



TUGAS AKHIR-VI180629

RANCANG BANGUN SISTEM *MONITORING PLANT* PENGENDALIAN pH DAN KELEMBABAN TANAH BERBASIS *WEB SERVER* MENGGUNAKAN *RASPBERRY PI 3 B+*

**I Putu Dedi Semara Putra
NRP. 10 51 16 00 00 0066**

**Dosen Pembimbing:
Dr. Ir. Purwadi Agus Darwito, Msc
NIP. 19620822 198803 1 001**

**PROGRAM STUDI D3 TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020**



TUGAS AKHIR-VI180629

RANCANG BANGUN SISTEM *MONITORING PLANT* PENGENDALIAN pH DAN KELEMBABAN TANAH BERBASIS WEB SERVER MENGGUNAKAN *RASPBERRY PI 3 B+*

**I Putu Dedi Semara Putra
NRP. 10 51 16 00 00 0066**

**Dosen Pembimbing:
Dr. Ir. Purwadi Agus Darwito, Msc
NIP. 19620822 198803 1 001**

**PROGRAM STUDI D3 TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020**



FINAL PROJECT-VI180629

***DESIGN AND DEVELOPMENT MONITORING
PH AND SOIL MOISTURE CONTROL SYSTEM
BASED WEB SERVER USING RASPBERRY PI 3
B +***

**I Putu Dedi Semara Putra
NRP. 10 51 16 00 00 0066**

Advisor Lecturer:
Dr. Ir. Purwadi Agus Darwito, Msc
NIP. 19620822 198803 1 001

***STUDY PROGRAM OF DIII INSTRUMENTATION TECHNOLOGY
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2020***

LEMBAR PENGESAHAN I

**RANCANG BANGUN SISTEM *MONITORING PLANT*
PENGENDALIAN pH DAN KELEMBABAN TANAH
BERBASIS *WEB SERVER* MENGGUNAKAN *RASPBERRY*
*PI 3 B+***

TUGAS AKHIR

Oleh :

I Putu Dedi Semara Putra

NRP. 10 51 16 000 00 066

Surabaya, 18 Januari 2020

Mengetahui/Menyetujui

Pembimbing 1



Dr. Ir. Purwadi Agus Darwito, Msc

NIP. 19620822 198803 1 001

Mengetahui,
Kepala Departemen
Teknik Instrumentasi FV-ITS



Dr. Ir. Totok Soehartanto, DEA

NIP. 19650309 199002 1 001

LEMBAR PENGESAHAN II

RANCANG BANGUN SISTEM *MONITORING PLANT* PENGENDALIAN pH DAN KELEMBABAN TANAH BERBASIS *WEB SERVER* MENGGUNAKAN *RASPBERRY* *PI 3 B+*

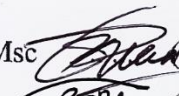
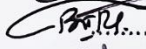


TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
Pada
Program Studi DIII Teknologi Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh:

I Putu Dedi Semara Putra
NRP. 1051160000066

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Ir. Purwadi Agus Darwito, Msc  (Pembimbing I)
2. Brian Raafi'u S.ST., M.T.  (Ketua Penguji)
3. Herry Sufyan Hadi, S.T, M.T.  (Penguji I)
4. Murry Raditya, ST, M.T.  (Penguji II)

SURABAYA
Januari, 2020

**RANCANG BANGUN SISTEM *MONITORING PLANT*
PENGENDALIAN pH DAN KELEMBABAN TANAH
BERBASIS *WEB SERVER* MENGGUNAKAN *RASPBERRY
PI 3 B+***

Nama Mahasiswa : I Putu Dedi Semara Putra
NRP : 10511600000066
Departemen : Teknik Instrumentasi ITS
Dosen Pembimbing : Dr. Ir. Purwadi Agus Darwito, Msc

ABSTRAK

Iklm tropis di Indonesia mengakibatkan terdapat perbedaan kadar pH dan kelembaban tanah di setiap daerah. Dibutuhkan sebuah *plant* yang dapat mengontrol dan memantau kadar pH dan kelembaban tanah. Sistem pemantauan dapat memberikan informasi keberlangsungan *plant* secara *continue*. Dikembangkan sebuah sistem pemantauan berbasis *Web Server* untuk mempermudah proses pemantauan secara *online*. *Plant* pengendalian kadar pH dan kelembaban tanah menggunakan komponen sensor kelembaban tanah, sensor pH, *raspberrypi 3 B+*, *LCD 20x4*, serta *solenoid valve* dan *motor servo* sebagai aktuatornya. Hasil dari pembacaan sensor akan ditampilkan pada *LCD 20x4* pada panel box dan *web server Apache 2*. Hasil pembacaan sensor disimpan pada *database MySQL*. Berdasarkan hasil pengujian penulis, data dapat ditampilkan pada *LCD 20x4* serta dapat diakses melalui *Web Server* pada *smartphone* dan PC. Terdapat *delay* pembacaan pada *LCD 20x4* sebesar 3 ms serta *delay* pada akses *web server* sebesar 0,00024813 ms. Didapatkan maksimal pengguna yang mengakses *web server* lebih dari 10 orang dan maksimal jarak akses adalah 24,75m. *Delay* pada *Web Server* dipengaruhi oleh kecepatan akses internet dan *delay* pada *LCD 20x4* dipengaruhi oleh resistansi pada kabel.

Kata Kunci : Sistem Pemantauan, *Database*, *Raspberry Pi 3 B+*,
Web server

**DESIGN AND DEVELOPMENT MONITORING PH AND
SOIL MOISTURE CONTROL SYSTEM BASED WEB
SERVER USING RASPBERRY PI 3 B +**

Author's Name : I Putu Dedi Semara Putra
NRP : 10511600000066
Departement : Teknik Instrumentasi ITS
Supervisor : Dr. Ir. Purwadi Agus Darwito, Msc

ABSTRACT

The tropical climate in Indonesia results in differences in pH and soil moisture levels in each region. A plant that can control and monitor pH and soil moisture is needed. The monitoring system can provide information on plant continuity continuously. A web server-based monitoring system was developed to simplify the online monitoring process. The plant controls pH and soil moisture using components of soil moisture sensors, pH sensors, raspberrypi 3 B +, LCD 20x4, and solenoid valves and servo motors as actuators. The results of the sensor readings will be displayed on a 20x4 LCD on the panel box and web server. The sensor readings are stored in a MySQL database. Based on the author's test results, the data can be displayed on a 20x4 LCD and can be accessed through a Web Server on smartphones and PCs. There is a reading delay on the 20x4 LCD of 3 ms and a delay on web server access of 0,00024813 ms. Obtained a maximum of users who access the web server more than 10 people and the maximum access distance is 24,75m. The delay on the Web Server is affected by the speed of internet access and the delay on the LCD 20x4 is affected by the resistance of the cable.

Keywords : Monitoring System, Database MySQL, Raspberry Pi 3 B +, web server

KATA PENGANTAR

Puji dan syukur senantiasa dipanjatkan kepada Allah SWT Yang Maha Agung dan Maha Bijaksana. Atas berkah, petunjuk, dan karunia-Nya penulis telah mengerjakan Tugas Akhir dan menyelesaikan laporan dengan judul **“Rancang Bangun Sistem Monitoring Plant Pengendalian pH dan Kelembaban Tanah Berbasis Web Server Menggunakan Raspberry Pi 3 B+”**.

Laporan Tugas Akhir ini disusun sebagai pertanggung jawaban selama pengerjaan Tugas Akhir dan sebagai salah satu syarat memenuhi mata kuliah VI180629 Tugas Akhir program studi D3 Teknologi Instrumentasi, Departemen Teknik Instrumentasi ITS.

Selama pengerjaan dan penyusunan laporan Tugas Akhir ini, penulis telah mendapat banyak dukungan serta bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini, terima kasih diucapkan kepada:

1. Bapak Dr. Ir. Totok Soehartanto, DEA selaku Kepala Departemen Teknik Instrumentasi ITS.
2. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc. selaku dosen pembimbing dan Kepala Laboratorium Instrumentasi Industri yang selalu memberikan motivasi dan ilmu kepada penulis.
3. Dr. Ridho Hantoro, S.T., M.T. selaku dosen wali yang telah memberikan waktunya 7 semester terakhir.
4. Orang tua, terutama papa dan mama, I Nyoman Janji dan Ni Nyoman Sulasih yang tidak henti-hentinya memberi semangat dan doa untuk menyelesaikan Tugas akhir ini.
5. Teman baik penulis, Anak Agung Gede Agung Aditya Satwikayadnya yang telah memberikan dukungan secara lahir batin dan bantuan dengan maksimal serta senantiasa mendoakan keberhasilan penulis.
6. I Made Pande Ari Wijaya teman baik penulis dari bali yang selalu membagi ilmu pemrograman nya.
7. Rahmad Bayu Rifaly yang memberikan do'a, *support*, dan waktunya.
8. Teman-teman TPKH ITS.

9. Teman-teman Komunitas Sing Seni Surabaya.
10. Teman-teman Teknik Instrumentasi Angkatan 2016 yang selalu mendoakan dan *support*.

Semua pihak yang telah membantu dalam pelaksanaan dan penyusunan laporan Tugas Akhir ini. Penulis menyadari bahwa terdapat beberapa kekurangan dalam laporan Tugas Akhir ini. maka dari itu kritik dan saran yang membangun sangat penulis harapkan agar penyusunan laporan selanjutnya lebih baik lagi.

Semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi kita semua.

Surabaya, 18 Januari 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	
TUGAS AKHIR	i
FINAL PROJECT	iii
LEMBAR PENGESAHAN I	v
LEMBAR PENGESAHAN II	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR NOTASI	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
BAB II TINJAUAN PUSTAKA	3
2.1 <i>Monitoring System</i>	3
2.2 <i>WebServer</i>	4
2.3 <i>PHP (Hypertext Preprocessor)</i>	7
2.4 <i>Raspberry Pi 3B+</i>	9
2.5 <i>Program Thonny Python</i>	11
2.6 <i>Database</i>	11
2.7 <i>Komunikasi I2C</i>	12
2.8 <i>Sistem Kontrol</i>	13
2.8.1 <i>Sistem Kontrol Loop Terbuka (Open-Loop Control System)</i>	13
2.8.1 <i>Sistem Kontrol Loop Tertutup (Closed-Loop Control System)</i>	14
2.9 <i>Pengendalian Mode On-Off</i>	15
2.10 <i>Respon Sistem</i>	15
BAB III METODOLOGI PENELITIAN	17
3.1 <i>Perancangan Sistem</i>	17
3.2 <i>Studi Literatur</i>	18

3.3 Identifikasi dan Pemodelan Sistem.....	19
3.4 Perancangan Model Sistem.....	20
3.4.1 Perancangan Alat.....	20
3.4.2 Perancangan Hardware.....	21
3.4.3 Perancangan <i>Software</i>	25
BAB IV HASIL DAN PEMBAHASAN.....	37
4.1 Hasil Pengujian <i>Hardware</i>	37
4.2 Hasil Pengujian <i>Software</i>	40
4.2.1 Pengujian <i>Input</i> Sinyal ADC Ke <i>Raspberry</i>	41
4.2.2 Pengujian pada layar LCD.....	42
4.2.3 Pengujian Pada <i>phpMyAdmin</i>	43
4.2.4 Pengujian Kualitas <i>Web Server</i>	46
4.2.5 Uji <i>Quality of Service (QoS)</i>	51
4.3 Pembahasan.....	54
BAB V PENUTUP.....	59
5.1 Kesimpulan.....	59
5.2 Saran.....	59
DAFTAR PUSTAKA.....	61
LAMPIRAN A SPESIFIKASI KOMPONEN PLANT.....	63
LAMPIRAN B PERCOBAAN THING VIEW.....	71
LAMPIRAN C PROGRAM PLANT.....	75
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 2.1 Raspberry Pi 3 B+(raspberrypi.org, 2019).....	10
Gambar 2.2 Kontrol <i>Loop</i> terbuka.....	13
Gambar 2.3 Kontrol <i>Loop</i> tertutup	14
Gambar 3.1 <i>Flowchart</i> Sistem	17
Gambar 3.2 Diagram blok <i>monitoring</i>	19
Gambar 3.3 Design plan pengondisian kelembaban dan pH	20
Gambar 3.4 Design P&ID.....	21
Gambar 3.5 <i>Soil Moisture Sensor</i> (Dfrobot, 2019).....	22
Gambar 3.6 <i>PH Meter Kit SEN0161 Sensor</i> (Dfrobot, 2019)....	22
Gambar 3.7 Komponen pada panel.....	23
Gambar 3.8 Wiring Proses Monitoring.....	24
Gambar 3.9 <i>Data Flow Diagram Monitoring</i>	25
Gambar 3.10 <i>Login Putty</i>	27
Gambar 3.11 <i>Login VNC view</i>	27
Gambar 3.12 <i>Start up VNC view</i>	28
Gambar 3.13 <i>Start up putty</i>	28
Gambar 3.14 Tampilan <i>Upgrade</i>	29
Gambar 3.15 Tampilan <i>Update</i>	29
Gambar 3.16 Perintah nano	30
Gambar 3.17 Install mariadb	30
Gambar 3.18 <i>Install apache2</i>	31
Gambar 3.19 <i>Restart apache2</i>	31
Gambar 3.20 <i>Install phpMyAdmin</i>	32
Gambar 3.21 <i>login phpMyAdmin</i>	32
Gambar 3.22 Pembuatan tabel	33
Gambar 3.23 Tampilan LCD 20x4	34
Gambar 4.1 kalibrasi sensor Kelembaban tanah.....	37
Gambar 4.2 Kalibrasi Sensor pH tanah.....	38
Gambar 4.3 (a) Pembacaan di air (b) Pembacaan di udara.....	39
Gambar 4.4 Tampilan awal VNC viewer	40
Gambar 4.5 Tampilan saat plan dijalankan.....	41
Gambar 4.6 Tampilan <i>error</i> pada LCD	42
Gambar 4.7 Tampilan data monitoring kelembaban dan pH tanah pada LCD.....	43
Gambar 4.8 Tampilan pengujian MariaDB	44

Gambar 4.9	Tampilan <i>table</i> tugas_akhir.....	44
Gambar 4.10	Tampilan <i>monitoring</i> tugas_akhir pada <i>smartphone</i>	45
Gambar 4.11	Tampilan <i>monitoring</i> online tugas_akhir pada beberapa <i>device</i>	46
Gambar 4.12	Tampilan data pada <i>raspberry</i>	47
Gambar 4.13	Tampilan <i>table</i> kelembaban data pada <i>database</i> ..	47
Gambar 4.14	Tampilan <i>table</i> pH data pada <i>database</i>	48
Gambar 4.15	Tampilan <i>monitoring</i> pada <i>Web Server</i>	48
Gambar 4.16	pengujian sejauh SEKRETARIAT HIMADATA- ITS	49
Gambar 4.17	Pengujian sejauh Ruang Sidang Teknik Instrumentasi ITS	50
Gambar 4.18	Tampilan Wireshark.....	51
Gambar 4.19	Skema Penambahan Input Analog	55
Gambar 4.20	Perbandingan Beaglebone Black dan Raspberry 3 B+ (Educba, 2019)	56

DAFTAR TABEL

Tabel 4.1 Tabel pembacaan <i>input</i> sensor palsu dan <i>output</i> ADC	39
Tabel 4.2 Tabel pembacaan <i>input</i> sensor Kelembaban dan <i>output</i> ADC	39
Tabel 4.3 Tabel pembacaan <i>input</i> sensor pH dan <i>output</i> ADC ..	40
Tabel 4.4 Tabel kategori <i>delay</i> (Tedyyana, 2011)	52
Tabel 4.5 Tabel kategori <i>jitter</i> (Tedyyana, 2011)	53
Tabel 4.5 Tabel kategori <i>Packet loss</i> (Tedyyana, 2011)	54
Tabel 4.6 Tabel spesifikasi <i>Beaglebone Black</i> dan <i>Raspberry 3</i> (Educba, 2019)	56

DAFTAR NOTASI

$\bar{X}D$	= Rata-Rata <i>Delay</i>
$\sum D$	= Jumlah <i>Delay</i>
$\sum E$	= Jumlah Variasi <i>Delay</i>
$\sum A$	= Jumlah Data <i>In</i>
$\sum B$	= Jumlah Data <i>Out</i>
PL	= <i>Packet Loss</i>

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah negara dengan iklim tropis. Iklim tropis memiliki beberapa kelebihan diantaranya sumber daya alam yang melimpah, curah hujan yang cenderung tinggi dan tanah yang subur. Iklim tropis ini sangat menguntungkan untuk masyarakat Indonesia bercocok tanam (Merisa, 2017). Berdasarkan data dari BPS (badan pusat statistik) jumlah penduduk Indonesia yang bermata pencaharian sebagai petani sejumlah 35,7 juta orang atau 28,76% dari jumlah penduduk Indonesia. Sektor pertanian merupakan sektor utama yang berperan penting pada perekonomian nasional dalam menyerap tenaga kerja, sumber pertumbuhan ekonomi dan penyumbang devisa.

Tak hanya petani, karyawan dan seorang pengusaha pun melakukan kegiatan bercocok tanam sebagai hobi atau sebagai sumber bahan makanan organik. Kebanyakan masyarakat Indonesia yang bekerja sebagai karyawan maupun pengusaha memiliki sedikit waktu untuk bercocok tanam namun memiliki keinginan yang besar untuk melakukannya. Karyawan dan pengusaha biasanya tinggal di daerah perkotaan dimana lahan sangat sempit dan kualitas tanahnya juga kurang baik (Luthfy, 2015). Dalam memecahkan masalah tersebut harus diciptakan sebuah alat yang dapat memonitoring dan mengontrol kualitas tanah agar terjaga secara otomatis.

Kualitas tanah dapat dikatakan baik apabila kadar air dan pH nya terjaga di *set point* tertentu. Contohnya tanaman cabai, kadar air yang terdapat di dalam tanah harus di antara 40% sampai 60%. Serta kadar pH di dalam tanah sekitar 6 sampai 7 (Javamas, 2019). Untuk mengatur kualitas tanah diperlukan suatu rancang bangun alat yang dapat mengontrol kadar air dan pH secara otomatis dan dapat dimonitoring secara online melalui *smartphone* atau komputer.

Pada tugas akhir ini dibuatlah sebuah penelitian yang diharapkan dapat menjawab masalah dalam bidang bercocok tanam di daerah perkotaan dan daerah dengan tanah yang kurang

subur. Dengan menggunakan *Raspberry Pi 3 B+* sebagai *controler* serta diperuntukan sebagai *server* dari plant. Data pembacaan sensor akan di unggah menuju *database MySQL* dan di akses melalui *Web Server Apache2* yang terpasang pada *Raspberry*. Bahasa pemrograman yang digunakan untuk menjalankan sistem adalah *Python* dan *PHP*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, didapatkan beberapa rumusan masalah sebagai berikut:

- a. Bagaimana *monitoring* pH dan Kelembaban tanah pada proses pengondisian tanaman dengan kondisi *real time*?
- b. Bagaimana menampilkan hasil *monitoring* pada *LCD*?
- c. Bagaimana menampilkan hasil *monitoring* pada *Web Server* dan menyimpannya pada *database MySQL Raspberry*?

1.3 Tujuan

Tujuan utama dari penelitian ini adalah untuk memenuhi mata kuliah Tugas Akhir sebagai syarat kelulusan dari program studi Diploma 3 Teknik Instrumentasi industri serta hasil penelitian tersebut diharapkan dapat memberikan solusi terhadap rumusan masalah diatas sebagai berikut:

- a. Untuk memonitoring pH dan Kelembaban tanah pada proses pengondisian tanaman dengan kondisi *real time*.
- b. Untuk menampilkan hasil *monitoring* pada *LCD*.
- c. Untuk menampilkan hasil *monitoring* pada *Web Server* dan menyimpannya pada *database MySQL Raspberry*?

1.4 Batasan Masalah

Berdasarkan perumusan masalah di atas didapatkan beberapa Batasan masalah sebagai berikut:

- a. Perancangan alat pengatur pH dan Kelembaban tanah.
- b. Proses integrasi antara *output* sensor dan ADC.
- c. Perancangan program python pengatur pH dan Kelembaban tanah di *Raspberry Pi 3B+*.
- d. Perancangan *interface* pada *web server* dan *LCD 20x4*.

BAB II TINJAUAN PUSTAKA

2.1 *Monitoring System*

Penilaian (Evaluasi) merupakan tahapan yang berkaitan erat dengan kegiatan monitoring, karena kegiatan evaluasi dapat menggunakan data yang disediakan melalui kegiatan monitoring. Dalam merencanakan suatu kegiatan hendaknya evaluasi merupakan bagian yang tidak terpisahkan, sehingga dapat dikatakan sebagai kegiatan yang lengkap. Evaluasi diarahkan untuk mengendalikan dan mengontrol ketercapaian tujuan. Evaluasi berhubungan dengan hasil informasi tentang nilai serta memberikan gambaran tentang manfaat suatu kebijakan. Istilah evaluasi ini berdekatan dengan penafsiran, pemberian angka dan penilaian. Evaluasi dapat menjawab pertanyaan “Apa perbedaan yang dibuat” (SUBA, R.S.K., 2014).

Monitoring didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan, dan tindakan atas informasi suatu proses yang sedang diimplementasikan. Umumnya, *monitoring* digunakan dalam checking antara kinerja dan target yang telah ditentukan. *Monitoring* ditinjau dari hubungan terhadap manajemen kinerja adalah proses terintegrasi untuk memastikan bahwa proses berjalan sesuai rencana (on the track). *Monitoring* dapat memberikan informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah perbaikan yang berkesinambungan. Pada pelaksanaannya, *monitoring* dilakukan ketika suatu proses sedang berlangsung. Level kajian sistem *monitoring* mengacu pada kegiatan per kegiatan dalam suatu bagian misalnya kegiatan pemesanan barang pada supplier oleh bagian purchasing. Indikator yang menjadi acuan *monitoring* adalah *output* per proses / per kegiatan. Umumnya, pelaku *monitoring* merupakan pihak-pihak yang berkepentingan dalam proses, baik pelaku proses (self *monitoring*) maupun atasan / supervisor pekerja. Berbagai macam alat bantu yang digunakan dalam pelaksanaan sistem *monitoring*, baik observasi / interview secara langsung, dokumentasi maupun aplikasi visual.

