

**LAPORAN AKHIR  
PENELITIAN KOMPETITIF  
PENELITIAN DASAR**



**UBAYA**  
UNIVERSITAS SURABAYA

**PENGEMBANGAN SMART HYDROPONICS SYSTEM**

Ketua : Agung Prayitno, S.T., M.Eng (204037/0725107901)  
Anggota : Ir. Veronica Indrawati, M.T. (194029/0708046601)

**Universitas Surabaya  
November 2018**

## HALAMAN PENGESAHAN

Judul Penelitian : Pengembangan Smart Hydroponics System

Nama Rumpun Ilmu : Teknik Elektro

Ketua Peneliti :

- a. Nama Lengkap : Agung Prayitno, S.T., M.Eng.
- b. NIDN : 204037 / 0725107901
- c. Jabatan Fungsional : Lektor-200
- d. Fakultas /Program Studi : Fakultas Teknik / Teknik Elektro
- e. Nomor HP : 087854417676
- f. Alamat Surat (email) : prayitno\_agung@staff.ubaya.ac.id

Anggota Peneliti :

- a. Nama Lengkap : Ir. Veronica Indrawati, M.T.
- b. NIDN : 194029 / 0708046601
- c. Fakultas /Program Studi : Fakultas Teknik / Teknik Elektro

Lama Penelitian Keseluruhan : 1 tahun

Penelitian Tahun ke : 1 dari 1 tahun

Biaya Penelitian Keseluruhan : Rp. 15.000.000,-

Mengetahui,  
Dekan Fakultas Teknik,



(Dr. Drs. Amelia Santoso, M.T.)  
NPK: 193015

Surabaya, 29 Januari 2018

Ketua Peneliti,

(Agung Prayitno, S.T., M.Eng.)  
NPK: 204037

## DAFTAR ISI

Halaman Sampul.....	i
Halaman Pengesahan .....	ii
Daftar Isi .....	iii
Ringkasan .....	1
Bab I Pendahuluan .....	2
1.1. Latar Belakang dan Garis Besar Penelitian .....	2
1.2. Perumusan Masalah .....	3
1.3. Tujuan Khusus .....	3
1.4. Urgensi (Keutamaan) Penelitian .....	3
Bab II Tinjauan Pustaka .....	4
2.1. Penelitian Tentang Smart Hydroponics .....	5
2.2. Penelitian Pendahuluan Pengusul .....	5
Bab III Metode Penelitian .....	6
3.1. Pembuatan Kebun .....	7
3.2. Perancangan Sistem Kontrol .....	8
3.2.1. Sistem Kontrol Dengan Menggunakan PLC dan HMI Panel .....	8
3.2.1.1. pH Sensor .....	9
3.2.1.2. EC Sensor .....	9
3.2.1. 3. Analog level Sensor .....	10
3.2.1.4. Rangkaian Relay dan Penurun Tegangan .....	11
3.2.1.5. Mixer .....	12
3.2.1.6. Solenoid valve .....	12
3.2.1.7. Pompa .....	12
3.2.1.8. Wiring I/O pada PLC .....	13
3.2.19. Rancangan Program .....	14
3.2.10. Rancangan HMI .....	15
3.2.2. Sistem Kontrol Dengan Menggunakan Arduino, Raspberry Pi dan Mobile Application .....	15
3.2.2.1. Sistem Secara Umum .....	16
3.2.2.2. Perancangan Sistem Komunikasi .....	16
3.2.2.3. Perancangan Jaringan Komunikasi SZ03 .....	17

3.2.2.4. Perancangan Sistem Pada Master (Raspberry Pi) .....	19
3.2.2.5. Perancangan Sistem Pada Slave (Arduino) .....	20
3.2.2.6. Perancangan Database .....	27
3.2.2.7. Perancangan Mobile Application .....	28
Bab IV Hasil Pengujian Sistem .....	31
4.1. Pengujian Sistem Kontrol Dengan PLC dan HMI Panel .....	31
4.1.1. Pengujian loop control level air .....	31
4.1.2. Pengujian loop control nilai pH air .....	32
4.1.3. Pengujian loop control nutrisi dalam air ..	33
4.1.4. Pengujian keseluruhan .....	34
4.2. Pengujian Sistem Kontrol Dengan Menggunakan Arduino, Raspberry Pi dan Mobile Application .....	35
4.2.1. Pengujian Input Setpoint .....	35
4.2.2. Pengujian Database .....	36
Bab V Kesimpulan .....	38
Daftar Pustaka .....	39

## RINGKASAN

Laporan kemajuan ini menjelaskan tentang kemajuan penelitian “Pengembangan Smart Hydroponics System”. Penelitian ini merancang *smart hydroponics system* sebagai solusi untuk bercocok tanam modern dengan media air yang cocok untuk diterapkan pada daerah perkotaan yang mempunyai lahan sempit dengan memanfaatkan atap-atap gedung yang kosong untuk ruang produktif pertanian. *Smart hydroponics system* ini dilengkapi dengan *smart control* yang dapat mengatur pH air, nutrisi, dan kontrol kelembaban sesuai dengan karakteristik pertumbuhan tanaman yang akan ditanam, sehingga sistem ini dapat menghasilkan tanaman hidroponik yang berkualitas. Tahapan penelitian yang sudah dilakukan adalah pembuatan kebun, green house, perancangan kontrol system untuk pH, level air, nutrisi, dan kelembapan.

**Kata Kunci:** *smart hydroponics system*, kontrol pH, kontrol ppm

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang dan Garis Besar Penelitian

Indonesia merupakan negara agraris dimana sampai saat ini mayoritas penduduk di Indonesia masih menggantungkan hidup pada sektor pertanian. Berdasarkan data yang diperoleh dari Badan Pusat Statistik (BPS) pada sensus pertanian tahun 2013 jumlah petani di Indonesia mencapai 31,70 juta jiwa [1]. Negara Indonesia diuntungkan karena terletak di daerah beriklim tropis yang dilewati oleh garis khatulistiwa, dimana sinar matahari dapat menyinari sepanjang tahun sehingga dapat menanam sepanjang tahun. Meski Indonesia adalah negara pertanian, tetapi sejauh ini masih banyak berbagai masalah pertanian di Indonesia yang membuat sektor pertanian tersebut belum berkembang.

Sistem pertanian di Indonesia hingga saat ini masih bersifat konvensional, dimana sistem pertanian masih didominasi oleh sistem pertanian tradisional. Sistem pertanian konvensional di Indonesia mempunyai banyak permasalahan umum antara lain musim kemarau yang panjang yang menyebabkan kekeringan lahan, penggunaan teknologi yang masih sederhana, pengolahan tanah setelah panen memerlukan waktu yang lama agar tanah dapat ditanami kembali, serta banyaknya rumput atau tanaman liar pengganggu yang dapat tumbuh dengan mudahnya, tanaman yang ditanam juga rentan terhadap serangan hama wereng, dan hama lainnya. Asupan nutrisi tanaman dan tingkat keasaman tanah pada sistem pertanian konvensional masih sulit untuk dimonitoring. Selain permasalahan tersebut, pertanian di Indonesia masih terdapat permasalahan lain yaitu lahan yang semakin berkurang karena beralih fungsi menjadi perumahan dan area komersil.

Di daerah perkotaan khususnya kota-kota besar, lahan pertanian konvensional sudah mulai berkurang, bahkan hampir tidak ada lahan pertanian karena pesatnya pembangunan perumahan, apartemen, maupun gedung-gedung baru. Oleh karena itu profesi sebagai petani hampir tidak ada sehingga kebutuhan akan sayur-sayuran pun didatangkan dari luar kota. Dalam kesempatan ini penulis akan memberikan solusi pada permasalahan tersebut dengan mengembangkan teknik pertanian modern yang dapat diterapkan di kota-kota besar sebagai alternatif untuk bercocok tanam. Teknik bercocok tanam yang dapat diterapkan di kota-kota besar adalah dengan menggunakan metode hidroponik. Metode bercocok tanam ini tidak menggunakan media tanah, tetapi memanfaatkan air sebagai media tanam.

Sistem bercocok tanam *Smart Hydroponics System* merupakan solusi di atas, ide dan gagasan ini juga muncul dari keinginan pengusul dan tim untuk memanfaatkan atap-atap

gedung di kampus Universitas Surabaya yang memungkinkan untuk pertanian modern. Kami memilih atap-atap gedung atau atap rumah karena permukiman penduduk di kota-kota besar cenderung rapat dan mempunyai halaman yang sempit, sehingga dapat memanfaatkan atap gedung untuk tempat produktif pertanian. Dengan teknik bercocok tanam ini penulis berharap dapat menjadi salah satu solusi bagi pemenuhan kebutuhan pangan di perkotaan, minimal kebutuhan sayuran harian keluarga. Sehingga kebutuhan akan sayuran segar tidak perlu menempuh jarak ratusan kilometer untuk sampai di meja makan warga kota.

## 1.2. Perumusan Masalah

Berdasar latar belakang di atas, permasalahan pada penelitian ini dapat dirumuskan sbb:

- a. Bagaimana memanfaatkan atap gedung perkotaan untuk bercocok tanam dengan menggunakan teknik hidroponik?
- b. Bagaimana merancang *smart system* untuk kontrol dan monitoring pertanian teknik hidroponik?
- c. Bagaimana merancang aplikasi *smartphone* dengan berbagai fitur yang mendukung dan memudahkan pertanian teknik hidroponik?

## 1.3. Tujuan Khusus

Tujuan penelitian ini adalah memanfaatkan tempat yang ada namun belum dimanfaatkan, seperti atap gedung atau atap rumah di daerah perkotaan sebagai tempat untuk bercocok tanam dengan teknik hidroponik. *Smart Hydroponics System* ini mempunyai fitur sebagai berikut:

- Kemudahan dalam menentukan data kebutuhan tiap tanaman yang ditanam karena sekali klik pada menu aplikasi, sistem secara otomatis akan mengontrol dan memonitoring tanaman yang dipilih dengan data pH, data kepekatan nutrisi, suhu dan kelembaban yang sesuai untuk tanaman tersebut.
- Kontrol dan *monitoring* tingkat keasaman media tanam air, kepekatan larutan nutrisi, suhu dan kelembaban kebun melalui aplikasi *smartphone*. Sehingga bertani menjadi lebih fleksibel dan mudah.

## 1.4. Urgensi (Keutamaan) Penelitian

Urgensi dari penelitian ini adalah bahwa penelitian ini adalah penelitian aplikatif untuk membawa teknologi dalam peningkatan ketersediaan pangan melalui pertanian.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Tentang Smart Hydroponics**

Selain mempunyai beberapa keunggulan, bukan berarti sistem hidroponik tidak mempunyai kelemahan. Kelemahan inilah yang membuat tanaman hidroponik tidak dapat tumbuh secara maksimal. Pada sistem hidroponik, kelemahan yang sering muncul adalah pH larutan yang sering berubah sepanjang hari karena aktivitas biologis dari tanaman maupun kondisi lingkungan dan kadar nutrisi dalam air berkurang karena terserap akar tanaman. Jika pH larutan terlalu asam maupun terlalu basa maka akan menyebabkan tanaman hidroponik tidak dapat tumbuh dengan baik, karena larutan nutrisi dapat larut dengan baik pada *range* pH tertentu sehingga dapat terserap dengan baik oleh akar tanaman. Sedangkan kadar nutrisi juga perlu dijaga karena kadar nutrisi sebanding dengan *electrical conductivity* dari larutan. Hal ini penting karena aktivitas penyerapan nutrisi oleh akar adalah aktivitas ion. Selain itu jika sistem hidroponik diletakkan diluar ruangan (*outdoor*) maka tumbuhan hidroponik akan rentan terhadap panas dan kelembaban pada siang hari. Paparan panas matahari langsung dan kurangnya kelembaban akan mengakibatkan daun pada tanaman menjadi layu bahkan bisa membuat daun tumbuhan menjadi kering dan mati. Masalah lain yang sering menjadi kendala adalah setiap jenis tanaman mempunyai kebutuhan pH, kepekatan larutan nutrisi, *electrical conductivity* yang berbeda beda. Hal tersebut menuntut petani untuk cermat dan rutin untuk memonitor dan mengontrol variabel tersebut. Selama ini hal tersebut masih dilakukan manual dan cenderung tanpa perekaman data dengan baik. Karena manual petani akan kesulitan memonitor dan mengontrol kebun hidroponiknya jika berada di luar kota.

