

**APLIKASI SISTEM *MONITORING* PADA TANAMAN  
AEROPONIK BERBASIS ANDROID**

**SKRIPSI**

Disusun untuk memenuhi salah satu persyaratan dalam mendapatkan gelar  
Sarjana Teknik



Diajukan Oleh :

**VERA DWI PUSPITASARI**

**3332130692**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS SULTAN AGENG TIRTAYASA  
CILEGON - BANTEN  
2018**

## **PERNYATAAN KEASLIAN SKRIPSI**

Dengan ini saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

### **APLIKASI SISTEM *MONITORING* PADA TANAMAN AEROPONIK BERBASIS ANDROID**

adalah hasil karya sendiri dan sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari karya orang lain yang sudah dipublikasi dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Sultan Ageng Tirtayasa maupun di Perguruan Tinggi atau instansi manapun kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Cilegon, Juli 2018



**Vera Dwi Puspitasari**  
NIM. 3332130692

## PENGESAHAN PEMBIMBING

Skripsi dengan judul:  
**APLIKASI SISTEM *MONITORING* PADA TANAMAN AEROPONIK  
BERBASIS ANDROID**

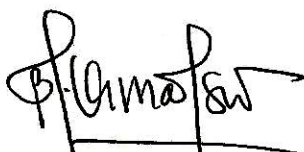
Dipersiapkan dan disusun oleh:  
**Vera Dwi Puspitasari**  
**3332130692**

Dibuat untuk melengkapi persyaratan menjadi Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Sultan Ageng Tirtayasa telah diperiksa dan disetujui oleh pembimbing skripsi.

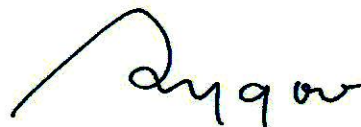
Cilegon, Juli 2018

Pembimbing I

Pembimbing II



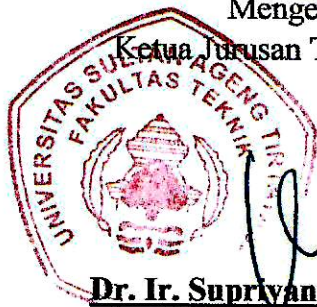
**Dr. Irma Saraswati, S.Si., M.T**  
NIP. 197807242003122001



**Anggoro Suryo P., S.Kom., M.Kom.**  
NIP. 198403042009121010

Mengetahui,

Ketua Jurusan Teknik Elektro



**Dr. Ir. Supriyanto, M.Sc., IPM.**  
NIP. 197605082003121002

## PENGESAHAN PENGUJI

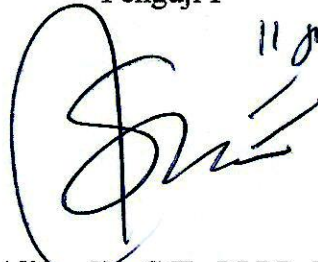
Skripsi dengan judul:  
**APLIKASI SISTEM *MONITORING* PADA TANAMAN AEROPONIK  
BERBASIS ANDROID**

Dipersiapkan dan disusun oleh:  
**Vera Dwi Puspitasari**  
3332130692

Telah dipertahankan di depan Dewan Penguji  
Pada Tanggal 3 Juli 2018

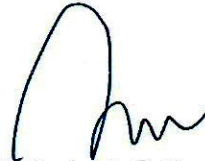
### Susunan Dewan Penguji

Penguji I



**Dr. Alimudin, S.T., M.M., M.T.**  
NIP. 1972204172008121004

Penguji II



**Rian Fahrizal, S.T., M.Eng.**  
NIP. 197510262005011001

Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar sarjana

Mengetahui,  
Ketua Jurusan Teknik Elektro



**Dr. Ir. Supriyanto, M.Sc., IPM.**  
NIP. 197605082003121002

## KATA PENGANTAR



Puji syukur ke hadirat Allah SWT, Tuhan semesta Alam yang telah memberikan nikmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **APLIKASI SISTEM *MONITORING* PADA TANAMAN AEROPONIK BERBASIS ANDROID**. Solawat serta salam semoga tercurah limpah bagi Nabi Muhammad Sallallahu Alaihi Wasallam, keluarga serta para pengikutnya yang setia hingga akhir masa.

Penyusunan skripsi ini merupakan salah satu syarat untuk menyelesaikan program studi S1 dan untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro, Universitas Sultan Ageng Tirtayasa. Selain itu, skripsi ini juga untuk menambah pengetahuan terhadap ilmu yang telah dipelajari di bangku perkuliahan dan menerapkan teori-teori ke dalam kehidupan sehari-hari.

Penulisan skripsi ini tidak akan terwujud tanpa adanya bantuan dari pihak lain. Oleh karena itu, dalam kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah banyak membantu serta membatu dalam pelaksanaan, penulisan, dan penyelesaian skripsi ini, yaitu:

1. Kedua orang tua tercinta yang selalu memberikan do'a, kasih sayang, semangat, nasehat, motivasi dan dukungan serta materi yang tak terhingga kepada penulis.
2. Bapak Dr. Supriyanto, M.Sc., IPM. selaku Ketua Jurusan Teknik Elektro Fakultas Teknis Universitas Sultan Ageng Tirtayasa.
3. Bapak Herudin, S.T., M.T. selaku dosen pembimbing akademik penulis.
4. Ibu Dr. Irma Saraswati, S.Si., M.T. selaku dosen pembimbing I sekaligus Koordinator Skripsi yang telah memberikan bimbingan kepada penulis selama menyelesaikan skripsi ini.
5. Bapak Anggoro Suryo Pramudyo, S.Kom., M.Kom. selaku dosen pembimbing II yang telah memberikan bimbingan kepada penulis selama menyelesaikan skripsi ini.

6. Seluruh dosen dan staff akademik Teknik Elektro Fakultas Teknik Universitas Sultan Ageng Tirtayasa.
7. Ibu Anggari Ayu yang telah berbagi ilmu untuk membantu penulis dalam menyelesaikan penelitian ini.
8. Benny Susilowati dan Anis Tri Zulaikah yang merupakan kakak dan adik penulis yang selalu memberikan do'a, semangat dan dukungannya.
9. Harmi Octaviyanti, Renjana Bening, Dwi Rahmat, Faza Fauzan, Wilda Annisa yang selalu memberikan semangat dan dukungan kepada penulis.
10. Teman-teman mahasiswa Teknik Elektro 2013.
11. Rekan-rekan asisten Laboratorium Jurusan Teknik Elektro Fakultas Teknik Universitas Sultan Ageng Tirtayasa.
12. Semua pihak yang telah banyak membantu yang tidak dapat penulis sebutkan satu persatu, terima kasih banyak atas bantuan, dukungan dan motivasinya selama melaksanakan studi dan menyelesaikan Skripsi ini.

Penulis menyadari bahwa dalam penyusunan Skripsi ini masih banyak terdapat kekurangan dan kesalahan. Untuk itu penulis mengharapkan saran dan kritik yang membangun untuk perbaikan dan pengembangan kedepan. Penulis berharap semoga penelitian ini menjadi sumbangsih yang bermanfaat bagi dunia sains dan teknologi di Indonesia, khususnya disiplin keilmuan yang Penulis dalami.

Cilegon, Juli 2018

Penulis

## ABSTRAK

Sistem pertanian Indonesia mengalami peningkatan yang signifikan seiring dengan perkembangan teknologi. Namun akibat dari perkembangan teknologi mengakibatkan lahan pertanian menjadi sempit. Untuk mengatasi lahan pertanian yang semakin sempit, mulai dikembangkannya sistem cocok tanam dengan sistem aeroponik. Perubahan iklim lingkungan yang tidak menentu dapat mengganggu produktifitas pertumbuhan dan menjadikan tanaman berkualitas rendah. Untuk itu diperlukan suatu sistem *monitoring* berbasis android untuk memantau perubahan nilai parameter, parameter tersebut antara lain suhu, kelembaban, intensitas cahaya serta kadar karbondioksida. Sistem *monitoring* berbasis android menggunakan pemrograman java dan bersifat *open source*, selain itu *user* dapat melakukan *monitoring* perkembangan tanaman bayam, dengan melihat data yang telah tersimpan ke dalam *database* sehingga *user* dapat mengetahui perkembangan dari tanaman. Aplikasi ini dapat diakses melalui *smartphone* dengan sistem operasi android dan bersifat *client server*.

**Kata kunci:** *monitoring*, aeroponik, *client server*, android, *open source*.

## **ABSTRACT**

Indonesia's agricultural system has improved significantly along technological developments. However, as a result of technological developments, the agricultural land becomes narrow. To resolve this problem, developed a farming system with aeroponic system. Unpredictable environmental climate changes can disrupt productivity growth and make quality crops low. Because of that we need an android based monitoring system to monitor changes in parameter values, such parameters include temperature, humidity, light intensity and carbon dioxide levels. Android based monitoring system using java programming and it has an open source operation, in addition users can also monitor the growth of spinach plants by looking at data that has been stored in storage data, so it can help the user to know the development of the plant. This app can be accessed via smartphone with android operating system and has client-server operation.

**Keywords :** monitoring, aeroponic, client server, android, open source.



## DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN SKRIPSI.....	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
KATA PENGANTAR.....	v
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR ISTILAH.....	xiv
 <b>BAB 1 PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	4
 <b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Sistem Aeroponik Tanaman Bayam.....	6
2.1.1 Suhu.....	7
2.1.2 Kelembaban.....	8
2.1.3 Intensitas Cahaya.....	8
2.1.4 Kadar Karbondioksida.....	9
2.2 Sistem <i>Monitoring</i> .....	9
2.3 Android.....	11
2.3.1 Android Studio.....	12
2.3.2 Aeromosis.....	13

2.4 Bahasa Pemrograman Java.....	13
2.5 <i>Unified Modeling Language</i> (UML).....	16
2.6 Basis Data.....	18
2.7 PHP .....	20
2.8 Kajian Pustaka.....	21

### **BAB III METODOLOGI PENELITIAN**

3.1 Metode Penelitian.....	24
3.2 Instrumen Peneltian.....	25
3.2.1 <i>Software</i> .....	25
3.2.2 <i>Hardware</i> .....	27
3.3 Perancangan Penelitian .....	28
3.3.1 Perancangan Sistem .....	29
3.3.2 Perancangan Arsitektur .....	33
3.3.3 Perancangan Tampilan Aplikasi dan Pemrograman Android Studio.....	34
3.4 Tempat dan Waktu Penelitian .....	43

### **BAB IV HASIL DAN PEMBAHASAN**

4.1 Menjalankan Aplikasi Aeromosis .....	44
4.2 Pengujian Antarmuka Aplikasi Aeromosis.....	46
4.3 Hasil Perancangan Antarmuka Aplikasi Aeromosis .....	49
4.4 Pengujian terhadap Berbagai Layar Perangkat .....	62
4.5 Pembahasan Aplikasi Aeromosis.....	64

### **BAB V PENUTUP**

5.1 Kesimpulan .....	67
5.2 Saran.....	68

<b>DAFTAR PUSTAKA</b> .....	69
-----------------------------	----

### **LAMPIRAN**

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Tanaman dengan Sistem Aeroponik .....	6
Gambar 2.2 Tahapan <i>Monitoring</i> .....	10
Gambar 2.3 Lapisan Pengembangan <i>Software</i> .....	11
Gambar 3.1 <i>Flowchart</i> Penelitian .....	24
Gambar 3.2 Android SDK <i>Manager Tool</i> .....	26
Gambar 3.3 Android Studio .....	26
Gambar 3.4 <i>Notepad++</i> .....	27
Gambar 3.5 Blok Diagram Perancangan Penelitian.....	28
Gambar 3.6 <i>Use Case Diagram</i> Aplikasi Android .....	29
Gambar 3.7 <i>Activity Diagram</i> Menu Suhu .....	29
Gambar 3.8 <i>Activity Diagram</i> Menu Kelembaban .....	30
Gambar 3.9 <i>Activity Diagram</i> Menu Intensitas Cahaya .....	31
Gambar 3.10 <i>Activity Diagram</i> Menu Karbondioksida.....	32
Gambar 3.11 <i>Activity Diagram</i> Menu Diskusi Aeroponik .....	33
Gambar 3.12 Rancangan Arsitektur Aplikasi Android .....	33
Gambar 3.13 Diagram Blok Aplikasi Sistem <i>Monitoring</i> .....	34
Gambar 3.14 Menu <i>Splashscreen</i> Aplikasi.....	36
Gambar 3.15 Menu Utama Aplikasi .....	37
Gambar 3.16 Tampilan Nilai Parameter .....	39
Gambar 3.17 Tampilan Submenu Data Grafik .....	42
Gambar 3.18 Tampilan <i>Form</i> Menu Diskusi Aeroponik .....	43
Gambar 4.1 <i>USB Debugging</i> .....	45
Gambar 4.2 <i>Select Deployment Target</i> .....	46
Gambar 4.3 Tampilan <i>SplashScreen</i> pada <i>Smartphone</i> Lenovo A6600d40 .....	50
Gambar 4.4 Tampilan Menu Utama pada <i>Smartphone</i> Lenovo A6600d40.....	50

Gambar 4.5 Tampilan <i>ImageSlider</i> Menu Utama .....	51
Gambar 4.6 Tampilan Menu Parameter Suhu.....	52
Gambar 4.7 Tampilan Nilai Parameter Suhu .....	53
Gambar 4.8 Tampilan Grafik <i>Line</i> Parameter Suhu.....	53
Gambar 4.9 Tampilan Menu Parameter Kelembaban.....	54
Gambar 4.10 Tampilan Nilai Parameter Kelembaban .....	55
Gambar 4.11 Tampilan Grafik <i>Line</i> Menu Parameter Kelembaban .....	56
Gambar 4.12 Tampilan Menu Parameter Intensitas Cahaya.....	56
Gambar 4.13 Tampilan Nilai Parameter Intensitas Cahaya .....	57
Gambar 4.14 Tampilan Grafik <i>Line</i> Menu Parameter Intensitas Cahaya .....	57
Gambar 4.15 Tampilan Menu Parameter Karbondioksida .....	58
Gambar 4.16 Tampilan Nilai Parameter Karbondioksida.....	59
Gambar 4.17 Tampilan Grafik <i>Line</i> Menu Parameter Karbondioksida.....	60
Gambar 4.18 Tampilan Menu Diskusi Aeroponik.....	61
Gambar 4.19 Tampilan Menu Keluar .....	62
Gambar 4.20 <i>Query</i> Data Grafik Parameter Suhu .....	65
Gambar 4.21 Tampilan pada <i>Smartphone</i> Samsung GT-S6310 .....	66

## DAFTAR TABEL

	Halaman
Tabel 2.1 Perbandingan Android Studio dengan Eclipse ADT .....	12
Tabel 4.1 Daftar <i>Layout xml</i> Aplikasi Aeromosis.....	47
Tabel 4.2 Daftar <i>File</i> Java Aplikasi Aeromosis .....	48
Tabel 4.3 Pengujian Antarmuka Menu Utama.....	49
Tabel 4.4 Pengujian Antarmuka Menu Parameter Suhu .....	52
Tabel 4.5 Pengujian Antarmuka Menu Parameter Kelembaban .....	54
Tabel 4.6 Pengujian Antarmuka Menu Parameter Intensitas.....	56
Tabel 4.7 Pengujian Antarmuka Menu Parameter Karbondioksida.....	58
Tabel 4.8 Pengujian Antarmuka Menu Diskusi Aeroponik .....	60
Tabel 4.9 Pengujian Antarmuka Menu Keluar .....	61
Tabel 4.10 Daftar Berbagai Perangkat yang digunakan.....	63
Tabel 4.11 Hasil Pengujian dari Berbagai Ukuran Layar Perangkat .....	63

## DAFTAR ISTILAH

<i>Activity Diagram</i>	Diagram yang menggambarkan <i>workflow</i> (aliran kerja) atau aktivitas dari sebuah sistem.
<i>Android</i>	Sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.
<i>Client server</i>	Sebuah paradigma dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi ke dalam dua pihak.
<i>Database</i>	Kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.
<i>Emulator</i>	Sebuah program yang membuat pengguna komputer bisa merasakan sistem operasi tertentu seperti Android, iPhone atau atau program lainnya yang tidak bisa dijalankan mandiri di komputer.
<i>File</i>	Kumpulan berbagai informasi yang berhubungan dan juga tersimpan di dalam <i>secondary storage</i> ,
<i>GUI</i>	Jenis antarmuka pengguna yang menggunakan metode interaksi pada peranti elektronik secara grafis (bukan perintah teks) antara pengguna dan komputer.
<i>ImageSlider</i>	Gambar yang berjalan ( <i>sliding</i> ) secara otomatis dalam tampilan <i>website</i> .
Java	Bahasa pemrograman yang dapat diaplikasikan pada internet.
<i>JSON</i>	Format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat ( <i>generate</i> ) oleh komputer.

<i>Monitoring</i>	Aktifitas yang ditujukan untuk memberikan informasi tentang sebab dan akibat dari suatu kebijakan yang sedang dilaksanakan agar kesalahan awal dapat segera diketahui dan dapat dilakukan tindakan perbaikan, sehingga mengurangi risiko yang lebih besar.
<i>Open Source</i>	Sistem pengembangan yang tidak dikoordinasi oleh suatu individu/lembaga pusat, tetapi oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber ( <i>source-code</i> ) yang tersebar dan tersedia bebas (biasanya menggunakan fasilitas komunikasi internet).
<i>Php</i>	Bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML.
<i>Smartphone</i>	Telepon genggam yang mempunyai kemampuan tingkat tinggi, terkadang dengan fungsi yang menyerupai komputer.
<i>Server</i>	Sebuah sistem komputer yang menyediakan jenis layanan ( <i>service</i> ) tertentu dalam sebuah jaringan komputer.
<i>Splashscreen</i>	Tampilan pertama program yang muncul sementara sebelum masuk ke menu utama.
<i>Use Case Diagram</i>	Sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.
<i>User</i>	Pengguna suatu sistem yang umumnya adalah manusia.
<i>Webhosting</i>	Jasa penyewaan server dan aplikasi/ <i>software</i> untuk keperluan web server.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Indonesia merupakan negara agraris yang mayoritas penduduknya bekerja pada bidang pertanian. Seiring dengan perkembangan teknologi yang signifikan sistem pertanian mulai mengembangkan sistem cocok tanam dengan macam-macam sistem, salah satu contoh sistem yang mulai dikembangkan adalah sistem aeroponik. Sistem aeroponik menggunakan media udara untuk bercocok tanam tanpa media tanah. Dengan cara menggantungkan tanaman pada udara kemudian nutrisi disemprotkan pada akar tanaman [1]. Tanaman yang cocok untuk sistem *aeroponik* adalah sayuran [2].

Objek yang digunakan pada penelitian ini adalah tanaman bayam, karena bayam memerlukan waktu pembudidayaan yang relatif singkat berkisar 25 - 35 hari setelah tanam [3]. Tanaman bayam dapat tumbuh dengan baik, jika beberapa parameter terpenuhi. Parameter yang diperlukan antara lain seperti suhu, kelembaban, kadar karbondioksida dan pencahayaan yang cukup. Jika salah satu parameter tersebut tidak terpenuhi atau berlebihan maka hasil panen pun tidak maksimal dan mengakibatkan tanaman bayam menjadi kering bahkan tidak akan tumbuh [4].

Pengaruh perubahan iklim dalam bidang pertanian ditandai dengan adanya bencana banjir, kekeringan dan bergesernya musim hujan. Perubahan iklim tersebut mengakibatkan produktifitas pertumbuhan berkurang dan menjadikan tanaman berkualitas rendah [5]. Untuk itu memerlukan suatu sistem pemantauan (*monitoring*) secara kontinu dari penanaman benih hingga waktu panen. *Monitoring* pertumbuhan tanaman secara manual memiliki keterbatasan yang disebabkan oleh faktor fisik manusia meliputi kelelahan, tidak kontinu, ketidakseragaman dan ketidakteelitian. Solusi dari permasalahan tersebut ialah dengan melakukan penerapan teknologi berupa *monitoring* jarak jauh. *Monitoring* jarak jauh sangatlah efektif karena data hasil *monitoring* langsung tersimpan ke dalam *database* [6].



Sudah banyak penelitian yang membahas mengenai sistem *monitoring* jarak jauh. Beberapa contoh penelitian mengenai sistem *monitoring* jarak jauh sebelumnya adalah sistem *monitoring* tanaman bayam yang memanfaatkan Zigbee sebagai standar komunikasi pengiriman data. Namun jarak atau *range* kerja dari ZigBee sekitar 76 meter. Data yang diterima *gateway* diteruskan ke server menggunakan platform M2M. Penelitian ini hanya meneliti beberapa parameter saja yaitu kelembaban tanah, sensor cahaya dan sensor ketinggian serta perlu adanya penambahan parameter untuk perkembangan tanaman bayam. Penelitian *monitoring* tanaman ini dapat dikembangkan menjadi aplikasi *mobile* [3].

Penelitian kedua yakni mengenai sistem *monitoring* pada tanaman cabai dengan dengan parameter suhu dan kelembaban berbasis LabVIEW. Pada *greenhouse* digunakan beberapa sensor seperti sensor DHT11 yang berfungsi sebagai pendeteksi suhu dan kelembaban lingkungan, sensor *soil moisture* sebagai pendeteksi kelembaban tanah dan penambahan sensor LDR sebagai pendeteksi intensitas cahaya, serta terdapat data *logger* yang menyimpan data hasil *monitoring* secara berkala sebagai bahan evaluasi. Hasil *monitoring* suhu dan kelembaban masih menggunakan layar LCD 4x20 karakter dan *front* panel LabVIEW 2013 [7].

Penelitian ketiga yaitu membahas tentang sistem *monitoring* pada tanaman cabai dan tomat berdasarkan hasil penyiraman tanaman. *Monitoring* penyiraman tanaman berdasarkan kelembaban tanah melalui SMS berbasis *mikrokontroler* merupakan salah satu cara yang dapat digunakan dalam hal mengawasi serta merawat tanaman tetap dalam kondisi yang baik. Penelitian ini kurang maksimal karena hanya membahas kelembaban tanah saja, dan tergantung pada kondisi jaringan seluler. Jaringan seluler yang kurang baik dapat menghambat penerimaan SMS [8].

Penelitian keempat berfokus pada bagaimana membangun sistem *monitoring* dan *controlling* air nutrisi *aquaponik* (PH dan EC) menggunakan Arduino Uno. Dengan sistem *aquaponik* ini dapat membantu *user* untuk *monitoring* PH dan EC dari jarak jauh sekaligus memberikan batasan agar kondisi air pada *aquaponik* tetap stabil. Untuk pengembangan sistem *monitoring* dan *controlling* air nutrisi *aquaponik* lebih lanjut dapat dikembangkan dalam lingkungan yang lebih luas agar

menghasilkan produk dalam jumlah yang lebih besar. Tampilan hasil *monitoring* masih dalam bentuk tampilan *website* [9].

Berdasarkan uraian dari penelitian yang telah dilakukan sebelumnya, penelitian ini difokuskan pada sistem *monitoring* saja dengan parameter suhu, kelembaban intensitas cahaya serta kadar karbondioksida. Pembeda penelitian ini dengan penelitian sebelumnya adalah menggunakan aplikasi android untuk *monitoring* jarak jauh pada tanaman bayam dengan sistem aeroponik. Android merupakan salah satu sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Android dibuat dengan menggunakan pemrograman Java dan bersifat *open source* sehingga dapat berkembang dengan cepat dan juga menyediakan platform yang terbuka [10]. Selain itu pada sistem aplikasi ini *user* dapat melakukan *monitoring* perkembangan tanaman bayam, dengan melihat data yang telah tersimpan ke dalam *database* sehingga *user* bisa mengetahui perkembangan dari tanaman bayam. Aplikasi ini diakses melalui *smartphone* dengan sistem operasi Android dan bersifat *client server*.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah dijelaskan, maka pada penelitian ini dirumuskan masalah yang perlu diperhatikan yaitu cara menampilkan nilai dan data grafik pada parameter suhu, kelembaban, intensitas cahaya serta kadar karbondioksida pada tanaman bayam aeroponik yang telah tersimpan ke dalam *database* dan dapat ditampilkan pada aplikasi android.

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah membuat sebuah aplikasi sistem *monitoring* sederhana pada tanaman bayam aeroponik berbasis android yang dapat dijalankan pada *smartphone*, dapat menampilkan nilai dan data grafik dari parameter suhu, kelembaban, intensitas cahaya serta karbondioksida.

#### 1.4 Manfaat Penelitian

Manfaat penelitian ini adalah untuk meningkatkan kualitas hasil panen terutama sayuran bayam dan mendapatkan hasil panen yang seragam, dapat memudahkan *user* dalam pemantauan tanpa perlu datang ke lokasi dan dapat mengetahui perkembangan pada tanaman bayam aeroponik yang telah tersimpan ke dalam *database*. Aplikasi ini dapat diakses melalui *smartphone* dengan sistem operasi Android dan bersifat *client server*.

#### 1.5 Batasan Masalah

Batasan pada penelitian ini adalah:

1. Variabel yang digunakan pada penelitian ini adalah suhu, kelembaban, kadar CO<sub>2</sub> serta intensitas cahaya.
2. Satuan yang digunakan untuk suhu adalah °C, kelembaban adalah %RH, intensitas cahaya adalah lux serta kadar karbondioksida adalah ppm.
3. Menggunakan *software* Android studio versi 3.0. Penelitian ini hanya membahas mengenai sistem *monitoring* berbasis android saja.
4. Data yang tersimpan pada *database* akan ditampilkan dalam bentuk grafik pada aplikasi android.
5. Aplikasi sistem *monitoring* ini tidak menampilkan notifikasi untuk *user*, jika terjadi perubahan nilai pada parameter.

#### 1.6 Sistematika Penulisan

Sistematika penulisan dari skripsi diuraikan dalam 5 bab, dengan sistematika sebagai berikut:

### BAB I PENDAHULUAN

Menjelaskan latar belakang, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan dari penelitian.

## BAB II TINJAUAN PUSTAKA

Membahas tentang dasar teori penunjang dan dasar-dasar ilmu yang berkaitan dengan penelitian.

## BAB III METODOLOGI PENELITIAN

Menjelaskan mengenai metode yang digunakan untuk memecahkan masalah yang telah dijelaskan pada perumusan masalah dan juga akan dibahas mengenai perancangan penelitian.

## BAB IV HASIL DAN PEMBAHASAN

Berisi penjelasan dan analisa dari metode yang digunakan dan hasil yang diperoleh.

## BAB V PENUTUP

Menjelaskan kesimpulan dan saran yang diperoleh setelah dilakukan penelitian. Kesimpulan diambil dari intisari bab-bab sebelumnya.

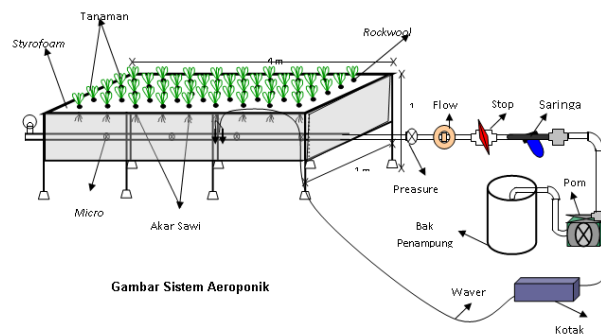
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Aeroponik Tanaman Bayam

Aeroponik merupakan suatu cara bercocok tanam sayuran di udara tanpa penggunaan tanah, nutrisi disemprotkan pada akar tanaman. Air yang berisi larutan hara disemprotkan menggunakan *mist spayer* dalam bentuk kabut hingga mengenai akar tanaman. Akar tanaman yang ditanam menggantung akan menyerap larutan hara tersebut. Aeroponik berasal dari kata *aero* yang berarti udara dan *ponic* yang artinya cara budidaya, jadi arti dari aeroponik adalah cara bercocok tanam di udara. Akar tanamannya menggantung di udara tanpa perantara media (bahan organik dan bahan anorganik).