Pada dasarnya, *monitoring* memiliki dua fungsi dasar yang berhubungan, yaitu *compliance monitoring* dan *performance monitoring*. *Compliance monitoring* berfungsi untuk memastikan proses sesuai dengan harapan / rencana. Sedangkan, *performance monitoring* berfungsi untuk mengetahui perkembangan organisasi dalam pencapaian target yang diharapkan. Umumnya, *output monitoring* berupa progress report proses. *Output* tersebut diukur secara deskriptif maupun non-deskriptif. *Output monitoring* bertujuan untuk mengetahui kesesuaian proses telah berjalan. *Output monitoring* berguna pada perbaikan mekanisme proses / kegiatan di mana *monitoring* dilakukan (Sanjana, P., and Mahalakshmi, 2014)

2.2 WebServer

Server dalam dunia komputer adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat scalable dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan atau *network operating system*. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (*printer*), dan memberikan akses kepada *workstation* anggota jaringan. Umumnya, di atas sistem operasi server terdapat aplikasi-aplikasi yang menggunakan arsitektur klien/ *server*. Contoh dari aplikasi ini adalah *DHCP Server*, *Mail Server*, *HTTP Server*, *FTP Server*, *DNS Server* dan lain sebagainya.

Pada saat mengakses sebuah halaman website, biasanya anda mengetik halaman tersebut di *browser* seperti *mozilla*, *chrome* dan lain-lain. Setelah anda meminta (biasanya dengan menekan enter) untuk dapat mengakses halaman tersebut, browser akan melakukan permintaan ke *web server*. Disinilah *web server* berperan, *web server* akan mencarikan data yang diminta *browser*, lalu mengirimkan data tersebut ke *browser* atau menolaknya jika ternyata data yang diminta tidak ditemukan.

Menghubungkan antara *Web Server* dan *Web Browser Internet* dan ini merupakan gabungan dari jaringan Komputer yang ada di seluruh dunia. Setelah semuanya terhubung secara fisik, *Protocol*

TCP, IP atau *networking protocol* yang memungkinkan semua komputer di dunia dapat berkomunikasi satu sama lainnya. Ketika browser meminta data web page kepada server maka instruksi dari permintaan data browser tersebut akan di kemas di dalam *TCP* yang merupakan satu protocol transport kemudian dikirim ke alamat, dalam hal ini adalah protocol berikutnya yaitu *HTTP* atau Hyper Text Transfer Protocol yang sudah kita kenal. *HTTP* ini merupakan sebuah protocol yang akan digunakan dalam *WWW* (*World Wide Web*) antar komputer yang saling terhubung dalam jaringan internet di dunia ini.

Web adalah tampilan pada *browser* dengan alamat domain khusus untuk sistem penelitian ini. *Web* dapat dibangun dengan menggunakan bahasa *HTML* dan *PHP* dengan style tampilan menggunakan bahasa *CSS*. *Web* tersebut disimpan pada satu komputer yang disebut *server*. *Server* menyimpan program *web* dan *database* untuk dapat diakses oleh admin atau *client* dari *browser*. *Website* dapat dibangun menggunakan program notepad/notepad++ atau adobe dreasweaver. *Webserver* adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari *HTTP* atau *HTTPS* pada *client* yang dikenal dan biasanya dikenal dengan nama *webbrowser* dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen *HTML*. Dalam bentuk sederhana *webserver* akan mengirim data *HTML* kepada permintaan *webbrowser* sehingga akan terlihat seperti pada umumnya yaitu sebuah tampilan *website*. Fungsi utama *webserver* adalah untuk melakukan atau akan tranfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. Halaman *web* yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan *webserver* berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman *web* termasuk yang di dalam berupa teks, video, gambar atau banyak lagi (Damayanti, 2017). Beberapa Jenis *webserver* di antanya adalah:

1. *Apache WebServer / The HTTP WebServer*
2. *Apache Tomcat*
3. *Microsoft windows Server 2008 IIS (Internet Information Services)*

4. *Lighttpd*
5. *Zeus WebServer*
6. *Sun Java System WebServer*

Dari daftar list di atas yang paling dominan dan digunakan hanya pada *Apache* dan *Microsoft Windows Server*. *Apache* merupakan *webserver* yang paling banyak dipergunakan di internet. Program ini pertama kali didesain untuk sistem operasi lingkungan *UNIX*. Namun demikian, pada beberapa versi berikutnya *Apache* mengeluarkan programnya yang dapat dijalankan di *Windows NT*. *Apache* mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunaanya. Dukungan *Apache* Beberapa dukungan *Apache* :

1. **Kontrol Akses.** Kontrol ini dapat dijalankan berdasarkan nama host atau nomor IP
2. ***CGI (Common Gateway Interface)*** Yang paling terkenal untuk digunakan adalah *perl (Practical Extraction and Report Language)*, didukung oleh *Apache* dengan menempatkannya sebagai modul (*mod_perl*)
3. ***PHP (Personal Home Page/PHP Hypertext Processor)*** Program dengan metode semacam *CGI*, yang memproses teks dan bekerja di *server*. *Apache* mendukung *PHP* dengan menempatkannya sebagai salah satu modulnya (*mod_php*). Hal ini membuat kinerja *PHP* menjadi lebih baik
4. ***SSI (Server Side Includes)***

Keunggulan *WebServer Apache* mempunyai kelebihan dari beberapa pertimbangan di atas :

1. *Apache* termasuk dalam kategori *freeware*.
2. *Apache* mudah sekali proses instalasinya jika dibanding *webserver* lainnya seperti *NCSA*, *IIS*, dan lain-lain.
3. Mampu beroperasi pada berbagai *platform* sistem operasi.
4. Mudah mengatur konfigurasinya, *Apache* mempunyai hanya empat file konfigurasi.
5. Mudah dalam menambahkan peripheral lainnya ke dalam platform *webserver*.
6. Dapat dijalankan menggunakan bahasa pemrograman *PHP*.
7. Mudah untuk melakukan *clear* dan *check* data.

Fasilitas atau Ciri Khas dari *Apache*. Fasilitas atau ciri khas dari *webserver Apache* adalah :

1. Dapat dijadikan pengganti bagi *NCSA webserver*.
2. Perbaikan terhadap kerusakan dan *error* pada *NCSA 1.3* dan *1.4*.
3. *Apache* merespon *webclient* sangat cepat jauh melebihi *NCSA*.
4. Mampu di kompilasi sesuai dengan spesifikasi *HTTP* yang sekarang.
5. *Apache* menyediakan *feature* untuk *multihomed* dan *virtual server*.
6. Kita dapat menetapkan respon *error* yang akan dikirim *webserver* dengan menggunakan file atau skrip.
7. *Server apache* dapat otomatis berkomunikasi dengan *client browser*nya untuk menampilkan tampilan terbaik pada *client browser*nya. *Webserver apache* secara otomatis menjalankan file *index.html*, halaman utamanya, untuk ditampilkan secara otomatis pada *client*nya.
8. *Webserver Apache* mempunyai level-level pengamanan.
9. *Apache* mempunyai komponen dasar terbanyak di antara *webserver* lain.
10. Ditinjau dari segi sejarah perkembangan dan prospeknya, *Apache webserver* mempunyai prospek yang cerah. *Apache* berasal dari *webserver NCSA* yang kemudian dikembangkan karena *NCSA* masih mempunyai kekurangan di bidang kompatibilitasnya dengan sistim operasi lain. Sampai saat ini, *webserver Apache* terus dikembangkan oleh tim dari *apache.org*.
11. Performasi dan konsumsi sumber daya dari *webserver apache* tidak terlalu banyak, hanya sekitar 20 MB untuk file-file dasarnya dan setiap daemونها hanya memerlukan sekitar 950 KB memory per child.
12. Mendukung transaksi yang aman (*secure transaction*) menggunakan *SSL (secure socket layer)*.
13. Mempunyai dukungan teknis melalui *web*.
14. Mempunyai kompatibilitas *platform* yang tinggi.
15. Mendukung *third party* berupa modul-modul tambahan.

Cara Kerja *WebServer* adalah ketika informasi *web* disimpan dalam dokumen yang disebut dengan halaman-halaman *web* (*webpages*). *Webpages* adalah file-file yang disimpan dalam komputer yang disebut dengan *server-server web* (*webservers*). Komputer-komputer membaca *webpage* disebut sebagai *webclient*. *Webclient* menampilkan page dengan menggunakan program yang disebut dengan *browser web* (*webbrowser*). (Rifqi, 2006)

Browser web adalah *software* yang digunakan untuk menampilkan informasi dari *server web*. *Software* ini kini telah dikembangkan dengan menggunakan *user interface grafis*, sehingga pemakai dapat melakukan '*point* dan *click*' untuk pindah antar dokumen. Suatu *browser* mengambil sebuah *webpage* dari *server* dengan sebuah *request*. Sebuah *request* adalah sebuah *request HTTP standart* yang berisi sebuah *page address*. Sebuah *page address* terlihat sebagai berikut : <http://www.yahoo.com>. Seluruh *webpage* berisi intruksi-intruksi bagaimana untuk ditampilkan. *browser* menampilkan page dengan membaca intruksi-intruksi ini. Intruksi yang paling umum untuk menampilkan disebut dengan tag HTML. Tag HTML contohnya , yaitu `<p>` ini adalah paragraph `</p>` (Damayanti, 2017)

2.3 PHP (*Hypertext Preprocessor*)

PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client). Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman *web* sederhana, tetapi juga *website* populer yang digunakan oleh jutaan orang seperti *wikipedia*, *wordpress*, *joomla*, dll.

Saat ini PHP adalah singkatan dari *Hypertext Preprocessor*, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs *web* dan bisa digunakan bersama dengan HTML. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP

adalah singkatan dari *Personal Home Page Tools*. Selanjutnya, diganti menjadi *FI (Form Interpreter)*. Sejak versi 3.0, nama bahasa ini diubah menjadi *PHP (Hypertext Preprocessor)*. Script *PHP* adalah bahasa program yang berjalan pada sebuah *webServer*, atau sering disebut *Server-side*. *PHP* adalah bahasa *Server-side* scripting yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis. *PHP* dapat digunakan dengan gratis dan bersifat *Open Source*. *PHP* dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi *GNU General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*. Maksud dari *Server-side* scripting adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di *Server* tetapi disertakan pada dokumen *HTML*. Pembuatan *web* ini merupakan kombinasi antara *PHP* sendiri sebagai bahasa pemrograman dan sebagai pembangun halaman *web*. (Fridayanthie & Mahdiati, 2016)

2.4 Raspberry Pi 3B+

Raspberry Pi merupakan komputer dalam satu singleboard. Operating System (OS) pada *Raspberry Pi* yaitu *Linux*. *Raspberry Pi* memiliki beberapa seri seperti *Raspberry Pi 1*, *2*, *3*, model *A*, model *A+*, model *B*, model *B+*. Seri yang akan digunakan dalam penelitian kali ini adalah *Raspberry Pi 3* model *B* yang merupakan seri terbaru. Berikut ini adalah spesifikasi dari *Raspberry Pi 3* model *B* (Christian, 2017)

- a. *Prosesor*
System on a Chip (SoC) berupa jenis chip jenis *Broadcom BCM2837R*. CPU *4x ARM Cortex-A53*, kecepatan prosesor *1.2 GHz*, GPU berupa *Broadcom VideoCore IV*, dengan RAM *1 GB LPDDR2 (900 MHz)*.
- b. *Slot Secure Digital Card (SD Card)*
Raspberry Pi 3 model B dilengkapi dengan slot *SD card* sebagai *hard drive* untuk menyimpan seluruh data.
- c. *Port USB*
Raspberry Pi 3 model B mempunyai 4 port *USB* tipe *2.0*.
- d. *Bluetooth*
Raspberry Pi 3 model B mempunyai jenis *Bluetooth 4.1 Classic* yang berfungsi sebagai media penghubung komunikasi dengan perangkat komunikasi lainnya.

- e. Konektor *HDMI*
Raspberry Pi 3 model B mempunyai *port HDMI* sebagai perantara *audio/video* yang akan ditampilkan pada sebuah monitor.
- f. *Output Audio Analog*
Port audio analog berfungsi sebagai penyedia keluaran *audio analog* untuk disambungkan pada perangkat *speaker* dengan *jack* sebesar 3,5 mm.



Alternate Function						Alternate Function
	3.3V PWR	1		2	5V PWR	
I2C1 SDA	GPIO 2	3		4	5V PWR	
I2C1 SCL	GPIO 3	5		6	GND	
	GPIO 4	7		8	UART TX	
	GND	9		10	UART RX	
	GPIO 17	11		12	GPIO 18	
	GPIO 27	13		14	GND	
	GPIO 22	15		16	GPIO 23	
	3.3V PWR	17		18	GPIO 24	
SPI0 MOSI	GPIO 10	19		20	GND	
SPI0 MISO	GPIO 9	21		22	GPIO 25	
SPI0 SCLK	GPIO 11	23		24	GPIO 8	SPI0 CS0
	GND	25		26	GPIO 7	SPI0 CS1
	Reserved	27		28	Reserved	
	GPIO 5	29		30	GND	
	GPIO 6	31		32	GPIO 12	
	GPIO 13	33		34	GND	
SPI1 MISO	GPIO 19	35		36	GPIO 16	SPI1 CS0
	GPIO 26	37		38	GPIO 20	SPI1 MOSI
	GND	39		40	GPIO 21	SPI1 SCLK

Gambar 2.1 Raspberry Pi 3 B+(Christian, 2017)

- g. *Raspberry Pi 3 model B* terdapat 40 buah pin. Pin pada *Raspberry Pi 3* terdiri dari beberapa bagian yaitu bagian

VCC, *GND* dan *GPIO* (*General Purpose Input/Output*). Terdapat 3 pin *VCC* dan 8 pin *GND*. Pin *GPIO* mulai dari *GPIO2* hingga *GPIO27*, pada pin *GPIO* terdapat fungsi lainnya.

2.5 Program *Thonny Python*

Thonny adalah ide baru (lingkungan pengembangan terintegrasi) yang dibundel dengan versi terbaru dari Raspbian dengan sistem operasi *PIXEL*. Penggunaan *Thonny* relatif lebih mudah untuk belajar kode. *Thonny* hadir dengan Python 3.6 bawaan, jadi tidak perlu memasang apa pun.

Program dapat ditemukan dengan cara membuka program pada *Raspberry*, dengan meng-klik Menu> Programming. Saat memulai *Thonny*, maka akan terdapat editor skrip dan shell baru. Seperti halnya dengan Python 2/3 IDLE, program dapat dimasukkan di editor skrip dan dijalankan pada shell. Kemudian shell dapat digunakan untuk berinteraksi langsung dengan program; mengakses variabel, objek, dan fitur program lainnya.

Thonny memiliki serangkaian fitur tambahan untuk mempelajari pemrograman. Salah satu fitur terbaik adalah mode debug yang kuat, tetapi mudah digunakan. Alih-alih menjalankan program, *Thonny* dapat melangkah melalui kode per baris. Terdapat variabel dan objek yang dapat dibuat, dan nilai-nilai diteruskan ke fungsi atau dinilai oleh pembanding. Terdapat beberapa hal yang cukup bagus di *Thonny*. Kemampuan untuk melangkah melalui program, membuatnya lebih mudah untuk memahami apa yang terjadi ketika dijalankan.

2.6 *Database*

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang kemudian disimpan.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*database management*

system / DBMS). *DBMS* merupakan sistem perangkat lunak yang memungkinkan pengguna basis data (*database user*) untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. Ada beberapa fungsi yang harus ditangani *DBMS* seperti pendefinisian data, menangani permintaan pengguna untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh *DBA (Database Administrator)*, menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun media penyimpanan (*disk*) dan juga menangani unjuk kerja semua fungsi secara efisien. Tujuan utama *DBMS* adalah untuk memberikan tinjauan abstrak data kepada pengguna. Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara dan juga bisa diakses secara efisien. Pertimbangan efisien di sini adalah rancangan struktur data yang kompleks tetapi masih bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya (Tedyyana, 2016).

2.7 Komunikasi I2C

Komunikasi I2C (*Inter-Integrated Circuit*) merupakan koneksi dibuat untuk menyediakan komunikasi antara perangkat-perangkat terintegrasi, seperti sensor, RTC, dan juga EEPROM. Komunikasi I2C bersifat *synchronous* namun berbedadengan SPI karena I2C menggunakan *protocol* dan hanya menggunakan dua kabel untuk komunikasi, yaitu *Synchronous clock (SCL)* dan *Synchronous data (SDA)*. Secara berurutan data dikirim dari *master* ke *slave* kemudian (setelah komunikasi *master* ke *slave* selesai) dari *slave* ke *master*. Perangkat I2C menggunakan 2 buah pin open - drain dua arah dengan memberikan pull-up resistor untuk setiap garis bus sehingga berlaku seperti AND menggunakan kabel. AVR dapat menggunakan 120 jenis perangkat untuk berbagi pada bus I2C yang masing-masing disebut sebagai *node*. Setiap *node* beroperasi sebagai *master* atau *slave*. *Master* merupakan perangkat yang menghasilkan *clock* untuk sistem, menginisiasi, dan juga memutuskan sebuah *transmisi*. *Slave* merupakan *node* yang menerima *clock* dan dialamatkan oleh *master*. Baik *master* dan *slave* dapat menerima dan mentransmisikan data. Data yang ditransmisikan diproses oleh *master* untuk diolah menjadi data yang sudah dapat diakses oleh *slave*.

I2C merupakan protocol komunikasi serial dimana setiap bit data ditransfer pada jalur SDA yang disinkronisasikan dengan pulsa *clock* pada jalur *SCL*. Jalur data tidak dapat berubah ketika jalur *clock* berada dalam kondisi *high*. Dalam I2C, setiap alamat atau data yang ditransmisikan harus dibentuk dalam sebuah paket dengan panjang 9 bit dimana 8 bit pertama disimpan dalam jalur SDA oleh *transmitter*, dan bit ke-9 merupakan *acknowledge* (atau *not acknowledge*) oleh receiver. I2C juga diistilahkan sebagai *Two wire Serial Interface (TWI)*, bergantung dari istilah yang digunakan oleh pabrik yang membuat perangkat. Salah satu perangkat yang digunakan dengan komunikasi I2C adalah *realtime clock (RTC)*. Perangkat ini menyediakan komponen jam, menit, dan detik, serta tahun, bulan, dan hari (SUBA, 2014).

2.3 Sistem Kontrol

Sistem kontrol merupakan suatu kumpulan cara atau metode yang dipelajari dari kebiasaan-kebiasaan manusia dalam bekerja, dimana manusia membutuhkan suatu pengamatan kualitas dari apa yang telah mereka kerjakan sehingga memiliki karakteristik sesuai dengan yang diharapkan pada mulanya. Terdapat dua jenis sistem kontrol, yaitu sistem kontrol *loop* terbuka dan sistem kontrol *loop* tertutup.

2.3.1 Sistem Kontrol *Loop* Terbuka (*Open-Loop Control System*)

Suatu sistem kontrol yang mempunyai karakteristik dimana nilai keluaran tidak memberikan pengaruh pada aksi kontrol disebut Sistem kontrol *loop* terbuka (*open loop control system*).



Gambar 2.2 Kontrol *Loop* terbuka (Triwiyatno, 2011)

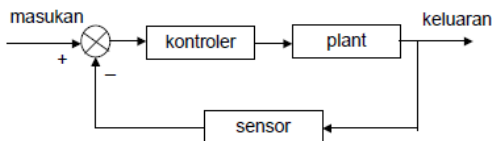
Sistem kontrol *loop* terbuka ini memang lebih sederhana, murah, dan mudah dalam desainnya, akan tetapi akan menjadi tidak stabil dan seringkali memiliki tingkat kesalahan yang besar bila diberikan gangguan dari luar. Kesalahan yang terjadi akibat tidak adanya *feedback* dari *output system*. Terkadang output yang

dihasilkan oleh sistem ini memiliki nilai yang lebih tinggi dari set point.

Contoh dari sistem *loop* terbuka adalah operasi mesin cuci. Penggilingan pakaian, pemberian sabun, dan pengeringan yang bekerja sebagai operasi mesin cuci tidak akan berubah (hanya sesuai dengan yang diinginkan seperti semula) walaupun tingkat kebersihan pakaian (sebagai keluaran sistem) kurang baik akibat adanya faktor-faktor yang kemungkinan tidak diprediksikan sebelumnya.

2.3.2 Sistem Kontrol *Loop* Tertutup (*Closed-Loop Control System*)

Sistem kontrol *loop* tertutup adalah identik dengan sistem kontrol umpan balik, dimana nilai dari keluaran akan ikut mempengaruhi pada aksi kontrolnya. Dibandingkan dengan sistem kontrol *loop* terbuka, sistem kontrol *loop* tertutup memang lebih rumit, mahal, dan sulit dalam desain. Akan tetapi tingkat kestabilannya yang relatif konstan dan tingkat kesalahannya yang kecil bila terdapat gangguan dari luar, membuat sistem kontrol ini lebih banyak menjadi pilihan para perancang sistem kontrol. Karena terdapat *feedback* yang akan diolah kembali oleh kontroler sehingga data yang didapatkan sesuai dengan setpoint. Dimana *feedback* itu didapatkan dari pembacaan sensor.



Gambar 2.3 Kontrol *Loop* tertutup (Triwiyatno, 2011)

Contoh dari sistem ini banyak sekali, salah satu contohnya adalah operasi pendinginan udara (AC). Masukan dari sistem AC adalah derajat suhu yang diinginkan si pemakai. Keluarannya berupa udara dingin yang akan mempengaruhi suhu ruangan sehingga suhu ruangan diharapkan akan sama dengan suhu yang diinginkan. Dengan memberikan umpan balik berupa derajat suhu ruangan setelah diberikan aksi udara dingin, maka akan didapatkan kesalahan (*error*) dari derajat suhu aktual dengan derajat suhu yang diinginkan. Adanya kesalahan ini membuat kontroler berusaha

memperbaikinya sehingga didapatkan *error* yang semakin lama semakin mengecil (Triwiyatno, 2011).

2.4 Pengendalian *Mode On-Off*

Mode kontrol *two-position* atau biasa disebut *mode* kontrol *on-off*. Dalam sistem kontrol *on-off*, elemen penggerak hanya memiliki dua posisi tetap, yang, dalam banyak kasus, hanya hidup dan mati. Kontrol jenis *on-off* relatif sederhana dan murah dan, untuk digunakan dalam sistem kontrol industri dan domestik. Dengan sinyal output dari pengontrol menjadi $u(t)$ dan sinyal *error* yang digerakkan menjadi $e(t)$. Dalam kontrol *on-off*, sinyal $u(t)$ tetap pada nilai maksimum atau minimum, tergantung pada apakah sinyal *error* yang digerakkan adalah positif atau negatif, sehingga di mana U_1 dan U_2 adalah konstanta. Nilai minimum U_2 biasanya nol atau $-U_1$. Pengontrol dua posisi pada umumnya adalah perangkat listrik, dan katup yang dioperasikan *solenoid* listrik banyak digunakan pada pengontrol tersebut (Ogata, 2010).