Beberapa penelitian tentang smart hydroponics telah dilakukan oleh beberapa peneliti di dunia. Dan Wang et al [2] mengembangkan sebuah desain Smart Monitoring and Control System untuk Aquaponics berbasis OpenWrt. System ini terbagi menjadi 3 bagian yaitu data akuisisi, mobile transfer dan aplikasi yang interaktif. Data yang didapat dari webcam dan sensor akan di analisa dan diproses oleh operator/pengguna, sedangkan pengguna juga dapat menggunakan smartphone untuk memonitor maupun mengontrol smart aquaponics tersebut. Kekurangan dari system ini adalah tidak adanya system cerdas yang dapat mengontrol system tersebut saat tidak adanya operator, sehingga keterlibatan operator masih tergolong sangat banyak dalam mengoperasikan system tersebut. Ageng Al Hilal Gajahyana Yosi[3] mengembangkan sebuah Prototype Hydroponic Greenhouse's Smart Controller yang berbasis ATMEGA328P dengan menggunakan teknologi Bluetooth secara keseluruhan, prototype ini



mengendalikan system greenhouse secara otomatis dengan pemrograman C pada mikrokontroler ATMEGA328P. data yang didapat oleh sensor juga dapat di monitor pada smartphone melalui bluetooth. Adapun kekurangan pada sistem ini adalah penggunaan bluetooth sebagai penyalur data untuk mobile application sehingga data tidak bisa didapat bila pengguna sedang berada jauh dari lokasi greenhouse.

## **2.2. Penelitian Pendahuluan Pengusul**

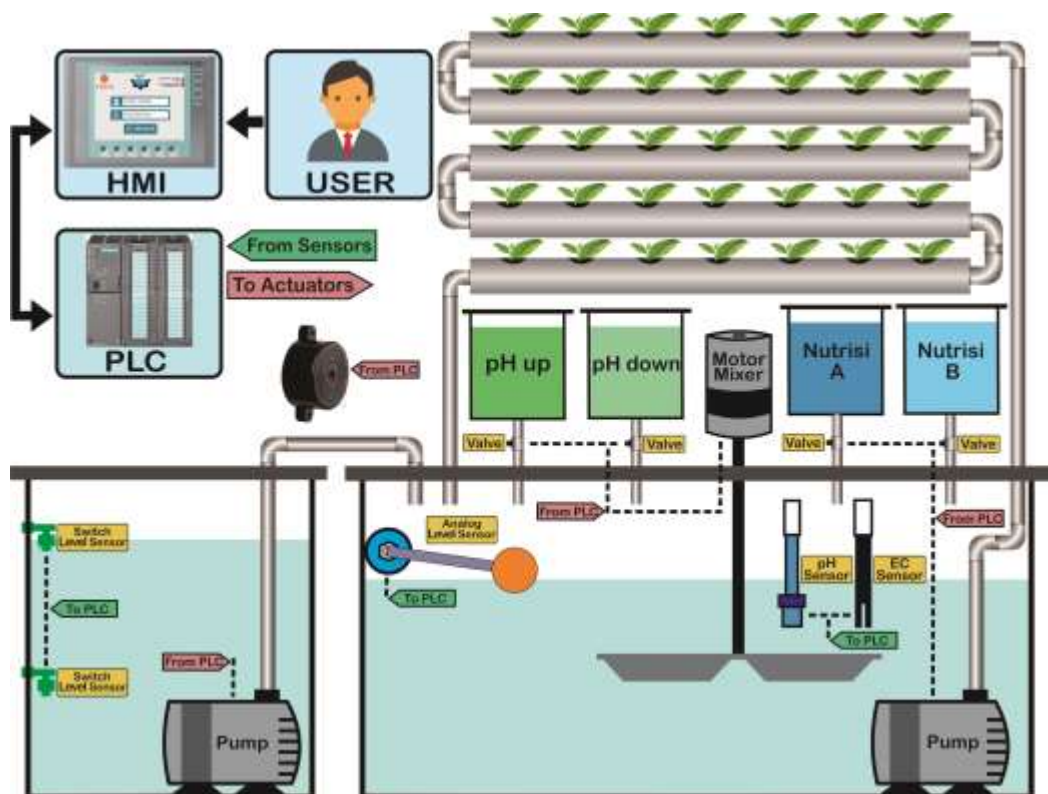
Penelitian di bidang smart system hydroponics ini merupakan penelitian pertama dari pengusul dalam aplikasi pertanian modern. Namun dengan pengalaman penelitian, pengusul yakin dapat menyelesaikan penelitian ini dan mendapatkan output penelitian yang lebih nyata.

Selama ini pengusul dan beberapa dosen di prodi TE Ubaya fokus pada pengembangan algoritma kontrol dan aplikasinya pada quadrotor. Sejak tahun 2013, pengusul (Agung Prayitno dan Veronica Indrawati) sudah fokus pada penelitian di bidang AR.Drone. Agung Prayitno [5] mendapatkan model dinamik AR.Drone 1.0 dengan pendekatan sistem fisik. Veronica Indrawati [6] dalam penelitiannya fokus pada pengembangan algoritma speech recognition yang akan digunakan sebagai pengganti dari remote control. Tahun 2014 dan 2015 pengusul dan anggota mendapatkan hibah penelitian DIKTI melalui skema Hibah Bersaing. Dalam penelitian ini, model dinamik AR.Drone 2.0 berhasil diperoleh dengan pendekatan data modeling [7] dan skema kontrol Fuzzy Logic Controller telah berhasil diaplikasikan untuk trajectory tracking [4]. Veronica [8] sukses mengimplementasikan waypoint navigation pada bidang x-y-z dan membandingkan skema beberapa skema fuzzy untuk position control dari drone [9]. Agung [10] merancang PID controller dan membandingkannya dengan [9]. Agung [11] mengimplementasi H-Infinity control untuk pitch dan roll dari AR.Drone.

### BAB III METODE PENELITIAN

Pengembangan Smart Hydroponics System dalam penelitian ini direalisasikan dengan menggunakan PLC/HMI dan menggunakan Arduino.

Skema Hydroponics System yang akan dirancang ditunjukkan pada Gambar 3.1. Secara umum sistem berisikan kebun hidroponik, bak nutrisi, sistem kontrol dan human machine interface. Kebun hidroponik dibuat dengan sistem pengairan Deep Flow Technique (DFT) dimana pipa paralon hidroponik disusun dalam posisi horisontal. Sistem ini memiliki keunggulan ketika listrik/pompa mati, di dalam pipa paralon masih terdapat sisa air yang memungkinkan sayuran untuk bertahan hidup selama berjam jam. Terdapat 3 loop kontrol dari sistem ini yaitu loop kontrol pH, loop kontrol nutrisi yang dinyatakan dalam electric conductivity larutan dan loop kontrol level air pada bak. Untuk mempermudah monitoring dan kontrol dibuat human machine interface dengan menggunakan HMI touch panel.



Gambar 3.1 Skema Smart Hydroponics System

### 3.1. Pembuatan Kebun

Bahan yang diperlukan untuk pembuatan kebun hidroponik adalah pipa PVC, penutup pipa, dan *netpot*. Pipa PVC yang digunakan mempunyai diameter 2,5 inch dengan panjang 2 m. Pipa ini kemudian dilubangi di sepanjang pipanya dengan diameter lubang 4,5 cm dengan jarak antar lubang 4.5 cm. Setiap lubang yang dibuat digunakan untuk menempatkan *netpot* (pot jaring) tanaman. Satu pipa berukuran 2 meter tersebut dapat menampung 13 *netpot*. Sistem ini memakai 5 tingkat pipa sehingga total keseluruhan dapat menampung 65 *netpot*. Pipa antar tingkat saling terhubung dengan menggunakan pipa 1 inch dan penutup pipa yang telah dimodifikasi. Penutup pipa tersebut dilubangi dan dipasang *fitting* yang berdiameter 1 inch. Hasil pembuatan kebun ditunjukkan pada Gambar 3.2.

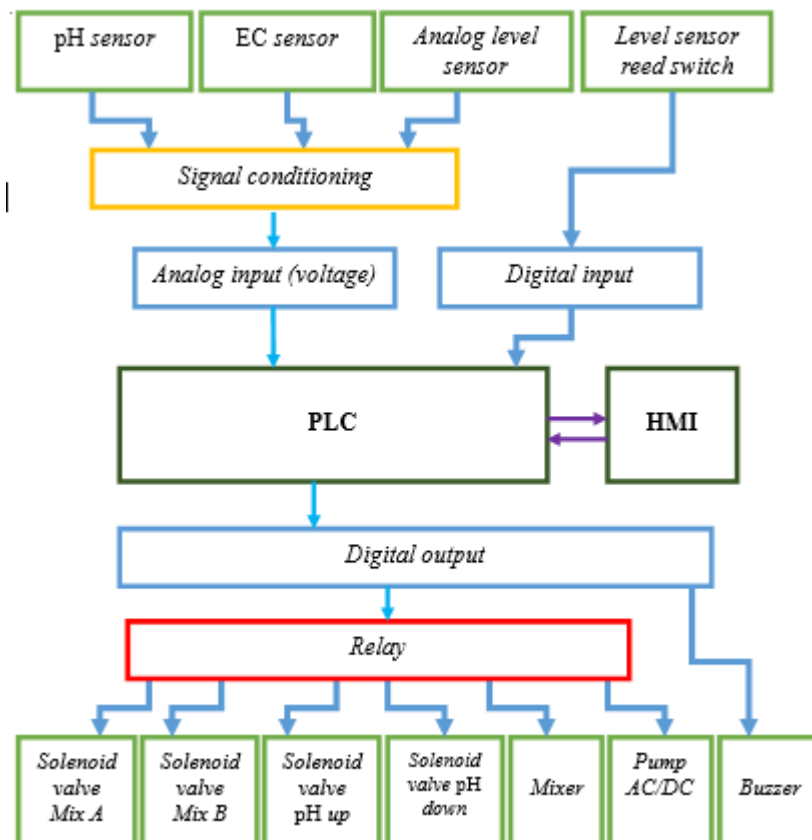


Gambar 3.2 Kebun hidroponik

## 3.2. Perancangan Sistem Kontrol

### 3.2.1. Sistem Kontrol Dengan menggunakan PLC dan HMI Panel

Secara umum skema hardware sistem kontrol yang akan dibuat untuk mengontrol kebun hidroponik ditunjukkan pada Gambar 3.3. Secara umum sistem ini terdiri dari sensor, modul input, kontroler, modul output dan human interface. Terdapat 3 buah sensor dengan luaran analog yaitu pH sensor, EC sensor dan level sensor yang masing masing memerlukan sinyal kondisioning untuk menyesuaikan dengan range dari modul input analog PLC. Sebagai batas level atas dan bawah pada bak nutrisi digunakan reedswitch yang bertipe digital. PLC yang digunakan adalah PLC Siemens S7-300 314C-2 PN/DP sedangkan HMI yang dipakai adalah HMI Siemens Touch Panel KTP600 Basic Color PN. PLC dan HMI akan diprogram dengan menggunakan *software* TIA Portal V13. Actuator yang digunakan adalah motor DC sebagai mixer, pump AC/DC, 4 solenoid valve untuk nutrisi A, nutrisi B, pH down dan pH up.



Gambar 3.2 Blok Diagram System

### 3.2.1.1. pH Sensor

Sensor pH perlu dikalibrasi dengan menggunakan larutan pH buffer. Luaran dari pH *sensor* berupa tegangan (Volt) yang kemudian diuji dengan larutan buffer menghasilkan data seperti Tabel 3.1.

