



# **RANCANG BANGUN SISTEM PERAWATAN TUMBUHAN KACANG-KACANGAN SECARA ADAPTIF PADA LINGKUNGAN *GREEN HOUSE* DENGAN MIKROKONTROLER ARDUINO**

**GREZIO ARIFIYAN PRIMAJAYA  
NRP 5110100196**

**Dosen Pembimbing I  
TOHARI AHMAD, S.Kom., MIT., Ph.D.**

**Dosen Pembimbing II  
Henning Titi Ciptaningtyas, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA**

*(Halaman ini sengaja dikosongkan)*



**UNDERGRADUATE THESES - KI091391**

**DESIGN AND IMPLEMENTATION OF ADAPTIVE  
NUTS PLANT MAINTENANCE SISTEM ON GREEN  
HOUSE ENVIRONMENT USING ARDUINO  
MICROCONTROLLER**

**GREZIO ARIFIYAN PRIMAJAYA  
NRP 5110100196**

**Advisor I  
Tohari Ahmad, S.Kom., MIT., Ph.D.**

**Advisor II  
Henning Titi Ciptaningtyas, S.Kom., M.Kom.**

**Informatics Departement  
Faculty of Information Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2014**

*(Halaman ini sengaja dikosongkan)*

# RANCANG BANGUN SISTEM PERAWATAN TUMBUHAN KACANG-KACANGAN SECARA ADAPTIF PADA LINGKUNGAN *GREEN HOUSE* DENGAN MIKROKONTROLER ARDUINO

Nama : Grezio Arifiyan Primajaya  
NRP : 51110100196  
Jurusa : Teknik Informatika – FTIf ITS  
Dosen Pembimbing I : Tohari Ahmad, S.Kom., MIT.,  
Ph.D.  
Dosen Pembimbing II : Henning Titi Ciptaningtyas, S.Kom.,  
M.Kom

## **Abstrak**

Kacang-kacangan memiliki peranan pokok sebagai pemenuh kebutuhan pangan. Permintaan terhadap kacang-kacangan tersebut setiap tahun mengalami peningkatan sejalan dengan pertambahan jumlah penduduk dan berkembangnya industri pangan. Disamping itu kacang-kacangan merupakan sumber protein nabati yang sangat dibutuhkan untuk meningkatkan kualitas sumberdaya manusia. Untuk itu maka pengembangan kacang-kacangan harus terus diupayakan dan ditingkatkan.

Perlu bantuan teknologi yang tepat guna dapat meningkatkan kualitas dari hasil panen kacang. Dikembangkan sebuah metode penanaman tanpa mengenal musim atau yang sering disebut *greenhouse*. Di dalam ruangan diharapkan dapat dibuat manipulasi cuaca sesuai kebutuhan tanaman. Penggunaan *green house* selama ini masih minim terutama untuk manipulasi suhu pada ruangan. Pengembangan *green house* dapat menguntungkan terutama bagi petani yang dihadapkan oleh kendala iklim yang kurang menentu akhir-akhir ini. Dengan adanya data pendukung syarat tumbuh yang lengkap maka diharapkan bisa menghasilkan kualitas kacang hijau yang terbaik. Untuk mengontrol kondisi ruangan *green house* diperlukan

bantuan teknologi mikrokontroler yang dapat mengukur dan mengatur variabel yang ada dalam syarat tumbuh tanaman.

Pada penelitian ini dibuat sebuah sistem perawatan menggunakan mikrokontroler Arduino Uno. Sistem perawatan mampu membuat kondisi lingkungan tanam menjadi ideal untuk tanaman kacang hijau.

***Kata Kunci***—Kacang hijau; *green house*; syarat tumbuh; mikrokontroler.

# DESIGN AND IMPLEMENTATION OF ADAPTIVE NUTS PLANT MAINTENANCE SISTEM ON GREENHOUSE ENVIRONMENT USING ARDUINO MICROCONTROLLER

Nama : Grezio Arifiyan Primajaya  
NRP : 51110100196  
Jurusan : Teknik Informatika – FTIf ITS  
Dosen Pembimbing I : Tohari Ahmad, S.Kom., MIT.,  
Ph.D.  
Dosen Pembimbing II : Henning Titi Ciptaningtyas, S.Kom.,  
M.Kom

## **Abstract**

Nuts have a main role as fulfilling the needs for food, demand has increased each year in line with population growth and development of the food industry. As are a source of vegetable protein, nuts are needed to improve the quality of human resources. For those reasons, the development of the nuts should be pursued and improved.

An appropriate technology is needed to increase the quality of the bean crop. A method of planting regardless of the season is called a green house. In this plantation room weather can be manipulated according to the needs of plants. The use of green house so far is minimal, especially for temperature manipulation in the room. Development of green house can be beneficial, especially for farmers who are confronted by a less erratic climatic constraints lately. Acurate data is required to have hight quality green bean. In order to control the green house room condition, we employ a microcontroller to measure and adjust the variables in the plantation requirements.

This study develops a maintenance system using an Arduino Uno microcontroller. This system is able to maintain the condition of the green bean plants to be ideal.

***Keyword*—Green beans; greenhouse; growing requirements; microcontroller.**



## LEMBAR PENGESAHAN

### RANCANG BANGUN SISTEM PERAWATAN TUMBUHAN KACANG-KACANGAN SECARA ADAPTIF PADA LINGKUNGAN *GREEN HOUSE* DENGAN MIKROKONTROLER ARDUINO

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelar Sarjana Komputer

pada

Bidang Studi Komputasi Berbasis Jaringan  
Jurusan Teknik Informatika  
Fakultas Teknologi Informasi

Oleh:

**GREZIO ARIFIYAN PRIMAJAYA**  
NRP. 5110100196

Disetujui oleh Pembimbing tugas Akhir:

TOHARI AHMAD, S.Kom., MIT., Ph.D.  
NIP. 197505252003121002

HENNING TITI CIPTANINGTYAS,  
S.Kom., M.Kom.  
NIP. 198407082010122004



**SURABAYA**  
**JULI, 2014**

*(Halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Bismillahirrohmanirohim.

Alhamdulillahilahirabil'alamin, segala puji bagi Allah Subhanahu Wata'alla, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tugas Akhir yang berjudul "***Rancang Bangun Sistem Perawatan Tumbuhan Kacang-kacangan secara Adaptif pada Lingkungan Greenhouse dengan Mikrokontroler Arduino***" dengan baik dan tepat waktu.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan-bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Kedua orang tua penulis, yang telah memberikan dukungan moral, spiritual dan material, semangat, perhatian, selalu setia dan sabar dalam menghadapi curhatan dari penulis, serta selalu memberikan doa yang tiada habisnya yang dipanjatkan untuk penulis.
3. Bapak Dr. Tohari Ahmad, S.Kom., MIT., selaku dosen pembimbing pertama, yang telah memberikan kepercayaan, dukungan, bimbingan, nasehat, perhatian, serta semua yang telah diberikan kepada penulis.
4. Ibu Henning Titi Ciptaningtyas, S.Kom., M.Kom., selaku dosen pembimbing kedua, atas bimbingan, arahan, bantuan serta ide untuk menyelesaikan Tugas Akhir ini.
5. Ibu Nanik Suciati, S.Kom., M.Kom., Dr.Eng.selaku ketua jurusan Teknik Informatika ITS, Prof Supeno Djanali. selaku dosen wali penulis, Bapak Abdul Munif, S.Kom, M.Sc selaku koordinator TA, Ibu Isye Ariesianti, S.Kom,

M.Phil. selaku koordinator KP, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya kepada penulis.

6. Pak Yudi, Pak Sugeng, Mas Jumali dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
7. Teman-teman admin lab NCC 2010 Noor Maulida ,Fajri, Tasha, Adik-adik 2011 Ade, Faris, Punggi, Risal, Billa, Deasy, Nisa.
8. Seluruh teman Teknik Informatika ITS angkatan 2010, empat tahun bersama kalian sadar atau tidak telah membentuk karakter dan kepribadian penulis.
9. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu persatu yang telah membantu terselesaikannya Tugas Akhir ini.

Kesempurnaan tentu sangat jauh tercapai pada Tugas Akhir ini, maka penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, Juni 2014

Grezio Arifiyan

## DAFTAR ISI

Abstrak.....	ix
Abstract.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan dan Manfaat.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 MySQL.....	5
2.2 PhoneGap.....	5
2.3 Mikrokontroler Arduino Uno.....	5
2.4 Sensor Suhu (LM35.....	7
2.5 Sensor Intensitas Cahaya (LDR).....	8
2.6 Sensor Kelembaban Tanah (SEN0057).....	9
2.7 Saklar Elektronik (Solid State Relay/SSR).....	9
2.8 Kran Otomatis (Katup Solenoida).....	10
2.9 Modul Koneksi Data (GSM/GPS/GPRS <i>Shield</i> ).....	10
2.10 Protokol Zigbee.....	11
2.11 Modul Xbee.....	13
2.12 <i>Fuzzy Logic</i> .....	14
2.13 <i>Xbee Shield</i> .....	14
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	17
3.1 Deskripsi Umum.....	17
3.2 Arsitektur Sistem.....	18
3.3 Perancangan Kebutuhan Arus Listrik.....	19
3.4 Perancangan Perangkat Keras.....	19
3.5 Perancangan Alir Data.....	23
3.6 Diagram Alir Aplikasi Sistem.....	24
3.7 Perancangan Basis Data Sistem.....	31

3.8	Perancangan Antarmuka Pengguna Telepon Genggam Pintar Android.....	32
3.9	Perancangan Antarmuka Pengguna Web .....	34
BAB IV IMPLEMENTASI .....		37
4.1	Lingkungan Implementasi.....	37
4.2	Implementasi Perangkat Keras data pembanding .....	38
4.3	Implementasi perangkat lunak pada perangkat keras data pembanding.....	39
4.4	Implementasi Perangkat keras Sistem Perawatan Kacang-kacangan.....	40
4.5	Implementasi Perangkat lunak pada perangkat keras sistem perawatan kacang-kacangan. ....	43
4.6	Implementasi Pada <i>Server</i> .....	50
4.7	Implementasi Antarmuka Pengguna Android.....	51
4.8	Implementasi Antar Pengguna Web.....	52
BAB V UJI COBA DAN EVALUASI .....		57
5.1	Uji coba perangkat keras data pembanding.....	57
5.2	Uji Coba Perangkat Keras Sistem Perawatan .....	60
BAB VI PENUTUP .....		75
6.1	Kesimpulan .....	75
6.2	Saran.....	76
DAFTAR PUSTAKA .....		77
BIODATA PENULIS .....		79
LAMPIRAN.....		81

## DAFTAR TABEL

Tabel 1. 1. Arduino Uno data sheet .....	6
Tabel 1. 2. Data <i>sheet</i> LM35 linear.....	8
Tabel 1. 3. Data <i>sheet</i> sensor kelembaban tanah.....	9
Tabel 1. 4. Data <i>sheet</i> GSM/GPS/GPRS-Shield .....	11
Tabel 1. 5. Perbandingan perangkat <i>wireless</i> .....	12
Tabel 1. 6. Perbandingan perangkat Xbee .....	13
Tabel 3. 1. Perancangan kebutuhan daya .....	19
Tabel 3. 2. Tabel simbol sensor .....	22
Tabel 3. 3. Entitas <i>smart_garden</i> .....	32
Tabel 4. 1. Implementasi.....	54
Tabel 5. 1. Prosedur uji coba penyimpanan data.....	58
Tabel 5. 2. Hasil uji coba penyimpanan data .....	58
Tabel 5. 3. Prosedur uji coba daya tahan baterai.....	59
Tabel 5. 4. Prosedur uji coba lampu 1 dan lumpu 2.....	61
Tabel 5. 5. Prosedur pengujian sistem penyiraman otomatis.....	62
Tabel 5. 6. Prosedur pengujian penyiraman ketika tanah kering	63
Tabel 5. 7. Prosedur uji coba kontrol kipas angin.....	63
Tabel 5. 8. Prosedur uji coba pengiriman data setiap 15 menit ..	64
Tabel 5. 9. Prosedur uji coba pengiriman data ketika ada syarat tumbuh tidak terpenuhi .....	65
Tabel 5. 10. Prosedur pengujian menampilkan data dalam bentuk grafik garis .....	65
Tabel 5. 11. Rentan kelembaban tanah .....	66
Tabel 5. 12. Hasil penyiraman pagi hari .....	67
Tabel 5. 13. Hasil penyiraman sore hari .....	67
Tabel 5. 14. Rata-rata nilai kelembaban tanah.....	67
Tabel 5. 15. Rentan nilai intensitas cahaya .....	68
Tabel 5. 16. Hasil uji coba pengaruh lampu .....	69
Tabel 5. 17. Hasil pengukuran suhu manual.....	71
Tabel 5. 18. Tabel pengukuran suhu.....	71
Tabel 5. 19. Hasil uji coba lama menampilkan data pada aplikasi Android.....	74

*(Halaman ini sengaja dikosongkan)*



## DAFTAR GAMBAR

Gambar 1. 1. Mikrokontroler Arduino Uno .....	6
Gambar 1. 2. Sensor LM35 .....	7
Gambar 1. 3. Sensor cahaya(LDR) .....	8
Gambar 1. 4. Sensor Kelembaban Tanah (SEN0057).....	9
Gambar 1. 5. Saklar Elektronik (Solid State Relay/SSR) .....	10
Gambar 1. 6. Kran Otomatis (Katup Solenoida).....	10
Gambar 1. 7. Modul Koneksi Data (GSM/GPS/GPRS-Shield)..	11
Gambar 1. 8. Xbee-Shield.....	15
Gambar 3. 1. Alur proses penggunaan sistem.....	18
Gambar 3. 2. Perangkat Keras Sistem Perawatan .....	20
Gambar 3. 3. Perangkat Keras Data Perbandingan .....	21
Gambar 3. 4. Penempatan Sensor .....	22
Gambar 3. 5. Perancangan Alir data .....	24
Gambar 3. 6. Diagram alir sistem .....	25
Gambar 3. 7. Diagram alir sensor .....	26
Gambar 3. 8. Diagram alir sensor suhu.....	27
Gambar 3. 9. Diagram alir sensor cahaya .....	28
Gambar 3. 10. Diagram alir kelembaban tanah.....	29
Gambar 3. 11. Diagram alir pengiriman data.....	30
Gambar 3. 12. Diagram alir penampilan data .....	31
Gambar 3. 13. Perancangan Basis Data Sistem .....	32
Gambar 3. 14. Perancangan Antarmuka Pengguna <i>Smartphone</i> <i>Android</i> .....	33
Gambar 3. 15. Perancangan antarmuka grafik log aktivitas .....	34
Gambar 3. 16. Perancangan antarmuka pengguna web grafik log aktivitas.....	35
Gambar 3. 17. Perancangan antarmuka tabel log aktivitas pengguna web .....	35
Gambar 4. 1. Perangkat keras data perbandingan	38
Gambar 4. 2. Perangkat data perbandingan terpasang .....	39
Gambar 4. 3. Pengambilan data perbandingan .....	40

Gambar 4. 4. Perangkat keras sistem perawatan kacang-kacangan .....	41
Gambar 4. 5. Perangkat koordinator terpasang.....	41
Gambar 4. 6. Perangkat <i>end device</i> .....	42
Gambar 4. 7. <i>End device</i> dan relay .....	43
Gambar 4. 8. m2msuport modul penguji .....	44
Gambar 4. 9. Implementasi Xbee koodinator .....	45
Gambar 4. 10. X-CTU .....	46
Gambar 4. 11. Pengujian gagal .....	47
Gambar 4. 12. Pin DOUT .....	47
Gambar 4. 13. Uji coba berhasil .....	48
Gambar 4. 14. <i>Setup firmware</i> .....	49
Gambar 4. 15. <i>Read firmware</i> .....	50
Gambar 4. 16. Tampilan utama aplikasi Android.....	51
Gambar 4. 17. Grafik log aktivitas pada Android.....	52
Gambar 4. 18. Tampilan log aktivitas halaman web.....	53
Gambar 4. 19. Tampilan tabel log aktivitas .....	54
Gambar 4. 20. Sistem perawatan .....	56
Gambar 5. 1. Grafik log aktivitas 17 juni.....	72
Gambar 5. 2. Grafik log aktivitas 21 juni .....	73

# **BAB I**

## **PENDAHULUAN**

Bab ini membahas garis besar penyusunan Tugas Akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan Tugas Akhir, dan sistematika penulisan.

### **1.1 Latar Belakang**

Komoditas kacang hijau mempunyai arti yang strategis karena menyediakan kebutuhan paling esensial bagi kehidupan, sebagai bahan pangan serta sumber protein nabati yang sangat dibutuhkan. Kebutuhan akan kacang hijau akan semakin meningkat sejalan dengan pertambahan jumlah penduduk dan berkembangnya industri pangan dan pakan. Di sisi lain produksi kacang hijau yang dihasilkan belum dapat memenuhi kebutuhan tersebut.

Untuk itu maka pengembangan kacang hijau harus terus diupayakan dan ditingkatkan guna memenuhi kebutuhan tersebut dan tentunya akan berdampak pula pada peningkatan pendapatan petani dan menggerakkan kegiatan perekonomian di wilayah pedesaan. Upaya-upaya pengembangan kacang hijau dilakukan melalui peningkatan luas area tanam, panen dan produktivitas, nilai tambah dan daya saing. Daerah-daerah sentra kacang hijau yang selama ini menurun luas tanamnya bahkan mengalihkan ke komoditas lain perlu diarahkan dan dibimbing untuk menanam kembali kacang hijau.

Kacang-kacangan memiliki peranan pokok untuk memenuhi kebutuhan pangan dan industri dalam negeri yang setiap tahun mengalami peningkatan sejalan dengan pertambahan jumlah penduduk dan berkembangnya industri pangan. Selain itu kacang-kacangan merupakan sumber protein nabati yang sangat dibutuhkan untuk meningkatkan kualitas sumber daya manusia Indonesia. Untuk itu maka

pengembangan kacang-kacangan harus terus diupayakan dan ditingkatkan.

Untuk meningkatkan kualitas dari kacang-kacangan tersebut perlu bantuan teknologi, penggunaan teknologi yang tepat guna akan membantu peningkatan kualitas dari hasil panen kacang, pada usulan tugas akhir ini difokuskan pada tanaman kacang hijau. Tanaman kacang hijau bergantung pada musim, tanaman kacang hijau akan mulai ditanam pada musim hujan dan dipanen ketika musim kering, petani memperkirakan pada saat musim panas dapat menjemur hasil panen. Hasil panen dijemur menggunakan bantuan sinar matahari, proses penjemuran ini dapat menentukan kualitas kacang hijau sendiri. Apabila penjemuran kurang maksimal dapat menimbulkan masalah ketika penimbunan kacang hijau, dikarenakan kacang masih dalam keadaan mentah dan belum diolah sehingga kacang tetap melakukan metabolisme yang menghasilkan panas. Panas yang dihasilkan tersebut menimbulkan kelembaban suhu pada area sekitar penimbunan kacang hijau yang dapat meimbulkan jamur pada kacang hijau tersebut.

Selain pengeringan hal yang dapat mempengaruhi kualitas dari hasil panen kacang hijau adalah iklim dan syarat tumbuh tanaman, syarat tumbuh tanaman kacang hijau antarlain:

Ketinggian tanah	: 5 – 700 dpl
Suhu	: 25 – 270 C
Lama paparan sinar	: >10 jam / hari
Curah hujan	: 50 – 200 / bulan
PH tanah	: 5,8 – 6,5
Kelembaban udara	: 50 – 89 %

Iklim yang tidak sesuai dapat merubah variabel pada syarat tumbuh tanaman misal suhu, kelembaban tanah dan intensitas cahaya yang dibutuhkan oleh tanaman kacang hijau. Untuk itu perlu dikembangkan sebuah metode penanaman tanpa mengenal musim atau yang sering disebut

*green house* dalam ruangan ini diharapkan dapat dibuat manipulasi cuaca sesuai kebutuhan tanaman, penggunaan *green house* selama ini masih minim. Pengembangan *green house* dapat menguntungkan terutama bagi petani yang dihadapkan oleh kendala iklim yang kurang menentu akhir-akhir ini dengan adanya data pendukung syarat tumbuh yang lengkap maka diharapkan bisa menghasilkan kualitas kacang hijau yang baik.

