



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**RANCANG BANGUN PROTOTIPE RUMAH KACA PINTAR  
BERBASIS IoT (INTERNET OF THINGS) UNTUK BUDIDAYA  
TANAMAN TOMAT**

Muhammad Irfan Abriyantoro  
0711164000151

Dosen Pembimbing  
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Surabaya 2020





TUGAS AKHIR - EE 184801

**RANCANG BANGUN PROTOTIPE RUMAH KACA PINTAR  
BERBASIS IoT (INTERNET OF THINGS) UNTUK BUDIDAYA  
TANAMAN TOMAT**

Muhammad Irfan Abriyantoro  
07111640000151

Dosen Pembimbing  
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Surabaya 2020



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi keseluruhan Tugas akhir saya dengan judul “**RANCANG BANGUN PROTOTIPE RUMAH KACA PINTAR BERBASIS IoT (INTERNET OF THINGS) UNTUK BUDIDAYA TANAMAN TOMAT**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Jakarta, Juni 2020

Muhammad Irfan Abriyantoro  
0711164000151

*-Halaman ini sengaja dikosongkan-*

# **RANCANG BANGUN PROTOTIPE RUMAH KACA PINTAR BERBASIS IoT (INTERNET OF THINGS) UNTUK BUDIDAYA TANAMAN TOMAT**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Telekomunikasi Multimedia  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**



**Dosen Pembimbing**

**Sri Rahayu, S.T., M.Kom.  
NIP.196802281997022001**

**SURABAYA  
JUNI, 2020**

*-Halaman ini sengaja dikosongkan-*



# **RANCANG BANGUN PROTOTIPE RUMAH KACA PINTAR BERBASIS IoT (INTERNET OF THINGS) UNTUK BUDIDAYA TANAMAN TOMAT**

Nama Mahasiswa : Muhammad Irfan Abriyantoro  
Dosen Pembimbing : Sri Rahayu S.T., M.Kom.

## **Abstrak**

Di era modern ini, makin banyak masalah yang dihadapi oleh masyarakat yang tinggal di perkotaan(urban). Aktivitas sehari-hari yang sangat padat dan serba cepat, namun tidak diimbangi dengan kemampuan mobilisasi yang memadai sering menjadi pemicu munculnya gangguan kesehatan akibat dari tekanan psikis yang tinggi dan buruknya kualitas udara yang penuh polutan. Tak heran banyak bermunculan komunitas yang mengkampanyekan hal-hal positif seperti gemar berolahraga, cinta lingkungan, asupan makanan sehat, dsb. Bahkan banyak mengkonsumsi Vitamin-C juga telah menjadi gaya hidup. Buah tomat sering menjadi sasaran konsumsi masyarakat urban untuk mendapatkan Vitamin-C alami yang murah. Selain Vitamin-C, tomat juga kaya betakaroten dan likopen sebagai antioksidan yang menjawab kebutuhan khas masyarakat urban.

Dalam Tugas Akhir ini, akan dibuat rancangan proto budidaya tanaman tomat yang dapat diimplementasikan oleh masyarakat kota yang terbatas luas lahan dan waktu pengelolaannya. Rancangan ini menggabungkan konsep *greenhouse* dan teknologi berbasis IoT, sehingga mensyaratkan adanya sejumlah *embedded-sensor*, jaringan internet, perangkat lunak dan divais elektronik lainnya.

Dengan teknologi *monitoring* tomat berbasis IoT ini, pemantauan dan perawatan kondisi tanaman dapat dilakukan dari jarak jauh. Tanaman tomat juga bisa dipanen lebih dari sekali dan hasil panen segar karena pengaturan lingkungan yang sesuai untuk pertumbuhan tomat. Hasil pengamatan pada parameter kelayakan jaringan juga menunjukkan bahwa teknologi IoT untuk pertanian khususnya tomat sangat layak untuk diterapkan.

**Kata Kunci:** *Internet of Things*, tomat, *cloud*, *sensor*, mikrokontroler, data.

*-Halaman ini sengaja dikosongkan-*

# **IoT-BASED SMART GREENHOUSE PROTOTYPE DESIGN FOR TOMATO CULTIVATION**

Student Name : Muhammad Irfan Abriyantoro  
Supervisor : Sri Rahayu S.T., M.Kom.

## **Abstract**

In this modern era, more and more problems are faced by people who live in urban areas (urban). Daily activities that are very dense and fast paced, but are not matched by adequate mobilization capabilities often trigger health problems due to high psychological pressure and poor air quality that is full of pollutants. No wonder so many communities have emerged that campaign for positive things like sports, love for the environment, intake of healthy food, etc. Even consuming lots of Vitamin-C has also become a lifestyle. Tomatoes are often the target of consumption in urban communities to get cheap natural Vitamin-C. In addition to Vitamin-C, tomatoes are also rich in beta-carotene and lycopene as antioxidants that answer the typical needs of urban communities.

In this Final Project, a prototype of tomato cultivation will be made that can be implemented by city communities with limited land area and management time. This design combines the concept of greenhouse and IoT-based technology, thus requiring a number of embedded sensors, internet networks, software and other electronic devices.

With this IoT-based tomato monitoring technology, monitoring and maintenance of plant conditions can be done remotely. Tomato plants can also be harvested more than once and the yield is fresh because of the appropriate environmental settings for tomato growth. Observations on network feasibility parameters also show that IoT technology for agriculture, especially tomatoes, is very feasible.

**Keywords:** *Internet of Things, tomato, cloud, sensor, microcontroller, data.*

*-Halaman ini sengaja dikosongkan-*

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan tugas akhir dengan judul “**Rancang Bangun Rumah Kaca Pintar Berbasis IoT (*Internet of Things*) Untuk Budidaya Tanaman Tomat**”. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak dari masa perkuliahan sampai penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini, oleh karena itu saya mengucapkan terima kasih kepada:

1. Ibu dan ayah penulis atas doa dan cinta yang tak henti pada penulis dalam keadaan apapun. Semoga Allah SWT senantiasa melindungi dan memberi mereka tempat terbaik kelak di surgaNya.
2. Ibu Sri Rahayu, S.T., M.Kom selaku dosen pembimbing yang telah memberikan arahan, bimbingan dan perhatiannya selama proses penyelesaian tugas akhir ini.
3. Seluruh dosen dan karyawan Departemen Teknik Elektro ITS yang telah memberikan banyak ilmu dan menciptakan suasana belajar yang luar biasa.
4. Teman-teman SMA saya yang juga berkuliah di Surabaya : Ibrahim, Risky, Rifqy, Kalief, Amyra, Sasqia, TM yang senantiasa mendukung perkuliahan saya selama 4 tahun ini.
5. Teman-teman seperjuangan e56 yang telah menemani dan memberikan dukungan selama masa kuliah sampai penyusunan tugas akhir ini.

Penulis telah berusaha maksimal dalam penyusunan tugas akhir ini. Namun tetap besar harapan penulis untuk menerima saran dan kritik untuk perbaikan dan pengembangan tugas akhir ini. Semoga tugas akhir ini dapat memberikat manfaat yang luas.

Jakarta, Juni 2020

Muhammad Irfan Abriyantoro

# DAFTAR ISI

PERNYATAAN KEASLIAN.....	iv
TUGAS AKHIR.....	iv
KATA PENGANTAR.....	xii
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xviii
DAFTAR TABEL.....	xx
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
<b>1.1 Latar Belakang</b> .....	<b>1</b>
<b>1.2 Rumusan Masalah</b> .....	<b>2</b>
<b>1.3 Batasan Masalah</b> .....	<b>2</b>
<b>1.4 Tujuan</b> .....	<b>3</b>
<b>1.5 Metodologi</b> .....	<b>3</b>
<b>1.6 Sistematika Penulisan</b> .....	<b>4</b>
<b>1.7 Relevansi</b> .....	<b>5</b>
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>7</b>
<b>2.1 Sistem Monitoring</b> .....	<b>7</b>
<b>2.2 Tanaman Tomat</b> .....	<b>7</b>
<b>2.2.1 Proses Tumbuh Tanaman Tomat</b> .....	<b>8</b>
<b>2.2.2 Merawat Tanaman Tomat</b> .....	<b>8</b>
<b>2.3 Jaringan Nirkabel</b> .....	<b>8</b>
<b>2.3.1 <i>Wireless Sensor Networks</i> (WSN)</b> .....	<b>9</b>
<b>2.3.2 Komunikasi dalam WSN</b> .....	<b>10</b>

<b>2.4</b>	<b>Internet of Things (IoT)</b> .....	11
2.4.1	Cara Kerja IoT .....	11
2.4.2	Arsitektur IoT .....	12
<b>2.5</b>	<b>Cloud</b> .....	13
2.5.1	Jenis <i>Cloud</i> Berdasarkan Pengguna .....	13
2.5.2	Jenis <i>Cloud</i> Berdasarkan Model Layanan .....	14
a.	Infrastructure-as-a-Service (IaaS) .....	14
b.	Platform-as-a-Service (PaaS) .....	14
c.	Software-as-a-Service (SaaS) .....	14
<b>2.6</b>	<b>Mikrokontroler</b> .....	15
<b>2.7</b>	<b>Quality of Service (QoS)</b> .....	17
2.7.1	<i>Throughput</i> .....	17
2.7.2	<i>Delay</i> .....	17
2.7.3	<i>Jitter</i> .....	18
2.7.4	<i>Packet Loss</i> .....	19
<b>2.8</b>	<b>Wireshark</b> .....	19
<b>2.9</b>	<b>Arduino IDE</b> .....	20
<b>2.10</b>	<b>Blynk</b> .....	22
<b>BAB III METODOLOGI PENELITIAN</b> .....		24
<b>3.1</b>	<b>Diagram Alir Penelitian</b> .....	24
<b>3.2</b>	<b>Identifikasi Permasalahan dan Solusi</b> .....	25
<b>3.3</b>	<b>Desain Sistem Budidaya Tomat Berbasis IoT</b> .....	26

<b>3.4</b>	<b>Pembuatan Sistem <i>Monitoring</i> IoT Tomat (SiMonTa)</b>	<b>27</b>
3.4.1	Perakitan Perangkat Keras	27
3.4.1.1	Perakitan Perangkat Keras	27
3.4.1.3	Realisasi Perangkat Keras	33
3.4.2	Pembuatan Perangkat Lunak	34
3.4.2.1	Penggunaan Bahasa Pemrograman	34
3.4.2.2	Pemrograman Mikrokontroler	35
3.4.2.3	Pemrograman Modul IoT	35
3.4.2.4	Realisasi Tampilan Antarmuka	36
3.4.3	Integrasi Sistem	36
3.4.4	Pengujian Sistem	37
3.4.4.1	Pengujian Fungsi Sensor	38
3.4.4.2	Pengujian Fungsi Aktuator	38
3.4.4.3	Pengujian Fungsi IoT	38
<b>3.5</b>	<b>Desain Lahan Tanaman Tomat</b>	<b>39</b>
3.5.1	Pembuatan Lahan Tomat Hidroponik	39
3.5.2	Pengaturan Lingkungan Tanaman Tomat	40
3.5.3	Pengeluaran Biaya Sistem	40
<b>3.6</b>	<b>Pengukuran Kualitas Jaringan (QoS)</b>	<b>41</b>
3.6.1	Parameter QoS	41
3.6.2	Wireshark	41
3.6.3	Skenario Pengujian QoS	42



3.6.4	Langkah Percobaan QoS .....	42
3.7	Standar Pengukuran Kualitas Jaringan (QoS) .....	43
3.7.1	Kriteria Umum Standar TIPHON.....	43
3.7.2	Parameter Kualitas Jaringan .....	44
<b>BAB IV</b>	<b>ANALISIS DATA DAN PEMBAHASAN .....</b>	<b>46</b>
4.1	Implementasi Sistem .....	46
4.1.1	Implementasi Perangkat Keras.....	46
4.1.2	Implementasi Perangkat Lunak.....	46
4.2	Hasil Pengujian Sistem .....	47
4.2.1	Hasil Uji Fungsi Sensor.....	47
4.2.2	Hasil Uji Fungsi Aktuator.....	48
4.2.3	Hasil Uji Fungsi IoT .....	48
4.3	Hasil Pengukuran Kualitas Jaringan (QoS) .....	50
4.3.1	Hasil Perbandingan Jumlah Sensor.....	50
4.3.2	Hasil Perbandingan Waktu Pengamatan .....	50
4.3.3	Hasil Perbandingan Jeda Waktu Transmisi .....	51
4.4	Analisa dan Pembahasan Sistem.....	52
4.5	Perbandingan Performansi Sistem .....	53
4.5.1	Hasil Pengukuran <i>Throughput</i> .....	53
4.5.1.1	<i>Throughput</i> Perbandingan Jumlah Sensor .....	53
4.5.1.2	<i>Throughput</i> Perbandingan Waktu Pengamatan .....	54
4.5.1.3	<i>Throughput</i> Perbandingan Jeda Transmisi.....	54
4.5.2	Hasil Pengukuran <i>Delay</i> .....	55

4.5.2.1	<i>Delay</i> Perbandingan Jumlah Sensor .....	55
4.5.2.2	<i>Delay</i> Perbandingan Waktu Pengamatan .....	56
4.5.2.3	<i>Delay</i> Perbandingan Jeda Transmisi .....	58
4.5.3	Hasil Pengukuran <i>Jitter</i> .....	59
4.5.3.1	<i>Jitter</i> Perbandingan Jumlah Sensor .....	59
4.5.3.2	<i>Jitter</i> Perbandingan Waktu Pengamatan .....	60
4.5.3.3	<i>Jitter</i> Perbandingan Waktu Pengamatan .....	60
4.5.4	Hasil Pengukuran <i>Packet Loss</i> .....	61
4.5.4.1	<i>Packet Loss</i> Perbandingan Jumlah Sensor .....	61
4.5.4.2	<i>Packet Loss</i> Perbandingan Waktu Pengamatan.....	62
4.5.4.3	<i>Packet Loss</i> Perbandingan Jeda Transmisi .....	62
4.6	Klasifikasi Standar QoS (TIPHON) .....	63
4.7	Informasi Tambahan SiMonTa.....	65
4.7.1	Cara Penggunaan SiMonTa .....	65
4.7.2	Implementasi SiMonTa.....	66
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>68</b>
5.1	<b>Kesimpulan .....</b>	<b>68</b>
5.2	<b>Saran.....</b>	<b>68</b>
<b>DAFTAR PUSTAKA .....</b>		<b>70</b>
<b>LAMPIRAN.....</b>		<b>72</b>
<b>RIWAYAT HIDUP.....</b>		<b>Error! Bookmark not defined.9</b>

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Jaringan Sensor Nirkabel .....	9
<b>Gambar 2. 2</b> Topologi Star vs Topologi Mesh .....	10
<b>Gambar 2. 3</b> Komponen Utama IoT .....	11
<b>Gambar 2. 4</b> Komponen Mikrokontroler .....	14
<b>Gambar 2. 5</b> <i>User Interface</i> Wireshark.....	19
<b>Gambar 2. 6</b> Tampilan Arduino IDE .....	20
<b>Gambar 2. 7</b> Tampilan Sketch Arduino IDE .....	21
<b>Gambar 2. 8</b> Tampilan Aplikasi Blynk.....	22
<b>Gambar 3. 1</b> Diagram Alir Penelitian .....	23
<b>Gambar 3. 2</b> Blok Diagram Sistem.....	25
<b>Gambar 3. 3</b> NodeMCU ESP8266.....	27
<b>Gambar 3. 4</b> Modul Sensor Suhu DHT22 .....	28
<b>Gambar 3. 5</b> Sensor Kelembaban Tanah YL-69.....	29
<b>Gambar 3. 6</b> 2-Channel Relay Module .....	30
<b>Gambar 3. 7</b> Pompa Air 5V .....	31
<b>Gambar 3. 8</b> Brushless DC Fan .....	32
<b>Gambar 3. 9</b> Realisasi Perangkat Keras.....	33
<b>Gambar 3. 10</b> <i>Pseudocode</i> dari Sistem IoT <i>Monitoring</i> Tomat.....	34
<b>Gambar 3. 11</b> <i>Pseudocode</i> dari Modul IoT SiMonTa.....	34
<b>Gambar 3. 12</b> Tampilan Aplikasi Blynk.....	35
<b>Gambar 3. 13</b> Diagram Alir Sistem .....	36
<b>Gambar 4. 1</b> Implementasi Perangkat Keras .....	43
<b>Gambar 4. 2</b> Implementasi Program pada NodeMCU.....	43
<b>Gambar 4. 3</b> Pengujian Fungsi Sensor.....	44
<b>Gambar 4. 4</b> Pengujian Fungsi Aktuator .....	45
<b>Gambar 4. 5</b> Tampilan Blynk Empat Sensor .....	46
<b>Gambar 4. 6</b> Grafik <i>Throughput</i> Berdasarkan Jumlah Sensor .....	50
<b>Gambar 4. 7</b> Grafik <i>Throughput</i> Berdasarkan Waktu Pengamatan .....	51
<b>Gambar 4. 8</b> Grafik <i>Throughput</i> Berdasarkan Jeda Transmisi .....	51
<b>Gambar 4. 9</b> Rata-rata <i>Delay</i> Berdasarkan Jumlah Sensor .....	52
<b>Gambar 4. 10</b> Nilai <i>Delay</i> Pada Perbandingan Jumlah Sensor .....	53
<b>Gambar 4. 11</b> Rata-rata <i>Delay</i> Berdasarkan Waktu Pengamatan .....	53
<b>Gambar 4. 12</b> Nilai <i>Delay</i> Pada Perbandingan Waktu Pengamatan .....	54
<b>Gambar 4. 13</b> Rata-rata <i>Delay</i> Berdasarkan Jeda Transmisi .....	55

<b>Gambar 4. 14</b>	Nilai Delay Pada Perbandingan Jeda Transmisi .....	55
<b>Gambar 4. 15</b>	Nilai <i>Jitter</i> Berdasarkan Jumlah Sensor .....	56
<b>Gambar 4. 16</b>	Nilai <i>Jitter</i> Berdasarkan Waktu Pengamatan .....	57
<b>Gambar 4. 17</b>	Nilai <i>Jitter</i> Berdasarkan Jeda Transmisi .....	57
<b>Gambar 4. 18</b>	Nilai <i>Packet Loss</i> Berdasarkan Jumlah Sensor .....	58
<b>Gambar 4. 19</b>	Nilai <i>Packet Loss</i> Berdasarkan Waktu Pengamatan .....	59
<b>Gambar 4. 20</b>	Nilai <i>Packet Loss</i> Berdasarkan Jeda Transmisi .....	59

## DAFTAR TABEL

<b>Tabel 3. 1</b> Keterangan Segmen Blok Diagram Sistem .....	25
<b>Tabel 3. 2</b> Spesifikasi NodeMCU ESP8266 .....	27
<b>Tabel 3. 3</b> Spesifikasi DHT22 .....	28
<b>Tabel 3. 4</b> Spesifikasi YL-69.....	29
<b>Tabel 3. 5</b> Spesifikasi 2-Channel Relay Module .....	30
<b>Tabel 3. 6</b> Spesifikasi Pompa Air 5V Submersible .....	31
<b>Tabel 3. 7</b> Spesifikasi Brushless DC Fan 5V.....	32
<b>Tabel 3. 8</b> Presentase dan Nilai QoS .....	40
<b>Tabel 3. 9</b> Kategori Throughput .....	40
<b>Tabel 3. 10</b> Kategori Delay .....	41
<b>Tabel 3. 11</b> Kategori Jitter .....	41
<b>Tabel 3. 12</b> Kategori Packet Loss .....	41
<b>Tabel 4. 1</b> Nilai QoS Berdasarkan Jumlah Sensor.....	47
<b>Tabel 4. 2</b> Nilai QoS Berdasarkan Waktu Pengamatan .....	48
<b>Tabel 4. 3</b> Nilai QoS Berdasarkan Jeda Transmisi .....	48
<b>Tabel 4. 4</b> Nilai Indeks TIPHON Setiap Sistem Sensor .....	60
<b>Tabel 4. 5</b> Nilai Indeks TIPHON Setiap Waktu Pengamatan.....	61
<b>Tabel 4. 6</b> Nilai Indeks TIPHON Setiap Jeda Transmisi .....	62

*-Halaman ini sengaja dikosongkan-*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini masyarakat yang tinggal di daerah perkotaan yang padat sering dihadapi dengan masalah yang banyak. Gangguan kesehatan akibat buruknya kualitas udara yang penuh polutan, juga perubahan iklim yang sangat cepat. Laju pertumbuhan populasi di perkotaan yang pesat juga menimbulkan masalah lingkungan, mulai dari konversi lahan sampai degradasi kualitas lingkungan. Urgensi pertanian kota atau *urban farming* meningkat ketika krisis ekonomi membahayakan keamanan pangan, khususnya bagi masyarakat kota. Jarak perkotaan yang jauh dari sumber produksi pangan menjadi alasan pentingnya pertanian perkotaan. Kesegaran bahan makanan yang tersedia seperti sayur dan buah mengalami degradasi kualitas selama transportasi sehingga usaha meningkatkan kualitas dan akses terhadap bahan makanan melalui kegiatan pertanian perkotaan perlu untuk dilakukan.