**Tabel 3.1: Hasil Kalibrasi pH Sensor**

<b>pH Buffer</b>	<b>Tegangan (V)</b>	<b>PIW (Word)</b>	<b>Output Scaling Sensor (0-100)</b>
4,00	1,20	3312	11,9375
6,86	1,99	5488	19,75
9,18	2,49	6864	24,75

Output scaling sensor adalah hasil perhitungan secara software yang dibuat dengan ladder diagram. Dari tabel tersebut kemudian dibuat sinyal conditioning secara software dengan menggunakan persamaan garis lurus yakni  $y = m \cdot x + c$ . Nilai  $y$  adalah pH buffer dan  $x$  adalah output scaling sensor. Untuk mendapatkan nilai  $m$  dan  $c$  digunakan metode least square dengan menyusun data sebagai berikut:

$$\begin{matrix} y \\ \left[ \begin{array}{l} 4,00 \\ 6,86 \\ 9,18 \end{array} \right] \end{matrix} = \begin{matrix} \theta \\ \left[ \begin{array}{ll} 11,9375 & 1 \\ 19,75 & 1 \\ 24,75 & 1 \end{array} \right] \end{matrix} \begin{matrix} g \\ \left[ \begin{array}{l} m \\ c \end{array} \right] \end{matrix}$$

Estimasi nilai  $g$  dilakukan menggunakan persamaan  $g = (\theta \cdot \theta)^{-1} \cdot \theta \cdot y$ , dan mendapatkan nilai  $m = 0,4009$  dan  $c = -0,8626$ . Sehingga persamaan sinyal conditioning untuk pH sensor adalah  $y = 0,4009 \cdot x - 0,8626$  dimana  $y$  merupakan nilai pH *sensor* dan  $x$  merupakan nilai *output scaling* dari PLC. Persamaan ini digunakan untuk menghitung nilai aktual pH *sensor* yang akan ditampilkan oleh HMI.

### 3.2.1.2. EC Sensor

EC *sensor* merupakan *sensor* yang dipakai untuk mendeteksi kandungan nutrisi dalam air. Satuan EC *sensor* yang digunakan pada sistem ini adalah ms/cm. Sama seperti kalibrasi pada pH *sensor*, kalibrasi pada EC *sensor* juga menggunakan larutan *buffer*. Larutan EC *buffer* mempunyai nilai 1.413 ms/cm dan 12.88 ms/cm. Hasil yang didapatkan dari kalibrasi EC *sensor* ditunjukkan oleh Tabel 3.1.

**Tabel 3.2: Hasil Kalibrasi EC Sensor**

<b>EC Buffer</b>	<b>Tegangan (V)</b>	<b>PIW (Word)</b>	<b>Output Scaling Sensor (0-100)</b>
1,413 ms/cm	0,16	448	1,617188
12,88 ms/cm	1,38	3808	13,75

Selanjutnya dirancang sinyal conditioning secara software dengan mencari persamaan garis lurus yakni  $y = m \cdot x + c$ . Dimana nilai y adalah EC buffer sedangkan x adalah output scaling sensor.

$$\begin{matrix}
 y & & \theta & & g \\
 \begin{bmatrix} 1,413 \\ 12,88 \end{bmatrix} & = & \begin{bmatrix} 1,617188 & 1 \\ 13,75 & 1 \end{bmatrix} & \begin{bmatrix} m \\ c \end{bmatrix}
 \end{matrix}$$

Estimasi nilai  $g$  dilakukan menggunakan persamaan  $g = (\theta^T \cdot \theta)^{-1} \cdot \theta^T \cdot y$ . Dari hasil perhitungan didapatkan persamaan  $y = 0,9451 \cdot x - 0,1154$  yang nantinya digunakan untuk nilai aktual EC sensor pada HMI.

### 3.2.1.3. Analog Level Sensor

Analog level sensor menggunakan *singleturn* potensiometer karbon bernilai 5K ohm bertipe B supaya sensor bekerja secara linier. Pemilihan *singleturn* potentiometer berdasarkan ketinggian bak nutrisi yang akan dikontrol level airnya. Analog level sensor dibuat dengan menggunakan beberapa part, seperti teflon plastik, kabel *single*, baut, dan pelampung. Teflon plastik akan dibubut untuk dudukan ke pemutar potensiometer, teflon ini berfungsi sebagai penyangga tuas yang terbuat dari kabel *single* (kopling). Diujung kabel single akan dipasang pelampung bulat yang berfungsi untuk menggerakkan potensiometer melalui tuas yang dipasang.

Untuk mendapatkan luaran potensiometer berupa nilai hambatan, selanjutnya digunakan rangkaian pembagi tegangan untuk mendapatkan luaran berupa tegangan. Karena tegangan masih terlalu kecil kemudian dibuat rangkaian non-inverting amplifier sehingga didapat data seperti pada Tabel 3.3.

**Tabel 3.3: Tabel bacaan sensor level**

<b>Ketinggian Air (Cm)</b>	<b>Rancangan (V)</b>	<b>Pengukuran (V)</b>	<b>Output Scalling Sensor (1-100)</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>0.22</b>	<b>0.32</b>	<b>3.234375</b>
<b>2</b>	<b>0.22</b>	<b>0.32</b>	<b>3.234375</b>
<b>3</b>	<b>0.55</b>	<b>0.64</b>	<b>6.40625</b>
<b>4</b>	<b>0.99</b>	<b>1.2</b>	<b>11.875</b>
<b>5</b>	<b>1.43</b>	<b>1.56</b>	<b>15.5625</b>
<b>6</b>	<b>1.65</b>	<b>1.8</b>	<b>17.875</b>
<b>7</b>	<b>1.98</b>	<b>2.16</b>	<b>21.375</b>
<b>8</b>	<b>2.2</b>	<b>2.39</b>	<b>23.625</b>
<b>9</b>	<b>2.53</b>	<b>2.64</b>	<b>26.25</b>
<b>10</b>	<b>2.75</b>	<b>2.93</b>	<b>29</b>
<b>11</b>	<b>2.97</b>	<b>3.14</b>	<b>31.25</b>
<b>12</b>	<b>3.19</b>	<b>3.39</b>	<b>33.5</b>
<b>13</b>	<b>3.41</b>	<b>3.63</b>	<b>35.75</b>
<b>14</b>	<b>3.63</b>	<b>3.81</b>	<b>37.75</b>
<b>15</b>	<b>3.85</b>	<b>4.12</b>	<b>40.75</b>
<b>16</b>	<b>4.07</b>	<b>4.22</b>	<b>41.75</b>
<b>17</b>	<b>4.29</b>	<b>4.48</b>	<b>44.25</b>
<b>18</b>	<b>4.51</b>	<b>4.69</b>	<b>46.25</b>
<b>19</b>	<b>4.62</b>	<b>4.85</b>	<b>48</b>
<b>20</b>	<b>4.73</b>	<b>5.05</b>	<b>50</b>

Dengan cara yang sama akan didapatkan persamaan untuk level sensor ini  $y = 0,3926 \cdot x - 0,6713$  yang nantinya digunakan sebagai nilai aktual *analog level sensor* pada *HMI*.

#### **3.2.1.4. Rangkaian Relay dan Penurun Tegangan**

Rangkaian *relay* pada sistem ini berfungsi sebagai saklar mekanik yang memisahkan output *PLC* yang mempunyai tegangan 24VDC dengan aktuator yang membutuhkan tegangan 12VDC seperti *solenoid valve ph up*, *solenoid valve pH down*, *solenoid valve Mix A*, *solenoid valve Mix B*, *DC motor (Mixer)*, pompa DC. Sedangkan untuk pompa AC akan digunakan *relay* berjenis elektromagnetik. Untuk *power supply* 12VDC pada sistem ini akan menggunakan modul *power supply* yang terpisah dengan *power supply PLC* yang mempunyai arus 10A. Sedangkan untuk penurun tegangan akan menggunakan IC LM7805 (*regulator*)

karena mempunyai tegangan *output* yang konstan dimana 12VDC akan diubah menjadi 5VDC untuk *power supply* pada *sensor*.

#### **3.2.1.5. Mixer**

Mixer menggunakan motor DC 12 volt yang dilengkapi dengan batangan *steinless steel* yang dilapisi selang plastik agar tidak korosi dan menggunakan *propeller* plastik sebagai pengaduk larutan.

#### **3.2.1.6. Solenoid valve**

*Solenoid valve* dipasang pada tangki larutan pH *up*, pH *down*, larutan *Mix A*, dan *Mix B*. Untuk menghubungkan tangki larutan dengan *solenoid valve* maka dipasang selang silikon berdiameter 6 mm supaya tidak mudah bocor ketika dialiri oleh larutan.

#### **3.2.1.7. Pompa**

Penelitian ini akan menggunakan 2 pompa yang berfungsi untuk mengalirkan air nutrisi dan menambah air ke bak nutrisi jika pada saat pengecekan terdapat pengurangan air. Kedua pompa tersebut berjenis pompa celup, dimana cara pemakaiannya adalah *body* pompa harus dicelupkan ke dalam air. Pompa yang digunakan adalah pompa DC *brushless* 12 volt dimana pompa ini dapat mengalirkan air setinggi 3 meter dengan laju aliran 240 liter air per jamnya, sehingga dapat dipakai pada sistem hidroponik yang telah dibuat. Sedangkan untuk penambahan air pada bak nutrisi akan menggunakan pompa AC 220 volt.



### 3.2.1.8. Wiring I/O pada PLC

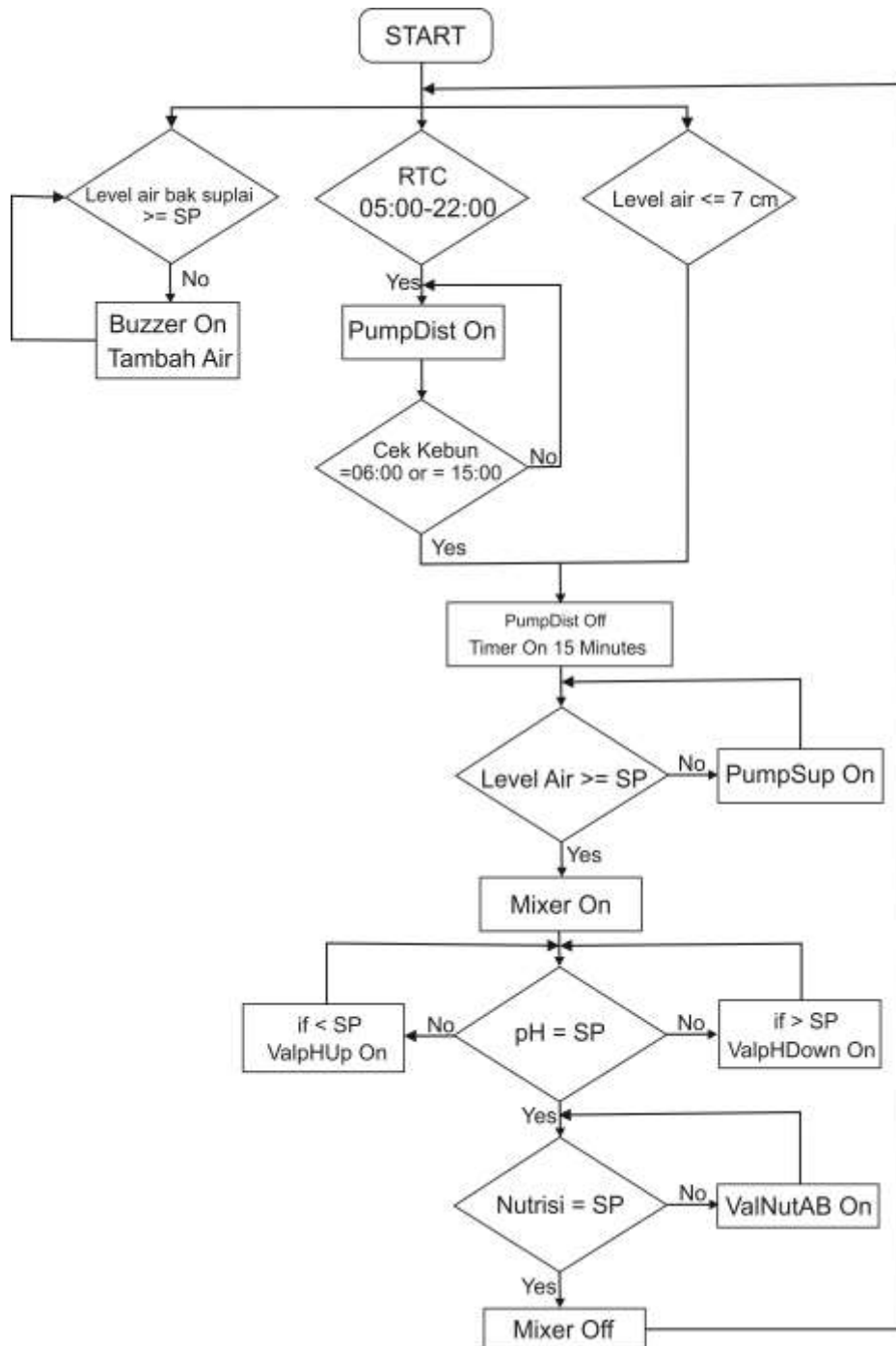
Setelah *sensor* dan aktuator sudah siap, maka selanjutnya akan dilakukan pengkabelan ke *PLC*. Untuk alamat *input/output* pada *PLC* dapat dilihat pada Tabel 3.4.