Sistem aeroponik dapat memberikan manfaat bagi petani yang tidak mempunyai lahan, karena aeroponik tidak membutuhkan tanah. Sehingga bisa dijadikan sebagai lahan di pekarangan rumah. Selain itu sistem aeroponik juga menghasilkan pertumbuhan tanaman yang cepat dan mendapatkan hasil produksi yang seragam [4]. Ilustrasi dari sistem aeroponik dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tanaman dengan Sistem Aeroponik [4]

Gambar 2.1 menjelaskan mengenai cara kerja dari sistem aeroponik. Selama perjalanan dari lubang *mist spayer* hingga sampai ke akar, butiran akan mengikat oksigen dari udara hingga kadar oksigen terlarut dalam butiran meningkat. Dengan demikian proses respirasi pada akar dapat berlangsung lancar dan menghasilkan banyak energi. Selain itu dengan pengelolaan yang terampil, produksi dengan sistem aeroponik dapat memenuhi kualitas, kuantitas dan kontinuitas [2].

Pada penelitian ini menggunakan sistem aeroponik yang diterapkan pada tanaman bayam karena memiliki masa panen yang singkat. Bayam (*Amaranthus viridis*) merupakan tanaman semusim dan tergolong sebagai tumbuhan C4 yang mampu mengikat gas CO<sub>2</sub> secara efisien sehingga memiliki daya adaptasi yang tinggi pada beragam ekosistem. Bayam mempunyai siklus hidup yang relatif singkat, yaitu sekitar 25 - 35 hari [3]. Sistem perakarannya adalah akar tunggang dengan cabang-cabang akar yang berbentuk bulat panjang yang menyebar ke semua arah.

Tanaman bayam cocok ditanam di dataran tinggi maka curah hujannya juga termasuk tinggi sebagai syarat pertumbuhannya. Tanaman bayam sangat cocok dibudidayakan pada dataran dengan curah hujan yang cukup tinggi yakni lebih dari 1500 mm/tahun serta dapat tumbuh baik jika pH tanah antara 6 - 7 [2]. Kebutuhan akan sinar matahari untuk tanaman bayam cukup besar. Pada tempat yang kurang cahaya matahari tumbuhan bayam menjadi kurus dan meninggi. Suhu udara yang sesuai untuk tanaman bayam berkisar antara 18 - 32 °C [11]. Kelembaban yang cocok untuk tanaman bayam berkisar antara 70 % [12], intensitas cahaya sebesar 8.000 - 9.000 lux di dalam ruangan dan 42.000 - 69.000 lux di luar ruangan [13] serta kadar karbondioksida 380 - 1.000ppm [14].

Faktor lingkungan menjadi salah satu faktor yang berpengaruh pada sistem aeroponik. Penelitian ini dilakukan pada miniatur *greenhouse* yang didalamnya terdapat *box* aeroponik. *Box* aeroponik tidak harus besar tetapi dapat digunakan dalam ukuran kecil. Pada *box* aeroponik dapat dipasang kipas atau *fan* yang berfungsi untuk mengalirkan udara keluar dari dalam *box* ketika suhu dan kelembaban tinggi [2]. Beberapa faktor lingkungan tersebut di antaranya adalah suhu, kelembaban, intensitas cahaya serta kadar karbondioksida.

### **2.1.1 Suhu**

Suhu merupakan faktor lingkungan yang berpengaruh terhadap pertumbuhan dan perkembangan tanaman. Suhu mempengaruhi beberapa proses fisiologis, tinggi rendahnya suhu di sekitar tanaman ditentukan oleh banyaknya intensitas cahaya yang ada serta kerapatan peletakan tanaman. Suhu sangat berpengaruh

terhadap pertumbuhan karena berkaitan dengan aktifitas enzim dan kandungan air dalam tubuh tumbuhan. Semakin tinggi suhu, maka semakin cepat laju transpirasi, sedangkan semakin rendah kandungan air pada tumbuhan sehingga proses tumbuhan semakin lambat. Suhu yang rendah dapat mempercepat pembentukan ruas yang lebih panjang daripada perlakuan suhu yang tinggi. Suhu yang tidak stabil dapat mempengaruhi pertumbuhan tanaman. Suhu yang digunakan pada sistem aeroponik berkisar 18 - 32 °C. Suhu yang terlalu tinggi akan mempengaruhi suhu larutan nutrisi pada bak penampungan. Pada larutan bersuhu tinggi kadar oksigen dalam larutan menurun yang mengakibatkan akar kekurangan energi untuk menyerap air [11].

### **2.1.2 Kelembaban**

Kelembaban dibutuhkan oleh tanaman agar tanaman tidak cepat kering karena penguapan. Kelembaban yang diperlukan tanaman berbeda-beda sesuai dengan jenisnya. *Relative Humidity* (RH) merupakan persentase kandungan air diudara pada suhu tertentu. Kondisi RH untuk sistem aeroponik khususnya bayam adalah 70 %. RH terlalu tinggi maka evaporasi dan daya serap akar tanaman untuk mendapatkan nutrisi berkurang. RH terlalu rendah (pada suhu tinggi), maka evaporasi akan berlangsung terlalu cepat dan tidak dapat diimbangi dengan penguasaan air oleh akar, dan menyebabkan tanaman menjadi layu. Tanaman menjadi gosong pucuk pada tepi daun. Warna hitam pada tepi daun dan menyebabkan penurunan kualitas tanaman [2].

### **2.1.3 Intensitas Cahaya**

Intensitas cahaya mempengaruhi pertumbuhan melalui proses fotosintesis, pembukaan *stomata*. Cahaya dapat mengubah *leukoplas* menjadi *kloroplas*. Intensitas cahaya yang cukup akan meningkatkan pembentukan *kloroplas*, sehingga tanaman yang mendapatkan cukup cahaya memiliki daun yang berwarna hijau. Tumbuhan yang sejenis dapat memiliki ukuran daun yang berbeda jika tumbuh di tempat dengan intensitas cahaya yang berbeda. Tanaman dengan intensitas cahaya yang cukup memiliki daun dengan jaringan *mesofil* lebih tebal daripada daun

dengan intensitas cahaya yang kurang. Selain itu, sistem perakarannya lebih lebat dibandingkan dengan sistem perakaran tumbuhan dengan intensitas cahaya yang kurang. Intensitas cahaya yang diperlukan oleh tanaman bayam sekitar 8.000 - 9.000 lux didalam ruangan dan 42.000 - 69.000 lux di luar ruangan [13].

#### **2.1.4 Kadar Karbondioksida**

Karbondioksida ( $\text{CO}_2$ ) merupakan salah satu gas penting dalam pertumbuhan tanaman, namun juga merupakan salah satu gas rumah kaca yang jumlahnya terus meningkat dari tahun ke tahun. Kenaikan kadar  $\text{CO}_2$  yang semakin meningkat akibat penggunaan gas-gas berbahaya saat ini tentu sangat mempengaruhi keadaan lingkungan sekitar juga. Seperti penggunaan bahan bakar kendaraan dan limbah pabrik yang tidak ramah lingkungan menjadi salah satu pemicu kenaikan kadar  $\text{CO}_2$ .  $\text{CO}_2$  yang menjadi bahan utama dalam fotosintesis cukup banyak diperlukan bagi tanaman, namun jika kenaikan kadar  $\text{CO}_2$  yang tidak disertai dengan penambahan jumlah tanaman juga dapat berpengaruh buruk bagi lingkungan sekitar. Kadar karbondioksida yang diperlukan oleh tanaman bayam adalah sekitar 380 - 1.000 ppm [14].

#### **2.2 Sistem *Monitoring***

Sistem adalah sekumpulan elemen yang saling berkaitan antara satu dengan yang lainnya membentuk suatu kesatuan agar mencapai suatu tujuan. Pendekatan sistem lebih menekankan pada prosedur, menjelaskan suatu jaringan kerja yang saling berhubungan untuk menyelesaikan tujuan dengan menekankan urutan operasi didalam sebuah sistem [15]. Menganalisa dari siklus perkembangan sistem, mendefinisikan kebutuhan fungsional dan mempersiapkan rancang bangun serta menggambarkan bagaimana suatu sistem dibentuk merupakan tahapan dari sistem yaitu konsep dari perancangan sistem [16]. Sistem memiliki beberapa karakteristik yang harus terpenuhi diantaranya:

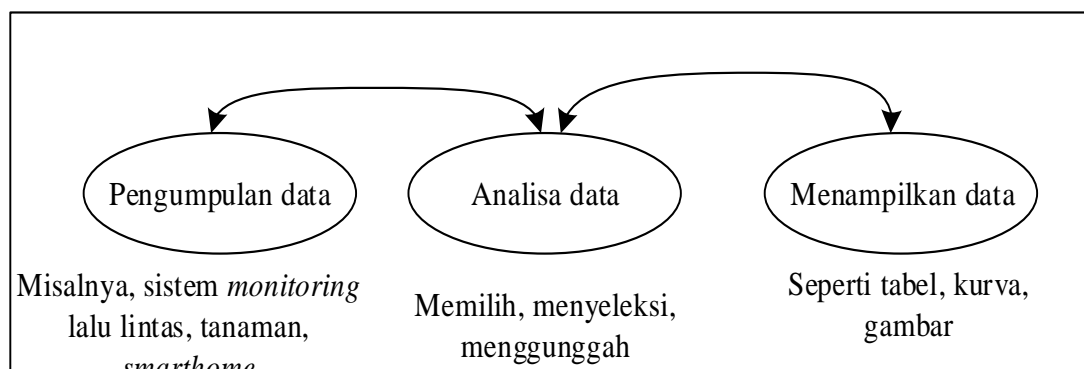
1. Batas sistem digunakan untuk membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luar.



2. Lingkungan luar sistem dari suatu sistem adalah apapun yang berada di luar batas dari sistem yang mempengaruhi operasi sistem.
3. Penghubung sistem media, penghubung antara satu subsistem dengan subsistem lainnya.
4. *Input* dari suatu sistem sebelum diproses, berupa data lapangan, sinyal dan sebagainya.
5. Hasil proses yang diolah dan diklasifikasikan menjadi keluaran yang diinginkan.
6. Pengolah sistem dapat menjadi bagaian pengolah dari sistem tersebut.
7. Sasaran sistem mempunyai tujuan atau sasaran, kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya [17].

*Monitoring* merupakan pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan yang fokus pada proses dan keluaran. *Monitoring* menyediakan data dasar untuk menjawab permasalahan. *Monitoring* bertujuan untuk mendapatkan umpan balik bagi kebutuhan program, proses pembelajaran yang sedang berjalan, dan mengetahui kebutuhan ini pelaksanaan program dengan mempersiapkan kebutuhan dalam pembelajara. Kebutuhan bisa berupa biaya, waktu, personel, dan alat. Secara garis besar tahapan *monitoring* dapat dilihat pada Gambar 2.2, yang terbagi menjadi tiga proses antara lain [18]:

1. Proses pengumpulan data *monitoring*.
2. Proses analisa data *monitoring*.
3. Proses menampilkan data hasil *monitoring*.



Gambar 2.2 Tahapan *Monitoring* [18]

### 2.3 Android

Android merupakan sebuah sistem operasi *mobile* yang berbasis *open source linux kernel* yang mencakup sistem operasi, *middleware*, dan aplikasi. Secara umum arsitektur android terdiri dari *application*, *application framework*, *libraries*, *android runtime* dan *linux kernel* [19]. Android merupakan platform yang sangat lengkap baik dalam segi sistem operasi, aplikasi dan *tool* pengembangan bahkan *market* aplikasi android sehingga membuat android mengalami perkembangan dalam segi teknologi maupun jumlah *device* yang ada didunia. Pengembangan perangkat lunak (*software engineering*) merupakan kumpulan suatu proses dengan metode-metode dan *device* yang memungkinkan dalam pengembangan *software* komputer yang berkualitas tinggi [20]. Terdiri dari empat lapisan dalam pengembangan *software* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Lapisan Pengembangan *Software* [21]

Fondasi lapisan yang berkaitan dengan rekayasa *software* adalah lapisan proses. Proses-proses perangkat lunak membentuk dasar bagi kendali manajemen proyek perangkat lunak dan membangun konteks dengan metode-metode teknis yang dapat diterapkan, produk-produk kerja (model, dokumen, data, laporan, formulir) tonggak waktu dibuat, kualitas dipastikan, dan perubahan dapat diatur dengan tepat. Proses secara umum terdiri dari komunikasi, perencanaan, pemodelan, konstruksi, dan penyerahan perangkat lunak ke pelanggan/pengguna. Metode rekayasa perangkat lunak menyajikan prosedur teknis untuk mengembangkan perangkat lunak yang mencakup sederetan tugas yang

menyertakan komunikasi, analisis kebutuhan, pemodelan rancangan, pembuatan program, pengujian, dan dukungan-dukungan terhadap pengguna [21].

### 2.3.1 Android Studio

Android studio adalah sebuah *Integrated Development Environment (IDE)* untuk platform android. Android Studio merupakan pengembangan dari eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android studio direncanakan untuk menggantikan eclipse ke depannya sebagai IDE resmi untuk pengembangan aplikasi android, sebagai pengembangan dari eclipse, android studio memiliki fitur-fitur baru dibandingkan dengan eclipse IDE. Berbeda dengan eclipse yang menggunakan ant, android studio menggunakan *gradle* sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut [22]:

1. Menggunakan *gradle-based build system* yang fleksibel.
2. Bisa mem-*build multiple APK*.
3. *Layout editor* yang lebih bagus.
4. *Template support* untuk *google services* dan berbagai macam tipe perangkat.
5. *Built-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. *Import library* langsung dari *Maven repository*.

Berikut adalah hasil perbandingan android studio dengan eclipse ADT dapat dilihat pada Tabel 2.1 [23].

Tabel 2.1 Perbandingan Android Studio dengan Eclipse ADT

<b>Feature</b>	<b>Android Studio</b>	<b>Eclipse ADT</b>
<i>Build system</i>	<i>Gradle</i>	<i>Apache Ant</i>
<i>Maven-based build independencies</i>	<i>Yes</i>	<i>No</i>
<i>Build variants and multiple-APK generation</i>	<i>Yes</i>	<i>No</i>
<i>Build variants and multiple-APK generation</i>	<i>Yes</i>	<i>No</i>
<i>Graphical layout editor</i>	<i>Yes</i>	<i>Yes</i>
<i>APK signing and keystore management</i>	<i>Yes</i>	<i>Yes</i>
<i>NDK support</i>	<i>Yes</i>	<i>Yes</i>

Android studio dapat diunduh pada halaman web resmi android studio yaitu <https://developer.android.com/studio/> didalam web resmi android studio tersebut sudah ada panduan lengkap dalam memasang dan menggunakan android studio. selain itu dalam web tersebut juga terdapat beberapa contoh dalam pembuatan aplikasi [24].

### 2.3.2 Aeromosis

Aeromosis merupakan singkatan dari *aerophonic monitoring system*. Aeromosis ini merupakan aplikasi sistem *monitoring* jarak jauh berbasis android pada tanaman aeroponik. Pada aplikasi ini memuat mengenai beberapa parameter, diantaranya suhu, kelembaban, intensitas cahaya, karbondioksida. Aplikasi aeromosis berfungsi untuk memudahkan *user* dalam pemantauan jarak jauh, dan sangat efektif karena bisa dilihat oleh semua *user* dan dimanapun *user* berada dengan cara memasang aplikasi aeromosis pada *smartphone*. Ukuran aplikasi aeromosis ini sangat kecil yaitu 1,40 MB. Dengan adanya koneksi internet, aplikasi aeromosis dapat memuat nilai dan data grafik dari parameter. Baik nilai dan data grafik disesuaikan dengan data yang masuk pada *database*.

Pembuatan aplikasi aeromosis menggunakan *software* android studio versi 3.0. *Software* android studio ini terdiri dari dua bahasa pemrograman yaitu xml yang berfungsi untuk membuat desain dari tampilan aplikasi dan java untuk menjalankan xml agar dapat dijalankan. Antara xml dengan java harus saling berkesinambungan agar aplikasi dapat berjalan sesuai dengan perancangan. Pembuatan aplikasi ini terdiri dari 20 *layout* xml dan 30 *file* java.

## 2.4 Bahasa Pemrograman Java

Java adalah bahasa pemrograman berorientasi objek yang serbaguna. Bahasa ini menyediakan sejumlah perluasan yang mendukung pengembangan aplikasi GUI (antarmuka kepada pemakai yang berbentuk grafis). Bahasa pemrograman java dapat berjalan pada sembarang platform yang memiliki *interpreter* dan lingkungan *runtime*. Walaupun java adalah bahasa penerjemah, program java harus dikompilasi terlebih dulu. *Compiler java (javac)* mengubah

kode sumber program menjadi *bytecode* yang dapat dieksekusi dalam lingkungan *runtime* java. Kemampuan *multithreading* java dapat digunakan untuk membangun aplikasi yang bisa melakukan beberapa pekerjaan atau aktivitas secara bersamaan. Lingkungan dinamis merupakan *class* yang dapat dimuat atau dapat diubah tanpa mengganggu program *class* tersebut. Berikut ini adalah karakteristik dari bahasa pemrograman java [25]:

a. *Simple*

Bahasa java merupakan bahasa sederhana yang masih cukup komprehensif untuk mengatasi berbagai macam pengembangan aplikasi perangkat lunak.

b. Berorientasi Objek

Berorientasi objek yaitu bahasa pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek.

c. Familiar

Java termasuk bahasa yang familiar bagi mayoritas *programmer* di komputer, sebab bahasa java tampak seperti bahasa C dan C++, dimana sebagian besar *programmer* telah akrab dengan bahasa C dan C++.

d. Kuat

Bahasa java termasuk bahasa yang kuat karena bahasa java lebih teliti mendeteksi kesalahan dan java memiliki *runtime exception handling* untuk membantu mengatasi kesalahan.

e. Aman

Bahasa java termasuk bahasa pemrograman yang aman untuk lingkungan aplikasi berbasis jaringan.

f. Arsitektur Netral

Bahasa java merupakan bahasa yang memiliki arsitektur netral dikarenakan bisa dieksekusi di berbagai perangkat lunak maupun berbagai platform dan sistem operasi.

g. *Portable*

Program java dapat digunakan di platform yang berbeda tanpa kompilasi ulang.

#### h. Kinerja Tinggi

Kinerja selalu menjadi pertimbangan. Java mencapai kinerja yang unggul dengan mengadopsi skema dimana eksekusi program dapat berjalan pada kecepatan penuh tanpa perlu memeriksa lingkungan *runtime*.

Pemrograman berorientasi objek merupakan paradigma untuk membuat program yang dapat bertahan terhadap kemungkinan perubahan yang berkaitan dengan pertumbuhan dan pertambahan unsur sistem. Berikut ini konsep dasar dari pemrograman berorientasi objek [25]:

##### a. Kelas

Struktur dasar pada java berupa *class*. Pemrograman java terdiri atas sekumpulan definisi *class*. Sebuah *class* adalah sebuah struktur data, serupa dengan *record* pada pascal atau *struct* pada bahasa C, dengan beberapa sifat tambahan.

##### b. Instance

Kelas pada java berhubungan dengan tipe terstruktur seperti pada bahasa C atau pascal. *Instance* merupakan keadaan tertentu sebuah *class*.

##### c. Pewarisan

Penggunaan pewarisan dapat menghilangkan duplikasi kode karena banyak atribut dan metode yang hanya perlu didefinisikan sekali dan dapat dipakai secara bebas dalam sub kelas.

##### d. Enkapsulasi

Enkapsulasi berguna untuk membatasi akses terhadap bagian-bagian dalam program. Enkapsulasi ini mendefinisikan perilaku untuk melindungi program dan data agar tidak dapat diakses sembarangan oleh program lain.

##### e. Polimorfisme

*Polimorfisme* yang berarti satu objek dengan banyak bentuk. *Polimorfisme* merupakan konsep sederhana yang memperbolehkan metode memiliki beberapa implementasi yang dipilih berdasarkan tipe objek yang dilewatkan pada pengerjaan metode.

Kelebihan Java yang pertama tentu saja multiplatform. Java dapat dijalankan dalam beberapa platform komputer dan sistem operasi yang berbeda.

Hal ini sesuai dengan slogannya yang udah dibahas sebelumnya. Yang kedua adalah OOP atau *Object Oriented Programming*. Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal ini akan memudahkan pemrograman menjadi lebih mudah. Kelengkapan *library* semakin beragam jika ditambah dengan karya komunitas Java. Setiap hal pasti memiliki kelebihan dan kekurangan [26].

## 2.5 Unified Modeling Language (UML)

UML adalah bahasa pemodelan visual, bukan dimaksudkan untuk menjadi suatu bahasa pemrograman visual, tetapi UML memberikan arah untuk bergerak ke arah kode. Bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum. UML mendefinisikan semantik dan notasi, bukan sebuah *tool*. Jadi dalam mengaplikasikan dapat menggunakan beberapa *tool* yang menggunakan UML sebagai bahasanya diantaranya yaitu *Rational Rose* dan *Enterprise Architect* [27]. Adapun jenis-jenis diagram antara lain :

### 1. Use Case Diagram

*Use case* adalah abstraksi dari interaksi antara *system* dan *actor*. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana *system* akan terlihat di mata *user*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.

### 2. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas *activity diagram*, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

### 3. State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa objek lain yang mengirim pesan. *State class* tidak digambarkan

untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

#### 4. *Class Diagram*

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur, deskripsi *class*, *package* dan objek beserta hubungan satu sama lain.

#### 5. *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Kegunaan dari *sequence diagram* untuk menunjukkan rangkaian pesan yang dikirim antara objek dengan interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

#### 6. *Component Diagram*

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*). Komponen piranti lunak adalah modul berisi kode, *source code* maupun *binary code*, *library* maupun *executable*, yang muncul pada *compile time*, *link time*, maupun *runtime*. Umumnya komponen terbentuk dari beberapa *class* atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

#### 7. *Deployment Diagram*

*Deployment/physical diagram* menggambarkan detail komponen di-*deploy* dalam infrastruktur sistem, komponen akan terletak (pada mesin, server atau piranti keras), kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya.



### 8. *Collaboration Diagram*

Menggambarkan kolaborasi dinamis seperti *sequence diagram*. Dalam menunjukkan pertukaran pesan *collaboration diagram* menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan menggunakan *sequence diagram*, tetapi jika penekanannya pada konteks menggunakan *collaboration diagram* [28].

## 2.6 Basis Data

Basis data adalah kumpulan data yang saling berhubungan secara logis dan dideskripsikan serta dirancang untuk memenuhi kebutuhan informasi dalam suatu organisasi. Basis data merupakan tempat penyimpanan data yang besar yang dapat digunakan secara bersamaan oleh banyak departemen dan banyak pengguna. DBMS (*Database Management System*) adalah sebuah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke basis data. DBMS merupakan perangkat lunak yang berinteraksi dengan program aplikasi pengguna dan basis data. Ada lima komponen utama dalam lingkungan DBMS yaitu [29]:

### 1. Perangkat Keras

DBMS dan aplikasi memerlukan perangkat keras untuk dapat berjalan. Perangkat keras dapat berupa satu komputer pribadi, satu *mainframe* maupun suatu jaringan komputer. Pemilihan perangkat keras untuk DBMS tergantung pada kebutuhan organisasi dan DBMS yang digunakan. Beberapa DBMS hanya dapat berjalan pada perangkat keras atau sistem operasi tertentu, sementara DBMS yang lain dapat berjalan pada berbagai perangkat keras dan sistem operasi.

### 2. Perangkat Lunak

Komponen perangkat lunak terdiri dari perangkat lunak DBMS dan program aplikasi, bersama dengan sistem operasi, termasuk perangkat lunak jaringan jika DBMS yang digunakan melalui jaringan. Biasanya, program aplikasi ditulis dalam bahasa pemrograman generasi ketiga (3GL), seperti

'C', C ++, Java, Visual Basic, COBOL, Fortran, atau Pascal, atau menggunakan bahasa generasi keempat (4GL), seperti SQL.

### 3. Data

Komponen yang paling penting dari lingkungan DBMS dari sudut pandang pengguna adalah data. Data bertindak sebagai jembatan antara komponen mesin dan komponen manusia. Basis data berisi data operasional dan metadata (data yang memberikan penjelasan mengenai data).

### 4. Prosedur

Prosedur mengacu pada instruksi dan aturan yang mengatur desain dan penggunaan basis data. Para pengguna sistem dan staf yang mengelola basis data memerlukan dokumentasi mengenai prosedur tentang cara menggunakan atau menjalankan sistem. Instruksi-instruksi yang biasa dibutuhkan oleh orang yang mengelola basis data ataupun pengguna yang menggunakan basis data meliputi:

- a. Cara masuk ke DBMS dan menggunakan fasilitas DBMS atau program aplikasi tertentu.
- b. Memulai dan menghentikan DBMS dan membuat salinan cadangan dari basis data.
- c. Menangani kegagalan atau kerusakan pada perangkat keras dan perangkat lunak.
- d. Mengubah struktur tabel, menata basis data di beberapa tempat penyimpanan, meningkatkan kinerja, atau arsip data ke penyimpanan sekunder.

### 5. Manusia (*People*)

Komponen terakhir yang terlibat dengan sistem DBMS adalah manusia. Ada 4 macam komponen manusia yang terlibat dalam lingkungan DBMS yaitu diantaranya [28]:

#### a. Pengelola Data dan Pengelola Basis Data

Pengelola data dan pengelola basis data umumnya berperan terkait dengan pengelolaan dan pengendalian DBMS dan datanya. Pengelola data adalah orang yang bertanggung jawab atas pengelolaan sumber

daya data yang termasuk perencanaan basis data, pengembangan dan pemeliharaan standar, kebijakan dan prosedur, dan perancangan basis data secara konseptual/logika, sedangkan pengelola basis data bertanggung jawab untuk realisasi basis data, termasuk perancangan basis data secara fisik dan implementasinya, keamanan dan pengaturan integritas, pemeliharaan sistem operasional, dan memastikan kepuasan pengguna terhadap kinerja aplikasi.

b. Perancang Basis Data

Dalam proyek-proyek perancangan basis data yang besar, berfungsi untuk membedakan antara dua jenis perancang yaitu perancang basis data *logical* dan perancang basis data fisik. Perancang basis data logikal bekerja dengan hal-hal yang berkaitan dengan mengidentifikasi data (yaitu, entitas dan atribut), hubungan antara data, dan kendala pada data yang akan disimpan dalam basis data.

c. Pengembang Aplikasi (*Application Developer*)

Pengembang aplikasi bertugas untuk mengembangkan aplikasi yang menyediakan fungsionalitas yang dibutuhkan oleh pengguna. Biasanya para pengembang aplikasi bekerja sesuai spesifikasi yang dihasilkan oleh sistem analis. Program aplikasi yang dikembangkan berisi untuk melakukan beberapa operasi pada basis data seperti mengambil data, memasukkan, memperbarui, dan menghapus data.

d. *End User*

*End user* adalah pengguna dari basis data yang dimana basis data tersebut telah dirancang, diimplementasikan, dipelihara untuk menyajikan informasi yang mereka butuhkan [23].

## 2.7 PHP

Kode PHP pada umumnya diproses, dan diterjemahkan menggunakan web server dengan sebuah modul PHP yang sudah terpasang pada server tersebut, sehingga memungkinkan PHP untuk ditanamkan pada sebuah dokumen *markup* HTML berektensi “.php”. PHP digunakan untuk menangani pemrosesan data yang

kompleks agar data dinamis dapat muncul di halaman web, misalnya seperti kalkulasi matematis, dan interaksi dengan *database*. PHP membuat *developer* dapat menjadikan HTML yang awalnya hanya berisi konten statis menjadi sebuah halaman yang responsif terhadap permintaan pengguna [30].

## 2.8 Kajian Pustaka

Penelitian sebelumnya dilakukan oleh Sugeng Wahono, pada tahun 2014 dengan judul eksperimen pengaturan suhu dan kelembaban pada rumah tanaman (*greenhouse*) dengan sistem humidifikasi. Berdasarkan ilmu pengkondisian udara hal tersebut dapat dibuat kondisi seperti musim pada tanaman tertentu sesuai yang diharapkan, yaitu dengan mengatur besarnya kelembaban dan suhunya. Hasil penelitian tersebut ialah menampilkan informasi suhu, kelembaban, jam dan tanggal terkini. Ada unsur humidifikasi yang kurang maksimal. Atap berpengaruh terhadap besarnya perubahan nilai *temperature* dan kelembaban pada *greenhouse* [1].

Penelitian sebelumnya dilakukan oleh Ramdani Marnisa, pada tahun 2015 dengan judul sistem *monitoring* tanaman menggunakan zigbee dan platform M2M. Penelitian ini membahas mengenai sistem *monitoring* jarak jauh. Beberapa contoh penelitian tentang sistem *monitoring* jarak jauh sebelumnya adalah sistem *monitoring* tanaman bayam yang memanfaatkan Zigbee sebagai standar komunikasi pengiriman data. Namun jarak atau *range* kerja dari ZigBee sekitar 76 meter. Data yang diterima *gateway* diteruskan ke server menggunakan platform M2M. Penelitian ini hanya meneliti beberapa parameter saja yaitu kelembaban tanah, sensor cahaya dan sensor ketinggian serta perlu adanya penambahan parameter untuk perkembangan tanaman bayam. Penelitian *monitoring* tanaman ini dapat dikembangkan menjadi aplikasi *mobile* [3].

