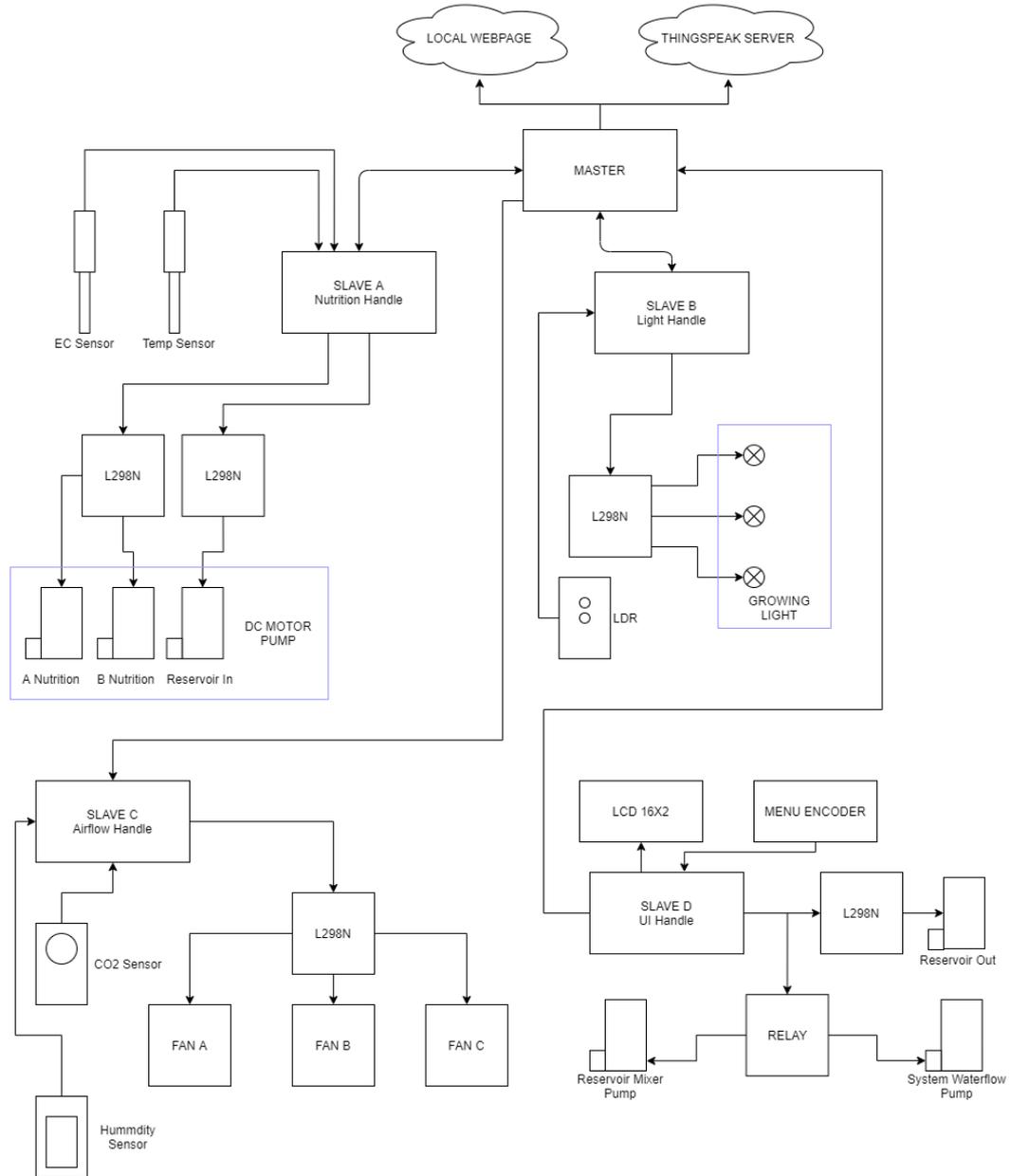


BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Diagram Blok



Gambar 3.1 Diagram Blok Sistem

Block Diagram pada gambar 3.1 menggambarkan bagaimana sistem akan diimplementasikan. Sistem ini secara garis besar terbagi menjadi 5 bagian utama. Bagian pertama adalah bagian *Master* dari komunikasi ke empat perangkat lainnya, Bagian ini berfungsi untuk mengatur bacaan sensor dan sebagai server dari HTML yang disimpan pada STIFFS Node Mcu ini.

Bagian kedua adalah *Slave A* yang dikhususkan untuk pengendalian kadar nutrisi sistem dengan bantuan beberapa sensor seperti sensor temperatur air, dan sensor *electrical conductivity* yang dimana kedua sensor tersebut berpengaruh terhadap nilai TDS (*Total Dissolved Solids*) dari cairan nutrisi yang terdapat di reservoir, dan nantinya akan dialirkan ke sistem setelah nilai TDS telah mencapai *setpoint*. Aktuator pada bagian ini adalah pompa DC yang terhubung dengan *motor driver* L298N, serta memiliki sistem kendali PID.

Pada bagian ketiga yang merupakan *Slave B*, terdapat sensor intensitas cahaya yang dalam sistem ini digunakan LDR atau *Light Dependent Resistor*, bertujuan untuk mengukur intensitas cahaya yang berasal dari LED maupun luaran sistem. Sistem pada bagian kedua ini tentu saja mengatur kestabilan intensitas cahaya agar tetap stabil dan memiliki porsi penyinaran yang sesuai dengan spesifikasi kebutuhan tanaman. Pada bagian ketiga ini, kendali yang digunakan juga PID dengan aktuator lampu LED yang terhubung dengan *motor driver* L298N. Pemilihan *motor driver* sebagai *dimmer* dari lampu LED pada sistem adalah dikarenakan membutuhkan arus yang cukup besar untuk menyalakannya.

Bagian keempat dari sistem ini yaitu *Slave C*, terdapat *CO₂ Sensor* dimana sensor ini akan membaca jumlah PPM dari gas CO₂ yang terdapat di dalam sistem, tepatnya pada bagian daun dari tanaman, tentunya kadar ini diatur untuk mendukung terjadinya proses fotosintesis serta menghindari terjadinya *tipburn* pada ujung daun tanaman. Selain itu juga terdapat sensor kelembaban untuk mengukur tingkat kelembaban sistem. Untuk aktuator dari bagian keempat sistem ini, antara lain terdapat L298N yang terkoneksi dengan *cooling fan*.

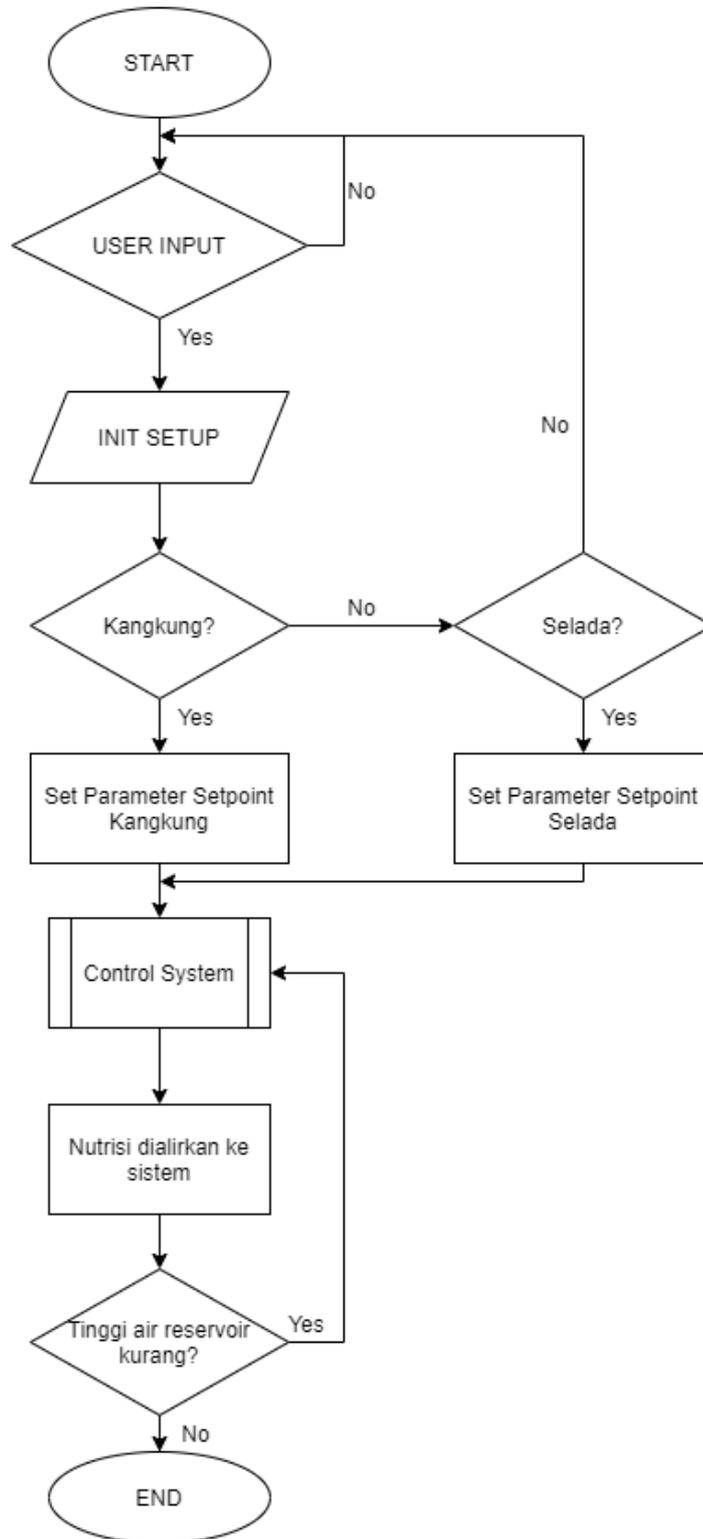
Bagian kelima adalah *microcontroller Slave D* yang berisi *user interface* (dalam hal ini berupa LCD 16x2, *rotary encoder* untuk memilih menu), Sensor ketinggian nutrisi yang terdapat dalam reservoir, relay pompa reservoir, dan bagian penentuan parameter nutrisi maupun intensitas cahaya yang tepat dan akan dikirimkan ke masing-masing *slave* untuk tanaman yang ditanam pada sistem.