Pengontrol secara *on-off* merupakan pengontrol paling sederhana dan tangguh (*robust*) namun nilai akurasi menjadi batasan ketika memerlukan pengontrolan akurasi tinggi karena *variabel* kendali (*controlled variable*) sistem selalu mengandung osilasi. Pengontrol *on-off* banyak diterapkan pada sistem pengontrolan di rumah tangga dan banyak digunakan seperti pada sistem penampung air, setrika listrik, *oven* listrik dan pemanas air (Hidayat, Rahmatika, Syafei, & Tumbelaka, 2018).

2.6 Respon Sistem

Dalam sistem pengendalian respon sistem adalah perubahan perilaku output terhadap perubahan sinyal *input*. Respon sistem berupa kurva ini akan menjadi dasar untuk menganalisa karakteristik sistem. Bentuk kurva respon sistem dapat dilihat setelah sistem mendapatkan sinyal *input*. Sinyal input yang diberikan untuk mengetahui karakteristik sistem disebut sinyal test. Ada 3 tipe *input* sinyal *test* yang digunakan untuk analisa sistem dari bentuk kurva respon:

- *Impulse signal*, (sinyal kejut sesaat) :
Respon suatu sistem ketika diberikan *input* berupa sinyal *impulse*.

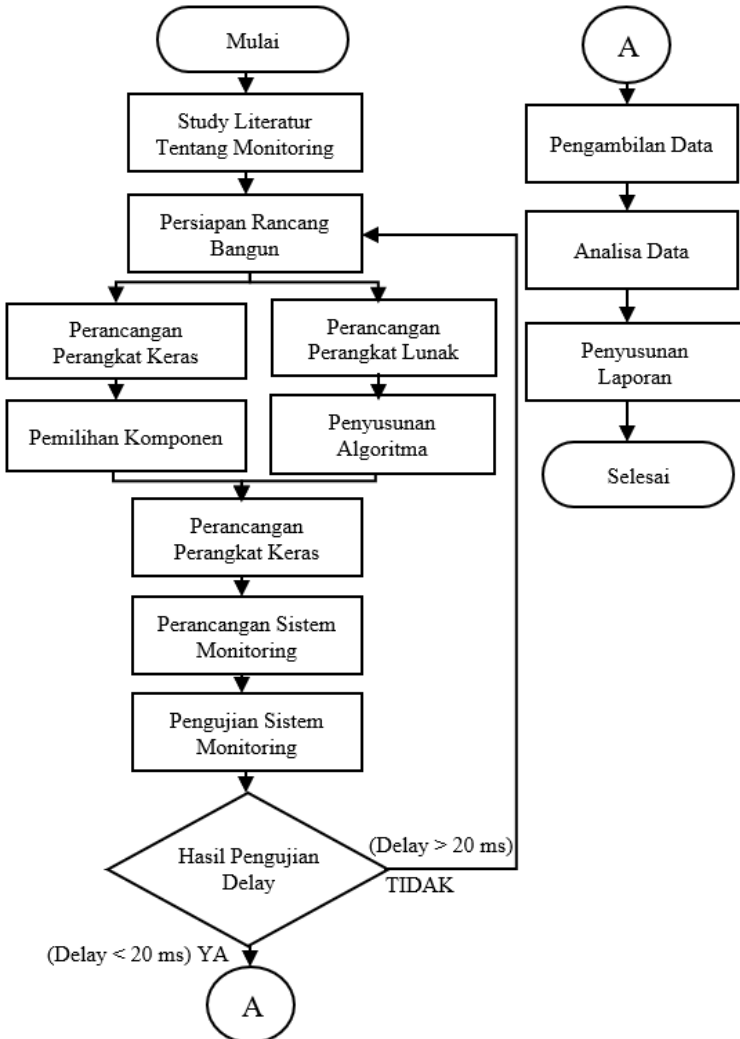
- *Step signal*, (sinyal input tetap DC secara mendadak) Respon suatu sistem ketika diberikan *input* berupa sinyal *unit step*.
- *Ramp signal*, (sinyal yang berubah secara berangsur)

Suatu sistem dianalisa kinerjanya menggunakan sinyal tes. Sinyal tes yang sering digunakan adalah *impulse*, *unit step* dan *ramp*. Respon *impulse* suatu sistem adalah fungsi alih sistem tersebut (Triwiyatno, 2011).

BAB III METODOLOGI PENELITIAN

3.1 Perancangan Sistem

Perancangan sistem pada tugas akhir ini memiliki langkah-langkah sebagai berikut ini yang dibuat dalam bentuk *flowchart*:



Gambar 3.1 *Flowchart* perancangan sistem

Gambar 3.1 merupakan *Flowchart* pengerjaan tugas akhir dari mulai hingga selesai. Tahap awal pengerjaan dimulai dengan studi literatur sebagai pemahaman materi yang menunjang tugas akhir. Ketika literatur didapatkan dilanjutkan ke identifikasi dan pemodelan sistem lalu dilanjutkan dengan perancangan model sistem. Dilanjutkan Pembuatan *hardware* dan *software* kemudian integrasi dan pengujian alat. Pengujian pada sistem ini adalah untuk melihat time respon antara yang ditampilkan pada software dengan hardware sudah sesuai yaitu dibawah 1 detik. Apabila belum terpenuhi maka dilakukan kajian pada pembuatan software dan integrasi. Apabila plant dapat dijalankan dan memenuhi set point maka dilanjutkan ke pengujian

Pengujian dilakukan sesuai dengan yang diharapkan dimana pembacaan yang ditampilkan pada *smartphone* dan *PC* dengan pembacaan pada layar *LCD* pada plant sama. Setelah data dapat diambil, diolah dan dianalisis barulah dapat diambil kesimpulan yang terstruktur.

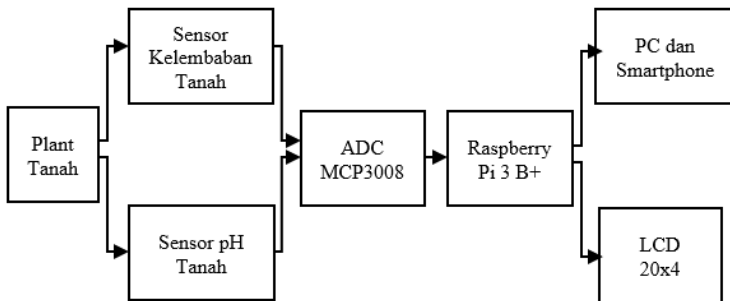
Dalam memudahkan proses *monitoring* dapat mengandalkan *smartphone* yang dapat memantau kondisi plant dari jarak jauh karena sudah berbasis IoT. *Raspberry Pi* pada plant ini digunakan sebagai penyalur data ke database sekaligus menjadi *server* yang akan dijadikan *web server*. *Webserver* yang digunakan adalah *apache 2* dimana aplikasi ini akan di install pada *raspberry*. Dan tujuan akhirnya adalah menampilkan data pada *Web Browser*.

3.2 Studi Literatur

Studi Literatur ini dilakukan dengan mencari materi penunjang dari buku, jurnal, dan artikel pada media cetak maupun online. Studi literatur ini juga untuk memperkuat pemahaman akan pembacaan sensor dan sistem *monitoring* berbasis *web*. Mencari dasar teori yang tepat dalam merancang dan dasar teori tentang monitoring dengan *LCD I2C*. Melakukan studi literatur yang berhubungan dengan pembuatan monitoring berbasis IoT yang dapat membantu pengerjaan plant. Pemilihan studi Literatur yang benar dapat memberikan kemudahan dalam pengerjaan plant. Kemudahan yang dimaksud adalah dasar dari kesalahan pengerjaan dan mencari solusinya.

3.3 Identifikasi dan Pemodelan Sistem

Identifikasi dan pemodelan *system* dilakukan untuk mendapatkan pemodelan hardware, *software*, dan elektrik dalam merancang sistem *monitoring* berbasis *web* pada pengondisian kelembaban dan pH tanah menggunakan *Raspberry Pi 3 B+*. Identifikasi dan pemodelan *system* dilakukan untuk mendapatkan pemodelan *hardware*, *software*, dan elektrik dalam merancang sistem *monitoring* berbasis *web* pada plan tanah menggunakan *Raspberry Pi 3*. Hal ini sangat penting sebelum dilaksanakannya perakitan alat agar tidak terjadi *error* yang signifikan.



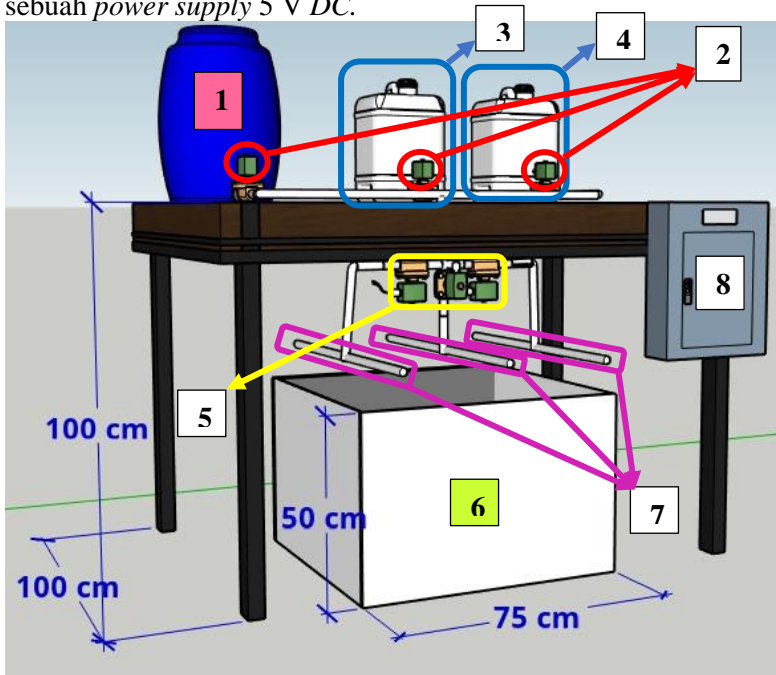
Gambar 3.2 Diagram blok *monitoring*

Diagram blok tersebut menjelaskan mengenai alur proses sistem monitoring. Sensor diletakkan pada plan yaitu sensor kelembaban dan sensor pH tanah. Sinyal yang dikirimkan sensor akan diproses oleh *raspberry pi* menjadi data pembacaan kelembaban dalam satuan persen, kadar pH dan ketinggian air drum dalam satuan centimeter. selanjutnya data tersebut ditampilkan pada LCD monitor 20x4 dan disimpan pada *database raspberry webservice*. Untuk mempermudah tampilan data dari *raspberry* ke *web* maka dibutuhkan *integrasi* antara *database* dan *Webservice*, yang selanjutnya bisa ditampilkan pada *interface* perangkat keras dengan membuka *localhost/IP/folder*. Kegiatan monitoring harus dalam satu akses point yang sama. Dikarenakan monitoring berbasis *IoT* ini menggunakan *localhost* yang dimiliki oleh *raspberry* itu sendiri. *Localhost* yang digunakan harus dapat mendukung masukan dari *database MySQL*

3.4 Perancangan Model Sistem

3.4.1 Perancangan Alat

Desain plan pengendalian pada pengondisian kadar kelembaban dan pH tanah ini dirancang sedemikian rupa dimana terdapat 3 blok yang membedakan lokasi sensor berada. Berikut adalah rancang bangun alat pengondisian kelembaban dan pH tanah yang terdiri sebuah meja yang dijadikan tempat bekerja dari *plant*. Terdapat 2 tangki putih yang berfungsi sebagai penyimpanan larutan asam dan basa, tangka biru sebagai penyimpanan air, tiga motor servo sebagai aktuator dan tiga *solenoid valve* sebagai pembuangan air campuran. Terdapat sebuah panelbox yang digunakan untuk menyimpan *raspberry*, *relay*, *ADC MCP3008* dan sebuah *power supply 5 V DC*.



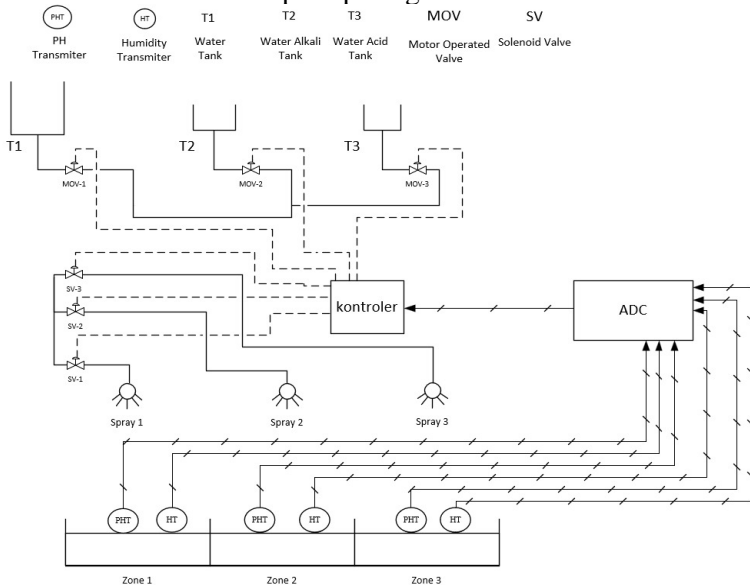
Gambar 3.3 Design plan pengondisian kelembaban dan pH tanah

Keterangan dari gambar diatas:

1. Tangki Air 150 liter
2. Motor *Servo* MG995
3. Tangki Asam 50 liter

4. Tangka Basa 50 liter
5. *Solenoid Valve* AC 220 Volt
6. *Box Plant*
7. *Spray*
8. *Panel Box*

Dari gambar 3.3 dapat kita rangkai sebuah *P&ID* yang nantinya dijadikan patokan pemasangan sensor dan aktuator. Dapat dibuat sebuah *piping flow diagram* dan *piping* pada plan pengendalian pH dan kelembaban tanah seperti pada gambar dibawah ini.



Gambar 3.4 Design P&ID

Dari *P&ID* diatas dapat kita bayangkan peletakan sensor dan aktuator pada plan pengatur kadar pH tanah dan kelembaban tanah otomatis.

3.4.2 Perancangan *Hardware*

Perancangan *hardware* diawali dengan melakukan pembuatan rangka meja. Kemudian tiga drum terbuat dari plastik dengan 2 warna berbeda. Terdapat dua drum putih akan diisi cairan asam dan cairan basa sedangkan satu drum biru diisi air. Digunakan tiga *motor servo* untuk mengatur keluaran air, air asam dan air basa. Pada pipa dibawah meja dilengkapi dengan tiga *solenoid valve*.

Dilanjutkan dengan pemasangan *wiring* pada *plant* dan pemasangan komponen pada panel *box*.

Adapun sensor yang digunakan pada sistem *monitoring plan* pengondisian kadar kelembaban dan pH tanah adalah sebagai berikut :

1. *Soil moisture sensor*

Soil Moisture Sensor Module adalah suatu modul sensor yang berfungsi untuk mendeteksi tingkat kelembaban tanah. Sensor ini bersifat kapasitif dan memiliki nilai pembacaan analog.



Gambar 3.5 *Soil Moisture Sensor* (Dfrobot, 2019)

2. *PH Meter Kit SEN0161 Sensor*

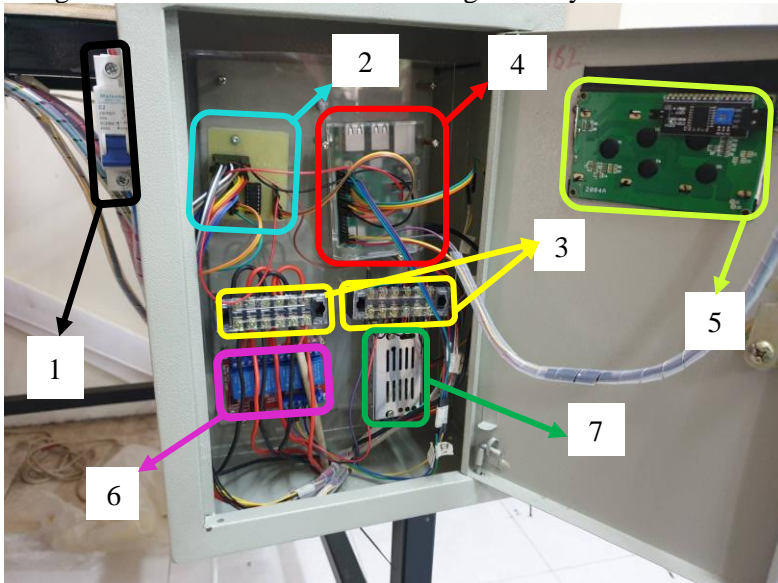
Sensor ini sering digunakan untuk sensing larutan yang membutuhkan akurasi tinggi. Input dari sensor ini yaitu tegangan sebesar 3.3v sampai 5v.



Gambar 3.6 *PH Meter SEN0161 Sensor* (Dfrobot, 2019)

Dilakukan perancangan panel dengan menggunakan komponen *MCB*, *relay 4 channel*, *Raspberry Pi 3 b+3 b+*, terminal AC, *ADC MCP 3008*, *power supply 5 V DC*, *adaptor*, kabel *USB*, jumper, kabel tunggal, kabel serabut dan *cable duct*.

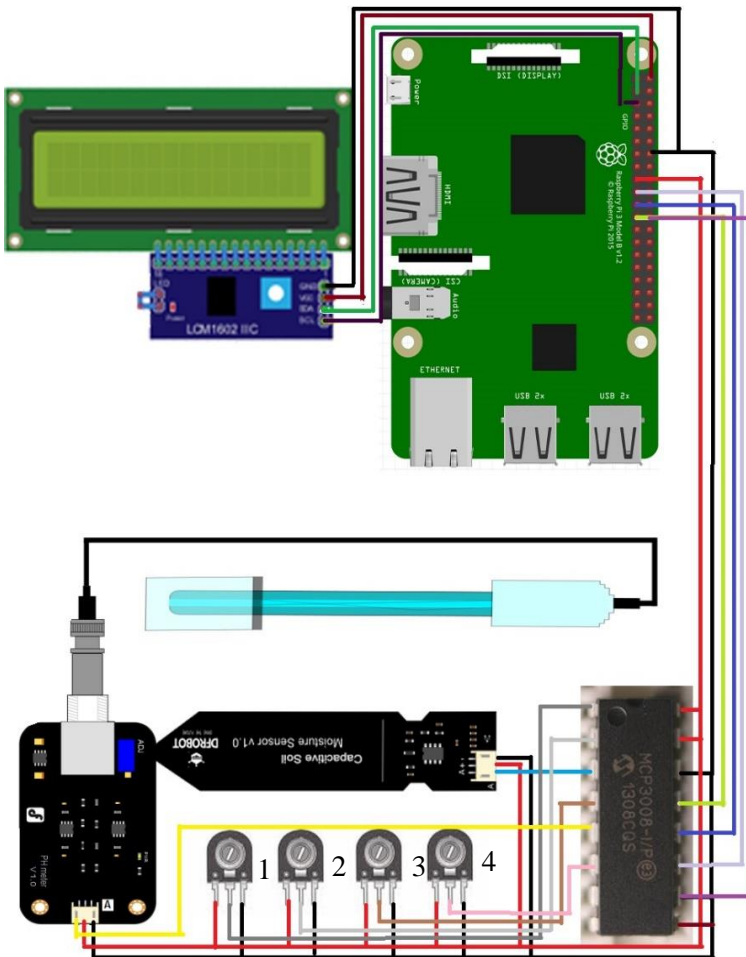
Sumber tegangan PLN dihubungkan ke *MCB* yang mana kabel dengan tegangan fasa dihubungkan ke sloket *MCB* dan netral ke terminal AC. Output dari *MCP* dihubungkan ke terminal AC. Terminal AC menghubungkan *Power Supply 5*, *Raspberry Pi 3 B+* dan *relay*. Ketika *MCB* dalam kondisi *ON* semua komponen mendapatkan daya. *Raspberry* mendapat daya dari adaptor *charger* dengan kabel *USB*. *Relay* mendapat *power* dari *power supply* dan *motor servo* mendapat daya dari *power supply*. Sedangkan *LCD* sendiri mendapat daya dari *Raspberry*. Sinyal sensor dibaca terlebih dahulu oleh *ADC* sebelum masuk *Raspberry*. *Raspberry* memberikan daya ke semua sensor dan *ADC* karena membutuhkan *VCC* sebesar *3.3V DC*. Jika menggunakan *5V DC* dari *power supply* akan menyebabkan terbakarnya komponen *ADC* dan menyebabkan *trouble* yang berkepanjangan. *Raspberry* juga mengatur bukaan *motor servo* dan mengirim sinyal ke *LCD*.



Gambar 3.7 Komponen pada panel

Gambar 3.7 dirancang dengan beberapa komponen seperti:

1. MCB
2. ADC MCP3008
3. Terminal 12 slot.
4. *Raspberry Pi 3 b+*
5. LCD display 20x4
6. Relay 4 channel
7. VCC 5V



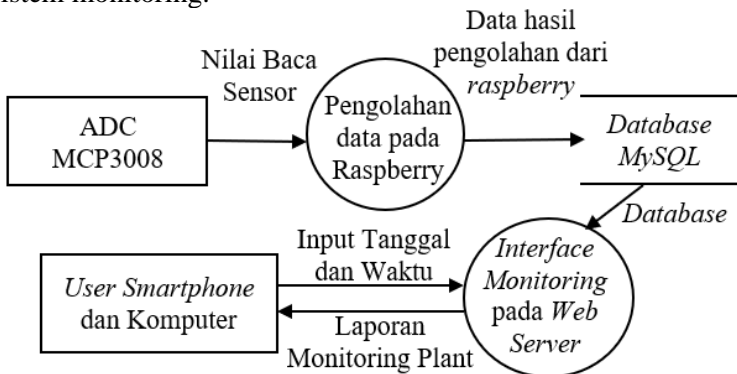
Gambar 3.8 Wiring Proses Monitoring

Gambar 3.8 adalah gambar proses monitoring secara keseluruhan dimana semua sensor diberikan $VCC\ 3.3\ V\ DC$ dan *LCD* diberikan $VCC\ 5V\ DC$. Berikut adalah keterangan dari wiring proses *monitoring*.