Namun aktivitas masyarakat perkotaan yang sangat padat dan serba cepat seperti bekerja, menghambat kemampuan untuk melakukan *urban farming*. Salah satunya adalah perawatan pertaniannya dimana pada tanaman tertentu membutuhkan perhatian khusus agar bisa berkembang dengan baik.

Selain terhambatnya kemampuan *urban farming* akibat dari aktivitas masyarakat perkotaan yang padat dan cepat, segi kesehatan pun turut menjadi masalah. Tingkat stress yang tinggi dan kualitas udara yang penuh polutan mengganggu kesehatan masyarakat. Salah satu cara untuk meminimalisir dampak buruk dari masalah tersebut bagi tubuh adalah dengan mengkonsumsi buah tomat yang mengandung vitamin-c alami serta betakaroten dan likopen sebagai antioksidan yang baik bagi tubuh.

Dikarenakan pentingnya *urban farming* dan masalah yang dihadapi oleh masyarakat perkotaan, maka dalam tugas akhir ini akan

dibuatkan sebuah sistem *monitoring* dan otomasi *urban farming* pada budidaya tanaman tomat berbasis *Internet of Things*. Dengan adanya sistem *monitoring* berbasis IoT ini diharapkan dapat membantu masyarakat dalam membuat *urban farming* untuk tanaman tomat, sehingga petani *urban farming* juga bisa meningkat.

## 1.2 Rumusan Masalah

Beberapa permasalahan yang akan dibahas dalam tugas akhir tentang sistem absensi berbasis IoT ini adalah sebagai berikut:

1. Bagaimana merancang dan membuat sistem *monitoring* dan otomasi *urban farming* pada budidaya tanaman tomat berbasis IoT.
2. Bagaimana mengatur lingkungan agar tetap pada kondisi optimum bagi tanaman tomat.
3. Bagaimana kualitas jaringan yang digunakan untuk menghubungkan perangkat ke jaringan internet.

## 1.3 Batasan Masalah

Adapun batasan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Objek yang digunakan sebagai implementasi sistem adalah tanaman tomat.
2. Parameter yang diamati pada tanaman tomat adalah parameter suhu dan kelembaban tanah
3. Menggunakan sensor suhu (DHT-22), sensor kelembaban tanah (YL-69), Relay Module, Cooling Fan, dan DC Water Pump sebagai pengukur parameter dan aktuator.
4. Menggunakan NodeMCU-ESP8266 sebagai mikrokontroler dan modul komunikasi (IoT).
5. Menggunakan koneksi Wi-Fi untuk menghubungkan modul IoT dengan jaringan internet
6. Menggunakan *cloud platform* Blynk
7. Pengukuran kinerja sistem menggunakan perangkat lunak wireshark.
8. Menggunakan standar pengukuran ITU-T dan TIPHON (ETSI).



## 1.4 Tujuan

Adapun tujuan dari pembuatan tugas akhir ini adalah untuk menjawab berbagai permasalahan, antara lain:

1. Menciptakan sistem *monitoring* yang mudah diakses dari manapun selama ada koneksi internet.
2. Meningkatkan efisiensi dalam perawatan pertanian.
3. Melakukan pengukuran kinerja jaringan pada sistem yang telah dibuat.

## 1.5 Metodologi

Dalam pelaksanaan Tugas Akhir ini, menggunakan metodologi yang dapat diuraikan sebagai berikut:

### 1. Studi Literatur

Kegiatan studi literatur dimulai dengan mencari dan mempelajari referensi yang berhubungan dengan sistem *monitoring*, IoT dan pemrograman arduino. Pada tahap ini juga mempelajari tentang penggunaan *cloud platform* Blynk yang digunakan sebagai basis data. Juga pendefinisian *Quality of Service* (QOS) dari sebuah jaringan nirkabel

### 2. Pendefinisian Permasalahan

Pada tahap ini semua permasalahan yang muncul harus diidentifikasi sebaik-baiknya. Masalah yang teridentifikasi seperti bagaimana konsep rancang bangun rumah kaca pintar berbasis IoT untuk selanjutnya dapat dipertimbangkan dalam kebutuhan baik perangkat keras maupun perangkat lunak sebagai sistem yang dibuat.

### 3. Perancangan Sistem

Dalam tahap ini dilakukan perancangan rangkaian perangkat keras yang digunakan agar dapat terintegrasi untuk membuat sistem *monitoring* tanaman berbasis IoT dan dapat terhubung dengan jaringan internet.

### 4. Realisasi Perangkat Lunak

Tahap ini terdiri dari analisa kebutuhan sistem, desain sistem, pengkodean sistem, pengujian sistem dan perawatan sistem perangkat lunak. Untuk merealisasikan hal tersebut

digunakan perangkat lunak Arduino IDE dan aplikasi berbasis web Blynk

## **5. Pengujian Sistem**

Dalam tahap pengujian sistem dilakukan implementasi terhadap alat yang dibuat seperti pengujian operasional dan pengujian jaringan.

## **6. Analisis Data dan Kesimpulan**

Dari hasil data skema pengujian pada tahap sebelumnya, dilakukan analisa perfromansi sistem untuk dihasilkan kesimpulan

## **7. Penulisan Buku Tugas Akhir**

Dalam tahap ini dilakukan penulisan buku laporan tugas akhir yang terdiri dari beberapa bagian, yaitu Bab 1 (Pendahuluan), Bab 2 (Tinjauan Pustaka), Bab 3 (Metodologi), Bab 4 (Hasil Data dan Analisa) dan Bab 5 (Penutup).

## **1.6 Sistematika Penulisan**

Laporan tugas akhir disusun dalam beberapa bab dengan perincian sebagai berikut:

### **BAB I : PENDAHULUAN**

Bab ini membahas tentang penjelasan mengenai latar belakang, permasalahan dan batasan masalah, tujuan, metode penelitian, sistematika pembahasan, dan relevansi.

### **BAB II : TINJAUAN PUSTAKA**

Bab ini membahas mengenai dasar teori yang digunakan untuk menunjang penyusunan tugas akhir ini. Seperti pemahaman tentang sistem *monitoring*, konsep IoT, *cloud platform*, juga komponen pendukung IoT seperti mikrokontroler dan modul IoT

### **BAB 3 : METODOLOGI PENELITIAN**

Bab ini membahas mengenai metode yang digunakan untuk merancang sistem *monitoring* tanaman tomat berbasis IoT, rumus-rumus yang dibutuhkan dalam perhitungan dan juga data-data pendukung perancangan sistem.

**BAB IV : ANALISA DATA DAN PEMBAHASAN**

Bab ini membahas tentang analisis dari hasil pengujian sistem. Hasil analisis meliputi pengujian QoS dan pengujian operasional.

**BAB V : KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dari hasil penelitian yang telah dilakukan

## **1.7 Relevansi**

Dengan dibuatnya rancang bangun prototipe rumah kaca pintar berbasis IoT untuk budidaya tanaman tomat sebagaimana yang dikerjakan dalam tugas akhir ini dapat dimanfaatkan untuk membantu petani tomat di perkotaan dalam hal *monitoring* dan perawatannya, dengan harapan proses dari pengiriman data hingga pemberian perintah dapat berlangsung dengan cepat sehingga mampu mendukung penanaman tomat secara mudah dengan implementasi IoT

*-Halaman ini sengaja dikosongkan-*

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem Monitoring.**

*Monitoring* atau pemantauan dapat didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan, dan penindakan atas informasi suatu proses yang sedang diimplementasikan. *Monitoring* dapat memberikan informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah optimasi atau perbaikan yang berkesinambungan. Pada pelaksanaannya, *monitoring* dilakukan ketika suatu proses sedang berlangsung.

Tujuan dari sistem *monitoring* dapat ditinjau dari segi objek dan subjek yang dipantau, serta hasil dari proses *monitoring* itu sendiri. Beberapa tujuan dari sistem *monitoring* yaitu memastikan proses yang dilakukan sesuai, menyediakan keakuratan data dan probabilitas tinggi bagi pelaku *monitoring* dan mengidentifikasi hasil yang tidak diinginkan pada suatu proses dengan cepat

Pada dasarnya, *monitoring* memiliki dua fungsi dasar yang berhubungan, yaitu memastikan proses sesuai dengan rencana, dan mengetahui perkembangan suatu objek dalam pencapaian target yang diharapkan. Umumnya *output* *monitoring* berupa *progress report*. Keluaran tersebut bertujuan untuk mengetahui kesesuaian proses telah berjalan. Ini juga berguna pada memperbaiki mekanisme proses atau kegiatan dimana *monitoring* berjalan.

Sistem *monitoring* dapat dilakukan dengan berbagai bentuk implementasi karena *monitoring* tidak memiliki acuan pasti sehingga pelaksanaannya mengacu ke arah improvisasi pengguna. Penggunaan sistem *monitoring*, disesuaikan dengan situasi dan kondisi pengguna. Beberapa bentuk sistem *monitoring* yang biasa digunakan seperti observasi proses kerja, membaca dokumentasi laporan, melihat *display* data kinerja, melakukan survey, juga inspeksi sampel kualitas dari suatu proses kerja.

#### **2.2 Tanaman Tomat**

Tomat (*Solanum lycoperscum L*) merupakan salah satu tanaman sayuran yang dapat tumbuh di seluruh bagian di dunia. Luas tanaman tomat di negara Tiongkok mencapai 5 juta ha dengan

produksi mendekati 129 juta ton atau lebih  $\frac{1}{4}$  luas tanaman tomat di dunia.

Tomat mengandung nutrisi seperti vitamin A, vitamin C, potassium, fosfor, magnesium dan kalsium. Disamping itu tomat juga mengandung antioksidan yang dapat mengurangi efek dari radikal bebas pada tubuh manusia.

### **2.2.1 Proses Tumbuh Tanaman Tomat**

Dari awal penanaman benih sampai panen, tanaman tomat memerlukan waktu kira-kira 8 minggu. Pada minggu pertama, mulai tumbuhnya batang, daun dan akar yang masih kecil. Minggu kedua, arah tumbuh batang sudah mulai tegak lurus dan pada batang daun terdapat rambut-rambut halus atau trikomata. Pada minggu ketiga, bagian batang mulai membentuk cabang-cabang. Panjang tanaman sudah mencapai 20 cm. Pada minggu keempat, diameter batang mulai membesar dan panjang tanaman bisa mencapai 45 cm. Pada minggu kelima, sudah muncul bunga pada batang dan setiap percabangan batang. Pada minggu keenam, tanaman tomat sudah mulai berbuah namun masih kecil. Pada minggu ketujuh, buah tomat mulai membesar dan berwarna hijau muda. Pada minggu kedelapan, buah tomat yang sudah berwarna merah tua bisa dipetik.

### **2.2.2 Merawat Tanaman Tomat**

Pertumbuhan dan perkembangan tanaman tomat dipengaruhi oleh beberapa faktor diantaranya adalah suhu dan kelembaban tanah. Apabila lingkungan kurang mendukung terhadap pertumbuhan dan perkembangan tanaman tomat ini maka akan mempengaruhi produktivitas tanaman tomat. Contohnya pada penyiraman tanaman, dimana tanah tempat tanaman tomat tumbuh memerlukan kelembaban tanah sekitar 50-60%. Suhu lingkungan juga perlu dijaga pada kisaran 20-30 derajat celcius agar tanaman dapat tumbuh sesuai dengan rencana

## **2.3 Jaringan Nirkabel**

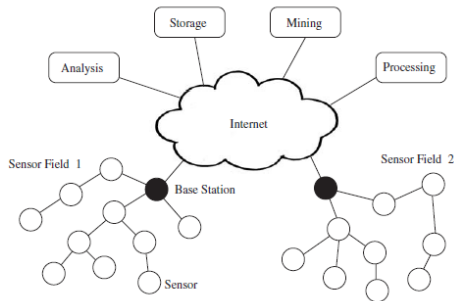
Jaringan Nirkabel merupakan sebuah jaringan yang memanfaatkan sinyal gelombang radio sebagai lapisan fisiknya. Perbedaan utama antara jaringan kabel dan nirkabel adalah peniadaan kabel sehingga pengguna memiliki mobilitas yang tinggi. Disamping itu, dikarenakan lapisan fisiknya bukan berupa benda seperti kabel,

maka perluasan jaringan tidak tergantung pada perangkat fisik yang banyak, namun cukup memberikan satu perangkat yang menjadi titik akses (*Access Point*). Dengan tidak bertambahnya perangkat setiap penambahan pengguna, teknologi ini dapat menghemat cukup banyak biaya. Namun terdapat beberapa kelemahan dari jaringan nirkabel bila dibandingkan dengan jaringan kabel atau *Wired Network*. Pertama ada error dikarenakan derau pada kanal transmisi, yang menjadi masalah cukup besar pada jaringan nirkabel. Juga kecepatan transmisi yang masih lebih cepat jaringan kabel dibandingkan dengan nirkabel.

Berdasarkan ukuran fisik area yang dapat dicakup, jaringan nirkabel terbagi menjadi beberapa kategori, yaitu *Wireless Personal Area Network* (WPAN) yang mencakup luasan daerah dengan radius 20 meter, kemudian *Wireless Local Area Network* (WLAN) yang dapat mencakup sebuah kawasan atau sebuah gedung, kemudian *Wireless Metropolitan Area Network* (WMAN) yang mencakup kawasan perkotaan atau antar gedung, dan yang terakhir *Wireless Wide Area Networks* (WWAN) yang mencakup antar Negara atau antar benua.

### **2.3.1 *Wireless Sensor Networks* (WSN)**

Aplikasi dari jaringan nirkabel digunakan dalam penggunaan sensor. Sensor sendiri merupakan sebuah komponen elektronik yang digunakan untuk mengukur parameter sebuah kejadian. Jaringan sensor nirkabel digunakan dalam Tugas Akhir ini. Implementasi jaringan sensor nirkabel terjadi karena banyaknya sensor yang mentransmisikan data ke sensor lain ataupun ke kontroller sebagai pusat data. Maka selain dari mengukur parameter, fungsi sensor bertambah menjadi tempat memproses, mentransmisikan dan menyimpan data. Contohnya pada gambar dibawah ini dimana terdapat dua sensor berada di daerah yang berbeda dan sama-sama terhubung ke internet melalui kontroller mereka (*Base Station*)



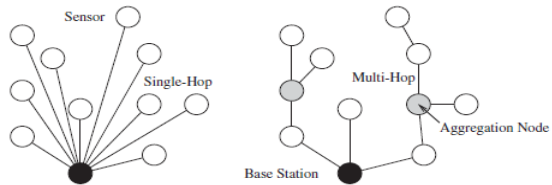
**Gambar 2. 1** Jaringan Sensor Nirkabel

Kemampuan dari sensor dalam jaringan sensor nirkabel bervariasi cukup luas. Misalnya satu sensor dapat mengukur satu parameter tertentu, sedangkan satu sensor lainnya dapat mengukur parameter yang berbeda-beda.

### **2.3.2 Komunikasi dalam WSN**

Sensor nirkabel menggunakan daya yang cukup rendah dibandingkan dengan komponen telekomunikasi lainnya dan data yang ditransmisikan juga rendah. Maka protokol IEEE 802.15.4 (Gutierrez, 2001) telah di desain untuk komunikasi jarak dekat untuk sensor nirkabel. Terdapat dua topologi yaitu topologi star dan topologi mesh. Pada topologi star, masing-masing sensor terhubung langsung ke controller. Apabila jarak antara sensor dan controller terlalu jauh dan daya yang digunakan menjadi besar, maka topologi mesh yang digunakan. Pada topologi mesh, sensor-sensor nirkabel saling terhubung untuk dapat mengirimkan data melalui sensor-sensor tersebut. Kedua topologi tersebut dapat dijelaskan melalui gambar dibawah ini





**Gambar 2. 2** Topologi Star vs Topologi Mesh

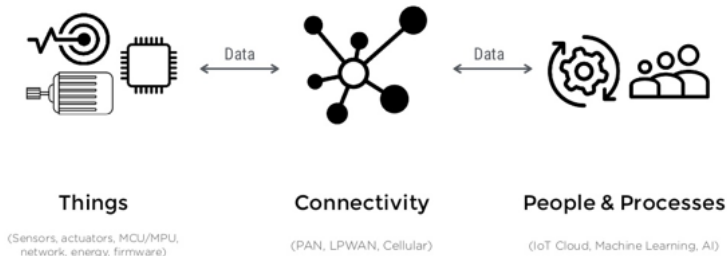
## 2.4 Internet of Things (IoT)

Konsep IoT pertama kali dicetus oleh salah satu anggota dari komunitas *Radio Frequency Identification* (RFID) di tahun 1999 dan semenjak itu terus berkembang dikarenakan pertumbuhan perangkat seluler, komunikasi tertanam dan *data analytics*. Definisi umum dari IoT adalah sebuah jaringan objek fisik yang saling terhubung melalui internet. IoT sendiri terbagi menjadi tiga kategori, yaitu *People-to-People*, *People-to-Machine/Things* dan *Machine/Things-to-Machine/Things*.