Pada penelitian sebelumnya sistem yang dibuat sebatas pembuatan kran otomatis, dengan mengukur suhu dan kelembaban tanah sistem yang telah dibuat dapat mengairi tanah atau tanaman secara otomatis sesuai dengan batas kelembaban tanah yang diatur [2]. Selain itu dalam penelitian yang telah dilakukan juga mengatur suhu yang ada di atap *green house* pada penelitian sebelumnya difokuskan dalam permasalahan tanaman tomat, buah tomat akan mudah retak jika berada dalam suhu yang panas.

Dalam penelitian ini penulis ingin membuat sebuah sistem yang dapat mengontrol dan mengukur kebutuhan tanaman kacang hijau selama berada di *green house*. Untuk membantu mengontrol diperlukan bantuan teknologi mikrokontroler yang dapat mengukur dan mengatur variabel yang ada dalam syarat tumbuh tanaman kacang hijau, sensor yang dibutuhkan berupa sensor suhu, kelembaban tanah, dan cahaya serta dibutuhkan algoritma *fuzzy* untuk membantu dalam mengontrol variabel yang dibutuhkan tanaman kacang hijau dalam proses pembenihan sampai masa panen tiba, yang membedakan dalam penelitian ini dengan penelitian sebelumnya adalah adanya komunikasi data yang dikirim ke *server* sebagai log aktivitas dari mikrokontroler data yang telah disimpan dapat diakses melalui perangkat bergerak dengan kerangka kerja Android.

## 1.2 Rumusan Permasalahan

Rumusan masalah pada penelitian ini terdapat beberapa masalah yang ingin diselesaikan oleh penulis, yaitu bagaimana membangun sistem yang dapat mengukur dan mengatur suhu, kelembaban, dan cahaya dan membandingkan dengan data syarat tumbuh tanaman kacang hijau pada proses produksi di ruangan *green house*?

## 1.3 Batasan Masalah

Dari masalah yang diuraikan oleh penulis, maka batasan masalah pada tugas akhir ini adalah:

1. Aplikasi *server* berupa aplikasi basis data menggunakan teknologi MySQL.
2. Data yang dikirimkan ke *server* merupakan data kondisi cahaya, suhu dan kelembaban tanah.
3. Data yang dikirimkan pada *server* setiap 15 menit sekali.
4. Data pembanding atau data syarat tumbuh tanaman didapat dari pengamatan lapangan (ladang tanaman kacang hijau).
5. Sistem diuji coba pada ruangan greenhouse Balai Penelitian Tanaman Aneka Kacang dan Umbi (BALITKABI).

## 1.4 Tujuan dan Manfaat

Tujuan dari pembuatan tugas akhir ini adalah:

1. Membuat sistem yang dapat membantu dalam proses produksi tanaman kacang hijau pada *green house*.
2. Sistem yang dibuat diharapkan dapat memanipulasi iklim pada ruangan untuk menunjang proses produksi tanaman kacang hijau.

Manfaat dari penelitian ini adalah sebagai sarana memantau dan mendukung produksi kacang hijau. Sistem yang dibuat dapat membantu produksi kacang hijau tanpa terkendala masalah iklim.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi penjelasan teori-teori yang berkaitan dengan pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 MySQL**

MySQL adalah sebuah perangkat lunak yang bersifat *open source*. MySQL merupakan sebuah relasi manajemen basis data yang *open source*, *enterprise level*, dan *multithread*. Pada penelitian ini, teknologi MySQL digunakan sebagai basis data untuk menyimpan data-data tentang suhu, kelembaban, dan cahaya yang dikirimkan lewat GPRS dan log aktivitas perintah yang dijalankan oleh sensor suhu dan cahaya. Selain itu, MySQL dapat diunduh secara gratis tanpa harus membayar.

#### **2.2 PhoneGap**

PhoneGap adalah sebuah kerangka kerja/*framework open source* yang dipakai untuk membuat aplikasi cross-platform mobile dengan HTML, CSS, dan JavaScript. PhoneGap menjadi suatu solusi yang ideal untuk seorang web developer yang tertarik dalam pembuatan aplikasi pada telepon genggam pintar. PhoneGap juga merupakan solusi ideal bagi mereka yang tertarik untuk membuat sebuah aplikasi yang dapat berjalan pada beberapa perangkat telepon genggam pintar dengan basis kode yang sama. Artinya, cukup hanya dengan 1 kali koding saja, anda bisa membuat aplikasi untuk smartphone iPhone, Android, Blackberry, Symbian dan Palm.

#### **2.3 Mikrokontroler Arduino Uno**

Arduino Uno merupakan sebuah kerangka kerja dari komputasi fisik yang bersifat *open source*. Arduino Uno tidak

hanya sekedar sebuah alat yang berjalan sendiri, tetapi Arduino Uno dapat dikombinasikan dengan perangkat sensor lain.



**Gambar 1. 1. Mikrokontroler Arduino Uno**

Diperlukan IDE (*Integrated Development Environment*), IDE adalah sebuah perangkat lunak yang berperan untuk menulis program, mengompilasi menjadi kode biner dan mengunggah ke dalam memori mikrokontroler. Ada banyak proyek dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino Uno. Selain itu juga, ada banyak modul-modul pendukung (sensor, tampilan, penggerak, dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino Uno. Arduino Uno berevolusi menjadi sebuah kerangka kerja karena menjadi pilihan dan acuan bagi banyak praktisi. Pada penelitian ini digunakan Arduino Uno seperti pada Gambar 1. 1. penggunaan Arduino Uno cukup mudah dibandingkan dengan Arduino versi yang lain, *data sheet* Arduino Uno dapat di lihat pada Tabel 1. 1.

**Tabel 1. 1. Arduino Uno data sheet**

<b>Mikrokontoler</b>	ATmega328
<b><i>Operating Voltage</i></b>	5V
<b><i>Input Voltage (recommended)</i></b>	7-12V
<b><i>Input Voltage (limits)</i></b>	6-20V
<b>Digital I/O Pins</b>	14 (of which 6 provide PWM output)



<b>Analog Input Pins</b>	6
<b>DC Current per I/O Pin</b>	40 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328), 0.5 KB digunakan sebagai boot loader
<b>SRAM</b>	2 KB (ATmega328)
<b>EEPROM</b>	1 KB (ATmega328)
<b>Clock Speed</b>	16 MHz

## 2.4 Sensor Suhu (LM35)

Gambar 1. 2 merupakan gambar sensor suhu, sensor LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika yang diproduksi oleh National Semiconductor.



**Gambar 1. 2. Sensor LM35**

LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali tanpa memerlukan rangkaian tambahan, data *sheet* dari sensor LM35 dapat dilihat pada tabel Tabel 1. 2.

Tabel 1. 2. Data sheet LM35 linear

<b>Nilai Output</b>	°C (Celsius)
<b>Akurasi</b>	+/- 0,5 °C saat suhu +25 °C
<b>Range nilai</b>	- 55 C – 150 °C
<b>Vcc input</b>	4 – 30 V
<b>Impedance</b>	0,1 ohm/1 mA

### 2.5 Sensor Intensitas Cahaya (LDR)

Sensor Cahaya LDR (*Light Dependent Resistor*) adalah salah satu jenis resistor yang dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Besarnya nilai hambatan pada Sensor Cahaya LDR (*Light Dependent Resistor*) tergantung pada besar kecilnya cahaya yang diterima oleh LDR itu sendiri. LDR sering disebut dengan alat atau sensor yang berupa resistor yang peka terhadap cahaya.



Gambar 1. 3. Sensor cahaya(LDR)

Biasanya LDR terbuat dari *cadmium* sulfida yaitu merupakan bahan semikonduktor yang resistansinya berubah-ubah menurut banyaknya cahaya (sinar) yang mengenainya. Penggunaan LDR pada penelitian ini digunakan untuk mengukur lama penyinaran yang didapat tanaman apabila tanaman kurang mendapat sinar yang cukup maka secara otomatis lampu akan dinyalakan, sensor LDR seperti Gambar 1. 3 dapat mudah digunakan atau dihubungkan dengan perangkat Arduino Uno.

## 2.6 Sensor Kelembaban Tanah (SEN0057)

Gambar 1. 4 merupakan sebuah sensor kelembaban tanah (SEN0057), SEN0057 digunakan untuk mengukur kelembaban tanah, cara kerja dari sensor ini adalah mengukur resistansi listrik yang dimiliki tanah saat dialiri listrik oleh sensor, sensor tegangan yang cukup besar, nilai dapat dilihat pada output voltage *signal* pada tabel Tabel 1. 3.



Gambar 1. 4. Sensor Kelembaban Tanah (SEN0057)

Tabel 1. 3. Data *sheet* sensor kelembaban tanah

<b>Power supply</b>	3.3 V or 5 V
<b>Output voltage signal</b>	0~4.2 V
<b>Current</b>	35 mA

## 2.7 Saklar Elektronik (Solid State Relay/SSR)

Relay seperti Gambar 1. 5 adalah sebuah saklar elektronik yang dapat dikendalikan dari rangkaian elektronik lainnya. Relay terdiri dari saklar, kumparan dan pegas. Relay terdapat 2 jenis di pasaran yakni *Normally Closed* dan *Normally Open*. *Normally Closed* berarti saklar relay akan terhubung dengan kontak ketika keadaan normal (tidak diberi tegangan). Sedangkan *Normally Open* berarti saklar relay tidak akan terhubung dengan kontak ketika keadaan normal. Dalam penelitian ini relay akan berfungsi sebagai aktuatur dari sistem.



**Gambar 1. 5. Saklar Elektronik (Solid State Relay/SSR)**

## **2.8 Kran Otomatis (Katup Solenoida)**

Solenoida Gambar 1. 6 yaitu katup listrik yang mempunyai koil sebagai penggeraknya dimana ketika koil mendapat tegangan maka koil tersebut akan berubah menjadi medan magnet sehingga menggerakkan katup pada bagian dalamnya ketika katup berpindah posisi maka pada lubang keluaran dari kran solenoida akan keluar cairan yang berasal dari persediaan, pada umumnya pipa solenoida mempunyai tegangan kerja 100/200 VAC namun ada juga yang mempunyai tegangan kerja DC.



**Gambar 1. 6. Kran Otomatis (Katup Solenoida)**

## **2.9 Modul Koneksi Data (GSM/GPS/GPRS Shield)**

*Shield* adalah istilah untuk modul tambahan pada Arduino. Karena cara menggunakan modul tambahan pada Arduino adalah dengan cara menumpuk di bagian atas Arduino, maka diberi istilah *shield* (tameng/perisai). Sedangkan GSM/GPS/GPRS-Shield adalah sebuah modul yang dapat memungkinkan Arduino

terhubung dengan internet, menerima dan mengirim sms. GSM Shield memiliki fitur GPRS (*General Packet Radio Service*), dengan fitur GPRS modul Arduino dapat mengirim data ke *server* untuk disimpan, bentuk GSM/GPS/GPRS shield tampak seperti pada Gambar 1. 7.



**Gambar 1. 7. Modul koneksi data (GSM/GPS/GPRS shield)**

**Tabel 1. 4. Data sheet GSM/GPS/GPRS shield**

<b>Power supply</b>	6 – 12 V
<b>Power consumption</b>	100 mA@7 V - GSM mode
<b>Quad-Band</b>	850/900/1800/1900 MHz
<b>GPRS slot</b>	multi-slot class 10
<b>GPS Power consumption (GSM engine in idle mode)</b>	- <i>Acquisition</i> 77 mA - <i>Tracking</i> 76 mA

## 2.10 Protokol Zigbee

Merupakan sebuah standar komunikasi wireless yang dikembangkan dari standar IEEE 802.15.4. Zigbee dikembangkan

oleh aliansi perusahaan Zigbee Alliance. Lebih dari 300 perusahaan tergabung dalam pengembangan dari standar komunikasi Zigbee. Beberapa keunggulan dari Zigbee adalah harga yang ditawarkan relative murah dan konsumsi energy yang rendah, beberapa keunggulan tersebut yang membuat Zigbee Alliance terus mengembangkan standar komunikasi perangkat ini. Tetapi Zigbee memiliki kelemahan dalam transfer data, data yang di transfer oleh perangkat Zigbee maksimal 250 Kbps, dengan kelemahan tersebut perangkat Zigbee jarang sekali digunakan untuk mentransfer data yang cukup besar. Berikut Tabel 1. 5 merupakan perbandingan perangkat Zigbee dengan standart komunikasi *wireless* lainnya

**Tabel 1. 5. Perbandingan perangkat *wireless***

<b>Standar</b>	<b>Bandwidth</b>	<b>Konsumsi daya</b>	<b>Kelebihan</b>	<b>Aplikasi</b>
Wi-fi	>54 Mbps	400 mA	Kecepatan transfer data	Internet
Bluetooth	1 Mbps	40 mA	Mampu beroperasi antar device	Komunikasi handset
Zigbee	250 Kbps	30 mA	Hemat energy dan murah	Remote kontrol, wireless sensor

Dalam komunikasi Zigbee terdapat 3 tipe perangkat Zigbee yaitu:

1. Zigbee Koordinator

Zigbee Koordinator adalah sebuah pusat dari jaringan yang menggunakan standart komunikasi Zigbee. Selain sebagai pusat jaringan sensor nirkabel. Zigbee koordinator juga berfungsi sebagai penghubung antara jaringan luar dengan jaringannya. Dalam sebuah jaringan Zigbee minimal ada 1(satu) buah Zigbee koordinator.

2. Zigbee Router  
Zigbee router selain dapat menjalankan tugas aplikasi, perangkat Zigbee router dapat sebagai perantara antara zigbee koordinator dengan Zigbee end Device.
3. Zigbee End Device  
Memiliki fungsi untuk menyampaikan data kepada Zigbee router atau Zigbee koordinator. Perangkat ini tidak bisa meneruskan data seperti Zigbee Router.

## 2.11 Modul Xbee

**Tabel 1. 6. Perbandingan perangkat Xbee**

	<b>Xbee Seri 1</b>	<b>Xbee Seri 1</b>
<b>Tegangan yang dikeluarkan</b>	1 mW (0dbm)	2 mW (3dbm)
<b>Kecepatan Komunikasi data</b>	250 Kbps	250 Kbps
<b>Tegangan yang dibutuhkan</b>	2.8-3.4 V	2.8-3.6 V
<b>Frekuensi</b>	ISM 2.4 GHz	ISM 2.4 GHz
<b>Temperatur alat</b>	-40 – 85 C	-40 – 85 C
<b>Bentu Jaringan</b>	Point to Point, Star, Mesh (With Digi Meshfirmware)	Point to Point, Star, Mesh
<b>Kanla yang tersedia</b>	16 Direct Squence Channels	16 Direct Squence Channels
<b>Pilihan pengaturan</b>	PAN ID, Chanel dan Source/Destination	PAN ID, Chanel dan Source/Destination

Xbee adalah sebuah merek yang dikeluarkan oleh perusahaan Digi International untuk sebuah modul komunikasi

radio. Perangkat Xbee yang diproduksi oleh Digi International memiliki 2 seri tipe yang berbeda. Setiap seri yang diproduksi memiliki standar komunikasi yang berbeda. Xbee seri 1 menggunakan standar komunikasi IEEE 802.15.4. sedangkan Xbee serie 2 dapat diatur menggunakan standar komunikasi IEEE 802.15.4 atau Zigbee. Berikut Tabel 1. 6 merupakan perbandingan antara Xbee seri 1 dan seri 2.

Dapat dilihat pada Tabel 1. 6 Xbee seri 1 memiliki beberapa kelebihan dibandingkan dengan Xbee seri 1. Oleh karena itu pada penelitian ini menggunakan Xbee seri 2 yang memiliki beberapa kelebihan. Dalam ukuran dan bentuk Xbee seri 1 dan Xbee seri 2 tidak ada perbedaan.

### **2.12 Fuzzy Logic**

*Fuzzy logic* adalah sebuah metodologi perhitungan dengan variabel kata-kata (*linguistic variable*), sebagai pengganti berhitung dengan bilangan. *Fuzzy logic* merupakan salah satu komponen pembentuk *soft computing*. *Fuzzy logic* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh seorang ilmuwan Amerika Serikat berkebangsaan Iran dari Universitas California di Berkeley pada tahun 1965. Dasar *fuzzy logic* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan menjadi ciri utama dari penalaran dengan *fuzzy logic* tersebut.

### **2.13 Xbee Shield**

*Xbee shield* digunakan untuk menghubungkan antara arduino dan perangkat Xbee. Penggunaan perangkat Xbee *shield* sangat mudah, karena Xbee *shield* sudah dirancng sesuai dengan perangkat Arduino Uno, Gambar Xbee *shield* bisa dilihat pada



Gambar 1. 8, Xbee *shield* membutuhkan kuat arus minimal 500 mA dalam tegangan 3,3 V.



**Gambar 1. 8. Xbee shield**

*(Halaman ini sengaja dikosongkan)*

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Perancangan merupakan bagian penting dari pembuatan perangkat lunak yang isinya berupa perencanaan-perencanaan secara teknis aplikasi yang dibuat. Sehingga bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Berawal dari deskripsi umum aplikasi hingga perancangan proses, alur dan implementasinya.

#### **3.1 Deskripsi Umum**

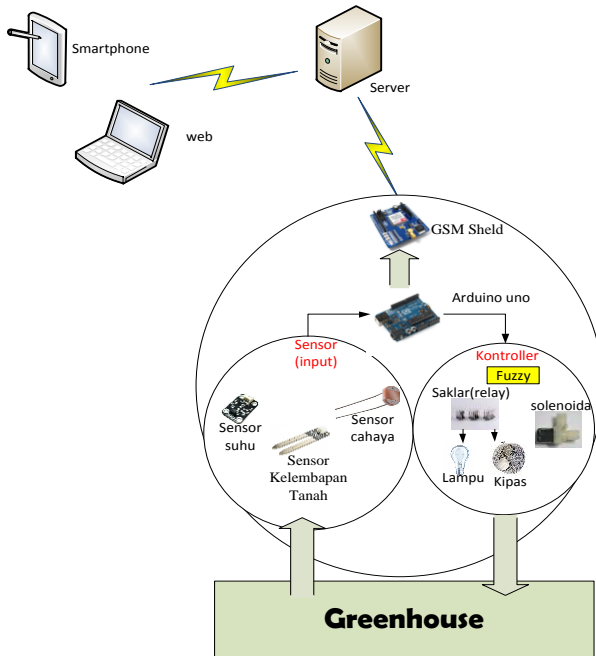
Pada tugas akhir ini akan dikembangkan suatu sistem yang dapat memantu suatu keadaan sebuah ruangan, pada tugas akhir ini diuji cobakan pada sebuah ruangan *green house* yang ada pada BALITKABI (Balai Penelitian Tanaman Kacang dan Umbi) diharapkan dengan adanya sistem yang di bangun dapat mempermudah pemantauan kondisi dalam *green house*. Dengan adanya pemantauan ini diharapkan proses pembenihan dapat dilakukan sepanjang tahun tanpa terkendala faktor cuaca yang kurang mendukung, prinsip kerja sistem yang di bangun pada tugas akhir ini adalah menempatkan beberapa modul sensor dalam ruangan *green house*, sensor berfungsi untuk mengambil data secara analog, sensor dikoordinir dengan mikrokontroler Arduino Uno selain sebagai koordinator Arduino Uno juga mengubah masukan analog modul sensor menjadi data digital. Data yang telah didapat digunakan untuk mengetahui kondisi ruangan *green house* memenuhi standar syarat tumbuh tanaman atau tidak, jika tidak memenuhi syarat tumbuh tanaman maka kontroler akan otomatis hidup dan akan mati ketika syarat tumbuh tanaman sudah mencukupi.