**Tabel 3.4: Pengalamatan I/O pada PLC**

No.	Jenis I/O	Nama <i>Sensor</i> Aktuator	Alamat I/O
1	<i>Input Analog</i>	<i>pH Sensor</i>	PIW800
2	<i>Input Analog</i>	<i>EC Sensor</i>	PIW802
3	<i>Input Analog</i>	<i>Level Sensor</i>	PIW804
4	<i>Input Digital</i>	<i>Level Up</i>	I136.0
5	<i>Input Digital</i>	<i>Level Down</i>	I136.1
6	<i>Output Digital</i>	<i>Pompa Distribusi</i>	Q136.0
7	<i>Output Digital</i>	<i>Motor Mixer</i>	Q136.1
8	<i>Output Digital</i>	<i>Buzzer</i>	Q136.5
9	<i>Output Digital</i>	<i>Pompa Supply Air (AC)</i>	Q136.6
10	<i>Output Digital</i>	<i>Valve AB Mix</i>	Q137.2
11	<i>Output Digital</i>	<i>Valve pH Down</i>	Q137.0
12	<i>Output Digital</i>	<i>Valve pH Up</i>	Q137.1

### 3.2.1.9. Rancangan Program

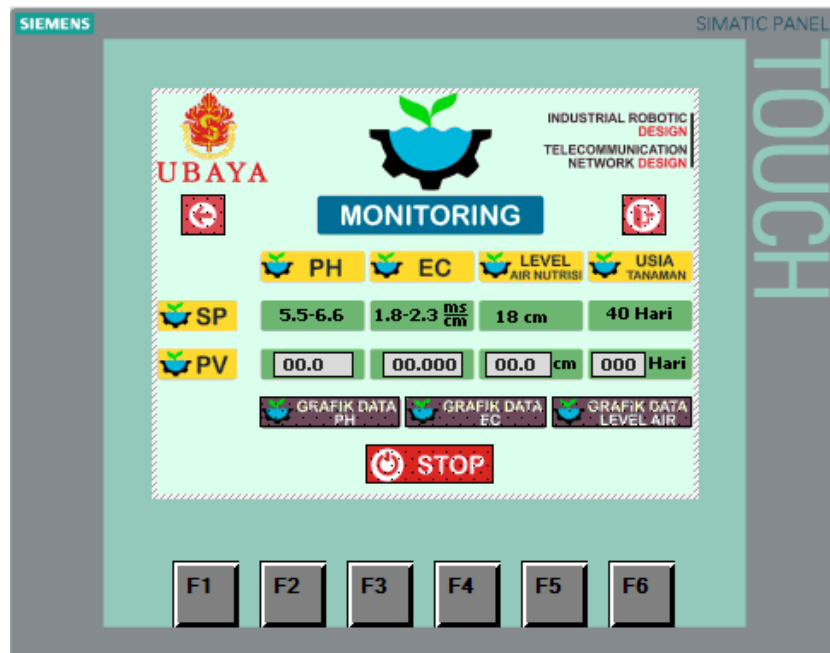
Program dibuat dengan ladder diagram PLC siemens dengan flowchart kontrol sebagai berikut.



Gambar 3.3 Flowchart Program

### 3.2.1.10. Rancangan HMI

Rancangan HMI mempunyai 5 menu utama yaitu *Login*, Menu Utama, Pilih tanaman, Info Tanaman, Lihat Kebun, *Logout*.

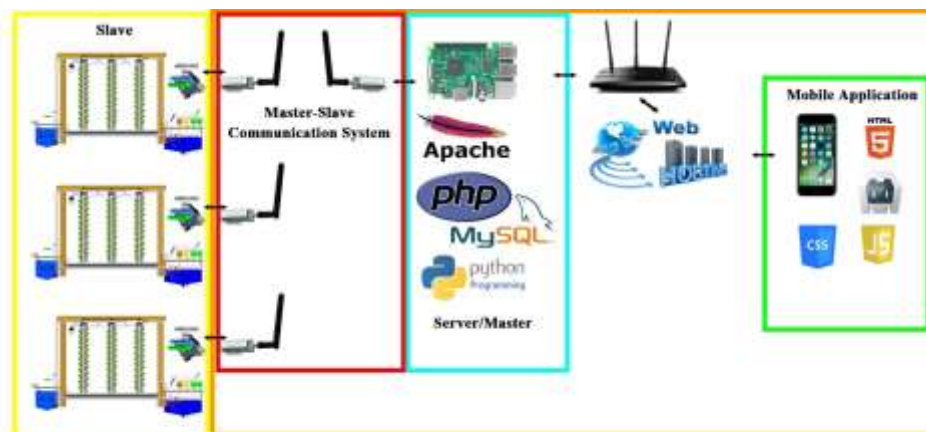


Gambar 3.4 Desain HMI

### 3.2.2. Sistem Kontrol Dengan menggunakan Arduino, Raspberry Pi dan Mobile Application

#### 3.2.2.1. Sistem Secara Umum

Perencanaan sistem kontrol ini meliputi: perencanaan program pada Arduino, perencanaan program python untuk pengolahan data dari SZ05 ke *database*, perencanaan *database* dari Raspberry Pi, perencanaan pembuatan *mobile application*. Secara umum, perencanaan sistem kontrol ini ditunjukkan pada Gambar 3.5



### Gambar 3.5. Desain sistem kontrol

Secara umum sistem pada Gambar 3.5 dapat dijelaskan sebagai berikut. Raspberry digunakan sebagai *server/master* dalam sistem ini. Sebagai *master*, Raspberry berkomunikasi dengan setiap *slave* melalui SZ05. Dalam sistem ini, *slave* merupakan kontroler (Arduino uno) untuk mengontrol variabel kebun yaitu pH, humidity, level larutan dan nutrisi tanaman.

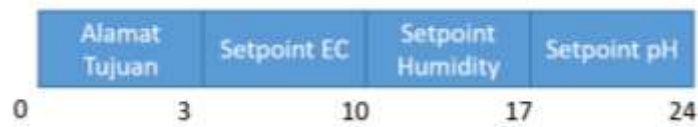
*Master* akan mengirimkan setpoint dari tabel “*spk*” pada database ke *slave*, sedangkan dari *slave* kontroler mengirimkan data sensor untuk disimpan ke dalam database dalam tabel “*dk*” dan “*cdk*”. Tabel pada database tersebut akan diakses oleh *mobile application* untuk menampilkan *present value* dari variabel, *setpoint value*, dan *historical data value*. Aplikasi ini dapat digunakan jarak jauh melalui internet, dengan mengakses Raspberry pi yang menggunakan Apache sebagai web server.

Setiap 10 menit Raspberry akan mengambil data *setpoint* semua kebun dari tabel “*spk*” kemudian dikirimkan melalui SZ05 ke *slave*. *Slave* akan mengambil data yang didapat sensor saat itu dan mengirimkan kembali kepada *master*. Data yang sudah diterima akan diolah Raspberry dan disimpan dalam tabel “*cdk*” dan “*dk*” pada *database*. *Mobile application* akan mengambil data-data yang ada pada database untuk ditampilkan pada aplikasi serta dapat merubah tabel “*spk*” pada database untuk merubah *setpoint* seperti yang diinginkan.

#### **3.2.2.2. Perancangan Sistem Komunikasi**

*Master* memberikan perintah/*command* kepada masing-masing *slave* secara bergantian dari *slave* 1 hingga terakhir. *Command* tersebut berisi *slave* tujuan, nilai *setpoint* sensor EC, nilai *setpoint* sensor humidity, dan nilai *setpoint* sensor pH. Setelah *slave* menerima perintah, maka *setpoint* yang diterima akan menggantikan *setpoint* yang lama. Kemudian *slave* akan membalas dengan mengirimkan *current value sensor* ke *master* yang kemudian diolah dan disimpan kedalam database.

Format pengiriman data dari *master* ke *slave* dapat kita lihat pada Gambar 3.6, pengiriman data berupa string sehingga 1 digit sama dengan 1 byte. Alamat tujuan dari kode dimasukkan pada byte 0 sampai 3, kemudian *setpoint* EC byte 4 sampai 10, *setpoint humidity* 11-17 dan *setpoint* pH 18-24.



Gambar 3.6: Format Data Master-Slave

Untuk mengirimkan kode dari *slave* ke *master*, format data yang digunakan adalah seperti yang ada pada Gambar 3.7, byte ke 0 sampai ke 6 berisikan data sensor EC, byte 7 hingga 13 adalah data sensor *humidity* dan byte 14 hingga 20 adalah data sensor pH.



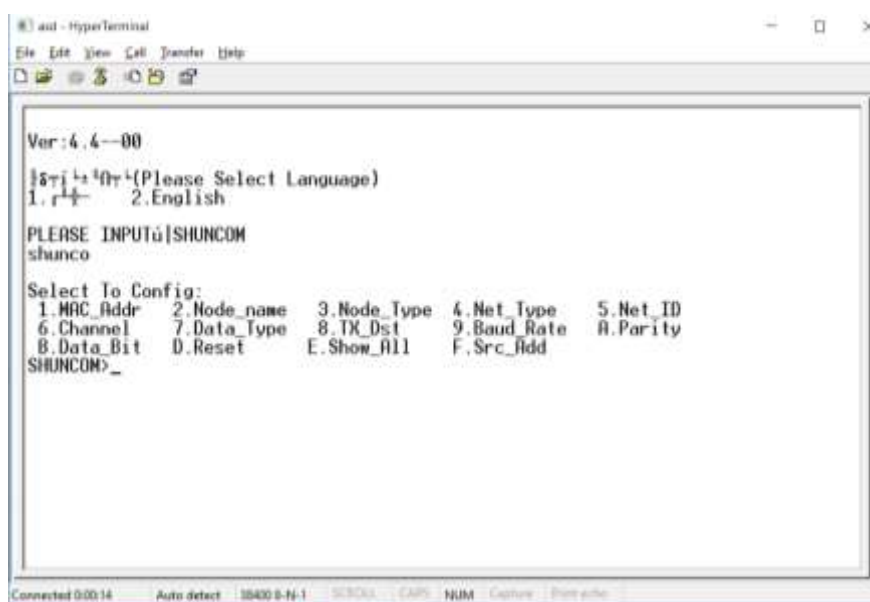
Gambar 3.7: Format Data Slave-Master

### 3.2.2.3. Perancangan Jaringan Komunikasi SZ05

Untuk menggunakan SZ05, diperlukan *development board* dan *hyperterminal* untuk melakukan konfigurasi. Untuk melakukan konfigurasi, sambungkan module SZ05 yang ingin dikonfigurasi ke *development board* yang sudah tersambung pada komputer/laptop. Kemudian buka aplikasi *hyperterminal*, lalu set baudrate di 38400, data bit 8, parity NONE, stop bit 1 dan flow NONE. Setelah itu tekan tombol CFG seperti yang ada pada Gambar 3.8 selama 3 detik. Setelah tombol CFG ditekan selama 3 detik, ketik 2 untuk melakukan pengaturan dalam bahasa inggris. Kemudian ketikkan SHUNCOM untuk masuk dalam mode pengaturan. Setelah itu akan muncul tampilan seperti pada Gambar 3.9 untuk melakukan konfigurasi.