IoT mengacu pada gagasan umum tentang berbagai hal terutama objek sehari-hari yang dapat dibaca, dikenali, ditemukan, dialamatkan dan/atau dikendalikan melalui internet. Objek sehari-hari tidak hanya mencakup perangkat elektronik yang kita biasa temui, tetapi hal-hal yang tidak berhubungan dengan elektronik seperti makanan, pakaian, furnitur, hewan, tanaman, air, dan lainnya. Berdasarkan cakupan dari IoT maka implementasinya pun sangat luas, tidak terbatas pada komponen elektronik.

### 2.4.1 Cara Kerja IoT

Tiga komponen utama dalam sistem IoT adalah *Things*, *Connectivity* dan *People & Process*.



**Gambar 2. 3** Komponen utama IoT

*Things* atau benda adalah objek seperti sensor, aktuator atau komputer sebagai sumber data yang digunakan. *Connectivity* atau jaringan adalah kanal transmisi dari sistem IoT, contohnya adalah jaringan selular, Wi-Fi, LPWAN. Kemudian *People & Process* atau manusia yang menggunakan dan menganalisa data yang sudah dikirim, ataupun juga bisa mengirimkan perintah kembali ke objek IoT.

#### 2.4.2 Arsitektur IoT

IoT memiliki layer-layer dalam implementasinya, seperti OSI layer yang memiliki layer-layer sesuai dengan fungsinya. Belum ada standar untuk IoT layer yang diakui dunia, namun dalam implementasi yang sudah dilakukan, layer-layer dalam IoT terbagi menjadi 4 layer.

##### a. Sensor Layer

Dalam layer ini terdiri dari objek-objek IoT seperti sensor, aktuator dan mikrokontroler yang mengambil dan menyimpan sementara data parameter yang diamati

##### b. Gateway and Network Layer

Layer ini berisi perangkat yang menghubungkan perangkat pada sensor layer dengan Management Service Layer. Contohnya adalah modul komunikasi ESP8266, Wi-Fi, NB-IoT, LoRa, dsb.

### **c. Management Service Layer**

Layer ini berfungsi sebagai manajemen perangkat dan sebagai pusat penyimpanan data (*Cloud Services*).

### **d. Application Layer**

Layer teratas ini berfungsi sebagai *user interface* untuk pengguna sistem IoT dimana pengguna bisa melihat hasil data ataupun memberikan perintah melalui *interface* ini

## **2.5 Cloud**

Istilah *cloud* pertama kali dicetuskan oleh John Gage dari Sun Microsystems di tahun 1984 yang mengatakan bahwa “Jaringan terdiri dari komputer-komputer”. Di era sekarang, *cloud* sendiri dapat diartikan sebagai jaringan server yang menyimpan data seperti aplikasi, dokumen, musik, video, dan sebagainya. Data-data tersebut dapat diakses melalui web. Contohnya adalah Google docs, aplikasi berbasis web yang penggunaanya tidak perlu menginstall software tersebut di komputernya. Tujuan dari penggunaan *cloud* adalah sebagai pengganti dari perangkat komputasi, basis data dan komunikasi menjadi lebih sederhana karena tidak memerlukan perangkat keras yang banyak.

### **2.5.1 Jenis Cloud Berdasarkan Pengguna**

Dalam usaha *cloud*, terdapat empat pembagian *cloud* berdasarkan jenis penggunaanya:

#### **a. Public Cloud**

Layanan *cloud* dimiliki oleh sebuah perusahaan dan mereka menyewakan ataupun memberikan secara gratis *storage cloud* mereka ke publik. Contoh dari Public Cloud adalah Azure (Microsoft) dan Google Cloud Platform

#### **b. Private Cloud**

Layanan *cloud* yang dimiliki oleh sebuah perusahaan dan hanya digunakan oleh perusahaan tersebut.

**c. Community Cloud**

Layanan cloud yang di miliki bersama oleh beberapa organisasi untuk mencapai kebutuhan bersama. Contohnya cloud untuk perusahaan-perusahaan di bidang kesehatan

**d. Hybrid Cloud**

Gabungan dari dua atau lebih jenis cloud (Public, private atau community)

**2.5.2 Jenis Cloud Berdasarkan Model Layanan**

Dalam bisnis *cloud* juga terdapat pembagian model bisnis berdasarkan jenis layanannya:

**a. Infrastructure-as-a-Service (IaaS)**

Sistem *cloud* meliputi server, jaringan, sistem operasi, dan *database* melalui *Virtual Machine*. Di sini pengguna mengatur komponen dari *cloud* seperti data, aplikasi, server dan lainnya secara mandiri.

**b. Platform-as-a-Service (PaaS)**

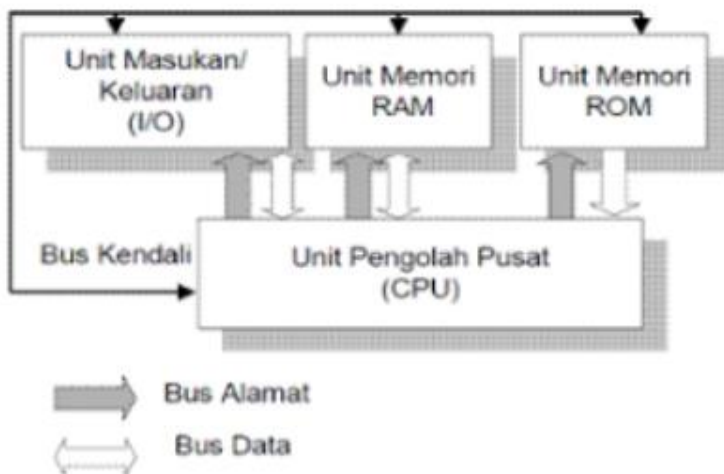
Meliputi komponen *cloud* seperti software untuk membuat atau menjalankan aplikasi. Di sini pengguna bisa langsung menggunakan software dan hanya mengatur software yang digunakan, tidak seperti SaaS yang harus bertanggung jawab atas semuanya.

**c. Software-as-a-Service (SaaS)**

Di bagian ini pengguna bisa menggunakan jasa *cloud* melalui jaringan internet, tidak seperti IaaS dan PaaS yang mengakses *cloud* melalui jaringan internal mereka. Pengguna SaaS tidak perlu memikirkan masalah server dan *database* karena itu menjadi tanggung jawab penyedia layanan, pengguna SaaS hanya menggunakan layanan. Contoh dari SaaS adalah Dropbox dan Google GSuite

## 2.6 Mikrokontroler

Mikrokontroler adalah sebuah sirkuit yang terintegrasi yang disatukan dengan komponen lainnya agar bisa melakukan operasi tertentu secara berulang-ulang kali tanpa deprogram ulang. Mikrokontroler terdiri dari prosesor mikro, *memory units*, *I/O interfaces*, ADC (Analog-to-Digital Conversion) dan beberapa modul kontrol dan komunikasi. Mikrokontroler digunakan untuk



berbagai aplikasi seperti otomasi, akuisisi data dan telekomunikasi, Keuntungan menggunakan mikrokontroler yaitu harga yang murah, dapat di program berulang kali, dan bentuk yang simpel. Mikrokontroler biasanya dikelompokkan kedalam satu kelompok tertentu tergantung spesifikasinya masing-masing. Misalnya MCS-51 yang diproduksi ATMEL dan AVR yang diproduksi Atmega.

**Gambar 2. 4** Komponen Mikrokontroler

Komponen pada mikrokontroler terdiri dari beberapa hal berikut:

**a. CPU (*Central Processing Unit*)**

CPU pada mikrokontroler terdiri dari dua bagian yaitu unit pengendali (*Control Unit*) dan unit logika (*Arithmetic-Logical*

Unit). Unit kontrol berfungsi untuk mengatur dan mengendalikan komponen pada mikrokontroler seperti contoh mengatur kapan alat input menerima dan mengirimkan data. Unit logika berfungsi untuk melakukan perhitungan yang dibutuhkan pada program

**b. RAM (Random Access Memory)**

RAM adalah memori yang dapat dibaca atau ditulis . Pada mikrokontroler, RAM berfungsi untuk menyimpan data pada saat program bekerja.

**c. ROM (Read Only Memory)**

ROM adalah memori yang hanya dapat dibaca. Pada mikrokontroler ROM berfungsi untuk menyimpan program yang diupload ke mikrokontroler. Bedanya dengan RAM, program yang disimpan pada ROM tidak akan hilang apabila mikrokontroler tidak tersambung dengan catu daya. Jenis ROM lainnya adalah PROM (*Programmable Read Only Memory*) dan EPROM (*Erasable Programmable Only Memory*). Yang biasa digunakan pada mikrokontroler adalah EPROM karena dapat diprogram ulang.

**d. Unit keluaran masuk (I/O)**

Jika mikrokontroler membutuhkan komponen eksternal untuk melakukan kerjanya, maka komponen tersebut terhubung melalui antarmuka I/O pada mikrokontroler. Contoh peralatan I/O yang biasa digunakan di mikrokontroler seperti sensor, LED dan modul komunikasi.

**e. Clock**

*Clock* berfungsi sebagai referensi waktu pada mikrokontroler dan untuk sinkronisasi antar komponen

**f. Bus Alamat**

Berfungsi sebagai lintasan pengalamanan instruksi ke komponen yang tersebar didalam mikrokontroler

**g. Bus Data**

Berfungsi sebagai lintasan keluar masuknya data dalam mikrokontroler

**h. Bus Kontrol**

Berfungsi sebagai pengendali komponen-komponen mikrokontroler agar tidak bertabrakan dalam proses bekerjanya.

**2.7 Quality of Service (QoS)**

*Quality of Service* adalah sebuah parameter kualitas untuk menentukan apakah sebuah jaringan telekomunikasi bekerja dengan baik atau tidak. Untuk mencapai sebuah pernyataan apakah sebuah jaringan mampu untuk menangani aliran paket data yang besar, dilakukan skema pengujian dengan trafik data yang berbeda agar dapat dicapai sebuah kesimpulan.

**2.7.1 Throughput**

*Throughput* adalah kecepatan (rate) transfer data yang diukur dalam satuan bps (bit per second). *Throughput* merupakan jumlah total paket datang dengan sukses yang diamati selama interval waktu tertentu. Untuk menghitung nilai persamaan perhitungan *Throughput* dapat menggunakan persamaan dibawah ini:

$$\textit{Throughput}(\text{bps}) = \frac{\text{Total paket data diterima (bit)}}{\text{Lama waktu pengiriman (s)}} \quad \dots (2.1)$$

Dimana satuan paket data yang diterima dalam bit dan lama waktu pengiriman dalam detik

**2.7.2 Delay**

Delay (Latency) merupakan waktu tunda yang terjadi karena proses transmisi data. Delay dipengaruhi oleh beberapa faktor diantaranya jarak, media fisik dan waktu proses yang dibutuhkan. Satuan dari delay adalah second (s). Semakin kecil delay berarti proses transmisi data semakin baik. Untuk menghitung nilai *delay* dapat menggunakan persamaan di bawah ini:

$$\text{Delay} = \text{Waktu diterima (s)} - \text{Waktu terkirim (s)} \dots (2.2)$$

Dimana satuan waktu terima dan terkirim dalam detik

### 2.7.3 Jitter

Jitter adalah variasi delay antar paket yang diakibatkan oleh panjang antrian dalam suatu pengolahan data dan penghimpunan ulang paket-paket data diakhir pengiriman akibat kegagalan sebelumnya. Delay antrian pada router dan switch menyebabkan jitter. Semakin besar beban trafik atau nilai variasi delay dalam suatu jaringan akan menyebabkan semakin besar pula peluang terjadinya tumbukan antar paket, sehingga nilai jitter akan semakin besar. Untuk menghitung nilai *jitter* dapat menggunakan persamaan dibawah ini:

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{Total paket yang diterima}} \dots (2.3)$$

Dimana satuan variasi delay dalam detik dan satuan paket yang diterima dalam bit

Total variasi delay merupakan perbandingan delay dengan delay rata-rata atau secara statistik disebut dengan standar deviasi. Standar deviasi digunakan untuk memperlihatkan seberapa besar perbedaan data yang ada dibandingkan dengan rata-rata dari data itu sendiri. Persamaan total variasi delay adalah dibawah ini:

$$\text{Total variasi delay} = \sqrt{s^2} = \left( \frac{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}{n(n-1)} \right) \dots (2.4)$$

Keterangan:

- $s^2$  = varians dari delay
- $n$  = banyaknya data
- $\sum_{i=1}^n x_i$  = Jumlah total data
- $\sum_{i=1}^n x_i^2$  = Jumlah total dari data yang dikuadratkan



$$(\sum_{i=1}^n x_i)^2 = \text{Jumlah total data dikuadratkan}$$

\*data yang dimaksud pada rumus merupakan nilai delay

#### 2.7.4 *Packet Loss*

Packet Loss merupakan sebuah parameter yang menunjukkan kegagalan dalam proses transmisi paket data untuk mencapai tujuan. Terdapat beberapa faktor yang menyebabkan kegagalan proses transmisi paket data tersebut, diantaranya adalah:

- a. Terjadinya overload trafik dalam jaringan.
- b. Tumbukan (*congestion*) dalam jaringan.
- c. Error yang terjadi pada media fisik.
- d. Kegagalan yang terjadi pada sisi penerima.

Persamaan perhitungan *Packet Loss*:

$$\text{Packet Loss} = \frac{(\text{Paket dikirim} - \text{Paket diterima})}{\text{Total paket yang dikirim}} \times 100\% \quad \dots (2.5)$$

Dimana satuan paket data dalam bit

## 2.8 **Wireshark**

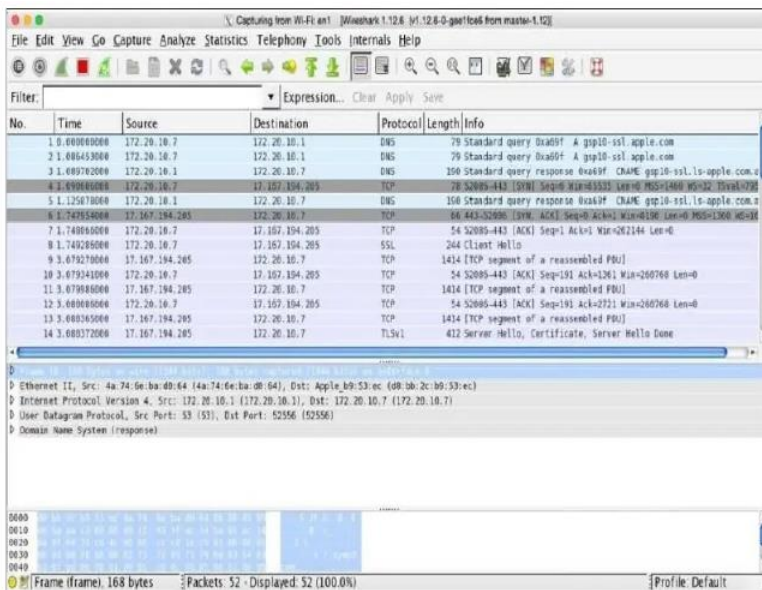
Wireshark adalah sebuah perangkat lunak untuk menangkap dan menganalisa paket data dalam sebuah jaringan telekomunikasi. Selain paket data, wireshark juga digunakan untuk menganalisa protokol komunikasi.

Wireshark mampu menangkap paket-paket data atau informasi yang berjalan dalam suatu jaringan yang terlihat dan semua jenis informasi ini dapat dengan mudah dianalisa memakai sniffing, dengan sniffing ini dapat diperoleh informasi penting seperti password email account milik orang lain. Wireshark dapat membaca data secara langsung dari Ethernet, Token-Ring dan 802.11 Wireless LAN

Salah satu aspek penting dari sniffing adalah di mana menempatkan sniffer paket ke dalam jaringan fisik untuk mencapai pemanfaatan maksimum darinya; sniffing sering disebut sebagai mengetuk ke dalam jaringan.. Menempatkan sniffer di tempat yang tepat dapat memengaruhi pengalaman analisis paket Anda secara

terperinci, yang pada akhirnya dapat menyebabkan hasil drastis jika dilakukan dengan benar

Secara garis besar kerja Wireshark dibagi menjadi tiga tahapan yaitu *collect*, *convert* dan *analyze*. Pada tahap *collect*, transmisi data pada suatu jaringan diamati dan direkam untuk dapat dilakukan perhitungan. Tahap *convert* yaitu mengubah bentuk data yang direkam agar bisa dianalisa menggunakan *software* tertentu. Yang terakhir adalah *analyze* yaitu tahap analisa dari data yang sudah direkam.

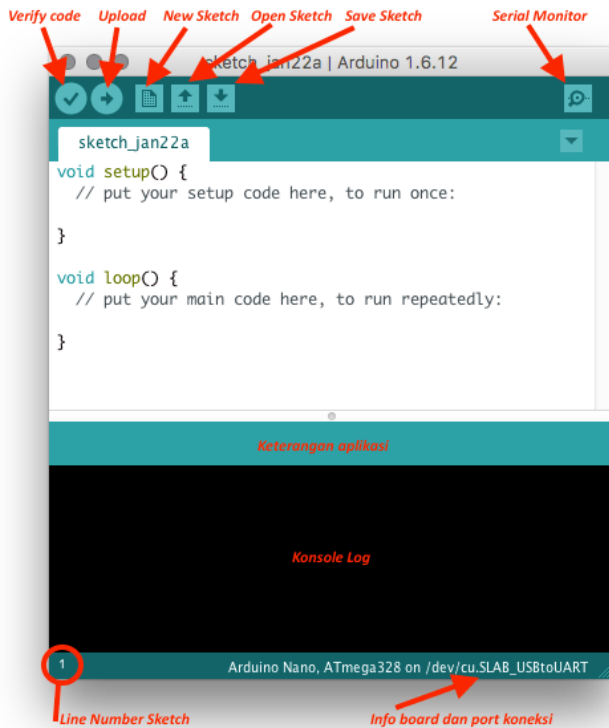


Gambar 2. 5 User Interface Wireshark

## 2.9 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak yang digunakan untuk memprogram board mikrokontroler Arduino dan mikrokontroler lainnya seperti NodeMCU dan ESpectro. Arduino IDE berfungsi sebagai *text-editor* untuk membuat, mengedit, memvalidasi dan meng-*upload*

kode program atau *sketch* pada board mikrokontroler. Ekstensi file *sketch* ini adalah *.ino*. Arduino IDE dibuat menggunakan bahasa pemrograman JAVA. *Library* atau bahasa pemrograman yang digunakan pada *software* untuk membuat *sketch* menggunakan C karena Arduino IDE bersifat *open source* yaitu semua orang bisa menggunakan dan mengedit *software* tersebut.