Penelitian sebelumnya dilakukan oleh Syarief Syafrizal, pada tahun 2016 dengan judul sistem *monitoring* suhu dan kelembaban tanaman cabai pada *greenhouse* berbasis labview. Penelitian ini membahas tentang sistem *monitoring* pada tanaman cabai dengan dengan parameter suhu dan kelembaban berbasis LabVIEW. Pada *greenhouse* digunakan beberapa sensor seperti sensor DHT11

yang berfungsi sebagai pendeteksi suhu dan kelembaban lingkungan, sensor *soil moisture* sebagai pendeteksi kelembaban tanah dan penambahan sensor LDR sebagai pendeteksi intensitas cahaya, serta terdapat data *logger* yang menyimpan data hasil *monitoring* secara berkala sebagai bahan evaluasi. Hasil *monitoring* suhu dan kelembaban masih menggunakan layar LCD 4x20 karakter dan *front panel* LabVIEW 2013 [7].

Penelitian sebelumnya dilakukan oleh Caesar Pats Yahwe, pada tahun 2016 dengan judul rancang bangun *prototype system monitoring* kelembaban tanah melalui SMS berdasarkan hasil penyiraman tanaman. Penelitian ini membahas mengenai tentang sistem *monitoring* pada tanaman cabai dan tomat berdasarkan hasil penyiraman tanaman. *Monitoring* penyiraman tanaman berdasarkan kelembaban tanah melalui SMS berbasis *mikrokontroler* merupakan salah satu cara yang dapat digunakan dalam hal mengawasi serta merawat tanaman tetap dalam kondisi yang baik. Penelitian ini kurang maksimal karena hanya membahas kelembaban tanah saja, dan tergantung pada kondisi jaringan seluler. Jaringan seluler yang kurang baik dapat menghambat penerimaan SMS [8].

Penelitian sebelumnya dilakukan oleh Amanda Fahmi Ma'arif, pada tahun 2016 dengan judul sistem *monitoring* dan *controlling* air nutrisi *aquaponik* menggunakan Arduino uno berbasis web server. Penelitian ini berfokus pada bagaimana membangun sistem *monitoring* dan *controlling* air nutrisi *aquaponik* (PH dan EC) menggunakan arduino uno. Dengan sistem *aquaponik* ini dapat membantu *user* untuk *monitoring* PH dan EC dari jarak jauh sekaligus memberikan batasan agar kondisi air pada *aquaponik* tetap stabil. Untuk pengembangan sistem *monitoring* dan *controlling* air nutrisi *aquaponik* lebih lanjut dapat dikembangkan dalam lingkungan yang lebih luas agar menghasilkan produk dalam jumlah yang lebih besar. Tampilan hasil *monitoring* masih dalam bentuk tampilan *website* [9].

Penelitian sebelumnya dilakukan oleh Atmoko Andri, pada tahun 2013 dengan judul sistem *monitoring* dan pengendalian suhu dan kelembaban ruang pada rumah walet berbasis android, web, dan SMS. Penelitian ini menghasilkan respon kecepatan pengendalian melalui perangkat berbasis web selama 2 detik, melalui perangkat berbasis android selama 2 detik, dan melalui perangkat berbasis SMS

selama 15 detik. *Monitoring* data suhu dan kelembaban dapat dipantau melalui web secara *realtime* dengan selang waktu selama 5 detik. Sistem yang dibangun bermanfaat untuk menggantikan proses pengecekan suhu dan kelembaban yang masih dilakukan secara manual [10].

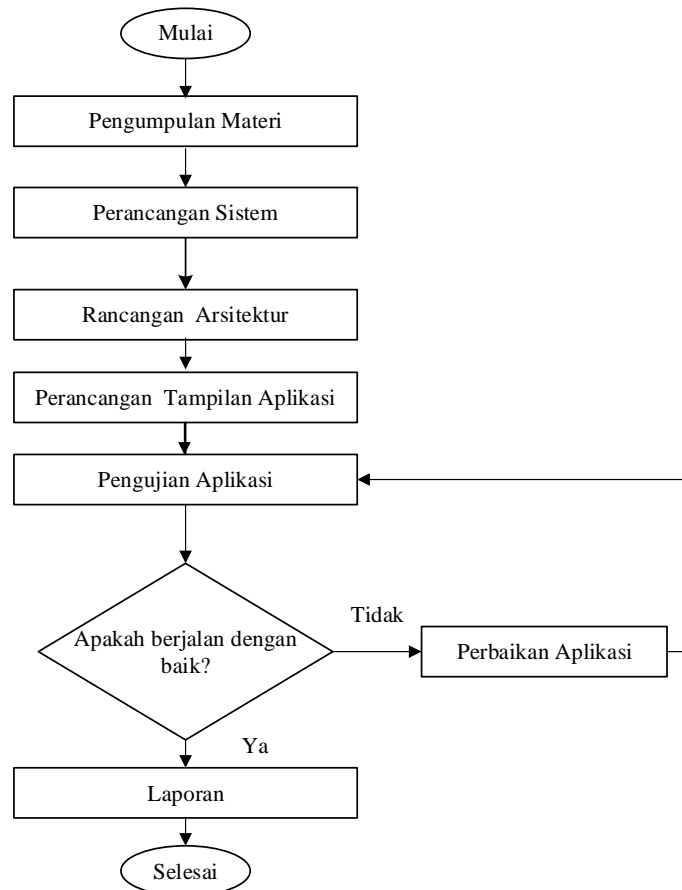
Penelitian selanjutnya dilakukan oleh Saputrra Apri, pada tahun 2012 dengan judul aplikasi *monitoring* hasil produksi dan pengingat jadwal perawatan tanaman kelapa sawit dengan perangkat *mobile* berbasis android. Penelitian ini menghasilkan sebuah aplikasi yang berfungsi untuk memudahkan *user* (pengusaha sawit) dalam memasukan dan menyimpan data produksi panen serta melihat perkembangan hasil produksi panen berupa tampilan grafik serta membantu *user* dalam mengingat jadwal perawatan sawit dan memberikan informasi tentang cara penanaman dan perawatan kelapa sawit [6].

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metode Penelitian

Bab ini menjelaskan tahap-tahap yang dilakukan dalam pelaksanaan penelitian. Pembahasan skripsi mengenai sistem *monitoring* suhu, kelembaban, intensitas cahaya serta kadar karbondioksida berbasis android pada tanaman aeroponik. Pada penelitian ini hanya difokuskan pada tanaman bayam karena memiliki masa panen yang singkat. Sistem *monitoring* diterapkan pada tanaman bayam dengan metode penanaman secara aeroponik dengan parameter suhu, kelembaban, intensitas cahaya serta kadar CO<sub>2</sub>. *Flowchart* dari penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* Penelitian

Pada Gambar 3.1 menjelaskan secara garis besar tentang tahapan yang akan dilakukan pada penelitian ini. Tahapan pertama yaitu pengumpulan materi mengenai dasar tanaman aeroponik terutama pada tanaman bayam, pertumbuhan tanaman bayam dalam keadaan baik pada keadaan suhu, kelembaban, intensitas cahaya serta kadar karbondioksida, teori pemrograman yang akan digunakan. Tahapan kedua yaitu memahami alur dari sistem tersebut berjalan. Tahapan ketiga yaitu merancang arsitektur dari *database* agar terhubung dengan aplikasi android.

Tahapan keempat yaitu merancang tampilan aplikasi, agar aplikasi berjalan dengan baik, yang berfokus pada struktur data, arsitektur *software*, GUI serta detail algoritma. Selain itu pemrograman atau *coding* merupakan hal paling terpenting karena menerjemahkan dari desain ke dalam bentuk mesin yang dapat dibaca. Jika desain dilakukan secara detail, pemrograman dapat diselesaikan secara mekanis. Tahap selanjutnya adalah tahap pengujian aplikasi. Tahapan ini untuk memastikan bahwa semua pernyataan sudah diuji dan pengujian pada eksternal fungsional untuk menemukan kesalahan-kesalahan sehingga aplikasi dapat diakses oleh semua *user*. Tahapan terakhir yakni menulis laporan sehingga dapat menarik kesimpulan dari penelitian ini.

## **3.2 Instrumen Penelitian**

Pada penelitian sistem *monitoring* berbasis android dibutuhkan beberapa instrumen untuk mempermudah kinerja proses pengerjaannya. Instrumen yang dibutuhkan berupa *software* dan *hardware*.

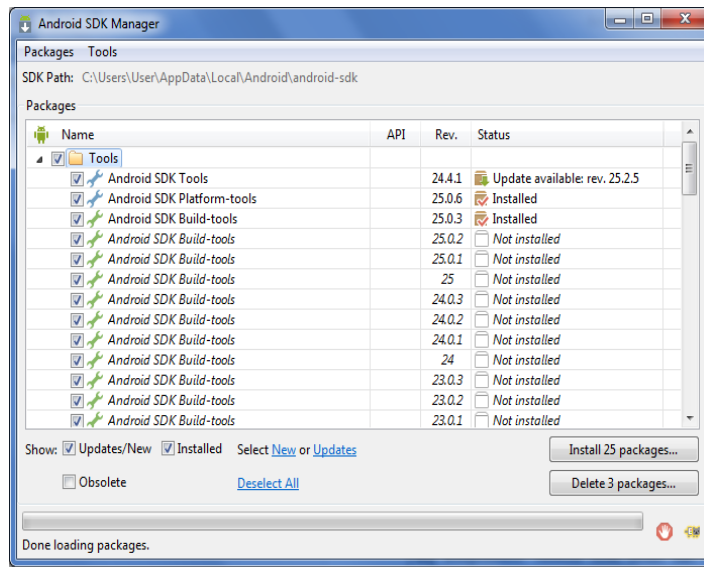
### **3.2.1 Software**

Pada pembuatan sistem aplikasi *monitoring* tanaman aeroponik ini membutuhkan beberapa *software* di antaranya:

#### **1. Android SDK Tool**

Android-SDK merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis *google* android. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial. Tampilan android SDK *tool* dapat dilihat pada Gambar 3.2.

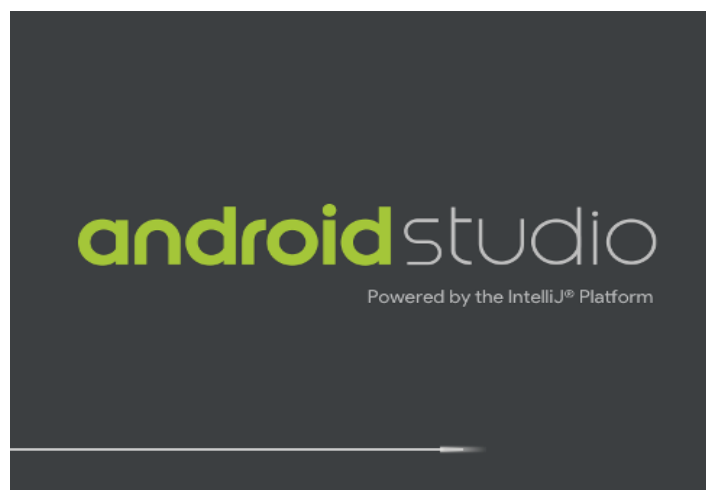




Gambar 3.2 Android SDK Manager Tool

## 2. Android Studio

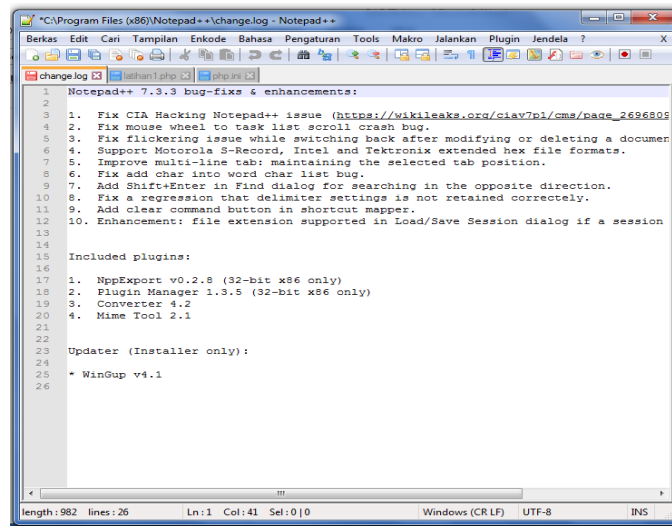
Android studio merupakan *software* yang digunakan untuk pengembangan aplikasi android dengan berbagai fitur yang lebih banyak. Setiap *project* dalam Android Studio berisi satu atau beberapa modul dengan *file* kode sumber dan *file* sumber daya. Tampilan android studio dapat dilihat pada Gambar 3.3.



Gambar 3.3 Android Studio

### 3. Text Editor

*Text editor* dapat digunakan untuk membuat program, mengedit *source code* bahasa pemrograman php dan java Pada penelitian ini menggunakan *notepad++*. Tampilan *notepad++* dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Notepad++*

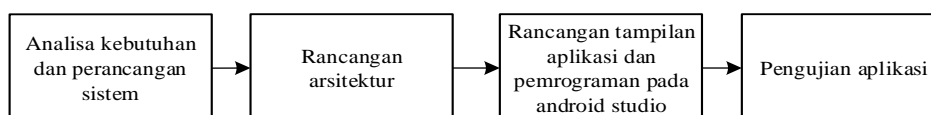
#### 3.2.2 Hardware

Dalam pembuatan aplikasi sistem *monitoring* ini membutuhkan perangkat keras (hardware) berupa PC dengan spesifikasi sebagai berikut:

1. Intel(R) Core(TM) i5-7200U CPU @2.50 GHz
2. RAM 4 GB
3. Hard Disk 1T
4. *Smartphone* dengan spesifikasi perangkat keras:
  - a. Versi Android : 6.0
  - b. CPU : 4 core 1,0 GHz
  - c. RAM : 2GB
  - d. Memori Telepon : 16GB
  - e. Ukuran Layar : 720 x 1280 pixel

### 3.3 Perancangan Penelitian

Perancangan penelitian sistem *monitoring* ini terbagi menjadi 4 tahap yaitu, perancangan sistem, perancangan arsitektur (penghubung antara *database* dengan android), perancangan tampilan aplikasi dan pemrograman pada android studio, serta pengujian aplikasi terhadap beberapa *smartphone*. Diagram alir perancangan penelitian dapat dilihat pada Gambar 3.5.

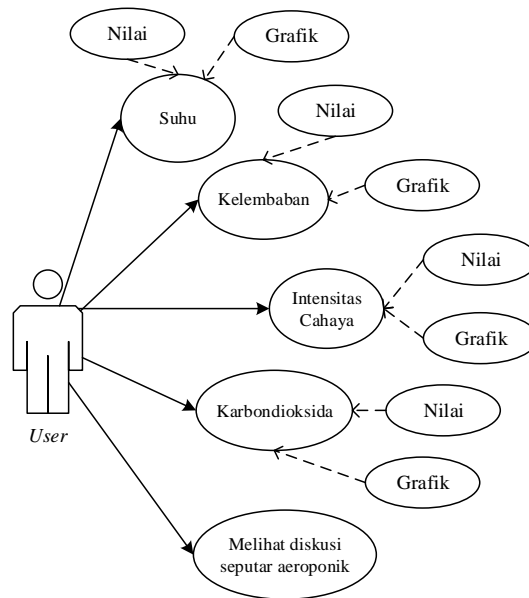


Gambar 3.5 Blok Diagram Perancangan Penelitian

Gambar 3.5 merupakan penjelasan secara singkat mengenai tahapan yang akan digunakan dalam penelitian ini. Tahap pertama dimulai dengan tahap analisa serta implementasi sistem. Tahap kedua merupakan tahap untuk menghubungkan *client* dengan server. Dalam proses menghubungkan *database* ke aplikasi sistem *monitoring* menggunakan format JSON (*JavaScript Object Notation*). Tahap ketiga membahas rancangan tampilan aplikasi dan pemrograman pada software android studio. Tahap keempat yaitu. Tahap terakhir yaitu pengujian terhadap sistem kerja aplikasi, apakah sistem berjalan sesuai rancangan atau tidak serta pada berbagai macam *smartphone*.

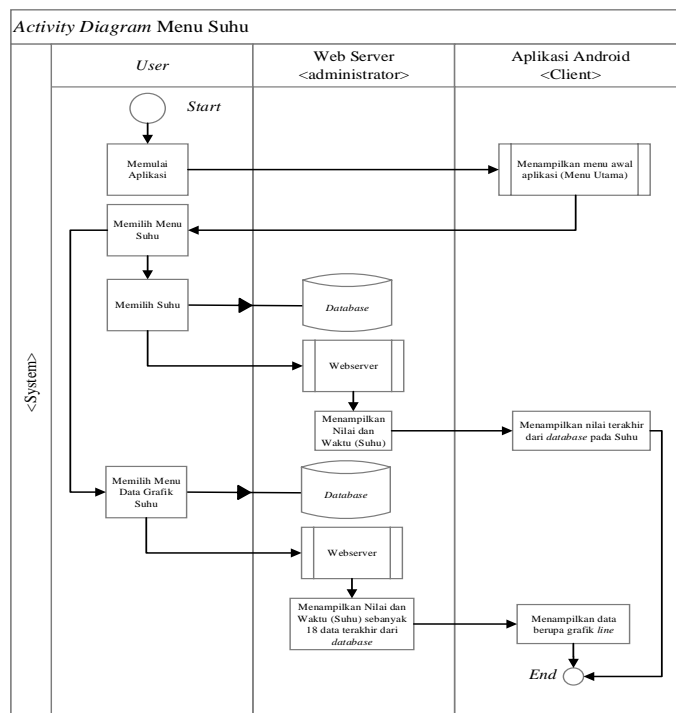
#### 3.3.1 Perancangan Sistem

*Use case diagram* aplikasi sistem *monitoring* pada Gambar 3.6 menggambarkan fungsi yang bisa dilakukan oleh *user*. Fungsi tersebut bertujuan untuk melihat nilai dan grafik *line* pada parameter suhu, kelembaban, intensitas cahaya, karbondioksida, diskusi berupa beberapa pertanyaan seputar aeroponik.



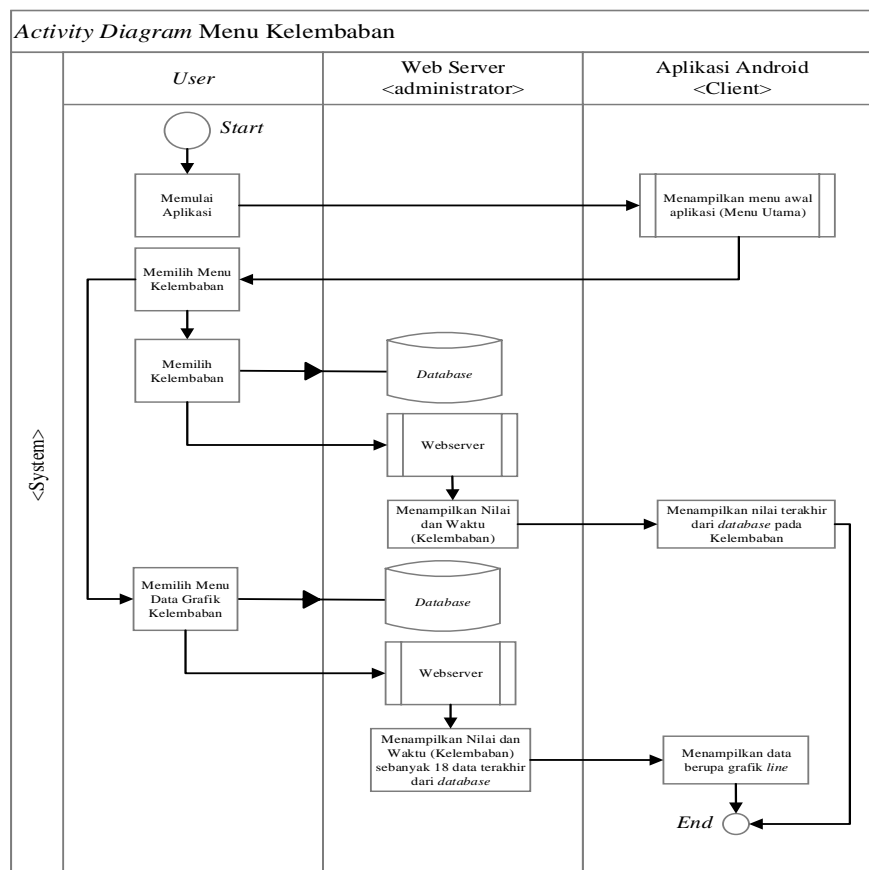
Gambar 3.6 Use Case Diagram Aplikasi Sistem Monitoring

Untuk *activity diagram* aplikasi sistem *monitoring* dibagi menjadi lima, yaitu menu parameter suhu, kelembaban, intensitas cahaya, karbondioksida, dan diskusi aeroponik. *Use case diagram* untuk menu parameter suhu dijelaskan pada Gambar 3.7.



Gambar 3.7 Activity Diagram Menu Suhu

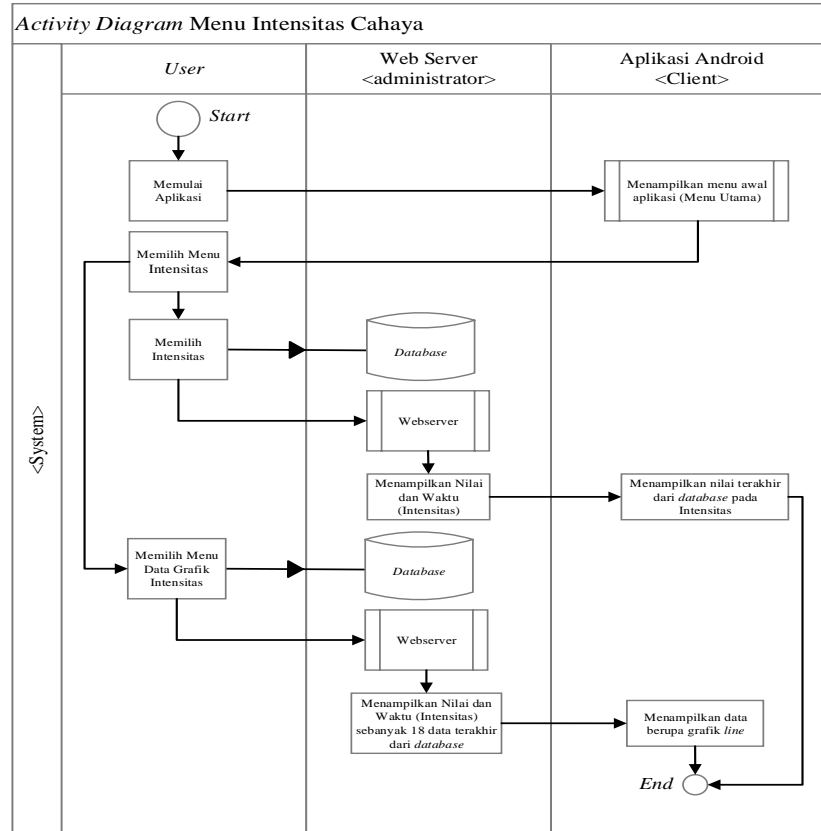
*Activity diagram* pada Gambar 3.7 menjelaskan mengenai kinerja sistem dari menu parameter suhu. *User* memilih menu parameter suhu maka sistem akan menampilkan menu parameter suhu dengan 2 pilihan yaitu suhu menampilkan nilai terakhir dari parameter, dan data grafik suhu menampilkan menu data grafik suhu sebanyak 18 data terakhir yang tersimpan pada *database*. *Use case diagram* yang kedua merupakan gambaran mengenai penjelasan dari kinerja sistem menu kelembaban secara singkat dan dijelaskan pada Gambar 3.8.



Gambar 3.8 Activity Diagram Menu Kelembaban

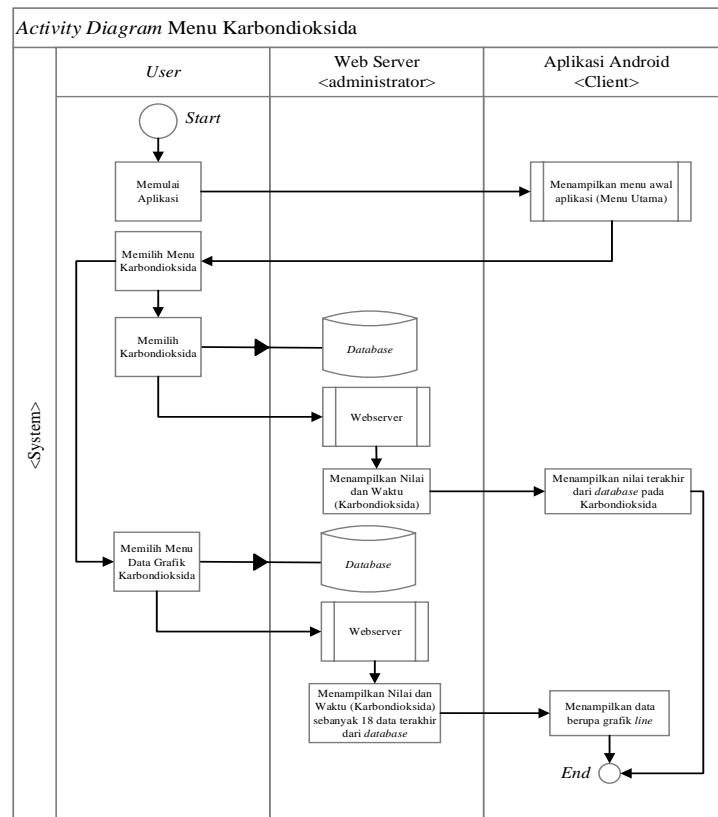
*Activity diagram* pada Gambar 3.8 menjelaskan mengenai kinerja sistem dari menu parameter kelembaban. *User* memilih menu parameter kelembaban maka sistem akan menampilkan menu parameter kelembaban dengan 2 pilihan yaitu kelembaban menampilkan nilai terakhir dari parameter, dan data grafik kelembaban menampilkan menu data grafik kelembaban sebanyak 18 data terakhir yang tersimpan pada *database*. *Use case diagram* yang ketiga merupakan gambaran

mengenai penjelasan dari kinerja sistem menu intensitas cahaya secara singkat dan dijelaskan pada Gambar 3.9.



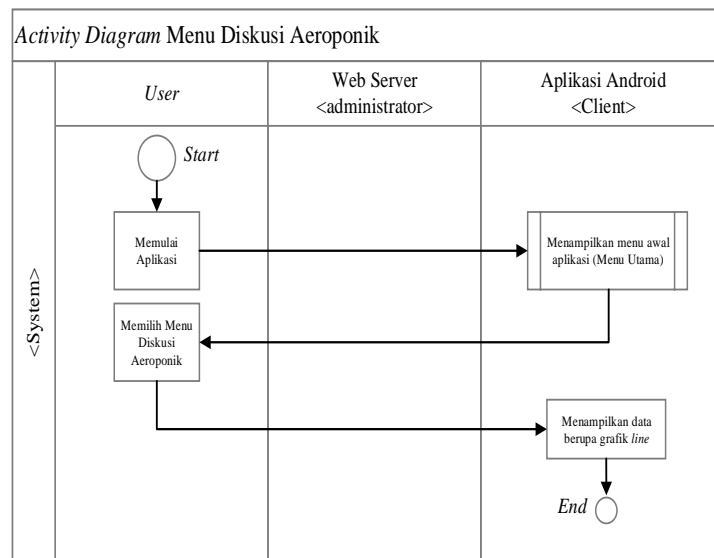
Gambar 3.9 Activity Diagram Menu Intensitas Cahaya

*Activity diagram* pada Gambar 3.9 menjelaskan mengenai kinerja sistem dari menu parameter intensitas cahaya. *User* memilih menu parameter intensitas cahaya maka sistem akan menampilkan menu parameter intensitas cahaya dengan 2 pilihan yaitu intensitas cahaya, dan data grafik intensitas cahaya. Jika menu intensitas cahaya yang dipilih, maka akan menampilkan nilai terakhir dari parameter sehingga *user* dapat melihat nilai parameter tersebut, dan apabila *user* memilih data grafik maka akan menampilkan menu data grafik intensitas cahaya sebanyak 18 data terakhir yang tersimpan pada *database*. *Use case diagram* yang keempat merupakan gambaran mengenai penjelasan dari kinerja sistem menu karbondioksida secara singkat dan dijelaskan pada Gambar 3.10.