1. █ $VCC\ 3.3V\ DC$
2. █ *Ground*
3. █ $VCC\ 5V\ DC$
4. █ *Input analog potensio 1 menuju channel 0 MCP3008*
5. █ *Input analog potensio 2 menuju channel 1 MCP3008*
6. █ *Input analog potensio 3 menuju channel 3 MCP3008*
7. █ *Input analog potensio 4 menuju channel 5 MCP3008*
8. █ *Input analog sensor RH menuju channel 2 MCP3008*
9. █ *Input analog sensor pH menuju channel 4 MCP3008*
10. █ *SCLK MCP3008 menuju SCLK raspberry*
11. █ *MISO MCP3008 menuju MISO raspberry*
12. █ *MOSI MCP3008 menuju MOSI raspberry*
13. █ *CE MCP3008 menuju CE0 raspberry*
14. █ *SDA LCD menuju SDA raspberry*
15. █ *SCL LCD menuju SCL raspberry*

3.4.3 Perancangan *Software*

Hal pertama yang dilakukan dalam pembuatan *software* adalah menginstall OS pada *Raspberry Pi 3 B+*. OS yang digunakan pada *Raspberry Pi 3 B+* ini adalah *Raspbian*. Sebelum melakukan pembuatan *software* dilakukan perancangan *data flow diagram* dari sistem monitoring. Berikut ini adalah *data flow diagram* dari sistem monitoring.



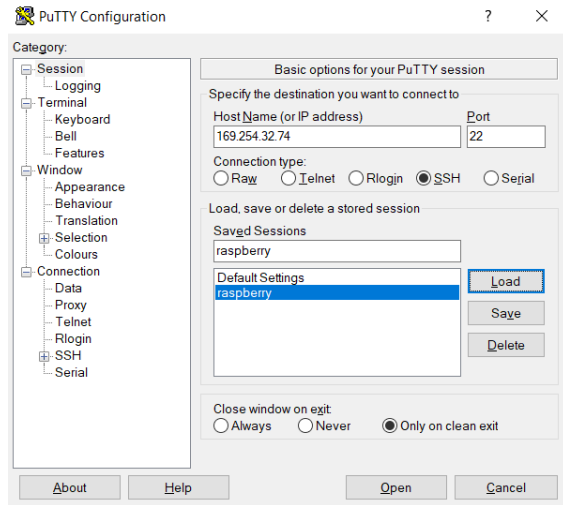
Gambar 3.10 *Data Flow Diagram Monitoring*

Gambar 3.10 menjelaskan bagaimana proses integrasi antara *raspberry* dengan *web server* dari *localhost* yang dimiliki oleh *raspberry*. *Server* yang digunakan adalah *server* dari *raspberry* itu sendiri. Proses bermula dari pembacaan sensor oleh *raspberry pi* yang selanjutnya akan diolah datanya saat proses kalibrasi. Hasil pengolahan data dikirim melalui *localhost* ke *MySQL* dan disimpan. Setelah proses penyimpanan, dilanjutkan ke proses penampilan data pada *smartphone* dan *PC*. Dimana proses penampilan ini menggunakan Bahasa *PHP* sebagai Bahasa yang akan membuat *interface* pada *localhost*. Pada *webserver* terdapat proses menerima dan meminta data. Karena pada user atau pada perangkat keras *smartphone* dan *PC* terdapat mode *search* yang digunakan untuk menampilkan data tertentu pada halaman *web*. Penggunaan *search* diperuntukan kepada user yang ingin mencari data secara spesifik dan menyeluruh.

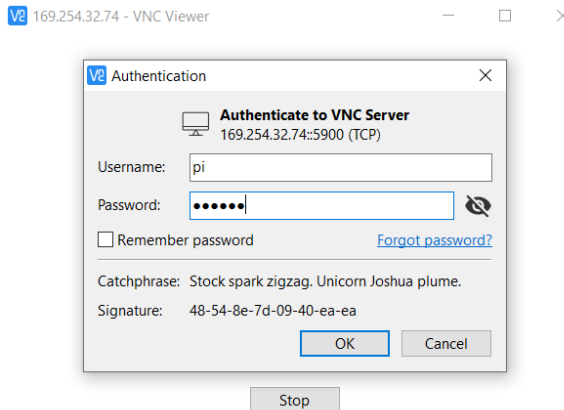
Langkah selanjutnya adalah membuat program pada *Raspberrry* dengan menggunakan bahasa *python*. Bahasa *python* digunakan untuk melakukan pembacaan sensor menampilkan hasil baca dan menghubungkan ke *database MySQL phpMyAdmin*. Pembuatan *listing code* menggunakan *software Thony Python*. Menginstal aplikasi di *Raspberrry Pi* harus menggunakan terminal. Terminal yang digunakan untuk mengaksesnya adalah *Putty*. Pada proses mengkoneksikan *raspberry* dengan *PC* ada beberapa cara yaitu dengan menggunakan koneksi ethernet atau dengan menggunakan koneksi *wifi*. Dimana pada koneksi tersebut terdapat sebuah *IP Adress* yang digunakan sebagai inisial perangkat yang akan diakses oleh *PC*. Namun bisa juga dengan menjalankan *raspberry* sebagai *PC* dengan menghubungkan *raspberry* dengan *Monitor, Mouse, Keyboard*. Proses penyimpanan data menggunakan *MySQL* menggunakan aplikasi *MariaDB*. *MariaDB* adalah aplikasi yang dikembangkan oleh *MySQL* sendiri untuk mempermudah pengguna dalam mengaksesnya. Proses penampilan data dilakukan pada *LCD I2C*. Terdapat beberapa prosedur atau langkah-langkah yang harus dilaksanakan agar *raspberry* dapat digunakan untuk membuat sebuah rancang bangun *monitoring* online kadar kelembaban dan pH tanah menggunakan *webserver*. Berikut adalah langkah-langkah mengakses *raspberry* dari *PC* sampai

menginstal MySQL, *phpMyAdmin*, *Localhost* dan mengintegrasikan *plant*.

1. Instal dan buka Aplikasi *Putty* atau *VNC view*



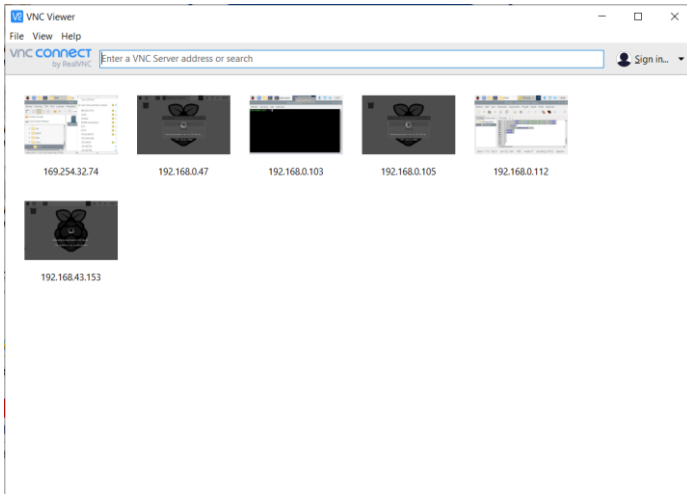
Gambar 3.11 *Login Putty*



Gambar 3.12 *Login VNC view*

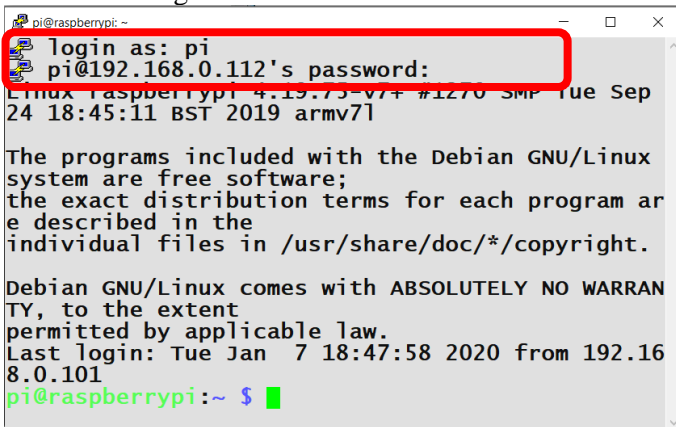
Ketika melakukan *login* ke *VNC* atau *Putty* pastikan koneksi *LAN* terhubung dan menggunakan kabel yang kualitasnya baik. Tujuannya adalah menghindari kehilangan data. Saat login ingat *password raspberrry* dengan baik. Karena jika terjadi kesalahan *login* secara berturut-turut maka user akan di blokir.

2. Tampilan aplikasi *VNC view* yang merupakan aplikasi yang dapat menampilkan gambar desktop *Raspberry* secara *real*.



Gambar 3.13 Start up *VNC view*

3. Kembali ke *putty* dan *login* akun untuk mempermudah menginstall aplikasi dengan terminal yang sudah terkoneksi dengan kabel LAN.

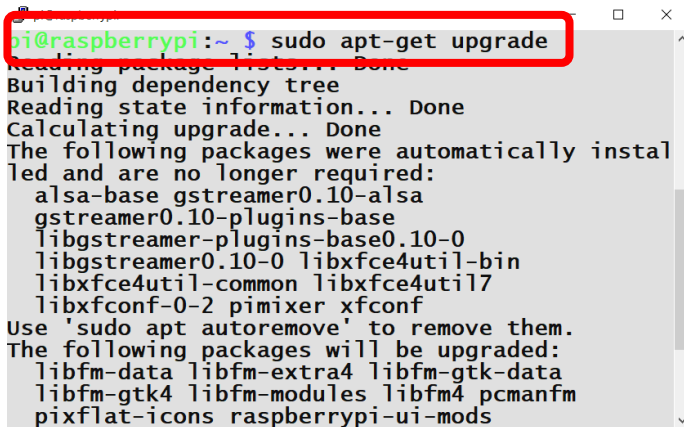


Gambar 3.14 Start up *putty*

Dua gambar diatas adalah tampilan awal setelah *login* berhasil dan *user* dapat mengakses *raspberry* pada laptop. Ketika terjadi

trouble coba lihat kabel LAN atau *setting raspberry* dengan menggunakan *monitor, keyboard* dan *mouse*. Karena raspberry pi adalah mini PC yang dapat diakses dengan perangkat tersebut.

4. *Upgrade raspberry* dengan perintah “`sudo get-apt upgrade`”



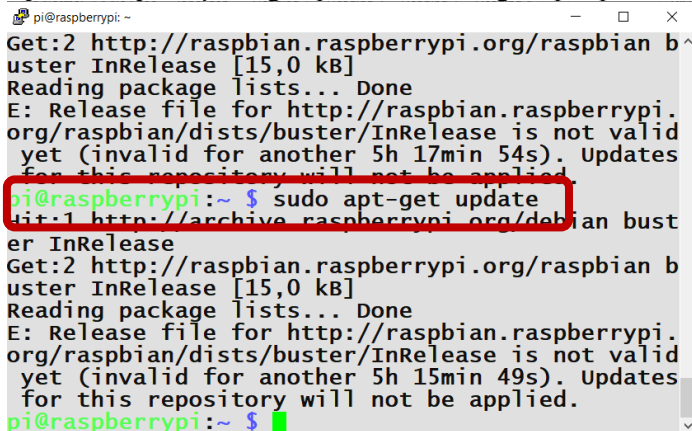
```

pi@raspberrypi:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed
and are no longer required:
  alsabase gstreamer0.10-alsabase
  gstreamer0.10-plugins-base
  libgstreamer-plugins-base0.10-0
  libgstreamer0.10-0 libxfce4util-bin
  libxfce4util-common libxfce4util7
  libxfceconf-0-2 pimixer xfconf
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  libfm-data libfm-extra4 libfm-gtk-data
  libfm-gtk4 libfm-modules libfm4 pcmanfm
  pixflat-icons raspberrypi-ui-mods

```

Gambar 3.15 Tampilan *Upgrade*

5. *Update raspberry* dengan perintah “`sudo get-apt update`”



```

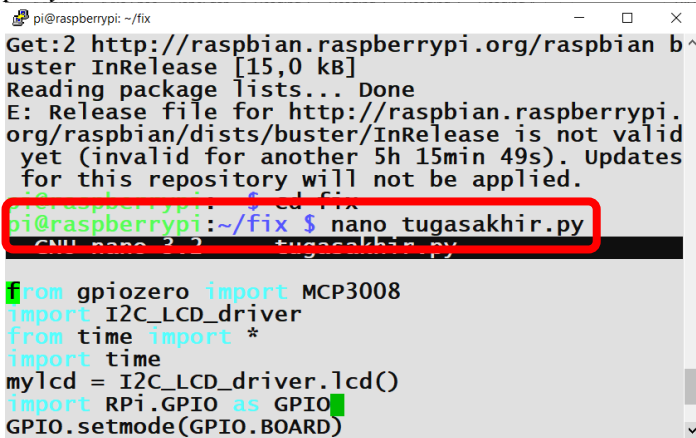
pi@raspberrypi:~ $ sudo apt-get update
Get:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Reading package lists... Done
E: Release file for http://raspbian.raspberrypi.org/raspbian/dists/buster/InRelease is not valid yet (invalid for another 5h 17min 54s). Updates for this repository will not be applied.
pi@raspberrypi:~ $ sudo apt-get update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Get:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Reading package lists... Done
E: Release file for http://raspbian.raspberrypi.org/raspbian/dists/buster/InRelease is not valid yet (invalid for another 5h 15min 49s). Updates for this repository will not be applied.
pi@raspberrypi:~ $

```

Gambar 3.16 Tampilan *Update*

Dua tampilan di atas digunakan untuk melakukan *upgrade* dan *Update raspberry* agar tidak terjadi *crash* saat proses *install* aplikasi karena terdapat aplikasi pendukung yang belum di *update* dan di *upgrade*.

6. Listing Code Sensing Sensor dan aktuator pada terminal *putty*



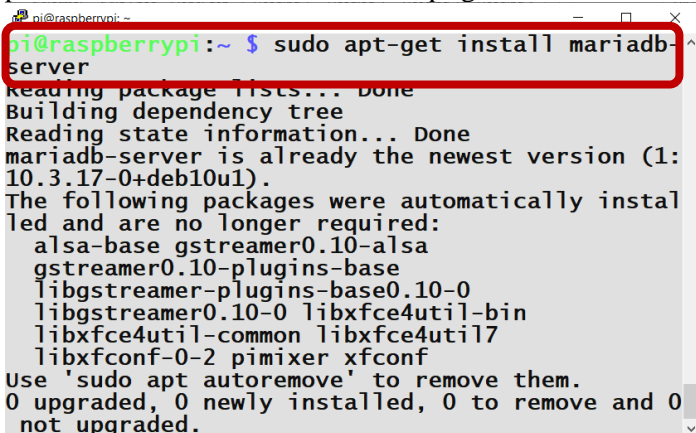
```

pi@raspberrypi: ~/fix
Get:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Reading package lists... Done
E: Release file for http://raspbian.raspberrypi.org/raspbian/dists/buster/InRelease is not valid yet (invalid for another 5h 15min 49s). Updates for this repository will not be applied.
pi@raspberrypi:~/fix $ nano tugasakhir.py
GNU nano 2.2.6 tugasakhir.py
from gpiozero import MCP3008
import I2C_LCD_driver
from time import *
import time
mylcd = I2C_LCD_driver.lcd()
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)

```

Gambar 3.17 Perintah nano

7. Menginstal *mariadb* atau *mysql* versi terbaru dengan perintah melakukan *install* “`sudo apt-get mariadb-server`”



```

pi@raspberrypi:~ $ sudo apt-get install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
mariadb-server is already the newest version (1:10.3.17-0+deb10u1).
The following packages were automatically installed and are no longer required:
  alsabase gstreamer0.10-alsa
  gstreamer0.10-plugins-base
  libgstreamer-plugins-base0.10-0
  libgstreamer0.10-0 libxfce4util-bin
  libxfce4util-common libxfce4util7
  libxfconf-0-2 pimixer xfconf
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

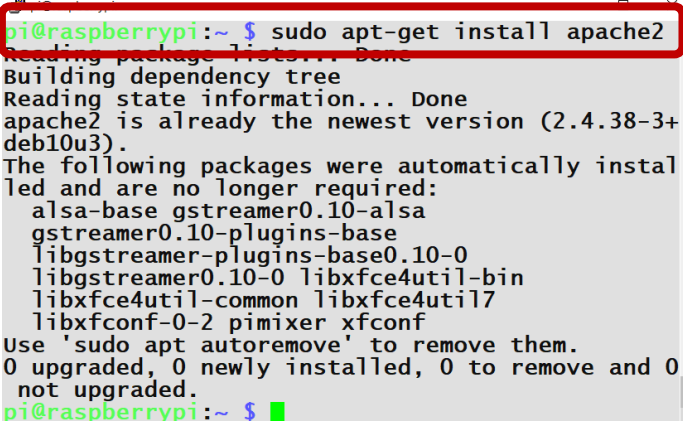
```

Gambar 3.18 Install *mariadb*

8. Menginstal *WebServer* dengan perintah “`sudo get-apt apache2`”

Ketika proses penginstalan terdapat baris yang memberi saran [Y/n]. ketik pada *putty* “Y” kemudian tekan tombol *enter* agar proses dapat berjalan. Kemudian pada

proses ini akan muncul tampilan konfigurasi dimana kita akan membuka akses point kita disana.



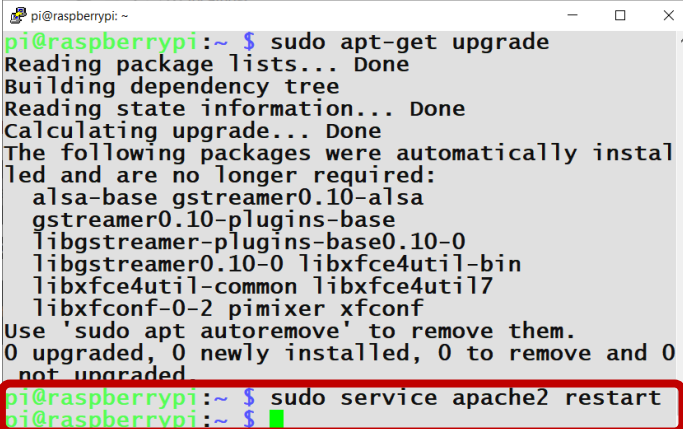
```

pi@raspberrypi:~ $ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.38-3+deb10u3).
The following packages were automatically installed and are no longer required:
  alsa-base gstreamer0.10-alsa
  gstreamer0.10-plugins-base
  libgstreamer-plugins-base0.10-0
  libgstreamer0.10-0 libxfce4util-bin
  libxfce4util-common libxfce4util7
  libxfconf-0-2 pimixer xfconf
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $

```

Gambar 3.19 *Install apache2*

- Melakukan *restart* pada *Apache2* “`sudo service apache2 restart`”



```

pi@raspberrypi:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  alsa-base gstreamer0.10-alsa
  gstreamer0.10-plugins-base
  libgstreamer-plugins-base0.10-0
  libgstreamer0.10-0 libxfce4util-bin
  libxfce4util-common libxfce4util7
  libxfconf-0-2 pimixer xfconf
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $ sudo service apache2 restart
pi@raspberrypi:~ $

```

Gambar 3.20 *restart apache2*

- Selanjutnya menginstall *phpmyadmin* dengan mengetikkan “`sudo apt install phpmyadmin`”

Ketika proses penginstallan terdapat baris yang memberi saran [Y/n]. ketik pada putty “Y” + enter. Kemudian pada proses ini akan muncul konfigurasi *password* pertama dan kedua untuk memverifikasi *password*

```

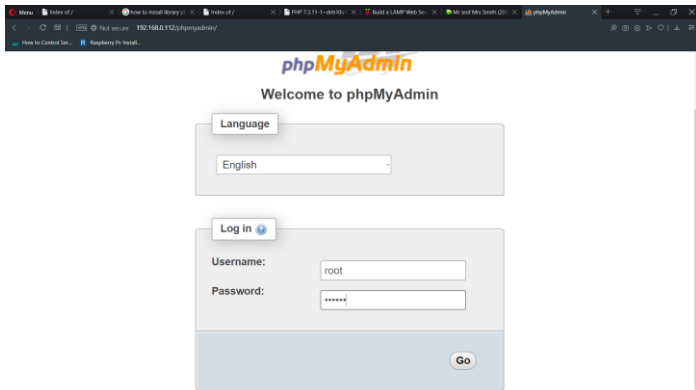
pi@raspberrypi:~ $ sudo apt-get install phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
phpmyadmin is already the newest version (4:4.6.6-5).
The following packages were automatically installed and are no longer required:
  alsa-base gstreamer0.10-alsa
  gstreamer0.10-plugins-base
  libgstreamer-plugins-base0.10-0
  libgstreamer0.10-0 libxfce4util-bin
  libxfce4util-common libxfce4util7
  libxfconf-0-2 pimixer xfconf
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

```

Gambar 3.21 *Install phpmyadmin*

11. Menampilkan *phpMyAdmin*

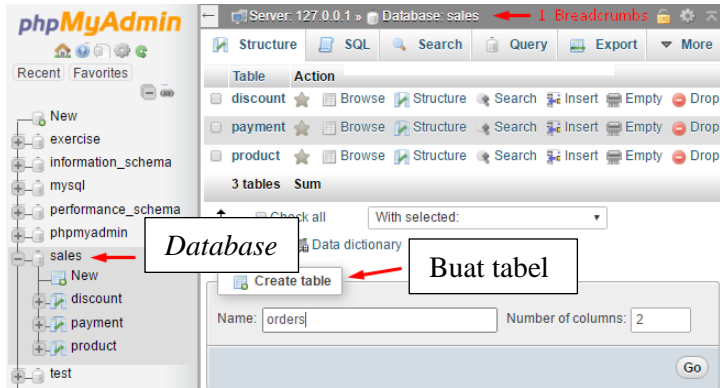
Cara menampilkan local host kita adalah dengan mengetik localhost/phpMyAdmin/ di broser. Masukkan kata sandi dan username.



Gambar 3.22 *login phpMyAdmin*

12. Membuat *structural table* di *phpMyAdmin*

Setelah *phpmyadmin* terpasang dengan sempurna, membuat struktur *database* sesuai dengan kebutuhan *monitoring*. Seperti pada gambar terdapat tiga *table* kelembaban dan tiga *table* pH. Dimana terdapat tiga blok yang membedakan daerah kontrol.