Data yang diambil akan dikirim ke sebuah *server* data akan ditampung pada basis data *server*, data akan ditampilkan kepada pengguna atau pemilik *green house* dalam bentuk aplikasi perangkat telepon genggam pintar berplatform Android. Aplikasi tersebut menampilkan log aktivitas sensor yang ada di dalam

*green house* dengan rentang waktu yang sudah ditentukan, data log aktivitas ditampilkan dalam bentuk grafik secara waktu nyata.

### 3.2 Arsirektur Sistem

Rancangan arsitektur sistem dapat dilihat pada Gambar 3. 1.



Gambar 3. 1. Alur proses penggunaan sistem

Pada sistem pemantauan *green house*, data hasil pemantauan akan dikirimkan kepada *server* menggunakan GPRS (*General Packet Radio Service*) pada perangkat mikrokontroler Arduino Uno perangkat modul sensor pengukur suhu, kelembaban, dan cahaya, GPRS digunakan untuk mengirim data ke *server* untuk disimpan sebagai data riwayat tanaman apabila ada perbedaan data pembandingan maka sistem mikrokontroler akan

mengontrol variabel misal kelembaban tanah kurang maka kran akan otomatis terbuka, jika cahaya dirasa kurang memenuhi maka mikrokontroler akan menyalakan lampu.

### 3.3 Perancangan Kebutuhan Arus Listrik

Perancangan kebutuhan daya digunakan untuk mengetahui kebutuhan daya listrik yang dibutuhkan oleh sistem, perancangan ini sangat penting karena sistem yang akan dibuat membutuhkan catu daya yang cukup karena setiap perangkat yang digunakan membutuhkan suplai arus listrik minimal, jika daya minimal tidak terpenuhi maka sistem tidak bisa berjalan sehingga sistem mengalami kegagalan. Selain kegagalan sistem, akibat kebutuhan daya tidak terpenuhi dapat menurunkan performa dari perangkat sensor yang terpasang, perhitungan dan perancangan kebutuhan daya listrik dapat dilihat pada Tabel 3. 1.

Tabel 3. 1. Perancangan kebutuhan daya

Nama Perangkat	Kebutuhan	
	Arus Listrik	Tegangan Listrik
Arduino Uno	40 mA	9 V
GSM/GPS/GPRS <i>shield</i>	100 mA	6 – 12 V
Xbee <i>shield</i>	500 mA	3,3 V
Xbee S2-pro	30 mA	2.8-3.4 V
Sensor suhu (LM35)	1 mA	4 – 30 V
Sensor cahaya (LDR)	1 mA	
Sensor kelembaban (SEN0057)	35 mA	5 V
Modul waktu (DS1307)	300 nA (0,3 mA)	
<b>Jumlah total</b>	<b>707,3 mA</b>	

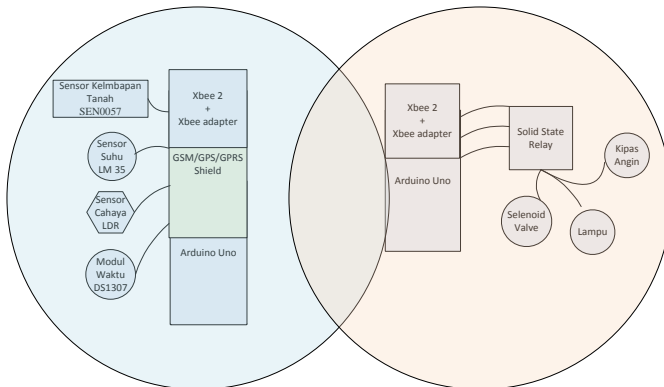
### 3.4 Perancangan Perangkat Keras

Perancangan perangkat keras menjelaskan secara umum bagaimana membangun perangkat keras yang akan dibangun pada sistem ini ada 3 jenis, yaitu perancangan perangkat keras sistem

perawatan, perancangan perangkat keras data pembanding dan perancangan penempatan sensor pada *green house*.

### 3.4.1 Perancangan Perangkat Keras Sistem Perawatan

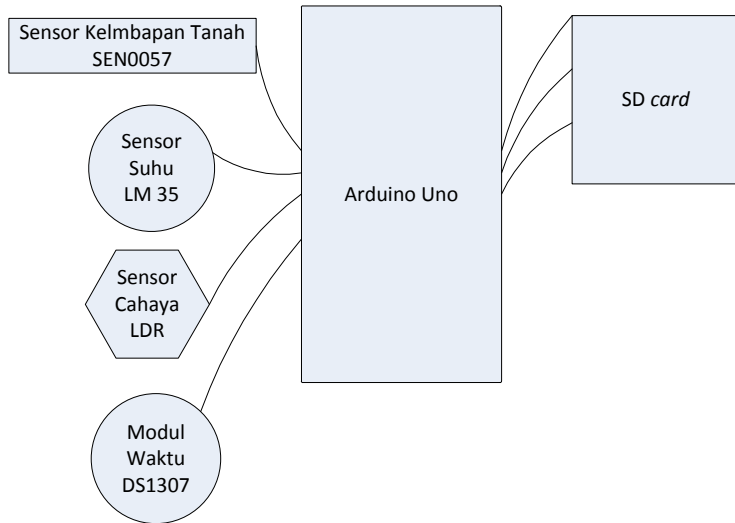
Perancangan perangkat keras sistem perawatan dapat dilihat pada Gambar 3. 2. Pada sistem perawatan menggunakan Arduino Uno, GSM/GPS/GPRS *shield*, relay, serta beberapa modul sensor (suhu (LM35), cahaya (LDR), dan kelembaban tanah (SEN0057)) serta ditambahkan modul waktu (jam digital (DS1307)). Penambahan modul waktu digunakan karena Arduino Uno tidak memiliki *local time*, *local time* digunakan sebagai pembatas sistem perawatan. Karena sistem dirancang untuk bekerja setiap hari antara jam 05:30 – 16:59 WIB.



**Gambar 3. 2. Perangkat Keras Sistem Perawatan**

### 3.4.2 Perancangan Perangkat Keras Data Pembanding

Perancangan perangkat keras data pembanding (seperti Gambar 3. 3) menggunakan satu Arduino Uno, serta beberapa modul sensor (suhu (LM35), cahaya (LDR), dan kelembaban tanah (SEN0057)) modul waktu (jam digital (DS1307) dan SD *card*(SD *card adapter* dan kartu memori/SD *card*), kartu memori/SD *card* digunakan untuk menyimpan data.

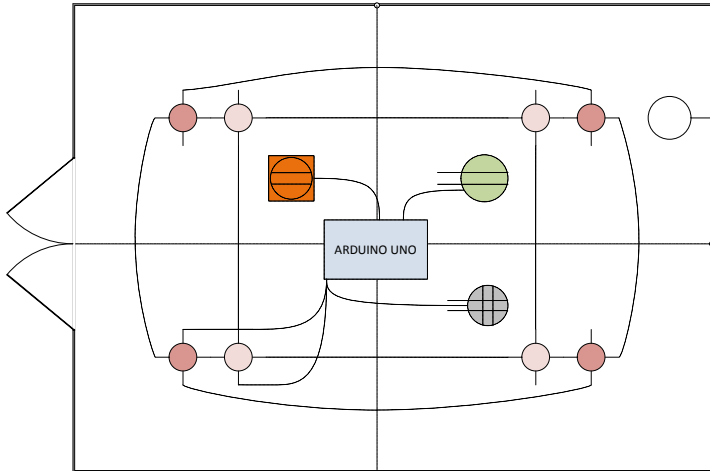


**Gambar 3. 3. Perangkat keras data pembanding**

Untuk merancang perangkat keras data pembanding modul dihubungkan dengan kabel (*jumper*) sedangkan untuk kartu memori/SD *card* dibutuhkan *breadboard* untuk membantu menghubungkan dengan Arduino Uno karena letak pin yang saling berjauhan selain itu dibutuhkan beberapa *jumper* untuk menghubungkan antara Arduino Uno dengan *breadboard*.

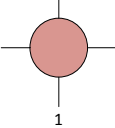
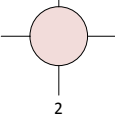

### 3.4.3 Perancangan Penempatan Sensor

Gambar 3. 4 merupakan perancangan penempatan sensor. Perancangan penempatan sensor sangat penting karena berpengaruh terhadap masukan pada sistem perawatan. Perancangan penempatan sensor tetap memperhatikan panjang kabel yang menghubungkan antara sensor dan Arduino Uno, panjang kabel dapat mempengaruhi akurasi sensor.


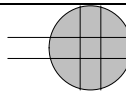
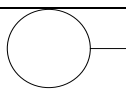
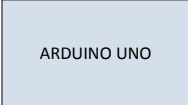


**Gambar 3. 4. Penempatan Sensor**

**Tabel 3. 2. Tabel simbol sensor**

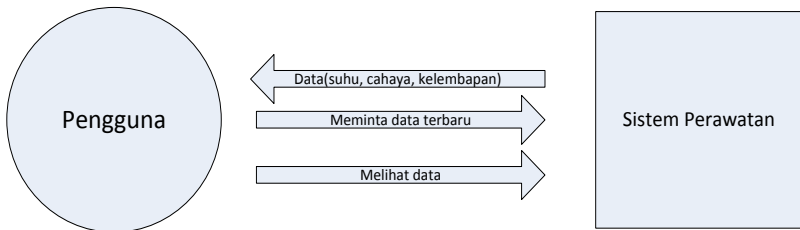
Label	Keterangan
	<p>Simbol lampu 1, lampu bersebelahan dengan lampu 2. Penggunaan 2 lampu untuk menyimulasikan kebutuhan cahaya pada ruangan sesuai dengan batas yang ditentukan.</p>
	<p>Simbol lampu 2, lampu 2 akan hidup ketika cahaya dalam ruangan terlalu gelap pada saat waktu tanaman melakukan fotosintesis, pemasangan lampu 1 dan 2 dengan cara digantung.</p>
	<p>Simbol sensor cahaya, sensor cahaya diletakkan di atas posisi terendah lampu. Sensor cahaya diharapkan tidak terpengaruh oleh cahaya lampu ketika terjadi kondisi ruangan kekurangan cahaya saat tanaman dalam proses fotosintesis di siang hari.</p>



	<p>Merupakan simbol sensor kelembaban tanah, sensor kelembaban tanah diletakkan atau ditancapkan pada tanah sekitar tanaman, kelembaban tanah dihitung dengan cara menghitung besarnya aliran listrik yang dapat di hantarkan oleh tanah.</p>
	<p>Simbol sensor suhu, sensor suhu bisa di letakkan dimana saja tetapi peletakkan akan lebih efektif di tengah ruangan. Diharapkan penempatan tersebut sensor suhu dapat secara maksimal membaca suhu dalam ruangan <i>green house</i></p>
	<p>Simbol kipas angin, kipas angin diletakkan didekat ventilasi udara atau celah sirkulasi udara.</p>
	<p>Simbol mikrokontroler Arduino Uno, mikrokontroler bisa diletakkan di dekat sumber listrik karena dalam mikrokontroler terdapat modul SSR (<i>Solid State Relay</i>) yang membutuhkan tegangan sebesar 12 V untuk mengoprasikannya. Selain itu catu daya untuk Arduino Uno menggunakan adapter 9 V, penggunaan adapter lebih efisien dari pada penggunaan baterai karena Arduino Uno memlakukan proses pengiriman data yang membutuhkan catu daya yang cukup besar.</p>

### 3.5 Perancangan Alir Data

Gambar 3. 5 menjelaskan proses secara umum pada sistem perawatan. Dalam perancangan diagram terdapat entitas pengguna dan proses utama sistem perawatan. Komunikasi data antara sistem perawatan dan pengguna membutuhkan perantara servis, servis mengirim data dari *server* keperangkat yang digunakan pengguna.



**Gambar 3. 5. Perancangan Alir data**

Pengguna dapat melihat data melalui proses yang dijalankan oleh sistem perawatan tanaman kacang hijau. Pengguna dapat mengetahui kondisi pada ruangan *green house* secara waktu nyata, data ditampilkan dalam bentuk grafik dengan rentang pembaruan data pada grafik sekitar 3 menit. Selain melihat data yang ditampilkan dalam grafik, pengguna dapat meminta data terbaru kondisi ruangan *green house* yang telah tersimpan pada *server*.

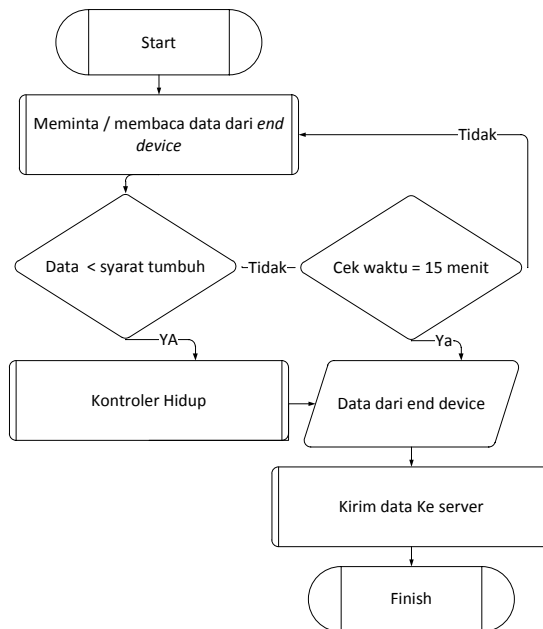
### 3.6 Diagram Alir Aplikasi Sistem

Alur dari setiap proses yang terdapat pada aplikasi digambarkan pada diagram alir untuk memudahkan pemahaman secara garis besar proses yang ada pada sistem, diagram alir sistem terdiri dari:

- Diagram alir aplikasi sistem.
- Diagram alir sensor.
- Diagram alir sensor suhu.
- Diagram alir sensor cahaya.
- Diagram alir sensor kelembaban tanah.
- Diagram alir sensor pengiriman data.
- Diagram alir penampilan data.

### 3.6.1 Diagram Alir Sistem

Masukan data untuk sistem berasal dari modul sensor atau *end device* yang telah terpasang masukan data didapat setiap waktu sesuai dengan *delay* pengambilan data yang sudah diatur pada sistem perawatan. Data yang didapat dari modul sensor akan selalu dicek apakah masukan data dari sensor sesuai dengan syarat tumbuh tanaman jika masukan tidak sesuai kontroler akan otomatis hidup sesuai dengan kebutuhan yang harus dipenuhi syarat tumbuh tanaman.



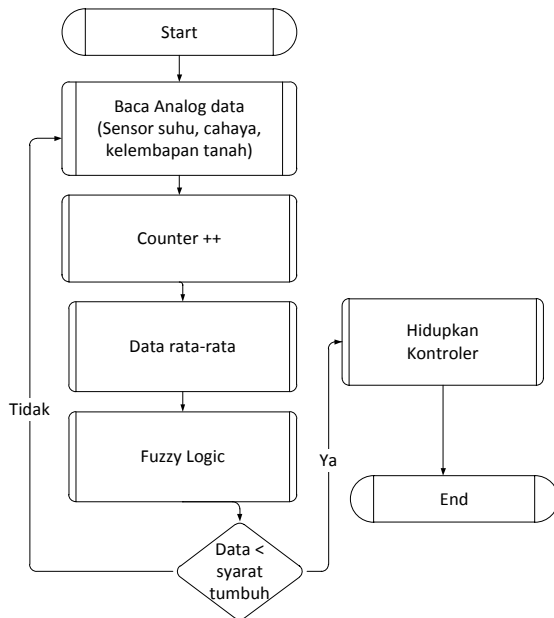
**Gambar 3. 6. Diagram alir sistem**

Seperti pada Gambar 3. 6 jika ada masukan data yang tidak sesuai akan dikirim ke *server*, data yang dikirim meliputi data kelembaban tanah, suhu dan cahaya. Semua masukan sistem akan dikirim ke *server* sebagai data log aktivitas. Apabila dalam rentang waktu 15 menit keadaan ruangan *green house* normal

maka data yang akan dikirimkan adalah data kondisi ruangan yang terakhir tiap 15 menit.

### 3.6.2 Diagram Alir Sensor

Masukan data pada sistem adalah sensor, diagram alir sistem dapat dilihat pada Gambar 3. 7, diperlukan masukan data yang stabil sebagai masukan sistem, sensor akan mengambil data sebagai masukan setiap 1 detik, data yang diperoleh akan ditampung dan dibagi dengan banyaknya data yang diambil atau dibagi dengan *counter*, kemudian diperoleh data rata-rata.



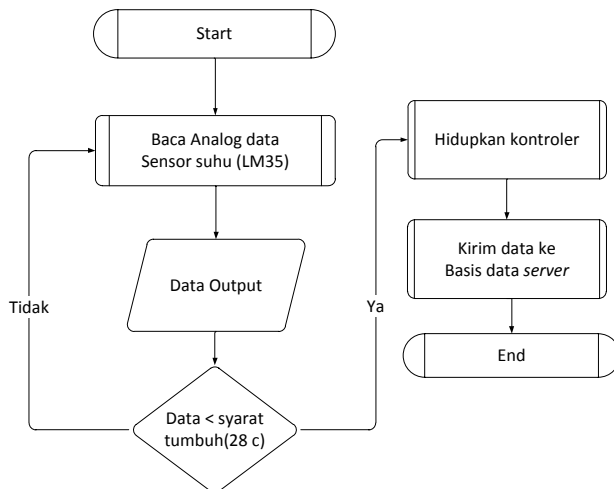
**Gambar 3. 7. Diagram alir sensor**

Setelah data rata-rata diperoleh maka data akan dimasukkan dalam algoritma *fuzzy*, data yang telah dimasukkan dalam algoritma *fuzzy* digunakan sebagai masukan sistem, jika

masuk data sensor tidak sesuai maka kontroler akan otomatis aktif.

### 3.6.3 Diagram Alir Sensor Suhu

Sensor suhu membaca keadaan suhu ruangan setiap 1 menit sistem pengolahan data pada sensor suhu dijelaskan pada sistem diagram alir sensor. Data yang telah diolah merupakan data output sensor dan masukan data sistem, diagram alir sensor dapat dilihat pada Gambar 3. 8.



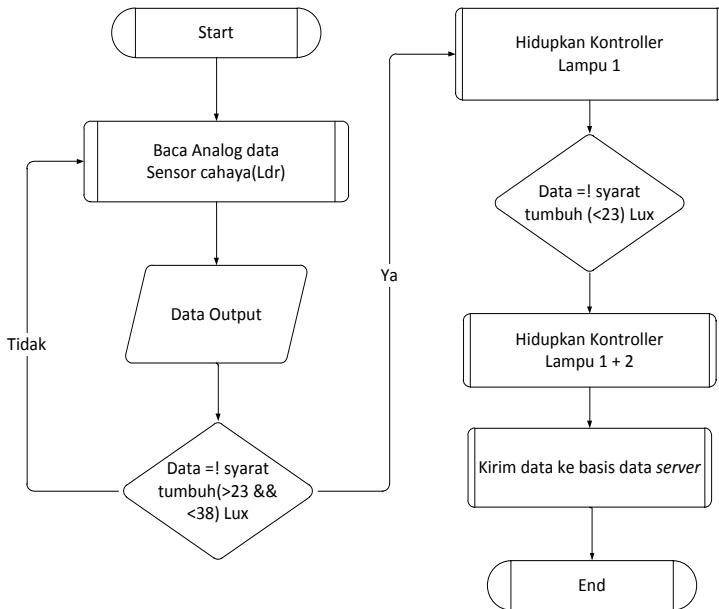
**Gambar 3. 8. Diagram alir sensor suhu**

Data keluaran akan dicek apabila data yang dihasilkan oleh sensor kurang dari 28 °C ( $< 28\text{ }^{\circ}\text{C}$ ) sistem akan otomatis menghidupkan saklar relay untuk menghidupkan kipas angin. Ketika saklar kipas angin hidup data akan dikirimkan ke *server* sebagai data riwayat atau data log aktivitas.