Gambar 3.10 Activity Diagram Menu Karbondioksida

*Activity diagram* pada Gambar 3.10 menjelaskan mengenai kinerja sistem dari menu parameter karbondioksida. *User* memilih menu parameter karbondioksida maka sistem akan menampilkan menu parameter karbondioksida dengan 2 pilihan yaitu karbondioksida, dan data grafik karbondioksida. Jika memilih menu karbondioksida, maka akan menampilkan nilai terakhir dari parameter sehingga *user* dapat melihat nilai parameter tersebut, dan apabila *user* memilih data grafik maka akan menampilkan menu data grafik karbondioksida sebanyak 18 data terakhir yang tersimpan pada *database*. *Use case diagram* yang kelima merupakan gambaran mengenai penjelasan dari kinerja sistem menu diskusi aeroponik secara singkat dan dijelaskan pada Gambar 3.11.

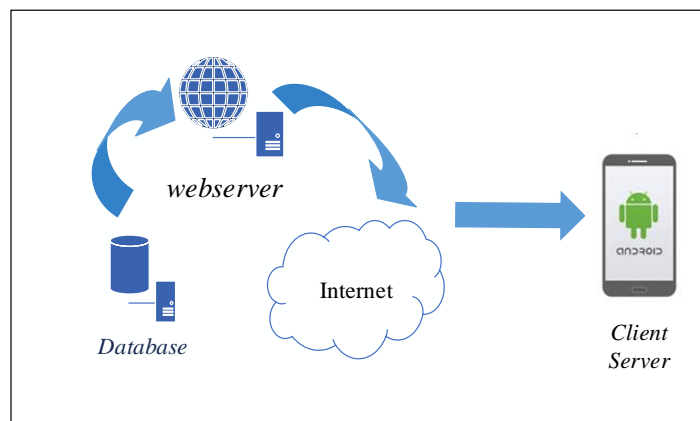


Gambar 3.11 Activity Diagram Menu Diskusi Aeroponik

Gambar 3.11 merupakan *activity diagram* dari menu diskusi aeroponik. Ketika *user* memilih menu tersebut maka sistem akan menampilkan menu diskusi yang berisi pertanyaan dan jawaban seputar aeroponik. Sehingga *user* dapat melihat tampilan dari menu tersebut.

### 3.3.2 Perancangan Arsitektur

Perancangan arsitektur pada penelitian aplikasi sistem *monitoring* ini bertujuan untuk menghubungkan *client* dengan server melalui internet. *Client* akan melakukan *request server* untuk mendapatkan informasi mengenai kondisi terakhir parameter. Penggambaran dari alur perancangan arsitektur dapat dilihat pada Gambar 3.12.



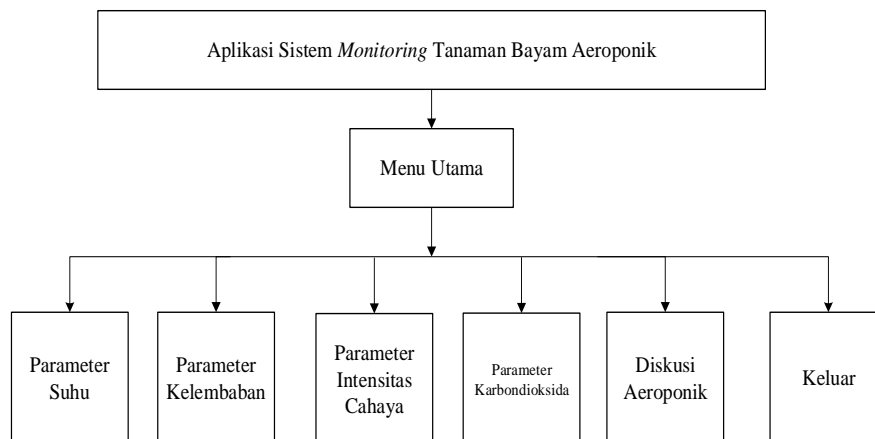
Gambar 3.12 Rancangan Arsitektur Aplikasi Android



Gambar 3.12 merupakan rancangan arsitektur dari aplikasi sistem *monitoring* tanaman aeroponik. Yang terdiri dari *database*, web server, internet dan *client server* yaitu *smartphone*. *Database* dibuat menggunakan MySQL, yang terhubung dengan web server. Agar saling terhubung antara *database*, web server, dan *smartphone* diperlukan jaringan internet.

### 3.3.3 Perancangan Tampilan Aplikasi dan Pemrograman Android Studio

Memodelkan suatu sistem merupakan hal yang paling utama. Banyak cara yang dipakai dalam memodelkan suatu sistem. Namun pada dasarnya pemodelan merupakan kombinasi antara *software* dengan *hardware* yang digunakan. Perancangan ini dimulai dengan membuat diagram blok dari sketsa tampilan dari isi aplikasi (gambaran isi dari aplikasi). Adapun diagram blok dapat dilihat pada Gambar 3.13.



Gambar 3.13 Diagram Blok Aplikasi Sistem *Monitoring*

Gambar 3.13 menjelaskan mengenai gambaran dari isi aplikasi sistem *monitoring* tanaman aeroponik yang terdiri dari menu utama yang berisi parameter suhu, kelembaban, intensitas cahaya, serta kadar CO<sub>2</sub>, diskusi aeroponik (berisi pertanyaan dan jawaban seputar aeroponik), dan keluar (keluar dari aplikasi). Gambar 3.13 dapat berjalan dengan adanya pemrograman pada *software* android studio. Tahap pemrograman pada android studio sangatlah penting dalam pembuatan aplikasi sistem *monitoring* aeromosis (*aerophonic monitoring system*) ini. Pada android studio terdapat dua bahasa pemrograman yaitu xml untuk membuat tampilan aplikasi dan java untuk menjalankan tampilan aplikasi agar dapat berjalan

sesuai dengan perancangan. Berikut langkah-langkah dalam pemrograman pada android studio:

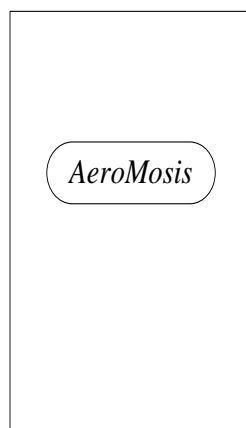
### 1. Pengumpulan Gambar

Langkah pertama yang harus dilakukan pada pemrograman adalah mengumpulkan beberapa gambar untuk membuat tampilan aplikasi aeromosis menjadi lebih menarik. Gambar yang telah dikumpulkan, kemudian dipindahkan kedalam *folder drawable*. Pemberian nama untuk gambar haruslah menggunakan huruf kecil. Jika menggunakan huruf kecil akan mengganggu jalannya aplikasi, selain pemberian nama yang harus menggunakan huruf kecil, ukuran dari gambar juga tidak boleh terlalu besar.

### 2. Menu Utama

Sebelum masuk kedalam menu utama, ketika membuka aplikasi akan masuk kedalam menu *splashscreen* terlebih dahulu. Menu *splashscreen* berisi logo yang menggambarkan aplikasi dengan tulisan aeromosis (*aerophonic monitoring system*). *Splashscreen* berdurasi 5 detik dan secara otomatis akan menuju kedalam menu utama. *Script/code* dibawah ini akan menampilkan menu *splashscreen* dengan durasi waktu 5 detik dan kemudian akan menuju menu utama secara otomatis.

```
{
public void run(){
try{
sleep(2000);
}catch (InterruptedException e){
e.printStackTrace();
}finally {
startActivity(new Intent(Splashscreen.this, MenuParameter.class));
finish();
}
};
thread.start();
}
```



Gambar 3.14 Tampilan *Splashscreen* Aplikasi

Gambar 3.14 merupakan tampilan awal pada aplikasi sistem *monitoring* tanaman aeroponik pada bayam dengan menerapkan *script* diatas, ketika aplikasi dijalankan, menu awal akan muncul berupa *splashscreen* dengan durasi 5 detik, kemudian secara otomatis akan menuju ke menu utama.

Menu utama terdiri dari *imageslider* dan beberapa menu pilihan. *Imageslider* memuat beberapa animasi yang dapat bergeser secara otomatis dengan *delay* setiap dua detik dan tiga detik untuk pindah dari gambar satu ke yang lain. Setelah gambar yang kelima secara otomatis akan berpindah ke gambar pertama. Berikut *script* untuk menjalankan menu *imageslider*.

```
public class CustomSwipeAdapter2 extends PagerAdapter {
    private int[] image_resource
    ={R.drawable.satu,R.drawable.two,R.drawable.three,R.drawable.empat
    ,R.drawable.lima};
    private Context ctx;
    private LayoutInflater inflater;

    public CustomSwipeAdapter2(Context ctx)
    {
        this.ctx= ctx;
    }
    @Override
    public int getCount() {

        return image_resource.length;
    }
}
```

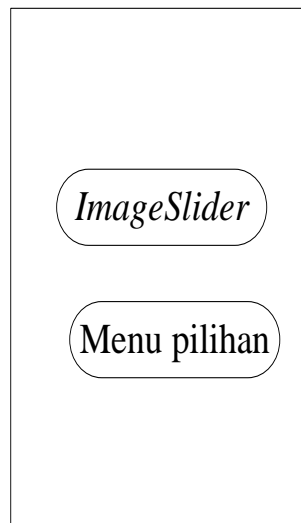
Selain *imageslider*, menu utama juga menampilkan beberapa menu pilihan diantaranya menu suhu, kelembaban, intensitas cahaya, karbondioksida, menu diskusi aeroponik dan menu keluar.

```

pindah = (ImageButton) findViewById(R.id.buttonsuhu);
pindah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MenuParameter.this,
MenuSuhu.class);
        startActivity(intent);
    }
});
pindah = (ImageButton) findViewById(R.id.butтонаeroponik);
pindah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MenuParameter.this,
MainQuestion.class);
        startActivity(intent);
    }
});

```

*Script/code* diatas akan menampilkan menu utama seperti pada Gambar 3.15.



Gambar 3.15 Menu Utama Aplikasi

### 3. Menu Parameter

Menu parameter ini mencakup menu suhu, kelembaban, intensitas cahaya, serta karbondioksida. Pada keempat menu parameter ini memiliki kesamaan dalam *activity*, yaitu terdiri dari dua menu pilihan. Menu pilihan tersebut ialah menu nilai parameter yang berisi nilai dan tanggal update terakhir serta menu data grafik yang berisi 18 data terakhir dari *database*. Grafik yang dimunculkan berupa grafik *line*. Perbandingan antara nilai dengan waktu untuk setiap parameter. Menghubungkan

data dari *database* dengan tampilan pada aplikasi, baik untuk nilai maupun data grafik dengan memasukkan *url* dari *database*.

```
public class ServerAPI {
    public static final String URL_DATASUHU =
    "http://verapuspita.000webhostapp.com/monitoring/current_suhu.php";
    public static final String URL_GRAFIK_SUHU =
    "http://verapuspita.000webhostapp.com/monitoring/get_datasuhu.php";

    public static final String URL_DATAKELEMBABAN =
    "http://verapuspita.000webhostapp.com/monitoring/current_kelembaban
    .php";
    public static final String URL_GRAFIK_KELEMBABAN =
    "http://verapuspita.000webhostapp.com/monitoring/get_datakelembaban
    .php";

    public static final String URL_DATAINTENSITAS =
    "http://verapuspita.000webhostapp.com/monitoring/current_intensitas
    .php";
}
```

Pengambilan data untuk nilai dan data grafik adalah dengan menggunakan *function* “mRequestQueue” pada *file* Java AppController.

```
public static synchronized AppController getInstance()
{return instance;}
private RequestQueue getRequestQueue()
{if (mRequestQueue == null)
    {mRequestQueue = Volley.newRequestQueue(getApplicationContext());}
return mRequestQueue;}
public <T> void addToRequestQueue(Request<T> req, String tag)
{req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
getRequestQueue().add(req);
}

public <T> void addToRequestQueue (Request<T> req)
{
    req.setTag(TAG);
    getRequestQueue().add(req);
}
public void cancelAllRequest(Object req)
{if (mRequestQueue != null)
    {mRequestQueue.cancelAll(req);}}
```

Pengambilan data untuk semua parameter terdiri dari beberapa tahap dan membutuhkan 2 *layout xml* dan 2 *file java*. *Layout xml* pertama berupa *layout* untuk menampilkan menu nilai dari parameter dan satu *layout xml* untuk menampilkan nilai yang diambil dari *database* dengan memanfaatkan *AdapterView* pada *ListView*. *Layout xml* akan berjalan jika terhubung dengan *file java*.

```

private void loadJson()
{pd.setMessage("Mengambil Data");
  pd.setCancelable(false);
  pd.show();
  JSONArrayRequest reqData = new
  JSONArrayRequest(Request.Method.POST,
  ServerAPI.URL_DATASUHU,null,
  new Response.Listener<JSONArray>() {
  @Override
  public void onResponse(JSONArray response) {
  pd.cancel();
  Log.d("volley","response : " + response.toString());
  for(int i = 0 ; i < response.length(); i++)
  {try {JSONObject data = response.getJSONObject(i);
  Suhu md = new Suhu();
  md.setNilai(data.getString("nilai" ));

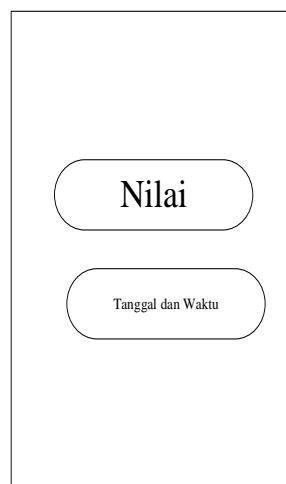
```

```

@Override
public HolderData onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
  View layout =
  LayoutInflater.from(parent.getContext()).inflate(R.layout.parameter_su
  hu,parent,false);
  HolderData holderData = new HolderData(layout);
  return holderData;}
@Override
public void onBindViewHolder(@NonNull HolderData holder, int position)
{
  Suhu md = mItems.get(position);
  holder.t_Suhu.setText(md.getNilai());
  holder.t_Waktu.setText(md.getWaktu());
  holder.md = md;}

```

Dua *script* diatas akan menampilkan tampilan nilai parameter seperti pada Gambar 3.16. Tampilan terdiri dari nilai dan waktu *update* terakhir yang termuat dalam *database*.



Gambar 3.16 Tampilan Nilai Parameter

Pengambilan data grafik juga membutuhkan beberapa tahap sehingga dapat menampilkan nilai sebanyak 18 data terakhir dari *database* dalam bentuk grafik *line*. Pengambilan data grafik membutuhkan 1 *layout xml* dan 2 *file java*. *Layout xml* berfungsi untuk menampilkan data grafik, 2 *file java* berfungsi untuk pengambilan data grafik.

```

public void load_data_from_server(){
    pd.show();
    String url = ServerAPI.URL_GRAFIK_SUHU;
    xAxis1 = new ArrayList<>();
    yAxis = null;
    yValues = new ArrayList<>();
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
url, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("String", response);
            try{
                JSONArray jsonArray = new JSONArray(response);
                for (int i=0; i<jsonArray.length(); i++){
                    JSONObject jsonObject = jsonArray.getJSONObject(i);
                    String suhu = jsonObject.getString("nilai").trim();
                    String waktu =
jsonObject.getString("waktu").trim();
                    xAxis1.add(waktu);
                    values = new Entry(Float.valueOf(suhu), i);
                    yValues.add(values);}
            }catch (JSONException e){
                e.printStackTrace();}
            LineDataSet lineDataSet = new LineDataSet(yValues, "Grafik
Suhu (18 data terakhir), \n\n" +
                "x: Waktu, \n\n" +
                "y: Nilai Suhu dalam satuan °C");
            lineDataSet.setColor(Color.rgb(0, 82, 159));
            yAxis = new ArrayList<>();
            yAxis.add(lineDataSet);
            String names[] = xAxis1.toArray(new String[xAxis1.size()]);
            data = new LineData(xAxis1, lineDataSet);
            chart.setData(data);
            chart.setDescription("");
            chart.animateXY(2000, 2000);
            chart.invalidate();
            pd.hide();}},
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    if(error != null){
                        Toast.makeText(getApplicationContext(), "Ups, terjadi kesalahan.",
Toast.LENGTH_LONG).show();
                        pd.hide();}}});

MySingletonsuhu.getInstance(getApplicationContext()).addToRequestQueue(
stringRequest);

```

```

private MySingletonsuhu(Context context) {
    mContext = context;
    mRequestQueue = getRequestQueue();

    mImageLoader = new ImageLoader(mRequestQueue,
        new ImageLoader.ImageCache() {
            private final LruCache<String, Bitmap>
                cache = new LruCache<String, Bitmap>(20);

            @Override
            public Bitmap getBitmap(String url) {
                return cache.get(url);
            }

            @Override
            public void putBitmap(String url, Bitmap bitmap) {
                cache.put(url, bitmap);
            }
        });
}

public static synchronized MySingletonsuhu getInstance(Context
context) {
    if (mInstance == null) {
        mInstance = new MySingletonsuhu(context);
    }
    return mInstance;
}

public RequestQueue getRequestQueue() {
    if (mRequestQueue == null) {

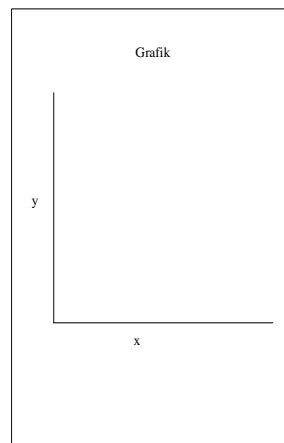
        mRequestQueue =
Volley.newRequestQueue(mContext.getApplicationContext());
    }
    return mRequestQueue;
}

public <T> void addToRequestQueue(Request<T> req) {
    getRequestQueue().add(req);
}

```

Dua *script/code* diatas berfungsi untuk menampilkan data grafik sebanyak 18 data terakhir yang termuat pada *database*. Data grafik yang ditampilkan berupa grafik *line* dengan *range* setiap 4 jam sekali dapat dilihat pada Gambar 3.17.





Gambar 3.17 Tampilan Submenu Data Grafik

#### 4. Menu Diskusi Aeroponik

Menu diskusi aeroponik berisi pertanyaan dan jawaban mengenai sistem seroponik. Dalam menu ini hanya terdiri dari 5 pertanyaan dan jawaban. Penambahan pertanyaan dan jawaban bisa dilakukan dengan cara menambahkan pada `activity_mainquestion.xml` seperti pada Gambar 3.18. Cara menambahkan pertanyaan dan jawaban dengan cara memasukan function “`<TextView>`”, dan diletakan sebelum function “`</LinearLayout>`”.

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Aeroponik merupakan suatu cara
bercocok tanam sayuran diudara tanpa penggunaan tanah,
nutrisi disemprotkan pada akar tanaman, air yang berisi
larutan hara atau nutrisi disemburkan dalam bentuk kabut
hingga mengenai akar tanaman. Akar tanaman yang ditanam
menggantung akan menyerap larutan hara tersebut. Air dan
nutrisi disemprotkan menggunakan irigasi sprinkler.\n\n"
    android:textColor="@color/colorTabStrip"

```

```
package com.example.pita.monitoringsystem.menu;

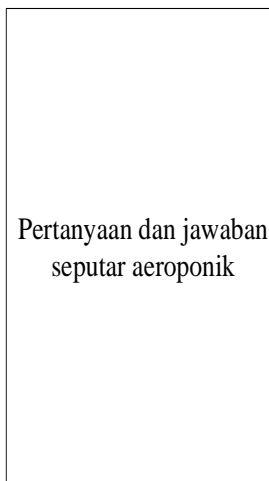
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import com.example.pita.monitoringsystem.R;

