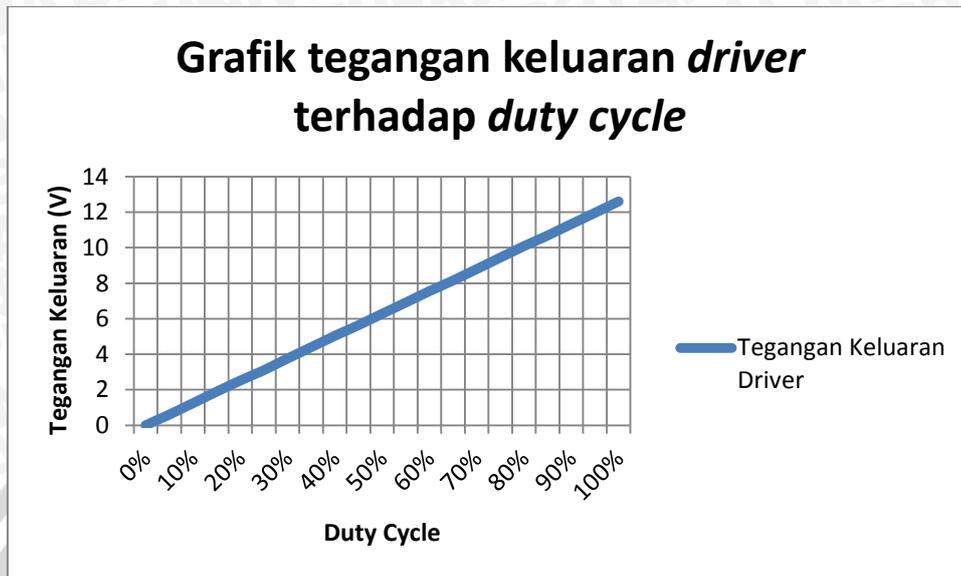
Data hasil pengujian tegangan keluaran *driver motor* (Volt) terhadap *duty cycle input driver motor* ditunjukkan dalam Tabel 3.2. Pada tabel tersebut terdapat *duty cycle* sebagai *input driver*. Sedangkan, *output driver* berupa tegangan yang diukur dengan menggunakan multimeter digital.

Tabel 3.2 Hasil Pengujian *Driver Motor (High Current 30 A DC Motor Driver)*.

No.	Duty Cycle (%)	Tegangan Keluaran Driver (V)	No.	Duty Cycle (%)	Tegangan Keluaran Driver (V)
1.	0	0	12.	55	6.91
2.	5	0.59	13.	60	7.55
3.	10	1.24	14.	65	8.15
4.	15	1.88	15.	70	8.79
5.	20	2.51	16.	75	9.44
6.	25	3.1	17.	80	10.08
7.	30	3.75	18.	85	10.68
8.	35	4.39	19.	90	11.32
9.	40	5.03	20.	95	11.96
10.	45	5.63	21.	100	12.61
11.	50	5.27	22.	-	-

Berdasarkan Tabel 3.2 dapat disimpulkan bahwa semakin besar nilai *duty cycle driver motor (High Current 30 A DC Motor Driver)* maka akan semakin besar pula nilai tegangan keluaran *driver motor (High Current 30 A DC Motor Driver)*.

Grafik hasil pengujian tegangan keluaran *driver motor* (volt) terhadap *duty cycle* input *driver motor* ditunjukkan dalam Gambar 3.6.



Gambar 3.6 Grafik hasil pengujian *Driver Motor* (*High Current 30 A DC Motor Driver*).

Sumber: Perancangan

Dari grafik tersebut didapatkan *gain driver motor* adalah:

$$m = \frac{y_{0.5} - y_{0.4}}{x_{0.5} - x_{0.4}} = \frac{5.27 - 5.03}{50 - 40} = 0.024$$

3.3.3 Karakterisasi Motor Pompa Mini DC 12V

Karakterisasi motor pompa mini DC 12V dilakukan untuk mengetahui karakter atau *gain* motor pompa tersebut. Hal tersebut diperoleh dengan mengamati debit keluaran motor pompa terhadap perubahan *duty cycle* input motor pompa mini DC 12V.

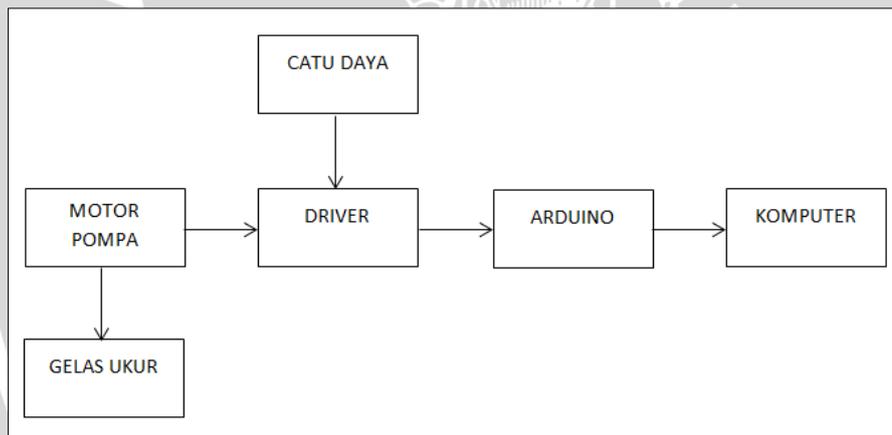
Peralatan yang digunakan terdiri atas:

1. Motor pompa mini DC 12 Volt
2. *Driver motor* (*High Current 30 A DC Motor Driver*)
3. Komputer/laptop
4. Arduino Mega 2560 (Mikrokontroller)
5. *Software* Arduino 1.6.7.
6. Gelas ukur
7. *Power supply* DC
8. Timer

Langkah pengujian meliputi:

1. Merangkai *driver motor* ke Arduino Mega 2560 dengan cara menghubungkan PWM ke pin PWM arduino, *enable driver motor* ke pin digital arduino, *Vcc* ke pin 5V arduino, dan *GND* ke pin *GND* arduino.
2. Menghubungkan masukan sumber dari *power supply* DC ke masukan *driver* dan keluaran *driver* M+ ke (+) motor pompa mini DC 12V dan *driver* M- ke (-) motor pompa mini DC 12V.
3. Menghubungkan rangkaian motor pompa dan arduino ke komputer/laptop.
4. Membuat program arduino yang sesuai dengan pengujian motor pompa mini DC 12V pada *software* Arduino 1.6.7.
5. Masukkan *duty cycle* secara berurutan mulai dari 0% hingga 100% dan catat data debit yang keluar dari pompa selama 1 menit.

Blok prosedur pengujian karakterisasi motor pompa mini DC 12V dapat dilihat pada Gambar 3.7.



Gambar 3.7 Blok Pengujian Motor Pompa Mini DC 12V.

Sumber: Perancangan

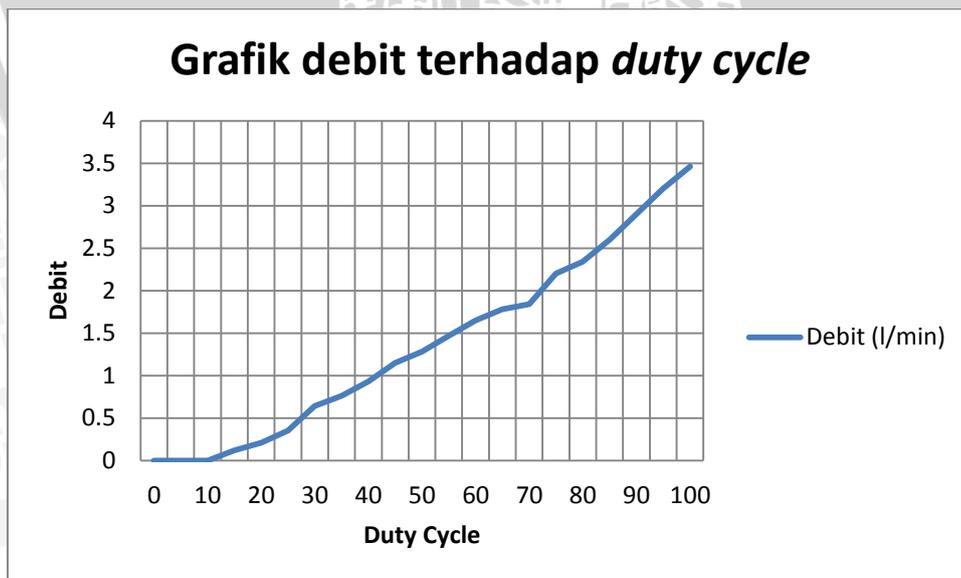
Data hasil pengujian debit keluaran motor pompa mini DC 12V (l/min) terhadap *duty cycle input* motor pompa mini DC 12V ditunjukkan dalam Tabel 3.3. Pada tabel tersebut terdapat *duty cycle* sebagai *input driver*. Sedangkan, *output driver* berupa debit yang diukur dengan menggunakan gelas ukur dan waktu yang dicatat menggunakan timer.

Tabel 3.3 Hasil Pengujian Motor Pompa Mini DC 12V.

No.	Duty Cycle (%)	Debit (l/min)	No.	Duty Cycle (%)	Debit (l/min)
1.	0	0	12.	55	1.47
2.	5	0	13.	60	1.65
3.	10	0	14.	65	1.78
4.	15	0.12	15.	70	1.84
5.	20	0.21	16.	75	2.2
6.	25	0.35	17.	80	2.34
7.	30	0.64	18.	85	2.6
8.	35	0.76	19.	90	2.9
9.	40	0.93	20.	95	3.2
10.	45	1.15	21.	100	3.46
11.	50	1.28	22.	-	-

Berdasarkan Tabel 3.3, dapat disimpulkan bahwa semakin besar nilai *duty cycle input* motor pompa mini DC 12V maka akan semakin besar pula nilai debit yang dikeluarkan. Pada nilai *input duty cycle* 0% hingga 10% memiliki nilai debit 0 l/min, pada data ini disebut dengan daerah *dead time*.

Grafik hasil pengujian debit motor pompa mini DC 12V (l/min) terhadap *duty cycle input* motor ditunjukkan dalam Gambar 3.8.



Gambar 3.8 Grafik Hasil Pengujian Motor Pompa Mini DC 12V.

Sumber: Perancangan

Dari grafik tersebut didapatkan *gain* motor pompa mini adalah:

$$m = \frac{y_{0.5} - y_{0.4}}{x_{0.5} - x_{0.4}} = \frac{1.28 - 0.93}{50 - 40} = 0.035$$

3.3.4 Karakterisasi Sensor pH SKU SEN0161

Karakterisasi sensor pH SKU SEN0161 dilakukan untuk mengetahui karakter sensor tersebut. Hal tersebut diperoleh dengan mengamati pH keluaran sensor pH SKU SEN0161 terhadap pH yang berasal dari pembacaan pH meter.