**Gambar 2. 6** Tampilan Arduino IDE

*Sketch* Arduino sendiri dibagi menjadi tiga bagian utama yaitu *Structure*, *Values* dan *Function*

### a. Structure

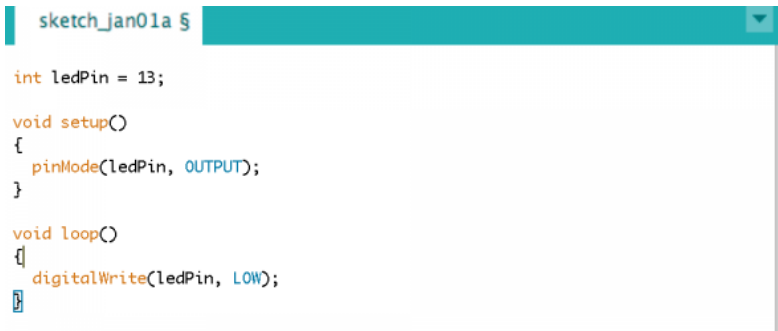
Berisi kerangka program terdiri dari fungsi `setup()` dan `loop()`. `Setup()` berfungsi sebagai pengawalan variabel, penggunaan library dan *pin mode*. `Loop()` berfungsi untuk melakukan pengulangan program pada fungsi `setup()`

### b. Values

Berisi konstanta yang sesuai dengan tipe data yang didukung oleh mikrokontroler

### c. Function

Segmentasi kode kedalam potongan-potongan modular yang memungkinkan sebuah kode melakukan tugas yang didefinisikan kemudian kembali ke asal dimana fungsi itu berada.

A screenshot of the Arduino IDE interface. The title bar shows 'sketch\_jan01a 5'. The code editor contains the following code:

```
int ledPin = 13;

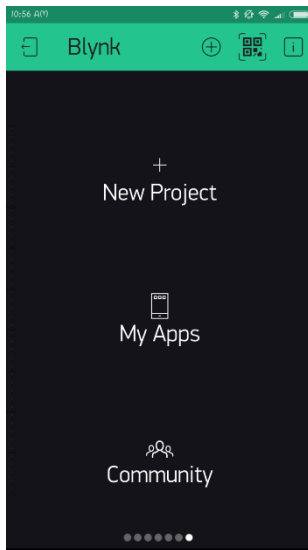
void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, LOW);
}
```

**Gambar 2. 7** Tampilan Arduino IDE

## 2.10 Blynk

Blynk adalah sebuah aplikasi berbasis web untuk smartphone yang digunakan sebagai *cloud platform* khusus untuk proyek IoT. Blynk terdapat pada *platform smartphone* iOS dan Android. Blynk digunakan untuk mengatur secara jarak jauh mikrokontroler seperti Arduino dan NodeMCU.

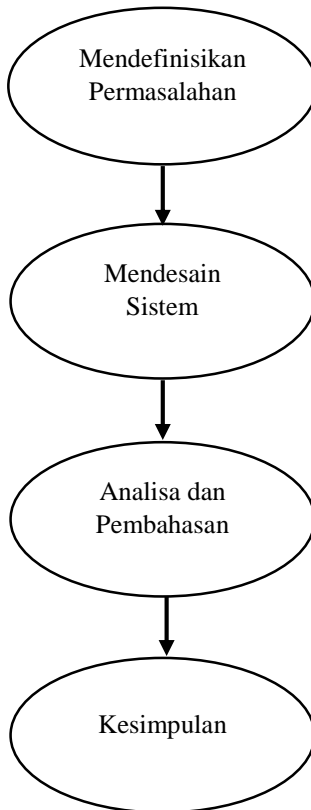


**Gambar 2. 8** Tampilan Aplikasi Blynk

## BAB III METODOLOGI PENELITIAN

### 3.1 Diagram Alir Penelitian

Berikut adalah diagram alir penelitian menuju sistem *monitoring* budidaya tanaman tomat



**Gambar 3. 1** Diagram Alir Penelitian

### 3.2 Identifikasi Permasalahan dan Solusi

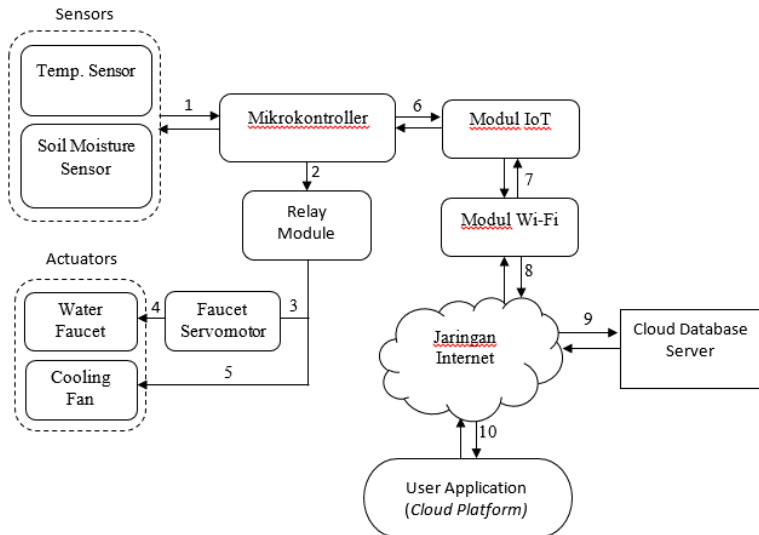
Saat ini masyarakat yang tinggal di daerah perkotaan yang padat sering dihadapi dengan masalah yang banyak. Gangguan kesehatan akibat buruknya kualitas udara yang penuh polutan, juga perubahan iklim yang sangat cepat. Laju pertumbuhan populasi di perkotaan yang pesat juga menimbulkan masalah lingkungan, mulai dari konversi lahan sampai degradasi kualitas lingkungan. Urgensi pertanian kota atau *urban farming* meningkat ketika krisis ekonomi membahayakan keamanan pangan, khususnya bagi masyarakat kota. Jarak perkotaan yang jauh dari sumber produksi pangan menjadi alasan pentingnya pertanian perkotaan. Kesegaran bahan makanan yang tersedia seperti sayur dan buah mengalami degradasi kualitas selama transportasi sehingga usaha meningkatkan kualitas

Tingkat stress yang tinggi dan kualitas udara yang penuh polutan mengganggu kesehatan masyarakat. Salah satu cara untuk meminimalisir dampak buruk dari masalah tersebut bagi tubuh adalah dengan mengonsumsi buah tomat yang mengandung vitamin-c alami serta betakaroten dan likopen sebagai antioksidan yang baik bagi tubuh

Namun penanaman tanaman tomat di perkotaan menemui beberapa masalah seperti lingkungan perkotaan yang tidak sesuai dengan lingkungan optimal untuk pertumbuhan tanaman tomat. Juga masyarakat perkotaan yang sibuk sehingga tidak bisa selalu melakukan perawatan tanaman tomat secara langsung.

Dikarenakan pentingnya *urban farming* dan masalah yang dihadapi oleh masyarakat perkotaan, maka dibuat solusi yaitu sebuah sistem *monitoring* budidaya tanaman tomat berbasis IoT. Dengan adanya sistem *monitoring* berbasis IoT diharapkan dapat membantu masyarakat dalam membuat *urban farming* untuk tanaman tomat.

### 3.3 Desain Sistem Budidaya Tomat Berbasis IoT



**Gambar 3. 2** Blok Diagram Sistem

**Tabel 3. 1** Keterangan Segmen Blok Diagram Sistem

Segmen	Medium	Keterangan
1	Kabel ( <i>Wired</i> )	-
2	Kabel ( <i>Wired</i> )	-
3	Kabel ( <i>Wired</i> )	-
4	Kabel ( <i>Wired</i> )	-
5	Kabel ( <i>Wired</i> )	-
6	PCB ( <i>Embedded</i> )	-
7	Udara ( <i>Wireless</i> )	Protokol IEEE 802.11
8	Udara ( <i>Wireless</i> )	Protokol TCP/IP
9	Kabel ( <i>Wired</i> )	-
10	Udara ( <i>Wireless</i> )	Protokol HTTP



Sistem yang akan dibuat dalam tugas akhir ini adalah sebuah sistem *monitoring* tanaman tomat dengan menggunakan konsep teknologi *Internet of Things* atau disingkat SiMonTa (SistemMonitoring Tomat) Komponen utama dalam pembuatan sistem ini adalah mikrokontroler dimana komponen ini berfungsi sebagai pengatur kerja seluruh perangkat keras pada sistem serta berperan dalam pengiriman data ke server melalui jaringan internet. Mikrokontroler membaca data yang diperoleh dari sensor suhu dan sensor kelembaban. Data yang ditangkap dikirimkan ke server *cloud* menggunakan modul IoT melalui jaringan internet. Pada *user interface*, data yang sebelumnya ditangkap dapat ditampilkan

Perintah juga dapat dikirimkan melalui *user interface* dimana bermula dari *smartphone* mengirimkan data perintah ke mikrokontroler melalui jaringan internet. Setelah perintah diterima mikrokontroler, akan diteruskan ke modul relay untuk menggerakkan pompa air dan kipas pendingin

### **3.4 Pembuatan Sistem *Monitoring* IoT Tomat (SiMonTa)**

Sub-bab ini menjelaskan tentang proses pembuatan prototipe SiMonTa mulai dari perangkat keras sampai perangkat lunak yang digunakan

#### **3.4.1 Perakitan Perangkat Keras**

Dalam perakitan perangkat keras, sebelumnya ditentukan terlebih dahulu model perangkat yang digunakan di setiap komponen

##### **3.4.1.1 Perakitan Perangkat Keras**

Perangkat keras yang dibutuhkan dalam membuat sistem absensi perkuliahan berbasis IoT ini adalah sebagai berikut:

## A. NodeMCU ESP8266

**Tabel 3. 2** Spesifikasi NodeMCU ESP8266

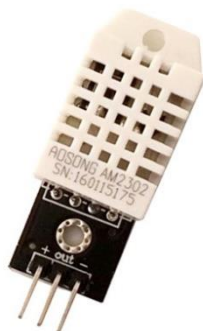
Spesifikasi	Nilai
Mikrokontroler/Chip	ESP-8266 32-bit
NodeMCU Model	Clone LoLin
Operating Voltage	3.3V
Input Voltage	4.5V-10V
Digital I/O Pins	11 Pin
Analog In Pins	1 Pin
Flash Memory / SRAM	4 MB / 64KB
Clock Speed	80 MHz
WiFi	IEEE 802.11 b/g/n
Frequency	2.4 GHz ~ 22.5 GHz
USB Connector	Micro USB
USB to Serial	CH340G

NodeMCU versi LoLin adalah mikrokontroler yang digunakan dalam tugas akhir ini. Mikrokontroler ini sudah terpasang modul Wi-Fi ESP8266 secara *embedded* sehingga memudahkan dalam pemrograman dan penghematan perangkat keras yang dibutuhkan.



**Gambar 3. 3** NodeMCU ESP8266

## B. Modul Sensor Suhu DHT-22



**Gambar 3. 4** Modul Sensor Suhu DHT22

**Tabel 3. 3** Spesifikasi DHT22

<b>Spesifikasi</b>	<b>Nilai</b>
Operating Voltage	3.5V – 5.5V
Operating Current	0.3mA (measuring) 60uA (standby)
Output signal	Digital
Sensing element	Polymer capacitor
Sensing periode	Average : 2s
Operating Range	Temperature -40~80 °C ; Humidity 0-100%RH
Resolution	16-bit
Dimensions (L * W * H):	22 x 28 x 5 mm

Sensor suhu yang digunakan dalam tugas akhir ini adalah DHT22. DHT22 memiliki NTC atau thermistor untuk mendeteksi suhu dan juga 8-bit mikrokontroler untuk mengubah data dari thermistor menjadi serial dan bisa dibaca oleh mikrokontroler seperti Arduino dan NodeMCU. Sensor ini mampu mengukur suhu dan kelembaban udara dengan rentang suhu -40°C to 80°C dan rentang kelembaban 0% - 100%, dan memiliki keakuratan  $\pm 1^\circ\text{C}$  and  $\pm 1\%$ . Spesifikasi dari DHT22 dijelaskan pada Tabel 3.3

### C. Sensor Kelembaban Tanah YL-69



**Gambar 3. 5** Sensor Kelembaban Tanah YL-69

**Tabel 3. 4** Spesifikasi YL-69

<b>Spesifikasi</b>	<b>Nilai</b>
Operating Voltage	3.3V – 5V
Operating Current	0 mA – 35 mA
Output Value	Dry soil 0~300 mV Humid soil 300~700 mV

Gambar diatas adalah sensor kelembaban tanah tipe YL-69. Sensor tersebut bekerja mengukur kelembaban tanah dengan mengukur perbedaan tegangan. Apabila kelembaban tanah tinggi maka tegangan yg diukur tinggi, begitu juga sebaliknya. Sensor ini terdiri dari pin pengukur tanah dan komparator LM-393 sebagai pengaman. Spesifikasi dari komponen tersebut dijelaskan pada Tabel 3.4

## D. Relay Module



Gambar 3. 6 2-Channel Relay Module

Modul relay digunakan dalam tugas akhir ini untuk menggerakkan aktuator seperti kipas pendingin dan pompa DC. Modul relay terdiri dari 1-channel, 2-channel, 4-channel sampai dengan 64-channel. Relay ini termasuk *optocoupler* yang berarti menggunakan prinsip opto-isolation yang berarti kontroler pada relay tidak berhubungan langsung dengan sumber tegangan. Aktuator yang menggunakan relay ini bekerja dengan sistem *normally-open* atau *normally-closed*. Berikut adalah spesifikasi dari Modul Relay yang digunakan:

Tabel 3. 5 Spesifikasi 2-Channel Relay Module

Spesifikasi	Nilai
Maximum Relay Output	DC-30V/10A ; AC-250V/10A
Relay Lifetime	100000 clicks
Input voltage	5 VDC
Terminals	Common terminal NO/NC
Isolation	Opto-Coupler Isolation

## E. Pompa Air Submersible



**Gambar 3. 7** Pompa Air 5V

Pompa air digunakan dalam tugas akhir ini untuk memompa dan mengalirkan air ke objek tanaman tomat. Pompa air yang digunakan adalah tipe submersible 3-5V karena kebutuhan yang tidak terlalu besar. Berikut spesifikasi dari pompa air:

**Tabel 3. 6** Spesifikasi Pompa Air 5V Submersible

<b>Spesifikasi</b>	<b>Nilai</b>
Voltage Rate	3-5VDC
Maximum Lift	40-110cm
Flow rate	80-120L/H
Water outlet diameter	Output-7.5 mm ; Input 4.7mm
Material	Engineering plastic
Driving mode	Brushless DC
Working life	Up to 500 hours
Diameter	24mm
Length	45mm
Height	33mm

## F. DC 5V Cooling Fan



**Gambar 3. 8** Brushless DC Fan

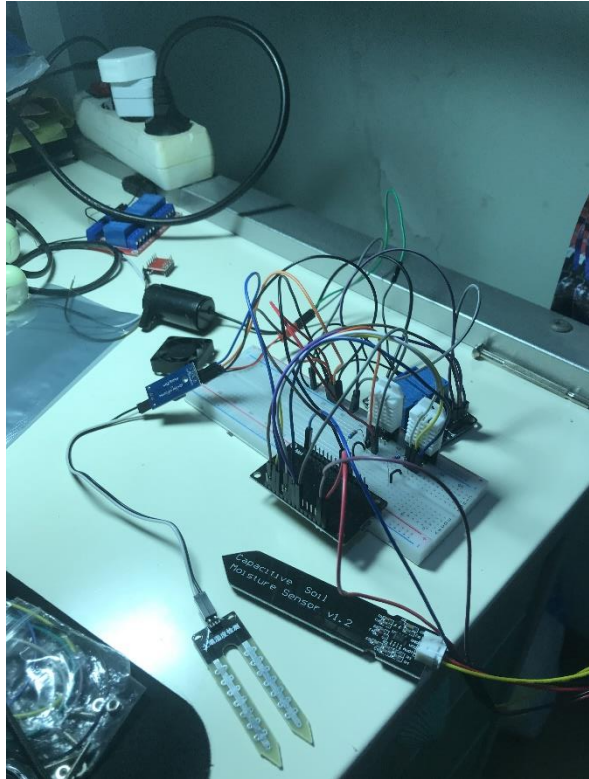
Kipas pendingin digunakan sebagai aktuator untuk menurunkan suhu pada lingkungan objek tanaman tomat. Kipas yang digunakan adalah *Brushless DC Fan* yang biasa digunakan untuk Raspberry Pi. Berikut spesifikasi kipas pendingin:

**Tabel 3. 7** Spesifikasi Brushless DC Fan 5V

<b>Spesifikasi</b>	<b>Nilai</b>
Operating Voltage	5V
Operating Current	0.2A
Driver type	Brushless DC
Dimension	30mm x 30mm x 8mm
Weight	6.2g

### 3.4.1.3 Realisasi Perangkat Keras

Berikut adalah realisasi dari komponen-komponen seperti mikrokontroler, sensor dan aktuator yang diimplementasikan ke objek tanaman tomat



**Gambar 3. 9** Realisasi Perangkat Keras

### **3.4.2 Pembuatan Perangkat Lunak**

Dalam perancangan sistem *monitoring* tomat berbasis IoT, juga dibutuhkan perangkat lunak untuk memrogram perangkat keras agar bekerja sesuai parameter.