### 3.6.4 Diagram Alir Sensor Cahaya

Sensor cahaya mengambil data setiap 1 menit data diolah dengan cara melakukan rata-rata data dan menggunakan algoritma

*fuzzy* untuk masukan pada sistem. Sistem akan otomatis menghidupkan saklar relay untuk lampu ketika data output tidak memenuhi syarat tumbuh tanaman, jika data keluaran  $> 23 \ \&\& \ < 38$  Lux maka rangkaian lampu 1 akan otomatis dihidupkan sedangkan jika data keluaran  $< 23$  Lux maka rangkaian lampu akan dihidupkan, diagram alir sensor dapat dilihat pada Gambar 3.9.

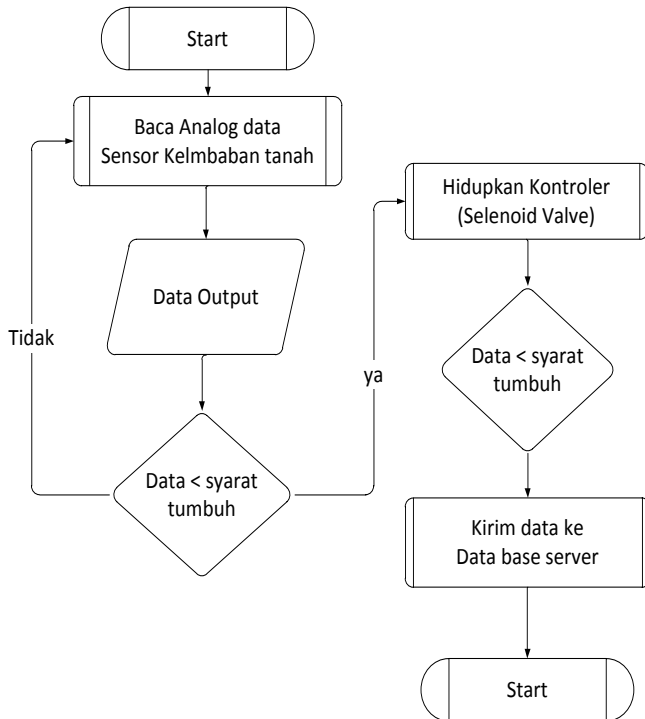


**Gambar 3. 9. Diagram alir sensor cahaya**

### 3.6.5 Diagram Alir Kelembaban Tanah

Sistem kerja pengambilan data pada sensor kelembaban tanah tidak berbeda dengan sistem pengambilan data pada sensor suhu dan cahaya. ketika masukan data sensor tidak sesuai atau tidak memenuhi syarat tumbuh tanaman maka sistem akan

otomatis menghidupkan kran solenoida, masukan data terakhir saat masukan data melebihi batas syarat tumbuh tanaman akan dikirim ke *server* sebagai data log aktivitas.

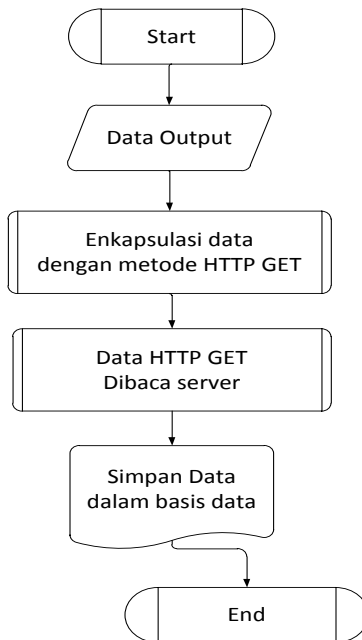


**Gambar 3. 10. Diagram alir kelembaban tanah**

Gambar 3. 10 merupakan gambar diagram alir pengambilan data dari sensor kelembaban tanah, sistem perawatan tidak mendukung *multithread*, sistem tidak bisa menangani masukan data sensor dalam 1 waktu dan melakukan kontrol secara bersamaan.

### 3.6.6 Diagram Alir Pengirima Data

Untuk pengiriman data menggunakan perintah HTTP GET. Perintah HTTP GET digunakan untuk meminta data dari *resource* yang spesifik, pada penelitian ini adalah halaman web tugas akhir. Sedangkan pada perangkat sistem perawatan melakukan perintah HTTP POST untuk mengirm data pada server.



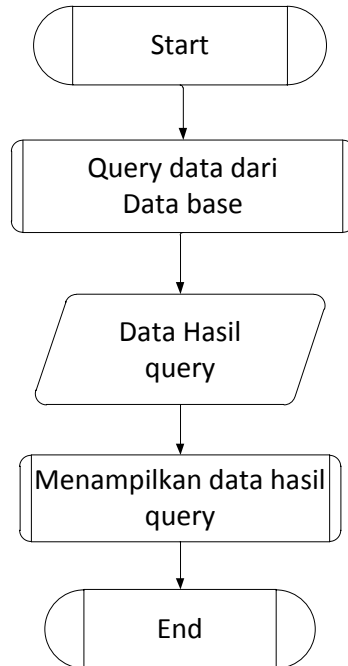
Gambar 3. 11. Diagram alir pengirima data

### 3.6.7 Diagram Alir Penampilan Data

Gambar 3. 12 merupakan diagram alir penampilan data log aktivitas, data log aktivitas berisi informasi dari sensor cahaya, suhu, dan kelembaban yang disimpan dalam basis data *server*. Data ditampilkan dalam aplikasi telepon genggam pintar berbasis Android sehingga pengguna dapat dengan mudah



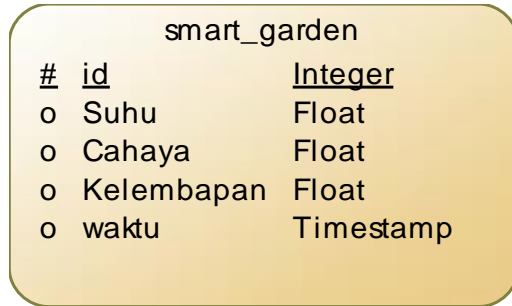
memantau keadaan ruangan *green house*. Data disajikan dalam bentuk grafik garis sehingga pengguna dapat dengan mudah memahami data yang disajikan.



**Gambar 3. 12. Diagram alir penampilan data**

### **3.7 Perancangan Basis Data Sistem**

Basis data dalam sistem dirancang menggunakan MySQL. CDM (*Conceptual Data Model*) dapat dilihat pada Gambar 3. 13. Entitas *smart\_garden* digunakan sebagai menyimpan data log aktivitas sistem yang telah dibuat penjelasan tentang entitas *smart\_garden* dapat dilihat pada Tabel 3. 3.



**Gambar 3. 13. Perancangan basis data sistem**

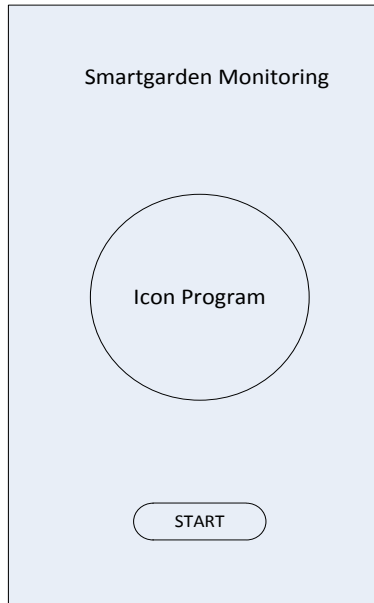
**Tabel 3. 3. Entitas smart\_garden**

Nama Properti	Tipe Data	Keterangan
<b>Id</b>	Int	Pengenalan entitas
<b>Suhu</b>	Float	Nilai suhu yang diterima
<b>Cahaya</b>	Float	Nilai cahaya yang diterima
<b>Kelembaban</b>	Float	Nilai kelembapan yang diterima
<b>Waktu</b>	Timestamp	Waktu data diterima

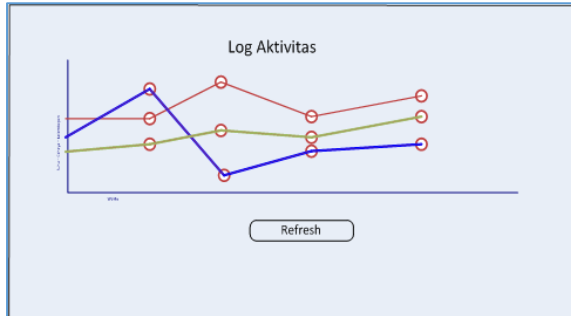
### **3.8 Perancangan Antarmuka Pengguna Telepon Genggam Pintar Android**

Sistem perawatan tidak memiliki desain antar muka, karena sistem perawatan menggunakan Arduino IDE untuk mengatur sistem atau mengunggah baris kode untuk sistem. Pengguna tidak bisa merubah sistem yang telah diatur terkecuali pengguna bisa mengunggah program untuk sistem yang baru. Konfigurasi dan sistem dan *server* dilakukan saat pertamakali saat alat akan dipasang.

Desain antar muka terdapat pada aplikasi pengguna, desain antar muka aplikasi perangkat bergerak berbasis Android dapat dilihat pada Gambar 3. 14 dan Gambar 3. 15. Gambar 3. 14 merupakan desain tampilan halaman depan untuk sistem pemantauan pada sistem.



**Gambar 3. 14. Perancangan antarmuka pengguna telepon genggam pintar Android**



**Gambar 3. 15. Perancangan antarmuka grafik log aktivitas**

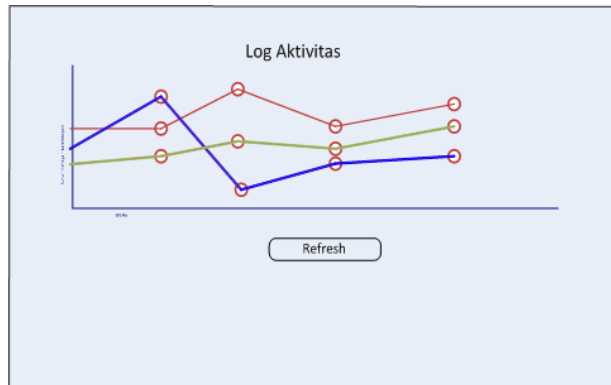
Antarmuka pada Gambar 3. 15 merupakan antar muka untuk menampilkan informasi pemantauan, data yang ditampilkan merupakan data yang ada dan sesuai dengan data pada basis data *server*. Antarmuka akan menampilkan 3 buah data, data suhu, intensitas cahaya, dan kelembaban tanah pada sumbu Y sedangkan pada sumbu X adalah waktu, waktu didapat ketika data dimasukkan kedalam tabel basis data dengan kata lain waktu yang ditampilkan dalam antarmuka pengguna adalah waktu *server*.

### **3.9 Perancangan Antarmuka Pengguna Web**

Desain perancangan pengguna web terdapa 2 bagian yaitu bagian antarmuka grafik dan tabel log aktivitas. Pengguna dapat memilih rentan waktu yang akan ditampilkan dalam tabel dan grafik log aktivitas.

#### **3.9.1 Perancangan Antarmuka Pengguna Web Grafik Log Aktivitas**

Data yang ditampilkan grafik log aktivitas pada web sama dengan yang ada pada pengguna Android. Perancangan antarmuka pengguna web dapat dilihat pada Gambar 3. 16



**Gambar 3. 16. Perancangan antarmuka pengguna web grafik log aktivitas**

### 3.9.2 Perancangan Antarmuka Tabel Log Aktivitas Pengguna Web

Terdapat kolom batas untuk menentukan rentang waktu yang ingin ditampilkan oleh pengguna, pengguna tampilan tabel dan kolom rantang waktu dapat dilihat pada Gambar 3. 17.

Kolom input batas bawah Tanggal			
Kolom input batas atas tanggal		Tombol pilih	
Suhu	Cahaya	Kelembapan	Waktu

**Gambar 3. 17. Perancangan antarmuka tabel log aktivitas pengguna web**

*(Halaman ini sengaja dikosongkan)*

## **BAB IV IMPLEMENTASI**

Bab ini merupakan bahasan mengenai implementasi dari perancangan perangkat lunak yang telah dijabarkan pada bab III. Implementasi meliputi perangkat sistem perawatan dan aplikasi pemantauan pada Android.

### **4.1 Lingkungan Implementasi**

Dalam merancang dan mengimplementasikan sistem perawatan pada tanaman kacang-kacangan digunakan beberapa perangkat pendukung sebagai berikut.

#### **4.1.1 Lingkungan Perangkat Keras**

Perangkat keras yang digunakan dalam mengembangkan sistem perawatan kacang-kacangan menggunakan beberapa perangkat pendukung sebagai berikut:

- Laptop ASUS seri K-42j dengan spesifikasi prosessor core i5 dengan RAM 6 GB untuk memori skunder dan 500 GB untuk memori skunder, serta VGA NVIDIA GFORCE 310 M 1 GB .
- 1 (Satu) buah perangkat Arduino Uno dengan.
- 1 buah Arduino Uno GSM/GPS/GPRS *shield* dari DFRobot.

#### **4.1.2 Lingkungan Perangkat Lunak**

Perangkat lunak yang dalam implementasi sistem perawatan kacang-kacangan, yaitu:

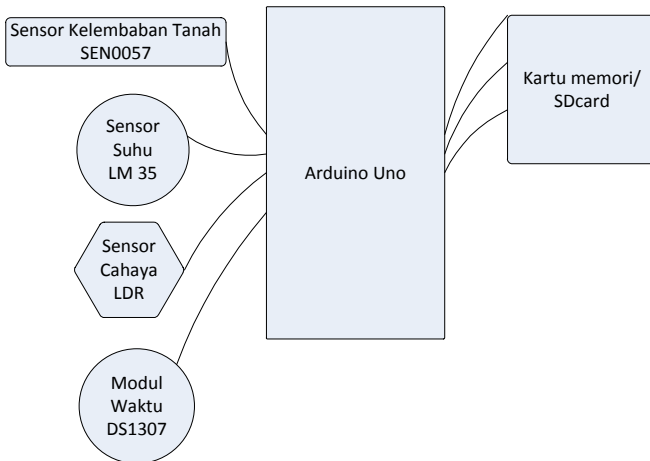
- Microsoft Windows 7 64-bit sebagai sistem operasi
- MySQL digunakan untuk implementasi basis data
- Mozilla Firefox digunakan sebagai browser dalam tahap uji coba aplikasi *server*
- Arduino Development Kit sebagai IDE untuk mengimplementasikan pada perangkat keras Arduino Uno.

- PhoneGap sebagai IDE untuk mengimplementasikan pada aplikasi telepon genggam pintar Android.
- Microsoft Visio 2010 untuk melakukan perancangan flowchart sistem monitoring.

#### 4.2 Implementasi Perangkat Keras Data Pemanding

Pada implementasi perangkat keras sistem perawatan di butuhkan beberapa perangkat keras, sebagai berikut:

- 1 (satu) buah Arduino Uno.
- 1 (satu) buah modul linear sensor suhu LM35.
- 1 (satu) buah modul sensor cahaya.
- 1 (satu) buah modul sensor kelembaban tanah.
- 1 (satu) buah modul jam/ DS1307.
- 1 (satu) buah SD *card* adapter.
- 1 (satu) buah SD *card* 2 GB
- Beberapa *jumper*.



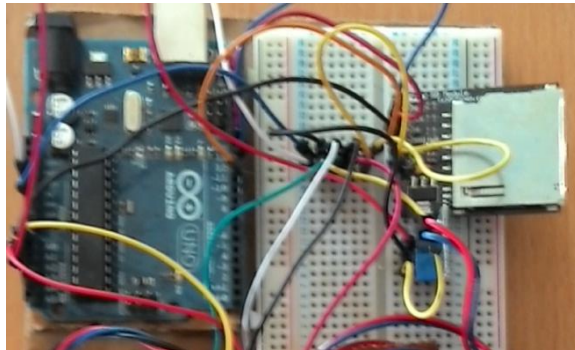
**Gambar 4. 1. Perangkat keras data pemanding**

Arduino Uno berfungsi sebagai pusat dari pengolah data yang diterima, data yang diterima dari 4 buah modul/sensor (waktu,



cahaya, kelembaban dan modul jam/waktu). Data yang diterima akan disimpan dalam *SD card*, data yang tersimpan dalam bentuk *.csv*. Perancangan perangkat keras data pembanding dapat dilihat pada Gambar 4. 1.

Perangkat keras data pembanding tidak terhubung dengan jaringan, karena fungsi dari rangkaian perangkat keras ini hanya mengambil data dan menyimpan data tersebut dalam sebuah kartu memori/*SD card*. Semua data dari modul akan tersimpan, waktu (tgl, jam, menit, detik) diambil dari modul DS1307, kemudian data suhu, cahaya, dan kelembaban. Untuk menulis data pada kartu memori/*SD card* Arduino Uno hanya bisa 1(satu) kali dalam penulisan data pada file. Perlu diingat Arduino Uno tidak mendukung *multithreading*. Apabila pemasangan modul telah dilakukan maka hasil akhir perangkat keras data pembanding dapat di lihat pada Gambar 4. 2.



Gambar 4. 2. Perangkat data pembanding terpasang

### 4.3 Implementasi Perangkat Lunak pada Perangkat Keras Data Pembanding

implementasi perangkat lunak pada perangkat keras data pembanding terdapat 1 fungsi yang digunakan untuk perintah menulis data pada file yang ada dalam *SD card*. Fungsi untuk menulis data pada memori dapat dilihat pada Gambar 4. 3.

Prosedur pengambilan data pembandingan	
Masukan: Analog masukan dari modul/ sensor	
	Buka file smartlog.csv
	Tulis data yang telah dibaca sensor pada SD card
	Tutup file smartlog.csv
Keluaran : Data tersimpan pada SD card	

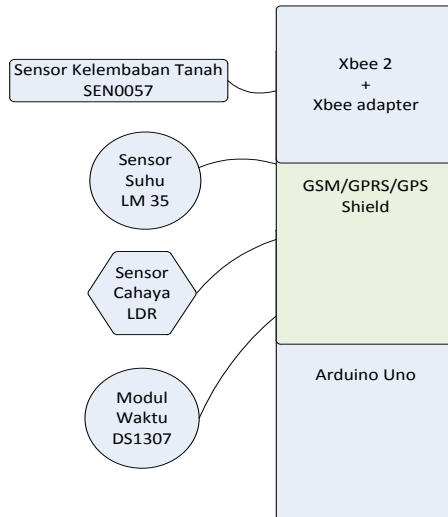
**Gambar 4. 3. Pengambilan data pembandingan**

#### **4.4 Implementasi Perangkat Keras Sistem Perawatan Kacang-Kacangan**

Implementasi perangkat keras pada sistem perawatan secara keseluruhan hampir sama dengan perangkat keras data pembandingan, tetapi yang membedakan pada perangkat keras sistem perawatan adalah penambahan beberapa kontroler, berikut beberapa perangkat keras untuk implementasi sistem perawatan:

- 1 (satu) buah Arduino Uno.
- 1 (satu) buah modul linear sensor suhu LM35.
- 1 (satu) buah modul sensor cahaya.
- 1 (satu) buah modul sensor kelembaban tanah.
- 1 (satu) buah modul jam/ DS1307.
- 1 (satu) buah *solenoid valve*/kran solenoid.
- 1 (satu) kipas angin.
- 2 (dua) buah rangkaian seri lampu.
- 4 (empat) buah relay yang tersambung dalam 1 (satu) buah modul relay 12 V.
- 1 (satu) buah GSM/GPS/GPRS *shield*.
- Beberapa *jumper*.
- 1 buah Xbee 2 pro series.

Untuk merangkai perangkat keras sistem perawatan sudah dijelaskan pada BAB III. Rangkaian sistem perawatan bisa dilihat pada Gambar 4. 4.