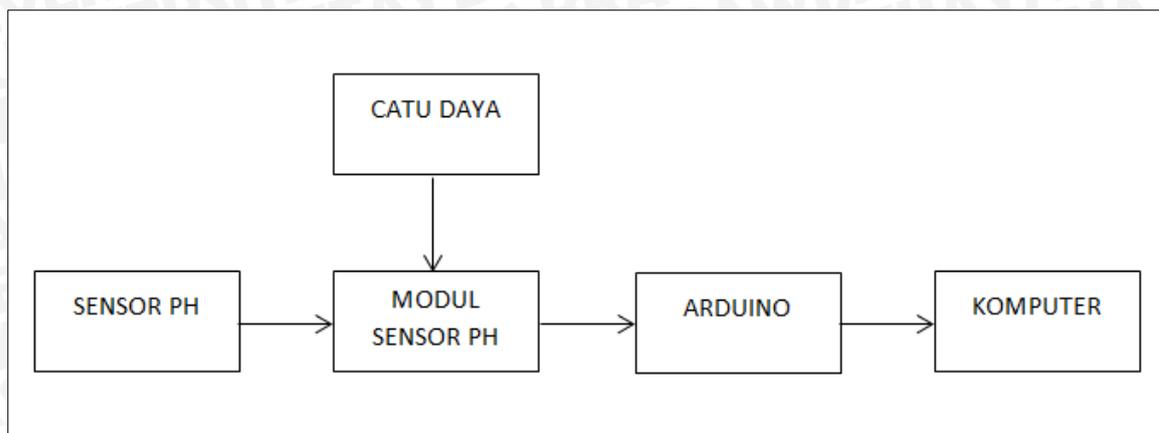
Peralatan yang digunakan terdiri atas:

1. Sensor pH SKU SEN0161
2. pH meter
3. Komputer/laptop
4. Arduino Mega 2560 (Mikrokontroler)
5. *Software* Arduino 1.6.7.
6. Larutan pH *buffer*
7. Kabel penghubung

Langkah pengujian meliputi:

1. Merangkai sensor pH SKU SEN0161 ke Arduino Mega 2560 dengan cara menghubungkan V+ ke pin 5V pada arduino, GND ke pin GND pada arduino, dan *output* sensor pH ke pin analog arduino.
2. Merangkai rangkaian sensor pH dan arduino ke laptop/komputer.
3. Membuat program yang sesuai dengan pembacaan sensor pH dengan menggunakan *software* Arduino 1.6.7.
4. Membuat larutan dengan menggunakan larutan pH *buffer* pada wadah yang sudah tersedia.
5. Memasukkan ujung sensor pH ke dalam wadah dengan larutan pH *buffer*.
6. Mencatat data pembacaan antara pH meter dan sensor pH SKU SEN0161.

Blok prosedur pengujian karakterisasi motor pompa mini DC 12V dapat dilihat pada Gambar 3.9.



Gambar 3.9 Blok Pengujian Sensor pH SKU SEN0161.

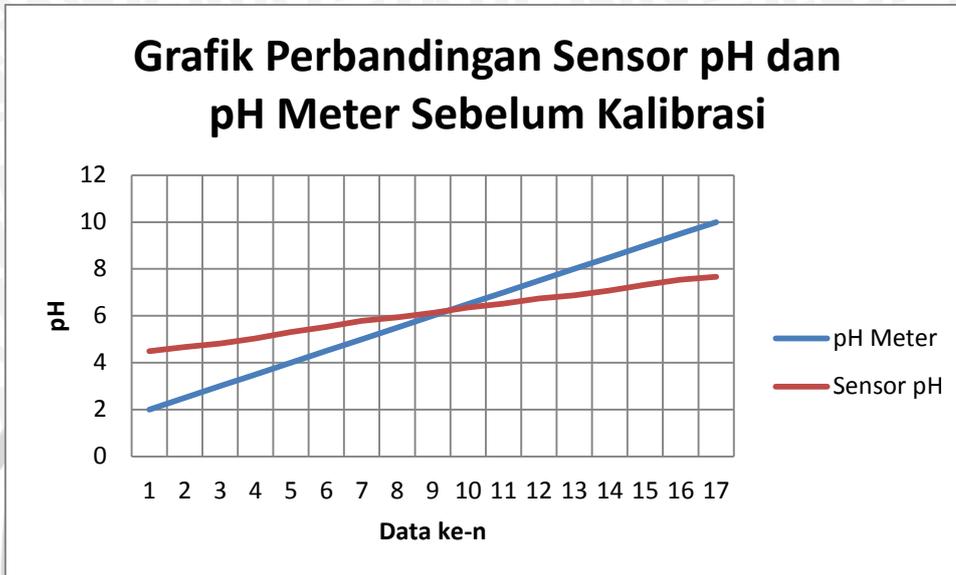
Sumber: Perancangan

Data hasil pengujian pembacaan sensor pH SKU SEN0161 terhadap pembacaan pH oleh pH meter ditunjukkan dalam Tabel 3.4. Pada tabel tersebut terdapat pH yang merupakan pembacaan data dari pH meter sebagai acuan. Sedangkan, terdapat pula pH hasil dari pembacaan dari sensor SKU SEN0161. Kemudian *error* yang dihasilkan merupakan perbandingan dari kedua pembacaan tersebut.

Tabel 3.4 Hasil Pengujian Sensor pH SKU SEN0161.

No.	pH Meter	Sensor pH SKU SEN0161	Error(%)
1.	2	4.49	124.5
2.	2.5	4.67	86.8
3.	3	4.81	60.34
4.	3.5	5.04	44
5.	4	5.31	32.75
6.	4.5	5.52	22.67
7.	5	5.78	15.6
8.	5.5	5.93	7.81
9.	6	6.13	2.16
10.	6.5	6.35	2.3
11.	7	6.52	6.85
12.	7.5	6.74	10.13
13.	8	6.87	14.12
14.	8.5	7.08	16.7
15.	9	7.33	18.56
16.	9.5	7.54	20.63
17.	10	7.67	23.3
Rata-rata			29.95

Berdasarkan hasil pengujian dalam Tabel 3.4, rata-rata *error* antara pH pembacaan dari pH meter dengan pembacaan pH oleh sensor pH SKU SEN0161 sebesar 29.95%. Grafik perbandingan pH hasil dari pembacaan pH meter dengan pembacaan pH oleh sensor pH SKU SEN0161 dapat dilihat dalam Gambar 3.10.



Gambar 3.10 Grafik Hasil Pengujian Sensor pH SKU SEN0161.

Sumber: Perancangan

Pada grafik tersebut terlihat bahwa selisih yang timbul antara kedua pembacaan pH cenderung membentuk garis linear, sehingga untuk mengurangi *error* pembacaan pH dapat menggunakan cara kalibrasi. Rumus umum untuk kalibrasi terdapat dalam persamaan (3.1) dan persamaan (3.2), kemudian hasil akhir rumus kalibrasi ditunjukkan oleh persamaan (3.3).

Rumus kalibrasi:

$$\sum x_i a_0 + \sum x_i^2 a_1 = \sum x_i y_i \tag{3.1}$$

$$f = a_0 + a_1 x \tag{3.2}$$

$$f = -9.043148957 + 2.464188979 x \tag{3.3}$$

Rumus kalibrasi tersebut dimasukkan ke dalam program untuk pembacaan sensor pH SKU SEN0161 agar mengurangi *error* pembacaan pH, sehingga akan menghasilkan data yang terdapat dalam Tabel 3.5. berikut:

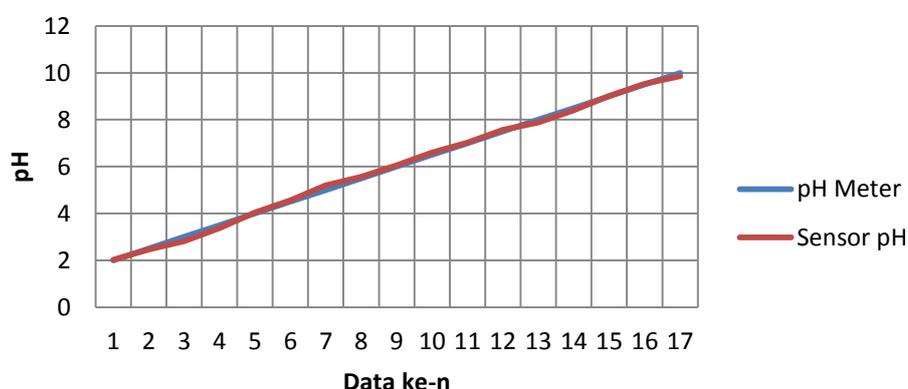
Tabel 3.5 Hasil Pengujian Sensor pH SKU SEN0161 Setelah Kalibrasi.

No.	pH Meter	Sensor pH SKU SEN0161	Error(%)
1.	2	2.02	1.05
2.	2.5	2.46	1.41
3.	3	2.8	6.34
4.	3.5	3.37	3.53
5.	4	4.04	1.04
6.	4.5	4.55	1.31
7.	5	5.19	3.99
8.	5.5	5.56	1.26
9.	6	6.06	1.03
10.	6.5	6.6	1.6
11.	7	7.02	0.33
12.	7.5	7.56	0.87
13.	8	7.89	1.42
14.	8.5	8.4	1.13
15.	9	9.01	0.21
16.	9.5	9.53	0.38
17.	10	9.85	1.42
Rata-rata			1.67

Berdasarkan hasil pengujian dalam Tabel 3.5, rata-rata *error* antara pH pembacaan dari pH meter dengan pembacaan pH oleh sensor pH SKU SEN0161 setelah kalibrasi sebesar 1.67%. Grafik perbandingan pH hasil dari pembacaan pH meter dengan pembacaan pH oleh sensor pH SKU SEN0161 setelah kalibrasi dapat dilihat dalam Gambar 3.11.

Error yang didapatkan haruslah sangat kecil, dikarenakan hasil pembacaan antara pH meter dengan hasil pembacaan dari sensor pH SKU SEN0161 tidak jauh berbeda. Dalam sistem ini membutuhkan umpan balik yang akurat dengan kepekaan yang tidak relatif tinggi, karena sistem akan dapat lebih stabil.

Grafik Perbandingan Sensor pH dan pH Meter Sesudah Kalibrasi



Gambar 3.11 Grafik Hasil Pengujian Sensor pH SKU SEN0161Setelah Kalibrasi.