3.2 Flow Chart Sistem

Alur dari sistem smart vertical farming ini dapat dilihat dari *Flow Chart* pada gambar 3.2, sistem ini di desain dengan input dari *user* sebagai pemilihan jenis tanaman yang akan ditanam, nantinya sistem akan menyesuaikan baik dari segi nutrisi, pencahayaan, hingga waktu panen yang akan dicapai. Setelah parameter-parameter tumbuh tanaman telah di set, hal yang selanjutnya dilakukan oleh sistem adalah memberikan parameter tersebut ke *microcontroller* dengan tugasnya masing-masing, lalu dijalankan pengendaliannya.

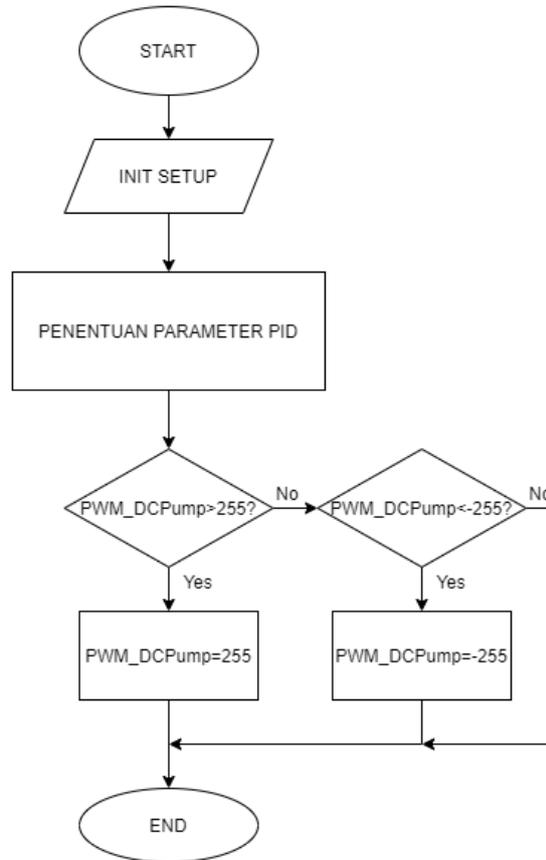
Setelah semua nilai yang dibaca oleh sensor telah sesuai dengan *setpoint*, selanjutnya sistem akan mengalirkan nutrisi yang sudah sesuai tersebut ditandai dengan dinyalakannya pompa sistem dengan *DC Pump* bertujuan mendorong nutrisi ke atas sistem. Sistem akan terus melakukan pengecekan terhadap tinggi air pada *reservoir* agar tidak terlalu surut dan merusak pompa air yang berada didalamnya. Jika terjadi penyurutan yang cukup besar, maka sistem akan memberikan notifikasi kepada user melalui LCD, dan tentunya kembali melakukan pengecekan terhadap nilai TDS yang tersisa dalam tangki *reservoir*.

MAIN MODULE



Gambar 3.2 Flow Chart Indoor Vertical Farming System

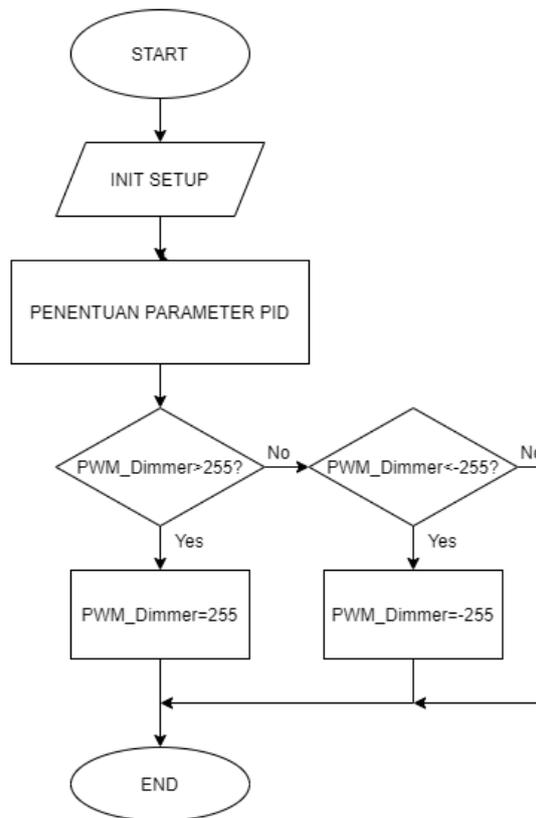
CONTROL SYSTEM NUTRITION TANK



Gambar 3.3 Modul Nutrition Tank Control

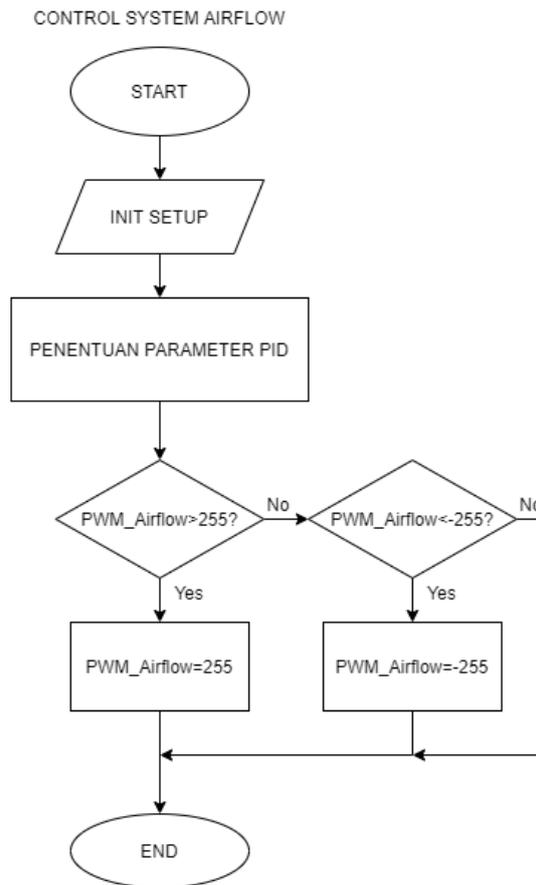
Modul sistem kendali yang bertujuan untuk menjaga stabilitas *TDS* atau *Total Dissolved Solids* di tangki nutrisi dapat terlihat pada gambar 3.3. Sistem ini memanfaatkan sistem kendali PID untuk menjaga kestabilan sistem, sistem secara langsung mengatur beberapa parameter nutrisi yang dibaca melalui sensor, mempercepat pencapaian ke *setpoint*, dan meminimalisir agar *error* yang dihasilkan kecil. Sehingga tanaman akan mendapatkan nutrisi yang optimum sesuai dengan pertumbuhannya.

CONTROL SYSTEM LIGHT INTENSITY



Gambar 3.4 Modul Light Intensity Control

Modul sistem kendali yang kedua pada sistem ini adalah bertujuan untuk mengatur intensitas cahaya, agar intensitasnya sesuai dengan kebutuhan tanaman. Modul kedua ini memanfaatkan sistem kendali PID untuk mempercepat waktu respon. Modul ini menggunakan *LED Super Bright* yang diatur intensitasnya menggunakan *motor driver* L298N, serta menggunakan *feedback* sensor berupa LDR