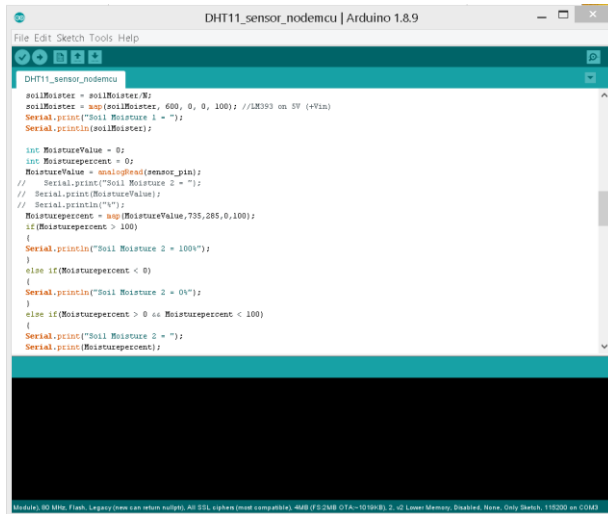
#### **3.4.2.1 Penggunaan Bahasa Pemrograman**

Bahasa pemrograman C digunakan untuk merancang program agar perangkat keras bekerja sesuai parameter yang di inginkan. Bahasa C lazim digunakan di *software* Arduino IDE untuk membuat program untuk mikrokontroler



### 3.4.2.2 Pemrograman Mikrokontroler

Program atau pseudocode yang digunakan pada *software* ini adalah untuk mengambil data dari sensor suhu dan kelembaban tanah.



```
File Edit Sketch Tools Help
DHT11_sensor_nodemcu | Arduino 1.8.9

DHT11_sensor_nodemcu

soilMoister = soilMoister/W;
soilMoister = map(soilMoister, 600, 0, 0, 100); //LM393 on 5V (+Vin)
Serial.print("Soil Moisture 1 = ");
Serial.println(soilMoister);

int MoistureValue = 0;
int Moisturepercent = 0;
MoistureValue = analogRead(sensor_pin);
// Serial.print("Soil Moisture 2 = ");
// Serial.println(MoistureValue);
// Serial.println("");
Moisturepercent = map(MoistureValue,755,285,0,100);
if(Moisturepercent > 100)
{
Serial.println("Soil Moisture 2 = 100%");
}
else if(Moisturepercent < 0)
{
Serial.println("Soil Moisture 2 = 0%");
}
else if(Moisturepercent > 0 && Moisturepercent < 100)
{
Serial.print("Soil Moisture 2 = ");
Serial.println(Moisturepercent);
}

(Monitor) 90 MHz, Flash, Legacy (you can return multiple), All SPI, esp8266 (most compatible), 4MB (FS:2MB OTA=1019KB), 2-GB (1MB memory, Disabled, None, Only Sketch, 115200 on COM4)
```

Gambar 3. 10 Pseudocode dari Sistem IoT Monitoring Tomat

### 3.4.2.3 Pemrograman Modul IoT

Program atau *pseudocode* yang digunakan agar mikrokontroler dapat mengirim data yang telah oleh sensor ke *cloud server*

```
#include <ESP8266WiFi.h>;
#include <BlynkSimpleEsp8266.h>;
#include <SimpleTimer.h>;
#include <DHT.h>;
SimpleTimer timer;

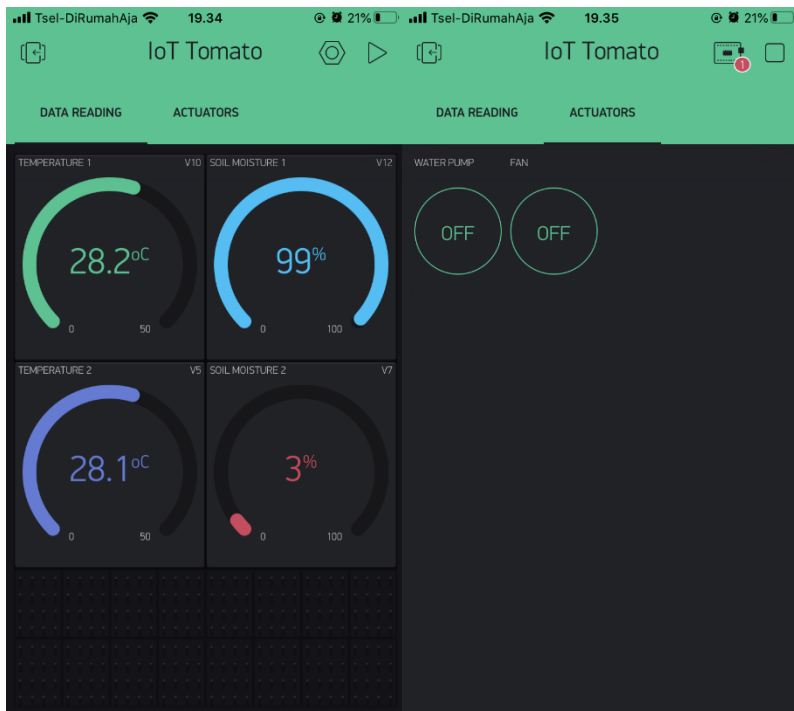
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "zVAcg0wF1rH1hAdMg0F_cS56vs0b66";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Ifancentury";
char pass[] = "qwertyui";
```

Gambar 3. 11 Pseudocode dari Modul IoT SiMonTa

### 3.4.2.4 Realisasi Tampilan Antarmuka

Hasil data yang ditangkap oleh sensor dan dikirim ke *cloud server* menggunakan ESP8266 dapat diamatin seperti gambar dibawah ini

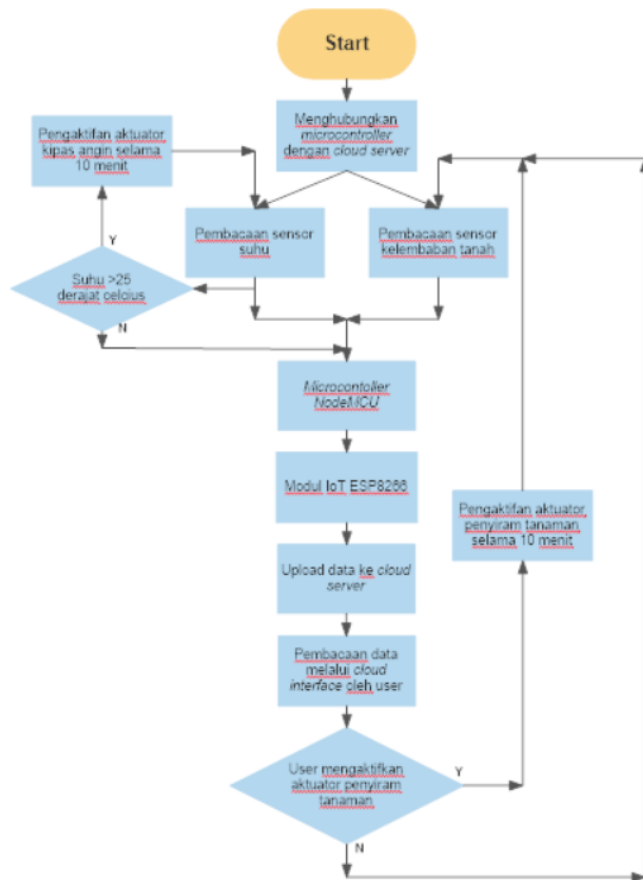


Gambar 3. 12 Tampilan Aplikasi Blynk

### 3.4.3 Integrasi Sistem

Setelah ditentukan perangkat keras beserta perangkat lunak untuk menunjang sistem *monitoring* tomat berbasis IoT, dilakukan integrasi antar komponen-komponen penunjang sistem.

Setelah terintegrasi maka dapat ditetapkan diagram alir sistem dari awal aktivasi sampai pengambilan data



Gambar 3. 13 Diagram Alir Sistem

### 3.4.4 Pengujian Sistem

Pengujian sistem dilakukan untuk membuktikan kemampuan sistem *monitoring* tomat berbasis IoT dalam menjalankan tugasnya sebagaimana tujuan mengapa sistem tersebut dibuat.

#### **3.4.4.1 Pengujian Fungsi Sensor**

Pengujian ini dilakukan untuk mengetahui apakah sensor dapat bekerja sesuai fungsinya dan mendapatkan data yang sesuai dengan kenyataan. Langkah pengujian dilakukan sebagai berikut :

1. Membuka *software* Arduino IDE
2. Menghubungkan NodeMCU dengan komputer/laptop yang berisi Arduino IDE
3. Meng-upload program ke NodeMCU menggunakan Arduino IDE
4. Setelah program berhasil di-upload, buka serial monitor di Arduino IDE
5. Amati hasil yang tertera pada serial monitor

#### **3.4.4.2 Pengujian Fungsi Aktuator**

Pengujian ini dilakukan untuk mengetahui apakah aktuator dapat bekerja sesuai fungsinya dan melakukan fungsi yang sesuai dengan kenyataan. Langkah pengujian dilakukan sebagai berikut:

1. Membuka *software* Arduino IDE
2. Menghubungkan NodeMCU dengan komputer/laptop yang berisi Arduino IDE
3. Meng-upload program ke NodeMCU menggunakan Arduino IDE
4. Setelah program berhasil di-upload, buka serial monitor di Arduino IDE
5. Buka aplikasi Blynk pada *smartphone*
6. Buka tab “Actuator”
7. Tekan tombol salah satu aktuator
8. Amati perubahan pada aktuator dan pada serial monitor

#### **3.4.4.3 Pengujian Fungsi IoT**

Pengujian ini dilakukan untuk mengetahui apakah mikrokontroler dapat mengirimdata ke *cloud server* secara tepat. Langkah pengujian dilakukan sebagai berikut:

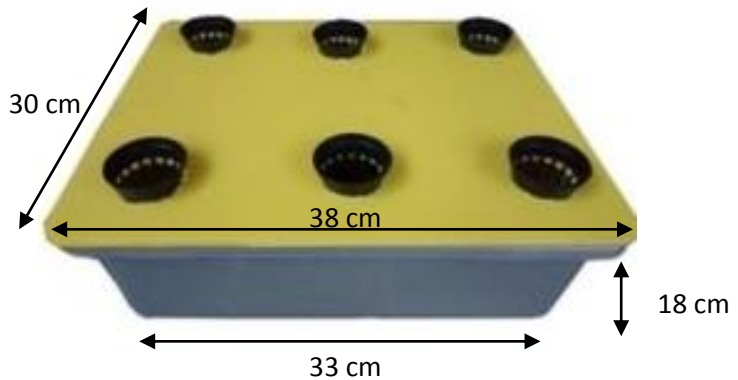
1. Menyalakan NodeMCU beserta sensor-sensor menggunakan catu daya
2. Buka aplikasi Blynk
3. Amati perubahan pada tab “Sensors”

### 3.5 Desain Lahan Tanaman Tomat

Berikut adalah desain sistem *monitoring*, sistem pengaturan lingkungan tanaman tomat dan pengeluaran biaya untuk realisasi sistem

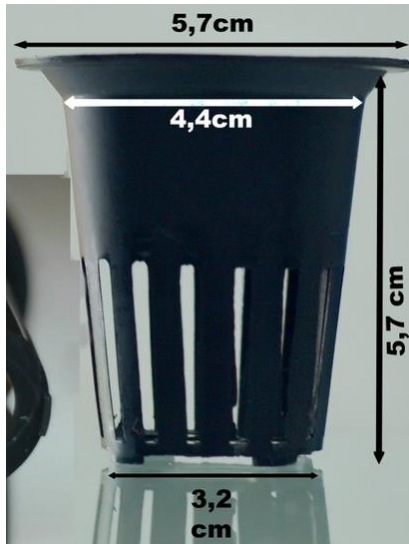
#### 3.5.1 Pembuatan Lahan Tomat Hidroponik

Pada tugas akhir ini, tanaman tomat akan dikembang biakan menggunakan cara hidroponik dengan sistem *wick*. Media tanam *rockwool* digunakan sebagai pengganti tanah.



**Gambar 3. 14** Lahan Tomat Hidroponik

Masing-masing *rockwool* diletakan kedalam netpot berdiameter 5,7 cm dan kedalaman 5,7 cm. Penyiraman tanaman dilakukan menggunakan kain flannel yang direndam dibawah netpot.



**Gambar 3. 14** Lahan Tomat Hidroponik

### **3.5.2 Pengaturan Lingkungan Tanaman Tomat**

Berdasarkan sifat tanaman tomat dalam pertumbuhannya yang telah dibahas pada sub-bab 2.2, maka sensor suhu, kelembaban tanah digunakan untuk mendeteksi suhu dan kelembaban tanah pada setiap netpot dan aktuator kipas pendingin dan penyiram tanaman untuk menurunkan suhu lingkungan agar tetap di kisaran 20-30 derajat celcius dan menjaga kelembaban *rockwool* di kisaran 50-60%.

### **3.5.3 Pengeluaran Biaya Sistem**

Setelah dijelaskan komponen-komponen yang akan dipakai dalam SiMonTa, berikut adalah total biaya yang dibutuhkan untuk merealisasikan perangkat keras untuk SiMonTa

**Tabel 3. 8** Biaya Perangkat Keras

No	Nama Perangkat	Harga Satuan (Rp)	Jumlah	Total (Rp)
1	NodeMCU ESP8266	48000	1	48000
2	DHT 22	42000	2	84000
3	YL-69	20000	1	20000
3	Kipas DC 5 V	23500	1	23500
4	Pompa Air Submersible 5V	12000	1	12000
5	Relay 2 Channel	21000	1	21000
6	Set Hidroponik Wick	105000	1	105000
Total				313500

### 3.6 Pengukuran Kualitas Jaringan (QoS)

Pengukuran QoS dilakukan untuk mengetahui kualitas jaringan pada sistem *monitoring* tomat berbasis IoT.

#### 3.6.1 Parameter QoS

Parameter kualitas jaringan yang diukur pada performansi SiMonTa nanti terdiri dari:

- a. *Throughput* : Laju data sistem
- b. *Delay* : Selisih waktu terima dengan waktu kirim
- c. *Jitter* : Variasi dari *delay*
- d. *Packet Loss* : Paket data yang hilang dalam transmisi

#### 3.6.2 Wireshark

Wireshark merupakan sebuah perangkat lunak yang memiliki fungsi untuk menganalisa jaringan. Pada perancangan sistem *monitoring* tomat berbasis IoT ini, Wireshark berfungsi untuk menampilkan statistik jaringan dalam waktu tertentu untuk kemudian diambil datanya sebagai pengukuran kualitas jaringan yang dipakai.

Wireshark dapat digunakan untuk mengukur parameter QoS yang dibahas pada sub-bab 3.6.1 karena memiliki fitur data analysis untuk melihat throughput. Wireshark juga memiliki filter seperti 'tcp.analysis.lost\_segment' untuk mencari packet loss dan filter 'tcp.analysis.ack\_rtt' untuk mencari delay atau *round-time-trip*.

### 3.6.3 Skenario Pengujian QoS

Jaringan yang digunakan pada koneksi antara SiMonTa dengan *cloud server* adalah jaringan internet. Kondisi standar SiMonTa adalah menggunakan 4 sensor, 2 sensor suhu dan 2 sensor kelembaban tanah. Kondisi pengiriman data standar SiMonTa adalah dengan jeda antar transmisi 1 menit. Skenario pengujian dari QoS dilakukan dalam beberapa kondisi yang berbeda. Berikut pembagian skenario pengujian QoS:

a. Berdasarkan Jumlah Sensor

Pengujian dilakukan pada SiMonTa berisi dua sensor dan empat sensor, dengan waktu pengamatan 60 menit dan jeda antar transmisi 1 menit. Pengujian ini dilakukan untuk mengetahui efek dari jumlah sensor terhadap kinerja jaringan

b. Berdasarkan Waktu Pengamatan

Pengujian dilakukan pada SiMonTa berisi empat sensor dengan waktu pengamatan 15 menit, 30 menit dan 60 menit, jeda antar transmisi 1 menit. Pengujian ini dilakukan untuk mengetahui apakah perbedaan waktu pengamatan berpengaruh terhadap kinerja jaringan

c. Berdasarkan Jeda Transmisi

Pengujian dilakukan pada SiMonTa berisi empat sensor dengan jeda antar transmisi 30 detik dan 1 menit, waktu pengamatan selama 1 jam. Pengujian ini dilakukan untuk mengetahui apakah ada efek dari jeda transmisi yang terlalu singkat atau terlalu lama terhadap kinerja jaringan

### 3.6.4 Langkah Percobaan QoS

Berikut adalah langkah untuk mengukur kualitas jaringan:



1. Jalankan sistem dengan menyalakan NodeMCU menggunakan catu daya
2. Buka *software* Wireshark
3. Pada bagian “Capture”, pilih “Local Area Connection”
4. Klik stop apabila pengamatan sudah selesai
5. Pada bagian “apply filter”, isi dengan “TCP”
6. Untuk menghitung *throughput*, buka tab “Statistics” kemudian pilih “Capture File Properties” kemudian amati “Average bits”
7. Untuk menghitung delay, pada bagian apply filter di isi dengan `tcp.analysis.ack_rtt`
8. Hasil pengamatan diexport kedalam format .csv, kemudian buka file menggunakan Microsoft Excel, kemudian delay dihitung dengan cara mengurangi selisih pada kolom time
9. Untuk menghitung jitter, didapat dari menghitung selisih dari hasil delay yang dihitung pada langkah ke-7
10. Untuk menghitung packet loss, pada *software* Wireshark, pada bagian apply filter isi dengan `tcp.analysis.lost_segment`.
11. Kemudian di bagian bawah terdapat informasi jumlah paket yang hilang.
12. Klik bagian “Statistics” kemudian pilih “Capture File Properties” untuk melihat jumlah paket yang terkirim dan diterima
13. Packet loss didapat dengan membagi antara paket yang hilang dengan jumlah paket yang terkirim

### **3.7 Standar Pengukuran Kualitas Jaringan (QoS)**

Standar yang digunakan adalah standar TIPHON dengan parameter yang terbagi dalam kriteria yang akan dijelaskan dalam sub-bab dibawah ini

#### **3.7.1 Kriteria Umum Standar TIPHON**

Standar yang digunakan adalah standar TIPHON dengan kriteria yang terbagi dalam indeks sebagai berikut:

**Tabel 3. 9** Persentase dan Nilai QoS

<b>Nilai</b>	<b>Persentase (%)</b>	<b>Skor(Indeks)</b>
3,8 – 4	95 – 100	Sangat Memuaskan
3 – 3,79	75 – 94,75	Memuaskan
2 – 2,99	50 – 74,75	Kurang Memuaskan
1 – 1,99	25 – 49,75	Jelek

### 3.7.2 Parameter Kualitas Jaringan

Dalam tugas akhir ini, parameter kualitas jaringan yang diukur terbagi menjadi tabel dibawah ini:

**Tabel 3. 10** Kategori Throughput

<b>Kategori</b>	<b>Throughput (bps)</b>	<b>Indeks</b>
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Jelek	<25	1

**Tabel 3. 11** Kategori Delay

<b>Kategori</b>	<b>Besar Delay (ms)</b>	<b>Indeks</b>
Sangat Bagus	<150	4
Bagus	150 s/d 300	3
Sedang	300 s/d 450	2

Jelek	>450	1
-------	------	---

**Tabel 3. 12** Kategori Jitter

<b>Kategori</b>	<b>Nilai Jitter (ms)</b>	<b>Indeks</b>
Sangat Bagus	0	4
Bagus	0 s/d 75	3
Sedang	75 s/d 125	2
Jelek	125 s/d 225	1

**Tabel 3. 13** Kategori Packet Loss

<b>Kategori</b>	<b>Packet Loss (%)</b>	<b>Indeks</b>
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1

## **BAB IV**

### **ANALISIS DATA DAN PEMBAHASAN**

Bagian ini membahas data hasil pengujian dan pengukuran beserta dengan analisa dari hasil pengujian dan pengukuran

#### **4.1 Implementasi Sistem**

Sub-bab ini menjelaskan hasil dari implementasi perangkat keras dan perangkat lunak pada objek tanaman tomat

##### **4.1.1 Implementasi Perangkat Keras**

Berikut adalah implementasi perangkat keras pada objek tanaman tomat.