Gambar 3.8. Tombol CFG pada Development Board



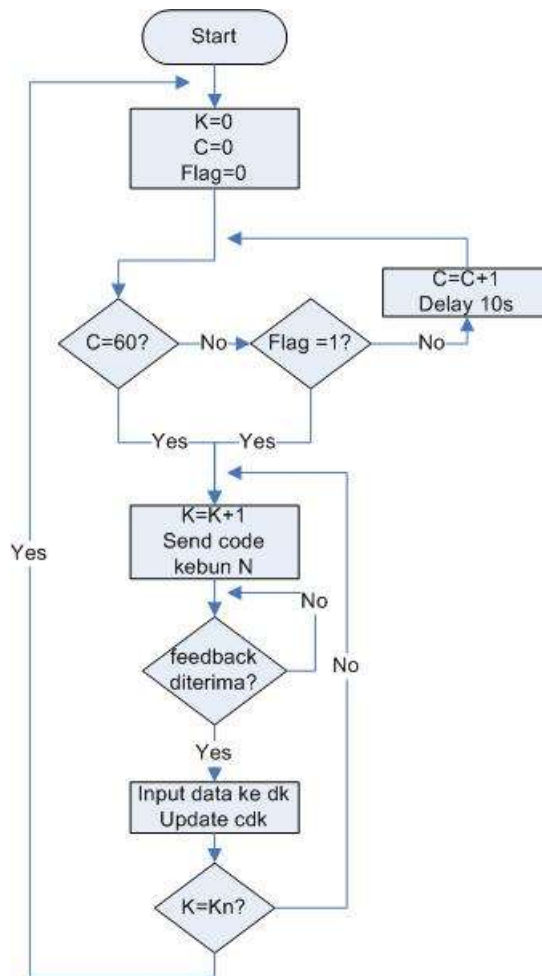
Gambar 3.9. Tampilan Konfigurasi SZ05

Konfigurasi yang disediakan adalah konfigurasi untuk *Mac address* untuk alamat dari setiap SZ05 dari 0000-FFFE. Untuk SZ05 yang digunakan oleh koordinator harus menggunakan alamat 0000. Node name hanya digunakan untuk nama dari SZ05. Untuk node type, ada 3 pilihan yang bisa digunakan, antara lain PAN\_Coord untuk menjadikan SZ05 sebagai koordinator, router untuk memberikan SZ05 fungsi sebagai router yaitu mengirimkan kembali pesan yang diterima ke alamat yang dituju bila jarak dari SZ05 tidak mencukupi dan *end device* sebagai *slave* (Arduino). Kemudian pada konfigurasi *network*, SZ05 sudah menyediakan 4 protokol jaringan yaitu *mesh*, *star*, *line* dan *peer*.

Pada 1 jaringan yang sama, seluruh SZ05 harus menggunakan protokol yang sama. Lalu konfigurasi pada network ID harus disamakan untuk setiap SZ05 yang digunakan. Pada *channel setting*, terdapat 16 pilihan frekuensi untuk digunakan pada setiap SZ05, yang dianjurkan dari manual book adalah 2.425GHZ, 2.450GHZ, 2.475GHZ dan 2.480 GHZ. *Channel* ini direkomendasikan untuk menghindari interferensi dari jaringan WIFI. Pada data type terdapat 2 pilihan yang dapat digunakan sesuai kebutuhan, ASCII dan HEX. Pada TX\_Type dilakukan konfigurasi untuk memilih tipe transmisi, yaitu *broadcast*, *master-slave* dan *point-point*. Untuk konfigurasi *master-slave*, *master* harus menyertakan 4 digit alamat *slave* pada setiap awal kode untuk memilih alamat yang dituju, tetapi *slave* tidak perlu menyertakan alamat dan hanya akan mengirimkan kepada *master/router*(bila ada). Kemudian pada *baudrate* dapat dipilih dari 1200 hingga 115200.

#### **3.2.2.4. Perancangan Sistem Pada Master (Raspberry Pi)**

Sistem pada Master secara garis besar dapat dilihat pada flowchart yang ada pada Gambar 3.10. Pada awal sistem, sistem akan memeriksa variabel *counter* dan *flag* yang terdapat pada database. Counter digunakan untuk menghitung waktu eksekusi data akuisisi, sedangkan flag digunakan untuk memeriksa perubahan database pada tabel bendera. Bila *counter* sudah mencapai 60 atau variabel flag bernilai 1 maka sistem akan mengirimkan kode untuk kebun 1, setelah *feedback* dari kebun 1 sudah diterima, *master* akan menginputkan data ke tabel “dk” dan melakukan perbaharuan data pada tabel “cdk”. Sistem akan melanjutkan kembali hingga kebun terakhir dan mengulang kembali sistem dari awal.



Gambar 3.10.Flowchart Master

Pada Raspberry Pi, digunakan bahasa pemrograman *python* untuk menerima data melalui SZ05. Untuk mengaktifkan serial komunikasi pada OS jessie kita harus mengaktifkan pada konfigurasi, *default port serial GPIO* pada OS jessie adalah *disabled*. Untuk mengaktifkan *port serial GPIO* maka kita harus membuka terminal pada raspberry untuk mengganti konfigurasi pada *config.txt* dengan menuliskan kode “Sudo nano /boot/config.txt”, lalu menuliskan pada *line* paling bawah “enable\_uart=1”.

Konfigurasi serial dari raspberry Pi dan SZ05 dilakukan dengan menggunakan kode seperti pada Gambar 3.11 dengan konfigurasi baudrate 9600, *parity none* dan *stop bit 1*. Kemudian, Raspberry Pi akan melihat nilai dari *flag* pada *database*, bila bernilai 0 dan variabel “hitung” dibawah 60, maka Raspberry akan menunggu selama sepuluh detik dan menambahkan



variabel “hitung” sebanyak 1. Selain itu, Raspberry akan melakukan proses data akuisisi yang sekaligus mengirimkan *setpoint* yang ada di *database* ke setiap *slave* yang dimiliki dan juga mengembalikan status *flag* dan variabel “hitung” kembali menjadi 0.

Untuk melihat nilai dari *flag* digunakan kode programing seperti pada Gambar 3.12. Baris ke-2 kode program digunakan untuk masuk ke *database* dengan username root dan password raspberry pada bagian data. Baris 4 dan 5 kode program digunakan untuk memilih kolom tanda dari tabel bendera, kemudian baris 6 dan 7 untuk memasukkan nilai dari kolom tanda ke variabel “tanda”.

```
ser = serial.Serial("/dev/ttyS0",  
                    baudrate = 9600,  
                    parity=serial.PARITY_NONE,  
                    stopbits=serial.STOPBITS_ONE,  
                    bytesize=serial.EIGHTBITS,  
                    timeout=10  
                    )
```

Gambar 3.11: Konfigurasi Komunikasi Serial pada Python

```
while 1:  
    db = MySQLdb.connect(host="localhost", user="root", passwd="raspberr", db="data")  
    cur= db.cursor()  
    time.sleep(1)  
    sql="SELECT `tanda` FROM `bendera` WHERE 1"  
    cur.execute(sql)  
    for row in cur.fetchall():  
        tanda=row[0]  
    cur.close()
```

Gambar 3.12: Pengambilan Nilai Flag dari Database

Kemudian logika untuk melakukan proses data akuisisi atau tidak dan juga untuk mengganti kembali flag pada *database* menjadi 0 dapat dilihat dari Gambar 3.13. Baris pertama adalah logika dari if statement yang digunakan, bila variabel “hitung” dibawah 60 dan variabel “tanda” adalah 0, maka variabel “hitung” akan bertambah 1 dan Raspberry akan menunggu selama 9 detik. Bila tidak ada perubahan pada *flag*, sistem ini akan melakukan data akuisisi setiap 10 menit sekali dengan perhitungan setiap *cycle* pemeriksaan 9 detik ditambah 1 detik

yang ada pada Gambar 3.12, yang dilakukan 60 kali sehingga menjadi 600 detik, yang berarti sensitifitas pemeriksaan *flag* adalah 10 detik. Pada baris ke 8 Gambar 3.10 terdapat kode untuk mengganti tabel bendera yang ada pada database.

```

if hitung <= 59 and tanda==0:
    hitung=hitung+1
    time.sleep(9)
else:
    db = MySQLdb.connect(host="localhost", user="root", passwd="raspberry", db="data")
    cur= db.cursor()
    time.sleep(1)
    sql="UPDATE `bendera` SET `tanda`=0 "
    cur.execute(sql)
    db.commit()
    cur.close()
    hitung=0

```

Gambar 3.13. IF-ELSE Statement

```

else:
    db = MySQLdb.connect(host="localhost", user="root", passwd="raspberry", db="data")
    cur= db.cursor()
    sql="SELECT `EC` FROM `spk` WHERE 1"
    cur.execute(sql)
    for row in cur.fetchall():
        EC1=row[0]
    cur.close()
    cur= db.cursor()
    sql="SELECT `HUM` FROM `spk` WHERE 1"
    cur.execute(sql)
    for row in cur.fetchall():
        HUM1=row[0]
    cur.close()
    cur= db.cursor()
    sql="SELECT `PH1` FROM `spk` WHERE 1"
    cur.execute(sql)
    for row in cur.fetchall():
        PH1=row[0]
    cur.close()
    cur= db.cursor()
    sql="SELECT `PH2` FROM `spk` WHERE 1"
    cur.execute(sql)
    for row in cur.fetchall():
        PH2=row[0]
    cur.close()
    cur= db.cursor()

```

Gambar 3.14. Pengambilan Data Set Point

Untuk mengirimkan set point saat ini kepada *slave*, Raspberry Pi mengambil *data set point* yang terdapat pada tabel “spk” dengan kode program seperti pada Gambar 3.14. pada baris 4 hingga 30. Program pada gambar tersebut adalah proses pengambilan nilai EC, pH dan *humidity* yang kemudian akan digabungkan menjadi 1 variabel yaitu kode 1 dengan kode program yang terdapat di Gambar 3.15. Pada awal setiap variabel kode terdapat kode yang bersifat konstan. Kode tersebut adalah alamat dari setiap *slave* yang akan dituju yaitu 0001, 0002 dan 0003, sedangkan Raspberry Pi tidak memiliki alamat

dikarenakan menggunakan modul koordinator SZ05.

```
kode1 = '0001%04d,%03d:%01d-%01d\n' % (EC11,HUM11,PH11,PH12)
kode2 = '0002%04d,%03d:%01d-%01d\n' % (EC21,HUM21,PH21,PH22)
kode3 = '0003%04d,%03d:%01d-%01d\n' % (EC31,HUM31,PH31,PH32)
```