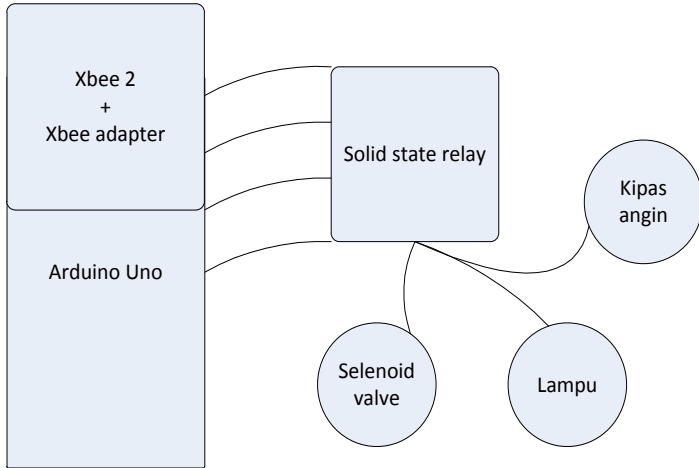


**Gambar 4. 4. Perangkat keras sistem perawatan kacang-kacangan**

Setelah perangkat keras terpasang seperti Gambar 4. 4 di atas, hasil dari rangkaian perangkat keras dapat dilihat pada Gambar 4. 5.



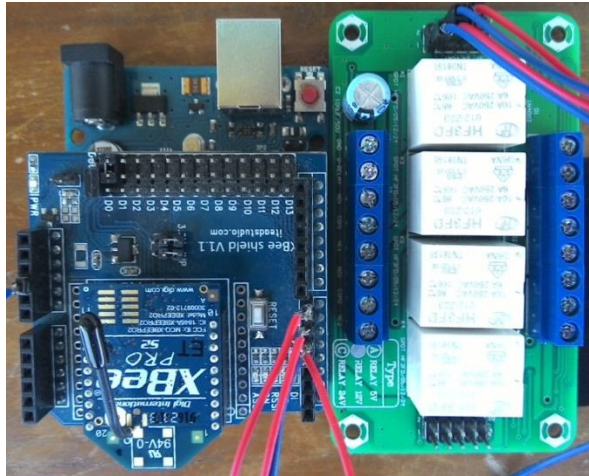
**Gambar 4. 5. Perangkat koordinator terpasang**



**Gambar 4. 6. Perangkat *end device***

Pada sistem ini GSM/GPS/GPRS *shield* sebagai sistem komunikasi data antara sistem perawatan dan basis data *server*, selain GSM/GPS/GPRS *shield* terdapat modul Xbee yang digunakan untuk komunikasi antara koordinator dan *end device*, fungsi dari Xbee *end device* adalah untuk mengontrol kontroler yang terhubung dengan solid state relay dapat di lihat pada Gambar 4. 7. Pada sistem perawatan juga terdapat 4 buah modul sensor (suhu, cahaya, kelembaban dan waktu) serata modul relay yang digerakkan dengan 12 V, saklar otomatis terhubung dengan mikrokontroler Arduino Uno, modul relay menerima perintah dari mikrokontroler Arduino Uno.

Data yang dibaca oleh sensor tidak dimasukkan kedalam SD *card* karena SD *card* adapter tidak dipasang pada perangkat keras sistem perawatan, tetapi data yang dibaca akan disimpan pada basis data *server* sesuai dengan implementasi perangkat lunak. Data yang tersimpan pada basis data yang ada di *server* akan di tampilkan pada perangkat pengguna.



Gambar 4. 7. End device dan relay

#### 4.5 Implementasi Perangkat Lunak pada Perangkat Keras Sistem Perawatan Kacang-Kacangan

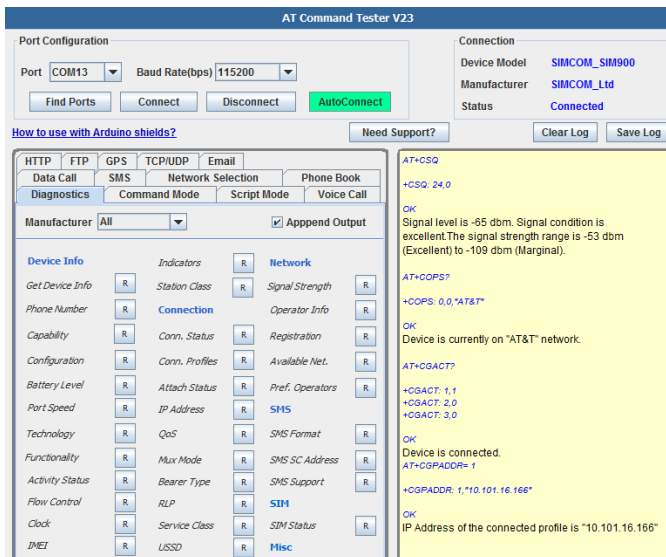
Implementasi perangkat lunak pada sistem perawatan terdapat beberapa 3 fungsi pada mikrokontroler Arduino Uno, fungsi yang digunakan yaitu: fungsi ambil data dari modul sensor pada Xbee koordinator, fungsi pengiriman data menggunakan modul GSM/GPS/GPRS *shield* fungsi pengoprasian kontroler menggunakan Xbee *end device*.

Sebelum implentasi perangkat lunak pada perangkat keras sistem perawatan khususnya modul GSM/GPS/GPRS *shield* perlu menggunakan modul penguji yang disediakan oleh m2msupport, tampilan modul penguji dapat di lihat pada Gambar 4. 8.

Hal yang perlu diperhatikan dalam menggunakan GSM/GPS/GPRS *shield* adalah catu daya yang digunakan harus setabil minimal menggunakan daya 6 - 12 V dan kuat arus 1000 mA, pada implementasi ini mengunkan catu daya 9 V dengan arus 1000 mA. Selain catu daya yang perlu diperhatikan adalah posisi saklar S1 dan saklar S2 berikut penjelasan posisi saklar yang

digunakan saat pertamakali penggunaan atau mengunggah *sketch* pada Arduino Uno:

- S1 pada posisi *Prog* dan S2 pada posisi Arduino Uno jika ingin mengendalikan modul GSM/GPS/GPRS *shield* menggunakan Arduino Uno. Setelah *sketch* terunggah pada perangkat Arduino Uno saklar S1 digeser ke posisi *Comm*.
- Untuk mengendalikan modul GSM/GPS/GPRS *shield* menggunakan modul pengujian maka S1 berada pada posisi *Prog* S2 pada posisi USB. Setelah *sketch* diunggah pada modul Arduino Uno yang terhubung dengan GSM/GPS/GPRS *shield* maka saklar S1 pada posisi *Comm*.



**Gambar 4. 8. m2msupport modul pengujian**

Pengujian modul GSM/GPS/GPRS *shield* menggunakan modul pengujian memudahkan untuk mengetahui kendala pada perangkat keras, perintah yang dijalankan untuk pengujian dan

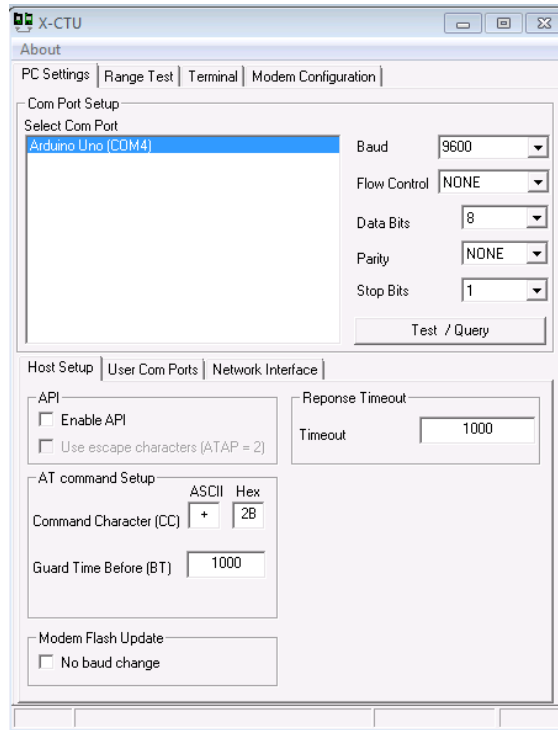
*sketch* adalah sama yaitu menggunakan AT-Comand. Indikasi jika program yang dijalankan berjalan dengan baik pada modul GSM/GPS/GPRS *shield* adalah indikasi lampu berwarna hijau pada label *start*, apabila lampu indikator tidak hidup maka perintah harus diulang/*restart*.

#### 4.5.1 Implementasi Perangkat Lunak pada Xbee Koordinator

Untuk implentasi perangkat lunak pada Xbee koordinator perlu memperbarui *firmware* menggunakan perangkat lunak X-CTU keluaran Digi, proses memperbarui *firmware* dapat dilihat pada Gambar 4. 10. Sedangkan untuk implementasi perangkat lunak pada perangkat Xbee koordinator dapat dilihat pada Gambar 4. 9.

Prosedur sistem perawatan	
Masukan: Analog masukan dari modul/ sensor	
	Baca masukan dari analog masukan dan digital masukan
	Jika waktu > 05:00 dan <17.00
	<ul style="list-style-type: none"> <li>- Nyalakan lampu dan kirim data kondisi <i>greenhouse</i> jika intensitas cahaya &lt; syarat tumbuh tanam</li> <li>- Nyalakan kipas dan kirim data kondisi <i>greenhouse</i> jika suhu &lt; syarat tumbuh tanam</li> <li>- Nyalakan kran dan kirim data kondisi <i>greenhouse</i> jika tanah &lt; dari syarat tumbuh</li> </ul>
	Jika waktu = 05:30-05:50 dan waktu = 17:30 - 17.50 <ul style="list-style-type: none"> <li>- Nyalakan penyiraman otomatis</li> </ul>
	Jika menit (00, 15, 30, dan 45) <ul style="list-style-type: none"> <li>- Kirim data kondisi <i>greenhouse</i></li> </ul>
Keluaran:Data kondisi <i>greenhouse</i> tersimpan	

**Gambar 4. 9. Implementasi Xbee koodinator**



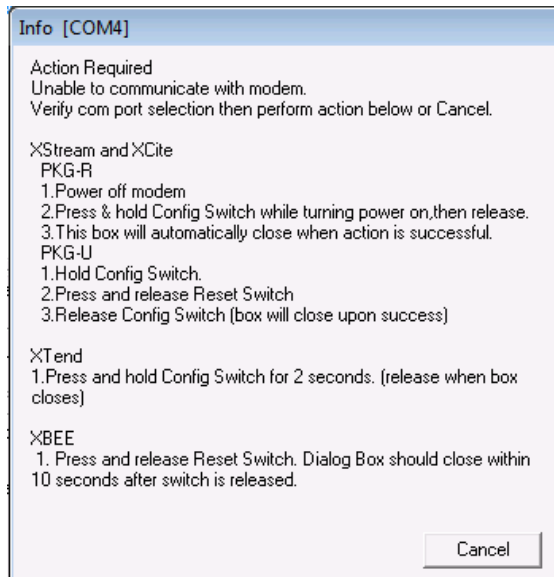
**Gambar 4. 10. X-CTU**

Pada penelitian ini digunakan *baud rate* 9600, setelah port terhubung lakukan *tes/query* untuk mengetahui apakah perangkat Xbee sudah *firmware* terbaru.

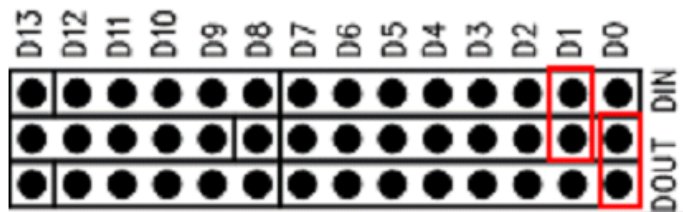
Jika gagal dalam test/query maka akan muncul peringatan seperti pada Gambar 4. 11 beberapa hal yang perlu dilakukan apabila gagal dalam test/query pada perangkat Xbee adalah:

- Pastikan Xbee *shield* mendapat catu daya yang cukup.
- Restart adapter Xbee.
- Pastikan pin *jumper* DIN dan DOUT pada posisi yang benar.



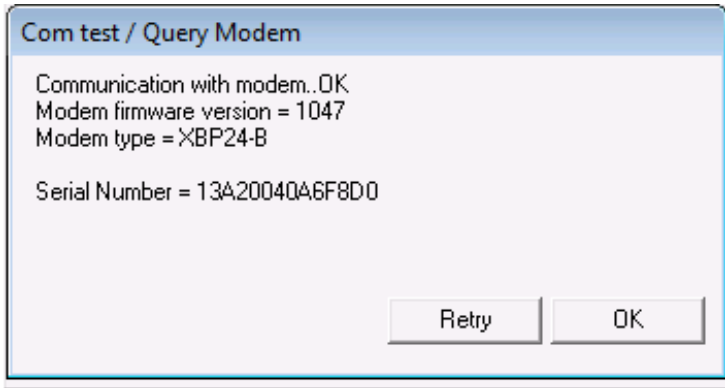


Gambar 4. 11. Pengujian gagal



Gambar 4. 12. Pin DOUT

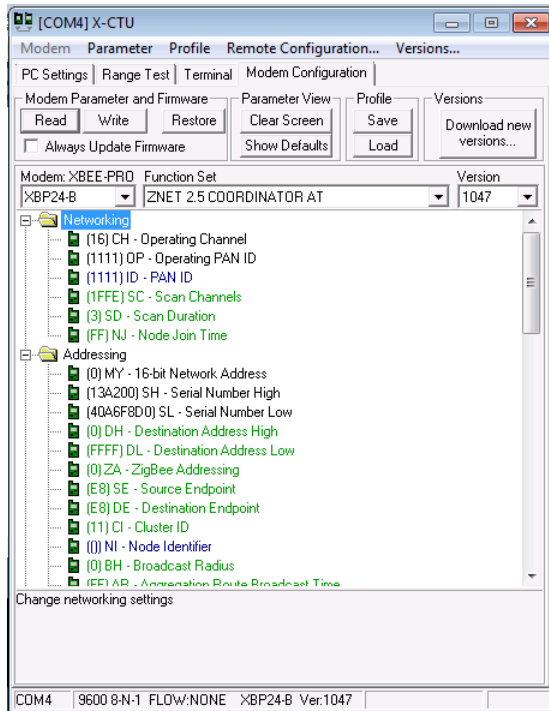
Gambar 4. 12 menunjukkan posisi jumper ketika modul Xbee diakses (tes dan memperbaiki *firmware*) Posisi *jumper* saat memperbaiki *firmware* telah diunggah dan menghubungkan Xbee dengan Xbee koordinator.



Gambar 4. 13. Uji coba berhasil

Gambar 4. 13 merupakan tampilan ketika perangkat lunak X-CTU berhasil melakukan tes/query pada perangkat Xbee, setelah proses pengujian selesai perangkat Xbee *firmware* dapat diperbarui. Pada penelitian ini menggunakan modem tipe XBP24-B dengan fungsi set ZNET 2.5 AT karena dalam implementasi perangkat lunak Xbee koordinator maka fungsi set menggunakan ZNET KOORDINATOR 2.5 AT dengan versi *firmware* 1047. Versi *firmware* antara koordinator dengan *end device* harus sama serta PAN ID harus diatur sama, berikut Gambar 4. 14. *Setup firmware* menjelaskan pengaturan *firmware* pada perangkat Xbee koordinator.

Terlihat pada Gambar 4. 14 menggunakan PAN ID diganti menjadi 1111. Pengaturan destination address High(DH) menjadi 0 dan destination address low (DL) dirubah menjadi 0 untuk pesan *broadcast* kepada perangkat Xbee *end device*.



Gambar 4. 14. Setup firmware

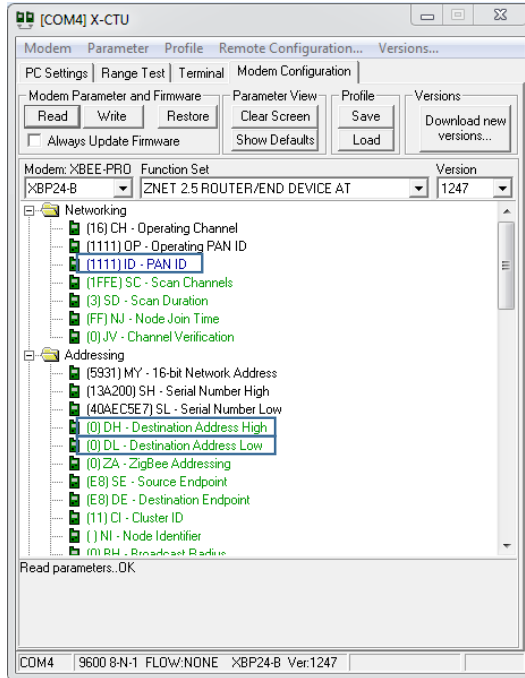
#### 4.5.2 Implementasi Perangkat Lunak pada Perangkat Xbee *End Device*

Untuk implementasi perangkat lunak pada perangkat Xbee *end device* langkah-langkah memperbarui *firmware* tidak jauh beda dengan memperbarui *firmware* Xbee koordinator.

Fungsi *read* berfungsi untuk melihat *firmware* yang telah digunakan sebelumnya dan *write* digunakan memperbarui *firmware* yang baru untuk perangkat Xbee *end device*.

Pada penelitian ini perangkat *end device* menggunakan *firmware* modem XBP24-B dengan *function set* ZNET 2.5 Router/END DEVICE AT versi 1247. Pengaturan PAN ID 1111 adalah PAN ID yang sama dengan PAN ID Xbee sedangkan

koordinasi *destination address High* (DH) dan *destination address low* (DL) disamakan dengan Xbee koordinator, secara otomatis Xbee *end device* akan mengikuti jaringan yang telah dibuat oleh Xbee koordinator yang telah dibuat sebelumnya, dapat dilihat pada Gambar 4. 15.



**Gambar 4. 15. Read firmware**

#### 4.6 Implementasi Pada Server

Implementasi *server* pada sistem perawatan pada *server* menggunakan MySQL. Implementasi pada *server* seperti yang sudah dijelaskan pada BAB II.

## 4.7 Implementasi Antarmuka Pengguna Android

Implementasi antarmuka pengguna terbagi menjadi 2 bagian halaman utama dan tampilan data grafik data.

### 4.7.1 Implementasi Antarmuka Halaman Utama

Gambar 4. 16 merupakan tampilan halaman utama pada aplikasi Android. Pengguna dapat menekan tombol monitoring atau memilih menu yang ada pada *sidebar* untuk melihat data log aktivitas.



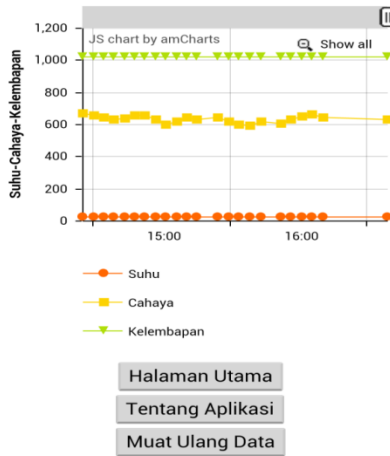
Pilih menu Monitoring untuk melihat Log aktifitas oleh alat yang sudah terpasang di greenhouse. data yang di tampilkan merupakan data suhu, intensitas cahaya dan kelembapan tanah. pengguna dapat memilih rentan waktu yang akan ditampilkan, data di tampilkan dalam bentuk grafik garis dan tabel data.

**Gambar 4. 16. Tampilan utama aplikasi Android**

### 4.7.1 Implementasi Antarmuka Grafik Log Aktivitas

Gambar 4. 17 atas merupakan data dari log aktivitas yang telah disimpan dalam basis data *server*, data ditampilkan dalam bentuk grafik garis. Pengguna dapat melihat dengan detail dari log aktivitas dengan mengeser *slide bar* yang ada di atas grafik, selain itu pengguna dapat melihat data log aktivitas dalam bentuk web untuk lebih detail dari data log aktivitas yang ditampilkan.