public class MainQuestion extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mainquestion);
    }
}
```

*Script/code* diatas akan menampilkan menu seperti pada Gambar 3.18.



Gambar 3.18 Tampilan *Form* Menu Diskusi Aeroponik

### 3.4 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Kendali Jurusan Teknik Elektro dari bulan April 2017 sampai dengan Juli 2018.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Penelitian ini membahas mengenai aplikasi sistem *monitoring* tanaman bayam aeroponik berbasis android dengan parameter suhu, kelembaban, intensitas cahaya serta kadar karbondioksida. Hasil yang diperoleh dari penelitian ini yaitu sistem *monitoring* jarak jauh dengan parameter tersebut dapat ditampilkan dalam aplikasi android, serta dapat melihat hasil *monitoring* sebelumnya yang telah tersimpan ke dalam *database*. Aplikasi android dibuat menggunakan *software* Android Studio versi 3.0.

Semua menu dapat berjalan karena adanya pengkodean (*coding*). Pengkodean (*coding*) merupakan proses penerjemahan desain atau rancangan aplikasi mulai dari menu utama sampai menu yang ada di dalamnya ke dalam bahasa pemrograman java untuk dijadikan sebuah kesatuan agar menjadi media yang dapat digunakan sesuai dengan perancangan. Pengujian (*testing*) sangat penting karena pada tahap ini bertujuan untuk memastikan menu dalam aplikasi dapat berjalan dengan benar dan untuk mengetahui kekurangan atau kesalahan yang harus diperbaiki. Pengujian aplikasi dilakukan menggunakan *smartphone*.

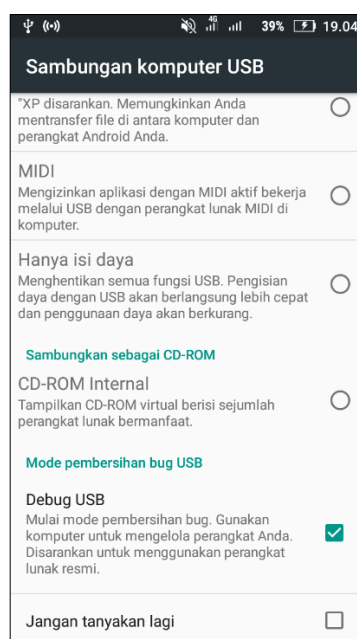
#### **4.1 Menjalankan Aplikasi Aeromosis**

Tahap yang paling penting agar aplikasi aeromosis dapat digunakan adalah tahap menjalankan aplikasi. Pada tahap ini aplikasi dijalankan menggunakan *smartphone* (*USB Debugging*), tahap ini juga merupakan tahap untuk menguji aplikasi dapat berjalan sesuai dengan perancangan. Aplikasi dijalankan pada *smartphone merk* Lenovo A6600d40 dengan sistem operasi android 6.0, API 23 dan ukuran layar 5 inci. Berikut langkah-langkah dalam menjalankan aplikasi menggunakan *smartphone*:

1. Mempersiapkan *smartphone* serta kabel Micro USB to USB. Kemudian membuka *Developer Mode* (Mode Pengembang) pada *smartphone*. Jika Mode Pengembang tidak muncul atau *smartphone* belum pernah digunakan

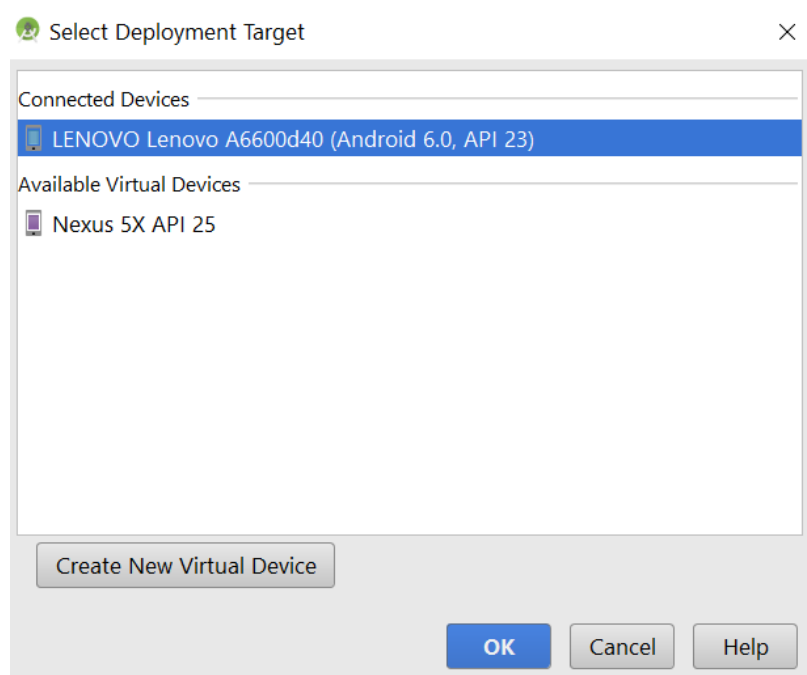
sebagai *emulator* pada *software* android studio, maka *user* harus masuk terlebih dahulu pada menu *settings* (pengaturan) terlebih dahulu, kemudian masuk ke menu *about phone*, pilih *build number* (tekan sebanyak 5 kali) sampai muncul pesan jika *smartphone* sudah menjadi *developer*.

2. Mengaktifkan *USB debugging* pada *smartphone* dengan cara menghubungkan *smartphone* dengan PC menggunakan kabel USB. Mengaktifkan *developer mode* melalui menu *settings*. Cari *USB debugging* dan kemudian aktifkan seperti pada Gambar 4.1.



Gambar 4.1 *USB Debugging*

3. Menjalankan aplikasi dengan cara *Run* pada *software* android studio, dan secara otomatis *smartphone* akan menjadi *emulator* dan muncul pada layar *software* android studio seperti pada Gambar 4.2. Setelah *smartphone* yang kita gunakan muncul kemudian klik OK, dan tunggu hingga proses *compile* selesai. Aplikasi akan muncul pada layar *smartphone*. Gambar 4.2 tidak akan muncul, bila *smartphone* tidak terhubung dengan *software* android studio atau ada kesalahan dalam pengaktifan *smartphone* sebagai *USB debugging*.



Gambar 4.2 *Select Deployment Target*

## 4.2 Pengujian Antarmuka Aplikasi Aeromosis

Aplikasi aeromosis dibuat dengan *software* android studio. *Software* android studio dilengkapi fitur yang lebih lengkap dari eclipse IDE. Dalam pembuatan aplikasi aeromosis *layout xml* dan *java* merupakan dua elemen terpenting yang harus ada. *Layout xml* digunakan untuk merancang tampilan dari aplikasi aeromosis sedangkan *file java* digunakan untuk menghubungkan tampilan *design* aplikasi aeromosis dengan *user* agar aplikasi aeromosis dapat berjalan sesuai yang diharapkan. Aplikasi aeromosis terdiri dari 20 *layout xml* dan 30 *file java*. Daftar *layout* dapat dilihat pada Tabel 4.1 dan daftar *file java* pada Tabel 4.2.

Tabel 4.1 Daftar *Layout xml* Aplikasi Aeromosis

No.	Nama <i>Layout</i>	<i>Role</i>
1.	activity_grafik_intensitas.xml	Menampilkan grafik <i>line</i> pada parameter intensitas
2.	activity_grafik_karbon.xml	Menampilkan grafik <i>line</i> pada parameter karbon
3.	activity_grafik_kelembaban.xml	Menampilkan grafik <i>line</i> pada parameter kelembaban
4.	activity_grafik_suhu.xml	Menampilkan grafik <i>line</i> pada parameter suhu
5.	activity_intensitas_.xml	Menampilkan nilai untuk parameter intensitas
6.	activity_karbon_.xml	Menampilkan nilai untuk parameter karbon
7.	activity_kelembaban_.xml	Menampilkan nilai untuk parameter kelembaban
8.	activity_mainquestion.xml	Menampilkan diskusi aeroponik
9.	activity_menu_intensitas.xml	Menampilkan submenu parameter intensitas
10.	activity_menu_karbon.xml	Menampilkan submen parameter karbon
11.	activity_menukelembaban.xml	Menampilkan submenu parameter kelembaban
12.	activity_menuparameter.xml	Menampilkan menu utama aplikasi
13.	activity_menusuhu.xml	Menampilkan submenu parameter suhu
14.	activity_splashscreen.xml	Menampilkan tampilan <i>splashscreen</i>
15.	activity_suhu_.xml	Menampilkan nilai untuk parameter suhu
16.	activity_swipelayout.xml	Menampilkan tampilan <i>imageslider</i>
17.	parameter_intensitas.xml	Menampilkan data dari <i>database</i> untuk parameter intensitas
18.	parameter_karbon.xml	Menampilkan data dari <i>database</i> untuk parameter karbon
19.	parameter_kelembaban.xml	Menampilkan data dari <i>database</i> untuk parameter kelembaban
20.	parameter_suhu.xml	Menampilkan data dari <i>database</i> untuk parameter suhu

Tabel 4.1 menjelaskan daftar *activity layout xml* yang digunakan pada aplikasi aeromosis ini. *Layout xml* tersebut merupakan *layout* desain dari tampilan aplikasi aeromosis. *Layout xml* dibagi menjadi dua *mode* yaitu *mode design* dan *mode text*. *Mode text* berisi *source code* atau barisan *coding* dari *text*. Tampilan *design* pada aplikasi aeromosis saja tidak akan berjalan baik tanpa adanya *file java*. *File java* berfungsi sebagai penghubung antara *design* tampilan dengan *user*

agar aplikasi berjalan dengan baik. Daftar *file* java yang digunakan pada pembuatan aplikasi aeromosis dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar *File* Java Aplikasi Aeromosis

No.	File Java	Fungsi
1.	Intensitas_Activity	Menjalankan <i>activity layout</i> intensitas <i>xml</i>
2.	Kelembaban_Activity	Menjalankan <i>activity layout</i> kelembaban <i>xml</i>
3.	Karbon_Activity	Menjalankan <i>activity layout</i> karbon <i>xml</i>
4.	Suhu_Activity	Menjalankan <i>activity layout</i> suhu <i>xml</i>
5.	AdapterIntensitas	Menjalankan <i>activity layout</i> parameter intensitas <i>xml</i>
6.	AdapterKarbon	Menjalankan <i>activity layout</i> parameter karbon <i>xml</i>
7.	AdapterKelembaban	Menjalankan <i>activity layout</i> parameter kelembaban <i>xml</i>
8.	AdapterSuhu	Menjalankan <i>activity layout</i> parameter suhu <i>xml</i>
9.	GrafikIntensitas	Menjalankan <i>activity layout</i> grafik intensitas <i>xml</i>
10.	GrafikKarbon	Menjalankan <i>activity layout</i> grafik karbon <i>xml</i>
11.	GrafikKelembaban	Menjalankan <i>activity layout</i> grafik kelembaban <i>xml</i>
12.	GrafikSuhu	Menjalankan <i>activity layout</i> grafik suhu <i>xml</i>
13.	MySingletonintensitas	Menampilkan grafik <i>line</i> pada parameter intensitas
14.	MySingletonkarbon	Menampilkan grafik <i>line</i> pada parameter karbondioksida
15.	MySingletonkelembaban	Menampilkan grafik <i>line</i> pada parameter kelembaban
16.	MySingletonsuhu	Menampilkan grafik <i>line</i> pada parameter suhu
17.	CustomSwipeAdapter2	Menjalankan <i>activity ImageSlider</i> pada menu utama
18.	MainQuestion	Menjalankan <i>activity layout xml</i> menu <i>aeroponik interview question</i>
19.	MenuIntensitas	Menjalankan <i>activity layout</i> menu intensitas <i>xml</i>
20.	MenuKarbon	Menjalankan <i>activity layout</i> menu karbon <i>xml</i>
21.	MenuKelembaban	Menjalankan <i>activity layout</i> menu kelembaban <i>xml</i>
22.	MenuSuhu	Menjalankan <i>activity layout</i> menu suhu <i>xml</i>
23.	MenuParameter	Menjalankan <i>activity layout</i> menu parameter <i>xml</i>
24.	Intensitas	Menjalankan fungsi untuk menampilkan nilai parameter intensitas
25.	Karbon	Menjalankan fungsi untuk menampilkan nilai parameter karbondioksida
26.	Kelembaban	Menjalankan fungsi untuk menampilkan nilai parameter kelembaban
27.	Suhu	Menjalankan fungsi untuk menampilkan nilai parameter suhu
28.	AppController	Menjalankan fungsi pengambilan data
29.	ServerAPI	Mengambil data dengan memasukan URL data
30.	Splashscreen	Menjalankan <i>activity layout xml splashscreen</i>

### 4.3 Hasil Perancangan Antarmuka Aplikasi Aeromosis

Aeromosis merupakan singkatan dari aeroponik *monitoring system*, aplikasi aeromosis ini digunakan untuk pemantauan jarak jauh tanaman bayam dengan parameter suhu, kelembaban, intensitas cahaya serta kadar karbondioksida. Aplikasi aeromosis menampilkan nilai dari parameter tersebut, data grafik yang berfungsi untuk melihat nilai dari parameter yang telah tersimpan ke dalam *database* yang berbentuk grafik *line*, beberapa pertanyaan dan jawaban seputar aeroponik.

Pengujian antarmuka dilakukan pada menu utama untuk memastikan menu pada antarmuka menu utama dapat berjalan sesuai dengan perancangan. Tampilan aplikasi yang sesuai dengan perancangan menggunakan *smartphone* dengan ukuran layar 5 inci, sistem operasi android 5.0.0 seperti pada merk hp Lenovo A6600d40. Hasil pengujian antarmuka aplikasi pada menu utama dapat dilihat pada Tabel 4.3

Tabel 4.3 Pengujian Antarmuka Menu Utama

No.	Test Case	Hasil yang diharapkan	Hasil Uji
1.	<i>ImageSlider</i>	Menampilkan beberapa foto yang dapat bergeser secara otomatis	Berhasil
2.	Menu Parameter Suhu	Menampilkan isi dari menu parameter suhu	Berhasil
3.	Menu Parameter Kelembaban	Menampilkan isi dari menu parameter kelembaban	Berhasil
4.	Menu Parameter Intensitas	Menampilkan isi dari menu parameter intensitas	Berhasil
5.	Menu Parameter Karbondioksida	Menampilkan isi dari menu parameter karbondioksida	Berhasil
6.	Menu Diskusi Aeroponik	Menampilkan menu diskusi aeroponik.	Berhasil
7.	Menu Keluar	Menampilkan <i>toast</i> berupa pilihan untuk meninggalkan aplikasi.	Berhasil

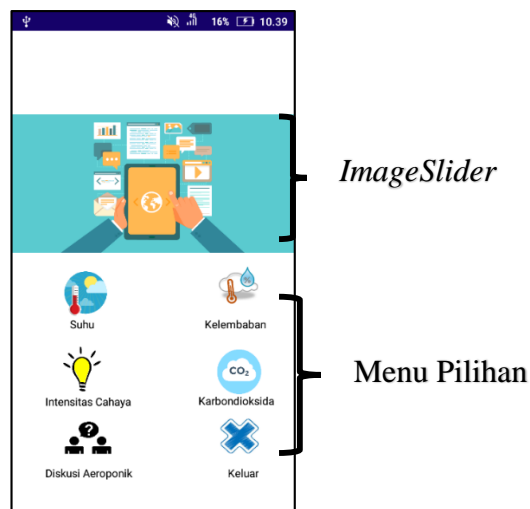
Tabel 4.3 merupakan hasil pengujian antarmuka pada menu utama, semua menu pada menu utama dapat berjalan sesuai dengan perancangan. Berikut tampilan dari menu utama aeromosis terlihat pada Gambar 4.3.





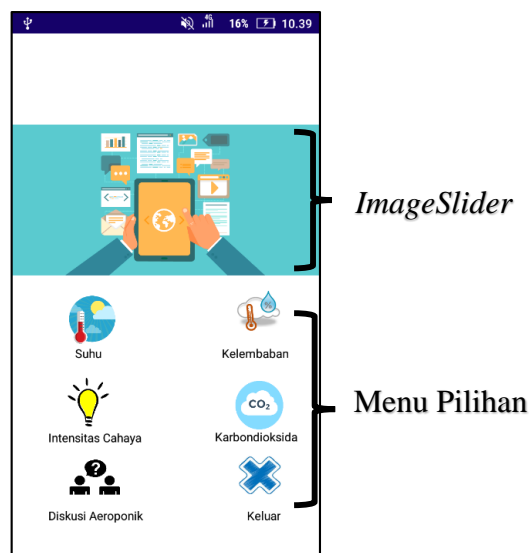
Gambar 4.3 Tampilan *SplashScreen* pada *Smartphone* Lenovo A6600d40

Gambar 4.3 merupakan tampilan *splashscreen*, yaitu tampilan awal ketika *user* memulai untuk membuka aplikasi sistem *monitoring* tanaman bayam aeroponik. Pada tampilan ini terdapat logo yang melambangkan aplikasi dengan tulisan aeromosis (aeroponik *monitoring system*). *Splashscreen* berdurasi sekitar 5 detik, kemudian secara otomatis akan menuju pada tampilan menu utama.



Gambar 4.4 Tampilan Menu Utama pada *Smartphone* Lenovo A6600d40

Tampilan menu pada Gambar 4.4 merupakan tampilan menu utama aplikasi aeromosis (aeroponik *monitoring system*). Menu utama terdiri dari *imageslider* dan beberapa menu pilihan. *Imageslider* pada menu utama ini memuat beberapa animasi yang dapat bergeser secara otomatis. Tampilan dari *imageslider* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Tampilan *ImageSlider* Menu Utama pada *Smartphone* Lenovo A6600d40

*Imageslider* pada Gambar 4.5 dapat bergeser secara otomatis dengan *delay* setiap dua detik dan tiga detik untuk pindah dari gambar satu ke yang lain. Setelah gambar kelima secara otomatis akan berpindah ke gambar pertama. Selain *imageslider* menu utama juga menampilkan beberapa menu pilihan diantaranya:

- i. Menu Parameter Suhu
- ii. Menu Parameter Kelembaban
- iii. Menu Parameter Intensitas Cahaya
- iv. Menu Parameter Karbondioksida
- v. Menu Diskusi Aeroponi
- vi. Menu Keluar

Penjelasan dari menu pilihan diatas adalah sebagai berikut:

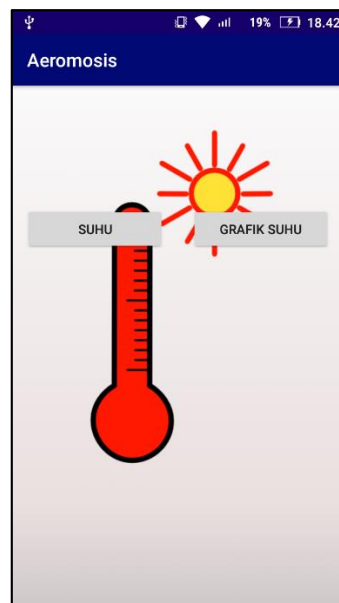
### 1. Menu Parameter Suhu

Aeromosis merupakan aplikasi *monitoring* tanaman aeroponik yang berfokus pada tanaman bayam. Pada aplikasi aeromosis menu parameter suhu ini menampilkan nilai dan grafik *line*. Pengujian antarmuka dilakukan pada menu parameter suhu untuk memastikan menu pada antarmuka menu parameter suhu dapat berjalan sesuai dengan perancangan. Hasil pengujian antarmuka aplikasi pada menu parameter suhu dapat dilihat pada Tabel 4.4.

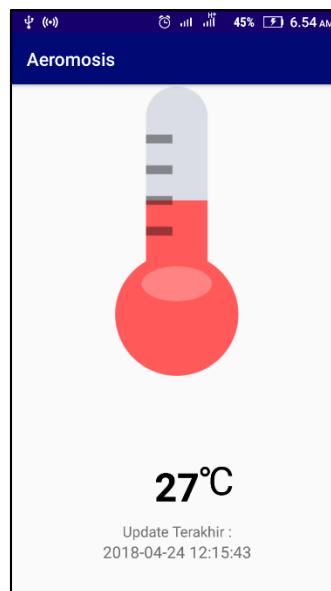
Tabel 4.4 Pengujian Antarmuka Menu Parameter Suhu

No.	Test Case	Hasil yang diharapkan	Hasil Uji
1.	Pilihan Suhu	Menampilkan nilai suhu secara <i>realtime</i>	Berhasil
2.	Pilihan Data Grafik Suhu	Menampilkan grafik dari menu suhu berupa grafik <i>line</i>	Berhasil

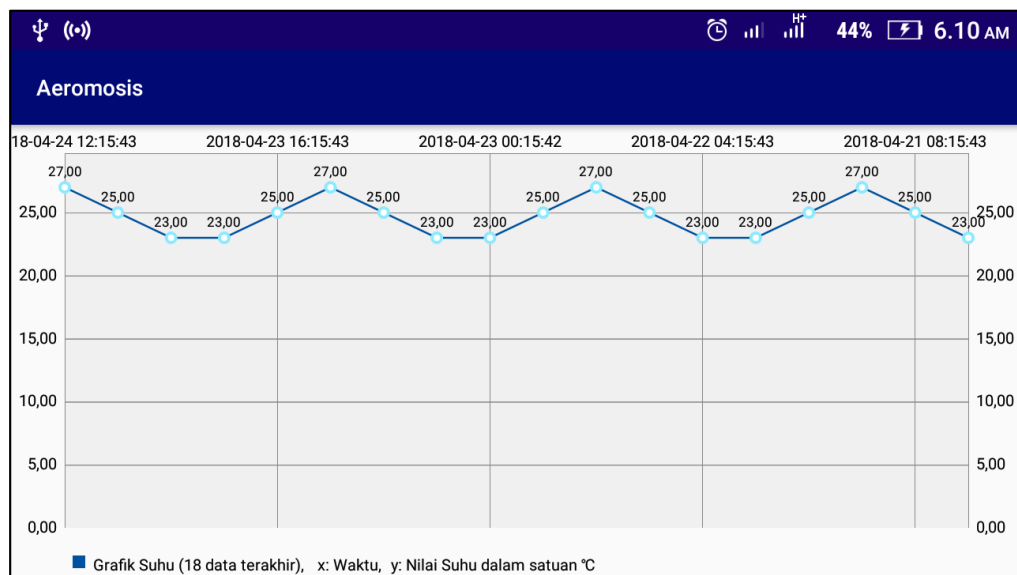
Tabel 4.4 merupakan hasil pengujian antarmuka pada menu parameter suhu, semua menu pada menu tersebut dapat berjalan sesuai dengan perancangan. Tampilan menu parameter suhu dapat dilihat dalam Gambar 4.6.

Gambar 4.6 Tampilan Menu Parameter Suhu pada *Smartphone* Lenovo A6600d40

Gambar 4.6 merupakan tampilan jika *user* memilih menu parameter suhu. Pada Gambar 4.6 berisi dua pilihan yaitu pilihan suhu yang berfungsi untuk menampilkan nilai parameter dan pilihan data grafik suhu untuk menampilkan data berupa tampilan grafik *line* sebanyak 18 data terakhir yang telah tersimpan pada *database*. Tampilan nilai suhu dapat dilihat pada Gambar 4.7 dan tampilan menu grafik pada Gambar 4.8.



Gambar 4.7 Tampilan Nilai Parameter Suhu pada *Smartphone* Lenovo A6600d40



Gambar 4.8 Tampilan Grafik *Line* Parameter Suhu

## 2. Menu Parameter Kelembaban

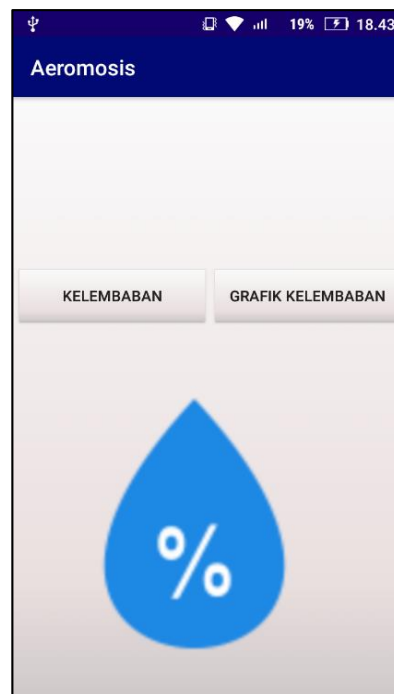
Tampilan menu parameter kelembaban tidak berbeda jauh dengan tampilan menu parameter suhu. Pada aplikasi aeromosis menu parameter kelembaban ini menampilkan nilai secara *real time* dan grafik *line*. Pengujian antarmuka dilakukan pada menu parameter kelembaban untuk memastikan menu pada antarmuka menu

parameter kelembaban dapat berjalan sesuai dengan perancangan. Hasil pengujian antarmuka aplikasi pada menu parameter kelembaban dapat dilihat pada Tabel 4.5.

Tabel 4.5 Pengujian Antarmuka Menu Parameter Kelembaban

No.	Test Case	Hasil yang diharapkan	Hasil Uji
1.	Pilihan Kelembaban	Menampilkan nilai kelembaban secara <i>realtime</i>	Berhasil
2.	Pilihan Data Grafik Kelembaban	Menampilkan grafik dari menu kelembaban berupa grafik <i>line</i>	Berhasil

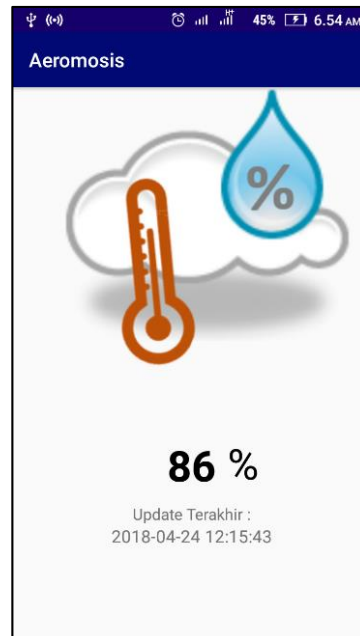
Tabel 4.5 merupakan hasil pengujian antarmuka pada menu parameter kelembaban, semua menu pada menu tersebut dapat berjalan sesuai dengan perancangan. Tampilan menu parameter kelembaban dapat dilihat dalam Gambar 4.9.



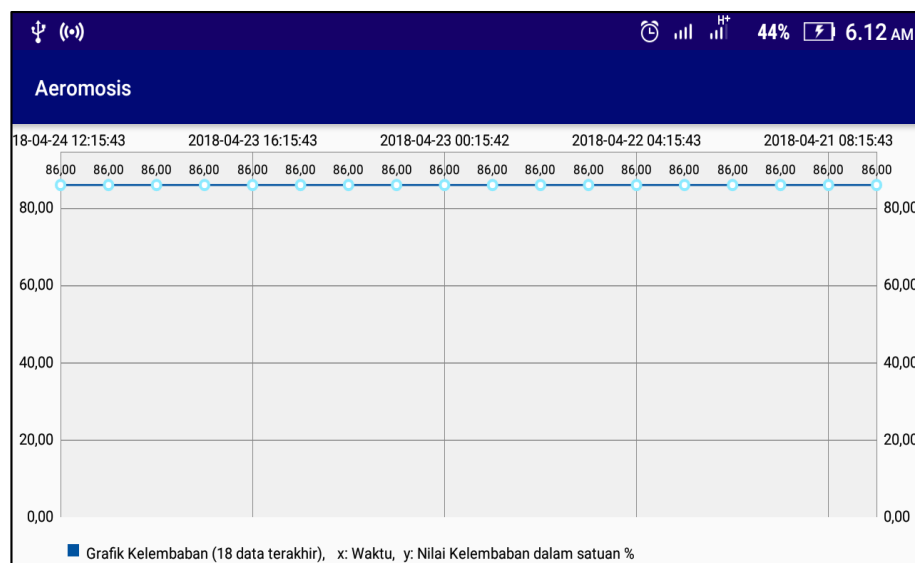
Gambar 4.9 Tampilan Menu Parameter Kelembaban pada *Smartphone* Lenovo A6600d40

Gambar 4.9 merupakan tampilan dari menu parameter kelembaban. Ketika *user* memilih menu kelembaban, tampilan pada menu kelembaban sama seperti pada menu suhu. Terdapat dua pilihan yaitu kelembaban dan data grafik kelembaban. Pilihan kelembaban berisi nilai kelembaban dan pilihan data grafik

kelembaban untuk menampilkan data berupa tampilan grafik *line* sebanyak 18 data terakhir yang telah tersimpan pada *database*. Tampilan nilai kelembaban dapat dilihat pada Gambar 4.10 dan tampilan menu grafik pada Gambar 4.11.



Gambar 4.10 Tampilan Nilai Parameter Kelembaban pada *Smartphone* Lenovo A6600d40



Gambar 4.11 Tampilan Grafik *Line* Menu Parameter Kelembaban pada *Smartphone* Lenovo A6600d40

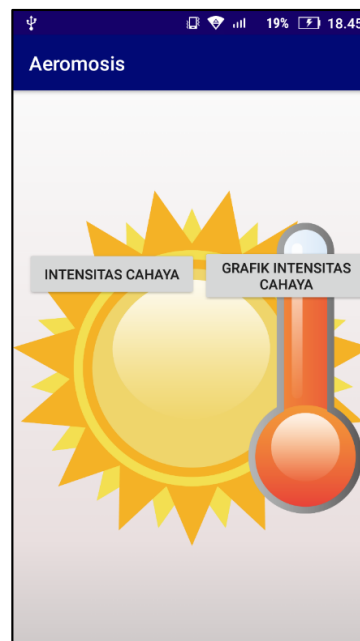
### 3. Menu Parameter Intensitas Cahaya

Tampilan pada menu intensitas cahaya tidak berbeda jauh dengan menu sebelumnya yaitu suhu dan kelembaban, yaitu menampilkan nilai secara *realtime* dan grafik *line*. Pengujian antarmuka dilakukan pada menu parameter intensitas cahaya untuk memastikan menu pada antarmuka menu parameter intensitas cahaya dapat berjalan sesuai dengan perancangan. Hasil pengujian antarmuka aplikasi pada menu parameter intensitas cahaya dapat dilihat pada Tabel 4.6.

Tabel 4.6 Pengujian Antarmuka Menu Parameter Intensitas

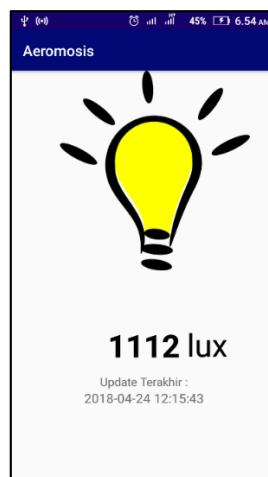
No.	Test Case	Hasil yang diharapkan	Hasil Uji
1.	Pilihan Intensitas	Menampilkan nilai intensitas secara <i>realtime</i>	Berhasil
2.	Pilihan Data Grafik Intensitas	Menampilkan grafik dari menu intensitas berupa grafik <i>line</i>	Berhasil

Tabel 4.6 merupakan hasil pengujian antarmuka pada menu parameter intensitas cahaya, semua menu pada menu tersebut dapat berjalan sesuai dengan perancangan. Tampilan menu parameter intensitas cahaya dapat dilihat dalam Gambar 4.12.

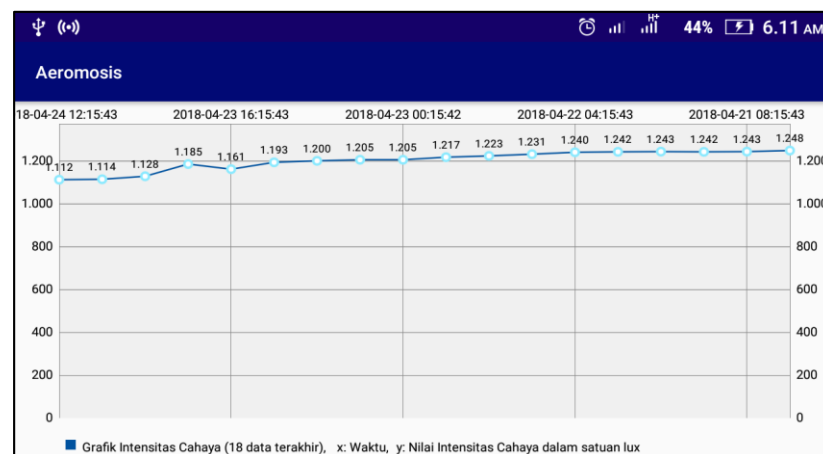


Gambar 4.12 Tampilan Menu Parameter Intensitas Cahaya pada *Smartphone* Lenovo A6600d40

Tampilan menu parameter intensitas cahaya dapat dilihat pada Gambar 4.12. Ketika *user* memilih menu intensitas cahaya, tampilan pada menu parameter ini akan sama seperti pada menu-menu sebelumnya yaitu suhu dan kelembaban. Terdapat dua pilihan yaitu intensitas cahaya dan data grafik intensitas cahaya. Pilihan intensitas cahaya berisi nilai dari intensitas cahaya dan pilihan data grafik intensitas cahaya berupa grafik *line* yang menampilkan data sebanyak 18 data terakhir yang telah tersimpan pada *database*. Tampilan nilai dari parameter intensitas cahaya dapat dilihat pada Gambar 4.13 dan tampilan menu grafik pada Gambar 4.14.



Gambar 4.13 Tampilan Nilai Parameter Intensitas Cahaya pada *Smartphone* Lenovo A6600d40



Gambar 4.14 Tampilan Grafik *Line* Menu Parameter Intensitas Cahaya pada *Smartphone* Lenovo A6600d40



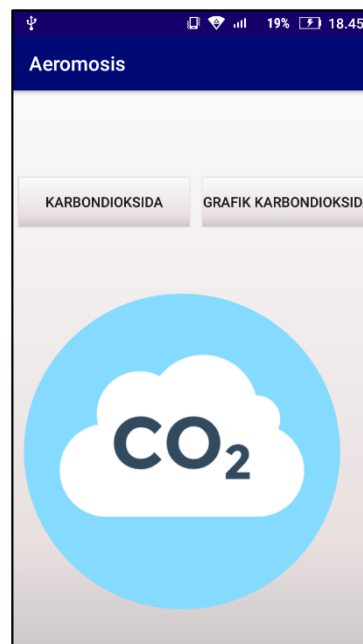
#### 4. Menu Parameter Karbondioksida

Tampilan pada menu karbondioksida tidak berbeda jauh dengan 3 menu sebelumnya yaitu suhu, kelembaban dan intensitas cahaya, yaitu menampilkan nilai secara *realtime* dan grafik *line*. Pengujian antarmuka dilakukan pada menu parameter karbondioksida untuk memastikan menu pada antarmuka menu parameter karbondioksida dapat berjalan sesuai dengan perancangan. Hasil pengujian antarmuka aplikasi pada menu parameter karbondioksida dapat dilihat pada Tabel 4.7.

Tabel 4.7 Pengujian Antarmuka Menu Parameter Karbondioksida

No.	Test Case	Hasil yang diharapkan	Hasil Uji
1.	Pilihan Karbondioksida	Menampilkan nilai karbondioksida secara <i>realtime</i>	Berhasil
2.	Pilihan Data Grafik Karbondioksida	Menampilkan grafik dari menu karbondioksida berupa grafik <i>line</i>	Berhasil

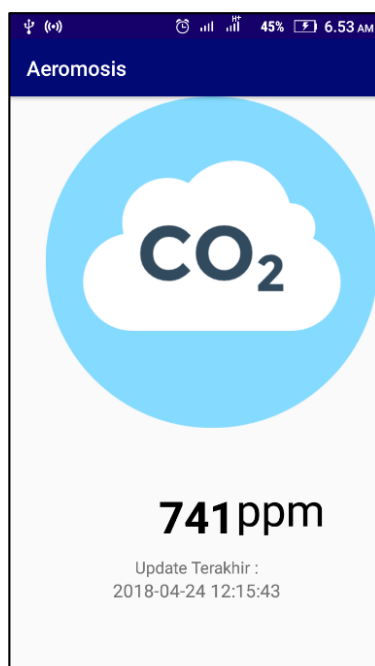
Tabel 4.7 merupakan hasil pengujian antarmuka pada menu parameter karbondioksida, semua menu pada menu tersebut dapat berjalan sesuai dengan perancangan. Tampilan menu parameter karbondioksida dapat dilihat dalam Gambar 4.15.



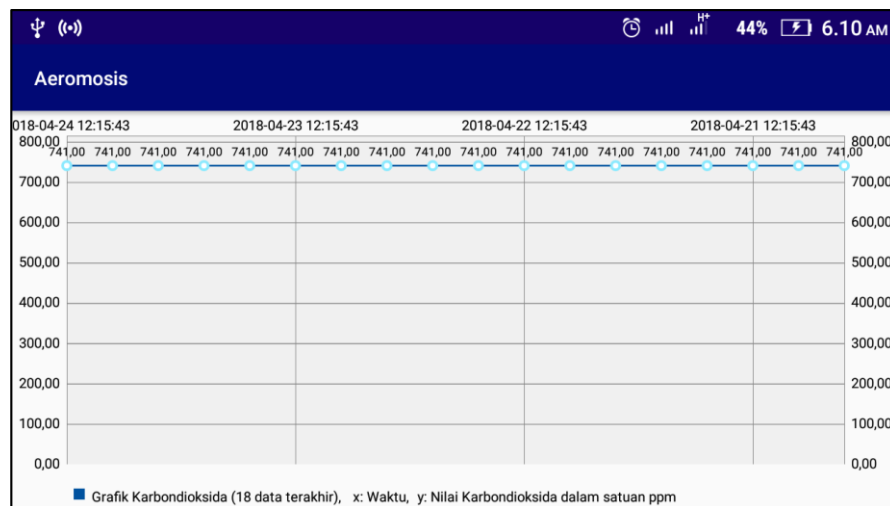
Gambar 4.15 Tampilan Menu Parameter Karbondioksida pada *Smartphone*

Lenovo A6600d40

Tampilan menu parameter karbondioksida dapat dilihat pada Gambar 4.15. Ketika *user* memilih menu karbondioksida, tampilan pada menu parameter ini akan sama seperti pada menu-menu sebelumnya yaitu suhu, kelembaban dan intensitas cahaya. Terdapat dua pilihan yaitu karbondioksida dan data grafik karbondioksida. Pilihan karbondioksida berisi nilai dari karbondioksida dan pilihan data grafik karbondioksida berupa grafik line yang menampilkan data sebanyak 18 data terakhir yang telah tersimpan pada *database*. Tampilan nilai dari parameter karbondioksida dapat dilihat pada Gambar 4.16 dan tampilan menu grafik pada Gambar 4.17.



Gambar 4.16 Tampilan Nilai Parameter Karbondioksida pada *Smartphone* Lenovo A6600d40



Gambar 4.17 Tampilan Grafik *Line* Menu Parameter Karbondioksida pada *Smartphone* Lenovo A6600d40

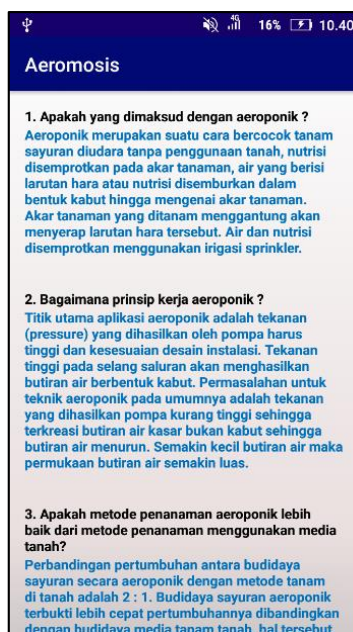
## 5. Menu Diskusi Aeroponik

Menu diskusi aeroponik berisi beberapa pertanyaan dan jawaban. Pengujian antarmuka dilakukan pada menu diskusi aeroponik untuk memastikan antarmuka menu diskusi aeroponik dapat berjalan sesuai dengan perancangan atau tidak. Hasil pengujian antarmuka aplikasi pada menu diskusi aeroponik dapat dilihat pada Tabel 4.8.

Tabel 4.8 Pengujian Antarmuka Menu Diskusi Aeroponik

No.	<i>Test Case</i>	Hasil yang diharapkan	Hasil Uji
1.	Menu Diskusi Aeroponik	Menampilkan isi menu diskusi aeroponik	Berhasil

Tabel 4.8 merupakan hasil pengujian antarmuka pada menu diskusi aeroponik, menu tersebut dapat berjalan sesuai dengan perancangan. Tampilan menu diskusi aeroponik dapat dilihat pada Gambar 4.18.



Gambar 4.18 Tampilan Menu Diskusi Aeroponik pada *Smartphone* Lenovo A6600d40

Menu ini berisi 5 pertanyaan dan jawaban mengenai seputar aeroponik. Untuk penambahan beberapa pertanyaan dan jawaban bias dilakukan dengan menambahkan pertanyaan dan jawaban pada *layout xml* pada *software* android studio. Pada menu ini dapat di-*scroll* keatas maupun kebawah. Ketika memilih tombol *back* pada *smartphone* maka akan kembali pada menu utama.

## 6. Menu Keluar

Menu keluar merupakan menu untuk keluar dari aplikasi. Menu ini menampilkan dialog pilihan untuk meyakinkan *user* sebelum keluar dari aplikasi. Pengujian antarmuka dilakukan pada menu keluar untuk memastikan antarmuka menu keluar dapat berjalan sesuai dengan perancangan. Hasil pengujian antarmuka aplikasi pada menu keluar dapat dilihat pada Tabel 4.9.

Tabel 4.9 Pengujian Antarmuka Menu Keluar

No.	Test Case	Hasil yang diharapkan	Hasil Uji
1.	Menu Keluar	Menampilkan dialog pilihan untuk meyakinkan <i>user</i> untuk keluar atau tidak	Berhasil

Tabel 4.9 merupakan hasil pengujian antarmuka pada menu keluar, menu tersebut dapat berjalan sesuai dengan perancangan. Tampilan menu keluar dapat dilihat pada Gambar 4.19.



Gambar 4.19 Tampilan Menu Keluar pada *Smartphone* Lenovo A6600d40

Jika memilih menu keluar maka akan muncul dialog pilihan untuk meyakinkan *user*. *User* tinggal memilih pilihan “YA” untuk keluar dari aplikasi aeromosis dan “TIDAK” apabila *user* ingin melanjutkan aplikasi aeromosis dan akan kembali pada menu utama. Selain menu keluar, untuk keluar dari aplikasi secara langsung tanpa ada pemberitahuan sebelumnya tinggal menekan tombol *back* pada *smartphone* saat berada pada menu utama aplikasi aeromosis.

#### 4.4 Pengujian terhadap Berbagai Layar Perangkat

Pengujian terhadap berbagai layar perangkat dilakukan untuk mengecek sistem dapat bekerja dengan baik atau tidak pada *smartphone* lainnya. Dalam penelitian ini menggunakan 10 macam *smartphone* dengan berbagai tipe dan spesifikasi yang berbeda-beda. Daftar berbagai perangkat yang digunakan dapat dilihat pada Tabel 4.10.

Tabel 4.10 Daftar Berbagai Perangkat yang digunakan

No.	Nama Perangkat	Ukuran Layar	Sistem Operasi	API
1.	Lenovo A6600d40	5 inci (720x1280 piksel)	Android 6.0	23
2.	Samsung GT-S6310	3.27 inci (320x480 piksel)	Android 4.1.2	16
3.	Xiaomi Redmi 4A	5 inci (720x1280 piksel)	Android 6.0.1	23
4.	Xiaomi Mi A1	5.5 inci (1920x1080 piksel)	Android 8.0	26
5.	Asus Z00RD	5 inci (720x1280 piksel)	Android 5.0.2	23
6.	Lenovo A7010a48	5 inci (720x1280 piksel)	Android 6.0	23
7.	Xiaomi Mi Max	6.44 inci (1920x1080 piksel)	Android 6.0.1	23
8.	Xiaomi Redmi Note 4	5.5 inci (1920x1080 piksel)	Android 6.0	23
9.	ASUS-X0011D	5.5 inci (1920x1080 piksel)	Android 7.1.1	25
10.	Freetel FTJ161A	4 inci (480x800 piksel)	Android 5.1	22

Jika aplikasi dapat dijalankan pada *smartphone* dan sesuai dengan perancangan maka akan dianggap berhasil, sedangkan jika aplikasi tidak dapat dijalankan atau bahkan dapat dijalankan namun tampilan layar tidak sesuai perancangan pada *smartphone* maka aplikasi dianggap gagal. Hasil pengujian dari berbagai ukuran layar perangkat dapat dilihat pada Tabel 4.11.

Tabel 4.11 Hasil Pengujian dari Berbagai Ukuran Layar Perangkat

No.	Nama Perangkat	Hasil Uji
1.	Lenovo A6600d40	Berhasil
2.	Samsung GT-S6310	Gagal
3.	Redmi 4A	Berhasil
4.	Xiaomi Mi A1	Berhasil
5.	Asus Z00RD	Berhasil
6.	Lenovo A7010a48	Berhasil
7.	Xiaomi Mi Max	Berhasil
8.	Xiaomi Redmi Note 4	Berhasil
9.	ASUS-X0011D	Berhasil
10.	Freetel FTJ161A	Berhasil

Hasil pengujian dari berbagai ukuran layar perangkat pada Tabel 4.11, diperoleh pengukuran yang berhasil 9 dan 1 gagal pada *smartphone*. Kegagalan pada pengujian ini disebabkan oleh tampilan layar yang tidak sesuai dengan

perancangan. Kegagalan ini terjadi karena aplikasi dijalankan pada *smartphone* dengan ukuran layar perangkat dibawah 4 inci dan sistem operasi pada *smartphone*

4.1.2. Ukuran layar dan sistem operasi pada *smartphone* sangat berpengaruh sekali pada aplikasi sistem *monitoring* ini.

#### 4.5 Pembahasan Aplikasi Aeromosis

Aeromosis merupakan aplikasi sistem *monitoring* pada tanaman aeroponik. Tanaman yang difokuskan pada penelitian ini adalah tanaman bayam dengan parameter suhu, kelembaban, intensitas cahaya, dan kadar karbondioksida. Sebelum melakukan pengujian aplikasi pada beberapa *smartphone*, dilakukan pengujian *file php* yang telah dibuat pada *software notepad++*. Kemudian *file php* disimpan dengan nama sesuai parameter yang berektensi “.php”.

Semua *file php* di-*upload* kedalam 000webhostapp (penyedia layanan *web hosting* gratis). Dalam pengaksesan *file php* yang telah tersimpan kedalam *folder monitoring* pada 000webhostapp cukup dengan memasukan alamat baik pada PC maupun *smartphone* “verapusita.000webhostapp.com/monitoring”. Dengan adanya koneksi internet, 000webhostapp ini dapat dibuka oleh semua *user* dan di mana saja tanpa takut terhalang oleh jaringan internet yang berbeda.

Aplikasi aeromosis dibuat menggunakan *software android studio* versi 3.0. Aplikasi aeromosis pada *software android studio* ini memerlukan 20 *activity layout* berektensi xml dan 30 *file java*. Jika terdapat salah satu kesalahan atau kekurangan *coding* aplikasi tidak akan berjalan sehingga pada saat pengujian akan mengalami *stop application* (keluar secara otomatis). *Software android studio* versi 3.0 ini sangat berat sehingga memerlukan waktu lama baik dalam *coding* maupun pengujian menggunakan *emulator* bawaan dari *software android studio*. Karena pengujian pada *emulator* memerlukan waktu yang lama, maka dalam hal pengujian aplikasi aeromosis dilakukan dengan menggunakan *smartphone* dengan sistem operasi android.

Pengujian ini juga menggunakan beberapa *smartphone* untuk mengecek aplikasi aeromosis berjalan sesuai dengan perancangan sistem. Sebelum mengakses aplikasi aeromosis disarankan kepada *user* untuk mengecek koneksi internet pada

*smartphone*. Karena apabila tidak ada koneksi internet, maka aplikasi tidak dapat memuat nilai parameter dan *data* grafik yang tersimpan dalam *database*.

Hasil dari *activity* nilai *current activity* pada parameter baik parameter suhu, kelembaban, intensitas cahaya dan kadar CO<sub>2</sub> berjalan sesuai dengan perancangan. Nilai parameter muncul beserta waktu dan tanggal *update* terakhir sesuai data dari *database*. Namun dalam pergantian *database* tidak bisa langsung memperbarui data selanjutnya, harus keluar terlebih dahulu dari *activity* submenu parameter baik parameter suhu, kelembaban, intensitas cahaya serta kadar CO<sub>2</sub>.

Hasil dari *activity* nilai data grafik pada parameter baik parameter suhu, kelembaban, intensitas cahaya serta kadar CO<sub>2</sub> tidak berbeda jauh dengan *activity* pada submenu parameter. Namun pada *activity* data grafik nilai yang dimunculkan berupa data grafik *line* yang memuat nilai sebanyak 18 data terakhir yang telah tersimpan pada *database*. Delapan belas data terakhir diperoleh dari *query* pada *file* data grafik baik parameter suhu, kelembaban, intensitas cahaya serta kadar CO<sub>2</sub>. Contoh *query* pada *file* data grafik salah satu parameter dapat dilihat pada Gambar 4.20, yang terletak pada *line* 54.