Gambar 3.15 Penggabungan Data Set Point

Proses data akuisisi dilakukan dengan memanfaatkan potongan kode program yang ada pada Gambar 3.16. Kode pada baris ke 4 adalah kode untuk mengirimkan variabel “kode1” secara serial melalui SZ05 ke *slave*, variabel “kode1” berisikan alamat *slave* 1 dan *setpoint-setpoint* dari EC, Humidity dan pH. Setelah itu *slave* akan mengirim kembali *current state* dari masing-masing sensor dan diterima dengan kode pada baris ke 5 dan disimpan ke variabel “ser” pada baris 6. Kemudian baris ke 8 hingga ke 10 adalah kode untuk mengolah data yang diterima dari *slave*. Baris ke 12 hingga akhir berfungsi untuk menyimpan masing-masing data ke *database* pada tabel “dk” dan “cdk” kolom EC, HUM dan PH dengan *value* yang sudah diolah sebelumnya.

```
db = MySQLdb.connect(host="localhost", user="root", passwd="raspberrypi", db="data")
cur = db.cursor()
time.sleep(1)
ser.write(kode1)
x=ser.readline(21)
ser.flushInput()

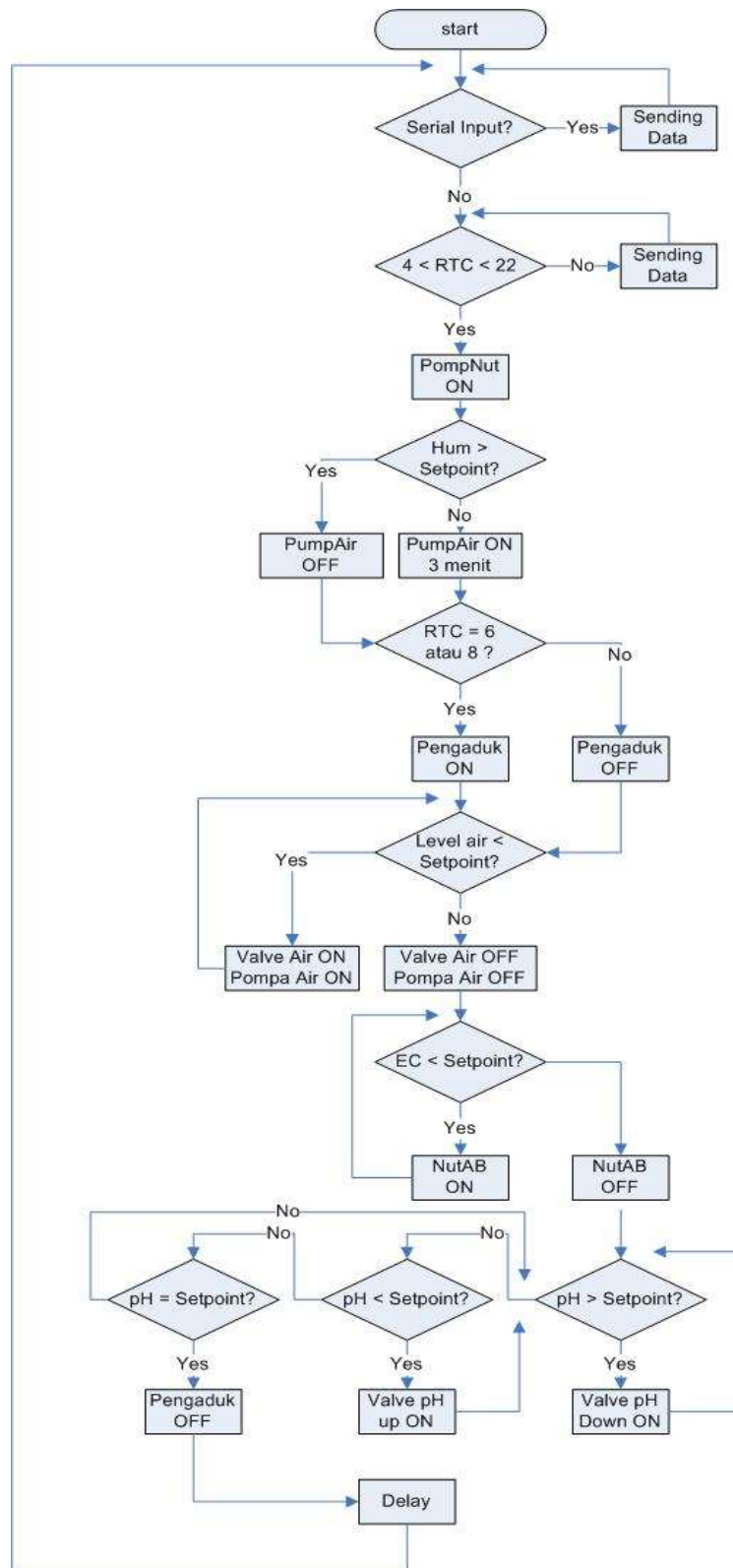
EC1 = s[0:7]
humidity1 = s[7:14]
pH1 = s[14:21]
sql = """INSERT INTO 'dk' (EC, HUM, PH) VALUES ('%s','%s','%s')""" % (EC1, humidity1, pH1)
try:
    cur.execute(sql)
    db.commit()
except:
    db.rollback()

cur.close()
cur = db.cursor()
time.sleep(1)
sql = """UPDATE 'cdk' SET EC=%s, HUM=%s, PH=%s, TIME=CURRENT_TIMESTAMP""" % (EC1, humidity1, pH1)
try:
    cur.execute(sql)
    db.commit()
except:
    db.rollback()
cur.close()
db.close() ...
```

Gambar 3.16: Menyimpan Data Dalam Database

### 3.2.2.5. Perancangan Sistem Pada Slave (Arduino)

Pada Gambar 3.17 dapat dilihat *flowchart* sistem pada arduino secara menyeluruh. Saat sistem mendeteksi adanya *input* serial, maka arduino akan mulai mengirimkan data sensor kepada Raspberry pi melalui SZ05. Selain itu, arduino hanya akan menjalankan sistem kontrol otomatis untuk menjaga setiap variabel pada kebun hidroponik dapat terus terjaga pada rentang *setpoint*. Proses pengiriman kode yang dilakukan ini hanya terjadi dalam hitungan sepersekian detik sehingga tidak mengganggu sistem kontrol kebun, tetapi eksekusi pengiriman data hanya dapat dilakukan paling cepat kurang lebih 3 menit satu kali, karena dapat dilihat pada *flowchart* pada Gambar 3.17, proses 1 *cycle* sebelum kembali ke pengecekan terdapat delay 3 menit dan delay lainnya yang menyebabkan terhambatnya waktu pengambilan data.



Gambar 3.17 Flowchart Sistem Slave

Pada umumnya, pemrograman pada *slave* hanya mendeteksi pengiriman data dari SZ05. *Slave* hanya akan mengirimkan data kepada *master* bila dan hanya bila variabel `stringComplete` bernilai “false” yang menyatakan bahwa pengiriman belum selesai dan juga masih ada penerimaan data serial pada saat yang bersamaan. Saat pengiriman data terjadi, potongan program pada Gambar 3.18 akan mengolah data yang diterima dari *master* akan dijadikan *set point* yang baru dari sistem yang akan dijalankan, apabila *set point* yang baru sama dengan yang lama maka *set point* dari sistem akan tetap sama. Kemudian *slave* akan mengirimkan kembali *current value* dari sensor saat ini. Selain itu sistem dari *slave* hanya akan menjalankan sistem hidroponiknya saja. *Cycle* dari sistem hidroponik yang sudah ada adalah 200 sampai 240 detik, sehingga pengiriman data hanya memungkinkan dilakukan paling cepat 4 menit 1 kali, dikarenakan pada saat siklus hidroponik berjalan, maka *slave* tidak dapat menerima komunikasi serial yang mengakibatkan *slave* baru akan menerima perintah dari master setelah siklus selesai. Pada Gambar 3.18, terdapat kode pemrograman untuk melakukan *zero padding* atau angka 0 diawal pada baris akhir dengan cara menggunakan logika `if`, sehingga apabila nilai dari sensor kurang dari 4 angka didepan koma, maka akan ditambahkan 0 pada depan nilai tersebut hingga terdapat 4 angka didepan koma. *Zero padding* digunakan dengan tujuan agar format data yang akan dikirimkan tetap sama, sehingga memudahkan pengolahan data pada sistem yang ada pada *master*.

```

void loop()
{
  if ((stringComplete==false) && (mySerial.available()))
  {
    while (mySerial.available())
    {
      char inChar = (char)mySerial.read();
      if (inChar == ':')
      {
        sEC=inputString;
        inputString="";
      }
      else if (inChar == ',')
      {
        sHum=inputString;
        inputString="";
      }
      else if (inChar == '-')
      {
        sPH=inputString;
        inputString="";
      }
      else if (inChar == '\n')
      {
        sPH2=inputString;
        stringComplete = true;
      }
      else
      {
        inputString += inChar;
      }
    }

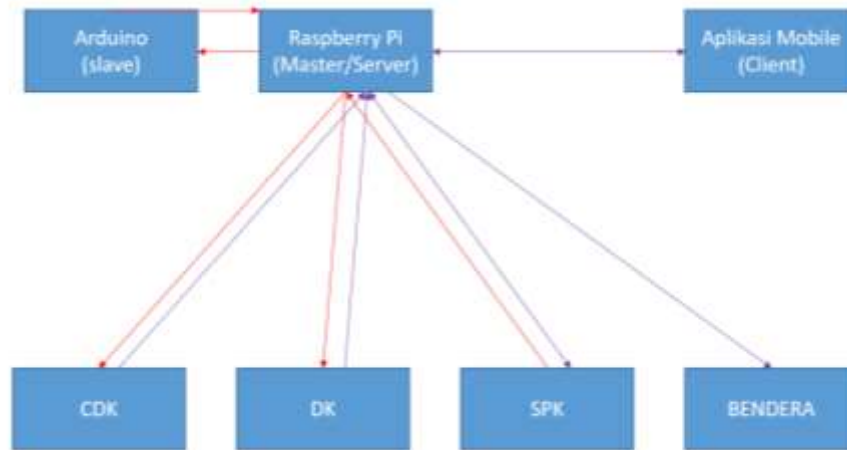
    if (stringComplete == true)
    {
      dtostrf(EC, 4, 2, valEC);
      dtostrf(h, 4, 2, valh);
      dtostrf(pH, 4, 2, valpH);
      if (EC<10)
      {
        sprintf(buffec, sizeof(buffec), "000%a", valEC);
      }
      if ((EC<100)&&(EC>10))
      {
        sprintf(buffec, sizeof(buffec), "00%a", valEC);
      }
      if ((EC<1000)&&(EC>100))
      {
        sprintf(buffec, sizeof(buffec), "0%a", valEC);
      }
      if (EC>1000)
      {
        sprintf(buffec, sizeof(buffec), "%a", valEC);
      }
    }
  }
}

```

Gambar 3.18 Program pada Slave

### 3.2.2.6. Perancangan Database

Setiap kebun memiliki 3 tabel pada database yaitu tabel “spk” untuk menyimpan *setpoint* kebun, tabel “dk” untuk menyimpan histori data kebun dan tabel “cdk” untuk menyimpan nilai sensor terakhir pada kebun. Pada desain database ini, setiap tabel bekerja secara individu sehingga tidak terdapat relasi antar tabel. Alur data dapat dilihat pada garis merah dan ungu pada Gambar 3.19, garis merah adalah alur pada sistem *master-slave* dan ungu untuk sistem *server-client*. Data yang didapatkan oleh *master*, akan diakses dan diinputkan kepada database “cdk” dan “dk” dimana “cdk” adalah *database current* data kebun dan “dk” adalah data kebun.



Gambar 3.19. Blok Diagram *Database Master-Slave* dan *Server-Client*

Dalam sistem *master-slave*, tabel “spk” hanya akan diakses untuk mengirimkan data *setpoint* yang ada pada tabel “spk” ke masing-masing *slave* untuk memerintahkan *slave* mengirimkan data pada sensor. Data yang diterima oleh *master* kemudian disimpan ke tabel “dk” dan “cdk”. Tabel dk digunakan untuk menyimpan histori data kebun sedangkan “cdk” hanya menyimpan data terakhir yang diterima dari kebun.

Kemudian, pada garis berwarna ungu dapat dilihat alur data antara *server* dan *client*. *Mobile application* akan mengakses seluruh tabel yaitu tabel “cdk”, “dk”, “spk” dan bendera. Aplikasi akan mengambil data pada tabel “cdk” dan “spk” untuk ditampilkan pada aplikasi, tabel “dk” diambil 10 data terakhir untuk dijadikan grafik, sedangkan bendera adalah *flag* untuk mengetahui bila *mobile application* melakukan *update* perubahan data pada *set point* kebun di tabel “spk”. Sehingga saat *set point* kebun berubah, Raspberry akan melakukan siklus pengiriman data untuk memberi *setpoint* yang baru kepada setiap kebun

### 3.2.2.7. Perancangan *Mobile Application*

Pembuatan *mobile application* bertujuan agar pengguna dapat melakukan *control* dan *monitoring* kebun secara langsung dengan hanya menggunakan *smartphone*. Pembuatan *mobile application* ini menggunakan cordova sebagai *development tools*. Pada penggunaan

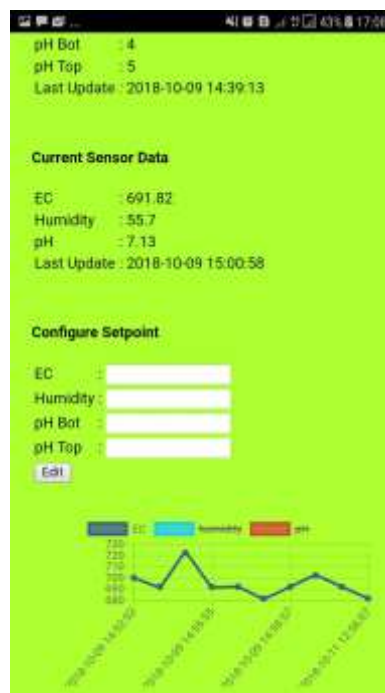


cordova, pembuatan aplikasi android dilakukan dengan mengekstrak *website* yang kemudian akan dibuat menjadi apk untuk diinstall di *smartphone*. Pada pembuatan *mobile application* ini, untuk mengakses *database* digunakan pemrograman PHP dan JavaScript, sedangkan untuk design menggunakan HTML dan CSS. Pada umumnya, aplikasi hanya akan membaca tabel bendera, “cdk” dan “spk” secara terus menerus dengan interval 10 detik. Aplikasi menampilkan data sensor saat ini, *set point* saat ini, grafik data kebun dengan story 10 data terakhir dan konfigurasi *set point* untuk setiap kebun dengan tampilan seperti pada Gambar 3.20 dan 3.21. Pada kolom *setpoint* terdapat informasi *setpoint* EC, humidity, pH atas, pH bawah dan waktu dari update terakhir. Sedangkan pada bagian *current sensor data*, terdapat data terakhir dari EC sensor, humidity, pH dan waktu pengambilan terakhir. Tombol pada bagian atas didesain untuk menghilang saat ditekan, sehingga bila halaman sedang berada pada kebun 1, maka tombol kebun 1 akan menghilang.