## Grafik Log Aktifitas



Gambar 4. 17. Grafik log aktivitas pada Android

### 4.8 Implementasi Antar Pengguna Web

Selain dengan menggunakan perangkat bergerak telepon genggam pintar, pengguna juga dapat melihat data log aktivitas dalam bentuk halaman web. Tampilan antar muka halaman web tidak jauh berbeda dengan aplikasi Android, untuk implementasi antar muka pengguna web terdapat beberapa tampilan yang ada pada halaman web, sebagai berikut:

#### 4.8.1 Implementasi Antarmuka Grafik Log Aktivitas

Implementasi grafik pada halaman web dapat dilihat pada Gambar 4. 18. Tampilan yang ada pada halaman web tidak jauh berbeda dengan tampilan grafik yang ada pada aplikasi perangkat Android.



**Gambar 4. 18. Tampilan log aktivitas halaman web**

#### 4.8.1 Implementasi Antarmuka Tabel Log Aktivitas

Implementasi log aktivitas pada halaman web dapat dilihat pada Gambar 4. 19. Pada implementasi data log aktivitas dalam bentuk tabel memungkinkan pengguna untuk memilih rentang waktu data yang akan ditampilkan, setelah rentang waktu dipilih maka data akan ditampilkan dalam bentuk tabel dan grafik garis. Dalam bentuk data yang ditampilkan dalam bentuk tabel memungkinkan pengguna untuk mengurutkan data suhu, cahaya, kelembaban tanah dan waktu.

Batas Bawah:	<input type="text"/>
Batas Atas:	<input type="text"/>
<input type="button" value="Submit Query"/>	

Show 10 ▾ entries Search:

22	551	479	2014-05-29 13:51:53
24	555	566	2014-05-29 12:22:54
25	450	400	2014-05-08 11:31:04
25	600	688	2014-05-29 11:52:11
26	400	800	2014-05-09 11:35:30
26	588	568	2014-05-29 12:26:00
26	400	800	2014-05-29 21:39:35
26	809	676	2014-06-02 00:49:07
27	564	389	2014-06-01 04:00:39
27.6	500	800	2014-05-29 21:40:32
<b>Suhu</b>	<b>Cahaya</b>	<b>Kelembapan</b>	<b>Waktu</b>

Showing 1 to 10 of 11 entries First Previous **1** 2 Next Last

**Gambar 4. 19. Tampilan tabel log aktivitas**

Dari hasil perancangan pada BAB III dan diimplementasikan pada BAB IV ada beberapa perancangan yang tidak dapat diterapkan atau diimplementasikan untuk membangun sistem perawatan. Untuk penjelasan implementasi yang dapat dan tidak dapat digunakan dalam sistem perawatan dapat dilihat pada Tabel 4. 1.

**Tabel 4. 1. Implementasi**

Nama Fitur	Implementasi	
	Berhasil	Tidak
Perangkat keras data pembanding		
- Sensor suhu (LM35)	Ya	
- Sensor cahaya (LDR)	Ya	
- Sensor kelembaban tanah (SEN0057)	Ya	
- SD card	Ya	
- Modul waktu (DS1307)		Tidak



Sistem perawatan			
• Perangkat <i>end device</i>			
	- Relay	Tidak jadi diimplementasikan	
	- Xbee adapter( <i>shield</i> )		
	- Xbee S-2 pro		
• Perangkat koordinator			
	- Sensor suhu (LM35)		Tidak
	- Sensor cahaya (LDR)	Ya	
	- Sensor kelembaban tanah (SEN0057)	Ya	
	- Modul waktu (DS1307)	Ya	
	- GSM/GPS/GPRS <i>shield</i>	Ya	
	- Xbee adapter ( <i>shield</i> )		Tidak
	- Xbee S-2 pro		Tidak
	- <i>Fuzzy logic</i>		Tidak
Sistem monitoring			
	• Aplikasi Android	Ya	
	• Aplikasi web	Ya	

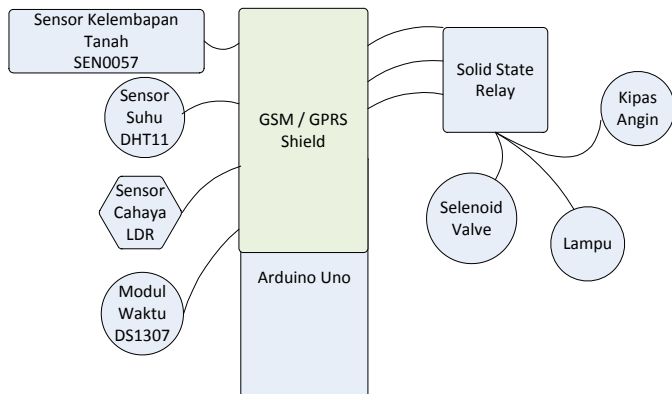
Pada perangkat keras data pembanding modul waktu tidak dapat digunakan, karena modul waktu (DS1307) membutuhkan 2 (dua) masukan analog pada pin 4 dan 5, sedangkan untuk menggunakan SD *card* adapter harus terhubung dengan analog pin 4 untuk bisa digunakan, dengan pertimbangan fungsionalitas maka modul waktu tidak digunakan untuk penentuan waktu pengambilan digunakan *delay* (jeda) untuk pengambilan data dan menuliskan data pada SD *card*.

Perangkat *end device* sudah berhasil diimplementasikan tetapi pada sistem perawatan perangkat *end device* tidak jadi digunakan karena pada seperangkat koordinator perangkat Xbee tidak digunakan. Keputusan untuk menghilangkan perangkat Xbee karena pada saat pengujian nilai keakuratan sensor menurun ketika perangkat koordinator dirangkai dengan rangkai Arduino Uno + GSM/GPS/GPRS *shield* + Xbee *shield* + Xbee + Sensor.

Pada perangkat koordinator penggunaan sensor suhu LM35 diganti dengan sensor DHT11, pengantian sensor ini karena pada saat pengujian pengambilan data nilai dari sensor LM35 – (minus) untuk itu dicari sensor dengan fungsi sama untuk pengukuran suhu.

Implementasi *fuzzy logic* tidak berhasil diimplementasikan, kendala terjadi ketika hasil rata-rata dari nilai sensor dimasukkan kedalam *fuzzy logic* nilai yang dihasilkan adalah nol. Sebelum implementasi pada sistem perawatan *fuzzy logic* dicoba pada sistem perawatan tetapi tanpa menggunakan perintah AT+Comand dalam uji coba ini nilai yang dihasilkan dari *fuzzy logic* sesuai dengan harapan, tetapi ketika *fuzzy logic* dijalankan bersama AT+Comand nilai dari sensor yang diolah dengan *fuzzy logic* menjadi 0 (nol).

Dari beberapa implementasi yang tidak dapat diimplementasikan maka sistem perangkat keras dari sistem perawatan dapat dilihat pada Gambar 4. 20. Pada perancangan sistem perawatan yang baru, menghilangkan perangkat *end device* dan perangkat koordinator. Sistem perawatan yang dibangun mirip dengan perangkat koordinator tetapi menghilangkan modul *Xbee shield + Xbee*.



**Gambar 4. 20. Sistem perawatan**

## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini akan dibahas mengenai hasil pengujian dan analisa kinerja yang telah dilakukan. Terdapat beberapa skenario ataupun test case yang akan dilakukan. Dengan beberapa kriteria dan parameter untuk mengukur keberhasilan pencapaian tujuan yang telah diuraikan sebelumnya.

#### **5.1 Uji coba perangkat keras data pembanding**

##### **5.1.1 Lingkungan Uji Coba**

Lingkungan uji coba menjelaskan perangkat keras serta perangkat lunak yang digunakan dalam menjalankan beberapa skenario uji coba. Perangkat lunak yang digunakan adalah:

IDE : Arduino IDE  
Sistem Operasi : Windows 7 Ultimate 64 Bit

Perangkat keras yang digunakan dalam pengujian aplikasi ini adalah:

Processor : Intel Core i5 2.4GHz. 1 GB GPU  
Memori : 6 GB DDR3

##### **5.1.2 Skenario Uji Coba**

Uji coba akan dilakukan dengan dua tipe pengujian dengan beberapa variasi parameter yaitu uji coba fungsionalitas dan uji coba performa. Uji coba dilakukan pada kebun yang sebelumnya telah disiram air sesuai kebutuhan tanaman. Ada beberapa jenis pendekatan yang akan dimodifikasi untuk dapat mengetahui hasil pengujian performa penyimpanan data dari alat yang telah dirangkai, daya tahan catu daya menggunakan baterai 9 V, dan keakuratan data yang diambil.

###### **5.1.2.1 Uji Coba Fungsionalitas**

Uji coba fungsionalitas dilakukan dengan cara menjalankan aplikasi yang telah dibuat. Uji coba fungsionalitas

meliputi semua proses yang berada pada diagram *use case* yang sudah dijelaskan pada BAB III, yaitu pengujian penyimpanan data selama 1 hari atau 24 jam.

### 5.1.2.2 Uji Coba Penyimpanan Data

Uji coba penyimpanan data ini dilakukan dengan menjalankan perangkat keras. Setelah implementasi perangkat lunak pada perangkat keras data pembanding, maka ketika perangkat keras data pembanding diaktifkan maka secara otomatis program akan berjalan dan perangkat keras akan menyimpan data yang telah didapat dari sensor dalam *file* berbentuk *.csv*. pada pengujian ini menggunakan catu daya baterai 9 V. Prosedur pengujian penyimpanan data dapat dilihat pada tabel 5.1.

**Tabel 5. 1. Prosedur uji coba penyimpanan data**

<b>Nama</b>	Uji coba penyampaian suara
<b>Tujuan uji coba</b>	Menguji penyimpanan data pada <i>SD card</i>
<b>Kondisi awal</b>	Perangkat keras menyimpan data setiap 1 menit.
<b>Masukan</b>	Suhu, cahaya, dan kelembaban tanah pada kebun tempat uji coba
<b>Keluaran</b>	Data yang tersimpan dalam file berbentuk <i>.csv</i>
<b>Hasil uji coba</b>	Berhasil

Data hasil penyimpanan pada 10 menit awal yang tersimpan di *SD card* bisa dilihat pada Tabel 5. 2.

**Tabel 5. 2. Hasil uji coba penyimpanan data**

Menit	Suhu (°C)	Cahaya (Lux)	Kelembaban (Volt)
1	27	118	4,99
2	27	115	4,99

<b>3</b>	27	113	4,99
<b>4</b>	27	116	4,99
<b>5</b>	26	115	4,99
<b>6</b>	27	115	4,99
<b>7</b>	27	116	4,99
<b>8</b>	27	116	4,99
<b>9</b>	27	116	4,99
<b>10</b>	27	115	4,99

### 5.1.2.3 Uji Coba Daya Tahan Baterai

Pengujian daya tahan baterai dilakakun bersamaan dengan menjalankan perangkat keras data pembanding, pengujian alat dimulai jam 15.30 WIB tanggal 05 juni 2014 daerah Surabaya.

**Tabel 5. 3. Prosedur uji coba daya tahan baterai**

<b>Nama</b>	Uji coba daya tahan baterai
<b>Tujuan uji coba</b>	Menguji daya tahan baterai
<b>Kondisi awal</b>	Perangkat keras menyimpan data setiap 1 menit.
<b>Masukan</b>	Perangkat keras yang terhubung dengan baterai
<b>Keluaran</b>	Daya tahan baterai
<b>Hasil uji coba</b>	Gagal

Pada uji coba menggunkan baterai gagal data terakhir didapat pada pukul 02.15 WIB pada tanggal 06 juni 2014, dari uji coba yang dilakukan bahwa penyimpanan file pada kartu memori/SD card memerlukan daya yang besar sehingga daya yang ada pada baterai tidak cukup untuk menjalankan perangkat keras data pembanding selama 24 jam atau lebih.

## 5.2 Uji Coba Perangkat Keras Sistem Perawatan

### 5.2.1 Lingkungan Uji Coba

menjelaskan perangkat keras serta perangkat lunak yang digunakan dalam menjalankan beberapa skenario uji coba. Perangkat lunak yang digunakan adalah:

IDE : Arduino IDE  
Sistem Operasi : Windows 7 Ultimate 64 Bit

Perangkat keras yang digunakan dalam pengujian aplikasi ini adalah:

Prosesor : Intel Core i5 2.4GHz, 1 GB GPU  
Memori : 6 GB DDR3

### 5.2.2 Skenario Uji Coba

Uji coba akan dilakukan dengan 2 tipe pengujian dengan beberapa parameter, yaitu uji coba fungsionalitas dan uji coba performa. Pengujian performa meliputi daya tahan baterai atau catu daya yang digunakan. Uji coba fungsionalitas meliputi setiap bagian perangkat keras yang dirangkai dengan Arduino Uno, selain pengujian tiap perangkat keras, pengujian dilakukan sistem secara keseluruhan.

#### 5.2.2.1 Uji Coba Fungsionalitas

Sebelum membahas tentang pengujian perangkat keras sistem perawatan, sistem perawatan secara keseluruhan sudah berjalan dengan baik. Sistem perawatan mampu untuk melakukan perintah perawatan dengan baik, tetapi yang perlu diketahui adalah sistem perawatan tidak bisa menjalankan perintah secara *multithread*. Sistem perawatan hanya bisa memenuhi 1 kondisi saja, ketika terjadi kondisi gelap dan kondisi tanah kering maka sistem perawatan akan menyiram tanah dulu setelah itu sistem perawatan akan memenuhi syarat tumbuh cahaya yang kurang begitu juga bila terjadi kondisi suhu lingkungan yang panas, kelembaban tanah kering dan intensitas cahaya kurang maka sistem perawatan hanya bisa memenuhi 1 kondisi jika kondisi 1

terpenuhi maka sistem akan memenuhi kondisi yang lain yang masih kurang.

Uji coba fungsionalitas dilakukan dengan cara menjalankan perangkat keras untuk mengetahui dan memastikan nilai yang dihasilkan oleh sensor sudah sesuai, selain itu uji coba fungsionalitas untuk mengetahui sistem yang dibangun telah berhasil dan berjalan dengan baik.

Pengujian fungsi kontrol setiap alat yang terpasang pada sistem, Pengujian dilakukan dengan menjalankan perangkat keras yang telah diprogram untuk menjalankan sistem perawatan. Sistem perawatan dijalankan selama 2 hari pada tanggal 19 – 21 juni 2014. Pada tanggal 19 – 20 juni 2014, sistem perawatan dipasang pada tempat uji coba tanpa menghidupkan perangkat kontrol, fungsi kontrol dijalankan pada tanggal 20 – 21 juni 2014 pada pukul 22:00 WIB dan berakhir pada pukul 22:00 WIB tanggal 21. Tempat uji coba dilakukan di daerah Keputih, Surabaya, media tanam tanah dengan komposisi 40 % pasir. Untuk uji coba kontrol pertama adalah kontrol pada perangkat lampu 1 dan lampu 2. Lampu 1 dan 2 akan menyala ketika kondisi cahaya  $> 23$  Lux, pada uji coba ini lampu menyala pada pukul 05:00 – 05:30 WIB, saat kondisi cahaya kurang terang di pagi hari. Perlu diketahui sistem berjalan mulai pukul 05:00 – 17:00 WIB lampu bisa menyala ketika pada rentang waktu tersebut kondisi kurang terang atau mendung yang mengakibatkan kondisi tempat uji coba gelap. Pada uji coba ini tidak menemui kondisi gelap yang disebabkan oleh cuaca mendung. Prosedur uji coba dapat dilihat pada Tabel 5. 4.

**Tabel 5. 4. Prosedur uji coba memenuhi intensitas cahaya disiang hari**

<b>Nama</b>	Uji coba kontrol lampu dengan relay.
<b>Tujuan Uji Coba</b>	Menguji lampu pada sistem perawatan ketika cahaya tidak memenuhi syarat tumbuh tanaman pada jam 05:00 – 17:00.
<b>Kondisi Awal</b>	Sistem perawatan berjalan membaca waktu, suhu, cahaya dan kelembaban tanah.

<b>Masukan</b>	Waktu(jam) dan intensitas cahaya.
<b>Keluaran</b>	Lampu menyala setiap 5 menit sek.ali
<b>Hasil Uji Coba</b>	Berhasil

Hasil uji coba dapat dilihat pada data log aktivitas dari sistem perawatan yang ditampilkan pada grafik log menggunakan perangkat Android maupun website ketika lampu menyala. Uji coba kontrol selanjutnya adalah uji coba kran otomatis, kran otomatis akan menyala pada saat pagi dan sore hari. Lama kran hidup adalah 20 menit, untuk pagi hari adalah pukul 05:30 – 05:50 sedangkan untuk sore hari adalah pukul 16:30 – 16:50 hasil data hasil penyiraman pada tanah dapat dilihat pada tabel uji coba performa, pada uji coba kran otomatis ini tidak ditemukan kondisi tanah menjadi kering atau tanah tidak memenuhi syarat tumbuh tanaman, jadi uji coba penyiraman menggunakan kran otomatis dilakukan pada pagi dan sore hari saja, saat sistem perawatan melakukan penyiraman rutin di pagi dan sore hari, tetapi pada uji coba ini mencoba membuat kondisi tanah seolah-olah menjadi kering dengan cara mencabut sensor kelembaban tanah. Prosedur uji coba sistem penyiraman otomatis dan penyiraman saat tanah kering dapat dilihat pada Tabel 5. 5 dan Tabel 5. 6.

**Tabel 5. 5. Prosedur pengujian sistem penyiraman otomatis**

<b>Nama</b>	Uji coba sistem penyiraman
<b>Tujuan Uji Coba</b>	Menguji sistem penyiraman otomatis pada pagi dan sore hari (05:30 – 05:50 dan 16:30 – 16:50) WIB
<b>Kondisi Awal</b>	Sistem perawatan berjalan membaca waktu, suhu, cahaya dan kelembaban tanah
<b>Masukan</b>	Membaca modul waktu DS1307
<b>Keluaran</b>	Kran otomatis menyala dan mengairi tanah dengan air selama 20 menit.
<b>Hasil Uji Coba</b>	Berhasil



**Tabel 5. 6. Prosedur pengujian penyiraman ketika tanah kering**

<b>Nama</b>	Uji coba penyiraman otomatis ketika tanah kering
<b>Tujuan Uji Coba</b>	Menguji sistem penyiraman otomatis ketika tanah kering
<b>Kondisi Awal</b>	Sistem perawatan berjalan membaca modul waktu, suhu, cahaya dan kelembaban tanah
<b>Masukan</b>	Membaca sensor kelembaban tanah dan waktu
<b>Keluaran</b>	Kran otomatis menyala dan mengairi tanah dengan air selama 5 menit.
<b>Hasil Uji Coba</b>	Berhasil

Pada pengujian kontrol yang terakhir adalah pengujian kipas angin, kipas angin menyala ketika suhu melebihi syarat tumbuh tanaman (33 °C). pada uji coba ini kondisi suhu yang melebihi syarat tumbuh tanaman adalah pukul 11:30 – 14:30 WIB. dari data yang diambil pada tanggal 19 – 21 juni 2014, suhu rata-rata pada rentang waktu tersebut adalah 34 °C. dari data yang didapat pada tanggal 19 – 20 juni 2014 maka untuk sistem perawatan untuk kontrol kipas angin dirubah menjadi 10 menit untuk setiap kali kipas dihidupkan dari awal 5 menit. prosedur uji coba kontrol kipas angin dapat dilihat pada Tabel 5. 7.

**Tabel 5. 7. Prosedur uji coba kontrol kipas angin**

<b>Nama</b>	Uji coba kontrol kipas angin
<b>Tujuan Uji Coba</b>	Menguji kontrol kipas angina
<b>Kondisi Awal</b>	Sistem perawatan berjalan membaca modul waktu, suhu, cahaya dan kelembaban tanah
<b>Masukan</b>	Membaca sensor suhu DHT11
<b>Keluaran</b>	Kipas angin menyala selama 10 menit.
<b>Hasil Uji Coba</b>	Berhasil

### 5.2.2.2.1 Uji Coba Pengiriman Data Kondisi Ruangan

Pengujian ini bersamaan ketika sistem kontrol berjalan, karena ada 1 variabel yang tidak memenuhi syarat tumbuh tanaman maka kontrol akan berjalan dan mengirimkan data kondisi ruangan selain itu pengiriman data kondisi ruangan setiap 15 menit sekali akan dikirimkan ke basis data sebagai data log aktivitas. Tabel prosedur pengiriman data tiap 15 menit sekali dan saat kondisi tidak memenuhi syarat tumbuh tanaman dapat dilihat pada Tabel 5. 8 dan Tabel 5. 9. Untuk pengiriman kondisi ruangan setiap 15 menit tidak selalu berhasil, data hasil pengiriman dapat dilihat pada uji coba performa.