```

43 // $result = array();
44 // while($row = mysql_fetch_array($res)){
45 //     array_push($result, array('suhu'=>$row[1], 'waktu'=>$row[2]));
46 // }
47 // echo json_encode(array("result"=>$result));
48 // mysql_close($connect_db);
49 // }
50
51 include_once('connect.php');
52
53
54 $query = "SELECT nilai,waktu FROM suhu ORDER BY waktu DESC LIMIT 18";
55 //SELECT nilai,waktu FROM suhu WHERE waktu >= now() - INTERVAL 1 MONTH GROUP BY DAY(waktu);
56 $result = mysql_query($koneksi,$query);
57
58 $arraydata = array();
59
60 while($baris = mysql_fetch_assoc($result))
61 {
62     $arraydata[]=$baris;
63 }
64
65 echo json_encode($arraydata);
66
67

```

Gambar 4.20 *Query* Data Grafik Parameter Suhu

Berdasarkan pengujian terhadap beberapa *smartphone* seperti pada Tabel 4.11 yang telah dilakukan, aplikasi aeromosis berjalan cukup baik dan sesuai yang diharapkan jika dijalankan pada *smartphone* dengan ukuran layar diatas 4 inci dan



sistem operasi android pada *smartphone* 5.0.0 keatas pada semua *merk* dan tipe *smartphone*. Namun jika aplikasi aeromosis dijalankan pada ukuran layar *smartphone* dibawah 4 inci dan sistem operasi android dibawah 5.0.0 dapat dijalankan, tetapi tampilan pada perangkat tidak sesuai perancangan. Tampilan aplikasi aeromosis pada ukuran layar *smartphone* 4 inci dan sistem operasi android 4.1.2 API 16 dapat dilihat pada Gambar 4.21.



Gambar 4.21 Tampilan pada *Smartphone* Samsung GT-S6310

Pada Gambar 4.21 terdapat perbedaan tampilan disebabkan oleh ukuran layar 4 inci. Namun aplikasi dapat berjalan dengan baik. Hanya saja tampilannya sedikit berbeda. Kendala dalam pengembangan aplikasi aeromosis ini adalah banyaknya perangkat android yang beredar namun tiap perangkat tersebut memiliki ukuran layar serta sistem operasi android yang berbeda-beda. Hal ini menyebabkan tampilan di tiap perangkat kemungkinan berbeda, sehingga diperlukan penanganan ekstra agar tampilan konten dapat konsisten pada setiap perangkat android.

## **BAB V**

### **PENUTUP**

Penutup dari penelitian yang telah dibuat, pada bab V berisi tentang kesimpulan dan saran yang di buat berdasarkan penjelasan yang telah di uraikan pada bab-bab sebelumnya mengenai Aplikasi Sistem *Monitoring* Suhu, Kelembaban, Intensitas Cahaya dan Kadar CO<sub>2</sub> pada Tanaman Aeroponik Berbasis Android.

#### **5.1 Kesimpulan**

Berdasarkan hasil penelitan aplikasi sistem *monitoring* suhu, kelembaban, intensitas cahaya serta kadar karbondioksida pada tanaman bayam aeroponik berbasis android dapat diambil kesimpulan sebagai berikut:

1. Aplikasi sistem *monitoring* tanaman bayam aeroponik telah selesai sesuai dengan perancangan. Semua fungsi algoritma berjalan pada aplikasi berjalan dengan sangat baik pada *smartphone* dengan minimal ukuran layar pada *smartphone* 4 inci dan sistem operasi android 5.0.0 dan sistem sudah memenuhi spesifikasi kebutuhan tanaman.
2. Aplikasi sistem *monitoring* jarak jauh dapat menampilkan nilai parameter suhu, kelembaban, intensitas cahaya serta kadar karbondioksida dengan menampilkan nilai dan waktu. Selain itu aplikasi ini dapat juga menampilkan data grafik berupa tampilan grafik *line* sebanyak 18 data terakhir yang diambil dari *database*. Aplikasi ini sangat bermanfaat untuk pemantauan perkembangan pertumbuhan tanaman bayam aeroponik.

#### **5.2 Saran**

Ada beberapa saran untuk penelitian selanjutnya yang berkaitan dengan penelitian ini, yaitu:

1. Aplikasi sistem *monitoring* tanaman bayam aeroponik berbasis android perlu adanya penambahan beberapa parameter seperti jumlah oksigen, pH,

hasil panen, kondisi tanaman apakah sehat atau tidak dan sebagainya. Serta dapat diaplikasikan juga pada tanaman selain aeroponik.

2. Diharapkan pada penelitian selanjutnya dapat dikembangkan dengan adanya penambahan fitur dan kelengkapan dibagian server, sehingga dapat menyajikan informasi mengenai *monitoring* secara lengkap. Misalnya ada penambahan *video streaming*, penambahan notifikasi jika nilai dari parameter kelebihan atau kekurangan dan sebagainya.
3. Aplikasi ini perlu dikembangkan lebih lanjut agar dapat di *install* pada semua sistem operator selain android, misalna *Iphone Operating system*, *Windows Phone*, *Blackberry* dan sebagainya.
4. Perlu adanya penanganan ekstra agar tampilan aplikasi dapat konsisten pada setiap perangkat *smartphone*. Karena setiap perangkat tersebut memiliki ukuran layar yang berbeda-beda.
5. Perlu adanya penambahan fungsi *refresh* pada aplikasi, agar pada saat pergantian nilai atau grafik dapat langsung dilihat tanpa perlu menekan tombol bak pada aplikasi dan masuk kembali pada menu nilai dan grafik.

## DAFTAR PUSTAKA

- [1] Wahono, S., Sugiyanto, Elfita, “Eksperimen Pengaturan Suhu dan Kelembaban pada Rumah Tanaman (Greenhouse) dengan Sistem Humidifikasi”, Jurnal Teknik Mesin S-1, Vol. 2, No. 1, pp. 49-56, 2014.
- [2] Ferdiansyah, “Rancang Bangun Sistem Kontrol pada Budidaya Tanaman dengan Teknik Aeroponik”, Institut Teknologi Sepuluh November, pp. 11-17, 2012.
- [3] Ramdani M., Andrian R., Novian Anggit, “Perancangan Sistem Monitoring Tanaman Menggunakan Zigbee dan Platform M2M”, Universitas Telkom Bandung, pp. 1-7, 2014.
- [4] Y. Sutiyoso, “Aeroponik Sayuran, Budidaya dengan Sistem Pengabutan Swadaya”, Jakarta, 2003.
- [5] Ruminta, “Analisis Penuruna Produksi Tanaman Padi Akibat Perubahan Iklim Di Kabupaten Bandung Jawa Barat”, Universitas Padjajaran, Jurnal Kultivasi Vol. 15, No. 1, pp. 37-43, 2016.
- [6] T. Budioko, “Sistem Monitoring Jarak Jauh Berbasis Internet of Things Menggunaka Protokol MQTT”, Seminar Nasional Riset Teknologi Informasi, Vol. 1, pp. 353-358, 2016
- [7] Syarief, Syafrizal, “Sistem Monitoring Suhu dan Kelembaban Tanaman Cabai pada Greenhouse Berbasis LabVIEW”, Vol. 15, No. 2, pp. 135-140, 2016.
- [8] Yahwe P. Caesar, Isnawaty, L.M. Fid Aksara, “Rancang Bangun Prototype System Monitoring Kelembaban Tanah Melalui SMS Berdasarkan Hasil Penyiraman Tanaman”, semanTIK, Vol. 2, No. 1, pp. 97-110, 2016.
- [9] Fahmi, Amanda, Wijaya, Ikhmal Aldhi Ghani, Nurulli Abdul Sugiharto, Aldhyth, “Sistem Monitoring dan Controlling Air Nutrisi Aquaponik Menggunakan Arduino Uno Berbasis Web Server”, Vol. 1, No. 1, pp. 39-46, 2016.

- [10] Andri, Rachmad, “Sistem Monitoring dan Pengendalian Suhu dan Kelembaban Ruang pada Rumah Walet Berbasis Android, Web dan SMS”, Institut Teknologi Sepuluh November, pp. 1-9, 2013.
- [11] Wahyudi, ST, “Merancang Sistem Monitoring dan Kendali Fertigasi Tanaman Aeroponik Jarak Jauh Menggunakan Web Server Embedded”, Universitas Sultan Ageng Tirtayasa, 2015.
- [12] S. Ayub, “Rancang Bangun Sistem Aeroponik Secara Otomatis Berbasis Mikrokontroler”, Universitas Komputer Indonesia, 2016.
- [13] Z. Arifin, “Pengaruh Intensitas Cahaya Matahari dan Trikontanol terhadap Pertumbuhan dan Hasil Biji Bayam”, vol. 1 no. 1, pp. 1-10, 2007.
- [14] P. Megha, “Greenhouse Carbon Dioxide supplementation”, Oklahoma State University, pp. 1-6, 2017.
- [15] Hartono, Bambang. “Sistem Informasi Manajemen Berbasis Komputer”, Jakarta: PT. Rineka Cipta, 2013
- [16] S. Adi, D. Kristin, “Strukturisasi Entity Relationship Diagram dan Data Flow Berbasis Business Event-Driven”, Vol. 5, No. 1, pp. 26-34, 2014.
- [17] A. Saputra, Ananda, Dadang Syarif, “Aplikasi Monitoring Hasil Produksi dan Pengingat Jadwal Perawatan Tanaman Kelapa Sawit dengan Perangkat Mobile Berbasis Android”, Vol. 1, pp. 1-10, 2012.
- [18] Daryanto, Fauzi, “Monitoring Lampu Koridor Gedung A, D4 PENSITS dengan Menggunakan Wirelles Sensor Network (WSN)”, Institut Teknologi Sepuluh Nopember, pp. 1-8, 2011.
- [19] N. Safaat, “Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android (Revisi Kedua)”, Informatika, Bandung, 2014.
- [20] Lutfhi, Irfan, “Pengembangan Aplikasi Historiod Berbasis Android sebagai Media Pembelajaran Sejarah siswa SMA”, Universitas Negeri Yogyakarta, pp. 1-10, 2016.
- [21] Pressman, Roger, “Software Engineering: A Practitioner’s Approach (7th Edition)”, McGraw-Hill, New York, 2010.

- [22] Herdi, Hafizh, dalam artikel “Belajar Membuat Aplikasi Android Sederhana Menggunakan Android Studio”. Tersedia dari: <http://www.twoh.co/2014/09/belajar-membuat-aplikasi-android-sederhana-menggunakan-android-studio.html>. [URL dikunjungi pada 15 Juli 2017, Jam 15:30].
- [23] N. Septadi, “Perancangan Aplikasi Pemutar Musik Beserta Penampil Lirik Musik Berbasis Android”, Universitas STIKUBANK, 2016.
- [24] “Android Studio”. Tersedia dari: <http://developer.android.com/studio/>. [URL dikunjungi pada 5 Juli 2018, Jam 15:45].
- [25] Naughton, Patrick, “Java Handbook”, diterjemahkan oleh: Panji Gotama Yogyakarta, 1996.
- [26] S. Agus, “Pembuatan Aplikasi Mobile Broadcast Informasi Perkuliahan Berbasis Android”, Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra, pp. 1-8, 2012.
- [27] I. Pamungkas, “Desain dan Implementasi Pemutar Musik Berbasis Android. Manajemen Informatika”, AMIKOM, pp. 1-9, 2013.
- [28] M. Turland, “Php Architect’s Guide to Web Scraping with PHP. Introduction-Web Scraping Defined, str, 2”, Wikipedia dalam artikel “Android Studio”. 2010. Tersedia dari: [https://en.m.wikipedia.org/wiki/Android\\_studio.html](https://en.m.wikipedia.org/wiki/Android_studio.html). [URL dikunjungi pada 20 Desember 2017, Jam 16:30].
- [29] T. Connolly, C. Begg, “Database Systems: A Practical Approach to Design, Implementation, and Management, Fourth Edition Boston”: Pearson Education, 2005.
- [30] C. Hopkins, “Jump Start PHP”, Collingwood: Sitepoint, 2013.

# **LAMPIRAN**

# FILE PHP

## 1. File connect.php

```
<?php
define('HOSTNAME', 'pramudyo.com');
define('USERNAME', 'pramudyo_greenho');
define('PASSWORD', 'greenhouse123');
define('DB_SELECT', 'pramudyo_greenhouse');
$koneksi = new
mysqli(HOSTNAME,USERNAME,PASSWORD,DB
_SELECT) or die (mysqli_errno());
?>
```

## 2. File current\_suhu.php

```
<?php
include_once('connect.php');
$query = "SELECT nilai,waktu FROM suhu ORDER
BY waktu DESC LIMIT 1";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result))
{
    $arraydata[]=$baris;
}
echo json_encode($arraydata);
?>
```

## 3. File get\_data\_suhu.php

```
<?
include_once('connect.php');
$query = "SELECT nilai,waktu FROM suhu ORDER
BY waktu DESC LIMIT 18";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result)){
    $arraydata[]=$baris;}
echo json_encode($arraydata);
?>
```

## 4. File current\_kelembaban.php

```
<?php
include_once('connect.php');
$query = "SELECT nilai,waktu FROM kelembaban
ORDER BY waktu DESC LIMIT 1";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result))
{
    $arraydata[]=$baris;
}
echo json_encode($arraydata);
?>
```

## 5. File get\_datakelembaban.php

```
<?
include_once('connect.php');
$query = "SELECT nilai,waktu FROM kelembaban
ORDER BY waktu DESC LIMIT 18";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result)){
    $arraydata[]=$baris;}
echo json_encode($arraydata);
?>
```

## 6. File current\_intensitas.php

```
<?php
include_once('connect.php');
$query = "SELECT nilai,waktu FROM intensitas
ORDER BY waktu DESC LIMIT 1";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result))
{
    $arraydata[]=$baris;
}
echo json_encode($arraydata);
?>
```

## 7. File get\_dataintensitas.php

```
<?
include_once('connect.php');
$query = "SELECT nilai,waktu FROM intensitas
ORDER BY waktu DESC LIMIT 18";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result)){
    $arraydata[]=$baris;}
echo json_encode($arraydata);
?>
```

## 8. File current\_karbon.php

```
<?php
include_once('connect.php');
$query = "SELECT nilai,waktu FROM karbon
ORDER BY waktu DESC LIMIT 1";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result))
{
    $arraydata[]=$baris;
}
echo json_encode($arraydata);
?>
```

## 9. File get\_datakarbon.php

```
<?
include_once('connect.php');
$query = "SELECT nilai,waktu FROM suhu ORDER
BY waktu DESC LIMIT 18";
$result = mysqli_query($koneksi,$query);
$arraydata = array();
while($baris = mysqli_fetch_assoc($result)){
    $arraydata[]=$baris;}
echo json_encode($arraydata);
?>
```



# APLIKASI SISTEM MONITORING

```
1. Menu Splash Screen
activity_splashscreen.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/an
droid"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@android:color/background_light"
tools:context="com.example.pita.monitoringsystem.Spl
ashscreen">

<ImageView
android:id="@+id/imageview3"
android:layout_width="326dp"
android:layout_height="193dp"
android:src="@drawable/pita"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.413"
tools:ignore="DuplicateIds,HardcodedText" />
</android.support.constraint.ConstraintLayout>

2. Menu Utama
activity_menuparameter.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/an
droid"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@android:color/white"
android:orientation="vertical"
android:paddingBottom="16dp"
android:paddingTop="16dp"
tools:context=".menu.MenuParameter">

<android.support.v4.view.ViewPager
android:id="@+id/view_pager"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_marginBottom="250dp"></android.suppor
t.v4.view.ViewPager>

<android.support.constraint.ConstraintLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="250dp">

<ImageButton
android:id="@+id/buttonintensitas"
android:layout_width="48dp"
android:layout_height="57dp"
android:background="@drawable/lux"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.211"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
tools:ignore="MissingConstraints" />

<TextView
android:id="@+id/textintensitas"
android:layout_width="wrap_content"
android:layout_height="18dp"
android:text="Intensitas Cahaya"
android:textColor="@android:color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.171"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.63" />