Pada bagian *edit*, terdapat *edit box* untuk mengisi *set point* yang diinginkan untuk *set point* EC, humidity, pH batas atas, pH batas bawah dan tombol *edit* untuk *input*. *Input* pada edit box dibatasi dengan beberapa syarat agar dapat dimasukkan dengan tujuan menghindari adanya kesalahan *input* yang dapat berakibat fatal pada sistem kontrol. Batas yang digunakan pada variabel EC adalah 0-5000, humidity 0-100 sedangkan untuk pH atas dan pH bawah 1-14 dengan tambahan syarat pada *input* pH, pH atas tidak bisa lebih kecil dari pH bawah.



Gambar 3.20 Tampilan *Mobile Application (A)*



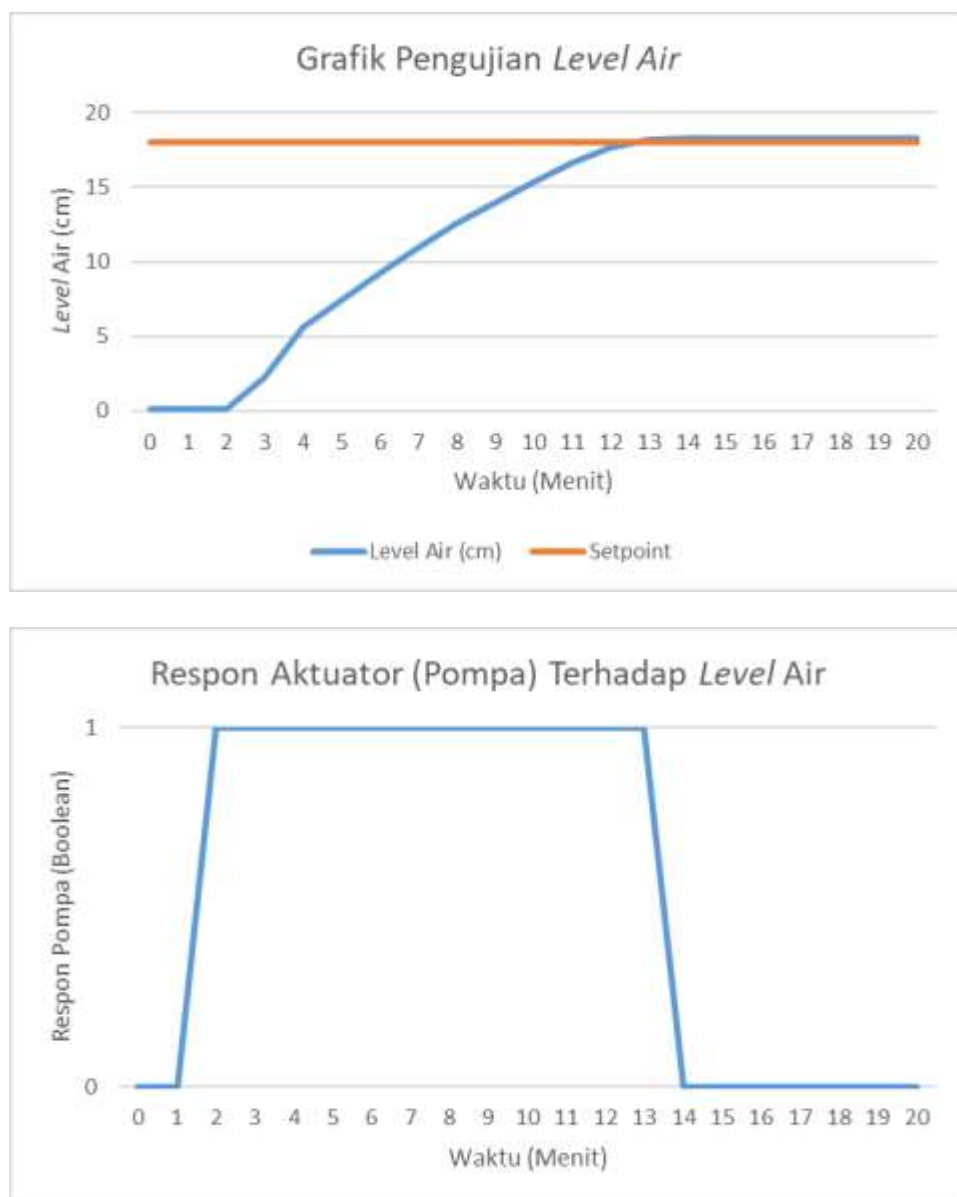
Gambar 3.21 Tampilan *Mobile Application (B)*

## BAB IV HASIL PENGUJIAN SISTEM

### 4.1. Pengujian Sistem Kontrol Dengan PLC dan HMI Panel

#### 4.1.1. Pengujian *Loop Control Level Air*

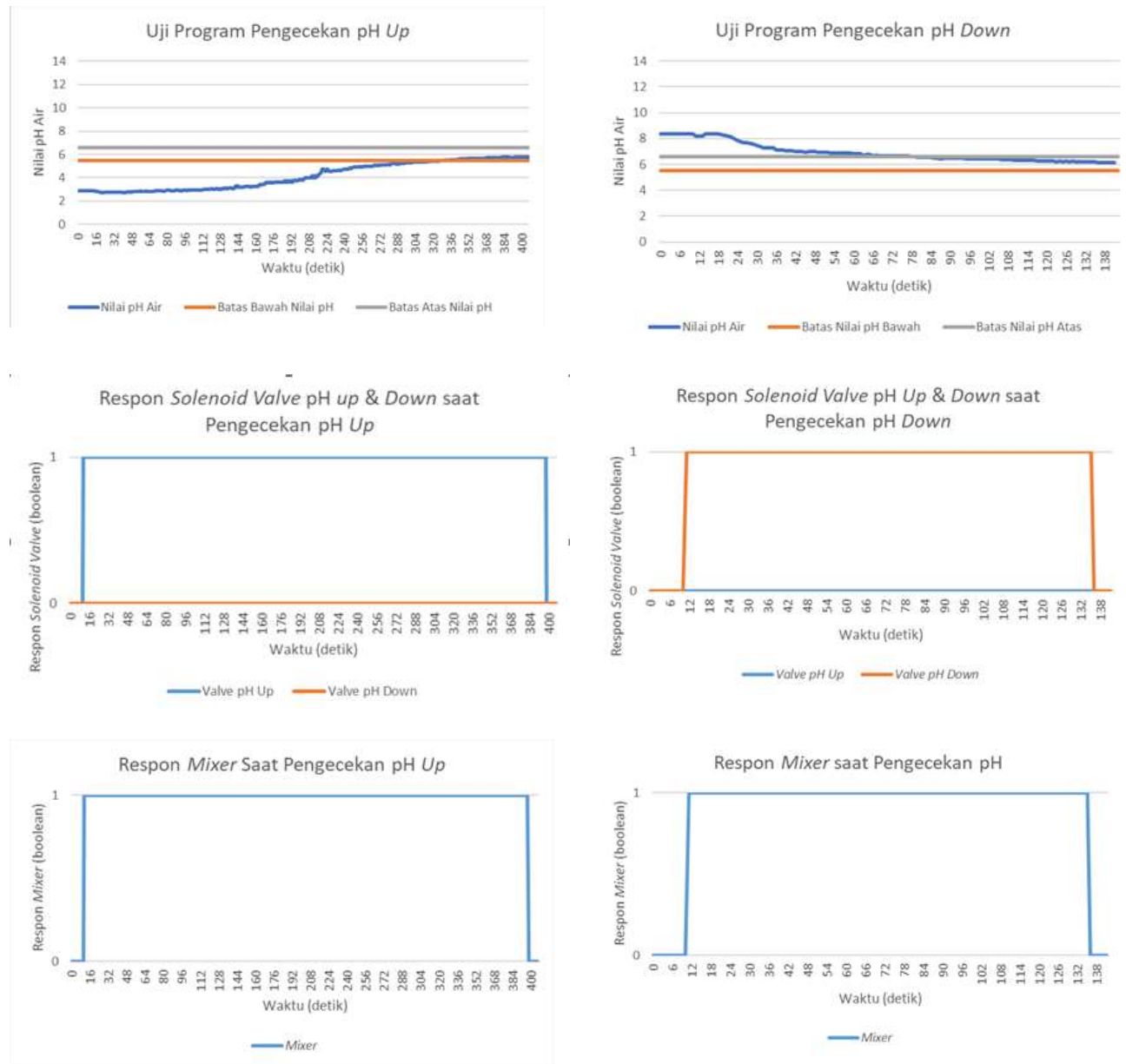
Tujuan pengujian untuk memastikan bahwa sistem kontrol level bekerja dengan baik memastikan level dapat mencapai setpoint pada saat pengecekan kebun. Dalam loop kontrol ini terdapat sensor level, kontroler dan pompa.



Gambar 4.1 Hasil Pengujian loop kontrol *Level Air*

### 4.1.2. Pengujian Loop Control Nilai pH Air

Pengujian ini untuk memastikan bahwa loop kontrol pH bekerja dengan baik. Pengujian dilakukan dengan setpoint yang lebih tinggi dari current value (mekanisme pH Up) dan setpoint lebih rendah dari current value (mekanisme pH Down). Dalam loop kontrol ini terdapat sensor pH, kontroler, valve pH Up dan valve pH Down. Berikut hasil pengujian pH Up dan pH Down



Gambar 4.2. Pengujian loop kontrol pH Up dan pH Down

### 4.1.3. Pengujian Loop Control Nutrisi Dalam Air

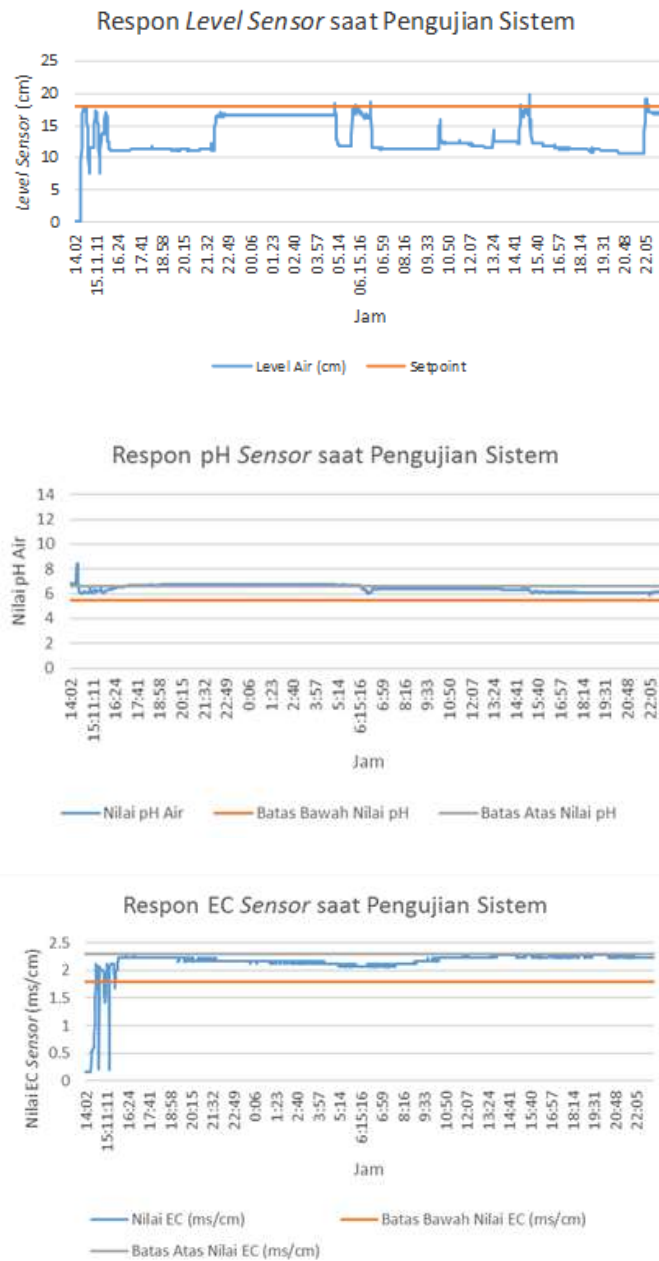
Pada pengujian ini bertujuan untuk mengetahui apakah loop kontrol nutrisi yang dirancang sudah berjalan sesuai yang diharapkan. Dalam loop kontrol ini terdapat sensor EC, kontroler, valve nutrisi A, valve nutrisi B.