Gambar 3.5 Modul Airflow Control

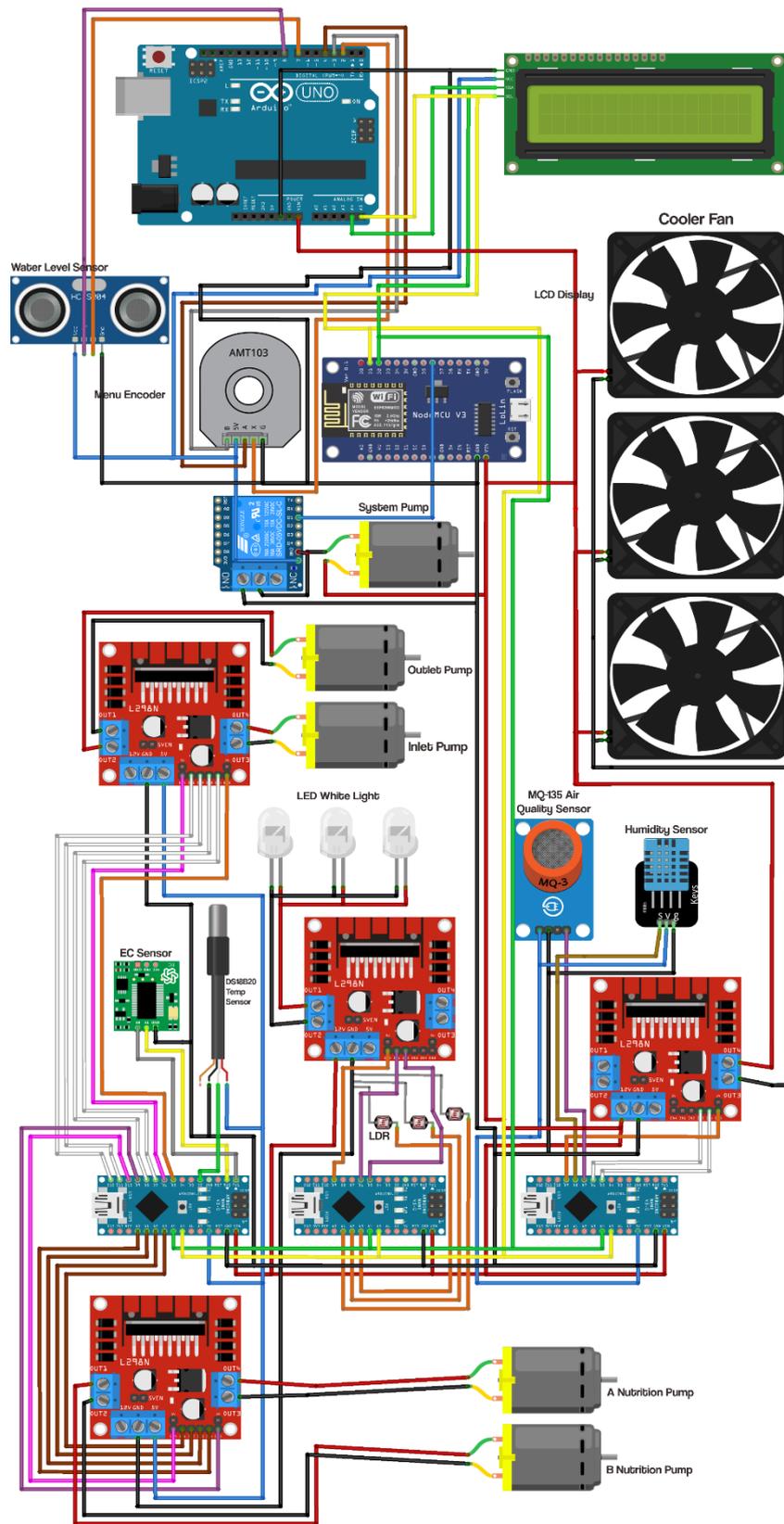
Modul sistem kendali yang ketiga pada sistem ini adalah bertujuan untuk mengatur *airflow* pada bagian atas tanaman. Diimplementasikan untuk mengurangi resiko *tipburn* pada ujung daun, serta menjaga agar CO₂ pada sistem cukup untuk kebutuhan tanaman. Pemanfaatan aktuator berupa *cooler fan* yang kecepatannya diatur menggunakan *motor driver* L298N, serta *feedback sensor* MQ-135, digunakan dalam modul ini.

3.3 Perancangan Hardware

3.3.1 Wiring Diagram

Wiring Diagram atau pengkabelan pada sistem ini dapat dilihat pada gambar 3.6 dibawah. Secara keseluruhan sistem memanfaatkan 5 microcontroller utama dengan 1 *master* dan 4 *slave*. *Master* yang merupakan pusat sistem yang mengatur *user interface* terhubung dengan LCD dengan modul I2C, *rotary encoder* sebagai alat untuk memilih pada menu utama, serta terhubung dengan relay untuk pompa sistem.

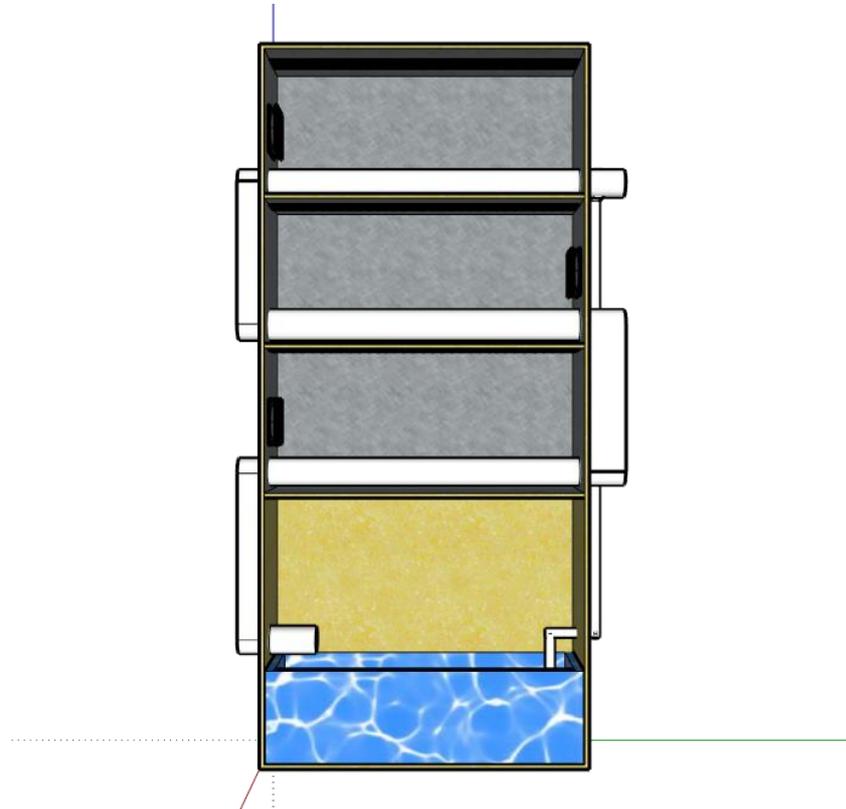
Ketiga *slave* lainnya memiliki fungsi dan tujuan masing-masing yang sudah dijelaskan pada blok diagram diatas, tiap *microcontroller* arduino nano terhubung dengan *motor driver* L298N, lalu terkoneksi dengan aktuatornya masing-masing yang berupa *cooler fan*, *LED Super Bright*, dan juga pompa DC.



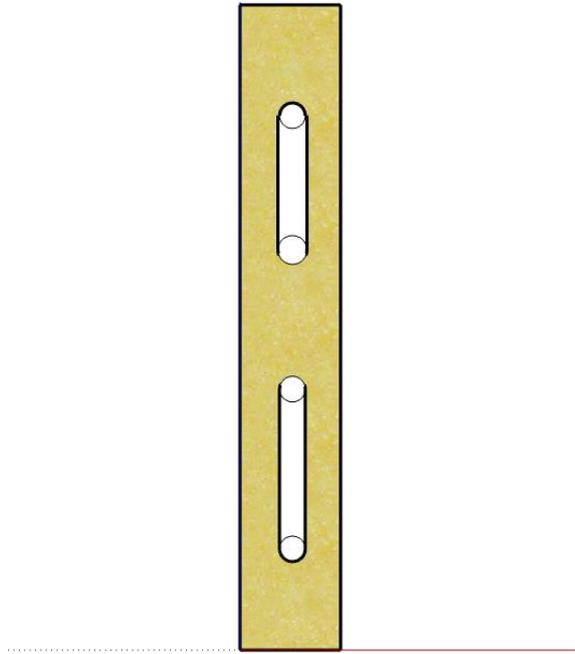
Gambar 3.6 Wiring Diagram

3.3.2 Desain Sistem

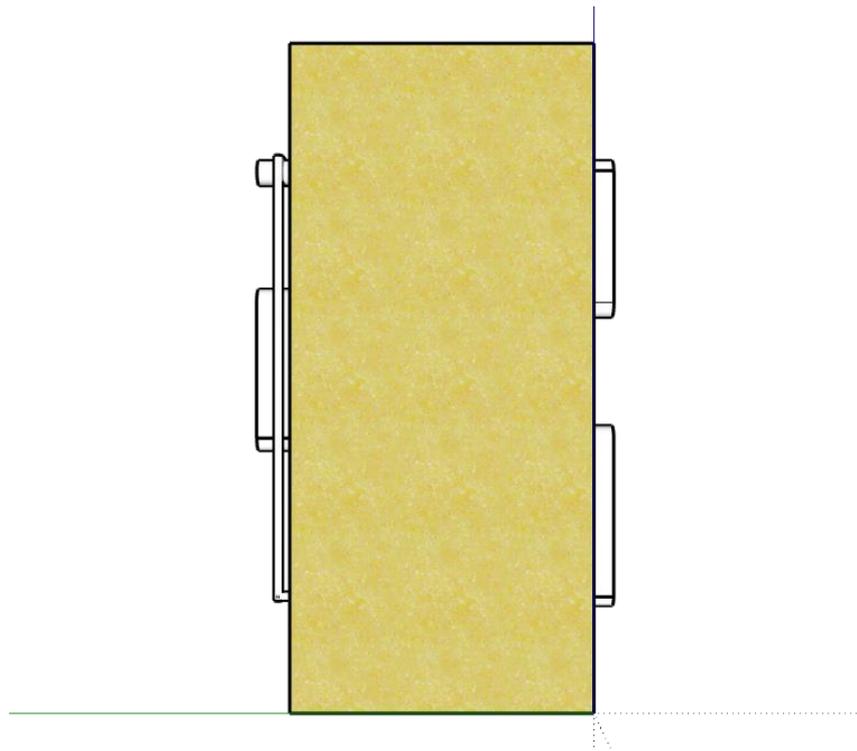
Beberapa detail sistem digambarkan menggunakan CAD (*Computer Aided Design*) dapat dilihat dengan beberapa gambar berikut.