Sumber: Perancangan

Dari grafik tersebut didapatkan *gain* sensor pH SKU SEN0161 adalah:

$$m = \frac{y_{0.5} - y_{0.4}}{x_{0.5} - x_{0.4}} = \frac{5.07 - 3.96}{5 - 4} = 1.11$$

3.3.5 Karakterisasi Sensor Ultrasonik SRF05

Karakterisasi sensor ultrasonik SRF05 dilakukan untuk mengetahui karakter sensor tersebut. Hal tersebut diperoleh dengan mengamati jarak keluaran sensor ultrasonik SRF05 terhadap jarak yang berasal dari pembacaan penggaris.

Peralatan yang digunakan terdiri atas:

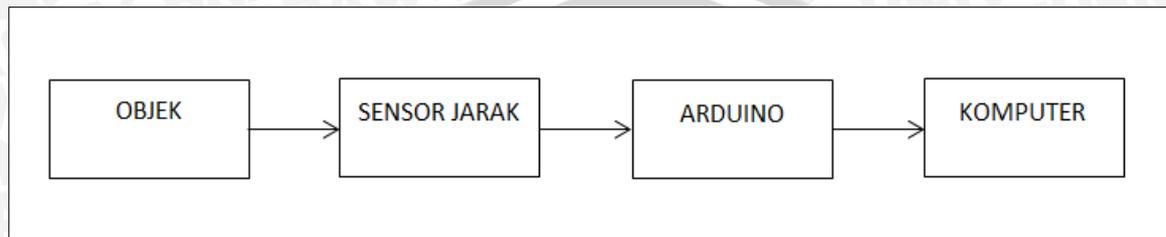
1. Sensor ultrasonik SRF05
2. Penggaris
3. Komputer/laptop
4. Arduino Mega 2560 (Mikrokontroler)
5. *Software* Arduino 1.6.7.
6. Benda penghalang
7. Kabel penghubung

Langkah pengujian meliputi:

1. Merangkai sensor ultrasonik SRF05 ke Arduino Mega 2560 dengan cara menghubungkan V+ ke pin 5V pada arduino, GND ke pin GND pada arduino, dan *output* sensor ultrasonik yakni *trig* dan *echo* ke pin digital arduino.
2. Merangkai rangkaian sensor ultrasonik dan arduino ke laptop/komputer.

3. Membuat program yang sesuai dengan pembacaan sensor ultrasonik SRF05 dengan menggunakan *software* Arduino 1.6.7.
4. Memberikan benda penghalang sebagai parameter untuk mendapatkan data.
5. Membandingkan data dengan menggunakan penggaris kemudian dicatat.

Blok prosedur pengujian karakterisasi sensor ultrasonik SRF05 dapat dilihat pada Gambar 3.12.



. Gambar 3.12 Blok Pengujian Sensor Ultrasonik SRF05.

Sumber: Perancangan

Data hasil pengujian sensor ultrasonik SRF05 terhadap pembacaan jarak dibandingkan dengan pembacaan jarak oleh penggaris ditunjukkan dalam Tabel 3.6. Pada tabel tersebut terdapat jarak yang merupakan pembacaan data dari penggaris sebagai acuan. Sedangkan, terdapat pula jarak hasil dari pembacaan dari sensor ultrasonik SRF05. Kemudian *error* yang dihasilkan yang merupakan perbandingan dari kedua pembacaan tersebut. Benda tersebut diletakkan di depan sensor sehingga jarak antara benda dengan sensor menjadi masukan data dan dibandingkan dengan pengukuran menggunakan penggaris secara manual.

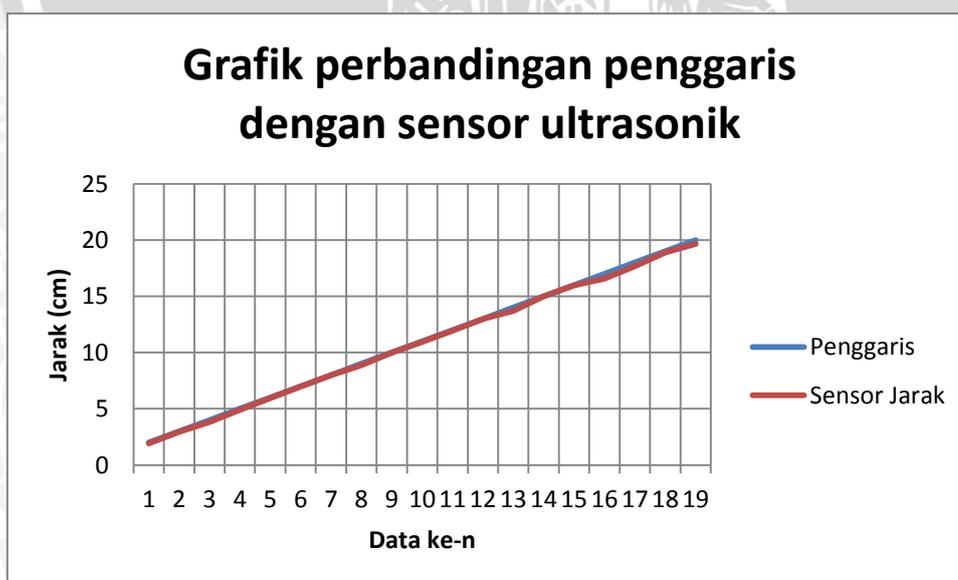
Dari percobaan membandingkan ini, kita dapat melihat keakuratan dari sensor ultrasonik dalam pembacaan jarak. Sehingga, setelah diaplikasikan di dalam sistem diharapkan akan bekerja sesuai dengan program yang telah dibuat. Sensor ini digunakan di dalam tangki II sebagai umpan-balik kontroler *on-off* untuk mengatur volume sisa larutan nutrisi yang tidak diserap oleh tanaman pada *plant* sistem hidroponik.

Objek yang digunakan merupakan objek yang dapat dijadikan sebagai penghalang sinyal dari sensor. Sehingga, sensor mengetahui jarak antara sensor dengan jarak objek yang ditempatkan sebagai penghalang.

Tabel 3.6 Hasil Pengujian Sensor Ultrasonik SRF05.

No.	Penggaris (cm)	Sensor Ultrasonik SRF05 (cm)	Error (%)
1.	2	1.93	3.5
2.	3	2.95	1.67
3.	4	3.83	4.25
4.	5	4.91	1.8
5.	6	5.95	0.83
6.	7	6.98	0.28
7.	8	8	0
8.	9	8.86	1.56
9.	10	9.97	0.3
10.	11	10.95	0.45
11.	12	11.93	0.58
12.	13	13	0
13.	14	13.71	2.07
14.	15	15	0
15.	16	15.98	0.12
16.	17	16.57	2.52
17.	18	17.67	1.83
18.	19	18.90	0.52
19.	20	19.67	1.65
Rata-rata			1.26

Berdasarkan Tabel 3.6 dapat disimpulkan bahwa semakin besar nilai jarak pembacaan oleh penggaris maka akan semakin besar pula nilai jarak pembacaan sensor ultrasonik SRF05. Grafik perbandingan jarak hasil dari pembacaan penggaris dengan pembacaan jarak oleh sensor ultrasonik SRF05 dapat dilihat dalam Gambar 3.13.



Gambar 3.13 Grafik Hasil Pengujian Sensor Ultrasonik SRF05.
Sumber: Perancangan

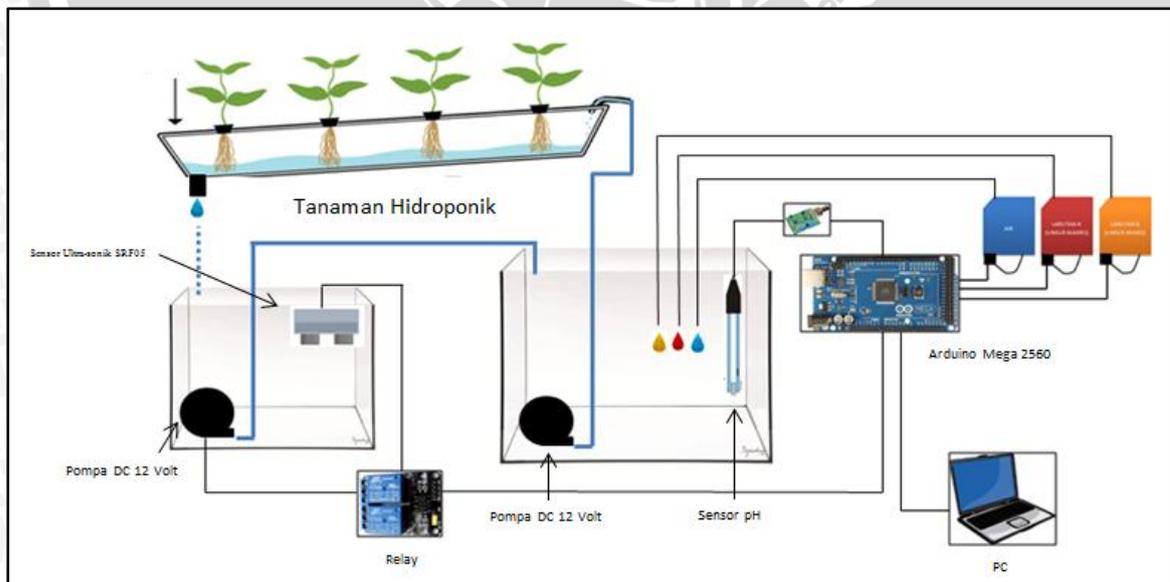
Dari grafik tersebut didapatkan *gain* sensor ultrasonik SRF 05 adalah:

$$m = \frac{y_{0.6} - y_{0.5}}{x_{0.6} - x_{0.5}} = \frac{5.95 - 4.91}{6 - 5} = 1.04$$

3.4 Pembuatan Perangkat Keras

3.4.1 Desain Media Hidroponik

Plant yang digunakan dalam penelitian ini adalah sebuah sistem hidroponik. Oleh karena itu, untuk sistem hidroponik yang digunakan menggunakan metode *Nutrient Film Technique* (NFT). Media hidroponik terdiri dari 2 buah *box* dan 1 pipa paralon dengan desain sendiri. *Box* dibagi menjadi 2 bagian, yaitu tangki I dan tangki II. Tangki I merupakan tangki pencampuran nutrisi berbahan *acrylic* dengan ukuran 40 cm x 30 cm x 10 cm. Tangki I dapat diisi air hingga volume maksimal 12 L, dengan ketebalan 3 mm. Di dalam Tangki I terdapat motor pompa mini DC 12V yang berfungsi untuk menyalurkan larutan nutrisi ke *plant* media hidroponik. *Plant* media hidroponik berbentuk pipa paralon dengan ketinggian 35 cm dan memiliki kemiringan 5 cm. Pipa paralon diberi lubang tanam dengan diameter yang sesuai dengan pot tanam hidroponik. *Plant* ini menghubungkan Tangki I dengan Tangki II. Tangki II merupakan tangki sisa larutan nutrisi berbahan *acrylic* dengan ukuran 40 cm x 30 cm x 25 cm. Tangki II dapat diisi air hingga volume maksimal 30 L, dengan ketebalan 3 mm.