**Gambar 4.1** Implementasi Perangkat Keras

##### **4.1.2 Implementasi Perangkat Lunak**

Berikut adalah implementasi perangkat lunak pada objek tanaman tomat

```
COM3
15:21:32.453 -> [14409] Connected to WiFi
15:21:32.453 -> [14409] IP: 172.20.10.2
15:21:32.453 -> [14409] Blynk v0.3.10 on NodeMCU
15:21:32.453 -> [14410] Connecting to blynk-cloud.com:8442
15:21:32.758 -> [14743] Ready (ping: 276ms).
```

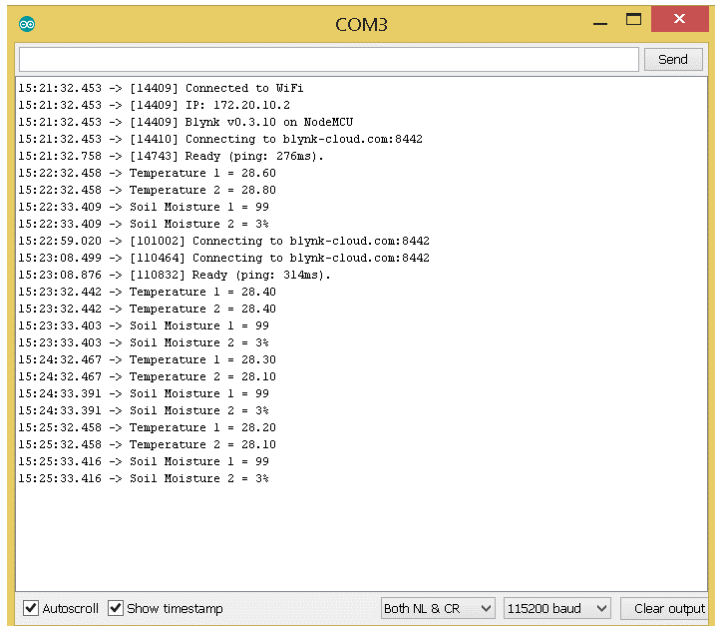
**Gambar 4.2** Implementasi Program pada NodeMCU

## 4.2 Hasil Pengujian Sistem

Pengujian sistem dibagi menjadi tiga bagian, yaitu pengujian sensor, pengujian aktuator dan pengujian komunikasi IoT. Berikut hasil dari masing-masing bagian

### 4.2.1 Hasil Uji Fungsi Sensor

Berdasarkan hasil pengujian, didapat pengamatan seperti pada gambar 4.1. Dimana data dari sensor suhu dan kelembaban tanah berhasil ditampilkan pada *Serial Monitor* Arudino IDE. Data yang muncul pada serial monitor juga dibandingkan dengan thermometer suhu lingkungan secara manual juga hygrometer tanah. Didapat data yang sama.



The image shows a screenshot of the Arduino IDE Serial Monitor window. The window title is "COM3". The output text is as follows:

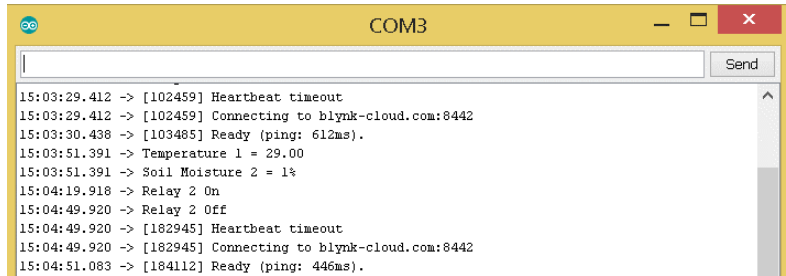
```
15:21:32.453 -> [14409] Connected to WiFi
15:21:32.453 -> [14409] IP: 172.20.10.2
15:21:32.453 -> [14409] Blynk v0.3.10 on NodeMCU
15:21:32.453 -> [14410] Connecting to blynk-cloud.com:8442
15:21:32.758 -> [14743] Ready (ping: 276ms).
15:22:32.458 -> Temperature 1 = 28.60
15:22:32.458 -> Temperature 2 = 28.80
15:22:33.409 -> Soil Moisture 1 = 99
15:22:33.409 -> Soil Moisture 2 = 3%
15:22:59.020 -> [101002] Connecting to blynk-cloud.com:8442
15:23:08.499 -> [110464] Connecting to blynk-cloud.com:8442
15:23:08.876 -> [110832] Ready (ping: 314ms).
15:23:32.442 -> Temperature 1 = 28.40
15:23:32.442 -> Temperature 2 = 28.40
15:23:33.403 -> Soil Moisture 1 = 99
15:23:33.403 -> Soil Moisture 2 = 3%
15:24:32.467 -> Temperature 1 = 28.30
15:24:32.467 -> Temperature 2 = 28.10
15:24:33.391 -> Soil Moisture 1 = 99
15:24:33.391 -> Soil Moisture 2 = 3%
15:25:32.458 -> Temperature 1 = 28.20
15:25:32.458 -> Temperature 2 = 28.10
15:25:33.416 -> Soil Moisture 1 = 99
15:25:33.416 -> Soil Moisture 2 = 3%
```

At the bottom of the window, there are several controls:  Autoscroll,  Show timestamp, a dropdown menu set to "Both NL & CR", a dropdown menu set to "115200 baud", and a "Clear output" button.

Gambar 4.3 Pengujian Fungsi Sensor

### 4.2.2 Hasil Uji Fungsi Aktuator

Berdasarkan hasil pengujian, didapat pengamatan seperti pada gambar 4.2. Dimana perintah dari aplikasi Blynk berhasil diterima oleh mikrokontroler dan diteruskan ke modul relay sebagai aktuator

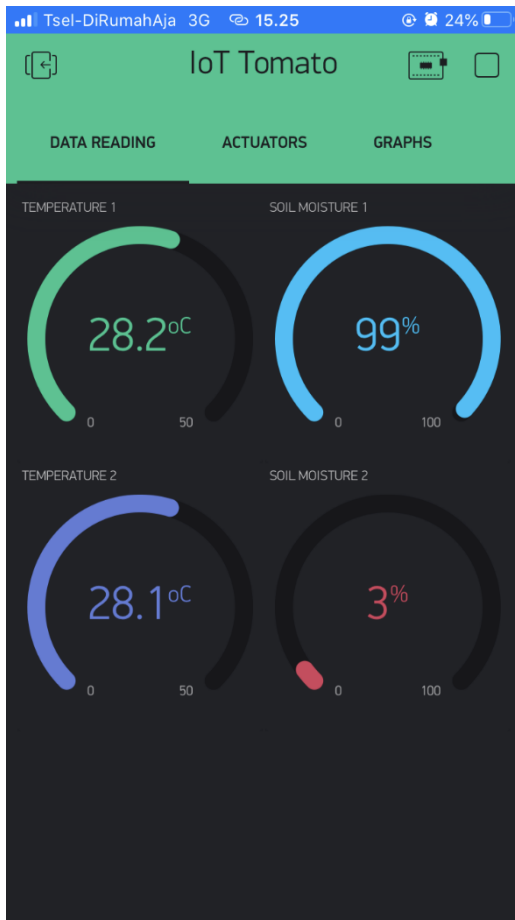


```
15:03:29.412 -> [102459] Heartbeat timeout
15:03:29.412 -> [102459] Connecting to blynk-cloud.com:8442
15:03:30.438 -> [103485] Ready (ping: 612ms).
15:03:51.391 -> Temperature 1 = 29.00
15:03:51.391 -> Soil Moisture 2 = 1%
15:04:19.918 -> Relay 2 On
15:04:49.920 -> Relay 2 Off
15:04:49.920 -> [182945] Heartbeat timeout
15:04:49.920 -> [182945] Connecting to blynk-cloud.com:8442
15:04:51.083 -> [184112] Ready (ping: 446ms).
```

**Gambar 4.4** Pengujian Fungsi Aktuator

### 4.2.3 Hasil Uji Fungsi IoT

Berdasarkan hasil pengujian, didapat pengamatan seperti pada Gambar 4.3. Dimana data yang di tangkap dapat ditransmisikan ke *cloud server* dan ditampilkan di aplikasi Blynk. Dapat diamati bahwa waktu pengukuran pada Gambar 4.1 sama dengan waktu pada Gambar 4.3, menandakan bahwa data yang ditampilkan sama.



**Gambar 4.5** Tampilan Blynk Empat Sensor

### 4.3 Hasil Pengukuran Kualitas Jaringan (QoS)

Pengukuran kualitas jaringan dibagi menjadi beberapa kondisi yang berbeda. Yang pertama berdasarkan jumlah sensor yang dipakai. Kedua berdasarkan waktu pengamatan. Ketiga berdasarkan jeda antar transmisi

#### 4.3.1 Hasil Perbandingan Jumlah Sensor

Pada sistem *monitoring* tomat menggunakan jumlah sensor yang berbeda, diukur parameter QoS seperti *throughput*, *delay*, *jitter* dan *packet loss*. Berikut hasil dari masing-masing parameter

**Tabel 4. 1** Nilai QoS Berdasarkan Jumlah Sensor

Parameter	Jumlah Sensor	
	Sistem Dua Sensor	Sistem Empat Sensor
Throughput (bps)	197	204
Delay (ms)	109	114
Jitter (ms)	0,238	0,233
Packet Loss (%)	0	0,72

#### 4.3.2 Hasil Perbandingan Waktu Pengamatan

Pada sistem *monitoring* tomat menggunakan masing-masing dua sensor suhu dan sensor kelembaban tanah, dengan waktu pengamatan yang berbeda, diukur parameter QoS seperti *throughput*, *delay*, *jitter* dan *packet loss*. Berikut hasil dari masing-masing parameter



**Tabel 4. 2** Nilai QoS Berdasarkan Waktu Pengamatan

Parameter	Waktu Pengamatan		
	Pengamatan 60 Menit	Pengamatan 30 Menit	Pengamatan 15 Menit
Throughput (bps)	204	183	265
Delay (ms)	114	138	117
Jitter (ms)	0,233	0.727	0.629
Packet Loss (%)	0,72	0	0.2

#### 4.3.3 Hasil Perbandingan Jeda Waktu Transmisi

Pada sistem *monitoring* tomat menggunakan masing-masing dua sensor suhu dan sensor kelembaban tanah, dengan jeda transmisi yang berbeda, diukur parameter QoS seperti *throughput*, *delay*, *jitter* dan *packet loss*. Berikut hasil dari masing-masing parameter

**Tabel 4. 3** Nilai QoS Berdasarkan Jeda Transmisi

Parameter	Jeda Transmisi	
	Jeda 30 Detik	Jeda 60 Detik
Throughput (bps)	251	204
Delay (ms)	166	114
Jitter (ms)	0,211	0,233
Packet Loss (%)	0	0,72

#### 4.4 Analisa dan Pembahasan Sistem

Dalam perakitan perangkat keras, komponen yang digunakan sesuai dengan harapan. Perawatan tanaman tomat pun cukup terbantu karena dapat dilakukan pengguna kapan saja dan dimana saja. Namun kendala terjadi pada pin mikrokontroler yang terbatas, menyebabkan tidak seluruh tanaman dapat diukur kelembaban tanahnya.

Pembuatan perangkat lunak untuk mengendalikan mikrokontroler juga dilakukan dengan lancar. Program untuk mengambil data dari sensor-sensor yang berkaitan pun cukup sederhana sehingga program keseluruhan tidak kompleks. Namun pada saat aktuator menyala, pengiriman data terhenti karena aktuator bekerja dengan rentang waktu *delay* sehingga menunda proses secara serial.

Berdasarkan hasil pengujian dari sistem yang terdiri dari pengujian sensor, aktuator dan komunikasi IoT, dapat diamati bahwa tiap bagian dari sistem dapat bekerja sesuai dengan fungsinya masing-masing. Sebagaimana tiap bagian dari sistem memenuhi parameter dari pengujian yang telah ditetapkan. Sensor dapat menangkap data secara konstan selama 60 menit waktu pengamatan dengan jarak antar transmisi selama satu menit. Aktuator juga merespon perintah yang diberikan dari aplikasi Blynk dengan aksi yang sesuai. Data yang diamati pada aplikasi Blynk juga sesuai dengan kondisi yang nyata di lapangan.

## 4.5 Perbandingan Performansi Sistem

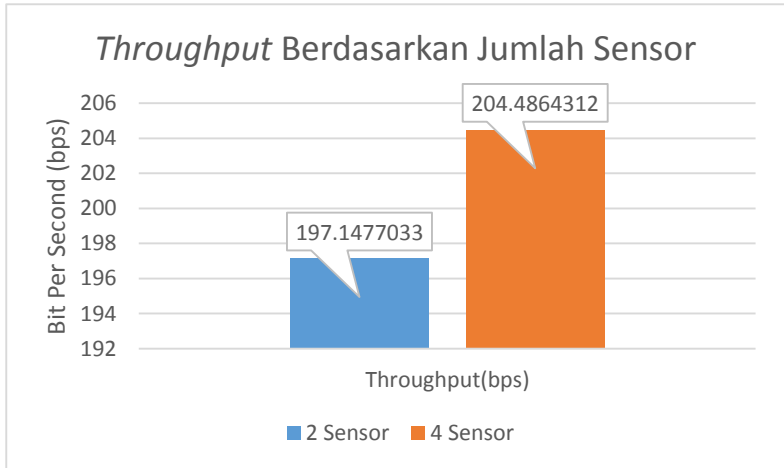
Berdasarkan hasil yang diperoleh pada sub-bab 4.3, dapat dirangkum berdasarkan masing-masing parameter

### 4.5.1 Hasil Pengukuran *Throughput*

Perhitungan laju data sistem atau *throughput* dilakukan pada tiga skenario yang berbeda :

#### 4.5.1.1 *Throughput* Perbandingan Jumlah Sensor

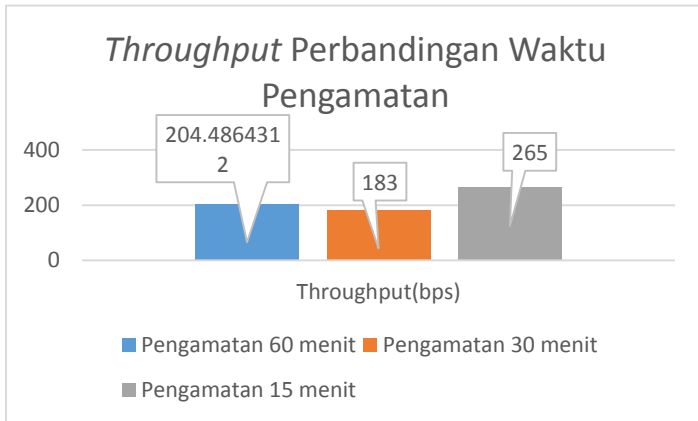
Jumlah sensor yang lebih banyak menyebabkan peningkatan pada laju data per detik. Peningkatan laju data terjadi karena jumlah paket data yang dikirim setiap menitnya pada SiMonTa empat sensor lebih besar dibandingkan SiMonTa dua sensor.



**Gambar 4.6** Grafik *Throughput* Berdasarkan Jumlah Sensor

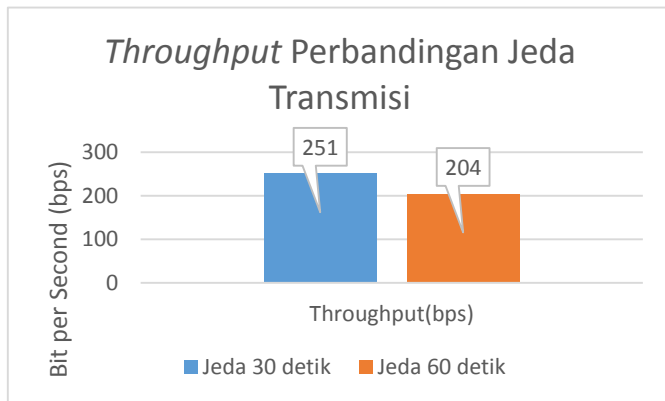
#### 4.5.1.2 *Throughput* Perbandingan Waktu Pengamatan

Terdapat peningkatan laju data pada pengujian dengan waktu pengamatan 15 menit. Peningkatan laju data pada pengujian dengan waktu pengamatan 15 menit terjadi karena penulis mengirimkan lebih banyak perintah untuk menyalakan aktuator ke SiMonTa melalui aplikasi Blynk dibandingkan dengan pengamatan yang lainnya.



**Gambar 4.7** Grafik *Throughput* Berdasarkan Waktu Pengamatan

#### 4.5.1.3 *Throughput* Perbandingan Jeda Transmisi



**Gambar 4.8** Grafik *Throughput* Berdasarkan Jeda Transmisi

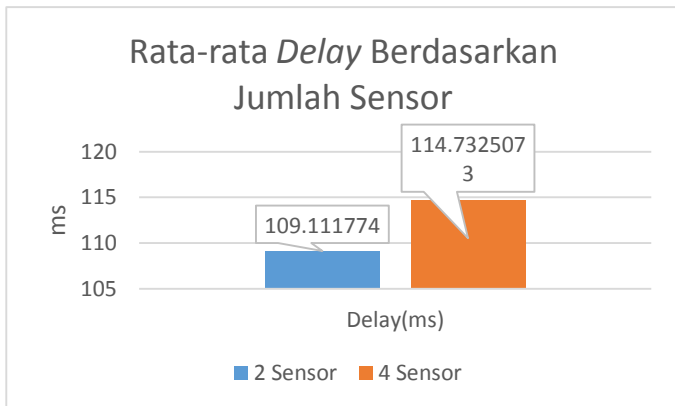
Terdapat peningkatan laju data pada SiMonTa dengan jeda transmisi 30 detik. Hal tersebut terjadi karena paket data yang dikirim setiap menitnya lebih banyak dibandingkan SiMonTa dengan jeda transmisi 60 detik.