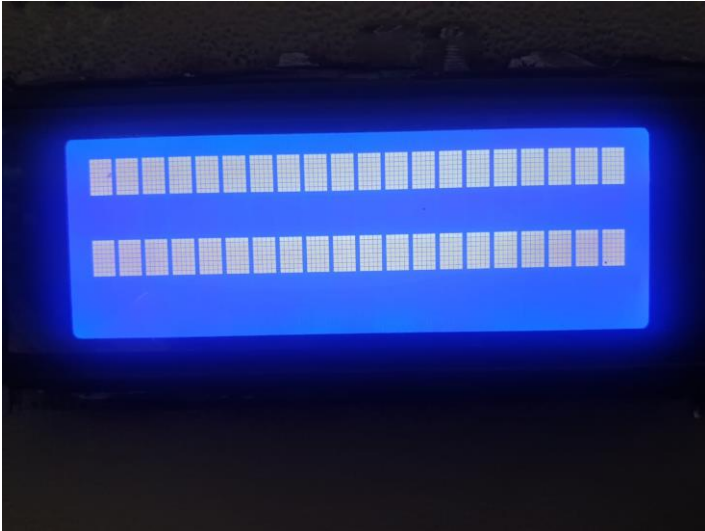


Gambar 3.23 Pembuatan tabel

Untuk bisa mengirim data dari pembacaan sensor yang terpasang pada *raspberry* ke *database* yang telah dibuat dibutuhkan sebuah koneksi internet. Koneksi internet yang dibutuhkan harus memiliki akses point yang sama dengan user yang akan memonitoring data. Karena akses yang digunakan adalah local host bukan public host yang pada dasarnya sangat jauh berbeda. Pada *raspberry* sudah terpasang modul *wifi* sehingga integrasi program python dan *database* lebih mudah.

13. Mengintegrasikan *software* dengan *hardware*

Yang pertama harus dilakukan adalah mengkoneksikan sensor dengan *ADC* untuk membaca nilai sensor dari *ADC* di *raspberry* dengan menggunakan kode *python*. Saat proses terintegrasi maka kita dapat menampilkan hasil pembacaan di terminal dan di layer *LCD 20x4*. Jika menggunakan *LCD I2C* diwajibkan untuk menginstall *library I2C* dan membolehkan akses *I2C* pada *raspberry*. Cara mengaktifkannya adalah masuk ke konfigurasi dan masuk ke *interface* lalu pilih *I2C* kemudian *Enable* konfigurasi tersebut. Kemudian mengintegrasikan *relay* dan *servo* dengan menggunakan perintah yang disebut *General Purpose Input Output (GPIO)*. *GPIO* tersebut akan mengirimkan perintah untuk menjalankan tiga *solenoid valve* yang terintegrasi dengan *relay* dan tiga *motor servo*.



Gambar 3.24 Tampilan LCD 20x4

14. Uji Pengambilan Data

Uji pengambilan data dilakukan untuk mendapatkan data dari sensor yang digunakan. Dalam hal ini menggunakan sensor kelembaban tanah dan pH tanah. Hasil dari pembacaan sensor ini yaitu berupa kadar air pada tanah dan kandungan pH dalam tanah. Dimana hasil pembacaan akan ditampilkan pada monitor dan pada *MySQL* yang terhubung dengan *server*. Dimana data pembacaan sensor akan diunduh dalam bentuk pdf pada localhost yang sudah tersedia.

15. Menganalisis Data

Analisis data digunakan untuk menjawab dan menjelaskan permasalahan yang telah ditemui. Dalam *system monitoring* pada plan pengondisian kelembaban dan pH ini perlu ada perbandingan antara data yang diambil dan ditampilkan pada layar monitor dengan *database* yang telah dibuat. Kemudian menjelaskan jika adanya error dalam *system*. Serta menjelaskan faktor penyebab terjadinya eror pada sistem dan solusinya seperti apa. Dan cara mengatasi masalah tersebut.

16. Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah menganalisis data yang telah didapat. Kesimpulan tersebut akan diolah dan mendapat saran kedepannya agar *system monitoring* berbasis *IoT* ini dapat berkembang dan lebih baik lagi.

Halaman ini sengaja dikosongkan

BAB IV HASIL DAN PEMBAHASAN

Rancang Bangun Sistem *Monitoring* Pengendalian pH dan Kelembaban Tanah Berbasis *Web Server* Menggunakan *Raspberry Pi 3 B+* merupakan sebuah alat yang dapat memantau kadar pH dan kelembaban tanah secara online. Alat ini menggunakan *web server* sebagai interface tools dan database *MySQL* sebagai penyimpanan data pembacaan.

4.1 Hasil Pengujian *hardware*

Pengujian Hardware adalah memastikan keadaan plant sesuai dengan rancangan baik P&ID atau *Wiring*. Memastikan *Raspberry* terhubung dengan aktuator dan sensor. Memastikan hasil pembacaan sensor sama dengan alat ukur yang valid. Hasil pembacaan sensor kelembaban tanah dan kadar ph tanah diuji dengan alat ukur *Three-Way Meter*.



Gambar 4.1 kalibrasi sensor Kelembaban tanah

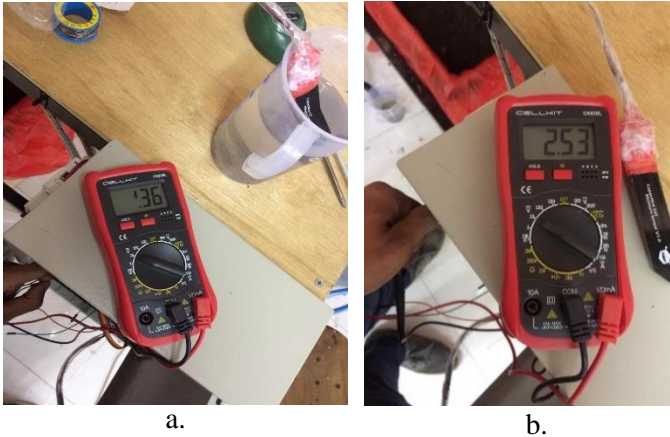
Kalibrasi sensor kelembaban tanah dilakukan dengan cara menambahkan beberapa air kedalam tanah dan dilakukan perhitungan secara standar



Gambar 4.2 kalibrasi sensor pH tanah

kalibrasi sensor pH tanah dilakukan dengan cara menuangkan larutan asam dan larutan basa ke tanah dengan jarak waktu tertentu. Dimana nilai yang di ambil dari sensor akan dibandingkan dengan alat ukur *Three-Way Meter*.

Selanjutnya adalah proses pengujian *hardware* salah satunya *ADC MCP3008*. Proses pengujiannya dilakukan dengan mengukur nilai keluaran *ADC* dengan nilai voltase *output* sensor. *ADC MCP3008* adalah *ADC 10 bit* maka nilai baca maksimalnya adalah 1024. Dapat disimpulkan ketika *ADC* mendapat tegangan penuh sebesar 3.3V maka nilai bacaan *ADC* adalah 1024. Ketika nilai pembacaan *ADC* sama dengan 1 maka nilai *output* dari *VCC* sama dengan 0,0032V. Dilakukan pengujian masukan tegangan dengan menggunakan *AVO meter* dan membandingkannya dengan hasil *konversi*.



Gambar 4.3 (a) Pembacaan di air (b) Pembacaan di udara

Pengujian dilakukan dengan melepas input analog sensor kemudian mengukurnya menggunakan *AVO meter* Digital. Berikut hasil data yang ditetapkan dari proses pengujian output tegangan.

a. *Output ADC MCP3008* dari *input potensio*

Nilai Tegangan Output Sensor Palsu (<i>Potensio</i>)	Nilai baca ADC	Nilai Konversi ADC ke Tegangan	Perbedaan pembacaan
3.2V	1024	3.3V	0,1V
2,7V	907	2,9V	0,2V
1,5V	563	1,8V	0,3V
2.1V	612	1,9V	0,2V
0,9V	344	1,1V	0,2V

Tabel 4.1 Tabel pembacaan *input* sensor palsu dan *output ADC*

b. *Output ADC MCP3008* dari *input* sensor Kelembaban

Nilai Tegangan Output Sensor Kelembaba	Nilai baca ADC	Nilai Konversi ADC ke Tegangan	Perbedaan pembacaan
2,56V	853	2,72V	0,16V
1,36V	487	1,56V	0,2V

Tabel 4.2 Tabel pembacaan *input* sensor kelembaban dan *output ADC*

c. *Output* ADC MCP3008 dari *input* sensor pH

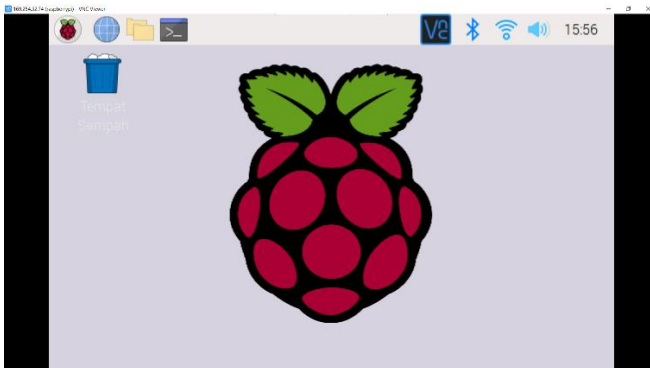
Nilai tegangan <i>output</i> sensor pH	Nilai baca ADC	Nilai Konversi ADC ke Tegangan	Perbedaan pembacaan
1,77V	532	1,72V	0,05V
0,61V	231	0,72V	0,11V

Tabel 4.3 Tabel pembacaan *input* sensor pH dan *output* ADC

Dari table diatas dapat diambil kesimpulan bahwa nilai baca ADC tidak bermasalah dan nilai tegangan yang masuk kedalam ADC MCP3008 tidak melebihi spesifikasi. Dan terjadinya eror pembacaan dikarenakan faktor lingkungan seperti resistansi pada tangan, kabel dan udara.

4.2 Hasil Pengujian *Software*

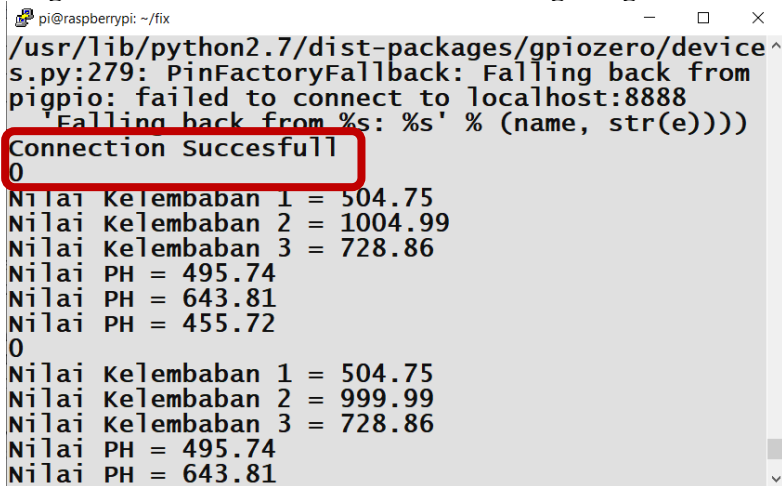
Pengujian *software* bertujuan untuk mengetahui apakah *software* yang telah dibuat dapat bekerja sesuai dengan yang direncanakan. Dalam hal ini, *software* akan diuji dengan cara menghubungkan *Raspberry Pi* dengan menggunakan kabel LAN. Pengujian dilakukan dengan membuka *software* remote desktop bernama VNC viewer. VNC viewer memudahkan kita untuk mengakses raspberry secara real. Berbeda dengan putty, putty hanya mengakses terminal yang ada di raspberry. VNC *viewer* hampir sama seperti *virtual box* tapi *virtual box* hanya satu prosesor dengan dua OS. Berikut adalah tampilan VNC view dengang OS raspberian.



Gambar 4.4 Tampilan awal VNC viewer

4.2.1 Pengujian Input sinyal ADC ke *raspberry*.

Pengujian dilakukan dengan menjalankan *script python* yang telah dibuat pada *terminal Raspberry pi* selama berhari-hari dan hasilnya program dapat berjalan dengan baik. Data dari 6 sensor dapat dibaca dan dikalibrasi. Terdapat tiga blok yang diatur pada Blok Satu terdapat sensor pH1 dan kelembaban1, pada Blok Dua terdapat sensor pH2 dan kelembaban2, pada Blok Tiga terdapat sensor pH3 dan kelembaban3. Pada setiap Blok juga terintegrasi dengan *motor servo* dan *Solenoid Valve* masing-masing.



```

pi@raspberrypi: ~/fix
/usr/lib/python2.7/dist-packages/gpiozero/device
s.py:279: PinFactoryFallback: Falling back from
pigpio: failed to connect to localhost:8888
'Falling back from %s: %s' % (name, str(e)))
Connection Successful
0
Nilai Kelembaban 1 = 504.75
Nilai Kelembaban 2 = 1004.99
Nilai Kelembaban 3 = 728.86
Nilai PH = 495.74
Nilai PH = 643.81
Nilai PH = 455.72
0
Nilai Kelembaban 1 = 504.75
Nilai Kelembaban 2 = 999.99
Nilai Kelembaban 3 = 728.86
Nilai PH = 495.74
Nilai PH = 643.81

```

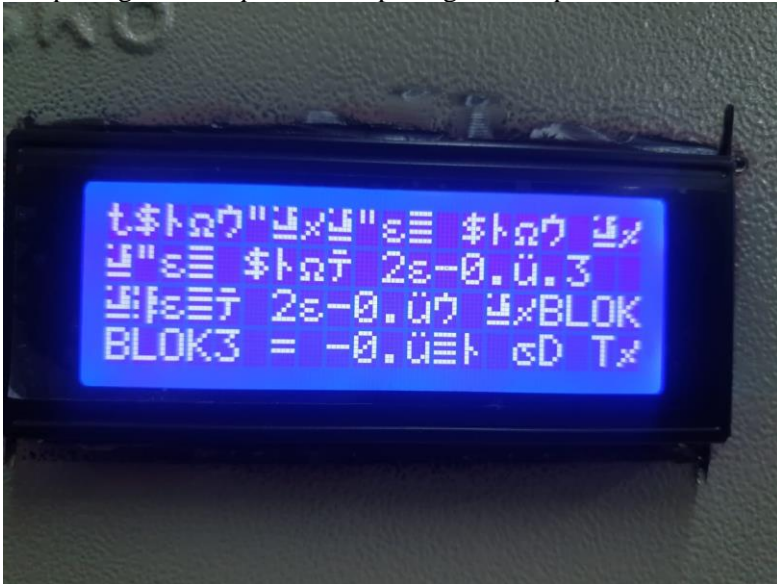
Gambar 4.5 Tampilan saat plan dijalankan

Data dapat dibaca dengan baik oleh *raspberry* dengan pengolahan data yang sesuai dengan kalibrasi sensor. Pada plan ini terdapat sinyal palsu yang bersifat *resistif*. Sinyal palsu tersebut didapat dari sebuah potensio. Jadi pada dasarnya potensio menghambat arus yang mengalir ke *ADC MCP3008*. Dengan sinyal palsu yang diberikan kepada *ADC MCP3008* yang didapat dari *potensio* 100K dan 10K *motor servo* satu dan tiga dapat digerakan. *Potensio* 10k digunakan untuk sinyal palsu pH1 dan kelembaban1 pada blok satu. Sedangkan sinyal palsu dengan potensio 100K digunakan untuk sinyal palsu pH3 dan kelembaban3 pada blok tiga. Kemudian pada blok dua menggunakan sensor asli dengan sifat kapasitif. Terdapat keuntungan sensor kapasitif dibandingkan dengan sensor resistif.

Sensor kapasitif respon nya lebih cepat daripada sensor resistif namun sering terjadi histerisis data.

4.2.2 Pengujian pada layar LCD

Sistem *monitoring* pada pengondisian pH dan kelembaban tanah dapat menyimpan data kadar pH dan kelembaban dalam *database* dan terjadi secara *real time*. *Database* yang digunakan adalah *MySQL Maria-DB*. *Database* ini terintegrasi langsung dengan *Raspberry* dan menyimpan data setiap 5 detik secara otomatis. Dimana hasil *monitoring* akan ditampilkan pada *LCD* dan perangkat *smartphone* atau perangkat komputer.



Gambar 4.6 Tampilan *error* pada *LCD*

Pada *LCD* sempat terjadi *trouble* dikarenakan kekurangan daya akibat beban yang digunakan sangat besar. Tegangan sebagian besar digunakan untuk menggerakkan tiga *motor servo* secara bersamaan dan menggerakkan tiga buah relay yang masing-masing diatur agar menjadi *normally open*. Solusi dari permasalahan tersebut adalah dengan menambahkan jumlah *VCC 5V DC*. Dari spesifikasi power supply tersebut hanya dapat mengeluarkan daya sebesar 5 Volt 2 Ampere. Dimana pada saat pemasangan wiring *motor servo*, *relay* dan *LCD* dirangkai secara seri dan

mengakibatkan kekurangan arus. Setelah sumber diganti dengan menancapkan kabel *VCC* ke *power supply* yang disediakan *raspberry*, *LCD* dapat berjalan secara normal dan terdapat *delay* yang dihitung wajar yaitu 3ms. *Delay* terjadi dikarenakan oleh pengiriman data yang melewati wiring yang Panjang dan memiliki tingkat resistansi yang cukup tinggi.



Gambar 4.7 Tampilan data monitoring kelembaban dan pH tanah pada *LCD*

4.2.3 Pengujian pada *phpMyAdmin*

Pengujian pertama dilakukan dengan cara melakukan pemantauan terhadap koneksi *raspberry* terhadap *server localhost* yang sudah dibuat. Menghubungkan *localhost* yang sama membutuhkan koneksi wifi yang sama atau akses *public* yang sama. Koneksi wifi yang digunakan adalah lab Instrumentasi Industri wifi dengan *IP Adress* 192.168.0.103. *Raspberry* dan perangkat yang akan melakukan proses pengiriman data dikoneksikan ke wifi lab *instrumentasi industry*. Dengan perintah “*sudo mysql -phpMyAdmin -p*” Dimana jika perangkat terhubung dengan server maka akan muncul tampilan *Welcome to the MariaDB monitor*. Tambahkan keyword “*USE MONITORING*” yang artinya akses akan dilanjutkan ke database monitoring yang telah dibuat sebelumnya. Jika akses berhasil maka akan muncul tampilan *database changed* Gambar 4.8 menunjukkan akses ke *table monitoring* berhasil dan data dapat diunggah ke *localhost*.

```

pi@raspberrypi:~ $ sudo mysql -phpmyadmin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.3.17-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE monitoring
Reading table information for completion of table and column names
You can turn off this feature to get a quicker s

```

Gambar 4.8 Tampilan pengujian MariaDB

Dilanjutkan ke pengujian kedua yaitu dengan mengupload data ke *phpMyAdmin* pada *username* “monitoring”. Nama table yang dituju adalah “tugas-akhir”. Pada table tugas akhir terdapat 6 kolom pada kolom pertama sampai kolom ketiga menunjukkan nilai baca sensor kelembaban satu sampai tiga dengan satuan persen (%). Pada kolom empat sampai enam menampilkan data sensor pH satu sampai tiga.

waktu	sensor_klmb_1	sensor_ph_1	sensor_klmb_2	sensor_klmb_3
2019-12-19 08:12:36	1	1	1	1
2019-12-22 20:28:57	42.2156	3.68847	1.12958	100
2019-12-22 20:29:29	44.3854	3.67513	1.12958	100
2019-12-22 20:29:35	42.2156	3.68847	1.12958	100
2019-12-22 20:31:35	42.2156	3.67513	1.12958	100
2019-12-22 20:31:40	42.2156	3.67513	1.12958	100

Gambar 4.9 Tampilan *table* tugas_akhir

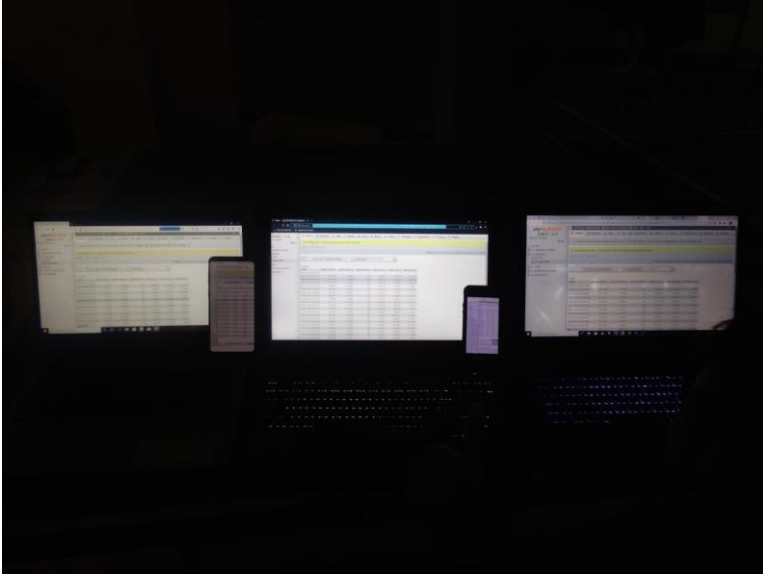
Data yang dimasukkan pada table tugas_akhir adalah data ADC yang sudah diolah dengan proses kalibrasi. Data tersebut sama dengan data yang ditampilkan pada *LCD 20x4*. Dilanjutkan ke pengujian ketiga dengan mengakses localhost secara bersamaan dengan koneksi wifi yaitu lab *instrumentasi industry* dengan IP Adress 192.168.0.112. cara mengaksesnya dengan brosing data pada aplikasi web yang terdapat pada smartphone atau laptop dengan memasukkan keyword “192.168.0.112/TA4.php”.