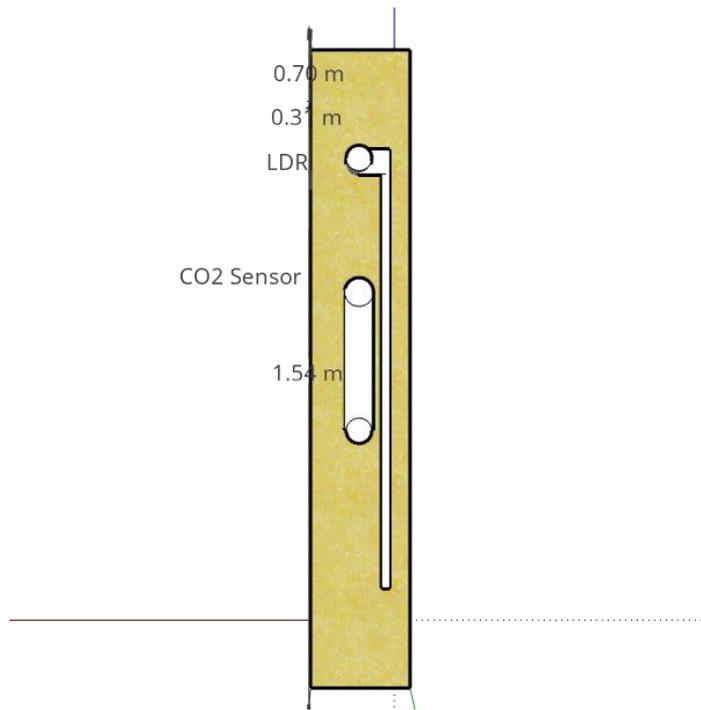
Gambar 3.7 Tampak Depan Sistem



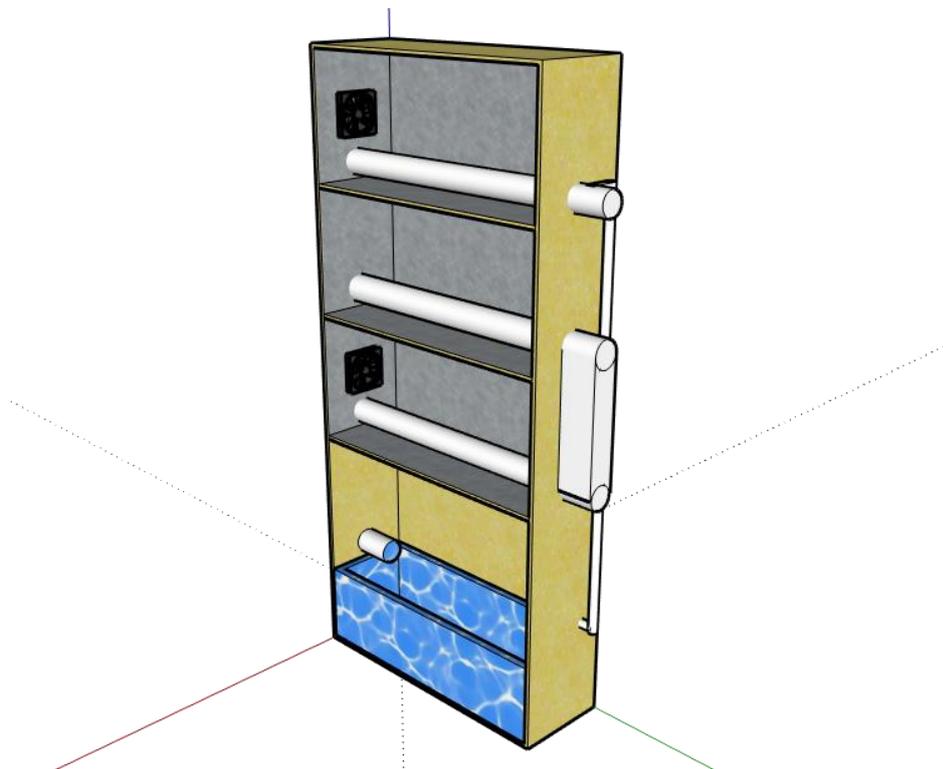
Gambar 3.8 Tampak Kiri Sistem



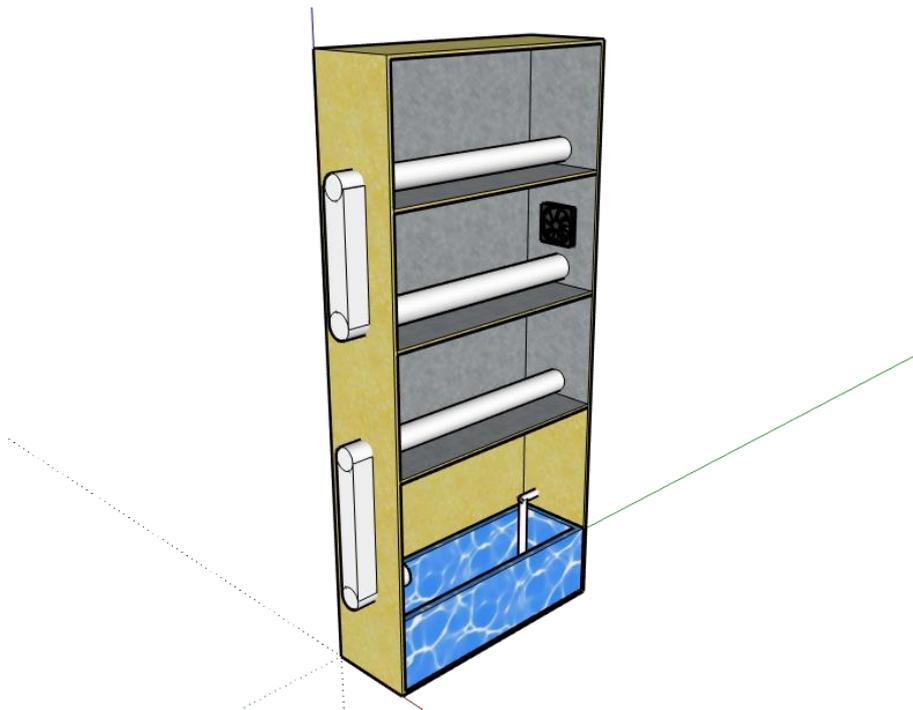
Gambar 3.9 Tampak Belakang Sistem



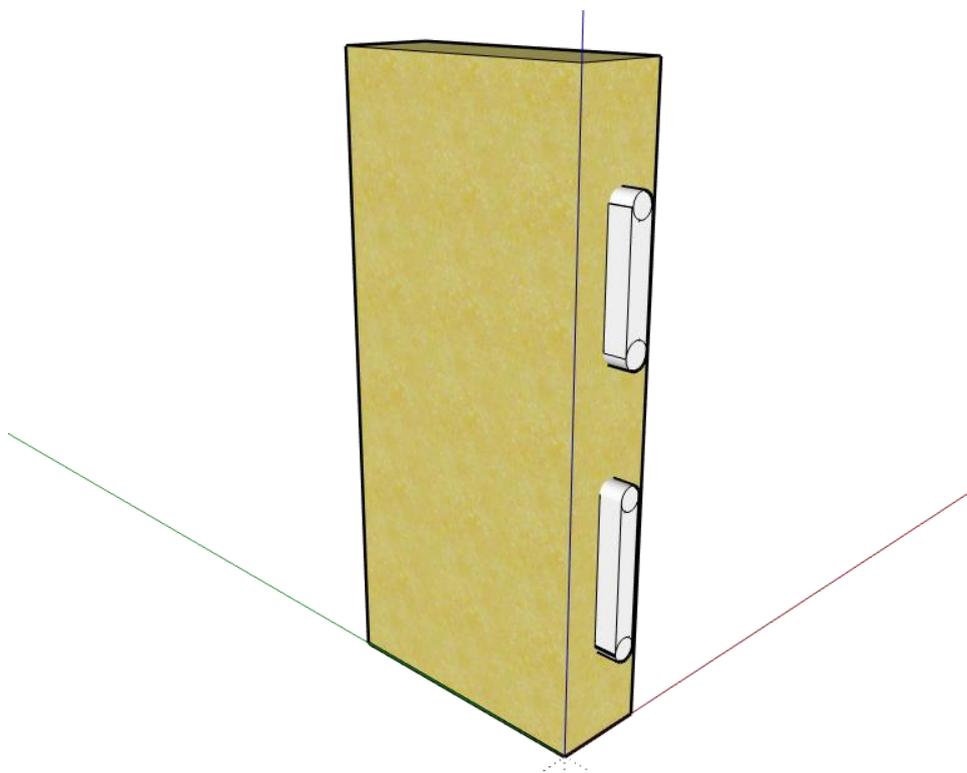
Gambar 3.10 Tampak Kanan Sistem



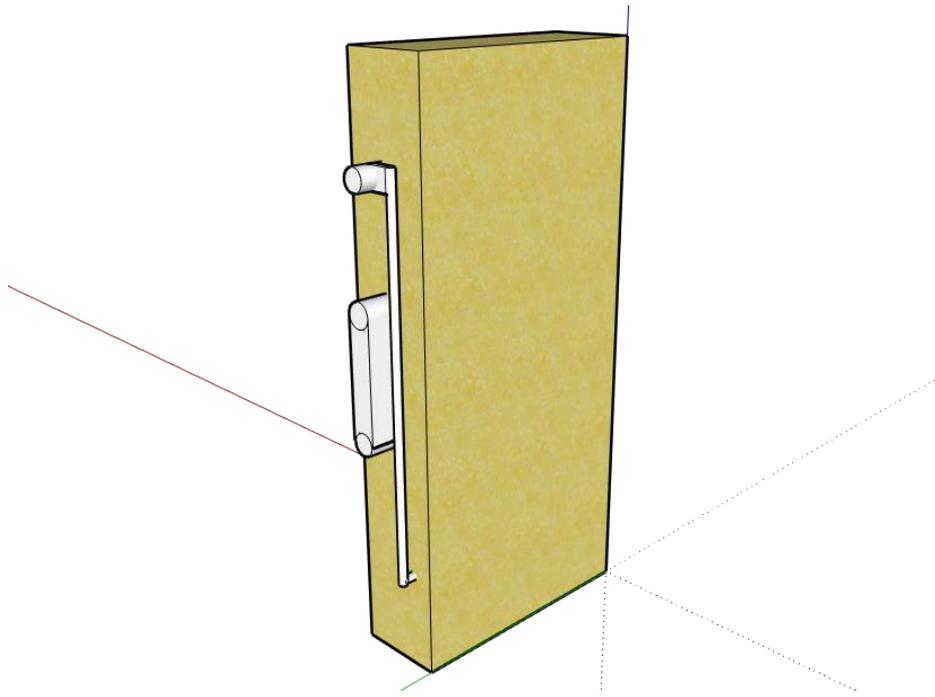
Gambar 3.11 Tampak Orthogonal Kanan Depan Sistem



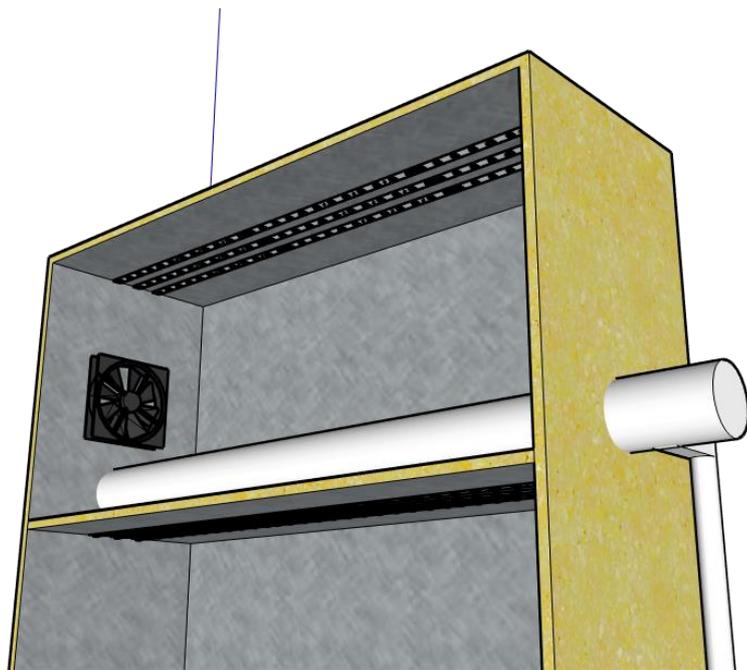
Gambar 3.12 Tampak Orthogonal Kiri Depan Sistem



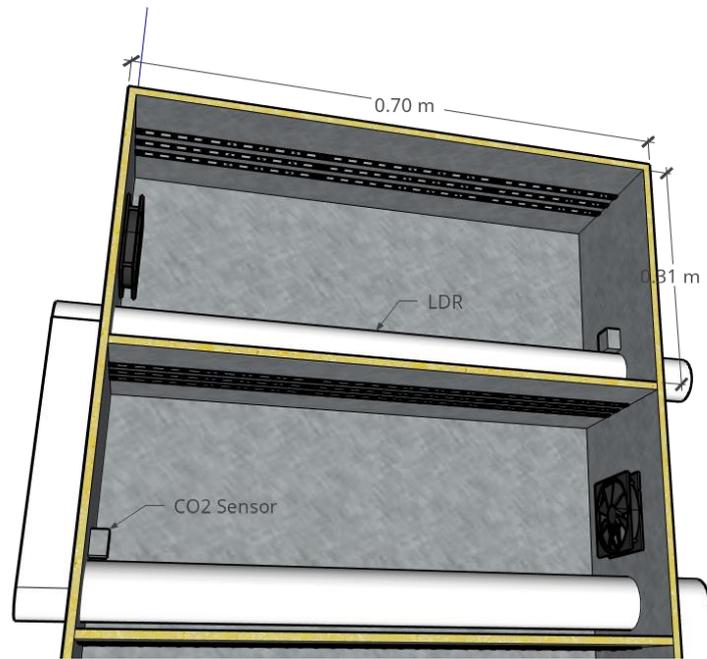
Gambar 3.13 Tampak Orthogonal Kanan Belakang Sistem



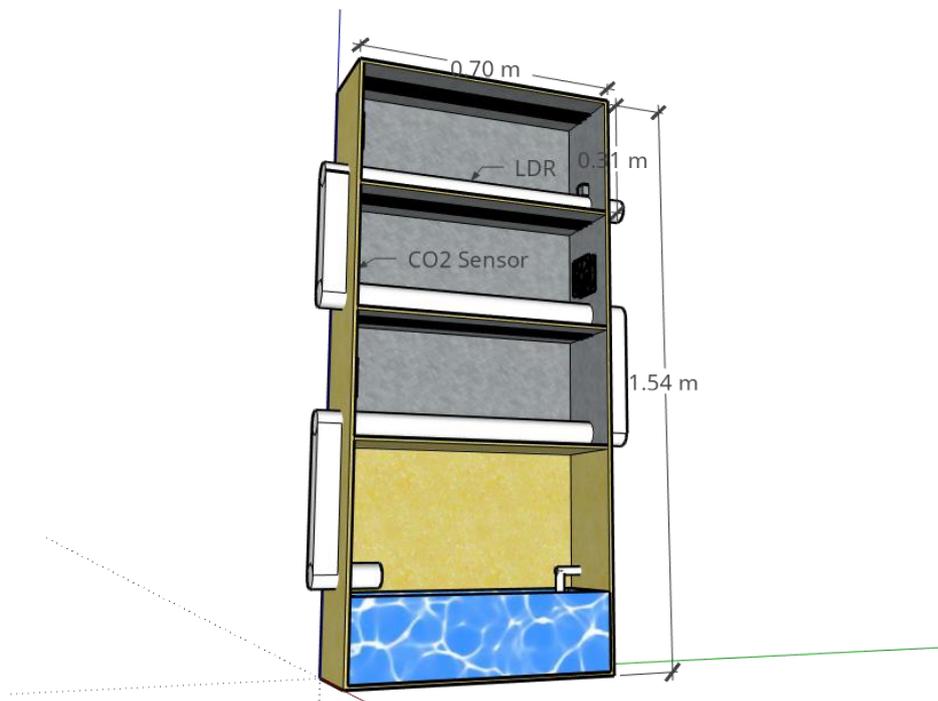
Gambar 3.14 Tampak Orthogonal Kiri Belakang Sistem



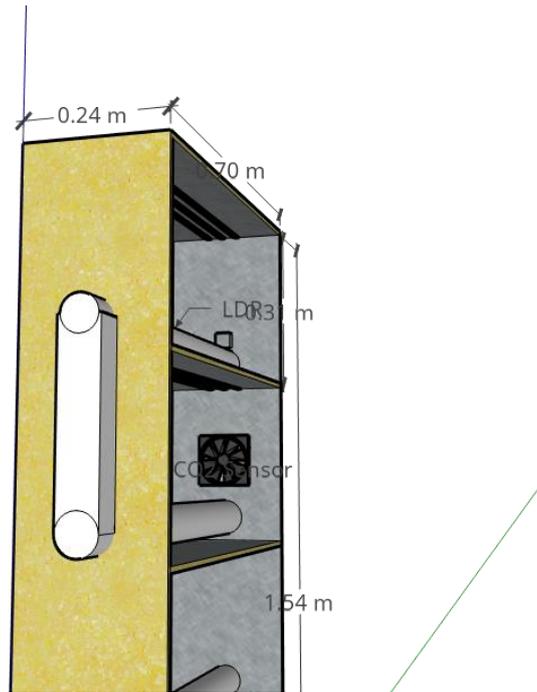
Gambar 3.15 Letak LED dan Cooler Fan pada Sistem



Gambar 3.16 Letak Sensor pada Sistem



Gambar 3.17 Detail Ukuran Panjang dan Tinggi Sistem



Gambar 3.18 Detail Ukuran Lebar Sistem

3.4 Penentuan Parameter

Pada sistem ini terdapat tiga bagian besar yaitu pencahayaan, *airflow*, serta nutrisi yang dikendalikan oleh sistem kendali PID. Maka dari itu untuk merancang sistem kendali agar respon sistemnya sesuai dengan yang diharapkan, maka dicarilah parameter dari masing-masing bagian terlebih dahulu

3.4.1 Parameter Pencahayaan Sistem

Parameter pencahayaan sistem dicari dengan penentuan fungsi alih terlebih dahulu, kemudian memasukkannya kedalam diagram blok pada *simulink*, serta melakukan *tuning* PID agar sistem mendapatkan respon sesuai dengan yang diharapkan.