#### 4.5.2 Hasil Pengukuran Delay

Perhitungan selisih waktu transmisi atau *round-time-trip* (RTT) atau *delay* dari SiMonTa terbagi menjadi tiga skenario yang berbeda. Berikut hasil dari pengukuran RTT tiap skenario

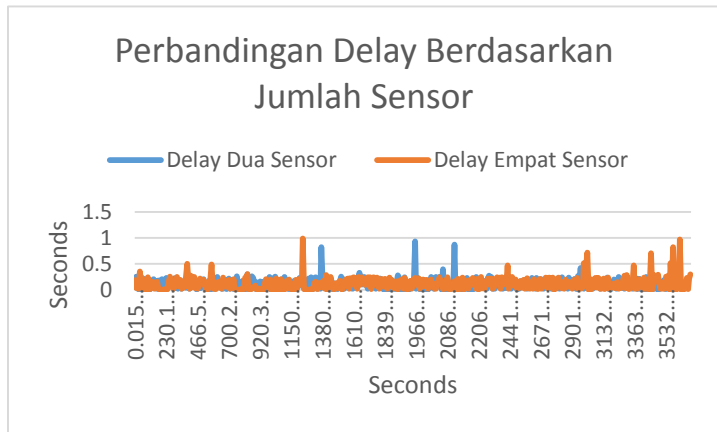
##### 4.5.2.1 Delay Perbandingan Jumlah Sensor

Delay atau *round-trip-time* pada SiMonTa berisi empat sensor sedikit lebih besar dibandingkan SiMonTa berisi dua sensor. Namun peningkatan *delay* pada SiMonTa 4 sensor tidak berpengaruh secara signifikan pada keseluruhan kualitas layanan sistem.



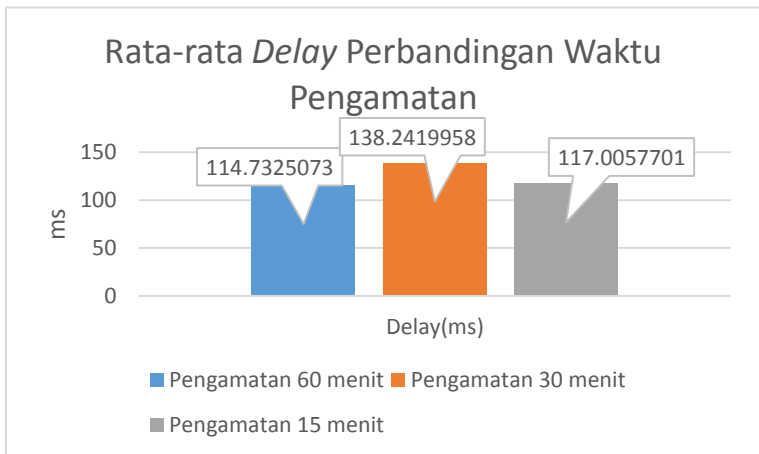
**Gambar 4.9** Rata-rata Delay Berdasarkan Jumlah Sensor

Pada Gambar 4.10, terlihat di beberapa waktu terdapat lonjakan nilai delay hampir mencapai satu detik antar transmisinya. Hal tersebut disebabkan mikrokontroler melakukan transmisi ulang karena paket data yang rusak atau *error* sebelumnya.



**Gambar 4.10** Nilai Delay Pada Perbandingan Jumlah Sensor

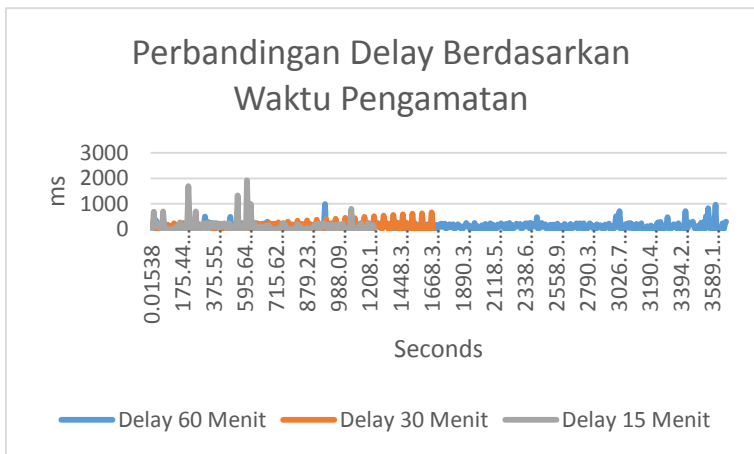
#### 4.5.2.2 Delay Perbandingan Waktu Pengamatan



**Gambar 4.11** Rata-rata Delay Berdasarkan Waktu Pengamatan

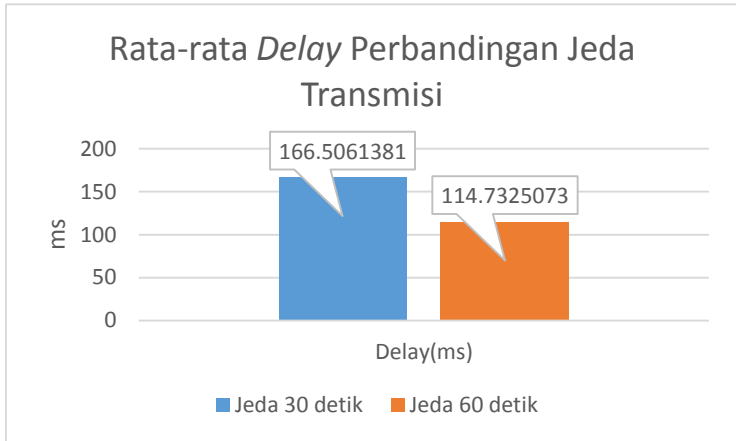
Nilai *delay* pada ketiga sistem tidak mengalami perubahan yang signifikan. Nilai *delay* rata-rata pada pengamatan 30 menit memiliki nilai yang paling besar diantara pengamatan lainnya. Hal tersebut disebabkan transmisi ulang dari mikrokontroler ke *cloud server* yang membuat waktu terima menjadi lebih lama. Ketiga skenario dilakukan dengan perlakuan yang sama

Pada Gambar 4.12, terlihat di beberapa waktu terdapat lonjakan nilai *delay* hampir mencapai satu detik antar transmisinya. Hal tersebut disebabkan mikrokontroler melakukan transmisi ulang karena paket data yang rusak atau *error* sebelumnya.

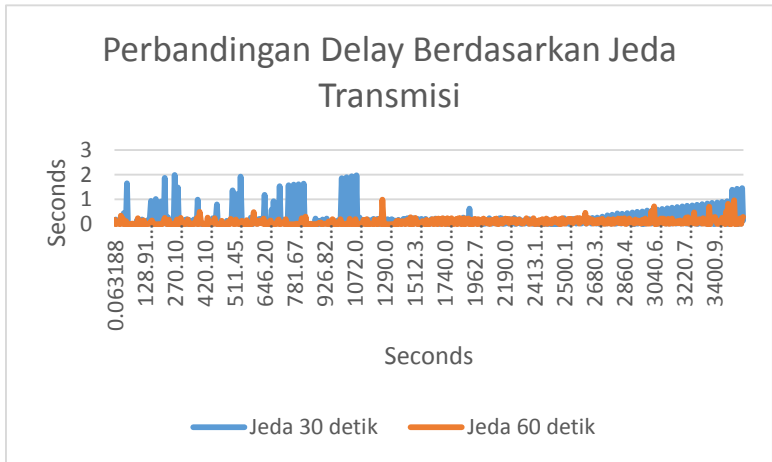


**Gambar 4.12** Nilai *Delay* Berdasarkan Waktu Pengamatan

### 4.5.2.3 Delay Perbandingan Jeda Transmisi



**Gambar 4.13** Rata-rata Delay Berdasarkan Jeda Transmisi



**Gambar 4.14** Nilai Delay Berdasarkan Jeda Transmisi

Gambar 4.13 menunjukkan adanya kenaikan nilai rata-rata delay pada SiMonTa dengan jeda transmisi 30 detik. Hal tersebut terjadi karena paket data yang dikirim setiap menitnya lebih banyak dibandingkan SiMonTa dengan jeda transmisi 60 detik.



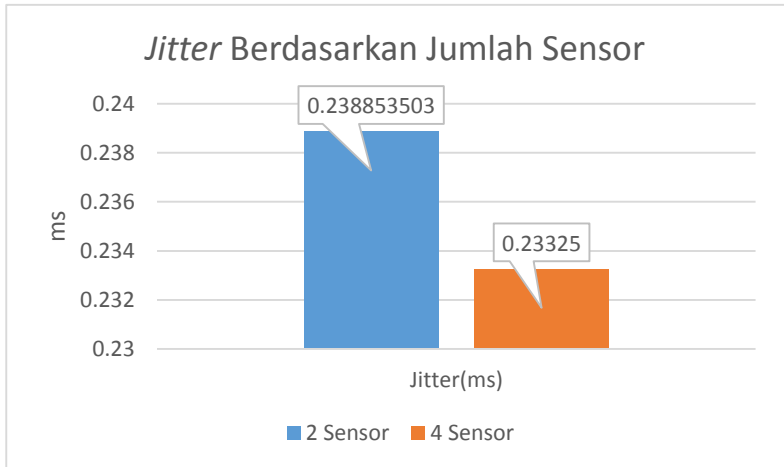
Gambar 4.14 menunjukkan beberapa kenaikan *delay* di beberapa waktu yang cukup signifikan. Hal tersebut disebabkan mikrokontroler melakukan transmisi ulang karena ada paket data yang rusak atau *error*.

#### 4.5.3 Hasil Pengukuran *Jitter*

Berikut adalah hasil pengukuran *jitter* dari setiap skenario pengukuran yang berbeda

##### 4.5.3.1 *Jitter* Perbandingan Jumlah Sensor

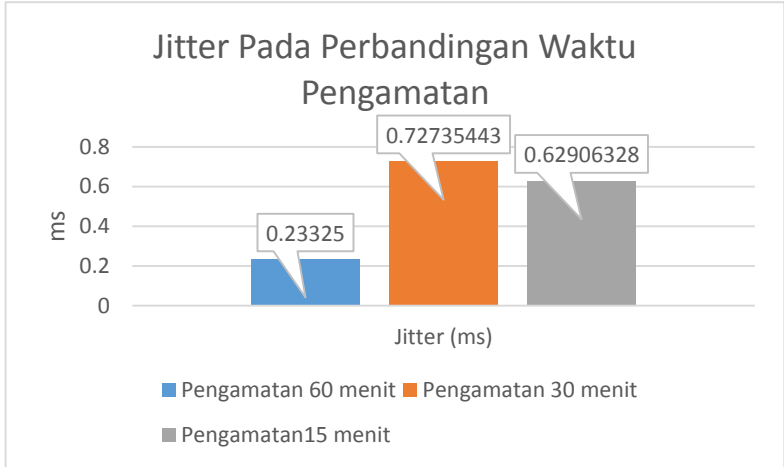
Kedua sistem memiliki nilai *jitter* yang tidak jauh berbeda, berkaca dari hasil pengukuran delay pada sub-bab 4.5.2.1.



**Gambar 4.15** Nilai *Jitter* Berdasarkan Jumlah Sensor

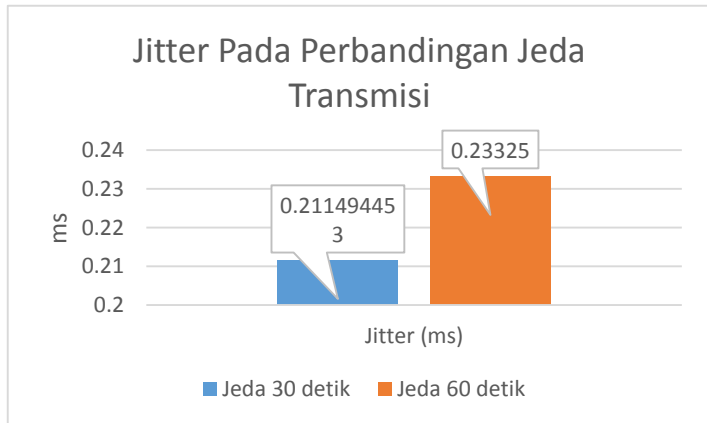
#### 4.5.3.2 *Jitter* Perbandingan Waktu Pengamatan

SiMonTa dengan pengamatan 30 menit memiliki nilai jitter yang paling besar dibandingkan dua pengamatan lainnya. Hal ini disebabkan nilai delay pada pengamatan 30 menit mengalami peningkatan yang tidak kecil.



Gambar 4.16 Nilai *Jitter* Berdasarkan Waktu Pengamatan

#### 4.5.3.3 *Jitter* Perbandingan Waktu Pengamatan



Gambar 4.17 Nilai *Jitter* Berdasarkan Jeda Transmisi

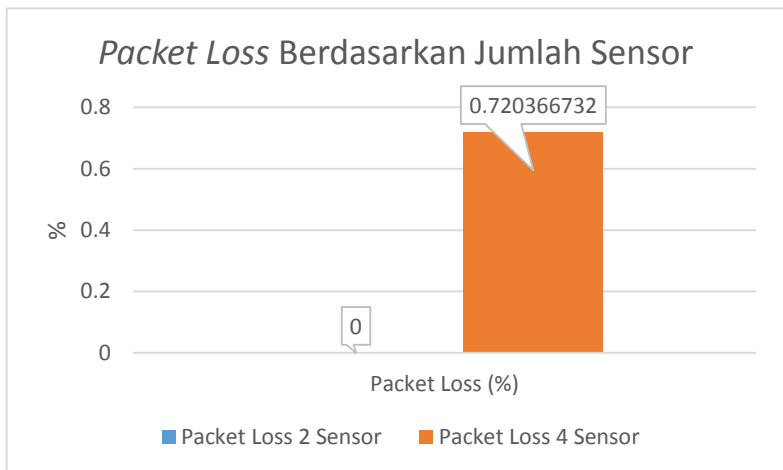
Gambar 4.16 menunjukkan nilai *jitter* yang tidak jauh berbeda antara kedua sistem walaupun nilai delay pada SiMonTa dengan jeda transmisi 30 detik lebih besar. Hal ini disebabkan paket data yang ditransmisikan pada SiMonTa jeda transmisi 30 detik lebih banyak sehingga pembagi dari perhitungan *jitter* juga menjadi lebih besar

#### 4.5.4 Hasil Pengukuran *Packet Loss*

Berikut adalah hasil pengukuran *packet loss* dari setiap skenario pengukuran yang berbeda

##### 4.5.4.1 *Packet Loss* Perbandingan Jumlah Sensor

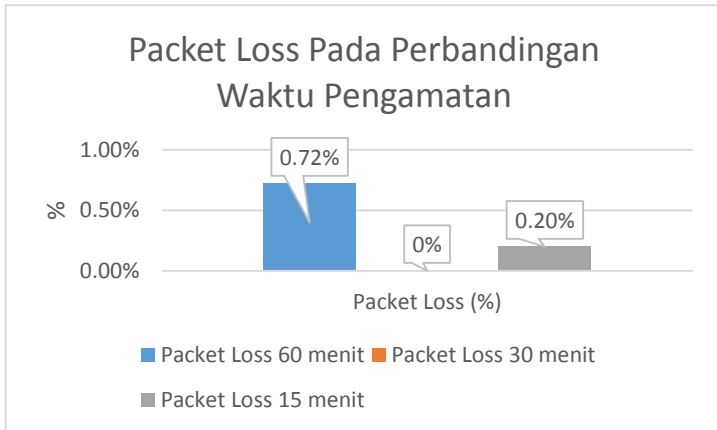
Pada SiMonTa dengan 4 sensor, terdapat paket yang hilang sebesar 0,72% dari total paket yang dikirimkan. Hal ini dapat terjadi karena gangguan jaringan antara modul Wi-Fi dengan *cloud server*.



**Gambar 4.18** Nilai *Packet Loss* Berdasarkan Jumlah Sensor

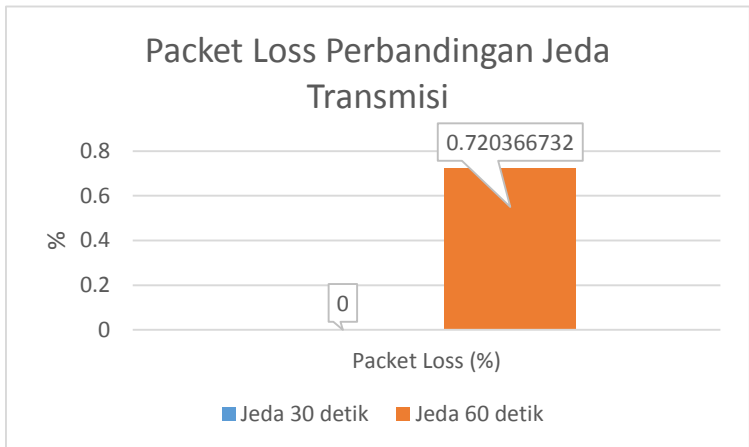
#### 4.5.4.2 Packet Loss Perbandingan Waktu Pengamatan

Sama seperti sub-sub-bab sebelumnya, namun terdapat pula paket yang hilang dalam transmisi data SiMonTa dengan waktu pengamatan 15 menit. Hal ini dapat terjadi karena gangguan jaringan antara modul Wi-Fi dengan *cloud server*.



Gambar 4.19 Nilai *Packet Loss* Berdasarkan Waktu Pengamatan

#### 4.5.4.3 Packet Loss Perbandingan Jeda Transmisi



Gambar 4.20 Nilai *Packet Loss* Berdasarkan Jeda Transmisi

Gambar 4.19 menunjukkan tidak adanya paket yang hilang pada SiMonTa dengan jeda transmisi 30 detik. Hal ini dapat terjadi karena transmisi ulang yang sukses pada SiMonTa jeda transmisi 30 detik.