<ImageButton
android:id="@+id/button suhu"
android:layout_width="57dp"
android:layout_height="57dp"
android:background="@drawable/sun"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.217"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.150"
tools:ignore="MissingConstraints" />

<TextView
android:id="@+id/textsuhu"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Suhu"
android:textColor="@android:color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.226"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.319" />

<ImageButton
android:id="@+id/button kelembaban"
android:layout_width="54dp"
android:layout_height="49dp"
android:layout_marginTop="36dp"
android:background="@drawable/humidity"
app:layout_constraintHorizontal_bias="0.857"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
tools:ignore="MissingConstraints" />

<TextView
android:id="@+id/textkelembaban"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Kelembaban"
android:textColor="@android:color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.875"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.319" />

<ImageButton
android:id="@+id/button karbon"
android:layout_width="56dp"
android:layout_height="57dp"
android:layout_marginBottom="12dp"
android:background="@drawable/codua"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.862"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

Splashscreen.java
package com.example.pita.monitoringsystem;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import
com.example.pita.monitoringsystem.menu.MenuParameter
;
public class Splashscreen extends Activity {
@Override
protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);

setContent(R.layout.activity_splashscreen);
Thread thread = new Thread() {
public void run() {
try{
sleep(2000);
} catch (InterruptedException e) {
e.printStackTrace();
} finally {
startActivity(new
Intent(Splashscreen.this, MenuParameter.class));
finish();
}};thread.start(); }
```

```

app:layout_constraintVertical_bias="0.537" />

<TextView
android:id="@+id/textkarbon"
android:layout_width="wrap_content"
android:layout_height="15dp"
android:text="Karbondioksida"
android:textColor="@android:color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.906"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.621" />

<ImageButton
android:id="@+id/buttonaeroponik"
android:layout_width="71dp"
android:layout_height="57dp"
android:background="@drawable/skripsi"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.226"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.831" />

<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="19dp"
android:text="Diskusi Aeroponik"
android:textColor="@android:color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.171"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.923" />

<ImageButton
android:id="@+id/buttonexit"
android:layout_width="65dp"
android:layout_height="51dp"
android:layout_marginRight="36dp"
android:background="@drawable/sk"
android:onClick="Clickexit"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.803" />

<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Keluar"
android:textColor="@android:color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.86"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.923" />
</android.support.constraint.ConstraintLayout>
</RelativeLayout>

```

```

MenuParameter.java
package com.example.pita.monitoringsystem.menu;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.view.ViewPager;
import android.view.View;
import android.widget.ImageButton;
import com.example.pita.monitoringsystem.R;
import java.util.Timer;
import java.util.TimerTask;
public class MenuParameter extends Activity {
    ViewPager viewPager;
    CustomSwipeAdapter2 adapter;
    ImageButton pindah;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menuparameter);
    viewPager = (ViewPager)
    findViewById(R.id.view_pager);
    adapter = new CustomSwipeAdapter2(this);
    viewPager.setAdapter(adapter);
}
3. Image Slider

```

```

Timer timer = new Timer ();
timer.scheduleAtFixedRate(new MyTimerTask(), 2000,
3000);

pindah = (ImageButton)
findViewById(R.id.button_suhu);
pindah.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v) {
Intent intent = new Intent(MenuParameter.this,
MenuSuhu.class);
startActivity(intent); });
pindah = (ImageButton)
findViewById(R.id.buttonkarbon);
pindah.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v) {
Intent intent = new Intent(MenuParameter.this,
MenuKarbon.class);
startActivity(intent); });
pindah = (ImageButton)
findViewById(R.id.buttonintensitas);
pindah.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v) {
Intent intent = new Intent(MenuParameter.this,
MenuIntensitas.class);
startActivity(intent); });
pindah = (ImageButton)
findViewById(R.id.buttonkelembaban);
pindah.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v) {
Intent intent = new Intent(MenuParameter.this,
MenuKelembaban.class);
startActivity(intent); });
pindah = (ImageButton)
findViewById(R.id.buttonaeroponik);
pindah.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v) {
Intent intent = new Intent(MenuParameter.this,
MainQuestion.class);
startActivity(intent); });
}

public class MyTimerTask extends TimerTask{
@Override
public void run() {
MenuParameter.this.runOnUiThread(new
Runnable() {
@Override
public void run()
{if (viewPager.getCurrentItem()==0) {
viewPager.setCurrentItem(1);
} else if (viewPager.getCurrentItem() == 1) {
viewPager.setCurrentItem(2);
} else if (viewPager.getCurrentItem() == 2) {
viewPager.setCurrentItem(3);
} else if (viewPager.getCurrentItem() == 3) {
viewPager.setCurrentItem(4);
} else {viewPager.setCurrentItem(0); } } });
}

public void Clickexit(View v) {
//exit dari aplikasi dengan konfirmasi AlertDialog
AlertDialog.Builder builder = new
AlertDialog.Builder(MenuParameter.this);
builder.setMessage("Apakah ingin keluar dari
aplikasi?");
builder.setCancelable(true);
builder.setPositiveButton("Iya", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int id) {
finish(); });
builder.setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int id) {dialog.cancel(); });
AlertDialog
alert = builder.create();
alert.show();
}
}
}

```

activity\_swipelayout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:gravity="top"
android:orientation="vertical"
android:layout_height="wrap_content">
```

```
<ImageView
android:id="@+id/image_view"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="25dp"
android:contentDescription="@string/todo" />
```

```
</LinearLayout>
```

```
CustomSwipeAdapter2.java
```

```
package com.example.pita.monitoringsystem.menu;
import android.content.Context;
import android.support.v4.view.PagerAdapter;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import com.example.pita.monitoringsystem.R;

public class CustomSwipeAdapter2 extends
PagerAdapter {
    private int[] image_resource
    = {R.drawable.satu, R.drawable.two, R.drawable.three, R.
```

#### 4. Menu Suhu

```
activity_menusuhu.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/globalbg"
tools:context=".menu.MenuSuhu">
```

```
<Button
android:id="@+id/button"
android:layout_width="150dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:text="Suhu"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.068"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.23" />
```

```
<ImageView
android:id="@+id/imageView5"
android:layout_width="wrap_content"
android:layout_height="456dp"
app:srcCompat="@drawable/suhhu"
tools:layout_editor_absoluteX="0dp"
tools:layout_editor_absoluteY="0dp"
tools:ignore="MissingConstraints" />
```

```
<Button
android:id="@+id/button2"
android:layout_width="150dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:text="grafik suhu"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.602"
app:layout_constraintStart_toEndOf="@+id/button"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.23" />
</android.support.constraint.ConstraintLayout>
```

```
MenuSuhu.java
```

```
package com.example.pita.monitoringsystem.menu;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import
com.example.pita.monitoringsystem.grafik.GrafikSuhu;
import com.example.pita.monitoringsystem.R;
```

```
drawable.empat, R.drawable.lima);
private Context ctx;
private LayoutInflater inflater;
public CustomSwipeAdapter2(Context ctx)
this.ctx= ctx;
@Override
public int getCount() {return
image_resource.length;}

@Override
public boolean isVisibleFromObject(View view,
Object o) {return (view==(LinearLayout)o);}

@Override
public Object instantiateItem(ViewGroup
container, int position) {
    inflater = (LayoutInflater)
ctx.getSystemService(Context.LAYOUT_INFLATER_SERVICE
);
    View item_view =
inflater.inflate(R.layout.activity_swipelayout
, container, false);
    ImageView imageView = (ImageView)
item_view.findViewById(R.id.image_view);
    imageView.setImageResource(image_resource[position]);
    container.addView(item_view);
    return item_view;
}
@Override
public void destroyItem(ViewGroup container, int
position, Object object) {
    container.removeView((LinearLayout)object);
```

```
import
com.example.pita.monitoringsystem.activity.Suhu_Acti
vity;
public class MenuSuhu extends AppCompatActivity
implements View.OnClickListener {
    Button suhu, grafik;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_menusuhu);
suhu = findViewById(R.id.button);
grafik = findViewById(R.id.button2);
suhu.setOnClickListener(this);
grafik.setOnClickListener(this);
@Override
public void onClick(View v) {
if(v == suhu){
startActivity(new Intent(getApplicationContext(),
Suhu_Activity.class));}else if(v ==
grafik){startActivity(new
Intent(getApplicationContext(),
GrafikSuhu.class);}}}
```

```
activity_suhu.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".activity.Suhu_Activity">
```

```
<ImageView
android:id="@+id/imageView6"
android:layout_width="wrap_content"
android:layout_height="321dp"
app:srcCompat="@drawable/suhuu" />
```

```
<android.support.v7.widget.RecyclerView
android:id="@+id/list_suhu"
android:layout_width="368dp"
android:layout_height="151dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent" />
```

```
</android.support.constraint.ConstraintLayout>
```

```
Suhu_Activity.java
```

```
package com.example.pita.monitoringsystem.activity;
import android.app.ProgressDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

```

import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.adapter.AdapterSuhu;
import com.example.pita.monitoringsystem.model.Suhu;
import
com.example.pita.monitoringsystem.util.AppController;
import
com.example.pita.monitoringsystem.util.ServerAPI;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
public class Suhu_Activity extends AppCompatActivity {
    RecyclerView mRecyclerView;
    RecyclerView.Adapter mAdapter;
    RecyclerView.LayoutManager mManager;
    List<Suhu> mItems;
    ProgressDialog pd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_suhu);
        mRecyclerView = (RecyclerView)
        findViewById(R.id.list_suhu);
        pd = new ProgressDialog(Suhu_Activity.this);
        mItems = new ArrayList<>();
        loadJson();
        mManager = new LinearLayoutManager(Suhu_Activity.this,
        LinearLayoutManager.VERTICAL, false);
        mRecyclerView.setLayoutManager(mManager);
        mAdapter = new AdapterSuhu(Suhu_Activity.this, mItems);
        mRecyclerView.setAdapter(mAdapter);
        private void loadJson() {
            pd.setMessage("Mengambil Data");
            pd.setCancelable(false);
            pd.show();
            JsonArrayRequest reqData = new
            JsonArrayRequest(Request.Method.POST,
            ServerAPI.URL_DATASUHU, null, new
            Response.Listener<JSONArray>() {
                @Override
                public void onResponse(JSONArray response) {
                    pd.cancel();
                    Log.d("volley", "response : " + response.toString());
                    for (int i = 0 ; i < response.length(); i++) {
                        try {JSONObject data = response.getJSONObject(i);
                            Suhu md = new Suhu();
                            md.setNilai(data.getString("nilai"));
                            md.setWaktu(data.getString("waktu"));
                            mItems.add(md);} catch (JSONException e) {
                                e.printStackTrace();}
                            mAdapter.notifyDataSetChanged();},
                            new Response.ErrorListener() {
                                @Override
                                public void onErrorResponse(VolleyError error) {
                                    pd.cancel();Log.d("volley", "error : " +
                                    error.getMessage());});
                                AppController.getInstance().addToRequestQueue(reqData)
                                ;}}

```

```

parameter_suhu.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">
<TextView
android:id="@+id/txtSuhu"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:gravity="center"
android:text="@string/currentDummySuhu"

```

Suhu.java

```

android:textAppearance="?android:attr/textAppearanceLarge"
android:textColor="@android:color/background_dark"
android:textSize="40sp"
android:textStyle="bold"
android:typeface="normal" />

```

```

<TextView
android:id="@+id/txtTitleLastUpdate"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:gravity="center"
android:text="@string/currentLastUpdate"
android:textAppearance="?android:attr/textAppearanceMedium"
android:textSize="15sp" />
<TextView
    android:id="@+id/txtWaktu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/currentDummyDate"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textSize="16sp" />
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

AdapterSuhu.java

```

package com.example.pita.monitoringsystem.adapter;

```

```

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

```

```

import com.example.pita.monitoringsystem.R;
import com.example.pita.monitoringsystem.model.Suhu;

```

```

import java.util.List;

```

```

public class AdapterSuhu extends

```

```

RecyclerView.Adapter<AdapterSuhu.HolderData> {

```

```

    private List<Suhu> mItems ;
    private Context context;

```

```

    public AdapterSuhu(Context context, List<Suhu>
items)

```

```

    {
        this.mItems = items;
        this.context = context;
    }

```

```

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        View layout =
        LayoutInflater.from(parent.getContext()).inflate(R.lay
out.parameter_suhu, parent, false);
        HolderData holderData = new
        HolderData(layout);
        return holderData;
    }

```

```

    @Override
    public void onBindViewHolder(@NonNull HolderData
holder, int position) {

```

```

        Suhu md = mItems.get(position);
        holder.t_Suhu.setText(md.getNilai());
        holder.t_Waktu.setText(md.getWaktu());

```

```

        holder.md = md;

```

```

    @Override
    public int getItemCount() {
        return mItems.size();
    }

```

```

class HolderData extends RecyclerView.ViewHolder{
    TextView t_Suhu, t_Waktu;
    Suhu md;
    public HolderData (View view){
        super(view);t_Suhu = (TextView)
view.findViewById(R.id.txtSuhu);
        t_Waktu = (TextView)
view.findViewById(R.id.txtWaktu);}
}

```

```

package com.example.pita.monitoringsystem.model;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

```

```

import com.example.pita.monitoringsystem.R;
public class Suhu extends AppCompatActivity {
    String nilai,waktu;
    public Suhu(){
    public Suhu(String nilai, String waktu) {
        this.nilai = nilai;
        this.waktu = waktu;}
activity_grafik_suhu.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res-
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".grafik.GrafikSuhu">

<com.github.mikephil.charting.charts.LineChart
android:id="@+id/chart"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
/>
</android.support.constraint.ConstraintLayout>

```

#### GrafikSuhu.java

```

package com.example.pita.monitoringsystem.grafik;
import android.app.ProgressDialog;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.pita.monitoringsystem.R;
import com.example.pita.monitoringsystem.util.ServerAPI;
import com.github.mikephil.charting.charts.BarChart;
import
com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.BarData;
import com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import
com.github.mikephil.charting.data.LineDataSet;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;

```

```

public class GrafikSuhu extends AppCompatActivity {
    private ProgressDialog pd;
    ArrayList<LineDataSet> yAxis;
    ArrayList<Entry> yValues;
    ArrayList<String> xAxis;
    Entry values ;
    LineChart chart;
    LineData data;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_grafik_suhu);
        pd = new ProgressDialog(GrafikSuhu.this);
        pd.setMessage("loading");
        chart = (LineChart) findViewById(R.id.chart);
        load_data_from_server();
        public void load_data_from_server(){
            pd.show();
            String url = ServerAPI.URL_GRAFIK_SUHU;
            xAxis1 = new ArrayList<>();
            yAxis = null;
            yValues = new ArrayList<>();

            StringRequest stringRequest = new
            StringRequest(Request.Method.POST, url, new
            Response.Listener<String>() {

```

#### activity\_menulembaban.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"

```

```

public String getNilai() {return nilai;}
public void setNilai(String nilai) {this.nilai =
nilai;}
public String getWaktu() {return waktu;}
public void setWaktu(String waktu) {
    this.waktu = waktu;}

@Override
public void onResponse(String response) {
    Log.d("String",response);
    try{JSONArray jsonArray = new JSONArray(response);
    for (int i=0; i<jsonArray.length(); i++){
        JSONObject jsonObject = jsonArray.getJSONObject(i);
        String suhu = jsonObject.getString("nilai").trim();
        String waktu = jsonObject.getString("waktu").trim();
        xAxis1.add(waktu);
        values =
        new Entry(Float.valueOf(suhu),i);
        yValues.add(values); }
    }catch (JSONException e){
        e.printStackTrace();}
    LineDataSet lineDataSet = new
    LineDataSet(yValues,"Grafik Suhu (18 data
    terakhir)");
    lineDataSet.setColor(Color.rgb(0,82,159));
    yAxis = new ArrayList<>();
    yAxis.add(lineDataSet);
    String names[] = xAxis1.toArray(new
    String[xAxis1.size()]);
    data = new LineData(xAxis1,lineDataSet);
    chart.setData(data);
    chart.setDescription("");
    chart.animateXY(2000,2000);
    chart.invalidate();
    pd.hide();
    }},
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        if(error != null){
            Toast.makeText(getApplicationContext(), "Ups,
            terjadi kesalahan.", Toast.LENGTH_LONG).show();
            pd.hide(); }
        }
    }
    MySingletonsuhu.getInstance(getApplicationContext())
    .addToRequestQueue(stringRequest);}

```

#### MySingletonsuhu.java

```

package com.example.pita.monitoringsystem.grafik;
import android.content.Context;
import android.graphics.Bitmap;
import android.support.v4.util.LruCache;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;
public class MySingletonsuhu {
    private static MySingletonsuhu mInstance;
    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;
    private static Context mContext;
    private MySingletonsuhu(Context context) {
        mContext = context;
        mRequestQueue = getRequestQueue();
        mImageLoader = new
        ImageLoader(mRequestQueue,new
        ImageLoader.ImageCache() {private final
        LruCache<String, Bitmap> cache = new
        LruCache<String, Bitmap>(20);
        @Override
        public Bitmap getBitmap(String url) {
            return cache.get(url);}
        @Override
        public void putBitmap(String url, Bitmap bitmap) {
            cache.put(url, bitmap); }
    });
    public static synchronized MySingletonsuhu
    getInstance(Context context) {if (mInstance ==
    null) {mInstance = new MySingletonsuhu(context);
    }return mInstance;}
    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null){mRequestQueue =
        Volley.newRequestQueue(mContext.getApplicationContext(
        )); }return mRequestQueue;}
    public <T> void addToRequestQueue(Request<T> req)
    {
        getRequestQueue().add(req);}
    public ImageLoader getImageLoader() {return
    mImageLoader;}
}

```

#### 5. Menu Kelembaban

```

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/globalbg"

```

```

tools:context=".menu.MenuSuhu">

<Button
android:id="@+id/button1"
android:layout_width="171dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:background="@drawable/globalbg"
android:text="Kelembaban"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.032"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.293" />

<Button
android:id="@+id/button2"
android:layout_width="170dp"
android:layout_height="wrap_content"
android:layout_marginEnd="4dp"
android:layout_marginTop="16dp"
android:background="@drawable/globalbg"
android:text="grafik Kelembaban"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.691"
app:layout_constraintStart_toEndOf="@+id/button"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.293" />

<ImageView
android:id="@+id/imageView2"
android:layout_width="387dp"
android:layout_height="310dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.97"
app:srcCompat="@drawable/kelem"
tools:ignore="MissingConstraints,RtlCompat" />
</android.support.constraint.ConstraintLayout>

MenuKelembaban.java
package com.example.pita.monitoringsystem.menu;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import com.example.pita.monitoringsystem.R;
import com.example.pita.monitoringsystem.activity.Kelembaban_Activity;
import com.example.pita.monitoringsystem.grafik.GrafikKelembaban;
public class MenuKelembaban extends AppCompatActivity
implements View.OnClickListener {
    Button kelembaban, grafik;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menukelembaban);

        kelembaban = findViewById(R.id.button1);
        grafik = findViewById(R.id.button2);
        kelembaban.setOnClickListener(this);
        grafik.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {if(v == kelembaban){
startActivity(new Intent(getApplicationContext(),
Kelembaban_Activity.class));}
else if(v == grafik){startActivity(new
Intent(getApplicationContext(),
GrafikKelembaban.class);}}}

activity_kelembaban.xml
<?xml version="1.0" encoding="utf-8" ?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".activity.Suhu_Activity">

<ImageView
android:id="@+id/imageView"
android:layout_width="391dp"
android:layout_height="wrap_content"
app:srcCompat="@drawable/humidity" />

```

```

<android.support.v7.widget.RecyclerView
android:id="@+id/list_kelembaban"
android:layout_width="368dp"
android:layout_height="208dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>

Kelembaban_Activity.java
package com.example.pita.monitoringsystem.activity;

import android.app.ProgressDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.example.pita.monitoringsystem.R;
import com.example.pita.monitoringsystem.adapter.AdapterKelembaban;
import com.example.pita.monitoringsystem.model.Kelembaban;
import com.example.pita.monitoringsystem.util.AppController;
import com.example.pita.monitoringsystem.util.ServerAPI;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class Kelembaban_Activity extends
AppCompatActivity {

    RecyclerView mRecyclerview;
    RecyclerView.Adapter mAdapter;
    RecyclerView.LayoutManager mManager;
    List<Kelembaban> mItems;
    ProgressDialog pd;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelembaban);

        mRecyclerview = (RecyclerView)
findViewById(R.id.list_kelembaban);

        pd = new
ProgressDialog(Kelembaban_Activity.this);
        mItems = new ArrayList<Kelembaban>();

        loadJson();

        mManager = new
LinearLayoutManager(Kelembaban_Activity.this,
LinearLayoutManager.VERTICAL, false);
        mRecyclerview.setLayoutManager(mManager);
        mAdapter = new
AdapterKelembaban(Kelembaban_Activity.this, mItems);
        mRecyclerview.setAdapter(mAdapter);
    }

    private void loadJson()
    {
        pd.setMessage("Mengambil Data");
        pd.setCancelable(false);
        pd.show();

        JsonArrayRequest reqData = new
JSONArrayRequest(Request.Method.POST,
ServerAPI.URL_DATAKELEMBABAN, null,
new Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray
response) {
                pd.cancel();
                Log.d("volley", "response : " +
response.toString());
                for(int i = 0 ; i <

```

```

response.length(); i++)
    {
        try {
            JSONObject data =
                response.getJSONObject(i);
            Kelembaban md = new
                Kelembaban();
            md.setNilai(data.getString("nilai"));
            md.setWaktu(data.getString("waktu"));
            mItems.add(md);
        } catch (JSONException e)
        {
            e.printStackTrace();
        }
    }
mAdapter.notifyDataSetChanged();
},
new Response.ErrorListener() {
    @Override
    public void
onErrorResponse (VolleyError error) {
    pd.cancel();
    Log.d("volley", "error : " +
error.getMessage());
}
});

AppController.getInstance().addToRequestQueue(reqData)
;

parameter_kelembaban.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
        <TextView
            android:id="@+id/txtkelembaban"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:gravity="center"
            android:text="@string/currentDummyKelembaban"
            android:textAppearance="?android:attr/textAppearan
ceLarge"
            android:textColor="@android:color/background_dark"
            android:textSize="40sp"
            android:textStyle="bold"
            android:typeface="normal" />
        <TextView
            android:id="@+id/txtTitleLastUpdate"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:gravity="center"
            android:text="@string/currentLastUpdate"
            android:textAppearance="?android:attr/textAppearan
ceMedium"
            android:textSize="15sp" />
    </LinearLayout>
</android.support.constraint.ConstraintLayout>

Kelembaban.java
package com.example.pita.monitoringsystem.model;
import android.support.v7.app.AppCompatActivity;
public class Kelembaban extends AppCompatActivity
{
    String nilai,waktu;
    public Kelembaban () {}
    public Kelembaban(String nilai, String waktu)
{
    this.nilai = nilai;
    this.waktu = waktu;}
    public String getNilai() {return nilai;}
    public void setNilai(String nilai) {this.nilai
= nilai;}
    public String getWaktu() {
    return waktu; }
    public void setWaktu(String waktu) {
    this.waktu = waktu;}
}

AdapterKelembaban.java
package com.example.pita.monitoringsystem.adapter;
import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.model.Kelembaban
;
import java.util.List;
public class AdapterKelembaban extends
RecyclerView.Adapter<AdapterKelembaban.HolderData>
{
    private List<Kelembaban> mItems ;
    private Context context;
    public AdapterKelembaban(Context context,
List<Kelembaban> items){
    this.mItems = items; this.context = context;}
    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {View layout =
LayoutInflater.from(parent.getContext()).inflate(R
.layout.parameter_kelembaban, parent, false);
    HolderData holderData = new
    HolderData(layout);
    return holderData;
}
    @Override
    public void onBindViewHolder(@NonNull
HolderData holder, int position) {
    Kelembaban md = mItems.get(position);
    holder.t_Kelembaban.setText(md.getNilai());
    holder.t_Waktu.setText(md.getWaktu());
    holder.md = md;
}
    @Override
    public int getItemCount() {
    return mItems.size();
}
    class HolderData extends
RecyclerView.ViewHolder
    {
        TextView t_Kelembaban,t_Waktu;
        Kelembaban md;
        public HolderData (View view)
        {
            super (view);
            t_Kelembaban = (TextView)
view.findViewById(R.id.txtkelembaban);
            t_Waktu = (TextView)
view.findViewById(R.id.txtWaktu);}}
}

activity_grafik_kelembaban.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".grafik.GrafikKelembaban">
<com.github.mikephil.charting.charts.LineChart
android:id="@+id/chart1"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

```

```

/>
</android.support.constraint.ConstraintLayout>

MySingletonkelembaban.java
package com.example.pita.monitoringsystem.grafik;

import android.content.Context;
import android.graphics.Bitmap;
import android.support.v4.util.LruCache;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;

public class MySingletonkelembaban {

    private static MySingletonkelembaban mInstance;
    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;
    private static Context mContext;

    private MySingletonkelembaban(Context context) {
        mContext = context;
        mRequestQueue = getRequestQueue();

        mImageLoader = new
ImageLoader(mRequestQueue,
            new ImageLoader.ImageCache() {
                private final LruCache<String,
Bitmap>
                    cache = new
LruCache<String, Bitmap>(20);

```

```

GrafikKelembaban.java
package com.example.pita.monitoringsystem.grafik;
import android.app.ProgressDialog;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.pita.monitoringsystem.R;
import com.example.pita.monitoringsystem.util.ServerAPI;
import com.github.mikephil.charting.charts.BarChart;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.BarData;
import com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class GrafikKelembaban extends
AppCompatActivity {
    private ProgressDialog pd;
    ArrayList<LineDataSet> yAxis;
    ArrayList<Entry> yValues;
    ArrayList<String> xAxis1;
    Entry values;
    LineChart chart;
    LineData data;

    @Override
    protected void onCreate(Bundle savedInstanceState)
{super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_grafik_kelembaban
); pd = new ProgressDialog(GrafikKelembaban.this);
    pd.setMessage("loading");
    chart = (LineChart) findViewById(R.id.chart1);

```

## 6. Menu Intensitas Cahaya

```

activity_menu_intensitas.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"

```

```

@Override
public Bitmap getBitmap(String
url) {
    return cache.get(url);
}

@Override
public void putBitmap(String
url, Bitmap bitmap) {
    cache.put(url, bitmap);
}
});
public static synchronized MySingletonkelembaban
getInstance(Context context) {
    if (mInstance == null) {
        mInstance = new
MySingletonkelembaban(context);
    }
    return mInstance;
}

public RequestQueue getRequestQueue() {
    if (mRequestQueue == null) {
        mRequestQueue =
Volley.newRequestQueue(mContext.getApplicationContext())
;
    }
    return mRequestQueue;
}

public <T> void addToRequestQueue(Request<T>
req) {
    getRequestQueue().add(req);
}

public ImageLoader getImageLoader() {
    return mImageLoader;
}

load_data_from_server();
public void load_data_from_server(){
pd.show();
String url = ServerAPI.URL_GRAFIK_KELEMBABAN;
xAxis1 = new ArrayList<>();
yAxis = null;
yValues = new ArrayList<>();
StringRequest stringRequest = new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
@Override
public void onResponse(String response) {
Log.d("String", response);
try{
JSONArray jsonArray = new JSONArray(response);
for (int i=0; i<jsonArray.length(); i++){
JSONObject jsonObject =
jsonArray.getJSONObject(i);
String kelembaban =
jsonObject.getString("nilai").trim();
String waktu =
jsonObject.getString("waktu").trim();
xAxis1.add(waktu);
values = new Entry(Float.valueOf(kelembaban), i);
yValues.add(values);}}catch (JSONException e){
e.printStackTrace();}LineDataSet lineDataSet = new
LineDataSet(yValues, "Grafik Kelembaban (18 data
terakhir)");
lineDataSet.setColor(Color.rgb(0, 82, 159));
yAxis = new ArrayList<>();
yAxis.add(lineDataSet);
String names[] =
xAxis1.toArray(new String[xAxis1.size()]);
data = new
LineData(xAxis1, lineDataSet);
chart.setData(data);
chart.setDescription("");
chart.animateXY(2000, 2000);
chart.invalidate();
pd.hide();}},
new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
if(error != null)
{Toast.makeText(getApplicationContext(), "Ups,
terjadi kesalahan.", Toast.LENGTH_LONG).show();
pd.hide();}}});

MySingletonkelembaban.getInstance(getApplicationContext()
ntext()).addToRequestQueue(stringRequest);

```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/globalbg"
tools:context=".menu.MenuSuhu">