**Tabel 5. 8. Prosedur uji coba pengiriman data setiap 15 menit**

<b>Nama</b>	Uji coba pengiriman data
<b>Tujuan Uji Coba</b>	Menguji fitur pengiriman data setiap 15 menit
<b>Kondisi Awal</b>	Sistem perawatan berjalan membaca modul waktu, suhu, cahaya dan kelembaban tanah.
<b>Masukan</b>	Sistem membaca modul waktu.
<b>Keluaran</b>	Sistem mengirim data ke basis data <i>server</i> sebagai log aktivitas setiap 15 menit
<b>Hasil uji coba</b>	Berhasil

Untuk pengujian pengiriman data kondisi ruangan ketika ada salah satu variabel yang tidak memenuhi syarat tumbuh tidak selalu terkirim, data hasil uji coba dapat dilihat pada tabel uji performa.

**Tabel 5. 9. Prosedur uji coba pengiriman data ketika ada syarat tumbuh tidak terpenuhi**

<b>Nama</b>	Uji Coba Pengiriman Data
<b>Tujuan Uji Coba</b>	Menguji fitur pengiriman data ketika terjadi kondisi syarat tumbuh tidak terpenuhi
<b>Kondisi Awal</b>	Sistem perawatan berjalan membaca modul waktu, suhu, cahaya dan kelembaban tanah.
<b>Masukan</b>	Kondisi syarat tumbuh tidak terpenuhi.
<b>Keluaran</b>	Sistem mengirim data ke basis data <i>server</i> .
<b>Hasil Uji Coba</b>	Berhasil

#### 5.2.2.2.2 Uji Coba Aplikasi Android dan Website

Uji coba ini dilakukan setelah proses pengiriman data log aktivitas dikirim ke basis data *server*. Pengujian dilakukan menggunakan perangkat Android yang telah terinstal aplikasi *smart garden* yang dibangun khusus untuk memantau log aktivitas selain aplikasi pada Android pengguna juga dapat melihat log aktivitas pada website.

**Tabel 5. 10. Prosedur pengujian menampilkan data dalam bentuk grafik garis**

<b>Nama</b>	Uji coba menampilkan data dalam grafik
<b>Tujuan Uji Coba</b>	Menguji fitur menampilkan data bentuk grafik garis
<b>Kondisi Awal</b>	Membaca data pada basis data <i>server</i>
<b>Masukan</b>	
<b>Keluaran</b>	Menampilkan data dalam bentuk grafik garis
<b>Hasil Uji Coba</b>	Berhasil

Pengujian yang dilakukan adalah menampilkan data log aktivitas pada perangkat Android dan website dalam bentuk grafik garis

data yang ada pada basis data *server*, khusus untuk aplikasi web pengguna dapat menentukan rentan waktu yang ingin ditampilkan dalam grafik log aktivitas. Prosedur pengujian dapat dilihat pada Tabel 5. 9.

#### 5.2.2.2 Uji Coba Performa Sistem Perawatan

Uji coba peforma bertujuan untuk mengetahui performa dari sistem perawatan dan pengaruh dari perlakuan yang diberikan terhadap kondisi tempat uji coba. Pengujian performa yang akan diuji adalah uji coba performa pengiriman data setiap 15 menit, pengaruh penyiraman terhadap tanah, pengaruh penggunaan kipas angin, uji coba pengiriman data ketika syarat tumbuh tanam tidak terpenuhi.

Selain uji performa, uji coba pengiriman data digunakan untuk menegetahui selisih waktu pengiriman dan waktu tersimpan data pada basis data *server*. Pada uji coba yang pertama adalah, uji coba sistem melakukan penyiraman pada tanah, pengujian untuk mengetahui apakah sistem bisa membuat kondisi tanah menjadi ideal sesuai dengan syarat tumbuh tanaman. Pengukuran dilakukan setelah sistem penyiraman otomatis selesai beroperasi, yaitu pada pagi dan sore hari, dengan adanya pengukuran ini dapat diketahui kenaikan nilai kelembaban tanah dari pagi sampai sore hari, perlu diingat selain pengaruh banyaknya volume air yang disiramkan tetapi komposisi tanah juga berpengaruh, data hasil pengukuran pengaruh penyiraman terhadap kelembaban tanah dapat dilihat pada Tabel 5. 12 dan Tabel 5.13 untuk rata-rata perubahan kelembaban tanah dapat dilihat pada Tabel 5. 14.

**Tabel 5. 11. Rentan kelembaban tanah**

Rentan nilai Kelambaban (Volt)	Katagori
3,53 – 5	Kering
1,58 – 3,53	Lembab
0,36 – 1,58	Basah(dalam air)

Tabel 5. 11 merupakan data pembandingan untuk nilai kelembaban tanah, katagori tanah yang cukup lembab adalah 1,58 – 3,53 Volt kondisi tersebut sesuai dengan syarat tumbuh tanaman kacang hijau yang membutuhkan keadaan tanah lembab. Pada ujicoba penyiraman pagi hari nilai kelembaban tanah 3,79 Volt, tanah dalam kategori kering. Setelah dilakukan penyiraman nilai dari kelembaban pada tanah turun menjadi 2,06 Volt, nilai tersebut sudah memenuhi syarat tumbuh tanaman kacang hijau yang membutuhkan kondisi tanah yang cukup lembab.

**Tabel 5. 12. Hasil penyiraman pagi hari**

Waktu	Lama Penyiraman	Kondisi (Nilai kelembaban) Volt	
		Sebelum	Sesudah
05:30 – 05:50	20 menit	3,79	2,06

**Tabel 5. 13. Hasil penyiraman sore hari**

Waktu	Lama Penyiraman	Kondisi (Nilai kelembaban) Volt	
		Sebelum	Sesudah
16:30 – 16:50	20 menit	2,03	1,91

Dari uji coba yang dilakukan penyiraman pada pagi dan sore hari perubahan nilai kelembaban tanah tidak terlalu signifikan terhadap kondisi kelembaban tanah, kelembaban tetap terjaga dari pagi sampai sore hari perubahan nilai kelembaban tanah dapat dilihat pada tabel Tabel 5. 14.

**Tabel 5. 14. Rata-rata nilai kelembaban tanah**

Waktu (06:00 – 17.00)	Nilai Rata-Rata Kelembaban Tanah (Volt)
07:00 – 08:00	2,08
08:00 – 09:00	2,11
09:00 – 10:00	1,99

10:00 – 11:00	1,95
11:00 - 12:00	1,94
12:00 – 13:00	1,95
13:00 – 14:00	1,95
14:00 – 15:00	2,00
15:00 – 16:00	2,01

Uji coba performa yang kedua adalah menjalankan fungsi sistem perawatan memenuhi lingkungan yang intensitas cahaya yang kurang, uji coba ini berfungsi untuk mengetahui apakah sistem perawatan dapat memenuhi kebutuhan cahaya yang kurang terhadap kondisi tempat uji coba serta untuk mengetahui apakah penggunaan lampu sudah dapat memenuhi syarat tumbuh tanaman, pengujian dilakukan dengan skenario pengambilan data selama 10 menit pada beberapa kondisi. Dari uji coba yang dilakukan didapat rentang nilai untuk menentukan kondisi keadaan terang sekali, terang, redup, gelap, gelap sekali.

Pengambilan data menggunakan sensor LDR tidak terlalu akurat karena sensor LDR tergantung pada arah cahaya yang memapari sensor LDR. Penggunaan sensor protodiode mungkin lebih baik untuk pengambilan data intensitas cahaya dengan akurasi tinggi.

**Tabel 5. 15. Rentang nilai intensitas cahaya**

Kondisi	Range Nilai (Lux)	Keterangan
Terang sekali	51150 – 4217	Terpapar sinar matahari
Terang	4217 – 38	Dalam ruangan uji coba cerah tidak berawan
Redup	38 – 23	dalam ruangan uji coba berawan
Gelap	23 – 2	Malam hari masih ada sinar bulan
Gelap sekali	2 - 0	Malam hari tidak ada sinar bulan

Tabel 5. 15 merupakan tabel rentang nilai, rentang nilai didapat dari pengamatan lapangan. Range nilai cahaya dapat digunakan untuk menentukan batas syarat tumbuh tanaman kacang hijau.

Uji coba perawatan kita kondisi intensitas cahaya tidak memenuhi syarat tumbuh perlau dilakukan karena bermanfaat untuk mengetahui seberapa signifikan sistem perawatan bisa membantu memenuhi kondisi syarat tumbuh yang kurang. Data hasil pengujian pada kondisi lingkungan gelap dapat dilihat pada Tabel 5. 16, pada hasil uji coba ketika lampu dinyalakan oleh sistem perawatan ketika kondisi ruangan gelap sudah cukup untuk memenuhi syarat tumbuh tanaman syarat minimum yang dibutuhkan oleh tanaman adalah kondisi redup dengan rentang nilai 38 – 23 Lux, sedangkan pada uji coba nilai rata-rata intensitas cahaya sebelum lampu menyala adalah 20 Lux (gelap) setelah lampu menyala nilai rata-rata intensitas cahaya pada tempat uji coba adalah 38 Lux (redup) . Untuk nilai rata-rata ditunjukkan pada baris terakhir.

**Tabel 5. 16. Hasil uji coba sistem perawatan ketika kondisi lingkungan gelap.**

<b>Kondisi Ruangan (Lux)</b>				<b>Keterangan</b>
<b>Gelap</b>		<b>Gelap Sekali</b>		
<b>Sebelum lampu menyala</b>	<b>Sesudah lampu menyala</b>	<b>Sebelum lampu menyala</b>	<b>Sesudah lampu menyala</b>	
17	39	1	145	Kondisi pada tempat uji coba pada rentang gelap, pada uji coba lampu dinyalakan dari hasil yang
20	38	1	146	
20	37	1	147	
20	41	1	148	
20	41	1	148	
20	41	1	148	
20	41	1	148	

20	38	1	148	didapat penggunaan lampu dapat menaikkan intensitas cahaya pada tempat uji coba
20	37	1	148	
20	36	1	148	
19	38	1	147	
19	38	1	146	
<b>20</b>	<b>38</b>	<b>1</b>	<b>259,1429</b>	

Semakin tinggi nilai cahaya maka semakin terang, kondisi pada tempat uji coba adalah cukup terang. Kondisi terang sudah cukup memenuhi syarat tumbuh tanaman kacang hijau, paparan sinar yang dibutuhkan tanaman kacang hijau adalah >10 jam.

Pada uji coba berikutnya adalah dilakukan pada tempat yang memiliki intensitas cahaya gelap sekali. Hasil uji coba dapat dilihat pada Tabel 5. 16 untuk pengujian kondisi gelap sekali dilakukan pada malam hari karena pada pengujian siang hari pada tempat uji coba tidak ditemukan kondisi gelap sekali. Dari hasil uji coba yang dilakukan ketika sistem menyalakan lampu, sistem yang dibuat mampu membuat kondisi lingkungan tempat uji coba memenuhi syarat tumbuh tanaman kacang hijau menjadi kondisi terang. Pengaruh intensitas cahaya lampu yang diberikan tergantung pada lampu yang digunakan pada pengujian ini digunakan lampu 18 W sebanyak 2 (dua) buah, besar kecilnya lampu yang digunakan disesuaikan dengan kebutuhan ruangan tempat kacang hijau ditanam.

Uji coba selanjutnya adalah pengujian menurunkan suhu lingkungan tanam, uji coba ini dilakukan untuk mengetahui seberapa besar sistem dapat menurunkan suhu pada lingkungan tanam menggunakan kipas angin. Uji coba ini dilakukan dengan menggunakan 2 pengukuran dalam uji coba ini, pengukuran secara manual dan menggunakan sensor. Dengan adanya 2 cara pengukuran yang berbeda diharapkan didapat hasil yang akurat dengan cara membandingkan hasil dari pengambilan data secara



manual dan menggunakan sensor. Uji coba dilakukan bersamaan ketika sistem kontrol kipas dinyalakan.

**Tabel 5. 17. Hasil pengukuran menurunkan suhu lingkungan menggunakan termometer**

Jarak (meter)	Suhu (°C)	Suhu awal (°C)
0 – 1	32	33
1 – 2	32,5	
2 – 3	32,5	
3 – 4	33	

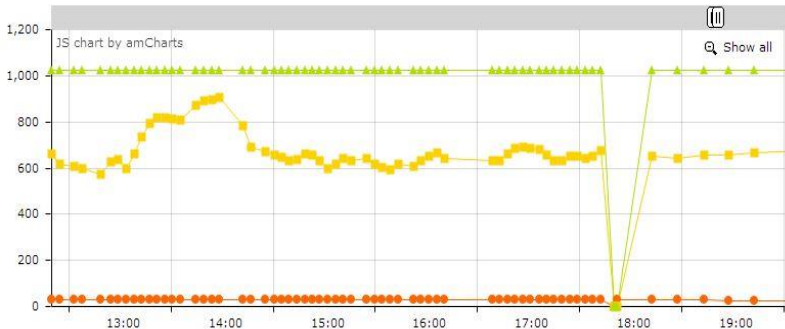
Pengukuran manual menggunakan termometer untuk mengukur pengaruh kipas angin dalam tempat uji coba, data hasil pengukuran dapat dilihat pada Tabel 5. 17. Hasil pengukuran menggunakan sensor dapat dilihat pada Tabel 5. 18, hasil dari pengukuran secara manual dan menggunakan sensor pada pengukuran menggunakan termometer terdapat selisih penurunan suhu 0,5 – 1 °C pada jarak 0 – 3 meter. Dari uji coba yang dilakukan penggunaan kipas angin tidak dapat menurunkan suhu secara signifikan, suhu ruangan tetap seperti seperti kondisi ketika kipas angin belum dinyalakan yaitu 33 °C meskipun terdapat sedikit perbedaan hasil pengukuran antara pengukuran manual dan menggunakan sensor, tetapi penggunaan kipas angin cukup membantu untuk memperlancar sirkulasi udara pada tempat uji coba.

**Tabel 5. 18. Tabel pengukuran suhu**

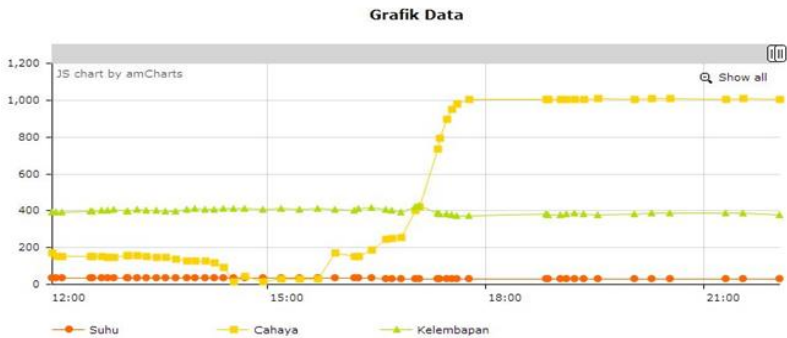
Jarak (meter)	Suhu (C)	Suhu awal (C)
0 – 1	33	33
1 – 2	33	
2 – 3	33	
3 – 4	33	

Setelah pengujian kontrol perawatan (menyiram, menerangi dan menurunkan suhu) selesai maka dilakukan pengamatan terhadap sistem secara keseluruhan terutama pengamatan terhadap pengiriman data sebelum pengamatan dilakukan sistem telah diuji coba pada 2 (dua) tempat yang berbeda yaitu pada Lab. NCC dan gang makam no:15 Surabaya. Data hasil uji coba ditampilkan dalam bentuk grafik seperti Gambar 5. 1 dan Gambar 5. 2, data yang ditampilkan pada grafik adalah data yang tersimpan setiap 15 menit dan ketika syarat tumbuh tidak terpenuhi. Dari pengamatan yang dilakukan didapatkan data yang tersimpan pada basis data *server* tidak tersimpan sesuai harapan, tidak semua data log aktivitas tersimpan pada basis data *server*. pada Gambar 5. 2 terlihat pada rentang waktu 18:00 terdapat data yang kosong, seharusnya ada data log aktivitas yang tersimpan setiap 15 menit sekali. Pengiriman data log aktivitas dari sistem perawatan tergantung pada sinyal yang ada, jika ketika pengiriman data dilakukan perangkat tidak dapat sinyal yang cukup untuk mengirim data ke basis data *server* data gagal tersimpan pada basis data *server*.

Grafik Data



Gambar 5. 1. Grafik log aktivitas 17 juni



**Gambar 5. 2. Grafik log aktivitas 21 juni**

Pengamatan yang dilakuakn pada 2 (dua) tempat yang berbeda ada beberapa data yang tidak terkirim sempurna dan, selisih waktu antara data satu dan data yang lain terpaut cukup jauh. Lebih dari 15 menit. Untuk perbedaan waktu antara *server* dan sistem perawatan adalah 2 menit, sistem perawatan lebih lambat 2 menit daripada waktu pada *local time* basis data *server*. Dapat dilihat kerapatan data yang ditampilkan oleh Gambar 5. 1 lebih teratur daripada Gambar 5. 2, ini disebabkan yang terkirim dan tersimpan pada basis data *server* lebih sedikit mengalami kegagalan.

Dari uji coba yang telah dilakukan didapatkan hasil sistem perawatan berjalan cukup baik, fungsi perangkat lampu dan penyiraman otomatis sudah cukup baik untuk membuat kondisi yang ideal untuk syarat tumbuh tanam kacang hijau. Untuk penggunaan kipas angin tidak dapat merubah kondisi suhu ruangan cukup signifikan tetapi kondisi pada ruangan uji coba sebetulnya masih dalam katagori suhu normal atau suhu ruangan masih dalam batas normal untuk syarat tumbuh tanam. Untuk penyimpanan data pada basis data *server* dari sisi sistem tidak ada kendala, tetapi kendala lebih pada perangkat penyedia layanan jaringan.

### 5.2.2.2.3 Uji Coba Performa Perangkat Android dan Aplikasi Web

Untuk uji coba performa perangkat Android dan aplikasi web, dilakukan 5 kali uji coba pada perangkat Android dengan 3 tempat berbeda. Pengujian dilakukan untuk mengetahui seberapa cepat perangkat Android menampilkan grafik data. 3 tempat uji coba kecepatan perangkat menampilkan grafik data adalah lab NCC, tempat uji coba sistem perawatan (keputih gang makam no 15) dan warung makan “Bonek”. Data hasil uji coba dapat di lihat pada Tabel 5. 19. Hasil uji coba lama menampilkan data pada aplikasi Android.

**Tabel 5. 19. Hasil uji coba lama menampilkan data pada aplikasi Android**

Nama tempat	Delay pada uji coba ke(detik)					Rata –rata (detik)
	1	2	3	4	5	
<b>Lab NCC</b>	5	6	3	5	4	4,6
<b>Gang makam</b>	8	8	2	4	6	5,6
<b>Warung Bonek</b>	4	4	7	3	5	4,6

Dari hasil uji coba yang dilakukan pada 3 tempat berbeda didapat rata-rata waktu yang dibutuhkan untuk menampilkan data grafik log aktivitas adalah 5 detik, ketersediaan layanan jaringan juga berpengaruh untuk menampilkan data dari basis data *server*.