A. Fungsi Alih Pencahayaan Sistem

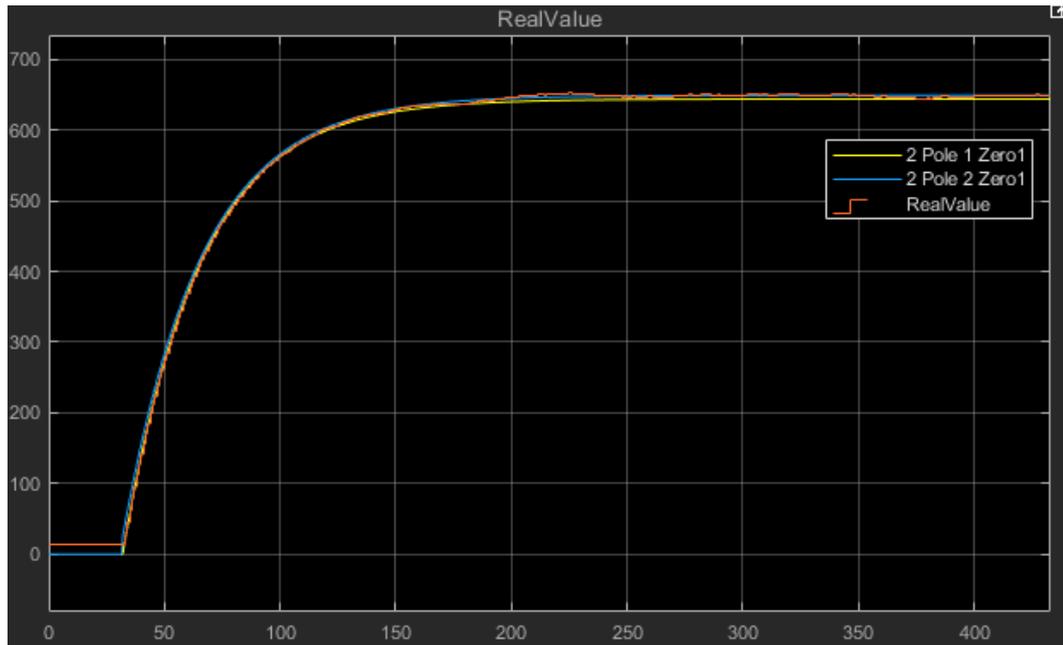
Dengan menggunakan *Matlab Identification Toolbox*, dan memasukkan *input* dan *output* yang dibaca dari sensor pada sistem, didapatkan fungsi alih orde dua dengan satu dan dua *zero*, antara lain hasil yang didapatkan sebagai berikut:

$$\text{Fungsi Alih Pencahayaan} = \frac{0.07666s+0.03734}{s^2+0.5171s+0.01478} \quad (4)$$

Fungsi Alih dengan menggunakan satu *zero* dan dua *pole*, memperoleh nilai *fit to estimation data*: **97.21%**. kemudian dicoba kembali menggunakan fungsi alih dengan dua *zero* dan dua *pole*, kemudian didapatkan hasil sebagai berikut:

$$\text{Fungsi Alih Pencahayaan} = \frac{0.1026s^2 + 0.1374s + 0.04878}{s^2 + 0.6754s + 0.01915} \quad (5)$$

Fungsi Alih dengan menggunakan dua *zero* dan dua *pole* ini memperoleh nilai *fit to estimation data* sebesar: **98.79%**, yaitu 1,58% lebih tinggi dibandingkan dengan fungsi alih yang menggunakan satu *zero*. Untuk memastikan estimasi yang dihasilkan dari *Matlab Identification Toolbox* ini sesuai dengan respon sistem aslinya, maka dilakukan perbandingan menggunakan grafik. Pada grafik 3.19 dibandingkan ketiga nilai yang berbeda antara nilai asli sistem, fungsi alih 1, dan 2 *zero*.

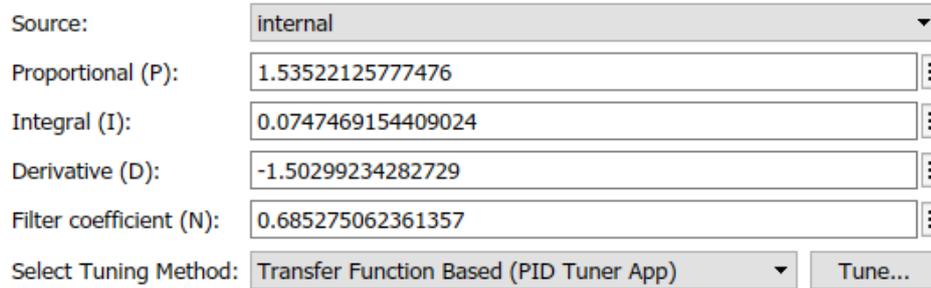


Gambar 3.19 Grafik Perbandingan Fungsi Alih dengan 1 dan 2 Zero
Time Sample = 0.01s

Estimasi fungsi alih sistem mendapatkan ketiga grafik memiliki perbedaan yang cukup kecil, yang paling jauh berbeda adalah pada grafik dengan 2 *pole* dan 1 *zero*, sehingga data tersebut di eliminasi. Serta untuk penentuan parameter PID didapatkan melalui persamaan fungsi alih dengan 2 *pole* dan 2 *zero*

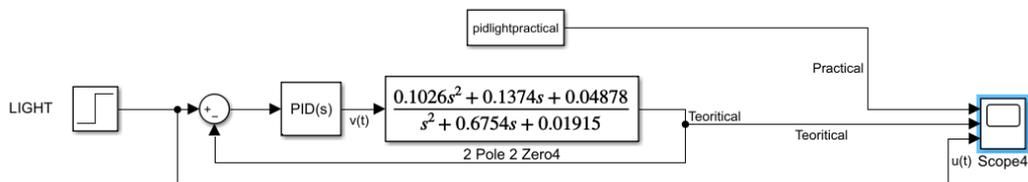
B. Penentuan Parameter PID Pencahayaan Sistem

Dengan menggunakan simulink pada Matlab, maka dibentuklah blok diagram dengan memasukkan fungsi alih pada persamaan (5) kedalam blok *simulink* serta melakukan tuning pada blok PID berbasis *PID Tuner* seperti yang ditunjukkan pada gambar 3.20.



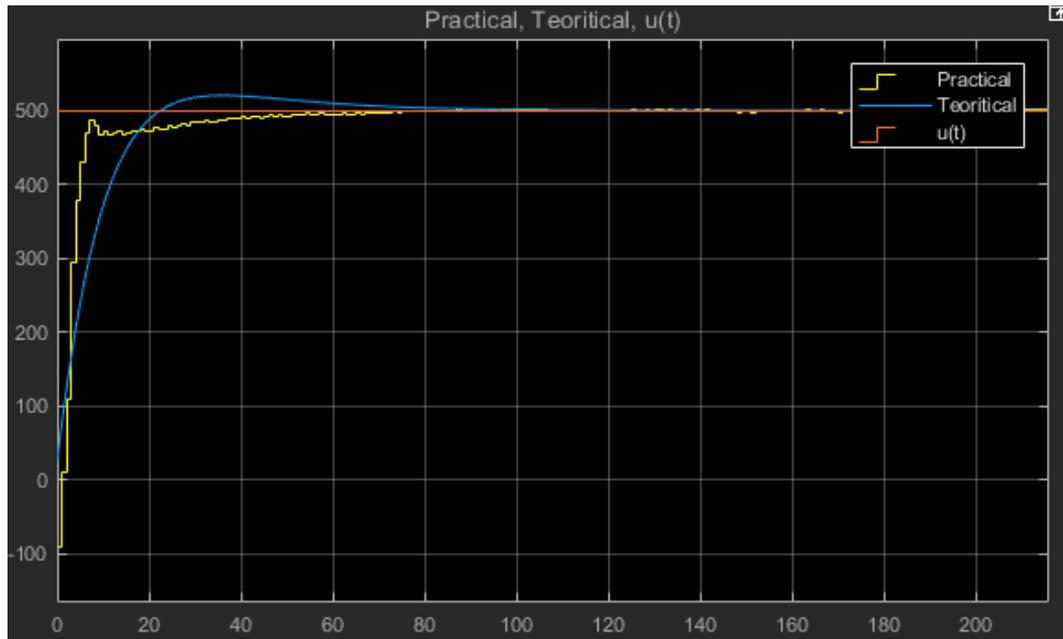
Gambar 3.20 Gambar Jendela PID Tuner Pencahayaan

Setelah itu dilakukan lagi pengujian berupa membandingkan hasil Teori dan Prakteknya, dengan memasukkan nilai nilai dari **Kp: 1.53522125777476**, **Ki: 0.0747469154409024**, dan **Kd: -1.50299234282729** yang didapatkan dari *tuning* PID, ke blok diagram pada *simulink* dan ke *arduino code* sistem, lalu kedua respon dibandingkan. Diagram blok yang digunakan untuk membandingkan kedua sistem ini dapat dilihat pada gambar 3.21, serta hasil perbandingan grafiknya dapat dilihat pada gambar 3.22.