```

```
<Button
```



```

android:id="@+id/button1"
android:layout_width="171dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:text="Intensitas Cahaya"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.075"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.293" />

<Button
android:id="@+id/button2"
android:layout_width="170dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:text="Grafik Intensitas Cahaya"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.691"
app:layout_constraintStart_toEndOf="@+id/button"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.293" />

<ImageView
android:id="@+id/imageView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
app:srcCompat="@drawable/cahaya" />
</android.support.constraint.ConstraintLayout>

MenuIntensitas.java
package com.example.pita.monitoringsystem.menu;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.activity.Intensitas_Activity;
import
com.example.pita.monitoringsystem.grafik.GrafikIntensitas;
public class MenuIntensitas extends
AppCompatActivity implements View.OnClickListener
{
    Button intensitas, grafik;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu_intensitas);
        intensitas = findViewById(R.id.button1);
        grafik = findViewById(R.id.button2);
        intensitas.setOnClickListener(this);
        grafik.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {if(v == intensitas){
        startActivity(new Intent(getApplicationContext(),
        Intensitas_Activity.class));}else if(v == grafik){
        startActivity(new Intent(getApplicationContext(),
        GrafikIntensitas.class);}}

activity_intensitas.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".activity.Suhu_Activity">

<ImageView
android:id="@+id/imageView"
android:layout_width="391dp"
android:layout_height="272dp"
app:srcCompat="@drawable/lux" />

<android.support.v7.widget.RecyclerView
android:id="@+id/list_intensitas"
android:layout_width="368dp"
android:layout_height="217dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
app:layout_constraintBottom_toBottomOf="parent"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
/>
</android.support.constraint.ConstraintLayout>

Intensitas_Activity.java
package
com.example.pita.monitoringsystem.activity;
import android.app.ProgressDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import
android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.JsonArrayRequest;
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.adapter.AdapterIntensitas;
import
com.example.pita.monitoringsystem.adapter.AdapterKelembaban;
import
com.example.pita.monitoringsystem.model.Intensitas;
import
com.example.pita.monitoringsystem.model.Kelembaban;
import
com.example.pita.monitoringsystem.util.AppController;
import
com.example.pita.monitoringsystem.util.ServerAPI;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
public class Intensitas_Activity extends
AppCompatActivity {
    RecyclerView mRecyclerview;
    RecyclerView.Adapter mAdapter;
    RecyclerView.LayoutManager mManager;
    List<Intensitas> mItems;
    ProgressDialog pd;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intensitas);
        mRecyclerview = (RecyclerView)
        findViewById(R.id.list_intensitas);
        pd = new ProgressDialog(Intensitas_Activity.this);
        mItems = new ArrayList<Intensitas>();
        loadJson();

        mManager = new
        LinearLayoutManager(Intensitas_Activity.this,
        LinearLayoutManager.VERTICAL, false);
        mRecyclerview.setLayoutManager(mManager);
        mAdapter = new
        AdapterIntensitas(Intensitas_Activity.this, mItems);
        mRecyclerview.setAdapter(mAdapter);

        private void loadJson() {
            pd.setMessage("Mengambil Data");
            pd.setCancelable(false);
            pd.show();

            JsonArrayRequest reqData = new
            JsonArrayRequest(Request.Method.POST,
            ServerAPI.URL_DATAINTENSITAS, null,
            new Response.Listener<JSONArray>() {
                @Override
                public void onResponse(JSONArray response) {
                    pd.cancel();
                    Log.d("volley", "response : " +
                    response.toString());
                    for(int i = 0 ; i < response.length(); i++){try {
                        JSONObject data = response.getJSONObject(i);
                        Intensitas md = new Intensitas();
                        md.setNilai(data.getString("nilai"));
                        md.setWaktu(data.getString("waktu"));
                        mItems.add(md); } catch (JSONException e) {
                            e.printStackTrace();}
                    mAdapter.notifyDataSetChanged();},
                    new Response.ErrorListener() {
                        @Override
                        public void onErrorResponse(VolleyError error) {
                            pd.cancel();Log.d("volley", "error : " +
                            error.getMessage());});}

```

```
AppController.getInstance().addToRequestQueue(reqData);}
```

```
parameter_intensitas.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">
```

```
<TextView
android:id="@+id/txtintensitas"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:gravity="center"
android:text="@string/currentDummyIntensitas"
android:textAppearance="@android:attr/textAppearance
ceLarge"
android:textColor="@android:color/background_dark"
android:textSize="40sp"
android:textStyle="bold"
android:typeface="normal" />
```

```
<TextView
android:id="@+id/txtTitleLastUpdate"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:gravity="center"
android:text="@string/currentLastUpdate"
android:textAppearance="@android:attr/textAppearance
ceMedium"
android:textSize="15sp" />
```

```
<TextView
android:id="@+id/txtWaktu"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:text="@string/currentDummyDate"
android:textAppearance="@android:attr/textAppearance
ceMedium"
android:textSize="16sp" />
```

```
</LinearLayout>
</android.support.constraint.ConstraintLayout>
```

```
Intensitas.java
package com.example.pita.monitoringsystem.model;
import android.support.v7.app.AppCompatActivity;
public class Intensitas extends AppCompatActivity {
String nilai, waktu;
public Intensitas() {}
public Intensitas(String nilai, String waktu) {
this.nilai = nilai;
this.waktu = waktu;
public String getNilai() {
return nilai;
public void setNilai(String nilai) {
```

```
activity_grafik_intensitas.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".grafik.GrafikIntensitas">
```

```
<com.github.mikephil.charting.charts.LineChart
android:id="@+id/charti"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

```
this.nilai = nilai;
public String getWaktu() {
return waktu;
public void setWaktu(String waktu) {
this.waktu = waktu;}
```

```
AdapterIntensitas.java
package com.example.pita.monitoringsystem.adapter;
```

```
import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
```

```
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.model.Intensitas
;
```

```
import java.util.List;
```

```
public class AdapterIntensitas extends
RecyclerView.Adapter<AdapterIntensitas.ViewHolder>
{
```

```
private List<Intensitas> mItems ;
private Context context;
```

```
public AdapterIntensitas(Context context,
List<Intensitas> items)
{
this.mItems = items;
this.context = context;
}
```

```
@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
View layout =
LayoutInflater.from(parent.getContext()).inflate(R
.layout.parameter_intensitas, parent, false);
ViewHolder holderData = new
ViewHolder(layout);
return holderData;
}
```

```
@Override
public void onBindViewHolder(@NonNull
ViewHolder holder, int position) {
Intensitas md = mItems.get(position);
holder.t_Intensitas.setText(md.getNilai());
holder.t_Waktu.setText(md.getWaktu());
holder.md = md;}
```

```
@Override
public int getItemCount() {return mItems.size();}
class ViewHolder extends RecyclerView.ViewHolder
{TextView t_Intensitas, t_Waktu;
Intensitas md;
public ViewHolder(View view){
super(view);
t_Intensitas = (TextView)
view.findViewById(R.id.txtintensitas);
t_Waktu = (TextView)
view.findViewById(R.id.txtWaktu);}}
```

```
GrafikIntensitas.java
package com.example.pita.monitoringsystem.grafik;
```

```
import android.app.ProgressDialog;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;
```

```
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.util.ServerAPI;
import
com.github.mikephil.charting.charts.BarChart;
import
com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.BarData;
import
com.github.mikephil.charting.data.BarDataSet;
```

```

import com.github.mikephil.charting.data.BarEntry;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class GrafikIntensitas extends
AppCompatActivity {
    private ProgressDialog pd;

    ArrayList<LineDataSet> yAxis;
    ArrayList<Entry> yValues;
    ArrayList<String> xAxis1;
    Entry values;
    LineChart chart;
    LineData data;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_grafik_intensitas
);

        pd = new
ProgressDialog(GrafikIntensitas.this);
        pd.setMessage("Loading");

        chart = (LineChart)
findViewById(R.id.chart1);
        load_data_from_server();
    }

    public void load_data_from_server() {
        pd.show();
        String url =
ServerAPI.URL_GRAFIK_INTENSITAS;
        xAxis1 = new ArrayList<>();
        yAxis = null;
        yValues = new ArrayList<>();

        StringRequest stringRequest = new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String
response) {
                Log.d("String", response);

                try{
                    JSONArray jsonArray = new
JSONArray(response);
                    for (int i=0;
i<jsonArray.length(); i++){
                        JSONObject jsonObject =
jsonArray.getJSONObject(i);
                        String intensitas =
jsonObject.getString("nilai").trim();
                        String waktu =
jsonObject.getString("waktu").trim();
                        xAxis1.add(waktu);
                        values = new
Entry(Float.valueOf(intensitas), i);
                        yValues.add(values);
                    }
                } catch (JSONException e){
                    e.printStackTrace();
                }
                LineDataSet lineDataSet = new
LineDataSet(yValues, "Grafik Intensitas (18 data
terakhir)");

                lineDataSet.setColor(Color.rgb(0,82,159));
                yAxis = new ArrayList<>();
                yAxis.add(lineDataSet);
                String names[] =
xAxis1.toArray(new String[xAxis1.size()]);
                data = new
LineData(xAxis1, lineDataSet);
                chart.setData(data);
                chart.setDescription("");
                chart.animateXY(2000,2000);

                chart.invalidate();
                pd.hide();
            }
        },
        new Response.ErrorListener() {
            @Override
            public void
onErrorResponse(VolleyError error) {
                if(error != null){

                    Toast.makeText(getApplicationContext(), "Ups,
terjadi kesalahan.", Toast.LENGTH_LONG).show();
                    pd.hide();
                }
            }
        });

        MySingletonintensitas.getInstance(getApplicationContext()
ntext()).addToRequestQueue(stringRequest);}

MySingletonintensitas.java
package com.example.pita.monitoringsystem.grafik;
import android.content.Context;
import android.graphics.Bitmap;
import android.support.v4.util.LruCache;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;
public class MySingletonintensitas {
    private static MySingletonintensitas
mInstance;
    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;
    private static Context mContext;

    private MySingletonintensitas(Context context)
{
        mContext = context;
        mRequestQueue = getRequestQueue();

        mImageLoader = new
ImageLoader(mRequestQueue,
new ImageLoader.ImageCache() {
            private final LruCache<String,
Bitmap>
cache = new
LruCache<String, Bitmap>(20);

            @Override
            public Bitmap getBitmap(String
url) {
                return cache.get(url);
            }

            @Override
            public void putBitmap(String
url, Bitmap bitmap) {
                cache.put(url, bitmap);
            }
        });

        public static synchronized
MySingletonintensitas getInstance(Context context)
{
            if (mInstance == null) {
                mInstance = new
MySingletonintensitas(context);
            }
            return mInstance;
        }

        public RequestQueue getRequestQueue() {
            if (mRequestQueue == null) {

                mRequestQueue =
Volley.newRequestQueue(mContext(getApplicationContext()
));
            }
            return mRequestQueue;
        }
        public <T> void addToRequestQueue(Request<T>
req) {
            getRequestQueue().add(req);
        }
        public ImageLoader getImageLoader() {
            return mImageLoader;
        }
}

```

## 7. Menu Karbondioksida

```

activity_menu_karbon.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
xmlns:tools="http://schemas.android.com/tools"

```

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/globalbg"
tools:context=".menu.MenuSuhu">

```

```

<Button
android:id="@+id/buttonk"
android:layout_width="171dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:background="@drawable/globalbg"
android:text="Karbondioksida"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.032"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.145" />

```

```

<ImageView
android:id="@+id/imageView7"
android:layout_width="317dp"
android:layout_height="317dp"
android:layout_marginEnd="32dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.691"
app:layout_constraintStart_toEndOf="@+id/button"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.84"
app:srcCompat="@drawable/codua" />
tools:ignore="MissingConstraints" />

```

```

<Button
android:id="@+id/buttonk2"
android:layout_width="170dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:background="@drawable/globalbg"
android:text="grafik Karbondioksida"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.691"
app:layout_constraintStart_toEndOf="@+id/button"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.145" />
</android.support.constraint.ConstraintLayout>

```

```

MenuKarbon.java
package com.example.pita.monitoringsystem.menu;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import com.example.pita.monitoringsystem.R;
import com.example.pita.monitoringsystem.activity.Karbon_Activity;
import com.example.pita.monitoringsystem.grafik.GrafikKarbon;
public class MenuKarbon extends AppCompatActivity
implements View.OnClickListener {
    Button karbon, grafik;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu_karbon);
    karbon = findViewById(R.id.buttonk);
    grafik = findViewById(R.id.buttonk2);
    karbon.setOnClickListener(this);
    grafik.setOnClickListener(this);
}

```

```

@Override
public void onClick(View v) {if(v == karbon){
startActivity(new Intent(getApplicationContext(),
Karbon_Activity.class)); }
else if(v == grafik){
startActivity(new Intent(getApplicationContext(),
GrafikKarbon.class);}}
}

```

```

activity_karbon.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

```

```

tools:context=".activity.Suhu_Activity">

```

```

<ImageView
android:id="@+id/imageView"
android:layout_width="391dp"
android:layout_height="322dp"
app:srcCompat="@drawable/codua" />

```

```

<android.support.v7.widget.RecyclerView
android:id="@+id/list_karbon"
android:layout_width="368dp"
android:layout_height="183dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"/>
</android.support.constraint.ConstraintLayout>

```

```

Karbon_Activity.java
package

```

```

com.example.pita.monitoringsystem.activity;
import android.app.ProgressDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.example.pita.monitoringsystem.R;
import com.example.pita.monitoringsystem.adapter.AdapterKarbon;
import com.example.pita.monitoringsystem.adapter.AdapterKelembaban;
import com.example.pita.monitoringsystem.model.Karbon;
import com.example.pita.monitoringsystem.model.Kelembaban;
import com.example.pita.monitoringsystem.util.AppController;
import com.example.pita.monitoringsystem.util.ServerAPI;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
public class Karbon_Activity extends
AppCompatActivity {
    RecyclerView mRecyclerview;
    RecyclerView.Adapter mAdapter;
    RecyclerView.LayoutManager mManager;
    List<Karbon> mItems;
    ProgressDialog pd;
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_karbon);
    mRecyclerview = (RecyclerView)
findViewById(R.id.list_karbon);
    pd = new
ProgressDialog(Karbon_Activity.this);
    mItems = new ArrayList<Karbon>();
    loadJson();
    mManager = new
LinearLayoutManager(Karbon_Activity.this,
LinearLayoutManager.VERTICAL, false);
    mRecyclerview.setLayoutManager(mManager);
    mAdapter = new
AdapterKarbon(Karbon_Activity.this, mItems);
    mRecyclerview.setAdapter(mAdapter);
}
private void loadJson() {
    pd.setMessage("Mengambil Data");
    pd.setCancelable(false);
    pd.show();
    JsonArrayRequest reqData = new
JsonArrayRequest(Request.Method.POST,
ServerAPI.URL_DATAKARBON, null,
new Response.Listener<JSONArray>() {
@Override
public void onResponse(JSONArray response)

```

```

(pd.cancel();
Log.d("volley", "response : " +
response.toString());
for(int i = 0 ; i < response.length(); i++){try {
JSONObject data = response.getJSONObject(i);
Karbon md = new Karbon();
md.setNilai(data.getString("nilai"));
md.setWaktu(data.getString("waktu"));
mItems.add(md);} catch (JSONException e)
{e.printStackTrace();}}
mAdapter.notifyDataSetChanged();},
new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
pd.cancel();Log.d("volley", "error : " +
error.getMessage());}});
AppController.getInstance().addToRequestQueue(req
ata);}}

```

#### parameter\_karbon.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
android:layout_width="match_parent"
android:layout_height="match_parent">

```

```

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

```

```

<TextView
android:id="@+id/txtkarbon"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:gravity="center"
android:text="@string/currentDummyKarbon"
android:textAppearance="?android:attr/textAppearan
ceLarge"
android:textColor="@android:color/background_dark"
android:textSize="40sp"
android:textStyle="bold"
android:typeface="normal" />

```

```

<TextView
android:id="@+id/txtTitleLastUpdate"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:gravity="center"
android:text="@string/currentLastUpdate"
android:textAppearance="?android:attr/textAppearan
ceMedium"
android:textSize="15sp" />

```

```

<TextView
android:id="@+id/txtWaktu"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:text="@string/currentDummyDate"
android:textAppearance="?android:attr/textAppearan
ceMedium"
android:textSize="16sp" />
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

#### activity\_grafik\_karbon.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-
auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".grafik.GrafikIntensitas">

```

```

<com.github.mikephil.charting.charts.LineChart
android:id="@+id/chartk"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

```

#### Karbon.java

```

package com.example.pita.monitoringsystem.model;
import android.support.v7.app.AppCompatActivity;
public class Karbon extends AppCompatActivity {
String nilai,waktu;
public Karbon(){}
public Karbon(String nilai, String waktu) {
this.nilai = nilai;
this.waktu = waktu; }

public String getNilai() {return nilai;}
public void setNilai(String nilai) {
this.nilai = nilai;}
public String getWaktu() {
return waktu; }
public void setWaktu(String waktu) {
this.waktu = waktu;}

```

#### AdapterKarbon.java

```

package com.example.pita.monitoringsystem.adapter;
import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.model.Karbon;
import java.util.List;
public class AdapterKarbon extends
RecyclerView.Adapter<AdapterKarbon.ViewHolderData> {
private List<Karbon> mItems ;
private Context context;
public AdapterKarbon(Context context,
List<Karbon> items){
this.mItems = items;
this.context = context;}

```

```

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
View layout =
LayoutInflater.from(parent.getContext()).inflate(R
.layout.parameter_karbon,parent,false);
ViewHolder holderData = new
ViewHolder(layout);return holderData; }
@Override
public void onBindViewHolder(@NonNull
ViewHolder holder, int position) {
Karbon md = mItems.get(position);
holder.t_Karbon.setText(md.getNilai());

holder.t_Waktu.setText(md.getWaktu());holder.md =
md; }

```

```

@Override
public int getItemCount() {
return mItems.size();}
class ViewHolderData extends RecyclerView.ViewHolder{
TextView t_Karbon,t_Waktu;
Karbon md;
public ViewHolderData (View view){
super (view);
t_Karbon = (TextView)
view.findViewById(R.id.txtkarbon);
t_Waktu = (TextView)
view.findViewById(R.id.txtWaktu)

```

```

/>
</android.support.constraint.ConstraintLayout>

```

#### GrafikKarbon.java

```

package com.example.pita.monitoringsystem.grafik;
import android.app.ProgressDialog;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.pita.monitoringsystem.R;
import
com.example.pita.monitoringsystem.util.ServerAPI;
import
com.github.mikephil.charting.charts.BarChart;
import

```

```

com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.BarData;
import
com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import
com.github.mikephil.charting.data.LineDataSet;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;

```

```

public class GrafikKarbon extends
AppCompatActivity {
    private ProgressDialog pd;
    ArrayList<LineDataSet> yAxis;
    ArrayList<Entry> yValues;
    ArrayList<String> xAxis1;
    Entry values ;
    LineChart chart;
    LineData data;@Overrideprotected void
onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_grafik_karbon);
    pd = new
ProgressDialog(GrafikKarbon.this);
    pd.setMessage("loading");
    chart = (LineChart)
findViewById(R.id.chartk);
    load data from server();
    public void load_data_from_server(){
    pd.show();
    String url = ServerAPI.URL_GRAFIK_KARBON;
    xAxis1 = new ArrayList<>();
    yAxis = null;
    yValues = new ArrayList<>();
    StringRequest stringRequest = new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
    Log.d("String",response);try{
    JSONArray jsonArray = new JSONArray(response);
    for (int i=0; i<jsonArray.length(); i++){
    JSONObject jsonObject =
jsonArray.getJSONObject(i);
    String karbon =
jsonObject.getString("nilai").trim();
    String waktu =
jsonObject.getString("waktu").trim();
    xAxis1.add(waktu); values = new
Entry(Float.valueOf(karbon),i);
    yValues.add(values);}
    }catch (JSONException e){
    e.printStackTrace();}
    LineDataSet lineDataSet = new
LineDataSet(yValues,"Grafik Karbondioksida (18
data terakhir)");
    lineDataSet.setColor(Color.rgb(0,82,159));
    yAxis = new ArrayList<>();
    yAxis.add(lineDataSet);
    String names[] = xAxis1.toArray(new
String[xAxis1.size()]);
    data = new LineData(xAxis1,lineDataSet);
    chart.setData(data);
    chart.setDescription("");
    chart.animateXY(2000,2000);
    chart.invalidate();
    pd.hide();}},

```

## 8. Menu Diskusi Aeroponik

activity\_mainquestion.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/globalbg"
android:paddingEnd="3dp"
android:paddingRight="3dp"
android:scrollbarThumbVertical="@drawable/vertical_s
crollview_bg"
android:scrollbarTrackVertical="@drawable/vertical_s
crollview_line"
tools:context=".menu.MainQuestion">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
android:padding="16dp">

```

```

new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
if(error != null){
Toast.makeText(getApplicationContext(), "Ups,
terjadi kesalahan.", Toast.LENGTH_LONG).show();
pd.hide();}}};
MySingletonkarbon.getInstance(getApplicationContext()
t()).addToRequestQueue(stringRequest);}

```

MySingletonkarbon.java

```

package com.example.pita.monitoringsystem.grafik;
import android.content.Context;
import android.graphics.Bitmap;
import android.support.v4.util.LruCache;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;

```

public class MySingletonkarbon {

```

    private static MySingletonkarbon mInstance;
    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;
    private static Context mContext;
    private MySingletonkarbon(Context context) {
    mContext = context;
    mRequestQueue = getRequestQueue();
    mImageLoader = new
ImageLoader(mRequestQueue,
new ImageLoader.ImageCache() {
    private final LruCache<String,
Bitmap> cache = new
LruCache<String, Bitmap>(20);
    @Override
    public Bitmap getBitmap(String
url) {
    return cache.get(url);
    }
    @Override
    public void putBitmap(String
url, Bitmap bitmap) {
    cache.put(url, bitmap);
    }
    });
    public static synchronized MySingletonkarbon
getInstance(Context context) {
    if (mInstance == null) {
    mInstance = new
MySingletonkarbon(context);
    }return mInstance;
    }
    public RequestQueue getRequestQueue() {
    if (mRequestQueue == null){mRequestQueue =
Volley.newRequestQueue(mContext(getApplicationContext()
));
    }return mRequestQueue;}
    public <T> void addToRequestQueue(Request<T> req)
{
    getRequestQueue().add(req);}
    public ImageLoader getImageLoader() {return
mImageLoader; }
}

```

```

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="1. Apakah yang dimaksud dengan
aeroponik ?"
android:textColor="@color/colorBlack"
android:textStyle="bold" />

```

```

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Aeroponik merupakan suatu cara
bercocok tanam sayuran diudara tanpa penggunaan
tanah, nutrisi disemprotkan pada akar tanaman, air
yang berisi larutan hara atau nutrisi disemburkan
dalam bentuk kabut hingga mengenai akar tanaman.
Akar tanaman yang ditanam menggantung akan menyerap
larutan hara tersebut. Air dan nutrisi disemprotkan
menggunakan irigasi sprinkler.\n\n"
android:textColor="@color/colorTabStrip"

```

```

        android:textStyle="bold" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="2. Bagaimana prinsip kerja aeroponik ?"
    android:textColor="@color/colorBlack"
    android:textStyle="bold" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Titik utama aplikasi aeroponik adalah tekanan (pressure) yang dihasilkan oleh pompa harus tinggi dan kesesuaian desain instalasi. Tekanan tinggi pada selang saluran akan menghasilkan butiran air berbentuk kabut. Permasalahan untuk teknik aeroponik pada umumnya adalah tekanan yang dihasilkan pompa kurang tinggi sehingga terkreasi butiran air kasar bukan kabut sehingga butiran air menurun. Semakin kecil butiran air maka permukaan butiran air semakin luas.\n\n"
    android:textColor="@color/colorTabStrip"
    android:textStyle="bold" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="3. Apakah metode penanaman aeroponik lebih baik dari metode penanaman menggunakan media tanah?"
    android:textColor="@color/colorBlack"
    android:textStyle="bold" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Perbandingan pertumbuhan antara budidaya sayuran secara aeroponik dengan metode tanam di tanah adalah 2 : 1. Budidaya sayuran aeroponik terbukti lebih cepat pertumbuhannya dibandingkan dengan budidaya media tanam tanah, hal tersebut disebabkan terpenuhinya nutrisi yang dibutuhkan sayuran secara terus menerus.\n\n"
    android:textColor="@color/colorTabStrip"
    android:textStyle="bold" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="4. Nutrisi seperti apa yang diperlukan oleh sistem aeroponik?"
    android:textColor="@color/colorBlack"
    android:textStyle="bold" />

```

## 9. Connect to Webserver

```

AppController.java
package com.example.pita.monitoringsystem.util;
import android.app.Application;
import android.text.TextUtils;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;
public class AppController extends Application {
    private static final String TAG =
AppController.class.getSimpleName();
    private static AppController instance ;
    RequestQueue mRequestQueue;
    @Override
    public void onCreate() {
        super.onCreate();
        instance = this;
    }
    public static synchronized AppController
getInstance() {
        return instance;
    }
    private RequestQueue getRequestQueue()
    {
        if(mRequestQueue == null){
            mRequestQueue
=Volley.newRequestQueue(getApplicationContext());
            return mRequestQueue;
        }
        public <T> void addToRequestQueue(Request<T> req,
String tag){
        req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
        getRequestQueue().add(req);
    }
    public <T> void addToRequestQueue(Request<T> req)
    {
        req.setTag(TAG);
        getRequestQueue().add(req);
    }
}

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nutrisi aeroponik merupakan pupuk yang telah diformulasikan khusus dari garam-garam mineral yang larut dalam air, mengandung unsur-unsur hara penting yang diperlukan tanaman bagi tumbuh dan berkembang. Nutrisi ini terdiri dari 2 bagian yaitu bagian A dan bagian B, biasanya disebut AB mix. Cara penggunaannya sangat mudah, hanya dengan mencampurkan bagian A dan bagian B dengan air, satu persatu secara terpisah, sesuai petunjuk yang diberikan produsen nutrisi tersebut untuk menjadikan larutan stok atau pekatan. Larutan stok ini perlu dicairkan lagi dengan air jika hendak digunakan.\n\n"
    android:textColor="@color/colorTabStrip"
    android:textStyle="bold" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="5. Jenis tanaman apa saja yang sering dibudidayakan secara aeroponik?"
    android:textColor="@color/colorBlack"
    android:textStyle="bold" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Jenis tanaman yang sering dibudidayakan secara aeroponik pada umumnya berupa sayuran daun yang waktu panennya sekitar satu bulan setelah pindah tanam. Harga jual komoditas tersebut juga dipilih yang dapat memberikan keuntungan maksimal. Tanaman rempah penyedap masakan, seperti oreano, parsley, thyme, dill dan basil dapat diusahakan dalam volume kecil.\n\n"
    android:textColor="@color/colorTabStrip"
    android:textStyle="bold" />

```

```

</LinearLayout>
</ScrollView>

```

```

MainQuestion.java
package com.example.pita.monitoringsystem.menu;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import com.example.pita.monitoringsystem.R;
public class MainQuestion extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mainquestion);
    }
}

```

```

public void cancelAllRequest(Object req) {
    if (mRequestQueue != null) {
        mRequestQueue.cancelAll(req);
    }
}

```

```

ServerAPI.java
package com.example.pita.monitoringsystem.util;
public class ServerAPI {
    public static final String URL_DATASUHU =
"http://verapuspita.000webhostapp.com/monitoring/cur
rent_suhu.php";
    public static final String URL_GRAFIK_SUHU =
"http://verapuspita.000webhostapp.com/monitoring/get
_data suhu.php";
    public static final String URL_DATAKELEMBABAN =
"http://verapuspita.000webhostapp.com/monitoring/cur
rent_kelembaban.php";
    public static final String URL_GRAFIK_KELEMBABAN =
"http://verapuspita.000webhostapp.com/monitoring/get
_datakelembaban.php";
    public static final String URL_DATAINTENSITAS =
"http://verapuspita.000webhostapp.com/monitoring/cur
rent_intensitas.php";
    public static final String URL_GRAFIK_INTENSITAS =
"http://verapuspita.000webhostapp.com/monitoring/get
_dataintensitas.php";
    public static final String URL_DATAKARBON =
"http://verapuspita.000webhostapp.com/monitoring/cur
rent_karbon.php";
    public static final String URL_GRAFIK_KARBON =
"http://verapuspita.000webhostapp.com/monitoring/get
_datakarbon.php";
}

```