The screenshot shows a mobile browser interface. At the top, the status bar displays 'TSEL', signal strength, Wi-Fi, time '6:06 PM', and battery '56%'. The address bar contains '192.168.0.112', which is circled in red. Below the address bar, the page title 'Plan Tanah' is visible. A search bar contains the text 'Keyword Tanggal & wakt' and has 'Cari!' and 'Refresh' buttons. A table with the following data is displayed:

waktu	Kelembaban 1	Kelembaban 2	Kelembaban 3
2020-01-08 17:00:41	71.1454	0	58.7384
2020-01-08 17:00:36	70.9044	0	58.7384
2020-01-08 17:00:30	70.1811	0	59.0611
2020-01-08 17:00:25	71.1454	0	58.7384
2020-01-08 17:00:20	70.1811	0	58.7384
2020-01-08 17:00:14	70.1811	0	58.7384
2020-01-07 22:41:19	70.9044	0	58.7384
2020-01-07 22:41:14	71.1454	0	58.0929
2020-01-07 22:41:09	70.1811	0	58.0929
2020-01-07 20:14:25	71.1454	0	47.6039
2020-01-07 20:14:19	71.1454	0	47.6039
2020-01-07 20:14:14	71.1454	0	47.6039
2020-01-07 18:23:49	72.3508	0	48.4107
2020-01-07 18:23:44	73.5562	0	49.0562
2020-01-07 18:16:01	73.2508	0	48.7225

Annotations in the image include a red circle around the IP address in the address bar, a red box labeled 'IP Adress' pointing to it, a blue circle around the search bar, and a blue box labeled 'Search Tools' pointing to the table.

Gambar 4.10 Tampilan *monitoring* tugas_akhir pada *smartphone*



Gambar 4.11 Tampilan *monitoring* online tugas_akhir pada beberapa *device*

Gambar diatas menunjukkan ada beberapa *device* yang dapat masuk ke *localhost* yang telah dibuat. Terdapat tiga laptop dan dua *smartphone* baik *android* maupun *iphone* yang dapat mengakses data tersebut. Akses tersebut berada pada satu router. Ketika *raspberry* dimatikan maka koneksi akan terputus dan perangkat lain tidak dapat melakukan akses *database*. Jarak komunikasi data sekitar kurang lebih 24,7 meter. Jarak koneksi dipengaruhi oleh koneksi *wifi* yang digunakan. Apabila *wifi* yang digunakan memiliki jarak koneksi 100 Meter maka jarak akses data adalah 100 meter.

4.2.4 Pengujian Kualitas *Web Server*

Pengujian kualitas *web server* dilakukan dengan membandingkan pembacaan pada *web server* dengan pembacaan pada *raspberry* (*putty*). Pengujian juga dilakukan dengan mencari jarak akses terjauh *user* dengan *raspberry*. Tujuan dari pengujian ini adalah untuk mengetahui adanya *data error* atau keterlambatan pengiriman data dari *raspberry* ke *database* dan dari *database* ke *web server*. Berikut adalah gambar hasil pengujian tampilan monitoring pada *raspberry*, *database* dan *web server*.

```

pi@raspberrypi: ~/fix
Nilai Kelembaban 1 = 37.6351
Nilai Kelembaban 2 = 1.1672
Nilai Kelembaban 3 = 56.8019
Nilai PH = 6.6366
Nilai PH = 12.1593
Nilai PH = 2.2211
2020-01-23 13:13:22
0
Nilai Kelembaban 1 = 38.1173
Nilai Kelembaban 2 = 0.5002
Nilai Kelembaban 3 = 56.9633
Nilai PH = 6.6366
Nilai PH = 11.9592
Nilai PH = 2.2344
2020-01-23 13:13:27
0
Nilai Kelembaban 1 = 38.3583
Nilai Kelembaban 2 = 0.5002
Nilai Kelembaban 3 = 57.2860
Nilai PH = 6.6232
Nilai PH = 0.0067
Nilai PH = 2.2211
2020-01-23 13:13:32
0

```

Gambar 4.12 Tampilan pengambilan data pada *raspberry*

Time	Kelembaban 1	Kelembaban 2	Kelembaban 3
2020-01-23 13:03:18	38.5994	6.63657	1.16724
2020-01-23 13:04:34	38.5994	6.62323	0.500244
2020-01-23 13:04:39	36.6708	6.66325	1.16724
2020-01-23 13:06:17	38.3583	6.63657	1.16724
2020-01-23 13:06:55	38.3583	6.72995	2.00098
2020-01-23 13:07:01	38.8405	6.62323	1.16724
2020-01-23 13:07:06	37.6351	6.62323	0.500244
2020-01-23 13:08:49	38.5994	6.50318	1.16724
2020-01-23 13:08:54	38.3583	6.56987	0.500244
2020-01-23 13:08:59	38.1173	6.62323	0
2020-01-23 13:10:45	37.6351	6.62323	0
2020-01-23 13:10:50	38.5994	6.66325	1.16724
2020-01-23 13:10:56	38.5994	6.63657	1.16724
2020-01-23 13:11:01	38.5994	6.62323	1.16724
2020-01-23 13:11:46	38.3583	6.63657	0
2020-01-23 13:13:17	38.5994	6.63657	0
2020-01-23 13:13:22	37.6351	6.63657	1.16724
2020-01-23 13:13:27	38.1173	6.63657	0.500244
2020-01-23 13:13:32	38.3583	6.62323	0.500244

Gambar 4.13 Tampilan *table* kelembaban data pada *database*

2020-01-23 13:03:18	56.9633	0.00666992	2.23442
2020-01-23 13:04:34	56.8019	0.00666992	2.23442
2020-01-23 13:04:39	56.6406	12.2793	2.24776
2020-01-23 13:06:17	56.9633	0.00666992	2.18107
2020-01-23 13:06:55	56.8019	0.00666992	2.23442
2020-01-23 13:07:01	57.9315	0.00666992	2.22108
2020-01-23 13:07:06	57.286	0.00666992	2.23442
2020-01-23 13:08:49	57.9315	0.00666992	2.22108
2020-01-23 13:08:54	57.9315	12.2793	2.23442
2020-01-23 13:08:59	57.286	11.9592	2.23442
2020-01-23 13:10:45	58.7384	0.00666992	2.23442
2020-01-23 13:10:50	57.286	0.00666992	2.1944
2020-01-23 13:10:56	57.286	0.00666992	2.23442
2020-01-23 13:11:01	57.286	0.00666992	2.23442
2020-01-23 13:11:46	56.9633	11.9992	2.23442
2020-01-23 13:13:17	59.3838	0.00666992	2.12771
2020-01-23 13:13:22	56.8019	12.1593	2.22108
2020-01-23 13:13:27	56.9633	11.9592	2.23442
2020-01-23 13:13:32	57.286	0.00666992	2.22108

Gambar 4.14 Tampilan *table* pH data pada *database*

Plan Tanah

Keyword Tanggal &/ waktu

waktu	Kelembaban 1	Kelembaban 2	Kelembaban 3	pH 1
2020-01-23 13:13:32	38.3583	0.500244	57.286	6.62323
2020-01-23 13:13:27	38.1173	0.500244	56.9633	6.63657
2020-01-23 13:13:22	37.6351	1.16724	56.8019	6.63657
2020-01-23 13:13:17	38.5994	0	59.3838	6.63657

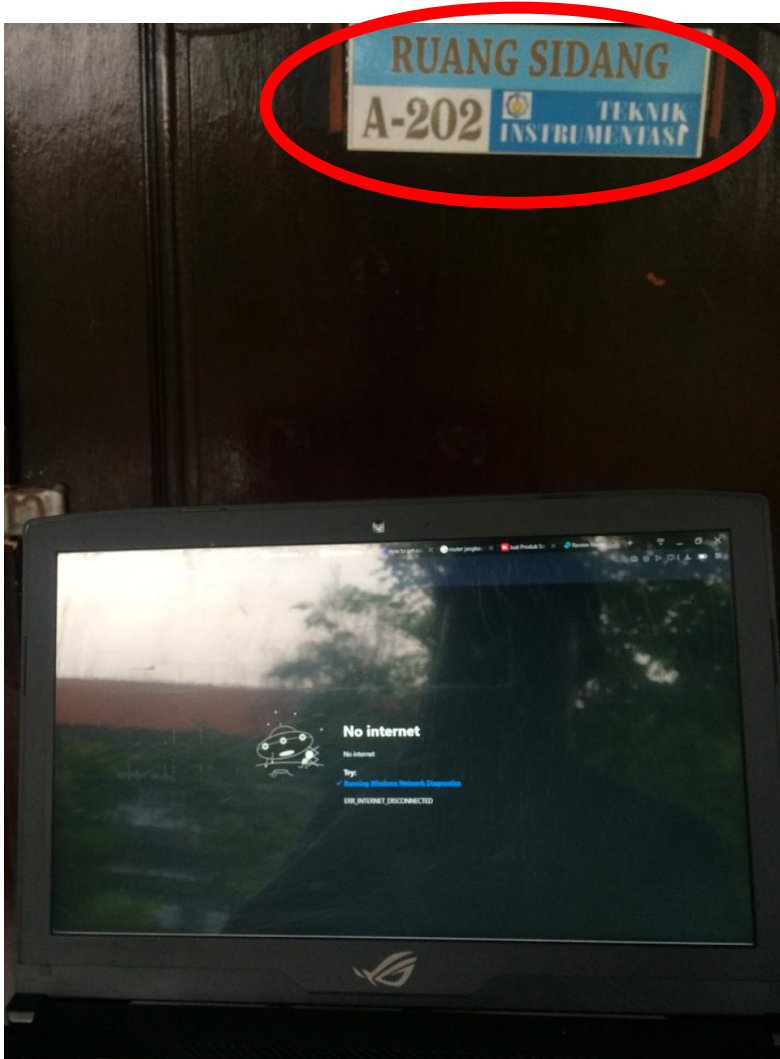
Gambar 4.15 Tampilan *monotoring* pada *web server*

Gambar diatas menunjukkan bahwa hasil pembacaan pada *raspberry*, *database* dan *web server* tidak mengalami masalah seperti kehilangan data atau *delay*. Pengujian selanjutnya adalah

pengujian jarak akses yang akan dilakukan dengan cara berjalan membawa perangkat keras yang digunakan sebagai *user* menjauh dari *plant*. Dengan tujuan mencari jarak terjauh terjadinya *lost connecting*. Berikut adalah gambar hasil pengujian jarak akses dengan cara menjauhkan *user*.



Gambar 4.16 Pengujian sejauh SEKRETARIAT HIMADATA-ITS



Gambar 4.17 Pengujian sejauh Ruang Sidang Teknik Instrumentasi ITS

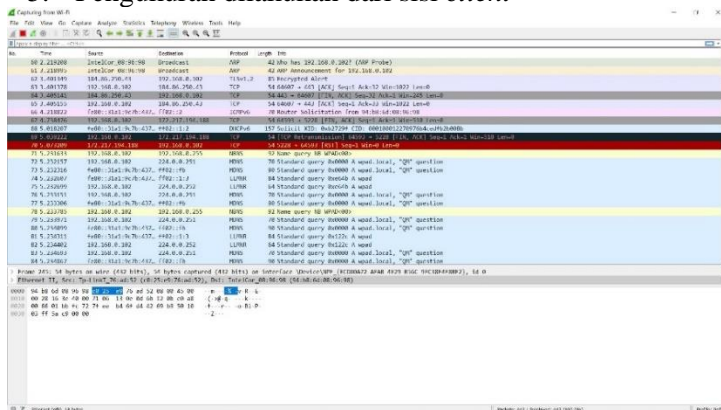
Gambar 4.16 dan Gambar 4.17 menjelaskan bahwa data *web server* tidak dapat diakses dari tempat tersebut. Sekertariat HIMADATA-ITS berada sejauh 24,75 meter. Data tersebut didapat dari perhitungan pengukuran kramik lantai. 27 kramik dengan lebar 40 cm dan 31 kramik dengan lebar 45 cm. pada saat

user berada di ruang Siding Teknik Instrumentasi Industri koneksi terputus dan akses *web server* terputus. Hal ini terjadi akibat akses terhalang oleh tembok dan atap Gedung. Ketinggian akses koneksi adalah kira-kira 10 Meter tanpa halangan tembok. Dari data pengujian tersebut didapatkan sebuah kesimpulan bahwa jenis *web server* tidak berpengaruh terhadap jarak dan batas user pengguna. Jarak dan jumlah total pengguna dipengaruhi oleh router yang dipergunakan.

4.2.5 Uji *Quality of Service* (QoS)

Quality of Service (QoS) adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwith*, mengatasi *jitter* dan *delay*. Parameter QoS adalah *latency*, *jitter*, *packet loss*, *throughput*, *MOS*, *echo cancellation* dan *PDD*. QoS sangat ditentukan oleh kualitas jaringan yang digunakan. Terdapat beberapa factor yang dapat menurunkan nilai QoS, seperti : *Redaman*, *Distorsi*, dan *Noise* (Tedyyana, 2011). Adapun parameter QoS yang digunakan dalam pengukuran meliputi *delay*, *jitter*, *throughput* dan *packet loss*. Adapun metode pengambilan data sampelnya yaitu :

1. Waktu pengambilan data dibatasi kurang dari 1 menit.
2. Perangkat lunak yang digunakan adalah *wireshark*.
3. Pengukuran QoS dilakukan pada parameter *delay*, *jitter*, dan *packet loss*.
4. Pengujian dilakukan pada data yang diakses di *database*
5. Pengukuran dilakukan dari sisi *client*



Gambar 4.18 Tampilan *Wireshark*

Pengujian *QoS* ini diawali dengan uji *delay* pada akses *wifi*. *Delay* adalah waktu tunda saat pengiriman paket yang diakibatkan oleh proses transmisi dari satu titik lain yang menjadi tujuannya. Ada beberapa faktor yang mempengaruhi *delay* yaitu faktor lingkungan dan kesalahan pada sistem operasi. *Delay* diperoleh dari selisih waktu kirim antara satu paket *TCP* dengan paket lainnya. Untuk menghitung rata-rata *delay* dapat menggunakan rumus sebagai berikut :

$$\bar{X}D \text{ (Delay rata-rata)} = \frac{\sum D(\text{jumlah delay})}{\sum A(\text{jumlah data})} \dots\dots\dots 4.1$$

Dari capture data yang telah dilakukan dengan wireshark maka didapatkan rata-rata *delay* dengan cara perhitungan sebagai berikut:

$$\begin{aligned} \bar{X}D &= \frac{1,98621}{80} \\ &= 0,024813 \mu\text{s} \\ &= 0,00024813 \text{ ms} \end{aligned}$$

Tabel 4.2 Tabel kategori *delay* (Tedyyana, 2011)

Kategori Lisensi	Besar Delay
Sangat bagus	<9 ms
Bagus	9 s/d 50 ms
Jelek	50 s/d 450 ms
Sangat jelek	>450 ms

Tabel 4.2 adalah tabel yang menunjukkan kualitas latensi berdasarkan besar delaynya. Sehingga dapat disimpulkan jaringan yang digunakan termasuk dalam kategori latensi yang sangat bagus. Dilanjutkan dengan pengujian *jitter*. *Jitter* didefinisikan sebagai variasi *delay* yang diakibatkan oleh panjang queue dalam suatu pengolahan data dan *reassemble* paket-paket data di akhir pengiriman akibat kegagalan sebelumnya. Menghitung *jitter* dapat menggunakan rumus berikut :

$$Jitter = \frac{\sum E(\text{jumlah variasi delay})}{\sum A(\text{jumlah data in})} \dots\dots\dots 4.2$$

Untuk dapat mengetahui total variasi dapat dihitung dengan rumus sebagai berikut :

$$\text{Total variasi delay} = (\text{delay } 2 - \text{delay } 1) + (\text{delay } 3 - \text{delay } 2) + \dots + (\text{delay } n - \text{delay } (n-1)) \dots \dots \dots 4.3$$

Tabel 4.3 Tabel kategori *jitter* (Tedyana, 2011)

Kategori Degradasi	Besar Jitter
Sangat bagus	0 ms
Bagus	0 s/d 75 ms
Jelek	75 s/d 125 ms
Sangat jelek	>125 ms

Tabel 4.3 adalah tabel yang menunjukkan kategori degradasi berdasarkan besar *jitter*nya. Dari capture data yang telah dilakukan dengan wireshark maka didapatkan nilai *jitter*nya dengan cara perhitungan sebagai berikut:

$$\begin{aligned} \text{Jitter} &= \frac{0,000007}{49} \\ &= 0,00000014 \mu\text{s} \\ &= 0,0000000014 \text{ ms} \end{aligned}$$

Sehingga dapat dikategorikan nilai *jitter* pada jaringan yang digunakan sangat bagus yaitu mendekati nilai 0 ms. Pengujian terakhir pada proses pengujian *Quality of Service (QoS)* adalah pengujian *packet loss*. *Packet loss* adalah jumlah paket data yang hilang per detik. *Packet loss (PL)* dapat disebabkan oleh sejumlah faktor, mencakup penurunan signal dalam media jaringan, melebihi batas saturasi jaringan, paket yang *corrupt* yang menolak untuk transit, dan kesalahan perangkat keras jaringan. Factor lingkungan juga berpengaruh terhadap *packet loss*. Factor lingkungan meliputi cuaca, jarak akses, gangguan perangkat, kondisi ruangan dan lain-lain. *Packet loss* dapat dihitung dengan rumus sebagai berikut :

$$PL = \frac{\sum B(\text{jumlah data out}) - \sum A(\text{jumlah data in})}{\sum A(\text{jumlah data in})} \times 100\% \dots \dots \dots 4.4$$

Tabel 4.4 Tabel kategori *jitter* (Tedyyana, 2011)

Kategori <i>Degradasi</i>	Besar Loss
Sangat bagus	0 %
Bagus	3 %
Jelek	15 %
Sangat jelek	25 %

Tabel 4.4 adalah tabel yang menunjukkan kategori degradasi berdasarkan besar *packet loss*nya. Dari capture data yang telah dilakukan dengan wireshark maka didapatkan nilai *packet loss*nya dengan menggunakan rumus diatas.

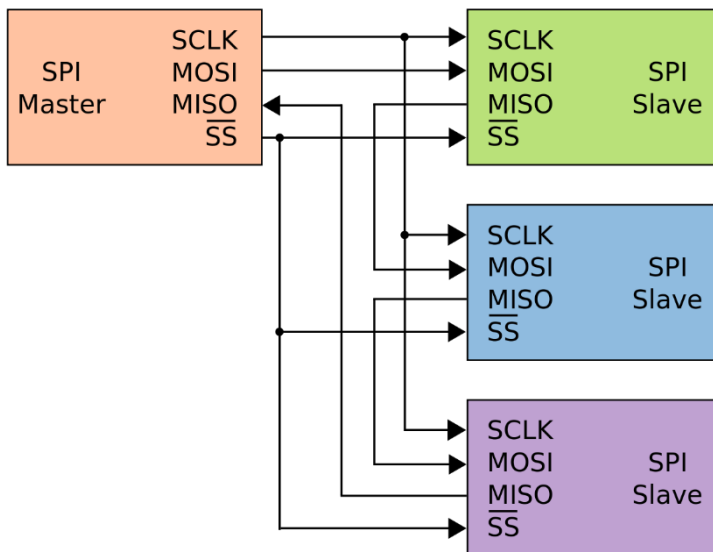
$$PL = \frac{80-80}{80} \times 100\% \\ = 0 \%$$

Sehingga *packet loss* dapat dikategorikan pada jaringan yang digunakan sangat bagus yaitu 0 %. Dari ketiga hasil pengujian dapat dikatakan koneksi *raspberry* dengan *localhost* sangat bagus. Namun, kondisi ini dapat berubah dikarenakan beberapa faktor. Faktor-faktor diantaranya adalah kondisi lingkungan dan sistem operasi yang dijalankan.

4.3 Pembahasan

Pada sistem *monitoring plant* pengendalian pH dan kelembaban tanah hal yang paling pertama dilakukan adalah pemilihan sensor dengan standar yang sesuai dengan kondisi. Kondisi yang dimaksud adalah kondisi dimana sensor dapat bertahan dengan berbagai macam kondisi lingkungan dan memiliki safety yang dapat dikatakan cukup. Dilakukan pemilihan *ADC* yang dapat digunakan sebagai converter sinyal dari analog menuju digital. Tujuannya adalah mengubah sinyal analog menjadi sinyal digital dikarenakan sensor terkadang tidak langsung memiliki *ADC* langsung. *ADC* yang paling sering ditemui saat berhubungan dengan *raspberry* adalah *ADC MCP3008*, *ADC* ini memiliki keunggulan diantaranya memiliki 8 input analog dan *ADC* ini dapat menerima tegangan sinyal analaog sebesar 3.3 volt. *ADC*

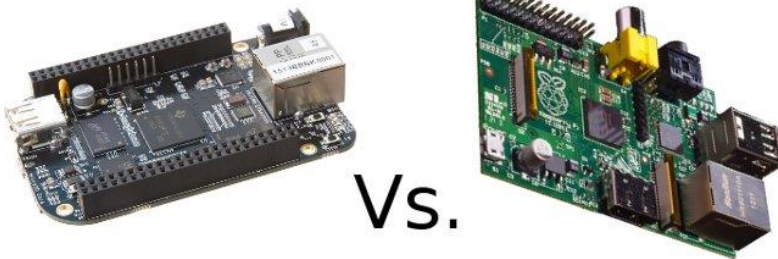
MCP3008 memiliki port *MISO*, *MOSI*, *SCLK* dan *CS*. Dipilihlah *ADC MCP3008* sebagai *ADC* pada plant pengondisian kadar pH dan Kelembaban tanah ini. *ADC* ini juga dapat dihubungkan *ADC* yang sama dengan tujuan menambah jumlah *input analog* pada *plant*. Berikut skema perancangan penambahan *input analog* pada *ADC*.



Gambar 4.19 Skema Penambahan Input Analog

Dari gambar diatas dapat diketahui *raspberry pi* sendiri menjadi *master* dan *ADC MCP3008* menjadi *slave*. *MISO* terakhir dimasukan ke *raspberry pi* sebagai input sensor. *VCC* yang digunakan harus lebih dari 2,5 Volt dan kurang dari 3 volt. *Raspberry* memiliki 40 *pin out* dan memiliki 2 *pin MOSI*, *MISO*, *SCLK* dan *SS* (*C0* dan *C1*). *Raspberry pi* memiliki spesifikasi yang dapat menunjang *plant* dan dapat dikembangkan kembali. Dibandingkan dengan *beaglebone black*, *raspberry* memiliki harga yang terpaut jauh. *Raspberry pi* memiliki harga 600 ribu sampai 700 ribu. Sedangkan *beaglebone black* memiliki harga 1,2 juta sampai 1,4 juta. Walaupun spesifikasi yang diberikan oleh *beaglebone black* sangat bagus tapi pada *plant* ini tidak terlalu

memerlukan spesifikasi tinggi. Berikut adalah perbandingan spesifikasi antara *raspberry pi* dan *beaglebone black*.