Gambar 3.14 Gambar Desain Media Hidroponik.

Sumber: Perancangan

3.4.2 Prinsip Kerja Alat

Prinsip kerja alat dalam Gambar 3.14 adalah sebagai berikut:

1. Sistem diberi dengan catu daya 12V untuk digunakan mencatu motor *peristaltic pump* 12V DC dan motor pompa mini DC 12V. Sedangkan, catu daya 5V untuk digunakan mencatu rangkaian sensor pH, rangkaian sensor ultrasonik, *relay*, dan rangkaian *driver motor*.
2. Catu daya Arduino Mega 2560 didapat dari laptop/komputer dengan menghubungkan *port* komunikasi antara arduino dengan komputer menggunakan kabel USB dengan kecepatan transfer data sebesar 9600 *bit per second* (bps).
3. Kadar larutan nutrisi menggunakan kadar keasaman (pH) yang diukur dengan menggunakan sensor pH.
4. Volume larutan nutrisi sisa diukur dengan menggunakan sensor ultrasonik.
5. Sensor pH dihubungkan dengan modul rangkaian penguat karena tegangan keluaran sensor sangat kecil sehingga harus diperkuat agar dapat dibaca oleh arduino dengan rentang 0V hingga 5V.
6. Sinyal keluaran dari modul rangkaian penguat ini digunakan sebagai masukan arduino yang kemudian diproses menggunakan kontroler PID.
7. Keluaran arduino berupa sinyal *pulse width modulation* (PWM) diberikan kepada rangkaian *driver motor*. Rangkaian ini memiliki fungsi untuk menguatkan sinyal PWM arduino dengan tegangan sebesar 0V hingga 5V menjadi 0V hingga 12V.
8. Terdapat 3 pompa untuk mengatur kondisi larutan nutrisi sesuai dengan yang diinginkan. Dimana 2 pompa berjalan dalam satu kondisi dan 1 pompa dengan kondisi lainnya.
9. Pompa makro dan mikro akan mengalirkan larutan makro dan larutan mikro dengan konsentrasi yang sama jika kadar keasaman (pH) larutan nutrisi hidroponik terlalu basa atau $\text{pH} > 6$.
10. Pompa air akan mengalirkan larutan air jika kadar keasaman (pH) larutan nutrisi hidroponik terlalu asam atau $\text{pH} < 6$.
11. Jika pH sudah sesuai dengan parameter yang diinginkan maka pompa mini DC 12V Tangki I akan menyala dan mengalirkan larutan nutrisi ke dalam *plant media* hidroponik.
12. Larutan nutrisi akan jatuh ke dalam Tangki II.

13. Pompa mini DC 12V Tangki II akan menyala apabila volume larutan sisa nutrisi sudah mencapai batas yang telah ditentukan dan mengalirkan larutan nutrisi Tangki II kembali ke dalam Tangki I.

3.4.3 Rangkaian Catu Daya

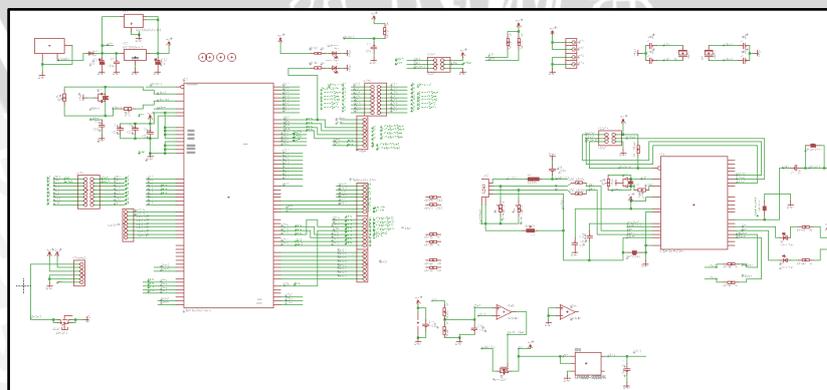
Rangkaian catu daya yang digunakan adalah *switching power supply* yang memiliki tegangan keluaran sebesar 12V. Rangkaian *switching power supply* terdiri dari rangkaian penyearah gelombang dan penguat arus. Rangkaian penyearah menggunakan *diode* yang berguna untuk mengkonversikan tegangan arus bolak-balik (AC) menjadi tegangan arus searah (DC). Rangkaian penguat arus menggunakan transistor. Catu daya 12V untuk motor pompa DC diambil dari *switching power supply* 12V dengan arus maksimal sebesar 30 Ampere. *Switching power supply* ditunjukkan dalam Gambar 3.15.



Gambar 3.15 *Switching Power Supply*.

3.4.4 Konfigurasi Pin Arduino Mega 2560

Pada alat ini digunakan mikrokontroler Arduino Mega 2560 sebagai pusat utama dalam proses pengendalian sistem. Konfigurasi I/O ditunjukkan dalam Gambar 3.16.



Gambar 3.16 Skematik Konfigurasi I/O Arduino Mega 2560.

Sumber: datasheet Arduino Mega 2560

Tabel masing-masing kegunaan pin Arduino Mega 2560 dalam sistem ini dapat ditunjukkan dalam Tabel 3.7.

Tabel 3.7 Konfigurasi Pin Arduino Mega 2560 dalam Sistem.

No.	Pin	Fungsi
1.	Vin	Sebagai tegangan masukan untuk mencatu daya alat
2.	Gnd	Sebagai <i>ground</i> untuk mencatu daya alat
3.	2	Sebagai masukan digital <i>echo</i> dari sensor ultrasonik SRF05
4.	3	Sebagai masukan digital <i>trigger</i> dari sensor ultrasonik SRF05
5.	4	Sebagai masukan digital R_PWM Pompa 1
6.	5	Sebagai masukan digital L_PWM Pompa 1
7.	6	Sebagai masukan digital R_PWM Pompa 2
8.	7	Sebagai masukan digital L_PWM Pompa 2
9.	8	Sebagai masukan digital R_PWM Peristaltik Air
10.	9	Sebagai masukan digital L_PWM Peristaltik Air
11.	10	Sebagai masukan digital R_PWM Peristaltik Mikro
12.	11	Sebagai masukan digital L_PWM Peristaltik Mikro
13.	12	Sebagai masukan digital R_PWM Peristaltik Makro
14.	13	Sebagai masukan digital L_PWM Peristaltik Makro
15.	24	Sebagai masukan digital R_EN Pompa 1
16.	25	Sebagai masukan digital L_EN Pompa 1
17.	26	Sebagai masukan digital R_EN Pompa 2
18.	27	Sebagai masukan digital L_EN Pompa 2
19.	28	Sebagai masukan digital R_EN Peristaltik Air
20.	29	Sebagai masukan digital L_EN Peristaltik Air
21.	30	Sebagai masukan digital R_EN Peristaltik Mikro
22.	31	Sebagai masukan digital L_EN Peristaltik Mikro
23.	32	Sebagai masukan digital R_EN Peristaltik Makro
24.	33	Sebagai masukan digital L_EN Peristaltik Makro
25.	A0	Sebagai masukan digital/analog dari Output Sensor pH Modul
26.	36	Sebagai masukan digital Relay 2 (Pin 4)
27.	37	Sebagai masukan digital Relay 1 (Pin 3)
28.	20	Sebagai masukan digital SDA dari LCD Display Module
29.	21	Sebagai masukan digital SCL dari LCD Display Module

3.4.5 Konfigurasi Pin Driver Motor

Rangkaian *driver motor* di dalam sistem terdiri dari 2 bagian, yakni bagian *driver motor* untuk mengendalikan larutan nutrisi dan juga bagian *driver motor* untuk mengendalikan pompa penyalur. Untuk bagian *driver motor* yang mengendalikan larutan nutrisi terdiri dari rangkaian untuk mengatur larutan makro, larutan mikro, dan air. Sedangkan, untuk bagian *driver motor* yang mengendalikan pompa penyalur dihubungkan ke *relay* untuk menggerakkan pompa yang mengalirkan larutan nutrisi dari Tangki I ke dalam *plant* dan larutan nutrisi dari Tangki II ke dalam Tangki I. Rangkaian *driver motor* yang digunakan adalah *driver motor H-Bridge* BTS7960 30A. Mode pengontrolan yang digunakan adalah mode sinyal *pulse width modulation* (PWM) dan dicatu dengan catu daya eksternal berupa *switching power supply* 12V. Tabel konfigurasi pin dari *driver motor* tersebut dapat dilihat dalam Tabel 3.8.

Tabel 3.8 Konfigurasi pin *Driver Motor H-Bridge* BTS7960 30A dalam Sistem.

No.	Pin	Fungsi
1.	Vin	Sebagai tegangan masukan untuk mencatu daya alat
2.	Gnd	Sebagai ground untuk mencatu daya alat
3.	D13	Sebagai R_PWM Makro
4.	D12	Sebagai L_PWM Makro
5.	D11	Sebagai R_PWM Mikro
6.	D10	Sebagai L_PWM Mikro
7.	D9	Sebagai R_PWM Air
8.	D8	Sebagai L_PWM Air
9.	D7	Sebagai R_PWM Pompa1
10.	D6	Sebagai L_PWM Pompa1
11.	D5	Sebagai R_PWM Pompa2
12.	D4	Sebagai L_PWM Pompa2
13.	D24	Sebagai masukan R_EN Makro
14.	D25	Sebagai masukan L_EN Makro
15.	D26	Sebagai masukan R_EN Mikro
16.	D27	Sebagai masukan L_EN Mikro
17.	D28	Sebagai masukan R_EN Air
18.	D29	Sebagai masukan L_EN Air
19.	D30	Sebagai masukan R_EN Pompa1
20.	D31	Sebagai masukan L_EN Pompa1
21.	D32	Sebagai masukan R_EN Pompa2
22.	D33	Sebagai masukan L_EN Pompa2

3.4.6 Konfigurasi Pin Relay

Relay yang digunakan adalah *relay* dengan kapasitas tegangan 12V. *Relay* digunakan untuk mengaktifkan pompa dalam Tangki I untuk dialirkan ke dalam *plant* media hidroponik. Pompa tersebut akan aktif apabila kondisi pH dalam Tangki I berada pada nilai $6 \leq \text{pH} \leq 6.09$ dan akan non-aktif apabila kondisi pH dalam Tangki I berada pada nilai $0 < \text{pH} \leq 5.99$ dan pada nilai $6.1 \leq \text{pH} < 14$. *Relay* juga digunakan untuk mengaktifkan pompa dalam Tangki II untuk dialirkan kembali ke dalam Tangki I. Pompa tersebut akan aktif apabila kondisi volume larutan nutrisi berada pada jarak $\geq 10\text{cm}$ dan akan non-aktif apabila kondisi volume larutan nutrisi berada pada jarak $17\text{cm} \leq \text{jarak} \leq 18\text{cm}$. Konfigurasi pin *Relay* dapat ditunjukkan dalam Tabel 3.9 berikut.