Gambar 3.21 Blok Diagram Perbandingan Respon Pencahayaan

Hasil pengujian kedua sistem (teori dan praktik) didapatkan respon sistem yang tidak jauh berbeda. Dengan begitu parameter yang didapatkan cocok dan cukup baik untuk menangani spesifikasi sistem yang diharapkan dengan respon sistem yang tidak terlalu cepat dan juga tidak terlalu lama.



Gambar 3.22 Respon Sistem Pencahayaan

3.4.2 Parameter Airflow Sistem

Pada saat dilakukan pengujian pertama terhadap aktuator serta sensor dari Airflow sistem, ditemukan bahwa penggunaan *fan* hanya dapat menaikkan kadar ppm CO₂ pada sistem indoor hingga maksimal pada kisaran 600-650 ppm pada posisi sensor 1, serta 400-450 ppm pada posisi sensor 2 dan 3 (posisi sensor dapat dilihat pada gambar 4.2, 4.3, dan 4.4). Padahal sistem membutuhkan kadar CO₂ untuk berfotosintesis sekitar 1200-1500 ppm.

Kadar oksiden yang terbaca oleh sensor dikendalikan sesuai dengan setpoint yang dibutuhkan oleh tanaman. Parameter *airflow* sistem dicari dengan penentuan fungsi alih terlebih dahulu, kemudian memasukkannya kedalam diagram blok pada *simulink*, serta melakukan *tuning* PID agar sistem mendapatkan respon sesuai dengan yang diharapkan.

A. Parameter PID Airflow

Sama halnya dengan sistem pencahayaan, penentuan parameter pada sistem airflow juga dilakukan dengan menggunakan *Matlab Identification Toolbox*, dan memasukkan *input dan output* yang dibaca dari sensor pada sistem, didapatkan fungsi alih orde dua dengan satu dan dua *zero*, antara lain hasil yang didapatkan sebagai berikut:

$$\text{Fungsi Alih Airflow Posisi 1} = \frac{-0.0362s+0.000106}{s^2+0.04567s+0.0001589} \quad (6)$$

Fungsi Alih dengan menggunakan satu *zero* dan dua *pole*, memperoleh nilai *fit to estimation data*: **93.76%**, dicoba kembali menggunakan fungsi alih dengan dua *zero* dan dua *pole*, kemudian didapatkan hasil sebagai berikut:

$$\text{Fungsi Alih Airflow Posisi 1} = \frac{0.16s^2-0.04061s+0.0001189}{s^2+0.05095s+0.000178} \quad (7)$$

Fungsi Alih dengan menggunakan dua *zero* dan dua *pole* ini memperoleh nilai *fit to estimation data* sebesar: **94.24%**, yaitu 0.48% lebih tinggi dibandingkan dengan fungsi alih yang menggunakan satu *zero*.

Fungsi alih pada persamaan (7) karena memiliki nilai *fit to estimation data* paling tinggi yaitu sebesar 94.24%. Dimana selanjutnya untuk penentuan parameter PID dengan posisi sensor 1 didapatkan melalui persamaan fungsi alih dengan 2 *pole* dan 2 *zero* ini.

Selanjutnya dilakukan penentuan parameter pada sistem airflow dengan posisi sensor 2, didapatkan data sebagai berikut:

$$\text{Fungsi Alih Airflow Posisi 2} = \frac{0.101s+0.003035}{s^2+0.09348s+0.001762} \quad (8)$$

Fungsi Alih dengan menggunakan satu *zero* dan dua *pole* memperoleh nilai *fit to estimation data*: 86.39%.

$$\text{Fungsi Alih Airflow Posisi 2} = \frac{1.503s^2+0.06589s+0.001319}{s^2+0.04311s+0.0007662} \quad (9)$$

Fungsi Alih dengan menggunakan dua *zero* dan dua *pole* memperoleh nilai *fit to estimation data*: 87.98%. Fungsi alih ini akan digunakan untuk proses selanjutnya karena memiliki nilai *fit to estimation data* lebih besar

Selanjutnya dilakukan penentuan parameter pada sistem airflow dengan posisi sensor 3, didapatkan data sebagai berikut:

$$\text{Fungsi Alih Airflow Posisi 3} = \frac{6.198s+0.1337}{s^2+3.639s+0.07747} \quad (10)$$

Fungsi Alih dengan menggunakan satu *zero* dan dua *pole* memperoleh nilai *fit to estimation data*: 87.01%

$$\text{Fungsi Alih Airflow Posisi 3} = \frac{1.502s^2+0.07083s+0.001435}{s^2+0.04636s+0.0008329} \quad (11)$$

Fungsi Alih dengan menggunakan dua *zero* dan dua *pole* memperoleh nilai *fit to estimation data*: 88.52% Fungsi alih ini akan digunakan untuk proses selanjutnya karena memiliki nilai *fit to estimation data* lebih besar.

B. Penentuan Parameter PID Airflow Sistem

Dengan menggunakan simulink pada Matlab, maka dibentuklah blok diagram dengan memasukkan fungsi alih pada persamaan (7), (9), dan (11) kedalam blok simulink serta melakukan tuning pada blok PID berbasis *PID Tuner* seperti yang ditunjukkan pada gambar 3.24.

The image displays three identical screenshots of the PID Tuner interface. Each screenshot shows the following parameters:

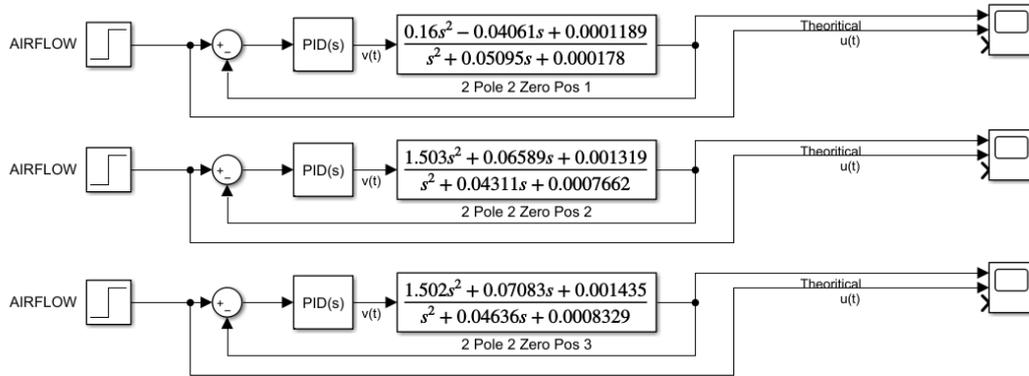
- Source: internal
- Proportional (P): 0.489144105380056
- Integral (I): 0.00182841886270847
- Derivative (D): -15.643543029223
- Filter coefficient (N): 0.0020999797642787
- Select Tuning Method: Transfer Function Based (PID Tuner App)

The Derivative (D) field is highlighted in yellow in the second and third screenshots.

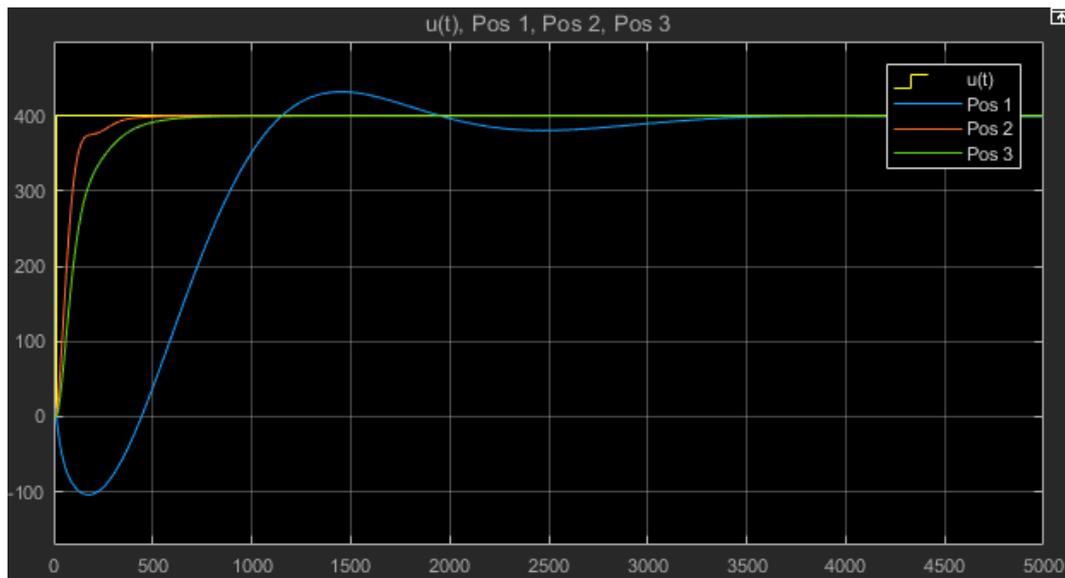
Gambar 3.23 Gambar Jendela PID Tuner Airflow dengan Posisi Sensor 1, 2, dan 3

Setelah itu dilakukan lagi pengujian berupa membandingkan hasil Teori dari ketiga posisi sensor, dengan memasukkan nilai nilai dari **K_p**, **K_i**, dan **K_d** yang didapatkan dari tuning PID, ke blok diagram pada simulink, lalu ketiga respon

dibandingkan. Diagram blok yang digunakan untuk membandingkan ketiga sistem ini dapat dilihat pada gambar 3.25, serta hasil perbandingan grafiknya dapat dilihat pada gambar 3.26.