BEAGLEBONE BLACK

RASPBERRY PI 3 B+

Gambar 4.20 Perbandingan *Beaglebone Black* dan *Raspberry 3 B+* (Educba, 2019)

Tabel 4.5 Tabel spesifikasi *Beaglebone Black* dan *Raspberry 3* (Educba, 2019)

Spesifikasi	<i>Beaglebone Black</i>	<i>Raspberry 3 B+</i>
<i>Chip</i>	<i>TI AM3359</i>	<i>Broadcom BCM2835 SoC full HD multimedia applications processor</i>
<i>CPU</i>	<i>1 GHz ARM Cortex-A8</i>	<i>700 MHz Low Power ARM1176JZ-F Applications Processor</i>
<i>GPU</i>	<i>PowerVR SGX530</i>	<i>Dual Core VideoCore IV® Multimedia Co-Processor</i>
<i>Memory</i>	<i>512 MB DDR3</i>	<i>512MB SDRAM</i>
<i>Ethernet</i>	<i>onboard 10/100 Ethernet RJ45 jack</i>	<i>onboard 10/100 Ethernet RJ45 jack</i>
<i>USB 2.0</i>	<i>Dual USB Connector</i>	<i>Dual USB Connector</i>
<i>Video Output</i>	<i>16b HDMI, 1280 x 1024 (MAX) 1024 x 768 x 1280 x 720, 1440 x 900, 1920 x 1080@24 Hz w/EDID Support</i>	<i>HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)</i>

<i>Audio Output</i>	<i>Via HDMI Interface, Stereo</i>	<i>3.5mm jack, HDMI</i>
<i>Onboard storage</i>	<i>2 GB 8-bit embedded MMC on-board flash version microSD card 3.3 V Supported</i>	<i>SD, MMC, SDIO card slot</i>
<i>Operating system</i>	<i>Linux, Android, Cloud9 IDE on Node.js w/ BoneScript library, plus more</i>	<i>Linux</i>
<i>Dimensions</i>	<i>86.40 × 53.3 mm (3.402 × 2.10 in)</i>	<i>8.6cm x 5.4cm x 1.7cm</i>
<i>pin</i>	<i>Beaglebone Black has 92 connections as (3 x I2C), CAN bus, SPI bus, (4 Timer), (5 serial port), (65 GPIO pin), (8 x PWM output), (7 x analog input 12 bit)</i>	<i>Raspberry Pi 3 B+ has 40 connections as (8 x GPIO pin, (1 x UART interface), (1 x SPI bus), (1 x I2C bus).</i>

Dari table 4.5 dapat diketahui bahwa spesifikasi dari *beaglebone black* lebih tinggi dibandingkan *raspberry*. Mulai dari spesifikasi RAM, CPU, GPU, dan lain-lain. Penggunaan *database MySQL* diperuntukan sebagai penyimpanan data agar dapat diakses oleh *web server*. *Web server* yang digunakan adalah *Apache2*. *Apache2* digunakan dikarenakan dapat menggunakan Bahasa *PHP* sebagai *interface*. Dari hasil pengujian didapatkan nilai delay sebesar 0,00024813 ms dan jarak akses terjauh adalah kurang dari 24,75 Meter. Tidak terdapat loss packet pada pengiriman data dan dibuktikan pada Gambar 4.12 sampai Gambar 4.15. Pada *LCD 20x4* tidak terdapat lost packet tetapi terdapat *error* dikarenakan kekurangan *VCC* dan dapat diselesaikan dengan cara penggantian *VCC*.

Halaman ini sengaja dikosongkan

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan, maka didapatkan beberapa kesimpulan sebagai berikut:

- a. Proses *monitoring* berbasis *web server* pada pengendalian pH dan kelembaban tanah dapat dilakukan secara real time dengan *Delay* sebesar 0,00024813ms. sedangkan *Delay* pada *LCD 20x4* sebesar 3ms.
- b. *LCD* dapat menampilkan data pembacaan sensor tanpa kehilangan data. Namun saat Motor Servo dan *Relay* secara bersamaan mendapat *supply* tegangan, terjadi *error* pada *LCD* akibat menurunnya *supply* tegangan. Namun saat *VCC* diganti, *LCD* tidak mengalami *error* pada kondisi yang sama.
- c. Hasil *monitoring* dapat ditampilkan pada *webserver* dengan tampilan *user* yang nyaman dan jelas. Serta terdapat *search tool* untuk mencari data. Hasil *monitoring* pada plan pengendalian pH dan kelembaban tanah ini berupa data yang diolah oleh *raspberry* dan disimpan di *database MySQL*.

5.2 Saran

Adapun saran yang dapat diambil selama pengerjaan tugas akhir adalah sebagai berikut:

- a. Menggunakan peralatan keselamatan pada saat terjadi *troubleshooting* panel.
- b. Menggunakan *server* Publik untuk memperluas jarak akses.
- c. Melakukan pengujian dengan menggunakan beberapa contoh tanah dari beberapa daerah
- d. Melakukan kalibrasi dengan alat yang memiliki standar kalibraator.
- e. Memperhatikan pemasangan *ground* pada plant. Karena *grounding* sangat penting untuk *safety*.
- f. Menggunakan *MCB* sebagai *safety*.
- g. Menambahkan *power supply external* pada *panel box*.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- B.L. Theraja. 2005. *Electrical Technology*. New Delhi: Nirja Contruction & Development Co. (P) Ltd.
- Christian, F. (2017). Modul pembelajaran raspberry pi. yogyakarta: Universitas Sanata Dharma. Diakses dari https://repository.usd.ac.id/12017/2/135114037_full.pdf
- Damayanti. (2017). Rancang Bangun Web Server Sebagai Data Storage And Display Dari Raspberry Pi. palembang: Politeknik negri sriwijaya. Diakses dari <http://eprints.polsri.ac.id/4548/>
- Dfrobot, (2019). All Sensor Product. Diakses dari <https://www.dfrobot.com/category-36.html>
- Educba, (2019). Difference Between Raspberry Pi 3 b+ versus BeagleBone Black. Diakses dari <https://www.educba.com/raspberry-pi-3-vs-beaglebone-black/>
- Fadli, R., Rifqi, A., & Fransisco, J (2016). Network Traffic Management , Quality of Service (Qos), Congestion Control dan Frame Relay. Depok: Universitas Gunadharma. Diakses dari https://agry_alfiah.staff.gunadarma.ac.id
- Fernando, Erick dkk. (2017). Arsitektur Teknologi Webserver Berbasis Mini PC Dengan Raspberry Pi. *Jurnal Sistem Informasi Indonesia (JSII)*. 2 . 135–138. Diakses dari http://aisindo.org/wp-content/uploads/2019/10/JSII-AISINDO-vol-2-no-1-September-2017_FIX.pdf
- Fridayanthie, E. W., & Mahdiati, T. (2016). Rancang Bangun Sistem Informasi Permintaan Atk Berbasis Intranet (Studi Kasus: Kejaksaan Negeri Rangkasbitung). *Revista Brasileira de Ergonomia*. 9(2), 10. Diakses dari <https://doi.org/10.5151/cidi2017-060>
- Hidayat, D., Rahmatika, M., Syafei, N. S., & Tumbelaka, B. Y. (2018). Simulasi Pengontrol On/Off pada Sistem Kendali Umpan Balik dengan Model Fisis Elektronik. *TELKA - Telekomunikasi, Elektronika, Komputasi Dan Kontrol*, 4(1), 43–53. Diakses dari <https://doi.org/10.15575/telka.v4i1.80>

- Javamas Agro, (2019). Kelembaban Tanah Ideal. Diakses dari <https://www.javamas.com/kelembaban-tanah-ideal/>
- Luthfy, P. (2015). Rancang Bangun Sistem Otomasi Hidroponik NFT. e-Proceeding of Applied Science. 1. 75-78. Diakses dari <http://openlibrary.telkomuniversity.ac.id>
- Maulana, Y. Y., Mahmudin, D., Wijaya, R. I., & Wiranto, G. (2016). Monitoring Kualitas Air Secara Real-Time Terintegrasi. Jurnal Elektronika Dan Telekomunikasi, 15(1), 23. diakses dari <https://doi.org/10.14203/jet.v15.23-27>
- Merisa, A. (2019). Rancang Bangun Sistem Kontrol dan Monitoring Iklim Mikro Pada Miniatur Greenhouse Berbasis Radio NRF24101 dan Mikrokontroler Arduino Nano. Yogyakarta: Universitas Negeri Yogyakarta, 1-3. diakses dari http://digilib.uin-suka.ac.id/35486/1/13620022_BAB-I_V_DAFTAR-PUSTAKA.pdf
- Sanjana, P., and Mahalakshmi (2014). Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor. (IJCSIT) International Journal of Computer Science and Information Technologies. 5(6), 7107-7109. Diakses dari <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.658.6805&rep=rep1&type=pdf>
- SUBA, R.S.K. (2014). Smart Monitoring System Using Raspberry-Pi and Smartphone. Elkomika. 7(1). 72-84. Diases dari <https://ejurnal.itenas.ac.id/index.php/elkomika/article/view/2317/1922>
- Tedyyana, A. (2016). Membuat Web Server Menggunakan Dinamic Domain Name System Pada Ip Dinamis. Teknologi Informasi & Komunikasi Digital Zone. 7. 1-10 . Diakses dari <https://media.neliti.com/media/publications/53833-ID-membuat-web-server-menggunakan-dinamic-d.pdf>
- Triwiyatno, A. (2011). Buku Ajar Sistem Kontrol Analog, 2Diakses dari <https://media.neliti.com/media/publications/53833-ID-membuat-web-server-menggunakan-dinamic-d.pdf>

LAMPIRAN A SPESIFIKASI KOMPONEN PLANT

1. RASPBERRY PI 3 B+



- SoC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
- Storage: Micro-SD
- GPIO: 40-pin GPIO header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- Dimensions: 82mm x 56mm x 19.5mm, 50g

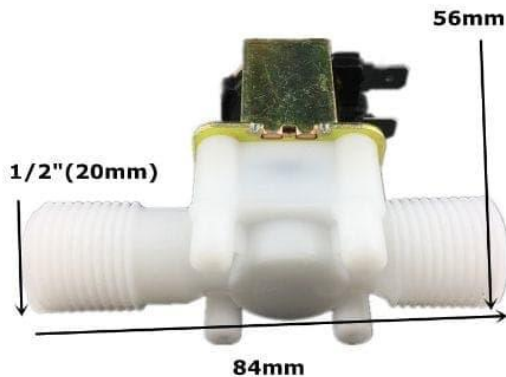
2. MOTOR SERVO



Motor servo MG995 adalah salah satu tipe motor servo dimana motor servo ini termasuk motor servo standard (servo rotation 108 °) adalah jenis yang paling umum dari motor servo, dimana putaran poros outputnya terbatas hanya 90 ° kearah kanan dan 90 ° kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180 °

- Operating Voltage : 4.8~7.2V
- Operating Current : 100mA
- Operating Speed : 0.17sec/60degree (4.8V)~0.13sec/60degree (6.0V)
- Torque : 13.0kg/cm (4.8V)~15.0kg/cm (6.0V)
- Dead Band Width : 4usec
- Temperature Range : -30~+60°C
- Cable Length : 30cm
- Servo Type : Analog Servo

3. SELENOID VALVE



Solenoid valve plastik seperti kran listrik untuk memutuskan listrik /menutup katup air. Posisi normal kran/valve tertutup, jika diberi tegangan AC 220V maka kran/valve terbuka dan air mengalir.

Specification:

- Operation mode: normally closed
- Voltage: AC 220V
- Power: 8W
- Current: 0.6A
- Material: metal + plastic
- Screw diameter: approx. 20mm
- Size: approx. 84 x 57mm (L x H)
- Coil size: approx. 34 x 23mm (W x H)
- Color: as picture shows
- Inlet and outlet: hose barbs for 1/2" (outer diameter) hose
- Pressure: 0.02- 0.8Mpa
- Max fluid temperature: 100C
- Valve type: diaphragm (operated by Servo)
- Usage: water and low viscosity fluids
- Model: ZE-4F90

4. SPESIFIKASI SENSOR KELEMBABAN TANAH



Specification

- Operating Voltage: 3.3 ~ 5.5 VDC
- Output Voltage: 0 ~ 3.0VDC
- Operating Current: 5mA
- Interface: PH2.0-3P
- Dimensions: 3.86 x 0.905 inches (L x W)
- Weight: 15g

5. SPESIFIKASI SENSOR PH TANAH



Specification

- Module Power : 5.00V
- Module Size : 43mm×32mm
- Measuring Range:0-14PH
- Measuring Temperature :0-60 °C
- Accuracy : $\pm 0.1\text{pH}$ (25 °C)
- Response Time : $\leq 1\text{min}$
- pH Sensor with BNC Connector
- PH2.0 Interface (3 foot patch)
- Gain Adjustment Potentiometer
- Power Indicator LED
- Cable Length from sensor to BNC connector:660mm

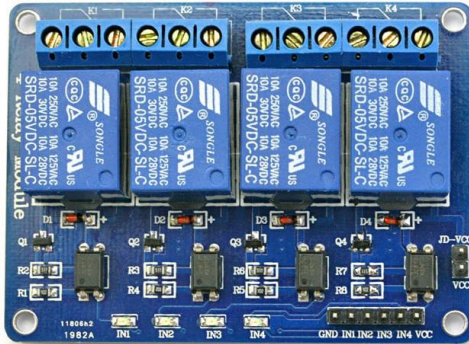
6. POMPA AT-106



Spesifikasi

- Model : AT-105
- Power : 500 watt
- Tinggi Maksimum : 3.0 meter
- Daya maksimum : 3000 liter/jam
- Voltase : 220-240 volt

7. RELAY



Spesifikasi:

- Input relay 5V DC
- Maksimum load 250VAC/10A 30VDC/10A
- Dilengkapi dengan optocoupler isolation untuk melindungi board microcontroller dari tegangan AC
- Memiliki LED indikator
- Menggunakan terminal block sehingga pemasangan kabel menjadi mudah
- Output keluaran 4 channel maksimal 10A

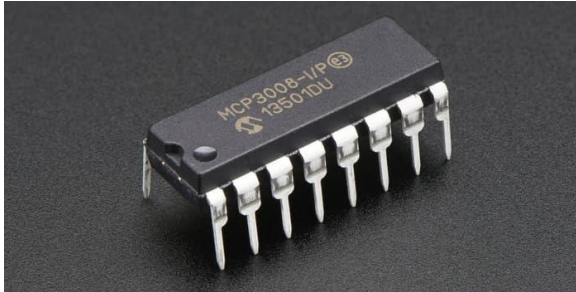
Output memiliki 3 pin terminal block yang ditandai dengan NO, COM dan NC.

NO (Normally Open) = Tidak ada arus yang dialirkan (OFF), Jika ada signal High / Low dari microcontroller maka ON

COM (Common) = Sumber tegangan yang akan dihubungkan (Bisa arus AC maupun DC 10A max)

NC (Normally Close) = Arus dialirkan (ON), Jika ada signal High / Low dari microcontroller maka OFF

8. ADC MCP 3008



Features

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- 4 (MCP3004) or 8 (MCP3008) input channels
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200 ksps max. sampling rate at $V_{DD} = 5V$
- 75 ksps max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
- 5 nA typical standby current, 2 μA max.
- 500 μA max. active current at 5V
- Industrial temp range: $-40^{\circ}C$ to $+85^{\circ}C$
- Available in PDIP, SOIC and TSSOP packages

Applications

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

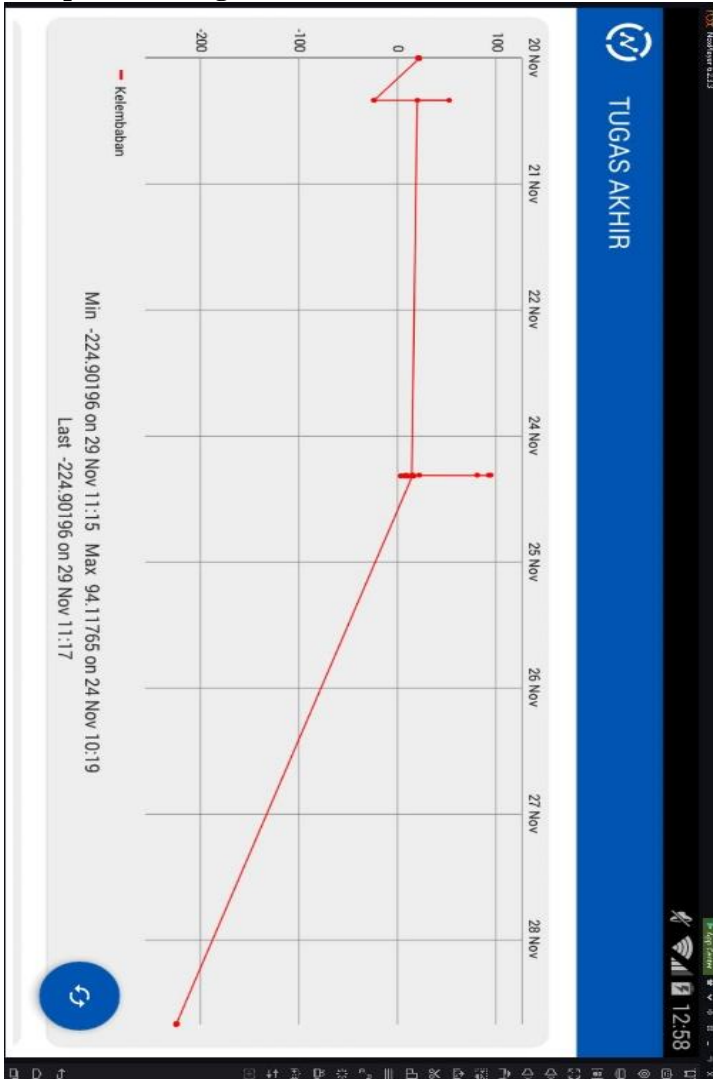
9. POWER SUPPLY 5 V DC



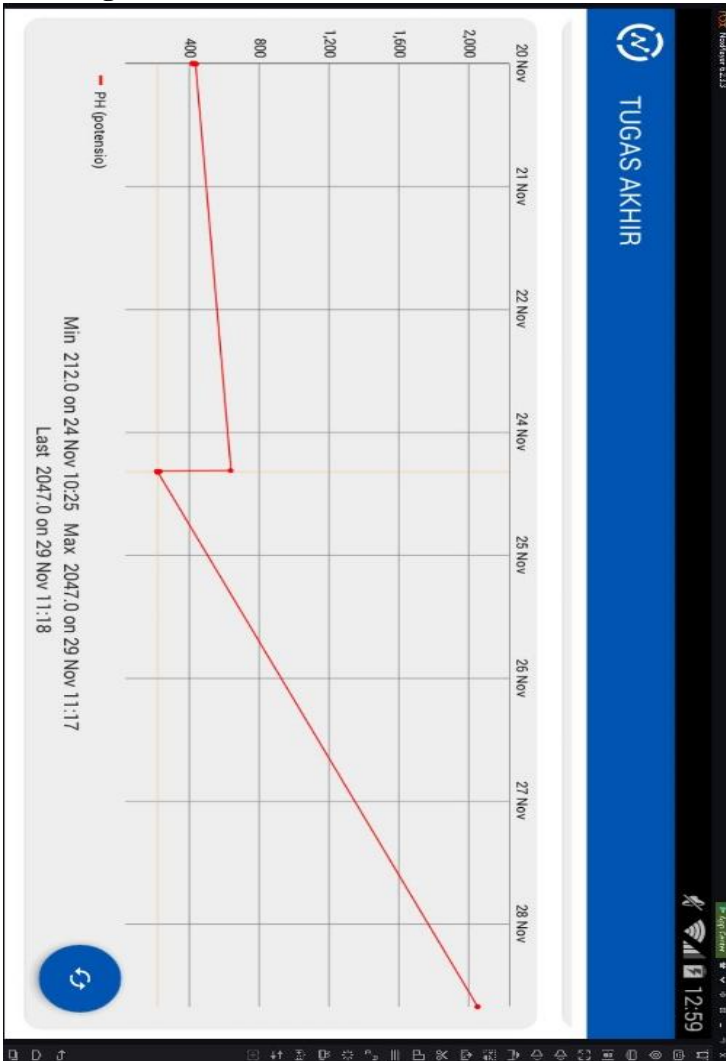
- Merubah tegangan AC 220 Volt menjadi DC 5Volt
- Medium Quality - 90% Real Rated Capacity ~ 40A
- Digunakan untuk bermacam-macam perangkat elektronik yg menggunakan tegangan 5 Volt
- Kualitas Power Supply Baik, menggunakan komponen berkualitas
- Menggunakan casing aluminium untuk pembuangan panas yang optimal
- Sangat disarankan untuk Module Running Text dan Videotron
- Grade B Power Supply

LAMPIRAN B PERCOBAAN THING VIEW

1. Percobaan Monitoring Data kelembaban Menggunakan Aplikasi Thing View



2. Percobaan Monitoring Data pH Menggunakan Aplikasi Thing View



3. Table Database Kelembaban Yang Telah Terdownload

12/26/2019

169.254.32.74 / localhost / monitoring / t

Current selection does not contain a unique column. Grid edit, checkbox, Ed

Showing rows 500 - 620 (621 total, Query took 0.0104 seconds.)

```
SELECT * FROM `tugas_akhir`
```

waktu	sensor_klmb_1	sensor_klmb_2	sensor_klmb_3
2019-12-23 15:18:42	45.3497	2.42054	38.0831
2019-12-23 15:18:47	43.18	0.484107	36.6308
2019-12-23 15:18:53	42.2156	1.12958	36.1467
2019-12-23 15:18:59	37.6351	0.484107	33.4034
2019-12-23 15:19:04	28.9562	0.161369	38.5672
2019-12-23 15:19:09	43.18	0.161369	36.1467
2019-12-23 15:19:15	42.4567	1.12958	36.1467
2019-12-23 15:19:20	42.2156	0.484107	36.1467
2019-12-23 15:19:25	41.4924	0.484107	35.5012
2019-12-23 15:19:31	42.2156	0.484107	36.1467
2019-12-23 15:19:36	42.2156	0.161369	36.3081
2019-12-23 15:19:41	43.18	1.12958	35.9853
2019-12-23 15:19:47	42.2156	0.484107	36.1467
2019-12-23 15:19:52	44.1443	0.484107	37.2763
2019-12-23 15:19:57	42.2156	0.484107	36.1467
2019-12-23 15:22:11	42.2156	0.161369	36.1467
2019-12-23 15:22:17	42.2156	0.484107	38.0831
2019-12-23 15:22:22	42.2156	0.484107	36.1467
2019-12-23 15:22:27	42.2156	0.484107	35.9853
2019-12-23 15:22:33	42.2156	0.484107	38.0831
2019-12-23 15:22:38	41.4924	0.484107	34.8557
2019-12-23 15:22:43	34.9832	2.42054	38.0831
2019-12-23 15:22:49	46.0729	1.12958	38.7286