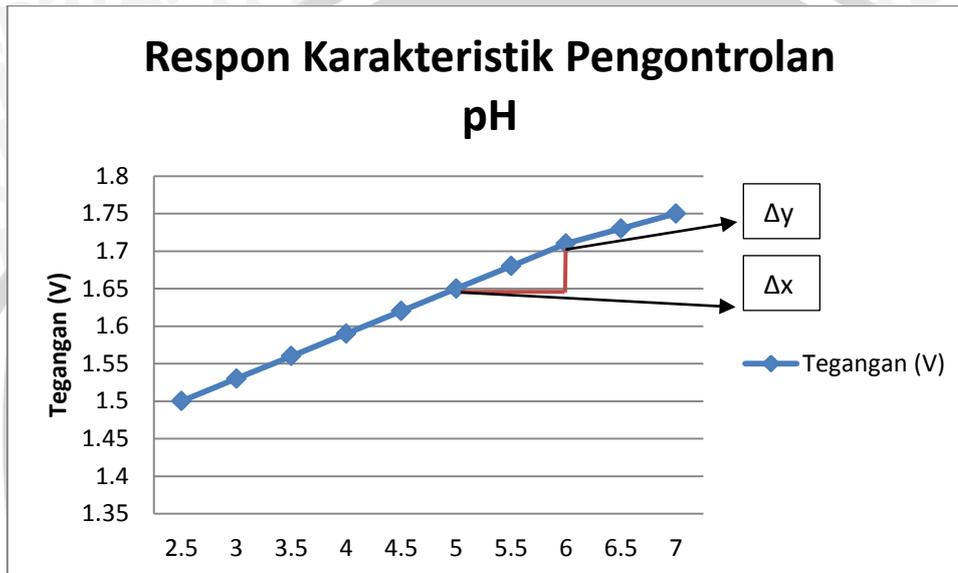
Tabel 3.9 Konfigurasi pin *Relay* dalam Sistem.

No.	Pin	Fungsi
1.	Vin	Sebagai tegangan masukan untuk mencatu daya alat
2.	Gnd	Sebagai ground untuk mencatu daya alat
3.	D34	Sebagai masukan Relay 1
4.	D35	Sebagai masukan Relay 2

3.5 Perancangan Algoritma

3.5.1 Penentuan Karakteristik Pengontrolan pH

Untuk menemukan fungsi alih plant agar mendapatkan karakteristik *statis gain* keadaan mantap dengan cara diberikan unit step masukan pH PDAM hingga mencapai *setpoint* yang diinginkan kemudian dilihat bagaimana keadaan mantapnya. Untuk menemukan parameter kontroler PID digunakan metode *ziegler-nichols* orde 1, dimana seperti pada Gambar 3.17 berikut ini.



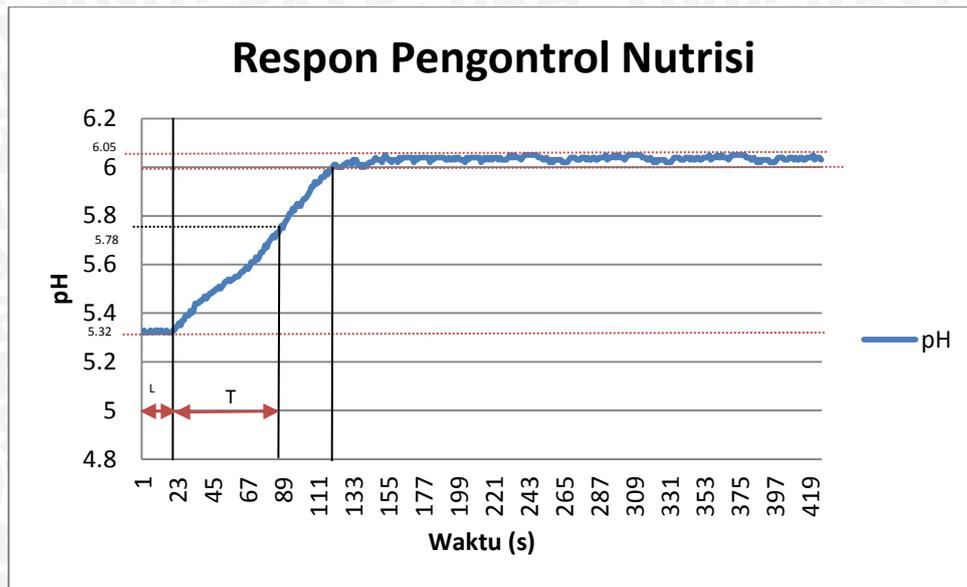
Gambar 3.17 Respon Karakteristik Pengontrolan pH.
Sumber: Perancangan

$$k = \frac{\Delta y}{\Delta u} = \frac{1.71 - 1.65}{6 - 5} = 0.06 \quad (3.4)$$

3.5.2 Parameter Kontroler PID dengan Metode Ziegler-Nichols

Dalam memenuhi tujuan performansi *loop* yang diinginkan, maka perlu ditambahkan sebuah kontroler di dalam sistem. Sedangkan, di dalam pengujian ini digunakan kontroler *proporsional*, *integral*, dan *differensial* (PID). Kontroler ini dibutuhkan dikarenakan karakteristik *plant* membutuhkan kontroler yang cepat dengan meminimalkan *overshoot*.

Langkah awal *tuning* parameter dengan menggunakan metode *ziegler-nichols* orde 1 adalah dengan menarik garis *tangent* pada titik infeksi grafik karakteristik *plant*. Kemudian mencari perpotongan garis *tangent* dengan garis nilai akhir, sehingga diperoleh nilai L (Keterlambatan transportasi proses) dan T (Konstanta waktu). Nilai L dan T tersebut nantinya akan digunakan untuk mencari parameter Kp, Ki, dan Kd.



Gambar 3.18 Metode Ziegler-Nichols Orde 1 pada *Plant*.
Sumber: Perancangan

Dari pengujian tersebut dapat dicari nilai parameter L dan juga T. Dalam grafik dapat diketahui bahwa nilai pH awal adalah 5.32 dan nilai pH akhir adalah 6.05. Untuk mencari nilai T dapat dilakukan dalam perhitungan sebagai berikut:

$$\Delta pH = 6.05 - 5.32 = 0.73$$

Sedangkan, nilai T ditentukan dari 63.2% nilai ΔpH yang didapatkan dari perhitungan sebagai berikut:

$$\begin{aligned} 63.2\% \Delta pH &= 5.32 + (63.2\% (6.05 - 5.32)) \\ &= 5.32 + 0.46136 \\ &= 5.78 \end{aligned}$$

Nilai T pada pH 5.78 = 91 detik.

Sehingga, pada saat 63.2% ΔpH , nilai T = 91 - 20 = 71 detik.

L = 20 detik dan T = 71 detik.

Untuk melakukan *tuning* parameter dari kontroler PID menggunakan rumus yang ada pada Tabel 3.10.

Tabel 3.10 Aturan Penalaan *Ziegler-Nichols* Berdasarkan Respon Unit Step Dari *Plant*

Type Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5L$

Berdasarkan pengujian kurva respon step berbentuk s diatas. Sehingga dapat diperoleh persamaan matematis fungsi alih sistem adalah sebagai berikut:

$$Y(s) = \frac{k}{Ts+1} \cdot e^{-sL} \quad (3.5)$$

$$= \frac{0.06}{71s+1} \cdot e^{-20s}$$

Setelah mendapatkan nilai fungsi alih sistem, kemudian dengan menggunakan rumus yang ada pada Tabel 3.10 maka akan didapatkan nilai parameter K_p , K_i dan K_d adalah sebagai berikut:

$$K_p = 1.2 \frac{T}{L} = 1.2 \times \frac{71}{20} = 4.26 \quad (3.6)$$

$$T_i = 2L = 2 \times 20 = 40 \quad (3.7)$$

$$T_d = 0.5L = 0.5 \times 20 = 10 \quad (3.8)$$

$$K_p = K_i \times T_i \quad K_i = \frac{K_p}{T_i} = \frac{4.26}{40} = 0.1065 \quad (3.9)$$

$$K_p = \frac{K_d}{T_d} \quad K_d = K_p \times T_d = 4.26 \times 10 = 42.6 \quad (3.10)$$

Dari perhitungan diatas didapatkan nilai parameter $K_p = 4.26$, $K_i = 0.1065$, dan $K_d = 42.6$. Sehingga akan mendapatkan persamaan transformasi *Laplace* berdasarkan dengan nilai K_p , K_i , dan K_d adalah sebagai berikut:



$$M(s) = [K_p + \frac{K_i}{s} + K_d s] E(s) \quad (3.11)$$

$$M(s) = [K_p s + K_i + K_d s^2] E(s) \quad (3.12)$$

3.5.3 Diskritisasi

Persamaan diatas belum dapat dimasukkan ke dalam mikrokontroler oleh karena itu persamaan kontinyu tersebut harus diubah ke dalam bentuk diskrit melalui Transformasi Z. Dalam transformasi Z dibutuhkan waktu *sampling* (T_s). Pada sistem ini menggunakan metode *Backward Difference* dikarenakan metode ini memiliki sensitifitas yang lebih rendah, sehingga sistem tidak akan mengalami perubahan atau osilasi saat diberi gangguan dan akan tetap mempertahankan performansinya. Dengan menggunakan metode *Backward Difference*, yaitu dengan mengganti s dalam persamaan analog bentuk s persamaan. Persamaan s tersebut setara dengan bentuk *Laplace*:

$$s = \frac{1-z^{-1}}{T_s} \quad (3.12)$$

Maka persamaan 3.12 disubstitusikan ke dalam persamaan 3.11 akan menjadi:

$$\begin{aligned} M(z) &= [K_p + \frac{K_i}{\frac{1-z^{-1}}{T_s}} + K_d (\frac{1-z^{-1}}{T_s})] E(z) \\ &= [K_p + \frac{K_i T_s}{1-z^{-1}} + \frac{K_d}{T_s} (1-z^{-1})] E(z) \end{aligned} \quad (3.13)$$

Kemudian didapatkan rumus masing-masing parameter kontroler PID, adalah sebagai berikut:

$$\text{Kontroler Proporsional: } C_p(z) = K_p \cdot E(z) \quad (3.14)$$

$$\begin{aligned} \text{Kontroler Integral: } C_i(z) &= \frac{K_i T_s}{1-z^{-1}} \cdot E(z) \\ &= [C_i(z) z^{-1} + K_i T_s \cdot E(z)] \end{aligned} \quad (3.15)$$