Gambar 4.3 Pengujian loop kontrol nutrisi

#### 4.1.4. Pengujian Keseluruhan

Pada pengujian ini, sistem yang dibuat dipasang pada kebun hidroponik. Pengujian dilakukan selama 24 jam lebih, hal ini berfungsi untuk mengetahui apakah sistem dapat berjalan dengan baik atau tidak.

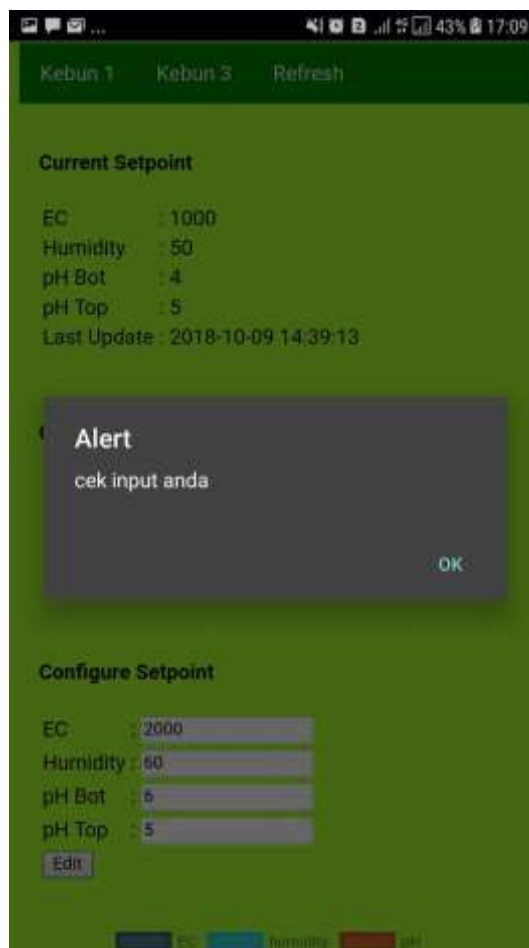


Gambar 4.4. Pengujian Keseluruhan

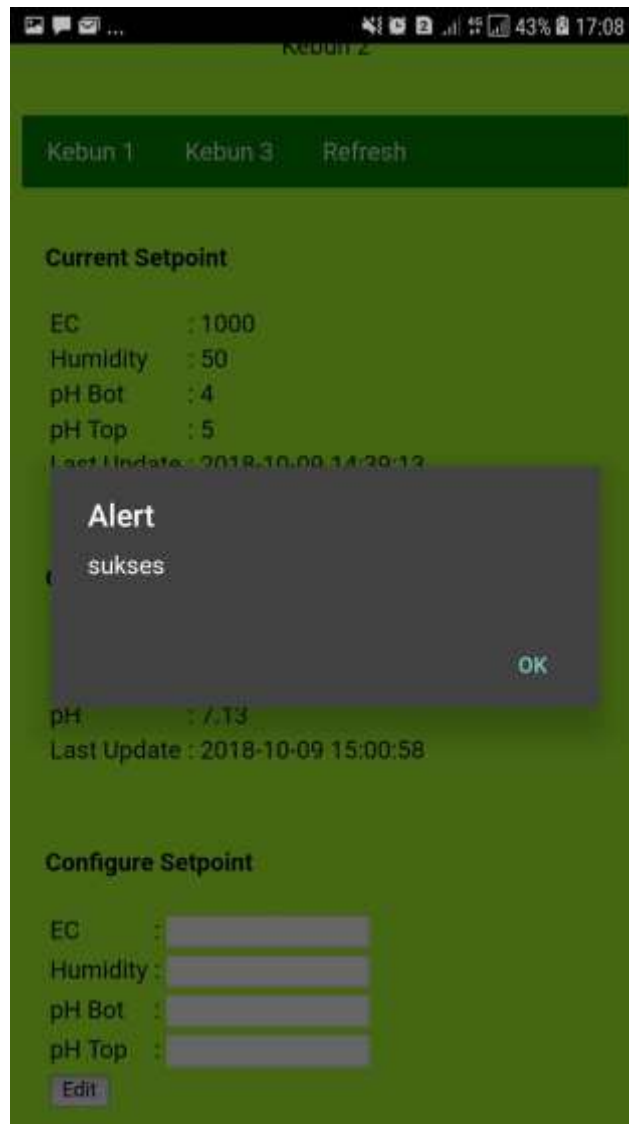
## 4.2. Pengujian Sistem Kontrol Dengan menggunakan Arduino, Raspberry Pi dan Mobile Application

### 4.2.2. Pengujian *Input Set Point*

Untuk menguji fitur ini, dilakukan pengujian dengan melakukan *input* data yang sesuai dan tidak sesuai secara bergantian sebanyak 10 kali pada setiap kondisi, dibawah batas bawah, diatas batas atas dan diantara batas bawah dan atas, kemudian juga dilakukan uji pada syarat pH Bot dan Top yang memiliki syarat pH Bot harus lebih kecil dari pH Top. Pengujian dikatakan berhasil apabila aplikasi dapat membedakan input yang tidak sesuai dan yang sesuai.



Gambar 4.5. Alert Salah Input



Gambar 4.6. Alert input benar

#### 4.2.2. Pengujian Database

Proyek ini memanfaatkan *database* untuk melakukan data akuisisi, sehingga sangat penting reliabilitas dari *database* untuk menampung *input* data dari sistem. Pengujian pada *database* dengan melakukan pengiriman data konstan secara terus menerus dan melakukan data akuisisi setiap 1 menit. Dari Gambar 4.7 dapat dilihat data akuisisi yang dilakukan sangat reliabel untuk setiap data yang diterima, untuk *response time* juga dapat dilihat pada *time stamp* berkisar 60 detik sampai 61detik sehingga penerimaan *database* dapat dikatakan reliabel.



EC	HUM	PH	time
1842.36	0045.64	0007.72	2018-10-04 18:04:11
1842.36	0045.64	0007.72	2018-10-04 18:05:11
1842.36	0045.64	0007.72	2018-10-04 18:06:11
1842.36	0045.64	0007.72	2018-10-04 18:07:11
1842.36	0045.64	0007.72	2018-10-04 18:08:11
1842.36	0045.64	0007.72	2018-10-04 18:09:11
1842.36	0045.64	0007.72	2018-10-04 18:10:12
1842.36	0045.64	0007.72	2018-10-04 18:11:12
1842.36	0045.64	0007.72	2018-10-04 18:12:12
1842.36	0045.64	0007.72	2018-10-04 18:13:12
1842.36	0045.64	0007.72	2018-10-04 18:14:12
1842.36	0045.64	0007.72	2018-10-04 18:15:12
1842.36	0045.64	0007.72	2018-10-04 18:16:12
1842.36	0045.64	0007.72	2018-10-04 18:17:12
1842.36	0045.64	0007.72	2018-10-04 18:18:12
1842.36	0045.64	0007.72	2018-10-04 18:19:12
1842.36	0045.64	0007.72	2018-10-04 18:20:12
1842.36	0045.64	0007.72	2018-10-04 18:21:12
1842.36	0045.64	0007.72	2018-10-04 18:22:12
1842.36	0045.64	0007.72	2018-10-04 18:23:12
1842.36	0045.64	0007.72	2018-10-04 18:24:12
1842.36	0045.64	0007.72	2018-10-04 18:25:12
1842.36	0045.64	0007.72	2018-10-04 18:26:12
1842.36	0045.64	0007.72	2018-10-04 18:27:12
1842.36	0045.64	0007.72	2018-10-04 18:28:12
1842.36	0045.64	0007.72	2018-10-04 18:29:12
1842.36	0045.64	0007.72	2018-10-04 18:30:12
1842.36	0045.64	0007.72	2018-10-04 18:31:12
1842.36	0045.64	0007.72	2018-10-04 18:32:12
1842.36	0045.64	0007.72	2018-10-04 18:33:12

Gambar 4.7. Hasil Pengujian Database

## **BAB V**

### **KESIMPULAN**

Dari hasil perancangan smart hydroponics system berbasis PLC dan HMI dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem kontrol mampu menjaga variabel yang dikontrol: pH, level dan nutrisi sesuai dengan rentang nilai yang diperlukan tanaman.
2. HMI yang dirancang mudah untuk digunakan dan mampu memonitoring kebun dengan baik.
3. Implementasi kontrol dan monitoring kebun dengan menggunakan mobile application dilakukan dengan menggunakan arduino uno sebagai slave, raspberry pi sebagai master/server.

## DAFTAR PUSTAKA

- [1] Badan Pusat Statistik 2013, *Laporan Hasil Sensus Pertanian 2013*, Sensus Pertanian 2013, Badan Pusat Statistik, dilihat 3 September 2017, < <https://st2013.bps.go.id/st2013esya/booklet/at0000.pdf>>
- [2] Dan Wang *et al.*, “Design of A Smart Monitoring and Control System for Aquaponics Based on OpenWrt,” *5<sup>th</sup> International Conference on Information Engineering for Mechanics and Materials (ICIMM 2015)*, Huhhot, Mongolia, 2015.
- [3] Ageng Al Hilal Gajahyana Yosi., “Prototype Hydroponic Greenhouse’s Smart Controller Berbasis ATMEGA328P dengan Bluetooth,” E-Jurnal Prodi Teknik Elektronika Universitas Negeri Yogyakarta, Yogyakarta, Indonesia, 2017.
- [4] Agung Prayitno, Veronica Indrawati, Gabriel Utomo. Trajectory Tracking of AR.Drone Quadrotor Using Fuzzy Logic Controller. *Journal Telkomnika*. Vol.12. No.4. December 2014. pp.819-828
- [5] Agung Prayitno. Perancangan Simulink Model Dari AR.Drone Sebagai Simulator Kontrol Quadrotor. *Proceeding SRITI – STMIK AKAKOM*, 2013
- [6] Veronica Indrawati. Penggunaan Algoritma Learning Vector Quantization Dalam Mengenali Suara Manusia Untuk Kendali Quadrotor. *Proceeding Sentika Universitas Atmajaya*, 2014.
- [7] Agung Prayitno, Veronica Indrawati. Model AR.Drone Dengan Indoor dan Outdoor Hull. *Proceeding CITEE UGM*, 2014.
- [8] Veronica Indrawati, Veronica Indrawati, Thomas Ardi Kusuma. Waypoint Navigation of AR.Drone Quadrotor Using Fuzzy Logic Controller. *Journal Telkomnika*. 2015;13(3): 930-939.
- [9] V.Indrawati, A.Prayitno, G.Utomo. Comparison of Two Fuzzy Logic Controller Schemes for Position Control of AR.Drone. *7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Chiangmai. Thailand. 2015.
- [10] Agung Prayitno, Veronica Indrawati, Ivan Immanuel Trusulaw. Comparison of PID and Fuzzy Controller for Position Control of AR.Drone. *Proceeding EECSI*, 2016.
- [11] Agung Prayitno, Veronica Indrawati, Clark Arron. H-Infinity Control for Pitch-Roll AR.Drone. *Journal Telkomnika*. 2016;14(3): 963-973.