#### 4.6 Klasifikasi Standar QoS (TIPHON)

Berdasarkan hasil yang diperoleh pada sub-bab 4.3, setiap perbandingan SiMonTa dapat dikategorikan sebagai berikut:

- a. Berdasarkan Jumlah Sensor

**Tabel 4. 4** Nilai Indeks TIPHON Setiap SistemSensor

Parameter	Sistem Dua Sensor		Sistem Empat Sensor	
	Nilai Parameter	Indeks	Nilai Parameter	Indeks
Throughput (bps)	197 bps	4	204 bps	4
Delay (ms)	109 ms	4	114 ms	4
Jitter (ms)	0,238 ms	3	0,233 ms	3
Packet Loss (%)	0%	4	0,72%	4
<b>Rata-Rata Indeks</b>	<b>3,75</b>		<b>3,75</b>	

Berdasarkan nilai indeks dari standar TIPHON, SiMonTa dengan kedua perbandingan diatas masuk ke dalam kategori ‘Memuaskan’

b. Berdasarkan Waktu Pengamatan

**Tabel 4. 5** Nilai Indeks TIPHON Setiap Waktu Pengamatan

Parameter	Pengamatan 60 Menit		Pengamatan 30 Menit		Pengamatan 15 Menit	
	Nilai Parameter	Indeks	Nilai Parameter	Indeks	Nilai Parameter	Indeks
Throughput (bps)	204 bps	4	183 bps	4	265 bps	4
Delay (ms)	114 ms	4	138 ms	4	117 ms	4
Jitter (ms)	0,233 ms	3	0,727 ms	3	0,629 ms	3
Packet Loss (%)	0,72%	4	0%	4	0,2%	4
<b>Rata-Rata Indeks</b>	<b>3,75</b>		<b>3,75</b>		<b>3,75</b>	

Berdasarkan nilai indeks dari standar TIPHON, SiMonTa dengan ketiga perbandingan diatas masuk ke dalam kategori ‘Memuaskan’

c. Berdasarkan Jeda Transmisi

**Tabel 4. 6** Nilai Indeks TIPHON Setiap Sistem Jeda Transmisi

Parameter	Sistem Jeda 30 detik		Sistem Jeda 60 detik	
	Nilai Parameter	Indeks	Nilai Parameter	Indeks
Throughput (bps)	251 bps	4	204 bps	4
Delay (ms)	166 ms	3	114 ms	4
Jitter (ms)	0,211 ms	3	0,233 ms	3
Packet Loss (%)	0%	4	0,72%	4
<b>Rata-Rata Indeks</b>	<b>3,5</b>		<b>3,75</b>	

Berdasarkan nilai indeks dari standar TIPHON, SiMonTa dengan kedua perbandingan diatas masuk ke dalam kategori ‘Memuaskan’

## **4.7 Informasi Tambahan SiMonTa**

Sub-bab ini menjelaskan cara menggunakan SiMonTa untuk melakukan budidaya dan perawatan tanaman tomat menggunakan teknologi IoT

### **4.7.1 Cara Penggunaan SiMonTa**

Berikut adalah *user manual* untuk menggunakan aplikasi SiMonTa

- a. Sistem dimulai dengan menghubungkan mikrokontroler ke catu daya.

- b. Setelah mikrokontroler terhubung dengan catu daya, maka sensor langsung aktif mengambil data dan aktuator siap melakukan kerja bila diberi perintah.
- c. Buka aplikasi Blynk pada *smartphone*
- d. Klik tombol ‘Play’ pada pojok kanan atas layar untuk mulai menghubungkan *smartphone* dengan mikrokontroler
- e. Klik tab ‘Data Reading’ untuk melihat kondisi lingkungan tanaman tomat
- f. Klik tab ‘Actuators’ untuk melihat tombol untuk mengaktifkan aktuator
- g. Klik tombol pada ‘Water Pump’ untuk mengaktifkan pompa air
- h. Klik tombol pada ‘Fan’ untuk mengaktifkan kipas pendingin

#### **4.7.2 Implementasi SiMonTa**

Berdasarkan hasil pengamatan dan dibandingkan dengan siklus tomat, maka penggunaan SiMonTa paling baik digunakan pada saat pertumbuhan tomat mulai minggu ketiga sampai panen, karena pada waktu tersebut pertumbuhan tomat memerlukan lingkungan yang stabil yaitu suhu sekitar 20-30 dan kelembaban tanah 50-60%. Penerapan SiMonTa juga bisa dilakukan setelah tanaman tomat sudah panen untuk pertama kali. Dengan menjaga kestabilan lingkungan, tanaman tomat bisa berbuah atau panen dua sampai tiga kali.



*-Halaman ini sengaja dikosongkan-*

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Setelah dilakukan pengkajian, pengujian, pengukuran dan pembahasan dari hasil yang diperoleh terkait dengan judul tugas akhir “Rancang Bangun Rumah Kaca Pintar Berbasis IoT (*Internet of Things*) Untuk Budidaya Tanaman Tomat”, dapat diambil beberapa kesimpulan sebagai berikut

1. Sistem *Monitoring* Tomat Berbasis IoT dapat berjalan dengan baik, diukur dari hasil sensor dan aktuator yang sesuai dengan keadaan lapangan.
2. Nilai *Throughput* terbesar berada pada SiMonTa dengan jeda transmisi 60 detik, waktu pengamatan 15 menit.
3. Semakin kecil jeda antar transmisi, maka sistem mengalami delay transmisi yang lebih besar.
4. Nilai *Jitter* terbesar berada pada SiMonTa dengan jeda transmisi 60 menit dan waktu pengamatan 30 menit.
5. SiMonTa yang mengalami *packet loss* hanya SiMonTa 4 sensor dengan jeda transmisi 60 detik waktu pengamatan 15 menit dan 60 menit.

#### **5.2 Saran**

Dari hasil pengujian, pengukuran dan juga pembahasan dari Sistem *Monitoring* Tomat Berbasis IoT, penulis dapat memberikan beberapa saran sebagai berikut

1. Mengatur jeda antar transmisi sekitar 1-5 menit agar kestabilan sistem terjaga.

*-Halaman ini sengaja dikosongkan-*

## DAFTAR PUSTAKA

- [1] Y. Sastro, "PERTANIAN PERKOTAAN : Peluang, Tantangan, dan Strategi Pengembangan," *Buletin Pertanian Perkotaan*, vol. 3, no. 1, pp. 29-36, 2013.
- [2] B. Salehi, R. Sharifi-Rad, F. Sharopov, J. Namiesnik, A. Roointan, M. Kamle, P. Kumar, N. Martins and J. Sharifi-Rad, "Beneficial Effects and Potential Risks Of Tomatoes Consumption For Human Health : An Overview," *Nutrition*, no. S0899-9007(18)31257-7, pp. 1-42, 2019.
- [3] K. K. Patel, S. M. Patel, P. G. Scholar and C. Salazar, "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges," vol. 6, no. 5, pp. 6122-6131, 2016.
- [4] Y. Guven, E. Coşgun, S. Kocaoğlu and H. Gezici, "Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications," *Research Inventy: International Journal of Engineering And Science*, vol. 6, no. 9, pp. 38-44, 2017.
- [5] A. Kaur, "AN OVERVIEW OF QUALITY OF SERVICE COMPUTER NETWORK," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 3, pp. 470-475, 2011.
- [6] D. Serpanos and M. Wolf, *Internet-of-Things (IoT) Systems : Architectures, Algorithms, Methodologies*, Atlanta: Springer International Publisher, 2018.
- [7] M. A. Zamora-Izquierdo, J. Santa, J. A. Martinez, V. Martinez and A. F. Skarmeta, "Smart farming IoT platform based on edge and cloud computing," *Biosystems Engineering*, pp. 4-17, 2019.

- [8] I. Marsic, *Wireless Networks : Local and Ad Hoc Networks*, New Jersey: Rutgers University, 2005, pp. 3-8.
- [9] G. Gridling and B. Weiss, *Introduction to Microcontroller*, Vienna: Styrian Press, 2007, pp. 1-5.
- [10] R. Srinisivan, *Safer tomato production methods: A field guide for soil fertility and pest management*, Vols. 10-740, Taiwan: AVRDC Publication, 2010.

## LAMPIRAN

### LAMPIRAN A : Program SiMonTa

```
#define BLYNK_PRINT Serial
#include <SPI.h>;
#include <ESP8266WiFi.h>;
#include <BlynkSimpleEsp8266.h>;
#include <SimpleTimer.h>;
#include <DHT.h>;
SimpleTimer timer;

char auth[] = "zVdcgO3wf1rMIIhAdMgUF_cS56vs0b6G";

char ssid[] = "Irfancentury";
char pass[] = "qwertyui";

#define DHTPIN1 4 // D2
#define DHTPIN2 12 //D6
#define soilMoisterPin 0 //D3
#define sensor_pin 17 //A0
const int Relay_1 = 5; // Pompa ke D1
const int Relay_2 = 14; // Fan ke D5
#define DHTTYPE DHT22 // DHT 22, AM2302, AM2321
DHT dht1(DHTPIN1, DHTTYPE);
DHT dht2(DHTPIN2, DHTTYPE);

void sendSensor()
{
  float h = dht1.readHumidity();
  float t1 = dht1.readTemperature(); // or dht.readTemperature(true) for
  Fahrenheit
  float t2 = dht2.readTemperature();
```

```

Serial.print("Temperature 1 = ");
Serial.println(t1);

Serial.print("Temperature 2 = ");
Serial.println(t2);

if (isnan(h) || isnan(t1) || isnan(t2)) {
  Serial.println("Failed to read from DHT sensor!");
}
int soilMoister = 0;
//digitalWrite (soilMoisterVcc, HIGH);
delay (500);
int N = 3;
for(int i = 0; i < N; i++) // read sensor "N" times and get the average
{
  soilMoister += digitalRead(soilMoisterPin);
  delay(150);
}
//digitalWrite (soilMoisterVcc, LOW);
soilMoister = soilMoister/N;
soilMoister = map(soilMoister, 600, 0, 0, 100); //LM393 on 5V (+Vin)
Serial.print("Soil Moisture 1 = ");
Serial.println(soilMoister);

int MoistureValue = 0;
int Moisturepercent = 0;
MoistureValue = analogRead(sensor_pin);
Moisturepercent = map(MoistureValue,735,285,0,100);
if(Moisturepercent > 100)
{
  Serial.println("Soil Moisture 2 = 100%");
}
else if(Moisturepercent < 0)
{
  Serial.println("Soil Moisture 2 = 0%");
}

```

```

else if(Moisturepercent > 0 && Moisturepercent < 100)
{
Serial.print("Soil Moisture 2 = ");
Serial.print(Moisturepercent);
Serial.println("%");
}
Blynk.virtualWrite(V5, t2); // Humidity for gauge
Blynk.virtualWrite(V10, t1); // Temperature for gauge
Blynk.virtualWrite(V12, soilMoister); // virtual pin V12
Blynk.virtualWrite(V7, Moisturepercent); // Humidity for graph
Blynk.virtualWrite(V8, t1); // Temperature for graph

    if (t1 && t2 > 25)
{
    Blynk.notify("Warning ==>> Fan ON");
    digitalWrite(Relay_2, LOW);
    Serial.println("Relay 2 On");
    delay(10000);
    Blynk.notify("Warning ==>> Fan Off");
    digitalWrite(Relay_2, HIGH);
    Serial.println("Relay 2 Off");
}
}

void setup()
{
    pinMode(Relay_1, OUTPUT);
    pinMode(Relay_2, OUTPUT);
    digitalWrite(Relay_1, HIGH);
    digitalWrite(Relay_2, HIGH);
    Serial.begin(115200); // See the connection status in Serial Monitor
    Blynk.begin(auth, ssid, pass);

    dht1.begin();
    dht2.begin();
    // Setup a function to be called every second

```



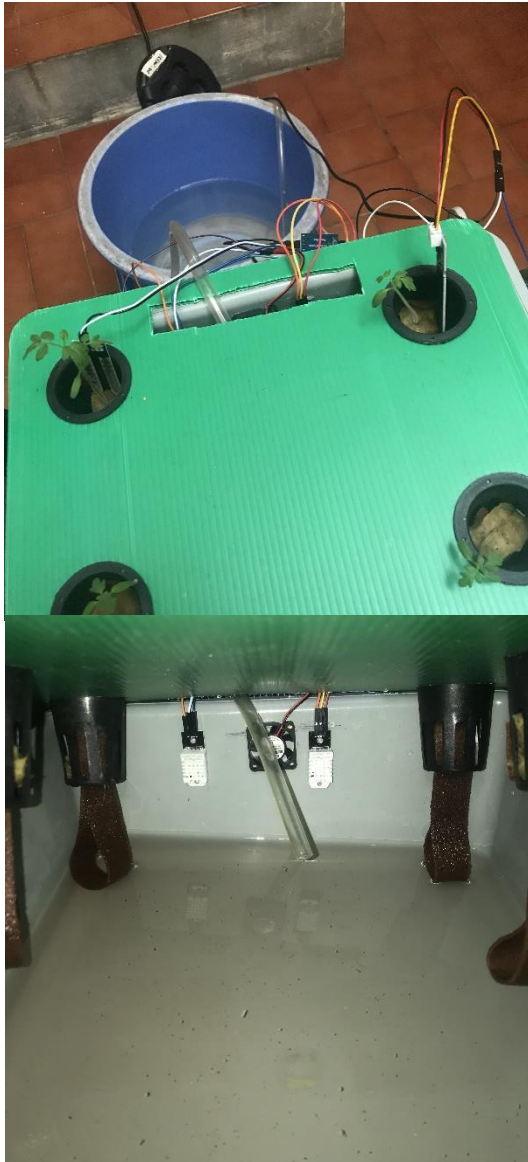
```

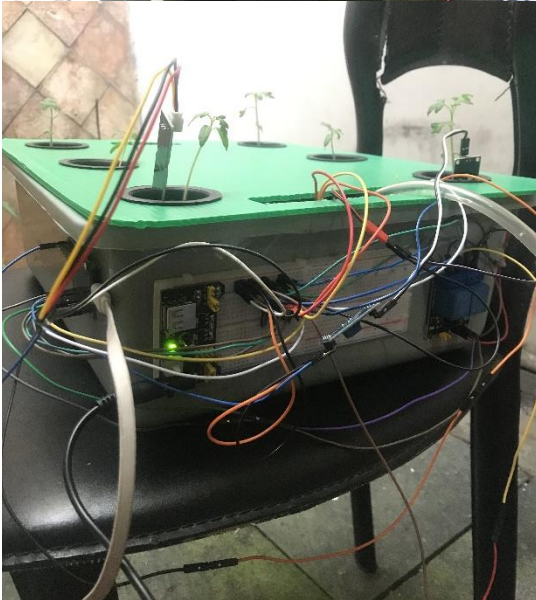
    timer.setInterval(30000L, sendSensor);
}

void loop()
{
    Blynk.run(); // Initiates Blynk
    timer.run(); // Initiates SimpleTimer
}
BLYNK_WRITE(V3)
{
    int pinValue = param.asInt();
    if (pinValue == 1)
    {
        Blynk.notify("Warning ==>> Pump ON");
        digitalWrite(Relay_1, LOW);
        Serial.println("Relay 1 On");
        delay(10000);
        Blynk.notify("Warning ==>> Pump ON");
        digitalWrite(Relay_1, HIGH);
        Serial.println("Relay 1 Off");
    }
}
BLYNK_WRITE(V4)
{
    int pinValue = param.asInt();
    if (pinValue == 1)
    {
        Blynk.notify("Warning ==>> Fan ON");
        digitalWrite(Relay_2, LOW);
        Serial.println("Relay 2 On");
        delay(10000);
        Blynk.notify("Warning ==>> Fan ON");
        digitalWrite(Relay_2, HIGH);
        Serial.println("Relay 2 Off");
    }
}

```

**LAMPIRAN B : Implementasi SiMonTa Pada Tanaman Tomat**





## LAMPIRAN C : PROPOSAL TUGAS AKHIR

Departemen Teknik Elektro  
Fakultas Teknologi Elektro – ITS

### EE 184801 TUGAS AKHIR – 6 SKS

Nama Mahasiswa : Muhammad Irfan Abriyantoro  
Nomer Pokok : 07111640000151  
Bidang Studi : Telekomunikasi Multimedia  
Tugas Diberikan : Semester Genap Tahun Ajaran 2019/2020  
Dosen Pembimbing : 1. Sri Rahayu, S.T., M.Kom.  
2. -

Judul Tugas Akhir : **Rancang Bangun Prototipe Rumah Kaca Pintar Berbasis IoT (Internet of Things) Untuk Budidaya Tanaman Tomat (IoT-based Smart Greenhouse Prototype Design for Tomato Cultivation)**

#### Uraian Tugas Akhir :

Di era modern ini, makin banyak masalah yang dihadapi oleh masyarakat yang tinggal di perkotaan(urban). Aktivitas sehari-hari yang sangat padat dan serba cepat, namun tidak diimbangi dengan kemampuan mobilisasi yang memadai sering menjadi pemicu munculnya gangguan kesehatan akibat dari tekanan psikis yang tinggi dan buruknya kualitas udara yang penuh polutan. Tak heran banyak bermunculan komunitas yang mengkampanyekan hal-hal positif seperti gemar berolahraga, cinta lingkungan, asupan makanan sehat, dsb. Bahkan banyak mengonsumsi Vitamin-C juga telah menjadi gaya hidup. Buah tomat sering menjadi sasaran konsumsi masyarakat urban untuk mendapatkan Vitamin-C alami yang murah. Selain Vitamin-C, tomat juga kaya betakaroten dan likopen sebagai antioksidan yang menjawab kebutuhan khas masyarakat urban.

Dalam Tugas Akhir ini, akan dibuat rancangan proto budidaya tanaman tomat yang dapat diimplementasikan oleh masyarakat kota yang terbatas luas lahan dan waktu pengelolannya. Rancangan ini menggabungkan konsep *greenhouse* dan teknologi berbasis IoT, sehingga mensyaratkan adanya sejumlah *embedded-sensor*, jaringan internet, perangkat lunak dan divais elektronik lainnya. Adapun untuk merealisasikan rancangan tersebut, akan di gunakan sensor suhu, sensor kelembaban tanah & sensor hujan untuk menangkap data kondisi lingkungan dalam *greenhouse*. Selain itu juga didukung oleh aktuator, mikrokontroler, modul komunikasi dan *cloud platform* sehingga membentuk sistem IoT yang lengkap. Sedangkan untuk mengetahui kinerja sistem yang dibuat, akan dilakukan pengujian apakah fungsi-fungsi semua elemen bisa bekerja dengan baik dan apakah tanaman tomat bisa tumbuh lebih baik dibanding tanaman tomat yang dikelola secara konvensional. Dengan rancangan yang diusulkan ini diharapkan dapat memberi manfaat berupa kegiatan positif yang bernilai kreatif bagi masyarakat urban dan sekaligus bisa menikmati hasil panennya.

Dosen Pembimbing

Sri Rahayu, S.T., M.Kom.  
NIP : 19680228199702200

Mengetahui,  
Ketua Program Studi S1

Menyetujui,  
Kepala Lab. Jaringan Telekomunikasi

Dedet C. Riawan, S.T., M.Eng., Ph.D.  
NIP : 197311192000031001

Dr. Ir. Achmad Affandi, DEA.  
NIP : 196510141990021001

## RIWAYAT HIDUP



Muhammad Irfan Abriyantoro dilahirkan di Jakarta pada tanggal 15 September 1998. Penulis merupakan anak tunggal dari pasangan Abrijendo dan Dyah Trianjayani. Penulis memulai Pendidikan formal di TK Besuki pada tahun 2002-2004, SDS Adik Irma pada tahun 2004 - 2010, SMP Labschool Kebayoran pada tahun 2010 – 2013 dan SMAN 8 Jakarta pada tahun 2013 - 2016 setelah kelulusan SMA, penulis melanjutkan studi S-1 Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Penulis berhasil diterima di Institut Teknologi Sepuluh Nopember . Pada tahun ketiga, penulis memilih bidang studi Telekomunikasi Multimedia. Penulis menulis buku tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana pada wisuda periode September 2020. Penulis dapat dihubungi melalui alamat *email* : irfan.abriantoro@gmail.com