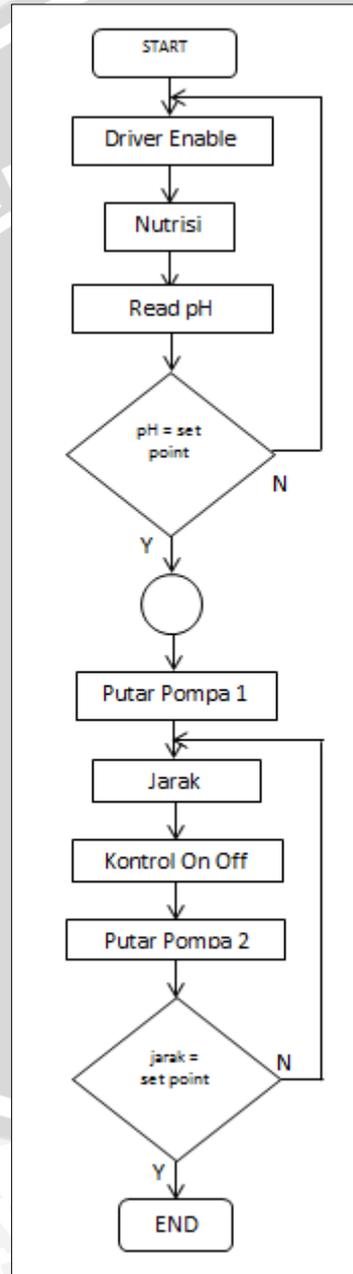
$$\begin{aligned} \text{Kontroler Diferensial: } C_d(z) &= \frac{K_d}{T_s} (1-z^{-1}) \cdot E(z) \\ &= \frac{K_d}{T_s} [E(z) - E(z) z^{-1}] \end{aligned} \quad (3.16)$$

Persamaan 3.14 hingga persamaan 3.16 kemudian diubah ke dalam persamaan beda sehingga akan didapatkan persamaan 3.17 seperti di bawah ini:

$$M(k) = K_p.E(k) + (C_i(k-1) + K_i.T_s E(k)) + \left(\frac{K_d}{T_s}(E(k) - E(k-1))\right) \quad (3.17)$$

Dimana (k-1) adalah kondisi sebelumnya. Persamaan diatas lalu dimasukkan ke dalam program mikrokontroler.

3.5.4 Flowchart Program



Gambar 3.19 Flowchart Program Keseluruhan Sistem.

Sumber: Perancangan

BAB IV

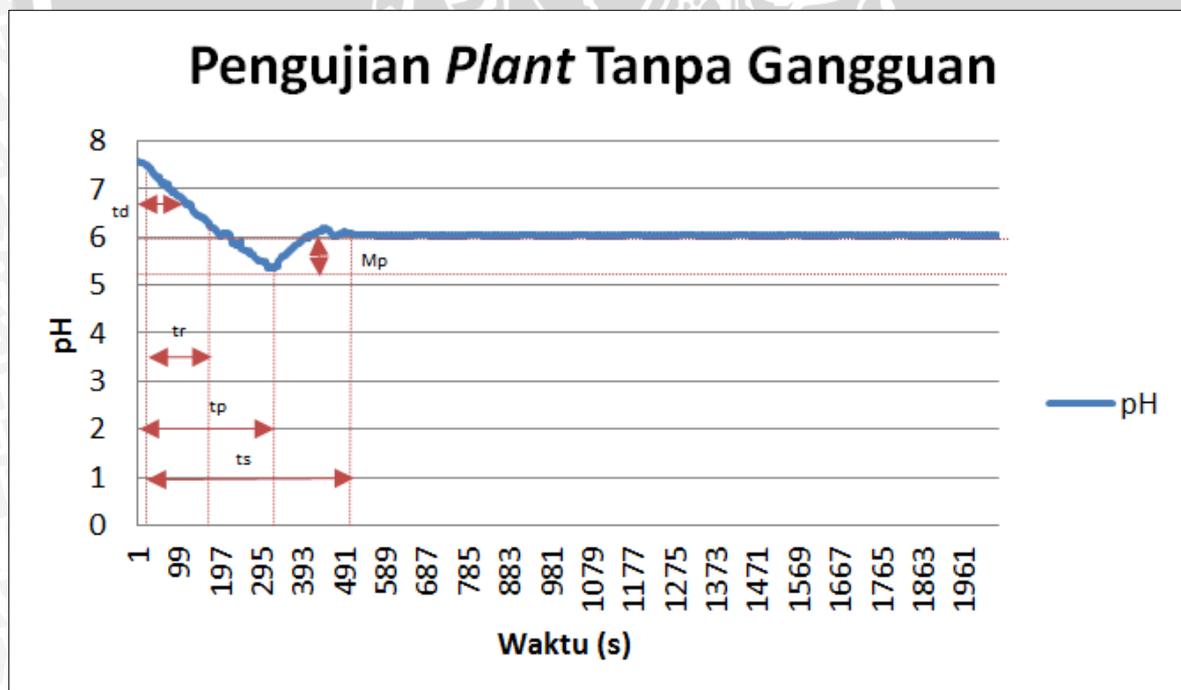
HASIL DAN PEMBAHASAN

Hasil dan pembahasan merupakan hasil dari pengujian keseluruhan sistem dengan parameter-parameter yang telah diberikan dari perhitungan. Parameter-parameter tersebut dilakukan pengujian apakah hasilnya sesuai dengan input yang sudah diberikan dan dapat diaplikasikan di dalam sistem yang telah dibuat. Selain itu juga untuk dapat melihat hasil respon yang dihasilkan. Pengujian keseluruhan sistem dibagi menjadi beberapa bagian yaitu:

1. Pengujian pH Larutan Nutrisi Tanpa Gangguan
2. Pengujian pH Larutan Nutrisi dengan Gangguan

4.1 Pengujian pH Larutan Nutrisi Tanpa Gangguan

Pengujian pengontrolan pH larutan nutrisi tanpa gangguan menggunakan kontroler PID $K_p = 4.26$, $K_i = 0.1065$, dan $K_d = 42.6$ didapatkan hasil *output* yang ditunjukkan pada Gambar 4.1 sebagai berikut.



Gambar 4. 1 Pengujian *Plant* Tanpa Gangguan

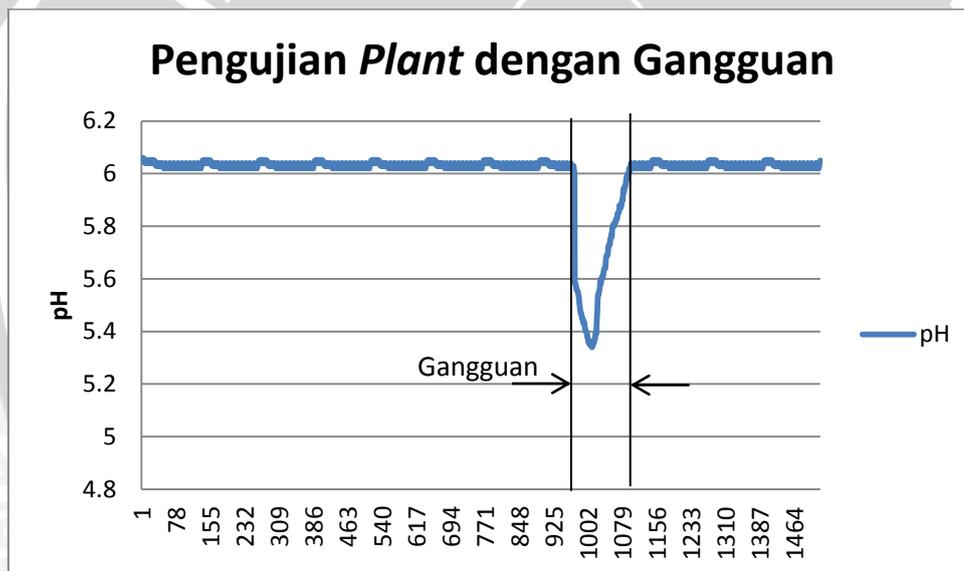
(Sumber : Perancangan)

Dari *output* hasil pengujian *plant* tanpa gangguan dengan menggunakan parameter kontroler PID tersebut diperoleh:

- *Settling time* (t_s) = 520 detik
- *Peak time* (t_p) = 312 detik
- *Rise time* (t_r) = 157 detik
- *Delay time* (t_d) = 107 detik
- *Maximum overshoot* (M_p) = 10.67%
- *error steady state* = 5%

4.2 Pengujian pH Larutan Nutrisi dengan Gangguan

Pengujian pengontrolan pH larutan nutrisi dengan gangguan larutan asam 100 mL dengan pH 4 menggunakan kontroler PID $K_p = 4.26$, $K_i = 0.1065$, dan $K_d = 42.6$ didapatkan hasil *output* yang ditunjukkan pada Gambar 4.2 sebagai berikut.



Gambar 4.2 Pengujian *Plant* dengan Gangguan

(Sumber : Perancangan)

Untuk Gambar 4.2 respon PID dengan gangguan larutan asam 100 mL dengan pH 4, gangguan terjadi pada detik ke 972 hingga detik ke 1090 dengan nilai penurunan pH sebesar 0.66, namun setelah detik ke 1091 nilai pH kembali mengikuti *setting point* dengan *error steady state* sebesar 5% dan *recovery time* sebesar 118 detik.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan pada perancangan alat dan hasil pengujian dapat diambil beberapa kesimpulan, yaitu:

1. Hasil pengujian dari respon sistem PID pada pengontrolan nutrisi dengan pH sebagai indikator, *plant* pengendali nutrisi didapatkan nilai $L = 20$ detik, $T = 71$ detik, dan $K = 0.06$.
2. Hasil pengujian untuk menentukan parameter PID dari sistem pengontrolan ini adalah sebagai berikut:
 - a. Respon sistem PID pada *setting point* tetap 6 dari keadaan awal 5.36; respon sistem telah menuju ke titik kestabilan untuk nilai parameter PID $K_p = 4.26$, $T_i = 40$, dan $T_d = 42.6$.
 - b. Pada pengujian pengontrolan nutrisi dengan pH sebagai indikator tanpa gangguan diperoleh *settling time* sebesar 520 detik dan *error steady state* sebesar 5%.
 - c. Pada pengujian pengontrolan nutrisi dengan pH sebagai indikator dengan gangguan larutan asam dengan nilai pH 4 sudah dapat mencapai kestabilan dengan *error steady state* sebesar 5% dan *recovery time* sebesar 118 detik.