## **BAB VI PENUTUP**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak, perangkat keras yang telah dirancang dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

### **6.1 Kesimpulan**

Pengamatan yang telah dilakukan dari tahap perancangan, kemudian implementasi, hingga uji coba perangkat lunak, dapat ditarik kesimpulan sebagai berikut:

1. Sistem perawatan yang yang dibuat dan dirangkai pada penelitian ini berfungsi sebagai alat pembantu monitoring keadaan *greenhouse* dan mempermudah dalam perawatan pada tanaman agar mendapat hasil yang maksimal.
2. Aplikasi pada perangkat Android untuk menampilkan grafik log aktivitas membutuhkan waktu rata-rata 4-6 detik untuk menampilkan data log aktivitas dan tergantung ketersediaan sinyal.
3. Sistem perawatan tidak bisa memenuhi kondisi syarat tumbuh secara bersamaan ketika lebih dari 1 syarat tumbuh yang tidak terpenuhi.
4. Penggunaan kipas angin dalam sistem perawatan tidak dapat menurunkan suhu lingkungan secara signifikan, tetapi cukup untuk memperlancar sirkulasi udara dan menjaga kesegaran udara.
5. Pengaruh penyiraman ketika kondisi tanah sudah cukup lembab tidak terlalu signifikan, karena tanah memiliki daya serap maksimal untuk menyerap air.
6. Penggunaan perangkat lampu sebagai pengganti cahaya matahari cukup efektif ketika ketika kondisi lingkungan

redup hingga gelap sekali. Besar kecilnya cahaya pada lampu tergantung pada lampu.

## 6.2 **Saran**

Beberapa saran yang diberikan untuk pengembangan sistem lebih lanjut dimasa yang akan datang jika dilihat dari hasil perancangan, implementasi, dan uji coba yang telah dilakukan antara lain:

1. Untuk uji coba penelitian selanjutnya diharapkan penggunaan Xbee bisa diterapkan karena adanya komunikasi data antar Arduino Uno dapat mempermudah dalam pembuatan sistem yang cukup besar yang menggunakan perangkat Arduino Uno.
2. Untuk uji coba sebaiknya dilakukan pada *greenhouse* dan terdapat tanaman sebagai bahan uji coba.
3. Ketersediaan layanan jaringan juga perlu diperhatikan jika ingin mengembangkan sistem cerdas yang selalu terhubung dengan jaringan.
4. Untuk lingkungan yang membutuhkan tingkat performa tinggi, dianjurkan untuk menggunakan perangkat mikrokontroler yang lebih baik semisal Raspberry Pi karena memiliki performa yang lebih baik daripada Arduino Uno.
5. Sebelum membangun perangkat keras perlu dilakukan perhitungan tentang kebutuhan daya perangkat keras tersebut secara datai dan teliti.

## DAFTAR PUSTAKA

- [1] Dinas Pertanian dan Tanaman Pangan Propinsi Jawa Barat, “Komonditas Ungulan (Kacang Hijau),” Dinas Pertanian dan Tanaman Pangan Propinsi Jawa Barat, [Daring]. Available: <http://diperta.jabarprov.go.id/index.php/subMenu/1694>. [Diakses 1 Maret 2014].
- [2] B. A. Handoko, Susanto dan N. Wakhidah, “<http://www.academia.edu/>,” [Daring]. Available: [http://www.academia.edu/3463411/SISTEM\\_PENGALIRAN\\_OTOMATIS](http://www.academia.edu/3463411/SISTEM_PENGALIRAN_OTOMATIS). [Diakses 5 Januari 2014].
- [3] F. Hahn, “Fuzzy controller decreases tomato cracking in greenhouses,” *Computers and Electronics in Agriculture*, vol. 77, no. 1, pp. 21-27, 2011.
- [4] S. D. Kaehler, “FUZZY LOGIC - AN INTRODUCTION,” [Daring]. Available: [http://www.seattlerobotics.org/encoder/mar98/fuz/fl\\_part1.htm](http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part1.htm). [Diakses 5 Maret 2014].
- [5] L. Welling dan L. Thomson, *PHP and MySQL Web Development*, Indianapolis: Sams Publishing, 2001.
- [6] D. Android.com, Februari 2013. [Daring]. Available: <http://developer.android.com/about/index.html>. [Diakses 7 Januari 2014].
- [7] Arduino.cc, Maret 2013. [Daring]. Available: <http://www.arduino.cc/>. [Diakses 7 Januari 2014].
- [8] Baskara, “Sensor Warna TCS3200 and TCS3210,” 1 Mei 2013. [Daring]. Available: <http://baskarapunya.blogspot.com/2013/05/sensor-warna-tcs3200-and-tcs3210.html>. [Diakses 1 Maret 2014].
- [9] Arduino.cc, “Arduino GSM Shield,” CREATIVE COMMONS, 2013. [Daring]. Available: <http://arduino.cc/en/Main/ArduinoGSMShield#.UxbXL8Z7IcI>. [Diakses 7 Januari 2014].

- [10] R. C. Gadri, A. Chavan, R. Sonawane dan S. Kamble, "Land Vehicle Tracking Application on Android Platform," *International Journal of Engineering Research and Applications (IJERA)*, pp. 1978-1982, 2012.
- [11] lionjogja, "20m.com," 2006. [Daring]. Available: <http://lionjogja.20m.com/relay.html>. [Diakses 7 Januari 2014].
- [12] A. Kadir, *Panduan Praktis Memplajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino*, Surabaya: ANDI, 2012.
- [13] Arduino.cc, "Arduino Uno," Maret 2013. [Daring]. Available: <http://www.arduino.cc/>. [Diakses 5 Maret 2014].



## BIODATA PENULIS



**Grezio Arifiyan Primajaya**, bisa dipanggil Grezio, lahir di kota Malang, 26 Agustus 1992. Penulis adalah anak ke-2 dari 2 bersaudara dibesarkan di kab. Malang, Jawa Timur. Penulis menempuh pendidikan SDN Gedogwetan III (1996 - 2004), SMP 02 Turen (2004 - 2007), SMA Negeri Kepanjen (2007 - 2010). Pada tahun 2010, penulis mengikuti SNMPTN dan diterima strata satu jurusan Teknik Informatika Fakultas

Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya angkatan 2010 yang terdaftar dengan NRP 5110100196. Selama menempuh kuliah, penulis juga aktif sebagai anggota departemen dalam negeri di Himpunan Mahasiswa Teknik Computer (HMTC) C-1A. Penulis juga aktif sebagai administrator lab NCC selama masa perkuliahannya. Pada beberapa acara kampus, penulis juga beberapa kali aktif menjadi panitia, baik sebagai anggota maupun koordinator.

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN

### 1. *Fuzzy Logic Test* arduino sketch

```
#include <Fuzzy.h>
#include <FuzzyComposition.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzyOutput.h>
#include <FuzzyRule.h>
#include <FuzzyRuleAntecedent.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzySet.h>

// Instanciando um objeto da biblioteca
Fuzzy* fuzzy = new Fuzzy();

//range data syarat tumbuh suhu
FuzzySet* dingin = new FuzzySet(15, 20, 20, 25);
FuzzySet* normal = new FuzzySet(20, 25, 25, 30);
FuzzySet* panas = new FuzzySet(25, 30, 30, 35);

//input fuzzy dari sampling
//range data sampling
FuzzySet* dinginOut = new FuzzySet(18, 23, 23, 30);
FuzzySet* normalOut = new FuzzySet(23, 28, 28, 35);
FuzzySet* panasOut = new FuzzySet(28, 35, 35, 35);

//range data syarat tumbuh cahaya
FuzzySet* terang = new FuzzySet(0, 275, 275, 300); //Terang
FuzzySet* redup = new FuzzySet(275, 300, 300, 700); //Redup
FuzzySet* gelap = new FuzzySet(700, 1024, 1024, 1024); //Gelap
//input fuzzy dari sampling
//range data sampling cahaya
FuzzySet* terangOut = new FuzzySet(0, 300, 300, 326); //Terang
FuzzySet* redupOut = new FuzzySet(300, 326, 326, 700); //Redup
FuzzySet* gelapOut = new FuzzySet(694, 700, 700, 1024); //Gelap
//(butuh nya fuzzy untuk menghendle 1 input dan 1 output)

//range data syarat tumbuh Kelembapan tanah
FuzzySet* basah = new FuzzySet(0, 275, 275, 300); //Terang
```

```

FuzzySet* sedang = new FuzzySet(275, 300, 300, 700); //Redup
FuzzySet* kering = new FuzzySet(700, 1024, 1024,
1024); //Gelap
//input fuzzy dari sampling
//range data sampling kelembapan tanah

FuzzySet* basahOut = new FuzzySet(0, 300, 300, 326); //Terang
FuzzySet* sedangOut = new FuzzySet(300, 326, 326,
700); //Redup
FuzzySet* keringOut = new FuzzySet(694, 700, 700,
1024); //Gelap
//(butuh nya fuzzy untuk menghendle 1 input dan 1 output)

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

  //Fuzzy input suhu
  FuzzyInput* suhu = new FuzzyInput(1);
  suhu->addFuzzySet(dingin);
  suhu->addFuzzySet(normal);
  suhu->addFuzzySet(panas);

  fuzzy->addFuzzyInput(suhu);

  //fuzzy out suhu
  FuzzyOutput* suhuOut = new FuzzyOutput(1);
  suhuOut->addFuzzySet(dinginOut);
  suhuOut->addFuzzySet(normalOut);
  suhuOut->addFuzzySet(panasOut);

  fuzzy->addFuzzyOutput(suhuOut);

  //Fuzzy input cahaya
  FuzzyInput* cahaya = new FuzzyInput(2);
  cahaya->addFuzzySet(terang);
  cahaya->addFuzzySet(redup);
  cahaya->addFuzzySet(gelap);

  fuzzy->addFuzzyInput(cahaya);

  //fuzzy out cahaya
  FuzzyOutput* cahayaOut = new FuzzyOutput(2);
  cahayaOut->addFuzzySet(terangOut);
  cahayaOut->addFuzzySet(redupOut);
  cahayaOut->addFuzzySet(gelapOut);

  fuzzy->addFuzzyOutput(cahayaOut);

```

```

//Fuzzy input kelembapan tanah
FuzzyInput* tanah = new FuzzyInput(3);
tanah->addFuzzySet(basah);
tanah->addFuzzySet(sedang);
tanah->addFuzzySet(kering);

fuzzy->addFuzzyInput(tanah);

//fuzzy out kelembapan tanah
FuzzyOutput* tanahOut = new FuzzyOutput(3);
tanahOut->addFuzzySet(basahOut);
tanahOut->addFuzzySet(sedangOut);
tanahOut->addFuzzySet(keringOut);

fuzzy->addFuzzyOutput(tanahOut);

//Fuzzy Rule suhu
//rule 1 suhu
FuzzyRuleAntecedent* IfsuhuDingin = new
FuzzyRuleAntecedent();
IfsuhuDingin->joinSingle(dingin);
FuzzyRuleConsequent* ThenDinginOut = new
FuzzyRuleConsequent();
ThenDinginOut->addOutput(dinginOut);

FuzzyRule* fuzzyRule1 = new FuzzyRule(1, IfsuhuDingin,
ThenDinginOut);
fuzzy->addFuzzyRule(fuzzyRule1);

//rule2 suhu
FuzzyRuleAntecedent* IfsuhuNormal = new
FuzzyRuleAntecedent();
IfsuhuNormal->joinSingle(normal);
FuzzyRuleConsequent* ThenNormalOut = new
FuzzyRuleConsequent();
ThenNormalOut->addOutput(normalOut);

FuzzyRule* fuzzyRule2 = new FuzzyRule(2, IfsuhuNormal,
ThenNormalOut);
fuzzy->addFuzzyRule(fuzzyRule2);

//rule3 suhu
FuzzyRuleAntecedent* IfsuhuPanas = new
FuzzyRuleAntecedent();
IfsuhuPanas->joinSingle(panas);
FuzzyRuleConsequent* ThenPanasOut = new
FuzzyRuleConsequent();
ThenPanasOut->addOutput(panasOut);

```

```

FuzzyRule* fuzzyRule3 = new FuzzyRule(3, IfsuhuPanas,
ThenPanasOut);
fuzzy->addFuzzyRule(fuzzyRule3);

//Fuzzy Rule cahaya
//rule 4 cahaya
FuzzyRuleAntecedent* IfcahayaTerang = new
FuzzyRuleAntecedent();
IfcahayaTerang->joinSingle(terang);
FuzzyRuleConsequent* ThenTerangOut = new
FuzzyRuleConsequent();
ThenTerangOut->addOutput(terangOut);

FuzzyRule* fuzzyRule4 = new FuzzyRule(4, IfcahayaTerang,
ThenTerangOut);
fuzzy->addFuzzyRule(fuzzyRule4);

//rule 5 cahaya
FuzzyRuleAntecedent* IfcahayaRedup = new
FuzzyRuleAntecedent();
IfcahayaRedup->joinSingle(redup);
FuzzyRuleConsequent* ThenRedupOut = new
FuzzyRuleConsequent();
ThenRedupOut->addOutput(redupOut);

FuzzyRule* fuzzyRule5 = new FuzzyRule(5, IfcahayaRedup,
ThenRedupOut);
fuzzy->addFuzzyRule(fuzzyRule5);

//rule 6 cahaya
FuzzyRuleAntecedent* IfcahayaGelap = new
FuzzyRuleAntecedent();
IfcahayaGelap->joinSingle(gelap);
FuzzyRuleConsequent* ThenGelapOut = new
FuzzyRuleConsequent();
ThenGelapOut->addOutput(gelapOut);

FuzzyRule* fuzzyRule6 = new FuzzyRule(6, IfcahayaGelap,
ThenGelapOut);
fuzzy->addFuzzyRule(fuzzyRule6);

//Fuzzy Rule kelembapan tanah
//rule 5 kelembapan
FuzzyRuleAntecedent* IftanahBasah = new
FuzzyRuleAntecedent();
IftanahBasah->joinSingle(basah);
FuzzyRuleConsequent* ThenBasahOut = new
FuzzyRuleConsequent();

```

```

ThenBasahOut->addOutput(basahOut);

FuzzyRule* fuzzyRule7 = new FuzzyRule(7, IftanahBasah,
ThenBasahOut);
fuzzy->addFuzzyRule(fuzzyRule7);

//rule 8 cahaya
FuzzyRuleAntecedent* IftanahSedang = new
FuzzyRuleAntecedent();
IftanahSedang->joinSingle(sedang);
FuzzyRuleConsequent* ThenSedangOut = new
FuzzyRuleConsequent();
ThenSedangOut->addOutput(sedangOut);

FuzzyRule* fuzzyRule8 = new FuzzyRule(8, IftanahSedang,
ThenSedangOut);
fuzzy->addFuzzyRule(fuzzyRule8);

//rule 9 cahaya
FuzzyRuleAntecedent* IftanahKering = new
FuzzyRuleAntecedent();
IftanahKering->joinSingle(kering);
FuzzyRuleConsequent* ThenKeringOut = new
FuzzyRuleConsequent();
ThenKeringOut->addOutput(keringOut);

FuzzyRule* fuzzyRule9 = new FuzzyRule(9, IftanahKering,
ThenKeringOut);
fuzzy->addFuzzyRule(fuzzyRule9);
}

void loop(){
int val = analogRead(0);
int vil = analogRead(1);
int vol = analogRead(2);
int dat;
dat=(500 *val) /1024;

fuzzy->setInput(1, dat);
fuzzy->fuzzify();
float outputSuhu = fuzzy->defuzzify(1);

fuzzy->setInput(2, vil);
fuzzy->fuzzify();
float outputCahaya = fuzzy->defuzzify(2);

fuzzy->setInput(3, vol);
fuzzy->fuzzify();

```

```

float outputTanah = fuzzy->defuzzify(3);

Serial.print("Suhu Fuzzy: ");
Serial.print(outputSuhu);
Serial.print("  Suhu: ");
Serial.print(dat);
Serial.print("  --||--");
Serial.print("  cahaya Fuzzy: ");
Serial.print(outputCahaya);
Serial.print("  cahaya: ");
Serial.print(vil);
Serial.print("  --||--");
Serial.print("  tanah Fuzzy: ");
Serial.print(outputTanah);
Serial.print("  tanah: ");
Serial.print(vol);
Serial.println();

  delay(1000); //just here to slow down the output for
easier reading
}

```

## 2. Relay Test Sketch

```

void setup(){
  Serial.begin(9600);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
}

void loop(){
  digitalWrite(8, HIGH);
  Serial.println("Pin 8 mati");
  delay(2000);
  digitalWrite(8, LOW);
  Serial.println("Pin 8 hidup");
  delay(2000);
  digitalWrite(9, HIGH);
  Serial.println("Pin 9 mati");
  delay(2000);
  digitalWrite(9, LOW);
  Serial.println("Pin 9 hidup");
}

```



```

delay(2000);
digitalWrite(10, HIGH);
Serial.println("Pin 10 mati");
delay(2000);
digitalWrite(10, LOW);
Serial.println("Pin 10 hidup");
delay(2000);
digitalWrite(11, HIGH);
Serial.println("Pin 11 hidup");
delay(2000);
    digitalWrite(11, LOW);
Serial.println("Pin 11 mati");
delay(2000);
}

```

### 3. Xbee end device arduino sketch

```

byte byteRead;
void setup(){
    Serial.begin(9600);
}
void relay(){
    if(Serial.available()){
        byteRead = Serial.read();
        if(byteRead == 88){
            Serial.println("Suhu Panas - Kipas");
            delay(50000);
        }
        if(byteRead == 120){
            Serial.println("Lampu 1 Hidup");
            delay(50000);
        }
        if(byteRead == 74){
            Serial.println("Lampu 2 + 1 Hidup");
            delay(50000);
        }
        if(byteRead == 90){
            Serial.println("Kran Hidup");
            delay(50000);
        }
    }
}
}

```

```
void loop() {
  relay();}
```

#### 4. AT+Comand kirim data ke server

```
void start_GSM() {
  //Configuracion GPRS Claro Argentina
  Serial.println("AT");
  delay(2000);
  Serial.println("AT+CREG?");
  delay(2000);
  Serial.println("AT+SAPBR=3,1,\"APN\", \"telkomsel\");
  delay(2000);
  Serial.println("AT+SAPBR=3,1,\"USER\", \"\");
  delay(2000);
  Serial.println("AT+SAPBR=3,1,\"PWD\", \"\");
  delay(2000);
  Serial.println("AT+SAPBR=3,1,\"Contype\", \"GPRS\");
  delay(2000);
  Serial.println("AT+SAPBR=1,1");
  delay(10000);
  Serial.println("AT+HTTPIPINIT");
  delay(2000);
  Serial.println("AT+HTTTPARA=\"CID\",1");
  delay(2000);
}

void send_GPRS() {
  Serial.print("AT+HTTTPARA=\"URL\", \"tugasakhir196.esy.
  es/insert.php?suhu=");
  Serial.print(suhu());
  Serial.print("&cahaya=");
  Serial.print(ldr());
  Serial.print("&kelembapan=");
  Serial.print(tanah());
  Serial.println("\");
  delay(2000);
  Serial.println("AT+HTTTPACTION=0"); //now GET
  action
  delay(2000);
}
```

## 5. DTH11 Sensor tester sketch

```
// Example testing sketch for various DHT
humidity/temperature sensors
// Written by ladyada, public domain

#include "DHT.h"

#define DHTPIN 2      // what pin we're connected to

// Uncomment whatever type you're using!
//#define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// Connect pin 2 of the sensor to whatever your DHTPIN
is
// Connect pin 4 (on the right) of the sensor to
GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1
(power) of the sensor

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250
  milliseconds!
  // Sensor readings may also be up to 2 seconds 'old'
  (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
```

```

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try
again).
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}

// Compute heat index
// Must send in temp in Fahrenheit!
float hi = dht.computeHeatIndex(f, h);

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t");
Serial.print("Heat index: ");
Serial.print(hi);
Serial.println(" *F");
}

```

## 6. DS1307 Sensor sketch

```

// Date and time functions using a DS1307 RTC
connected via I2C and Wire lib

#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 rtc;

void setup () {
  Serial.begin(57600);
#ifdef AVR
  Wire.begin();
#else
  Wire1.begin(); // Shield I2C pins connect to alt I2C
bus on Arduino Due
#endif
}

```