Gambar 3.24 Blok Diagram Perbandingan Respon Airflow



Gambar 3.25 Respon Sistem Airflow

3.4.3 Parameter Nutrisi Sistem

Parameter *TDS* atau *Total Dissolved Solids* pada sistem ini dicari dengan penentuan fungsi alih terlebih dahulu, kemudian memasukkannya kedalam diagram blok pada simulink, serta melakukan *tuning* PID agar sistem mendapatkan respon sesuai dengan yang diharapkan.

A. Parameter PID Nutrisi

Sama halnya dengan sistem pencahayaan dan *aiflow*, penentuan parameter pada sistem nutrisi juga dilakukan dengan menggunakan *Matlab Identification Toolbox*, dan memasukkan *input* dan *output* yang dibaca dari sensor pada sistem, didapatkan fungsi alih orde dua dengan satu dan dua *zero*, antara lain hasil yang didapatkan sebagai berikut:

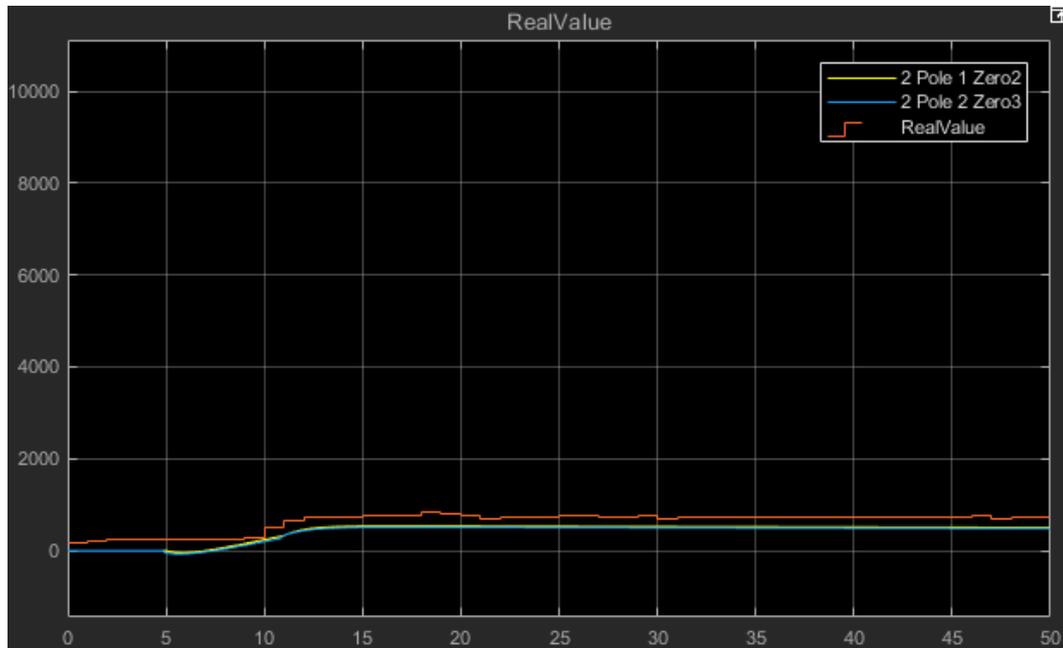
$$\text{Fungsi Alih Nutrisi} = \frac{-1.168s+0.875}{s^2+0.9728s+0.001715} \quad (12)$$

Fungsi Alih dengan menggunakan satu *zero* dan dua *pole*, memperoleh nilai *fit to estimation data*: **85.07%**, kemudian dicoba kembali menggunakan fungsi alih dengan dua *zero* dan dua *pole*, kemudian didapatkan hasil sebagai berikut:

$$\text{Fungsi Alih Nutrisi} = \frac{-0.5642s^2-1.089s+0.71}{s^2+0.831s+0.001444} \quad (13)$$

Fungsi Alih dengan menggunakan dua *zero* dan dua *pole* ini memperoleh nilai *fit to estimation data* sebesar: **85.76%**, yaitu 0.69% lebih tinggi dibandingkan dengan fungsi alih yang menggunakan satu *zero*. Untuk memastikan estimasi yang dihasilkan dari *Matlab Identification Toolbox* ini sesuai dengan respon sistem

aslinya, maka dilakukan perbandingan menggunakan grafik. Pada grafik dibandingkan ketiga nilai yang berbeda antara nilai asli sistem, fungsi alih 1, dan 2 zero.



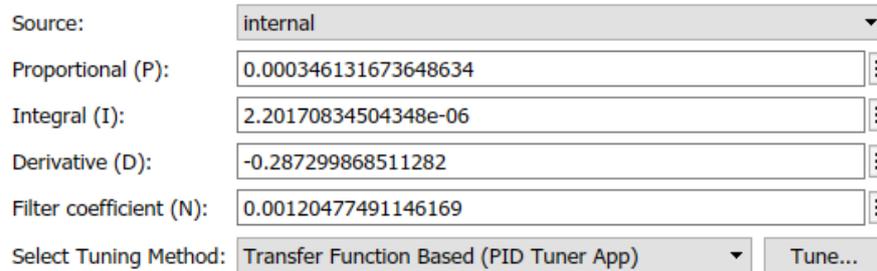
Gambar 3.26 Grafik Perbandingan Fungsi Alih Nutrisi dengan 1 dan 2 Zero
Time Sample = 1s

Estimasi fungsi alih sistem mendapatkan ketiga grafik memiliki perbedaan yang tidak terlalu jauh, oleh karena itu, digunakanlah fungsi alih pada persamaan (13) karena memiliki nilai *fit to estimation data* paling tinggi yaitu sebesar **85.76%**. Dimana selanjutnya untuk penentuan parameter PID didapatkan melalui persamaan fungsi alih dengan 2 *pole* dan 2 *zero* ini.

B. Penentuan Parameter PID Nutrisi Sistem

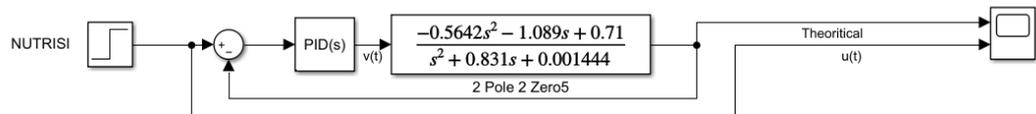
Dengan menggunakan simulink pada Matlab, maka dibentuklah blok diagram dengan memasukkan fungsi alih pada persamaan (13) kedalam blok simulink serta

melakukan tuning pada blok PID berbasis PID Tuner seperti yang ditunjukkan pada gambar 3.28.



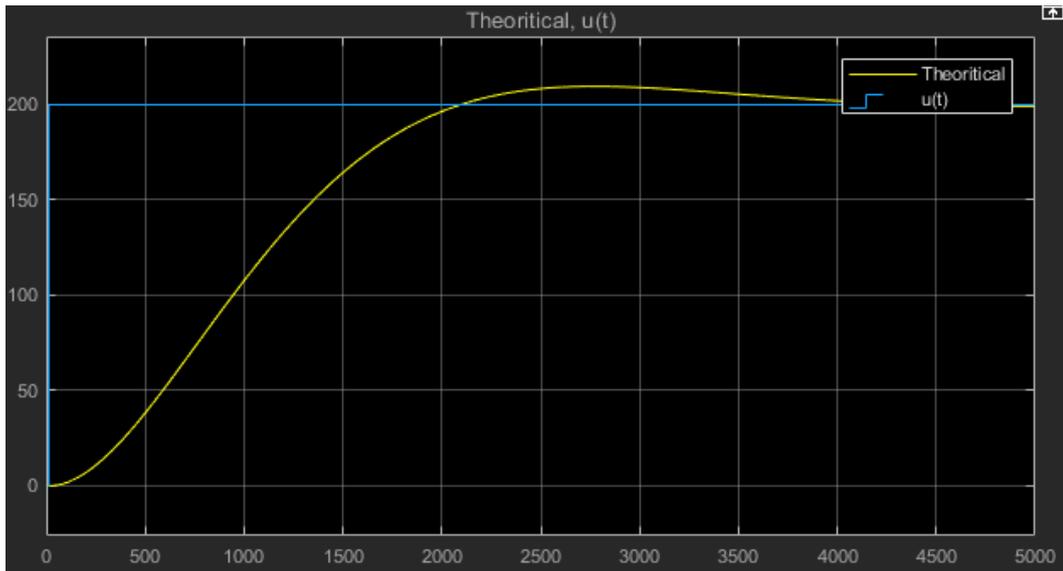
Gambar 3.27 Gambar Jendela PID Tuner Nutrisi

Setelah itu dilakukan lagi pengujian untuk melihat respon sistem, dengan memasukkan nilai nilai dari **Kp = 0.000346131673648634; Ki = 2.20170834504348e-06; dan Kd = -0.287299868511282**, yang didapatkan dari tuning PID, ke blok diagram pada simulink. Diagram bloknya dapat dilihat pada gambar 3.29



Gambar 3.28 Blok Diagram Respon Sistem Nutrisi

Hasil pengujian menunjukkan grafik sesuai yang diharapkan dengan respon sistem yang tidak terlalu cepat dan tidak terdapat overshoot yang banyak mengingat kapasitas tangki reservoir yang terbatas. Dengan begitu parameter yang didapatkan cocok dan cukup baik untuk menangani spesifikasi sistem yang diharapkan.



Gambar 3.29 Respon Sistem Nutrisi