5.2 Saran

Di dalam perancangan dan pembuatan alat dalam pengujian ini masih banyak terdapat kelemahan dan kekurangan. Oleh karena itu, terdapat beberapa hal yang dapat dijadikan saran untuk kesempurnaan alat lebih lanjut, sebagai berikut:

1. Untuk mempercepat pencampuran larutan nutrisi pada tangki I dapat ditambahkan kontroler *on off* motor pengaduk agar sistem berjalan lebih cepat.
2. Untuk mengoptimalkan larutan dalam proses penyerapan nutrisi ke dalam tanaman sebelum *plant* dapat ditambahkan sensor pengendali aliran air supaya oksigen dapat terdifusi ke dalam air sehingga akan membuat pertumbuhan tanaman semakin meningkat.
3. Untuk pengembangan lebih lanjut dapat dibandingkan dengan pertumbuhan tanaman tidak menggunakan media sistem hidroponik.

DAFTAR PUSTAKA

- Amruchly, A. 2014. *Metode Root Locus Untuk Mencari Parameter PID Pada Pengendalian Posisi Stamping Rod Berbasis Pneumatic Menggunakan Arduino Uno*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Anonim. 2016. *Arduino Mega 2560*. Diakses dari <http://www.arduino.cc/en/Main/ArduinoBoardMega2560> pada tanggal 7 Januari 2016.
- Arindya, R. 2014. *Instrumentasi dan Kontrol Proses*. Yogyakarta: Graha Ilmu.
- Ayu. 2014. *Pengendalian Kadar Keasaman (pH) Pada Proses Pengendapan Suhu Tahu Menggunakan Kontroler PID Berbasis ATmega328*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Kadir, A. 2005. *Algoritma Pemrograman Menggunakan C++*. Yogyakarta: Penerbit ANDI Yogyakarta.
- Kustanti, I. 2014. *Pengontrolan Kadar Keasaman (pH) Pada Sistem Hidroponik Stroberi Menggunakan Kontroler PID Berbasis Arduino Uno*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Mupasanta, S. 2016. *Pengontrolan Kadar Keasaman (pH) dan Aliran Air Pada Sistem Hidroponik Stroberi Menggunakan Kontroler PID*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Ogata, K. 2010. *Modern Control Engineering*, Pearson Education, Inc., publishing as Prentice Hall, One Lake Street, Upper Saddle River, New Jersey 07458. Fifth edition.
- Setiawan, I. 2008. *Kontrol PID untuk Proses Industri*. Jakarta: PT. Elex Media Komputindo.
- Supriati, Y. 2014. *Bertanam Tomat di Pot*. Jakarta: Penebar Swadaya.
- Suryani, R. 2015. *Hidroponik Budi Daya Tanaman Tanpa Tanah*. Yogyakarta: Penerbit ARCITRA.
- Soemarwanto. 2010. *Dasar Konversi Energi Elektrik*. Malang.
- Tugiyono, H. 1999. *Bertanam Tomat, Seri Pertanian*. Jakarta: Penerbit Niaga Swadaya.

UNIVERSITAS BRAWIJAYA

LAMPIRAN I

FOTO ALAT





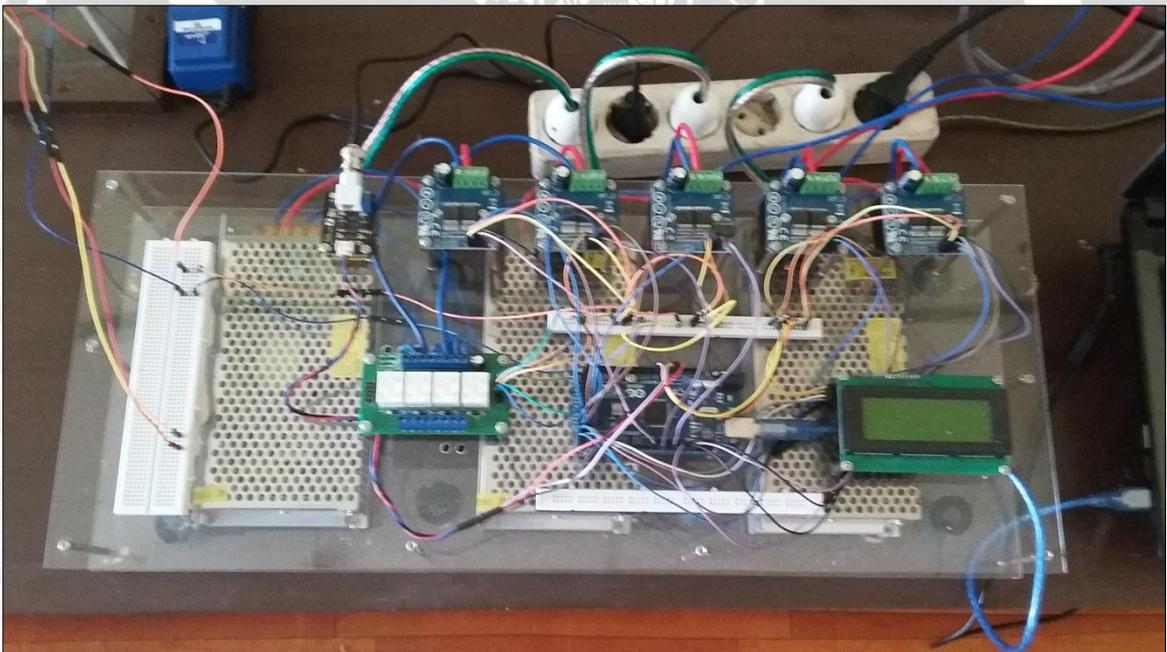
Gambar 1 Alat Pengontrol Nutrisi pada Sistem Tomat Hidroponik Tampak Bagian Atas



Gambar 2 Alat Pengontrol Nutrisi pada Sistem Tomat Hidroponik Tampak Bagian Depan



Gambar 3 Komponen Elektronika Tampak Samping

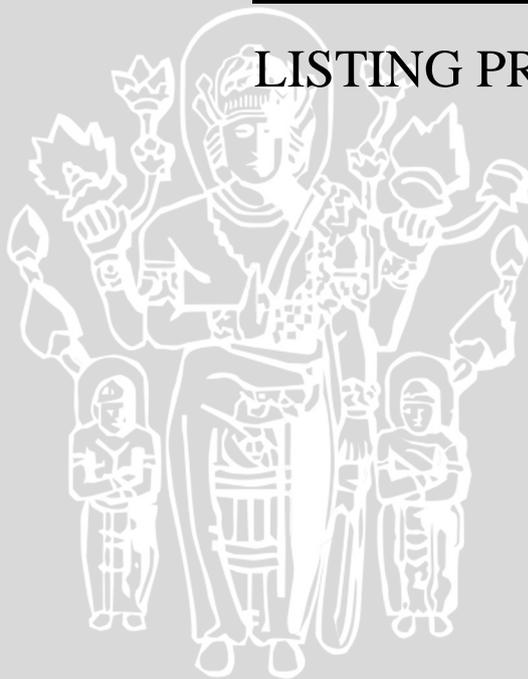


Gambar 4 Komponen Elektronika Tampak Atas

UNIVERSITAS BRAWIJAYA

LAMPIRAN II

LISTING PROGRAM



Listing Program Keseluruhan dengan Arduino Mega 2560

```
#include <SimpleTimer.h>
```

```
SimpleTimer timer;
```

```
double setPoint, error, dError, sError, lError, dTime, PID, kP, kI, kD, Ts, KP_PID,  
KI_PID, KD_PID, KI_PID_LAST, error1;
```

```
long pwm;
```

```
long delayCampur;
```

```
long delayPompa;
```

```
int LPWM_makro = 13; // H-Bridge leg 1 -> LPWM
```

```
int L_EN_makro = 33; // H-Brdge enable pin 1 -> L_EN
```

```
int RPWM_makro = 12; // H-Bridge leg 2 -> RPWM
```

```
int R_EN_makro = 32; // H-Bridge enable pin 2 -> R_EN
```

```
int LPWM_mikro = 11; // H-Bridge leg 1 -> LPWM
```

```
int L_EN_mikro = 31; // H-Brdge enable pin 1 -> L_EN
```

```
int RPWM_mikro = 10; // H-Bridge leg 2 -> RPWM
```

```
int R_EN_mikro = 30; // H-Bridge enable pin 2 -> R_EN
```

```
int LPWM_air = 9; // H-Bridge leg 1 -> LPWM
```

```
int L_EN_air = 29; // H-Brdge enable pin 1 -> L_EN
```

```
int RPWM_air = 8; // H-Bridge leg 2 -> RPWM
```

```
int R_EN_air = 28; // H-Bridge enable pin 2 -> R_EN
```

```
int LPWM_pompa1 = 5; // H-Bridge leg 1 -> LPWM
```

```
int L_EN_pompa1 = 25; // H-Brdge enable pin 1 -> L_EN
```

```
int RPWM_pompa1 = 4; // H-Bridge leg 2 -> RPWM
```

```
int R_EN_pompa1 = 24; // H-Bridge enable pin 2 -> R_EN
```

```
int LPWM_pompa2 = 7; // H-Bridge leg 1 -> LPWM
```

```
int L_EN_pompa2 = 27; // H-Brdge enable pin 1 -> L_EN
```

```
int RPWM_pompa2 = 6; // H-Bridge leg 2 -> RPWM  
int R_EN_pompa2 = 26; // H-Bridge enable pin 2 -> R_EN
```

```
int Relay1 = 34; // Arduino Pin to control the Relay  
int Relay2 = 35; // Arduino Pin to control the Relay
```

```
const int SensorPin = A0; //pin Input sensor pH  
float PH, voltage, pHValue;
```

```
#define echo 2  
#define trig 3  
float distance;
```

```
void setup()  
{  
  Serial.begin (9600);
```

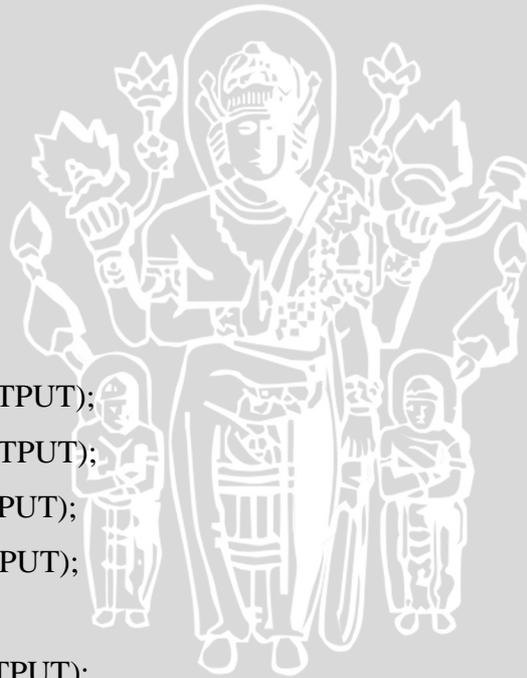
```
  pinMode(0,INPUT);
```

```
  pinMode(LPWM_makro, OUTPUT);  
  pinMode(RPWM_makro, OUTPUT);  
  pinMode(L_EN_makro, OUTPUT);  
  pinMode(R_EN_makro, OUTPUT);
```

```
  pinMode(LPWM_mikro, OUTPUT);  
  pinMode(RPWM_mikro, OUTPUT);  
  pinMode(L_EN_mikro, OUTPUT);  
  pinMode(R_EN_mikro, OUTPUT);
```

```
  pinMode(LPWM_air, OUTPUT);  
  pinMode(RPWM_air, OUTPUT);  
  pinMode(L_EN_air, OUTPUT);  
  pinMode(R_EN_air, OUTPUT);
```