4. Table Database pH Yang Telah Terdownload

alhost / monitoring / tugas_akhir | phpMyAdmin 4.6.6deb5

id edit, checkbox, Edit, Copy and Delete features are not available.

seconds.)

sensor_klmb_3	sensor_ph_1	sensor_ph_2	sensor_ph_3
38.0831	6.50318	3.24825	4.0353
36.6308	6.50318	13.6533	4.04864
36.1467	6.50318	13.6533	4.04864
33.4034	6.50318	13.6533	4.06198
38.5672	6.55653	13.6533	4.06198
36.1467	6.50318	13.6533	4.06198
36.1467	6.50318	13.6533	4.07532
36.1467	6.50318	13.6533	4.04864
35.5012	6.56987	13.6533	4.06198
36.1467	6.50318	13.6533	4.04864
36.3081	6.50318	13.6533	4.04864
35.9853	6.50318	13.6533	4.04864
36.1467	6.50318	13.6533	4.0353
37.2763	6.50318	13.6533	4.04864
36.1467	6.50318	13.6533	4.04864
36.1467	6.51652	13.6533	4.04864
38.0831	6.50318	9.91818	3.9686
36.1467	6.50318	13.6533	4.04864
35.9853	6.50318	13.6533	4.06198
38.0831	6.52986	8.94437	4.06198
34.8557	6.50318	13.6533	4.04864
38.0831	6.48984	2.23442	4.04864
38.7286	6.50318	3.18155	4.0353

LAMPIRAN C

PROGRAM PLANT

1. Program Library Untuk I2C

```
# -*- coding: utf-8 -*-
# Original code found at:
#
https://gist.github.com/DenisFromHR/cc863375a6e1
9dce359d
"""
Compiled, mashed and generally mutilated 2014-
2015 by Denis Pleic
Made available under GNU GENERAL PUBLIC LICENSE
# Modified Python I2C library for Raspberry Pi
# as found on
http://www.recantha.co.uk/blog/?p=4849
# Joined existing 'i2c_lib.py' and 'lccdriver.py'
into a single library
# added bits and pieces from various sources
# By DenisFromHR (Denis Pleic)
# 2015-02-10, ver 0.1
"""
# i2c bus (0 -- original Pi, 1 -- Rev 2 Pi)
I2CBUS = 1
# LCD Address
ADDRESS = 0x27
import smbus
from time import sleep
class i2c_device:
    def __init__(self, addr, port=I2CBUS):
        self.addr = addr
        self.bus = smbus.SMBus(port)
# Write a single command
    def write_cmd(self, cmd):
        self.bus.write_byte(self.addr, cmd)
        sleep(0.0001)
# Write a command and argument
    def write_cmd_arg(self, cmd, data):
        self.bus.write_byte_data(self.addr, cmd,
data)
```

```

        sleep(0.0001)
# Write a block of data
    def write_block_data(self, cmd, data):
        self.bus.write_block_data(self.addr, cmd,
data)
        sleep(0.0001)
# Read a single byte
    def read(self):
        return self.bus.read_byte(self.addr)
# Read
    def read_data(self, cmd):
        return self.bus.read_byte_data(self.addr,
cmd)
# Read a block of data
    def read_block_data(self, cmd):
        return self.bus.read_block_data(self.addr,
cmd)
# commands
LCD_CLEARDISPLAY = 0x01
LCD_RETURNHOME = 0x02
LCD_ENTRYMODESET = 0x04
LCD_DISPLAYCONTROL = 0x08
LCD_CURSORSHIFT = 0x10
LCD_FUNCTIONSET = 0x20
LCD_SETCGRAMADDR = 0x40
LCD_SETDDRAMADDR = 0x80
# flags for display entry mode
LCD_ENTRYRIGHT = 0x00
LCD_ENTRYLEFT = 0x02
LCD_ENTRYSHIFTINCREMENT = 0x01
LCD_ENTRYSHIFTDECREMENT = 0x00
# flags for display on/off control
LCD_DISPLAYON = 0x04
LCD_DISPLAYOFF = 0x00
LCD_CURSORON = 0x02
LCD_CURSOROFF = 0x00
LCD_BLINKON = 0x01
LCD_BLINKOFF = 0x00
# flags for display/cursor shift
LCD_DISPLAYMOVE = 0x08
LCD_CURSORMOVE = 0x00
LCD_MOVERIGHT = 0x04

```

```

LCD_MOVELEFT = 0x00
# flags for function set
LCD_8BITMODE = 0x10
LCD_4BITMODE = 0x00
LCD_2LINE = 0x08
LCD_1LINE = 0x00
LCD_5x10DOTS = 0x04
LCD_5x8DOTS = 0x00
# flags for backlight control
LCD_BACKLIGHT = 0x08
LCD_NOBACKLIGHT = 0x00
En = 0b00000100 # Enable bit
Rw = 0b00000010 # Read/Write bit
Rs = 0b00000001 # Register select bit
class lcd:
    #initializes objects and lcd
    def __init__(self):
        self.lcd_device = i2c_device(ADDRESS)
        self.lcd_write(0x03)
        self.lcd_write(0x03)
        self.lcd_write(0x03)
        self.lcd_write(0x02)
        self.lcd_write(LCD_FUNCTIONSET | LCD_2LINE
| LCD_5x8DOTS | LCD_4BITMODE)
        self.lcd_write(LCD_DISPLAYCONTROL
LCD_DISPLAYON)
        self.lcd_write(LCD_CLEARDISPLAY)
        self.lcd_write(LCD_ENTRYMODESET
LCD_ENTRYLEFT)
        sleep(0.2)
        # clocks EN to latch command
        def lcd_strobe(self, data):
            self.lcd_device.write_cmd(data | En |
LCD_BACKLIGHT)
            sleep(.0005)
            self.lcd_device.write_cmd(((data & ~En) |
LCD_BACKLIGHT))
            sleep(.0001)
        def lcd_write_four_bits(self, data):
            self.lcd_device.write_cmd(data |
LCD_BACKLIGHT)
            self.lcd_strobe(data)

```

```

# write a command to lcd
def lcd_write(self, cmd, mode=0):
    self.lcd_write_four_bits(mode | (cmd &
0xF0))
    self.lcd_write_four_bits(mode | ((cmd << 4)
& 0xF0))
# write a character to lcd (or character rom)
0x09: backlight | RS=DR<
# works!
def lcd_write_char(self, charvalue, mode=1):
    self.lcd_write_four_bits(mode | (charvalue
& 0xF0))
    self.lcd_write_four_bits(mode | ((charvalue
<< 4) & 0xF0))

# put string function with optional char
positioning
def lcd_display_string(self, string, line=1,
pos=0):
    if line == 1:
        pos_new = pos
    elif line == 2:
        pos_new = 0x40 + pos
    elif line == 3:
        pos_new = 0x14 + pos
    elif line == 4:
        pos_new = 0x54 + pos
    self.lcd_write(0x80 + pos_new)
    for char in string:
        self.lcd_write(ord(char), Rs)
# clear lcd and set to home
def lcd_clear(self):
    self.lcd_write(LCD_CLEARDISPLAY)
    self.lcd_write(LCD_RETURNHOME)
# define backlight on/off (lcd.backlight(1);
off= lcd.backlight(0)
def backlight(self, state): # for state, 1 =
on, 0 = off
    if state == 1:

self.lcd_device.write_cmd(LCD_BACKLIGHT)
    elif state == 0:

```

```

self.lcd_device.write_cmd(LCD_NOBACKLIGHT)
# add custom characters (0 - 7)
def lcd_load_custom_chars(self, fontdata):
    self.lcd_write(0x40);
    for char in fontdata:
        for line in char:
            self.lcd_write_char(line)

```

2. Program Tugas Akhir

```

from gpiozero import MCP3008
import I2C_LCD_driver
from time import *
import time
mylcd = I2C_LCD_driver.lcd()
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(29, GPIO.OUT)
GPIO.output(29, True)
PWM1 = GPIO.PWM(29, 50)
GPIO.setup(32, GPIO.OUT)
GPIO.output(32, True)
PWM2 = GPIO.PWM(32, 50)
GPIO.setup(38, GPIO.OUT)
GPIO.output(38, True)
PWM3 = GPIO.PWM(38, 50)
GPIO.setup(35, GPIO.OUT)
GPIO.output(35, True)
GPIO.setup(36, GPIO.OUT)
GPIO.output(36, True)
GPIO.setup(37, GPIO.OUT)
GPIO.output(37, True)
PWM1.start(0)
PWM2.start(0)
PWM3.start(0)
a = MCP3008(0)
b = MCP3008(1)
c = MCP3008(2)
d = MCP3008(3)
e = MCP3008(4)
f = MCP3008(5)

```

```

# importing connector
import mysql.connector as connector
from mysql.connector import Error
"""
    module 1:
    use to initiate any connection
    initiate_connection(par1, par2, par3, par4)
    par1: host name (local for local server)
    par2: database name
    par3: user name
    par4: password for corresponding user
    returning database connection
"""
def    initiate_connection(inHost,    inDatabase,
inUser, inPassword):
    connectorInstance = None
    try:
        connectorInstance =
connector.connect(host=inHost,port=3306,database
=inDatabase,user=inUser,password=inPassword)
    except Error as e:
        print('Connection Failed with error')
        print(e)
    finally:
        if connectorInstance is not None and
connectorInstance.is_connected():
            print('Connection Succesfull')
            return connectorInstance
        else:
            print('Connection Failed')
"""
Module 2:
Fetching all data into variable
FetchData(par1, par2, par3)
par1: connection instance
par2: table name in current database instance
par3: column name (* for selecting all column)
returning all data as array
"""
def FetchData(instance,tableName, colName):
    cursor = instance.cursor()
    text = 'SELECT '+colName+' FROM '+tableName

```



```

try:
    cursor.execute(text)
    rows = cursor.fetchall()
except Error as e:
    print(e)
finally:
    return rows
"""
Module 3:
Inserting data to database.
Database has 2 columns, strict to 2 values
inputted

    InputData(par1,par2,par3,par4)
    par1: database instance
    par2: table name
    par3: inserted data
    par4: inserted data
"""
def
InputData(instance,col1,col2,col3,col4,col5,col6
):
    cursor = instance.cursor()
    """SQL"""
    queryTarget = 'INSERT INTO
tugas_akhir(sensor_klmb_1,sensor_klmb_2,sensor_k
lmb_3,sensor_ph_1,sensor_ph_2,sensor_ph_3)
VALUES
('+str(col1)+','+str(col2)+','+str(col3)+','+str
(col4)+','+str(col5)+','+str(col6)+')'
    try:
        cursor.execute(queryTarget)
        instance.commit()
    except Error as e:
        print(e)
    finally:
        print(cursor.lastrowid)
"""
Module 4:
closing connection
close(par1)
par1: database instance

```

```

"""
def close(instance):
    instance.close()
    print('Database Connection Closed')
try:
    v
    initiate_connection("localhost","monitoring","ro
ot","123456")
    while True :
        bab1 = a.value * 1024
        bab2 = b.value * 1024
        bab3 = c.value * 1024
        bab4 = d.value * 1024
        bab5 = e.value * 1024
        bab6 = f.value * 1024
        kelembaban1 = (-((bab1-800)/415)*100)
        kelembaban2 = (-((bab2-1024)/620)*100)
        kelembaban3 = (-((bab3-1024)/620)*100)
        if kelembaban1 > 100: kelembaban1=100;
        elif kelembaban1 < 0: kelembaban1=0;

        if kelembaban2 > 100: kelembaban2=100;
        elif kelembaban2 < 0: kelembaban2=0;

        if kelembaban3 > 100: kelembaban3=100;
        elif kelembaban3 < 0: kelembaban3=0;

        ph1 = (bab4/75)
        ph2 = (bab5/74)
        ph3 = (bab6/75)

    InputData(v, kelembaban1, kelembaban2, kelembaban3,
    ph1, ph2, ph3)
    print('Nilai Kelembaban 1 = %3.2f' %bab1)
    print('Nilai Kelembaban 2 = %3.2f' %bab2)
    print('Nilai Kelembaban 3 = %3.2f' %bab3)
    print('Nilai PH = %3.2f' %bab4)
    print('Nilai PH = %3.2f' %bab5)
    print('Nilai PH = %3.2f' %bab6)
    mylcd.lcd_display_string((str('PLANT')),
1, 0)

```

```

        mylcd.lcd_display_string((str('HUM      |
PH')), 1, 8)
        mylcd.lcd_display_string((str('BLOK1    =
%3.1f' %kelembaban1)), 2, 0)
        mylcd.lcd_display_string((str('|%3.1f'
%ph1)), 2, 14)
        mylcd.lcd_display_string((str('BLOK2    =
%3.1f' %kelembaban2)), 3, 0)
        mylcd.lcd_display_string((str('|%3.1f'
%ph2)), 3, 14)
        mylcd.lcd_display_string((str('BLOK3    =
%3.1f' %kelembaban3)), 4, 0)
        mylcd.lcd_display_string((str('|%3.1f'
%ph3)), 4, 14)
        blok1    =    (ph1>7)    or    (ph1<6)    or
(kelembaban1<40)
        blok2    =    (ph2>7)    or    (ph2<6)    or
(kelembaban2<40)
        blok3    =    (ph3>7)    or    (ph3<6)    or
(kelembaban3<40)
        if blok1:
            GPIO.setmode(GPIO.BOARD)
            GPIO.setup(35, GPIO.OUT)
            GPIO.setup(36, GPIO.OUT)
            GPIO.setup(37, GPIO.OUT)
            GPIO.output(35,GPIO.HIGH)
            GPIO.output(36,GPIO.LOW)
            GPIO.output(37,GPIO.LOW)
            PWM1.ChangeDutyCycle(7-
((kelembaban1/60)*5))
            PWM2.ChangeDutyCycle(7+(((ph1-
7)/7)*4.5))
            PWM3.ChangeDutyCycle(7-(((ph1-
7)/7)*5))
        elif blok2:
            GPIO.setmode(GPIO.BOARD)
            GPIO.setup(35, GPIO.OUT)
            GPIO.setup(36, GPIO.OUT)
            GPIO.setup(37, GPIO.OUT)
            GPIO.output(36,GPIO.HIGH)
            GPIO.output(35,GPIO.LOW)
            GPIO.output(37,GPIO.LOW)

```

```

        PWM1.ChangeDutyCycle (7-
((kelembaban2/60)*5))
        PWM2.ChangeDutyCycle (7+(((ph2-
7)/7)*4.5))
        PWM3.ChangeDutyCycle (7-(((ph2-
7)/7)*5))
    elif blok3:
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(35, GPIO.OUT)
        GPIO.setup(36, GPIO.OUT)
        GPIO.setup(37, GPIO.OUT)
        GPIO.output(37,GPIO.HIGH)
        GPIO.output(36,GPIO.LOW)
        GPIO.output(35,GPIO.LOW)
        PWM1.ChangeDutyCycle (7-
((kelembaban3/60)*5))
        PWM2.ChangeDutyCycle (7+(((ph3-
7)/7)*5))
        PWM3.ChangeDutyCycle (7-(((ph3-
7)/7)*5))
    elif blok1 and blok2:
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(35, GPIO.OUT)
        GPIO.setup(36, GPIO.OUT)
        GPIO.setup(37, GPIO.OUT)
        GPIO.output(35,GPIO.HIGH)
        GPIO.output(36,GPIO.HIGH)
        GPIO.output(37,GPIO.LOW)
        PWM1.ChangeDutyCycle (7-
((kelembaban2/60)*5))
        PWM2.ChangeDutyCycle (7+(((ph2-
7)/7)*4.5))
        PWM3.ChangeDutyCycle (7-(((ph2-
7)/7)*5))
    elif blok1 and blok3:
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(35, GPIO.OUT)
        GPIO.setup(36, GPIO.OUT)
        GPIO.setup(37, GPIO.OUT)
        GPIO.output(35,GPIO.HIGH)
        GPIO.output(37,GPIO.HIGH)
        GPIO.output(36,GPIO.LOW)

```

```

        PWM1.ChangeDutyCycle(7-
((kelembaban1/60)*5))
        PWM2.ChangeDutyCycle(7+(((ph1-
7)/7)*4.5))
        PWM3.ChangeDutyCycle(7-(((ph1-
7)/6)*5))
    elif blok2 and blok3:
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(35, GPIO.OUT)
        GPIO.setup(36, GPIO.OUT)
        GPIO.setup(37, GPIO.OUT)
        GPIO.output(36,GPIO.HIGH)
        GPIO.output(37,GPIO.HIGH)
        GPIO.output(35,GPIO.LOW)
        PWM1.ChangeDutyCycle(7-
((kelembaban2/60)*5))
        PWM2.ChangeDutyCycle(7+(((ph2-
7)/7)*4.5))
        PWM3.ChangeDutyCycle(7-(((ph2-
7)/7)*5))
    elif blok1 and blok2 and blok3:
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(35, GPIO.OUT)
        GPIO.setup(36, GPIO.OUT)
        GPIO.setup(37, GPIO.OUT)
        GPIO.output(35,GPIO.HIGH)
        GPIO.output(36,GPIO.HIGH)
        GPIO.output(37,GPIO.HIGH)
        PWM1.ChangeDutyCycle(7-
((kelembaban2/60)*5))
        PWM2.ChangeDutyCycle(7+(((ph2-
7)/7)*4.5))
        PWM3.ChangeDutyCycle(7-(((ph2-
7)/6)*5))
        time.sleep(5)
        mylcd.lcd_clear()
except KeyboardInterrupt:
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(35, GPIO.OUT)
    GPIO.setup(36, GPIO.OUT)
    GPIO.setup(37, GPIO.OUT)
    GPIO.output(35,GPIO.LOW)

```

```

GPIO.output (36,GPIO.LOW)
GPIO.output (37,GPIO.LOW)
GPIO.cleanup ()
exit()

```

3. Program Interface Web Server Dengan Bahasa PHP

```

<?php
$conn = mysqli_connect("localhost", "root", "123456",
"monitoring");

```

```

function query($query) {
    global $conn;
    $result = mysqli_query($conn, $query);
    $rows = [];
    while( $row = mysqli_fetch_assoc($result) ) {
        $rows[] = $row;
    }
    return $rows;
}

```

```

}
function cari($keyword) {
    $query = "SELECT * FROM tugas_akhir
            WHERE
            waktu LIKE '%$keyword%'
            ";
    return query($query);
}

```

```

$tugas_akhir = query("SELECT * FROM tugas_akhir ORDER BY
waktu DESC");

```

```

if( isset($_POST["cari"]) ) {
    $tugas_akhir = cari($_POST["keyword"]);
}

```

```

?>
<!DOCTYPE html>
<html>
<head>
    <title>Tugas Akhir</title>
</head>
<body>

<h1>Plan Tanah</h1>

<form action="" method="post">
    <input type="text" name="keyword" autofocus
placeholder="Keyword Tanggal" autocomplete="off">
    <button type="submit" name="cari">Cari!</button>

</form>
<br>
<table border="1" cellpadding="10" cellspacing="0">
    <tr>
        <th>waktu</th>
        <th>Kelembaban 1</th>
        <th>Kelembaban 2</th>
        <th>Kelembaban 3</th>
        <th>pH 1</th>
        <th>pH 2</th>
        <th>pH 3</th>
    </tr>
    <?php foreach( $tugas_akhir as $row ) : ?>
    <tr>
        <td><?= $row["waktu"]; ?></td>
        <td><?= $row["sensor_klmb_1"]; ?></td>
        <td><?= $row["sensor_klmb_2"]; ?></td>
        <td><?= $row["sensor_klmb_3"]; ?></td>
        <td><?= $row["sensor_ph_1"]; ?></td>
        <td><?= $row["sensor_ph_2"]; ?></td>
        <td><?= $row["sensor_ph_3"]; ?></td>
    </tr>
    <?php endforeach; ?>

```

88

</body>

</html>

BIODATA PENULIS



Penulis bernama I Putu Dedi Semara Putra, lahir di Denpasar, 02 Mei 1998. Penulis menempuh pendidikan dasar di SDN 1 Baturiti. Kemudian dilanjutkan ke jenjang menengah pertama di SMPN 1 Baturiti. Selanjutnya dilanjutkan ke jenjang menengah atas di SMAN 1 Baturiti. Lalu penulis melanjutkan pendidikan ke perguruan tinggi negeri yaitu di Departemen

Teknik Instrumentasi FV-ITS

Selama di bangku perkuliahan penulis sudah mengikuti beberapa pelatihan, seperti Latihan Keterampilan Manajemen Mahasiswa Pra-Tingkat Dasar (LKMM Pra-TD), Pelatihan Karya Tulis Ilmiah Tingkat Dasar (PKTI-TD), pelatihan *trainer Instrumentation Paper Fighter*. Selain itu, penulis juga menjadi ketua Komunitas Sing Seni Surabaya dan menjadi koordinator Perlengkapan pada acara bakti social, Tirta Yatra dan Medical Day TPKH ITS 2017/2018. Penulis pernah menjadi anggota Sie Konsumsi pada acara SIMAKRAMA UPANAYANA Tahun 2018. Penulis juga pernah menjabat sebagai Wakil Kepala departemen Seni Minat Bakat TPKH ITS periode 2018/2019 dan penulis juga pernah magang STAF di Departemen PSDM TPKH ITS periode 2016/2017. Penulis juga sempat melakukan kerja praktek di PT.Perkebunan Nusantara X (Persero) Pabrik Gula Kremboong Sidoarjo, yang terletak di desa Kremboong kecamatan Krembung kabupaten Sidoarjo Jawa Timur.

Tugas akhir yang diambil penulis yaitu **RANCANG BANGUN SISTEM MONITORING ONLINE PLAN PENGATUR pH DAN KELEMBABAN TANAH BERBASIS MySQL MENGGUNAKAN RASPBERRY PI 3 B+**. Bagi pembaca yang ingin memberikan kritik, saran, maupun pertanyaan silahkan menghubungi penulis melalui email dedibaturiti07@gmail.com atau *idline*: dedisemara.