UNIVERSITAS BRAWIJAYA



```
pinMode(LPWM_pompa1, OUTPUT);
pinMode(RPWM_pompa1, OUTPUT);
pinMode(L_EN_pompa1, OUTPUT);
pinMode(R_EN_pompa1, OUTPUT);

pinMode(LPWM_pompa2, OUTPUT);
pinMode(RPWM_pompa2, OUTPUT);
pinMode(L_EN_pompa2, OUTPUT);
pinMode(R_EN_pompa2, OUTPUT);

pinMode(Relay1, OUTPUT); // Pump to plant
pinMode(Relay2, OUTPUT); // Pump to tank 1

pinMode(echo, INPUT);
pinMode(trig, OUTPUT);

delayCampur = 10000;
delayPompa = 2000;

PH = 0;
error = 0;
dError = 0;
sError = 0;
lError = 0;
dTime = 0;
PID = 0;

setPoint = 1.71;
kP = 4.26;
kI = 0.1065;
kD = 42.6;
Ts = 13;

timer.setInterval(1000, driverEnableTimer);
```



```
timer.setInterval(1000, nutrisiTimer);
timer.setInterval(1000, relayTimer);
timer.setInterval(1000, computeTimer);
timer.setInterval(1000, readPhTimer);
timer.setInterval(1000, jarakTimer);
}
void driverEnableTimer()
{
  driverEnable();
}
void nutrisiTimer()
{
  nutrisi();
}
void relayTimer()
{
  relay();
}
void computeTimer()
{
  compute();
}
void readPhTimer()
{
  readPh();
}
void jarakTimer()
{
  jarak();
}
void loop()
{
  timer.run();
}
```

UNIVERSITAS BRAWIJAYA

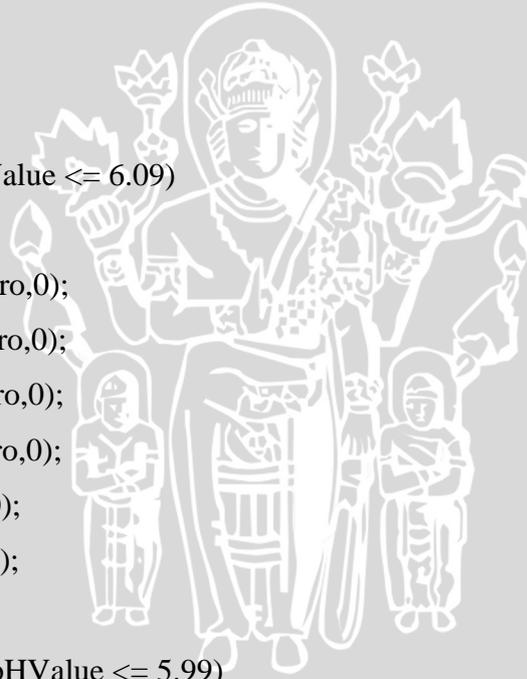


```
void driverEnable()
{
    digitalWrite(L_EN_makro, HIGH);
    digitalWrite(R_EN_makro, OUTPUT);
    digitalWrite(L_EN_mikro, HIGH);
    digitalWrite(R_EN_mikro, OUTPUT);
    digitalWrite(L_EN_air, HIGH);
    digitalWrite(R_EN_air, OUTPUT);

    digitalWrite(LPWM_makro, LOW);
    digitalWrite(LPWM_mikro, LOW);
    digitalWrite(LPWM_air, LOW);
}

void nutrisi()
{
    if (pHValue >=6 && pHValue <= 6.09)
    {
        analogWrite(RPWM_makro,0);
        analogWrite(LPWM_makro,0);
        analogWrite(RPWM_mikro,0);
        analogWrite(LPWM_mikro,0);
        analogWrite(RPWM_air,0);
        analogWrite(LPWM_air,0);
    }
    else if (pHValue > 0 && pHValue <= 5.99)
    {
        analogWrite(RPWM_makro,0);
        analogWrite(LPWM_makro,0);
        analogWrite(RPWM_mikro,0);
        analogWrite(LPWM_mikro,0);
        analogWrite(RPWM_air,pwm);
        analogWrite(LPWM_air,0);
    }
    else if (pHValue >= 6.1 && pHValue < 14)
```

UNIVERSITAS BRAWIJAYA

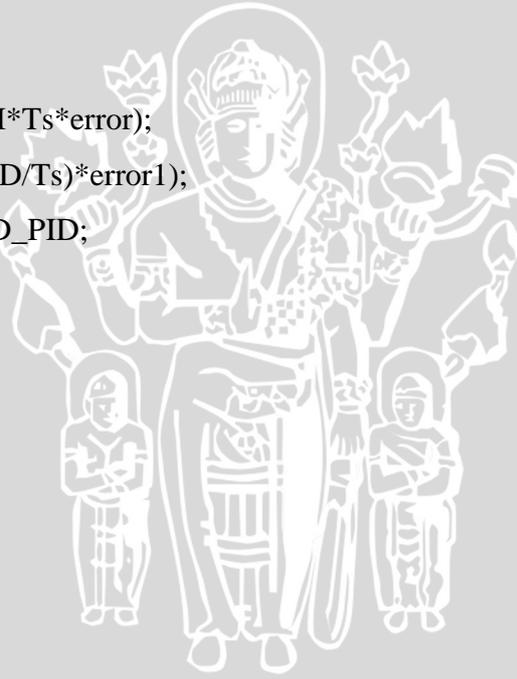


```
{
    analogWrite(RPWM_makro,pwm);
    analogWrite(LPWM_makro,0);
    analogWrite(RPWM_mikro,pwm);
    analogWrite(LPWM_mikro,0);
    analogWrite(RPWM_air,0);
    analogWrite(LPWM_air,0);
}
}
void compute()
{
    error = setPoint-voltage;

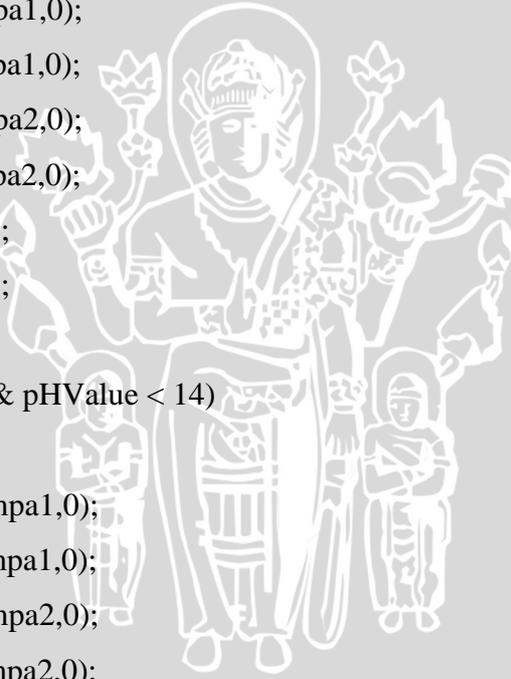
    KP_PID = kP*error;
    KI_PID = KI_PID_LAST + (kI*Ts*error);
    KD_PID = ((kD/Ts)*error)-((kD/Ts)*error1);
    PID = KP_PID + KI_PID + KD_PID;

    KI_PID_LAST = KI_PID;
    error1 = error;

    if(PID>255)
        pwm = 255;
    else if(PID<-255)
        pwm = 255;
    else
    {
        if(PID>0)
            pwm = (int) PID;
        else
            pwm = (int)(PID*(-1));
    }
}
void relay()
```



```
{
if (pHValue >=6 && pHValue <= 6.09)
{
analogWrite(RPWM_pompa1,255);
analogWrite(LPWM_pompa1,0);
analogWrite(RPWM_pompa2,0);
analogWrite(LPWM_pompa2,0);
digitalWrite(Relay1,HIGH);
digitalWrite(Relay2,LOW);
}
else if (pHValue > 0 && pHValue <= 5.99)
{
analogWrite(RPWM_pompa1,0);
analogWrite(LPWM_pompa1,0);
analogWrite(RPWM_pompa2,0);
analogWrite(LPWM_pompa2,0);
digitalWrite(Relay1,LOW);
digitalWrite(Relay2,LOW);
}
else if (pHValue >= 6.1 && pHValue < 14)
{
analogWrite(RPWM_pompa1,0);
analogWrite(LPWM_pompa1,0);
analogWrite(RPWM_pompa2,0);
analogWrite(LPWM_pompa2,0);
digitalWrite(Relay1,LOW);
digitalWrite(Relay2,LOW);
}
else if (distance >= 10)
{
analogWrite(RPWM_pompa1,0);
analogWrite(LPWM_pompa1,0);
analogWrite(RPWM_pompa2,255);
analogWrite(LPWM_pompa2,0);
```



```
digitalWrite(Relay1,LOW);
digitalWrite(Relay2,HIGH);
}
else if (distance >= 18 && distance <= 17)
{
  analogWrite(RPWM_pompa1,0);
  analogWrite(LPWM_pompa1,0);
  analogWrite(RPWM_pompa2,0);
  analogWrite(LPWM_pompa2,0);
  digitalWrite(Relay1,LOW);
  digitalWrite(Relay2,LOW);
}
}
void readPh()
{
  PH = 1023-analogRead(SensorPin);
  voltage = PH*5/1024;    //convert the analog into volt
  pHValue = -9.043148957+2.464188979*(PH*14/1024);//convert the read sensor into
  pH value

  Serial.print("Voltage:");
  Serial.print(voltage,2);
  Serial.print(" pH: ");
  Serial.println(pHValue,2);
  delay(1000);
}
void jarak()
{
  // Start Ranging-Generating a trigger of 10us burst
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
```

```
distance = pulseIn(echo, HIGH);
```

```
// Distance Calculation
```

```
distance = distance/58;
```

```
}
```



UNIVERSITAS BRAWIJAYA

LAMPIRAN III

DATA SHEET



